Departamento de Ingeniería de Sistemas Telemáticos
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DOCTORAL

**CONTRIBUTION TO DYNAMIC RISK
MANAGEMENT AUTOMATION BY AN ONTOLOGY-
BASED FRAMEWORK**

RAÚL RIESCO GRANADINO
TELECOMMUNICATION ENGINEER
2019

Departamento de Ingeniería de Sistemas Telemáticos
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DOCTORAL

**CONTRIBUTION TO DYNAMIC RISK
MANAGEMENT AUTOMATION BY AN
ONTOLOGY-BASED FRAMEWORK**

Author:

Raúl Riesco Granadino
Telecommunications Engineer

Director:

Victor A. Villagrà González
Doctor of Telecommunications Engineering

Madrid, October 2019

Title: Contribution to dynamic risk management automation by an ontology-based framework
Department: Departamento de Ingeniería de Sistemas Telemáticos
Author: Raúl Riesco Granadino
Director: Victor A. Villagrà González

COMMITTEE:

Chairman:	D. Julio Berrocal Colmenarejo	Doctor Ingeniero de Telecomunicación Catedrático de Universidad Universidad Politécnica de Madrid
Members:	D. Gregorio Martínez Pérez	Doctor en Informática Catedrático de Universidad Universidad de Murcia
	D. Fco. Javier López Muñoz	Doctor en Informática Catedrático de Universidad Universidad de Málaga
	D. Enrique Belda Esplugues	Doctor Ingeniero de Caminos, Canales y Puertos Universidad Politécnica de Valencia Subdirector General de Sistemas de Información y Comunicaciones para la Seguridad del Ministerio del Interior
Secretary:	Dña. Lourdes López Santidrián	Doctora en Informática Catedrática Escuela Universitaria Universidad Politécnica de Madrid
Substitutes:	D. Luis Javier García Villalba	Doctor en Informática Profesor Contratado Doctor Universidad Complutense de Madrid
	D. Eduardo García Pardo	Doctor en Informática Profesor Ayudante Doctor Universidad Rey Juan Carlos

Done the act of defense and reading of the Ph.D. Thesis in Madrid, on _____ the _____ of 2019, the committee agrees to grant the following qualification:

President

Members

Secretary

ACKNOWLEDGEMENTS

Thanks to my Thesis Director, Victor, for your coaching and support during these five years. Despite it has been a long way walking alone, you have given the needed support when it was really needed.

I want to thank my family, especially my wife, my oasis, for her tremendous support at all times. She has demonstrated that nothing is impossible and that we are a real team, thank you Diana for the whole life together and for being the light I need.

Our children, Javi and Claudia have also contributed so far, especially to keep me as creative as a kid.

My parents, my brother and sisters, as well as my parents in law, my sisters in law and nephews who also have trust me unconditionally.

At the Spanish National Cybersecurity Institute (INCIBE), I want to especially thank my colleague Blanca, which is a truly professional in charge of the ISMS of the entity. It was an honor to learn from her. I also want to especially thank my colleague Guillermo for the very interesting conversations related to the thesis insights. I also want to thank Nacho, Antonio, Raúl, Tatiana, Juan and the whole INTEL team, who also trust me along this (hard) way.

Last but not least, I want to thank my best friends Jorge, Raquel, Paco, Angélica, Gonzalo, Nuria and Jorgito, as they really believe in me.

ABSTRACT

One of the most important goals in an organization is to have risks under an acceptance level along the time. All organizations are exposed to real time security threats that could have an impact on their risk exposure levels harming the entire organization, their customers and their reputation. New emerging techniques, tactics and procedures (TTP) which remain undetected, the complexity and decentralization of organization assets, the great number of vulnerabilities proportional to the number of new type of devices (IoT) or still the high number of false positives, are only some examples of real risks for any organization. Risk management frameworks are not integrated and automated with Near Real Time (NRT) risk-related Cybersecurity Threat Intelligence (CTI) information. To enable such a dynamic, NRT and more realistic risk assessment and management processes, we created a new semantic version of STIX™ v2.0 for Cyber Threat Intelligence as it is becoming a de facto standard for Structured Threat Information Exchange. We selected an international leading organization in Cybersecurity to demonstrate new dynamic ways to support decision making at all levels while being under attack. Semantic reasoners could be our ideal partners to fight against threats having risks under control along the time, for that, they need to understand the data. Our proposal uses an unprecedented mix of standards to cover all levels of a DRM and ensure easier adoption by users.

At the same time, although cyber threat intelligence (CTI) exchange is a must for any organization due to the fact that no one can fight alone against all threats, the potential participants are often reluctant to share their CTI and prefer to consume only, at least in voluntary based approaches. Such behavior destroys the idea of information exchange. On the other hand, governments are forcing specific entities and operators to report them specific incidents depending on their impact, otherwise there could be sanctions to those operators which are not reporting them on time. Obligations and sanctions are usually discouraging participants to share information voluntarily which will just share and report what is strictly required. We propose a paradigm shift of cybersecurity information exchange by introducing a new way to encourage all participants involved, at all levels, to share relevant information dynamically within our DRM Framework. It will also contribute to the support and deployment of Dynamic Risk Management (DRM) frameworks along all our peers to share advanced intelligence, in the format of algorithms, beyond the exchange of Indicators of Compromise (IoC). Participants will have new and specific incentives to share, invest and consume threat intelligence and risk intelligence information depending on their different roles (producers, consumers, investors, donors and owner).

Our proposal leverages from standards like Structured Threat Information Exchange (STIX™), as well as W3C semantic web standards to enable a workspace of knowledge related to behavioral threat intelligence patterning to characterize tactics, techniques and procedures (TTP). At the same time, we propose the use of the Ethereum Blockchain to better incentivize the sharing of that knowledge between all parties involved as well as the creation of a standard CTI token as a digital asset with a promising value in the market. An experimentation was also performed to demonstrate its benefits and incentives but also its potential limits with regard to storage and cost of transactions.

The contribution of this paper is a Dynamic Risk Assessment and Management (DRA / DRM) framework based on ontologies. It includes an integrated, layered and networked architecture based on the Web Ontology Language (OWL), STIX™, a semantic reasoner, the use of semantic web rule language (SWRL) and the Ethereum Blockchain to approach an all-in-one solution at all levels (operational, tactic and strategic). It implements a hybrid Cyber Threat Intelligence and DRM Ontology as well as behavioral algorithms in the format of SWRL rules to infer new knowledge by the reasoner. As the dynamics is provided by the use of Intelligence Sharing, a paradigm shift based on the Ethereum Blockchain is also provided, to overcome all known issues of information sharing today,

included the exchange of algorithms or rules beyond the exchange of Indicators of Compromise (IoC).

Keywords- Cybersecurity, Dynamic Risk Assessment (DRA), Dynamic Risk Management (DRM), Cyber Threat Intelligence (CTI), Risk Intelligence, Semantic Web, Ontologies, OWL, SWRL, Countermeasures, Safeguards, Intelligence Sharing, Threats, Vulnerabilities, STIX™, TAXII™, Internet of Everything (IoE), Internet of Things (IoT), Indicators of Compromise (IoC), Ethereum Blockchain, Smart contract, Tokens.

RESUMEN

Una de las metas más importantes en una organización es mantener el nivel de riesgo en un nivel aceptable y asumible a lo largo del tiempo. Todas las organizaciones están expuestas a amenazas de seguridad en tiempo real que podrían tener un impacto en sus niveles de exposición al riesgo y dañar a la organización, a sus clientes y a su reputación. Las nuevas técnicas, tácticas y procedimientos (TTP, en inglés Techniques, Tactics and Procedures) emergentes que permanecen sin ser detectadas, la complejidad y la descentralización de los activos de la organización, la gran cantidad de vulnerabilidades proporcionales a la cantidad de nuevos tipos de dispositivos conectados (IoT) o la gran cantidad de falsos positivos, son sólo algunos ejemplos de los riesgos reales a los que se expone diariamente cualquier organización. Los sistemas de gestión de la seguridad de la información no están automatizados ni integrados con la información en tiempo real (o NRT – Near Real Time - próximo al tiempo real) de inteligencia de amenazas de ciberseguridad lo cual no permite tener una visión realista y dinámica del nivel de riesgo de nuestra organización. Para permitir procesos de gestión y evaluación de riesgos tan dinámicos, NRT y más realistas, se desarrolla en la presente tesis doctoral una nueva ontología o versión semántica del lenguaje STIX™ v2.0 (en inglés, Structure Threat Information Exchange) para la Inteligencia de Ciber-amenazas, ya que STIX™ se está convirtiendo en un estándar de facto para el intercambio de información de amenazas estructuradas aunque carece de la semántica necesaria en su versión actual. Se ha seleccionado el sistema de gestión de seguridad de la información de una organización líder internacional en Ciberseguridad para demostrar cómo esta nueva aproximación permitirá dinámicamente una mejor toma de decisiones informadas a todos los niveles, como por ejemplo, en situaciones bajo un ciber-ataque complejo. Se propone además el uso de un razonador semántico, como socio ideal o aliado para luchar contra las amenazas dinámicamente, para ello, el razonador necesitará comprender los datos, cuestión que se garantiza mediante el uso de ontologías. La propuesta de la presente tesis doctoral utiliza una combinación de estándares para cubrir todos los niveles de un Sistema de Gestión Dinámica de Riesgos (en inglés, DRM – Dynamic Risk Management) de tal manera que ello facilite la adopción por parte de los usuarios respecto de sistemas no estándares o propietarios.

Al mismo tiempo, el intercambio de inteligencia sobre amenazas (CTI) se considera imprescindible para cualquier organización debido al hecho de que nadie puede luchar solo contra todas las amenazas, a pesar de ello, los participantes a menudo son reacios a compartir su inteligencia y prefieren ser únicamente consumidores de la información. Tal comportamiento destruye la idea y esencia del intercambio voluntario de información. Por otro lado, los gobiernos están obligando a entidades y operadores específicos a reportar incidentes concretos dependiendo de su impacto, de lo contrario, podría haber sanciones para aquellos operadores que no informen a tiempo. Las obligaciones y sanciones generalmente desalientan a los participantes a compartir información voluntariamente, con lo cual se presume que acabarán compartiendo únicamente aquello que sea estrictamente necesario. Proponemos un cambio de paradigma en el intercambio de información de inteligencia de amenazas mediante la introducción de una nueva forma de incentivar a todos los participantes involucrados, a todos los niveles, a compartir información relevante dinámicamente dentro de nuestro marco o modelo de referencia en Gestión Dinámica de Riesgos (en inglés, DRM Framework). Este nuevo modelo de incentivos contribuirá al despliegue de este modelo de Gestión de Riesgos Dinámicos a lo largo de la comunidad de compartición de inteligencia, cuya información podría evolucionar al concepto de inteligencia avanzada, esto es, a intercambiar conocimiento real en forma de algoritmos (e.j. algoritmo de detección de comportamiento TTP) más allá del intercambio de indicadores de compromiso (IoC). Los participantes del sistema tendrán nuevos incentivos específicos para compartir, invertir y consumir información de inteligencia de amenazas e

inteligencia de riesgos dependiendo de sus diferentes roles (productores, consumidores, inversores, donantes y propietarios).

La propuesta, aprovecha una variedad de estándares como el intercambio de información de amenazas estructuradas (STIX™), así como los estándares de la web semántica del W3C para evolucionar a un espacio de conocimiento relacionado con la definición de conceptos más complejos y que requieren de mayor expresividad en la actualidad, como los patrones de inteligencia de amenazas y de comportamiento. Al mismo tiempo, se propone el uso de la cadena de bloques de Ethereum (en inglés, Ethereum Blockchain), ya que ésta provee de funcionalidades que permiten aportar este nuevo modelo de incentivos de manera garantizada y sin intervención humana. Por ejemplo mediante el uso de un nuevo token CTI estándar, que representará un nuevo activo digital cuyo valor será equivalente al valor que la información de inteligencia tenga en el mercado en cada momento. El valor de esta información se considera en crecimiento por la importancia que cada vez más, ésta información tiene para las organizaciones, sobre todo si llega a tiempo para evitar la materialización de determinados riesgos. Asimismo, la presente tesis doctoral incluye la realización de experimentos tanto para demostrar los beneficios de este nuevo modelo de intercambio, como sus carencias, especialmente con respecto al almacenamiento y el coste.

La contribución principal de esta tesis doctoral es un marco de Evaluación y Gestión de Riesgos Dinámicos (DRA / DRM) basado en ontologías. Incluye una arquitectura integrada, en capas y en red basada en el lenguaje de ontologías web (OWL), STIX™, un razonador semántico, el uso de lenguaje de reglas de la web semántica (SWRL) y la cadena de bloques de Ethereum para abordar una solución completa a los problemas principales y a todos los niveles (técnico, táctico y estratégico). Implementa una ontología híbrida de riesgos (DRM) y amenazas (CTI), así como algoritmos de comportamiento en formato de reglas SWRL para inferir nuevos conocimientos avanzados o inteligencia por parte del razonador. Como el dinamismo es proporcionado principalmente por el intercambio de información de inteligencia, se proporciona un nuevo modelo de intercambio basado en la cadena de bloques de Ethereum, ya que de no ser así, muchos de los problemas principales conocidos hoy en día seguirían sin ser resueltos, especialmente el problema del bajo nivel de incentivos para contar con mayor número de entidades que comparten información. El modelo propuesto incluye además como contribución, la posibilidad de intercambiar conocimiento en formato de algoritmos o reglas más allá del intercambio de Indicadores de Compromiso (IoC). De esta manera los receptores de la información podrán detectar por sí mismos estos comportamientos y no esperar a recibir únicamente los IoC detectados por terceros. Las reglas pueden incluso ser reglas de inferencia en gestión de riesgos o reglas de enriquecimiento de la información, entre muchas otras.

Palabras clave- Ciberseguridad, Evaluación dinámica de riesgos, Gestión dinámica de riesgos, Inteligencia de ciber-amenazas, Inteligencia de riesgo, Web semántica, Ontologías, OWL, SWRL, Contramedidas, Salvaguardas, Intercambio de inteligencia, Amenazas, Vulnerabilidades, STIX™, TAXII™, Internet de todo, Internet de las cosas, Indicadores de compromiso, Cadena de bloques de Ethereum, Contrato inteligente, Tokens.

*To my wife, Diana, my oasis, my support and my daily light that guides my way.
To my children, Javi and Claudia, who help me to keep creative.
And to my parents, who trust me blindly.*

With love.

TABLE OF CONTENTS

CHAPTER 1. INTRODUCTION	23
1.1. MOTIVATION.....	23
1.2. CONTRIBUTIONS	27
1.3. STRUCTURE OF THE MEMORY	30
CHAPTER 2. RISK MANAGEMENT	31
2.1. INTRODUCTION.....	31
2.2. THE ISMS FRAMEWORK	34
2.2.1 ISMS Objective	34
2.2.2 Scope.....	34
2.2.3 Steps to develop an ISMS.....	35
2.2.4 Recurring Process.....	35
2.3. RISK ASSESSMENT AND MANAGEMENT	36
2.3.1 Risk Management Process	37
2.3.2 Establishing the Context, Scope and Criteria	38
2.3.3 Risk Assessment.....	38
2.3.4 Risk Management / Treatment.....	40
2.3.5 Conclusion	41
2.4. STANDARDS AND METHODOLOGIES	41
2.4.1 ISO 27001.....	41
2.4.2 ISO 31000.....	43
2.4.3 OCTAVE	45
2.4.4 CRAMM	45
2.4.5 MAGERIT.....	46
2.4.6 MEHARI.....	46
2.4.7 NIST800-30.....	46
2.4.8 ITIL 4.0.....	46
2.4.9 CMMI 2.0	46
2.4.10 COBIT 5.0	46
2.5. EXISTING METHODS	46
2.6. RISK MANAGEMENT RESEARCH CHALLENGES	47
2.6.1 Risk Aggregation.....	48
2.6.2 Evolution	48
2.6.3 Legal Aspects and Business Rules	49
2.6.4 Cost.....	49
2.6.5 Requirements Engineering.....	49
2.6.6 Risk Ownership	49
2.6.7 Methods to Reduce and Manage Systems Complexity.....	50
2.6.8 Dynamic Risk Assessment and Management.....	50
2.6.9 Formal Interoperable Models to enable Comparisons and Compatibility between Multi-Disciplinary Environments.....	50
2.6.10 Statistical and Predictive Risk Analysis	50
2.6.11 Autonomous Detection and Remediation by a Man-Machine Effective Cooperation	51
2.6.12 Integrated Risk Metrics and Indicators.....	51
2.6.13 Visual Decision Making Governance Frameworks.....	51
2.6.14 New Incentives for Adoption of Risk Management Best Practices and Reducing Barriers (Especially for SME)	52
2.6.15 Catalyst for Improving Impact of Research	52
2.6.16 Expected Outcome.....	52
2.6.17 Timeline	52
2.7. CONCLUSIONS.....	53
CHAPTER 3. CYBER THREAT INTELLIGENCE	54
3.1. INTRODUCTION.....	54
3.2. CURRENT APPROACHES	56
3.2.1 Indicators of Compromise (IoC)	56

3.2.2 OPENIOC	57
3.2.3 IODEF	57
3.2.4 STIX™	57
3.3. PYRAMID OF PAIN	58
3.4. STIX™	58
3.4.1 Use Cases.....	59
3.4.2 Analyzing Cyber Threats.....	60
3.4.3 Guiding Principles	61
3.4.4 Architecture.....	63
3.4.5 STIX™ V1.1 Structure.....	63
3.4.6 STIX™ V1.1 Implementations	70
3.4.7 STIX™ V2 Structure.....	70
3.4.8 STIX™ V1.1 and V2 Comparison.....	95
3.5. EXCHANGE OF THREAT INTELLIGENCE INFORMATION	97
3.5.1 The Importance of Information Exchange	98
3.5.2 OSINT: Open Source Intelligence.....	99
3.5.3 ISAC: Information Sharing and Analysis Centers.....	100
3.5.4 Current Status of Information Sharing.....	101
3.5.5 TAXII™: A Oasis Proposal for Information Exchange.....	102
3.5.6 Research Challenges of Information Sharing.....	107
3.5.7 Interesting Information to Share per Level	109
3.6. CONCLUSIONS.....	110
CHAPTER 4. ONTOLOGIES AND SWRL TO REPRESENT KNOWLEDGE.....	112
4.1. INTRODUCTION.....	112
4.2. ONTOLOGY DEFINITION LANGUAGES.....	113
4.2.1 SHOE.....	115
4.2.2 RDF and RDF-Schema	115
4.2.3 OIL and DAML+OIL	117
4.2.4 OWL.....	117
4.2.5 OWL2.....	119
4.2.6 Discussion.....	121
4.3. BEHAVIORAL LANGUAGES SPECIFICATION	122
4.3.1 RULEML	123
4.3.2 SWRL.....	123
4.3.3 OWL-Services.....	128
4.3.4 Other Behavioral Languages' Specifications.....	129
4.3.5 Discussion.....	131
4.4. SEMANTIC REASONERS.....	133
4.4.1 Bossam	135
4.4.2 Pellet.....	136
4.4.3 Other Semantic Reasoners.....	139
4.4.4 Discussion.....	141
4.5. CONCLUSIONS.....	143
CHAPTER 5. GLOBAL ARCHITECTURE PROPOSED.....	144
5.1. INTRODUCTION.....	144
5.2. METHODOLOGY.....	147
5.3. EVALUATION OF ALTERNATIVES	148
5.4. RELATED WORK	152
5.5. APPROACH AND RESULTS.....	159
5.6. CONTRIBUTIONS	160
5.6.1 Architecture and Formal Model for CTI and Dynamic Risk Management.....	160
5.6.2 An Enhanced Exchange of Knowledge.....	162
5.6.3 A Pure Decentralized Infrastructure Based On Blockchain for Information Exchange ..	165
5.7. CORE COMPONENTS	167

5.7.1 (Dynamic) CTI and Risk Data-Set	169
5.7.2 (Dynamic) Risk and CTI Intelligence Knowledge	169
5.7.3 Dynamic Risk Management	169
5.7.4 Dynamic Risk and CTI Intelligence Information Sharing Module.....	173
5.7.5 Reasoner	174
5.7.6 Decision Making Level	174
5.8. EXTERNAL COMPONENTS	175
5.8.1 N-Dimensional Sensoring System.....	175
5.8.2 Knowledge Database	178
5.8.3 Visualization and Simulation Engine (Business Intelligence)	178
5.8.4 Anonymization, Authentication and Authorization System	179
5.9. INTEGRATED AND LAYERED ARCHITECTURE	179
5.9.1 Semantic Data Model.....	179
5.9.2 Semantic Business Logic (Reasoning) Model.....	180
5.9.3 Semantic Services / Application Model	181
5.10. CONCLUSIONS.....	182
CHAPTER 6. ONTOLOGIES PROPOSAL	184
6.1. INTRODUCTION.....	184
6.2. METHODOLOGY USED.....	185
6.3. TOOLS USED	187
6.3.1 OWL2.....	187
6.3.2 Protégé	187
6.4. CYBER THREAT INTELLIGENCE ONTOLOGY SPECIFICATION.....	187
6.4.1 Domain and Scope.....	187
6.4.2 Reusing Affected Ontologies.....	188
6.4.3 The Important Terms.....	188
6.4.4 Define Classes, Class Hierarchy, the Properties of the Classes and the Characteristics of the Properties	188
6.4.5 Create Instances of each Class and fill in their Properties	197
6.5. DYNAMIC RISK MANAGEMENT ONTOLOGY SPECIFICATION.....	198
6.5.1 Domain and Scope.....	198
6.5.2 Reusing Affected Ontologies.....	198
6.5.3 The Important Terms.....	199
6.5.4 Define Classes, Class Hierarchy, the Properties of the Classes and the Characteristics of the Properties	199
6.5.5 Create Instances of Each Class and Fill in Their Properties	215
6.6. CONCLUSIONS.....	217
CHAPTER 7. REASONING AND BEHAVIOR	218
7.1. INTRODUCTION.....	218
7.2. USE OF REASONING FOR DYNAMIC RISK AND CTI INTELLIGENCE BEHAVIORAL DOMAINS	218
7.3. TYPES OF REASONING RULES.....	220
7.3.1 Asset Valuation Rules	220
7.3.2 Threat Inventory Rules.....	221
7.3.3 Risk Inventory Rules	222
7.3.4 Data Enrichment Rules	222
7.3.5 Risk Assessment Rules	223
7.3.6 Risk Severity Rules.....	223
7.3.7 Security Policy Rules	223
7.3.8 Risk Management Rules	224
7.3.9 Threat Intelligence Rules	225
7.4. CONCLUSIONS.....	225
CHAPTER 8. INFORMATION EXCHANGE PROPOSAL.....	227

8.1. INTRODUCTION.....	227
8.2. RELATED WORK	229
8.3. APPROACH AND RESULTS.....	232
8.4. ROLES	233
8.5. MARKET GROWTH: MONTECARLO SIMULATION OF NEW INCENTIVES.....	234
8.6. CTI TOKEN	238
8.7. DECENTRALIZED ARCHITECTURE.....	239
8.8. CONCLUSIONS.....	242
CHAPTER 9. VALIDATION.....	243
9.1. INTRODUCTION.....	243
9.2. ORGANIZATION UNDER STUDY: A NATIONAL CSIRT	243
9.2.1 <i>Cyber Threat Intelligence Model</i>	243
9.2.2 <i>Risk Assessment and Management Model</i>	247
9.3. USE CASES.....	251
9.3.1 <i>Use Case 1: Dynamic Risk Management While Under Attack</i>	251
9.3.2 <i>Use Case 2: Exchange of Threat and Risk Intelligence</i>	253
9.4. PROTOTYPE VALIDATION	254
9.4.1 <i>Use Case 1 Prototype Validation</i>	254
9.4.2 <i>Use Case 2 Prototype Validation</i>	262
9.5. RESULTS.....	270
9.5.1 <i>Use Case 1 Results</i>	270
9.5.2 <i>Use Case 2 Results</i>	275
CHAPTER 10. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS	282
10.1. CONCLUSIONS.....	282
10.2. DISSEMINATION OF RESULTS	283
10.3. FUTURE RESEARCH DIRECTION.....	283
REFERENCES	285
ANNEX A. CYBER-OBSERVABLES STIX™ V2.....	292
A.I. ARTIFACT	292
A.II. AS - AUTONOMOUS SYSTEM.....	292
A.III. DIRECTORY.....	292
A.IV. DOMAIN NAME.....	292
A.V. EMAIL ADDRESS	293
A.VI. EMAIL MESSAGE.....	293
A.VII. FILE OBJECT.....	294
A.VIII. IPV4 ADDRESS OBJECT	298
A.IX. IPV6 ADDRESS OBJECT	298
A.X. MAC ADDRESS OBJECT.....	298
A.XI. MUTEX OBJECT	298
A.XII. NETWORK TRAFFIC OBJECT	299
A.XIII. PROCESS OBJECT	302
A.XIV. SW OBJECT	303
A.XV. URL OBJECT.....	303
A.XVI. USER ACCOUNT OBJECT	304
A.XVII. WINDOWS™ REGISTRY KEY FILE	304
A.XVIII. X.509 CERTIFICATE OBJECT.....	305
ANNEX B. CYBER THREAT INTELLIGENCE ONTOLOGY DETAILS	306
B.I. ATTACK PATTERN.....	306
B.II. BUNDLE	307
B.III. CAMPAIGN.....	308
B.IV. COURSE OF ACTION	309
B.V. CYBER OBSERVABLE.....	310
B.VI. DICTIONARY	357

B.VII. IDENTITY	358
B.VIII. INCIDENT	359
B.IX. INDICATOR.....	359
B.X. INFRASTRUCTURE	361
B.XI. INTRUSION SET	361
B.XII. KILL CHAIN PHASE	363
B.XIII. MALWARE	363
B.XIV. MARKING DEFINITION.....	376
B.XV. OBSERVED DATA	380
B.XVI. REPORT	381
B.XVII. SIGHTING.....	382
B.XVIII. THREAT ACTOR.....	384
B.XIX. TOOL	386
B.XX. VULNERABILITY	387
ANNEX C. DYNAMIC RISK MANAGEMENT ONTOLOGY DETAILS	389
C.I. ASSET	389
C.II. ASSET VALUATION	391
C.III. CONTEXT	392
C.IV. SAFEGUARDS	393
C.V. SECURITY EVENTS	399
C.VI. INCIDENT	400
C.VII. RISK MANAGEMENT	400
C.VIII. THREAT	401
C.IX. RISK SEVERITY	404
C.X. RISK SCOPE	405
C.XI. RISK.....	405
C.XII. RISK IMPACT	408
C.XIII. RISK OWNER.....	408
C.XIV. RISK PROBABILITY.....	409
C.XV. RISK ASSESSMENT.....	410
ANNEX D. SWRL REASONING RULES DETAIL	412
D.I. Asset Valuation Rules	412
D.II. Threat Inventory Rules	414
D.III. Risk Inventory Rules	417
D.IV. Data Enrichment Rules	418
D.V. Risk Assessment Rules	419
D.VI. Risk Severity Rules	420
D.VII. Security Policy Rules	422
D.VIII. Risk Management Rules.....	423
D.IX. Threat Intelligence Rules	425
ANNEX E. LIST OF ABBREVIATIONS AND ACRONYMS.....	427

LIST OF FIGURES

Figure 1 Levels of formality to approach Risk Management	33
Figure 2 ISMS framework (source: ENISA)	34
Figure 3 RM and ISMS (source: ENISA)	36
Figure 4 Components of the Risk Management process	38
Figure 5 Risk Assessment sub-phases: identification, analysis and evaluation	38
Figure 6 Relationships between the RM principles, framework and process (source: ISO 31000)	44
Figure 7 Research challenges of managing cyber risk topic	53
Figure 8. Kill Chain Phases.....	54
Figure 9. Some STIX™ use cases.....	60
Figure 10. Characterization of STIX: core constructs and main relationships.....	63
Figure 11. STIX™ Cyber Observable object.....	64
Figure 12. STIX™ Indicators object.....	65
Figure 13. STIX™ Incidents object.....	65
Figure 14. STIX™ TTP object.....	66
Figure 15. STIX™ Campaign object.....	67
Figure 16. STIX™ Threat Actor object.....	67
Figure 17. STIX™ Exploit Target object.....	68
Figure 18. STIX™ Course of action object.....	69
Figure 19. STIX™ Data markings object.....	70
Figure 20. STIX™ 2 twelve objects and relationships.....	71
Figure 21. STIX™ 2 legend of relationship types between objects.....	71
Figure 22. STIX™ 2 Attack pattern object and relationships.....	72
Figure 23. STIX™ 2 Campaign object and relationships.....	73
Figure 24. STIX™ 2 Course of action object and relationships.....	74
Figure 25. STIX™ 2 Identity object and relationships.....	75
Figure 26. STIX™ 2 Indicator object and relationships.....	76
Figure 27. STIX™ 2 Intrusion set object and relationships.....	78
Figure 28. STIX™ 2 Malware object and relationships.....	79
Figure 29. STIX™ 2 Observed data object.....	80
Figure 30. STIX™ 2 Report object.....	81
Figure 31. STIX™ 2 Threat Actor object and relationships.....	83
Figure 32. STIX™ 2 Tool object and relationships.....	84
Figure 33. STIX™ 2 Vulnerability object and relationships.....	85
Figure 34. STIX™ 2 relationship summary table.....	87
Figure 35. STIX™ 2 Patterning components.....	91
Figure 36. STIX™ V1.1 Object VS. STIX™ V2 Domain Object	95
Figure 37. STIX™ 2 Relationship Objects	96
Figure 38. STIX™ V1.1 Indicator vs. STIX™ 2 Indicator pattern	97
Figure 39 Table of MSM (Making Security Measurable) vs. Areas [Hernandez-Ardieta13]	101
Figure 40. TAXII collection and channels.....	103
Figure 41. TAXII 2.0 API Roots components and relationships	104
Figure 42. TAXII collections and channels.....	105
Figure 43 Examples of shared contents per sharing level (source: NIS WG2 – Information sharing)	110
Figure 44 Ontology stack (source: Wikipedia)	115
Figure 45 RDF Description (source: Dlib.org)	116
Figure 46 Language hierarchy [Fensel01]	117
Figure 47 OWL2 Structure.....	119
Figure 48 Ontology representation of some instances of our integrated CTI and DRM Framework	155
Figure 49 Information sharing topologies: a) Decentralized peer-to-peer sharing b) Centralized sharing through an instance.	157
Figure 50 Confederated information sharing topology a) mixed with either decentralized P-to-P b1), b2) or centralized topologies c1), c2) and c3)	157
Figure 51 Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX v2	161
Figure 52 Proposal of a new model of "Entity within the DRM Framework".....	162
Figure 53 Proposal of P-to-P pure decentralized Intelligence Sharing Network based on blockchain	166
Figure 54. General outline of the DRM Framework Architecture	168
Figure 55 Dynamic Risk and CTI Data-set.....	169
Figure 56 Dynamic Risk Management: DRA and DRM related modules.....	170
Figure 57 The reasoner: a key building block of the whole architecture	174
Figure 58 Decision Making level	175
Figure 59 Pyramid of pain [Bianco14].....	178
Figure 60 SWRL DNS Enrichment.....	181
Figure 61 SQWRL query example	182

Figure 62 CTI Ontology representation (STIX2.owl)	189
Figure 63 Example of some CyberObservable instances (STIX2.owl ontology)	198
Figure 64 DRM ontology overview (DRM.owl imports STIX2.owl)	200
Figure 65 Visualization of some DRM ontology instances	216
Figure 66 Automatic instances of class RiskAssessment, created by the reasoner.....	216
Figure 67 Visual ontology graph of the super-domain (STIX/CTI and Risk domains)	217
Figure 68 Asset dependency tree for asset valuation	221
Figure 69 Roles overlapping	234
Figure 70 Montecarlo simulation of benefits with an evolutionary tax and CTI token value (p_c).....	235
Figure 71 Montecarlo simulation to compare our new model versus current models.....	237
Figure 72 Architecture design options: 1. Client-Blockchain (Serverless), 2. Server-Blockchain (Clientless), 3. Client-Blockchain-Server	240
Figure 73 Client-Blockchain-Server design option which might need enhanced synchronization.	241
Figure 74 Intelligence Model of the leading organization (case study).....	246
Figure 75. Flowchart of Analysis and Risk Management phases of the entity under study (CSIRT).....	248
Figure 76. ISMS Risk categories of the entity under study	249
Figure 77 Graphical description representing the TTP of the Watering hole attack.....	252
Figure 78 SWRL Tactic rule to launch an Information Sharing of a "Detection rule".....	258
Figure 79 SWRL Strategic rule to select specific users under attack for an "Awareness Training".....	259
Figure 80 SWRL Enrichment rule of a Data Access Security Policy.....	259
Figure 81 SWRL Auto Residual Risk Calculation for Deliberated Malicious SW Distribution Risks.	260
Figure 82 Screenshot in Protégé tool of the Security Event created by a TTP detection rule.....	260
Figure 83 Results of a querying rule (SQWRL) for a Risk recalculation.....	261
Figure 84 Sequence diagram of the CTI Marketplace Smart contract deployment by its Owner (Wallet 1)...	263
Figure 85 Capture of a running-example where the system automatically mint and send 400 CTI tokens.	266
Figure 86 Sequence diagram of a CTI Data consumer listing CTI data rules of a specific producer.	267
Figure 87 Sequence diagram of two different type of investments: single investor (Wallet 4) and a CTI Data producer (Wallet 2)	268
Figure 88 Capture of the transaction of a CTI Data consumer reading Intelligence algorithms. Associated transfers are benefits to the owner and the CTI Data producer respectively.	269
Figure 89 Screenshot of Protégé tool [Protégé] of Dynamic Risk- Reassessments.....	272
Figure 90 Screenshot in Protégé tool of the instance created dynamically of class Information Sharing Control which at the same time is including the base64 encoded version of the SWRL attack detection TTP.	274
Figure 91 Screenshot in Protégé tool of the instance created dynamically of class HR Related Security Awareness Education and Cyber capabilities control as recommended training just for the CFO by the reasoner due to real time security events.	275
Figure 92 Capture of Remix IDE of a transaction in Ropsten mined and executed successfully to share a CTI enrichment algorithm.	276
Figure 93 Graph of gas used to write (store) Intelligence algorithms in the Marketplace in bulk operations..	277
Figure 94 Graph of gas cost to write (store) Intelligence algorithms in the Marketplace and the cost per rule in bulk operations	278
Figure 95 Graph of gas cost to read (query) Intelligence algorithms in the Marketplace and the cost per rule in bulk operations	278
Figure 96 Graph comparing the gas used between READ (querying) and WRITE (storing) Intelligence algorithms in bulk operations within the CTI Blockchain based Marketplace.	279
Figure 97 Asset dependency tree for asset valuation of our organization.....	412
Figure 98 SWRL rule to make automatic inventory of deliberated malicious SW distribution threats when a service in scope of our risk management framework depends on data and later on SW (OS, Browser, AdobeFlashPlugin) but also on HW (Personal Computers)	416
Figure 99 SWRL rule for auto classification of high severity risks.....	421
Figure 100 SWRL rule used in risk management for automatic classification. Once a high severity risk is detected, it is automatically classified for a mitigation strategy because the risk is not acceptable for the organization as is.....	424
Figure 101 Pattern based SWRL rule to detect a Security Event that will be triggered by a dropper executable delivered automatically after a network traffic redirection of an injected Javascript takes place. It has not specific IOC, all individuals are variables.....	426

LIST OF TABLES

Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions	26
Table 2 Solutions provided by our proposal to all open challenges and limitations identified in Table 1.....	28
Table 3 TAXII defined Endpoints (URLs and HTTP Methods) and the resources they operate on.....	106
Table 4 OWL2 syntax formats supported.....	120
Table 5 Table of characteristics and features provided by the reasoner within the DRM framework.....	219
Table 6 Solution provided to challenges of Table 1 by the use of SWRL and a Semantic Reasoner	220
Table 7 Example of an Asset Valuation rule about a "Data Inherit Valuation".....	221
Table 8 Example of a Threat Inventory rule to automate the identification of Bad Reputation threats associated to Classified Data	222
Table 9 Example of a Risk Inventory rule to automate the identification of Bad Reputation Risks	222
Table 10 Example of an Enrichment rule to automate the enrichment of DNS related data by its RDNS.....	222
Table 11 Example of a Risk Assessment rule to automate the assessment of Risks	223
Table 12 Example of a Risk Severity rule to automatically classify any Extreme Risks	223
Table 13 Example of a Security Policy rule to automate the detection, inventory and establishment of Data-Access-Control Policy.....	224
Table 14 Example of a Risk Management Mitigation Strategy for Extreme Risk Severity	224
Table 15 Example of a Threat Intelligence rule to automate the detection of a Security Event	225
Table 16 Properties of Attack pattern class (STIX2.owl).....	306
Table 17 Properties of Bundle class (STIX2.owl).....	307
Table 18 Properties of Campaign class (STIX2.owl).....	308
Table 19 Properties of Course of Action (CoA) class (STIX2.owl)	310
Table 20 Properties of Artifact cyber observable class (STIX2.owl).....	310
Table 21 Properties of Autonomous System (AS) cyber observable class (STIX2.owl).....	311
Table 22 Properties of Directory cyber observable class (STIX2.owl)	312
Table 23 Properties of Domain name cyber observable class (STIX2.owl)	313
Table 24 Properties of Email address cyber observable class (STIX2.owl)	314
Table 25 Properties of Email message cyber observable class (STIX2.owl)	315
Table 26 Properties of Mime Part Type class (STIX2.owl).....	317
Table 27 Properties of File cyber observable class (STIX2.owl)	319
Table 28 Properties of Archive File extension cyber observable class (STIX2.owl).....	321
Table 29 Properties of NTFS File extension cyber observable class (STIX2.owl)	323
Table 30 Properties of Alternate Data Stream type class (STIX2.owl).....	324
Table 31 Properties of PDF File extension cyber observable class (STIX2.owl)	325
Table 32 Properties of Raster Image File extension cyber observable class (STIX2.owl)	326
Table 33 Properties of Windows PE Binary File extension cyber observable class (STIX2.owl)	328
Table 34 Windows PE Binary types class.....	329
Table 35 Windows PE Optional header type class	329
Table 36 Windows PE Section type class.....	332
Table 37 Properties of IPv4Addr cyber observable class (STIX2.owl)	332
Table 38 Properties of IPv6Addr cyber observable class (STIX2.owl)	333
Table 39 Properties of Mac-addr cyber observable class (STIX2.owl).....	334
Table 40 Properties of MUTEX cyber observable class (STIX2.owl)	334
Table 41 Properties of Network Traffic cyber observable class (STIX2.owl).....	335
Table 42 Properties of HTTP Header extension class	338
Table 43 Properties of ICMP extensión class	340
Table 44 Properties of Network Socket extension class	341
Table 45 Properties of TCP extensión class	343
Table 46 Properties of Process cyber observable class (STIX2.owl).....	344
Table 47 Properties of Windows process extension class	346
Table 48 Properties of Windows Service extension class.....	348
Table 49 Properties of SW cyber observable class (STIX2.owl)	349
Table 50 Properties of URL cyber observable class (STIX2.owl).....	350
Table 51 Properties of User Account cyber observable class (STIX2.owl)	351
Table 52 Properties of Windows Registry Key cyber observable class (STIX2.owl)	352
Table 53 properties of X509 Certificate cyber observable	354
Table 54 Properties of X509 v3 extension type	356
Table 55 Properties of Dictionary.....	357
Table 56 Properties of Identity.....	358
Table 57 Properties of Indicator.....	360
Table 58 Properties of Intrusion set.....	361
Table 59 Properties of Kill Chain Phase	363
Table 60 Properties of Malware	364
Table 61 Properties of Marking Definition	377
Table 62 Properties of TLP	379

Table 63 Properties of Statements.....	379
Table 64 Properties of Observed data	380
Table 65 Properties of Report.....	381
Table 66 Properties of Sighting.....	382
Table 67 Properties of Threat Actor.....	384
Table 68 Properties of Tool.....	387
Table 69 Properties of Vulnerability	388
Table 70 Properties of Asset class (DRM.owl).....	389
Table 71 Properties of Asset Valuation class (DRM.owl).....	391
Table 72 Properties of Context class (DRM.owl)	392
Table 73 Properties of Risk Management class (DRM.owl).....	393
Table 74 Properties of Security Events class (DRM.owl).....	399
Table 75 Properties of Risk Management class (DRM.owl).....	400
Table 76 Properties of Threat class (DRM.owl)	401
Table 77 Properties of Risk severity class (DRM.owl)	404
Table 78 Properties of Risk scope class (DRM.owl)	405
Table 79 Properties of Risk class (DRM.owl).....	406
Table 80 Properties of Risk Impact class (DRM.owl)	408
Table 81 Properties of Risk Owner class (DRM.owl)	409
Table 82 Properties of Risk Probability class (DRM.owl).....	409
Table 83 Properties of Risk Assessment class (DRM.owl)	410
Table 84 AssetValuation#1 Data Inherit Valuation.....	412
Table 85 AssetValuation#2 SW Inherit Valuation.....	413
Table 86 AssetValuation#3 HW Inherit Valuation	413
Table 87 AssetValuation#4 Users Inherit Valuation	413
Table 88 AssetValuation#5 Networks Inherit Valuation	414
Table 89 AssetValuation#6 Real Estate Inherit Valuation.....	414
Table 90 ThreatInventory#1 Auto Bad Reputation Threat in Classified Data.....	414
Table 91 ThreatInventory#2 Auto Data Protection Threat.....	415
Table 92 ThreatInventory#3 Auto Unintentional User Error Threat.....	415
Table 93 ThreatInventory#4 Auto SW Vulnerabilities Threat	415
Table 94 ThreatInventory#5 Auto Deliberated Malicious SW Distribution Threat	416
Table 95 ThreatInventory#6 Auto Social Engineering Threat	416
Table 96 ThreatInventory#7 Auto Corporate Brand Image Damage Threat	417
Table 97 RiskInventory#1 Auto Bad Reputation Risk	417
Table 98 RiskInventory#2 Auto Deliberated Malicious SW Distribution Risk	418
Table 99 Enrichment#1 Auto IPv4 assignment by Domain match (DNS)	418
Table 100 Enrichment#2 Auto Domain assignment by IPv4 match (Reverse DNS)	418
Table 101 Enrichment#3 Auto HW-dependencies by Data Access	418
Table 102 RiskAssessment#1 Auto Potential Risk Assessment.....	419
Table 103 RiskAssessment#2 Auto Residual Risk Calculation for Bad Reputation Risk.....	419
Table 104 RiskAssessment#3 Auto Residual Risk for Deliberated Malicious SW Distribution	420
Table 105 RiskAssessment#4 Increasing Risks due to Security Events.....	420
Table 106 RiskSeverity#1 Extreme Risk Classification.....	421
Table 107 RiskSeverity#2 High Risk Classification.....	421
Table 108 RiskSeverity#3 Medium Risk Classification	421
Table 109 RiskSeverity#4 Low Risk Classification.....	422
Table 110 SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)	422
Table 111 RiskManagement#1 Mitigation Strategy for Extreme Risk Severity	423
Table 112 RiskManagement#2 Mitigation Strategy for High Risk Severity	423
Table 113 RiskManagement#3 Investigation Strategy for Medium Risk Severity	424
Table 114 RiskManagement#4 monitoring Strategy for Low Risk Severity.....	424
Table 115 ThreatIntelligence#1 Security Event Dropper Detection.....	425

LIST OF EXAMPLES

Example 1. A generic Attack pattern for spear phishing, referencing CAPEC	72
Example 2. A generic Campaign against Financial services.....	73
Example 3. A Course of Action (CoA) to mitigate a specific malware connection.....	74
Example 4. An Identity name	75
Example 5. An Indicator of a specific file hash belonging to a malware.....	77
Example 6 An Intrusion-set example.....	78
Example 7. An example of how a malware is described.....	79
Example 8. Observed data of a file object	80
Example 9. A standalone Report; the consumer may or may not already have access to the referenced STIX™ Objects.....	81
Example 10. A Bundle with a Report and the STIX™ Objects that are referred to by the Report.....	82
Example 11 An example of a Threat Actor description.....	83
Example 12. An example of the VNC Tool described in STIX™.....	85
Example 13 An example of the description of Vulnerability CVE-2016-1234.....	86
Example 14. Sighting with observed data (what exactly was seen) and where it was seen.....	88
Example 15 Statement marking.....	93
Example 16. TLP WHITE.....	93
Example 17. TLP GREEN.....	93
Example 18. TLP AMBER.....	93
Example 19. TLP RED.....	94
Example 20. This example marks the Indicator and all its properties with the Marking Definition referenced by the ID.....	94
Example 21. Granular marking considering marking ref and selectors.....	94
Example 22. Client makes a request for items 0-49 (50 items) and the server responds with 0-49.....	106
Example 23. A returned Bundle example.....	107
Example 24. Basic image artifact.....	292
Example 25. Basic AS Object.....	292
Example 26. Basic directory.....	292
Example 27. Basic FQDN.....	293
Example 28. Basic email address.....	293
Example 29. Simple email message with additional header properties.....	294
Example 30. Basic file with parent directory.....	294
Example 31. Basic unencrypted ZIP Archive.....	295
Example 32. NTFS File with a single alternate datastream.....	295
Example 33. Basic PDF file.....	296
Example 34. Simple image file with EXIF data.....	296
Example 35. EXE file.....	298
Example 36. IPv4 CDIR block.....	298
Example 37. IPv6 CDIR block.....	298
Example 38. Typical MAC address.....	298
Example 39. Malware Mutex.....	299
Example 40. Basic HTTP network traffic.....	299
Example 41. Basic TCP network traffic.....	299
Example 42. Basic HTTP Request.....	300
Example 43. Basic ICMP traffic.....	301
Example 44. Basic stream SOCKET.....	301
Example 45. Basic TCP traffic.....	302
Example 46. Basic process.....	302
Example 47. Basic Windows process.....	303
Example 48. Basic Windows service.....	303
Example 49. Typical SW instance.....	303
Example 50. Typical URL.....	304
Example 51. Basic UNIX account.....	304
Example 52. Simple registry key.....	305
Example 53. Basic X.509 certificate.....	305

LIST OF EQUATIONS

Equation 1 SWRL syntax: body and head	124
Equation 2 SWRL example rule to guess and uncle if knowing brothers of your parents	124
Equation 3 Example of the pseudocode of a simple SWRL rule handling an OWL ontology.....	164
Equation 4 Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX™ v2.....	164
Equation 5 Example of a CTI rule or algorithm in the format of SWRL and STIX™ v2 to define a TTP detection	165
Equation 6 Simple Risk Equation depending on impact and probability	170
Equation 7 New dynamic risk equation	173
Equation 8 Evolutionary benefit "B".....	235
Equation 9 Gross benefit or total income "I" of an entity x	236
Equation 10 Total cost "C" of an entity x.....	236
Equation 11 Evolutionary Benefit as a combination of cash and token prize	236
Equation 12 Cobb-Douglas utility function applied to our model.....	238
Equation 13 A version of Markowitz's mean-variance formula to choose between investing in stocks and our CTI token	238
Equation 14 Token prize equation	239
Equation 15 Risk Equation of our Organization under study.....	250
Equation 16 Residual Risk associated to that threat i, at time t=1	260
Equation 17 New values after risk re-assessment	261
Equation 18 Pseudocode of Constructor and token initialization	263
Equation 19 Pseudocode of the data set structure to store CTI Data Rules in the Blockchain	264
Equation 20 Pseudocode of the solidity function to share CTI Data by CTI Data producers	265
Equation 21 Wei - Ether - Gwei conversion ratio	265
Equation 22 Pseudocode of the solidity function used by CTI Data consumers to get access to CTI Data of a specific producer	267
Equation 23 Pseudocode of the solidity "fallback" function for investments in the "CTI".....	268

CHAPTER 1. INTRODUCTION

1.1. MOTIVATION

Nowadays security threats are affecting all types of entities ranging from large companies to small and medium enterprises (SME), but also to individuals. In 2018, employees of small organizations were more likely to be hit by email threats—including spam, phishing, and email malware—than those in large organizations [Symantec19].

The huge number of applications, the network connection of Industrial Control Systems (ICS), the huge volume of personal data handled by the network or the explosion of the number of connected devices worldwide, because of the IoE (internet of everything), are only some examples of the increasing complexity of the network to protect. Unintentional incidents might increase as well, while more complex contexts are considered. New risk measures have to be included during change management processes in order to prevent either unintentional or intentional incidents before they happen.

Routers and connected cameras were by far the main source of IoT attacks during last year, accounting for over 90 percent of all the attacks. The proportion of infected cameras used in attacks increased considerably during 2018. There was also a rapid increase in the number of ransomware infections on mobile devices, up by a third when compared to 2017. The U.S. was the worst affected by mobile ransomware, accounting for 63 percent of infections. It was followed by China (13 percent) and Germany (10 percent). Managing mobile device security continues to present a challenge for organizations. During 2018, one in 36 devices used in organizations were classed as high risk. This included devices that were rooted or jailbroken, along with devices that had a high degree of certainty that malware had been installed [Symantec19].

With the development of new tools and techniques, cyber-crime is consistently increasing in terms of number of attacks and level of damage caused to its victims developing new ways to gain unauthorized access to networks, programs and data [Bendovschi15].

Every year seems to bring a bigger number of attacks overall, but also a bigger number of attacks defeating the security of extremely large companies, thus affecting the information security, business continuity and customers' trust. There is a generally lack of understanding of the different types of attacks, characteristics and possible results, which may pose an obstacle in trying to defend the information security [Uma2013].

Organizations are getting better at detecting breaches. Over the past eight years, dwell times have decreased significantly – from a median dwell time of 416 days in 2011 to 78 days in 2018. That means that attackers are operating for just under three months, on average, before they are detected. Despite all this efforts, there is an increase in both Internal and External dwell times within EMEA, where the overall dwell time is still about 177 days which remained largely unchanged from 175 days in 2017 [Fireeye19].

Furthermore, there is an average of 279 days to really contain breaches. A breach lifecycle under 200 days costs \$1.2 million, however the average of data breaches are about \$ 3.9 million. The estimated cost per record lost is about \$150 and the healthcare is the most costly industry [Ponemon19].

Organizations, and in particular Boards, are taking cyber security far more seriously. This has been in part driven by regulation such as the General Data Protection Regulation (GDPR¹), but also due to increased recognition of the risk presented by targeted cyber

¹ <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679>

attackers. If you've been breached, you are much more likely to be targeted again and possibly suffer another breach. As more and more customers move to software as a service and cloud, attackers are following the data. Attacks against cloud providers, telecoms, and other organizations with access to large amounts of data, have increased [Fireeye19].

In cyber security, we have traditionally been focusing on identifying what we want to protect, and then building defenses around them along the time. When adversaries breach defenses, organizations adapt themselves to prevent those breaches to occur, and, of course, as we adapt, our adversaries adapt as well.

In 2018 there was a wide range of security challenges, ranging from simple misconfiguration issues to vulnerabilities in hardware chips. Poorly secured cloud databases continued to be a weak point for organizations. S3 buckets emerged as an Achilles heel for organizations, with more than 70 million records stolen or leaked as a result of poor configuration. This was on the heels of a spate of ransomware attacks against open databases such as MongoDB in 2017, which saw attackers wipe their contents and seek payment in order to restore them. There is anyway a common theme across these incidents: poor configuration.

While the overall number of targeted attacks was down somewhat last year, the most active groups stepped up their activity, attacking an average of 55 organizations over the past three years, up from 42 between 2015 and 2017. Spear-phishing emails remained the most popular avenue for attack and were used by 65 percent of all known groups. The most likely reason for an organization to experience a targeted attack was intelligence gathering, which is the motive for 96 percent of groups. Alongside the rise in popularity of living off the land tactics, the use of zero-day vulnerabilities declined in 2018, with only 23 percent of groups known to have exploited zero days, down from 27 percent in 2017. While still a niche area, the use of destructive malware continued to grow. Eight percent of groups were known to use destructive tools, a 25 percent increase over 2017 [Symantec19].

Supply chain attacks continued to be a feature of the threat landscape, with attacks increasing by 78 percent in 2018 [Symantec19]. These type of attacks exploit third-party services and software to compromise a final target, taking many forms, including hijacking software updates and injecting malicious code into legitimate software. Developers continued to be exploited as a source of supply chain attacks, either through attackers stealing credentials for version control tools, or by attackers compromising third-party libraries that are integrated into larger software projects.

The shortage of cybersecurity professionals is close to 3 million globally. This number may seem abstract, but it's having a real-world impact on companies and on the people who are responsible for their cybersecurity. According to the survey by ISC² [ISC2-18], 63% of respondents reported that their organizations have a shortage of IT staff dedicated to cybersecurity. Nearly 60% said that their companies are at moderate or extreme risk of cybersecurity attacks due to this shortage.

The risk goes beyond each own organization as there is a high risk of integrating a compromised entity into a parent organization through a merger or acquisition (M&A). During M&A activity, numerous due diligence and integration efforts are executed under aggressive deadlines to achieve financial and business objectives. To meet these objectives, leadership will sometimes accept risk by moving forward with integrating the organizations' computer networks without fully resolving security objectives. Despite intending to resolve incomplete or missed objectives over a longer timeframe, these objectives are often forgotten, reducing the security posture of the combined company [Fireeye19]. This provides attackers an opportunity to leverage the compromise of an acquired company to compromise the network of the acquiring company. In some cases, a single compromised email account could be leveraged to increase an attacker's access to the victim network.

Before an environment can be properly hardened and secured, organizations must first ensure that visibility and detection mechanisms are tuned for the current environment to reduce potential operational impacts. This, in turn, helps ensure that any planned security controls will be effective in mitigating risks related to an attacker compromising the existing infrastructure and underlying data. There are usually deficiencies in organizations' understanding of and visibility into their own environment directly led to failures in their ability to detect and respond to breaches. This allowed attackers to access critical systems without detection and consequently hampered organizations' ability to implement eradication steps in a short timeframe.

The formation of an incident response team reduces the cost of data breach by an average of \$ 360,000 [Ponemon19], as well as it helps to contain and mitigate the impact faster. In addition to the cost perspective, we have to take care about the image and reputation of our brands where any preventive action should be always considered better than a response.

Nowadays most the systems have been developed and deployed without enough security engineering processes as well as they are sometimes delivered to market before they have been security tested and patched. This is producing a fast rate on discovering new vulnerabilities or engineering malfunctions which lack of common and standardized way of reporting, sharing or responsible disclosing.

Because of the dynamic nature of the threats, the cyber threat landscape as well as the attack surface of any organization, change constantly, affecting all type of services, systems or processes at any level. Threats are also becoming more sophisticated as well as more polymorphic or even unpredictable. Human-motivated attacks are intelligence in nature. Threat Actors use new Techniques, Tactics and Procedures (TTP) [Oasis-STIX-TTP] with different motivation and objectives. Expert theories are becoming more and more useless as well as the use of signature based security appliances (e.g. Indicators of Compromise - IoC). Then, its corporate risk level (which is based on probability and potential impact of such threats) changes too.

Current frameworks and methodologies for Risk Assessment (RA) processes [ISO/IEC27005:2008] [ISO/IEC31000:2018] follow an iterative approach in which a partial snapshot of the organization assets and business processes is periodically taken for the estimation of its risk exposure and it is primarily based on expert and subjective theories. Risk management (RM) and its countermeasures, are also identified during these periodical reviews with the intent to keep the risk below an acceptable level. This real and dynamic behavior render these legacy frameworks and methodologies highly ineffective and unreliable for any organization or risk analyst. None of probability and impact variables used for risk estimation can depend just on a specific person experience. Experience is relevant and necessary, but it is not enough, we should take into account the real time threats in order to address them dynamically by investing in related countermeasures or responding with actions as earlier as possible, at the same time we measure their effectiveness. There is just one reality of an organization but current approaches, even by international leading organizations in the cybersecurity arena, are not addressing RA/RM processes by considering NRT (near real time) threats dynamically. Then, risk assessments (as demonstrated in our work by the example of an international lead organization), are not updated dynamically once a security event, an attack or an incident, is detected. RA/RM processes are usually disconnected from incident handling or from any other real time cybersecurity threat intelligence system. It also implies limitations to have more effective prevention safeguards, better detection mechanisms and more real ROI (return of investment) calculations over safeguards.

At the same time, a small cyber security flaw is usually enough for an attacker to success, especially if it is out of the scope of the Information Security Management System (ISMS), or out of the security perimeter. We need to count with enough and reliable

information on time to take informed decisions as well as to count with enough automation in order to handle such volume of data about threats and risks.

It is paramount to count with enough reliable sources (internal or external) as no one has enough knowledge to fight alone against all threats. Information sharing is then a must to any organization but there are still open challenges and limitations as shown in Table 1.

The table includes references from the bibliography to support each concept. All the open challenges can be grouped into the following groups:

- The lack of trust (infrastructure, admin and peers)
- The lack of incentives (business cases) provided to all roles simultaneously.
- The asymmetry between consumers and producers.
- The reliability and accuracy of CTI data.
- The lack of semantics (unambiguous data) to exchange knowledge (beyond single pieces of data).
- The effectiveness and efficiency of platforms (automation).

Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions

ID	Concept	References
1	Users reluctance to participate in cyber incident information sharing	[deFuentes17], [Ring14], [Tounsi18], [Riesco19b]
2	Users reluctance to share sensitive CTI data (e.g. due to privacy concerns)	[NIST16], [deFuentes17], [Riesco19a], [Tounsi18]
3	Trust issues between users and platform providers are mostly neglected	[deFuentes17], [Ring14], [Sauerwein17]
4	The lack of trust in the sharing infrastructure	[deFuentes17], [Leszczyna19], [Riesco19b]
5	Information sharing asymmetry: more consumers than producers	[Vishik13], [Ring14], [Riesco19b]
6	Lack of incentives for information sharing	[Vishik13], [Tosh18], [Riesco19b]
7	Lack of business models associated within cybersecurity initiatives	[Vishik13], [Tosh18], [Riesco19b]
8	Misapprehended costs	[Skopik16], [Ring14]
9	CTI data mostly limited to IoC instead of advanced intelligence	[Sauerwein17], [Ring14], [Riesco19a], [Riesco19b]
10	Low reliability, accuracy and quality of threat intelligence information data	[OSINTArmy12], [MISP16], [NATO01]
11	IoC as ephemeral data	[Riesco19a], [Bianco14]
12	Timing: Fast sharing is important but not enough	[Ring14], [Tounsi18]

ID	Concept	References
13	Unmanageable volume of (big) threat data	[Tounsi18]
14	CTI data taxonomies still lack of enough expressivity (e.g. TTP definition)	[Riesco19a], [OasisSTIX-TTP]
15	CTI data taxonomies lack of semantics (e.g. unambiguous and universal understanding by reasoners)	[Riesco19a]
16	Static approach (e.g. signatures) does not match the dynamic nature of new generation of threats	[Riesco19a], [Tounsi18]
17	Grained situational awareness need to be linked to information sharing	[Riesco19a], [Leszczyna19]
18	CTI not synchronized within information security management systems dynamically at all levels	[Riesco19a], [Leszczyna19]
19	There is no a common definition of threat intelligence sharing platform	[Sauerwein17]
20	The majority of platforms are closed source	[Sauerwein17]
21	Most platforms focus on data collection instead of analysis	[Sauerwein17]

Interoperability, standardization, expressiveness as well as the need of automation are some of our reasons to propose a **Dynamic Risk Management (DRM) Framework model** (see CHAPTER 5) **based on STIX™ v2** (see section 3.4.7 and ANNEX A), **ontologies** (see CHAPTER 6, ANNEX B and ANNEX C), **SWRL and a reasoner** (see CHAPTER 7 and ANNEX D). It helps to minimize the impact of different threat and risk management interpretations among different countries, at the same time it enables effective automation via machine to machine communication.

The need to count with reliable data from third parties, despite the lack of trust (infrastructure, admin and peers), the lack of incentives (business cases) and the asymmetry between consumers and producers in Information Sharing initiatives nowadays, are some of the reasons to add the use of the **Ethereum Blockchain** (see 3.5 and CHAPTER 8) to the DRM Framework proposal.

A summary of the contributions of the DRM Framework can be seen in Table 2.

1.2. CONTRIBUTIONS

The main contribution of this thesis is a **Dynamic Risk Management Framework**.

Existing standards, methodologies and best practices today are suggesting the need to support continuous processes by new formal approaches. Risk Assessment is usually executed at discrete time points (e.g. once per year), so it is not a real continuous process (as seen in Figure 3 RM and ISMS (source: ENISA)).

In this work, we propose a mix of 3 standards:

- **STIX™ [OasisSTIX1] as an Industry driven standard,**
- **OWL [W3C-OWL] and**
- **SWRL [W3C-SWRL]**

They will contribute to overcome all semantic expressiveness and limitations of STIX™ [OasisSTIX-TTP] [OasisSTIX-Campaign] [OasisSTIX-Incidents]. We consider that if our proposal is based on standards, it will be easier to implement and deploy by several organizations in the future.

Semantic ontologies [W3C-Ontology], supported by scientific language researchers as well as the World Wide Web Consortium (W3C), is a standard solution to create formal models, defining concepts, domains and relationships (even complex ones) guaranteeing unequivocal meaning.

They are considered **building blocks for semantic inference** [W3C-Inference], which is a mechanism to discover new relationships, to automatically analyze the content of the data and to manage knowledge. These inference based techniques are also important in discovering possible inconsistencies in the (integrated) data.

Our proposal of a Dynamic Risk Management framework, also includes **a reasoner** which will be described later (see 4.4 and CHAPTER 7). Its purpose is to **act dynamically** depending on the data. **It will also infer new knowledge depending on the semantics. All is done dynamically.**

We also propose to **connect Risk domain within Cyber Threat Intelligence domain**, that is, the real time threat landscape. That will enable the reception of dynamic inputs to learn and to act accordingly in NRT (near real time). It will contribute to the dynamics of the framework, **replacing assessments only at discrete points by dynamic re-assessments.**

The Framework will work at different levels (**operational, tactical and strategic**) depending on the data and type of roles involved. Because it is impossible to fight alone against all threats, organizations must share and consume reliable data from third parties as even manufacturers are doing. Due to there are still several open challenges and limitations (see Table 1) associated to Information Sharing initiatives, **we propose the use of the Ethereum Blockchain** within our DRM Framework to solve all identified issues. A summary of all solutions provided by our DRM Framework can be seen in Table 2.

Table 2 Solutions provided by our proposal to all open challenges and limitations identified in Table 1.

ID	Solution provided	IDs of Table 1 solved
1	CTI and risk data expressiveness provided by W3C semantic web ontologies and SWRL	10, 14, 15
2	Sharing of CTI and Risk behavioral context aware rules or algorithms beyond IoC	2, 9, 11, 12, 13, 16
3	Interoperability due to the use of semantic web ontologies	10, 14, 15
4	Interoperability and easy adoption due to the use of an OWL version of STIX™	14, 15, 18
5	Automation by using semantic web reasoners (analysis beyond data collection)	10, 12, 13, 17, 21
6	Quality of the data as inconsistencies are automatically detected by reasoners	10, 15, 21
7	Inference (new knowledge) provided by the use of semantic web reasoners	1, 2, 6, 19
8	Reduced volume of data needed (e.g. algorithms vs. IoC)	8, 9, 11, 12, 13
9	Situational awareness linked to information exchange provided by ontologies	15, 16, 17
10	Information sharing at all levels (operational, tactical and strategic)	15, 16, 17, 18
11	Evolutionary economic incentives ("Ether" and a new "CTI token") as business model	1, 5, 6, 7, 8

12	Blockchain inherent trusted network (validations and verifications of all transactions)	1, 3, 4, 12
13	Smart contract decentralized application (dApp) instanced in the blockchain	3, 4, 8, 19
14	Transparent running (open source) code (what, when, why, who)	3, 4, 10, 20, 21
15	Times-tamping of each transaction within blockchain blocks	3, 4, 12
16	Very low cost (just gas) (no infrastructure needed)	6, 7, 8
17	Interoperability between smart contracts (calls between dApps)	3, 4, 19, 20, 21
18	Interoperability between tokens	1, 6, 7, 19
19	dApp efficiency (EVM runs optimized code only) and decentralized availability	3, 8, 13, 19

Beyond that, our proposal is addressing several research challenges suggested by the biggest expert communities in the Cybersecurity arena² with regard to **Dynamic Risk** and **Information Sharing areas**.

The thesis provides the main contribution of a **Dynamic Risk Management Framework** with the following **original contributions** associated to it:

- **A new, layered and networked architecture** for Dynamic Risk Assessment and Management (DRA/DRM) based on STIX™, OWL ontologies, SWRL and a Pellet semantic reasoner. It includes a new approach of Information Sharing topology based on the Ethereum Blockchain.
- **A hybrid Cyber Threat Intelligence and DRM Ontology**, which represents the evolution and integration of the Cyber Threat Intelligence data within the DRA/DRM processes. For that, we implemented the OWL of a complete version of STIX™ v2.0 [OasisSTIX2] as well as a new DRM ontology. DRM and STIX2 ontologies are integrated (STIX2.owl ontology is imported by our DRM.owl ontology).
- **Definition of several SWRL rules as behavioral algorithms and axioms to support all of the business logic needed by the DRM framework.** We also used the Pellet Incremental **semantic reasoner** for this work.
- **A new enhanced and semantic approach to Information Sharing.** A semantic approach of Cyber Threat Intelligence sharing systems within Dynamic Risk Management (DRM) processes. For that, we support the exchange of CTI semantic web algorithms (beyond the exchange of IoC data) in the format of SWRL and an OWL enhanced version of STIX™ v2, at all levels (operational, tactic and strategic level).
- **A new incentive model of Information Sharing** to solve current open challenges, based on the Ethereum Blockchain. By solving this issues we would count with more dynamic and reliable CTI data, provided by more CTI Data producers for our DRM framework. New evolutionary economic incentives are provided by a combination of Ether and the creation of a new digital asset for threat intelligence, the CTI token (ERC20 compliant).
- **Simulations for the Information sharing model optimization**, and experimentation, to demonstrate its benefits and its limits, especially in terms of costs.

² <https://resilience.enisa.europa.eu/nis-platform/shared-documents/wg3-documents/strategic-research-agenda-final-v0.96/view>

1.3. STRUCTURE OF THE MEMORY

The memory is structured in chapters as follows:

- **Chapter 2** introduces the state of the art of Risk Management. It includes sections about the Information Security Management Systems (ISMS), Risk Assessment and Risk Management processes, available standards and methodologies as well as research challenges identified by the expert communities.
- **Chapter 3** introduces the state of the art of Cyber Threat Intelligence (CTI). It analyzes the importance of information sharing to always have the best information as possible on time, in order to fight effectively against new emerging threats. For that, different approaches to have a common language are described, especially the evolution of the most widespread de facto standard STIX™. Due to the need of exchange, different collaboration network approaches are described together with some research challenges identified by expert communities.
- **Chapter 4** introduces the building blocks of the semantic web. It describes the most used ontology languages, to describe knowledge, together with behavioral rule languages, to describe behaviors. A specific section of semantic reasoners is also included as it will be one of our main building blocks, contributing for the framework automation.
- **Chapter 5** describes the first contribution of the thesis, which is the proposed Dynamic Risk Management (DRM) Framework's global architecture. It includes the description of either the core components or the external components together with the description of its three main layers (Data, Business Logic and Services Layers).
- **Chapter 6** describes the second contribution of the thesis, which is the formal model to represent the knowledge within the DRM Framework, based on ontologies. Two different ontologies are defined.
- **Chapter 7** describes the third contribution of the thesis, which is the formal model to represent behaviors within the DRM Framework, based on a semantic reasoner and semantic web rule language (SWRL) rules.
- **Chapter 8** describes the fourth contribution of the thesis, which is a new model to support a more efficient and more effective exchange of information between DRM Framework's instances, based on semantic web and the Ethereum Blockchain. The model includes new type of roles, new incentives and a new decentralized architecture.
- **Chapter 9** shows different use cases, simulations and experiments in order to verify and validate our contributions.
- **Chapter 10** describes the main conclusions, the dissemination of results as well as some future research directions.
- **References** shows the main used references.
- **Annex A** shows the details of the Cyber-observables objects in STIX™V2.
- **Annex B** shows the details of the Cyber Threat Intelligence (CTI) Ontology.
- **Annex C** shows the details of the Dynamic Risk Management (DRM) Ontology.
- **Annex D** shows the details of the SWRL reasoning rules detail.
- **Annex E List of Abbreviations and Acronyms** shows the definition of most used acronyms along the thesis.

CHAPTER 2. RISK MANAGEMENT

2.1. INTRODUCTION

As described by ISO:

“Risks affecting organizations can have consequences in terms of economic performance and professional reputation, as well as environmental, safety and societal outcomes. Therefore, managing risk effectively helps organizations to perform well in an environment full of uncertainty”.

Understanding and managing risk is a key factor of the overall management of ICT systems, and risk should serve as a driver for decision makers and other stakeholders. Information security risk management is a particular concern with respect to ICT system security, software security and secure software development. Methods, techniques and tools for security risk assessment aid developers and stakeholders to identify the most critical threats and vulnerabilities, and to identify adequate means and options for mitigating security risks so as to maintain risks at an acceptable level for all stakeholders

There exist several standards, guidelines and industrial best practices for risk management, including ISO 31000 on risk management and ISO/IEC 27005 on information security risk management, that aid stakeholders in identifying, assessing and managing risks. There are currently two approaches on risk:

- ‘bottom up’ approaches start by assessing threats and vulnerabilities and decide on measures to implement based on what ‘the business’ may likely want, and
- ‘top down’ approach starts by assessing business objectives (risk is defined by ISO as ‘effect of uncertainty of objectives’) and deriving specifications for processes and information systems to ensure that the ‘effect of uncertainty’ is within acceptable bounds.

While ISO has long promoted the ‘bottom up’ approach, this may change over the coming years as both ISO 31k and 27k5 are being revised – WD2 of ISO 27005 will have a ‘top down’ approach in its Annex E.

Such guidelines and practices are most commonly applied for existing ICT and software systems, and less throughout software development. During the recent years there has been an increased focus on secure software development, the objective of which is to build security into systems already from the requirements and design phases. The Microsoft Security Development Lifecycle (SDL) is such a process, where threat modelling and risk identification is conducted during the software design phase.

Software and service systems of today are ever more dynamic, heterogeneous, compositional and evolving, where Future Internet systems like cloud services are a predominant example. Such characteristics, as well as the fact that stakeholders are multiple with sometimes conflicting interests, make many of the established methods and techniques for risk management and assessment inadequate and less fit for purpose.

Security is the capability of networks or information systems to resist accidents or illegal or malicious actions that compromise the availability, authenticity, integrity and confidentiality of the data stored or transmitted and of the services that these networks and systems offer or make accessible, with a specific level of confidence.

The objective is to protect the organization's mission, taking different security dimensions into account:

- **Availability:** Readiness of the services to be used when necessary. Lack of availability causes an interruption of services. Availability directly affects organization's productivity.
- **Integrity:** Maintenance of completeness and correctness of data. Without integrity, information may appear to be altered, corrupt or incomplete. Integrity directly affects the correct undertaking of an organization's functions.
- **Confidentiality:** Information must only reach authorized persons. Lack of confidentiality or secrecy could cause leaks of information as well as unauthorized accesses. Confidentiality is difficult to recover. Loss of confidentiality undermines the confidence of others in the organization, and may involve the breach of laws and contractual commitments related to the custody of the information.

These canonical dimensions of security may be extended with other ones that bring us closer to the users' perception of the security of their information systems:

- **Authenticity:** An entity is who it claims to be or guarantee the source from which the data originated. Against the authenticity of the information we can have manipulation of origin or of data. Against the authenticity of users accessing services, we can have spoofing.
- **Accountability:** Guarantee that it will be always possible to determine who did what and when. Accountability is essential to analyze incidents, prosecute attackers and learn from experience. Accountability maps into integrity of activity logs.

There are multiple ways of dealing with risks:

- preventing or reducing their likelihood,
- minimizing their consequences,
- sharing them with other organization (typically hiring a service or a cover insurance), or, ultimately,
- accepting them and foreseeing resources to act when needed.

Note that one legitimate option is to accept the risk. One frequently hears that absolute security does not exist; effectively, it is always necessary to accept a risk which however, must be known and subjected to the quality threshold required by the service. In fact, sometimes we accept operational risks of activities that may yield benefits surpassing the risks involved, or risks that we must face up to. That is the reason why broader definitions of risk are sometimes used:

“Effect of uncertainty on objectives. [ISO Guide 73]”:

“...Because all this is very delicate, it is not merely technical, and it includes the decision to accept a certain level of risk, it is essential to know in which conditions one is working and thus be able to ascertain to what level the system is trustworthy. This requires a methodical approach to make informed decisions, and be ready to explain rationally the decisions taken...”

This introduces some of the contributions of our work: decisions taken that can be rationally explained (by a reasoner), which are also understood either by

humans or machines. The level of automation needed for such amount of information must be high.

The analysis and management of risks is an essential part of the security process that allows the maintenance of a controlled environment, minimizing the risks to acceptable levels.

Given the complexity of the risk management process, the analytical team needs methodologies and tools to approach this process methodically; progressing from informal analyzes (e.g. designed by the entities themselves) to semi-formal analysis (e.g. ISO guides of good practices such as ISO 31000) to formal analyzes using specific internationally recognized languages (e.g. MAGERIT provided by spanish government).

Figure 1 Levels of formality to approach Risk Management

As a summary each entity might decide their own approach based on the level of formality needed. In case of higher formality, there is a need to use a methodology beyond best practices.

Trust is the strong hope that something will work as expected. Trust is a critical value in any organization providing services.

On the one hand, we strongly depend on information systems to achieve our objectives; but, on the other hand, their security is a recurring concern. People involved, who often are not technicians, wonder if these systems can be trusted. Trust is undermined with every failure, especially when investments do not translate into absence of incidents. Ideal systems do not fail. But the truth is that we accept to live with failing systems. The issue is not so much the lack of incidents in those systems, but the confidence that they are under control: to know what it might happen and what to do when it happens. The fear of the unknown is the main cause of distrust and, consequently, knowledge brings confidence: to know the risks in order to face and control them.

Resilience is the ability of systems to continue operating despite being subjected to an attack, even in a degraded or weakened state.

It includes the ability to quickly restore its essential functions after an attack.

It is about resisting so that the consequences do not imply a disaster looking for a point of support or minimum quality of service in adverse circumstances, a minimum of survival, accompanied by a plan of recovery of normal quality.

An attack includes from deliberate actions like accidents caused by people or by the environment.

Each entity defends itself in different ways:

- Availability: alternative means.
- Integrity: mechanisms for access control, detection and restoration of data.
- Confidentiality: preventive access control or reactive measures.

2.2. THE ISMS FRAMEWORK

2.2.1 ISMS Objective

Chief objective of Information Security Management (ISMS) is to implement the appropriate measurements in order to eliminate or minimize the impact that various security related threats and vulnerabilities might have on an organization. In doing so, Information Security Management will enable implementing the desirable qualitative characteristics of the services offered by the organization (i.e. availability of services, preservation of data confidentiality and integrity etc.).

Figure 2 ISMS framework (source: ENISA)

2.2.2 Scope

It is not only size but in particular the specific business activities of an organization that dictate its security related requirements on a legal, regulatory and operational level.

Small businesses with limited information systems infrastructure, whose operation does not demand handling, storage and processing of personal or confidential data, usually face minor risks or risks with lower likelihood or impact. These organizations are more likely not to maintain independent ISMS and usually deal with information security risks ad-hoc or as part of a wider Risk Management process.

Larger organizations and organizations such as banks and financial institutes, telecommunication operators, hospital and health institutes and public or governmental bodies have many reasons for addressing information security very seriously. Legal and regulatory requirements which aim at protecting sensitive or personal data as well as general public security requirements impel them to devote the utmost attention and priority to information security risks.

Under these circumstances the development and implementation of a separate and independent management process namely an Information Security Management System is the one and only alternative.

2.2.3 Steps to develop an ISMS

As shown in the Figure 2 ISMS framework (source: ENISA), the development of an ISMS framework entails the following 6 steps:

- Definition of Security Policy,
- Definition of ISMS Scope,
- Risk Assessment (as part of Risk Management),
- Risk Management,
- Selection of Appropriate Controls and
- Statement of Applicability

Steps 3 and 4, the **Risk Assessment and Management process, comprise the heart of the ISMS** and are the processes that “transform” on one hand the rules and guidelines of security policy and the targets; and on the other to transform objectives of ISMS into specific plans for the implementation of controls and mechanisms that aim at minimizing threats and vulnerabilities.

It is worth mentioning, that steps 3 and 4 are considered as a single entity, namely Risk Management.

The processes and activities related to the steps 5 and 6 do not concern information risks. They are rather related to the operative actions required for the technical implementation, maintenance and control of security measurements.

Appropriate controls may either be derived from existing (exhaustive) sets of controls or mechanisms, usually included in information security standards and guidelines, or be the outcome of a combination or adaptation of proposed controls to the specific organizational requirements or operational characteristics.

In both cases, step 6 is the documented mapping of the identified risks, applied to the specific organization with the technical implementation of security mechanisms the organization has decided to deploy.

2.2.4 Recurring Process

Finally it should be mentioned that although the ISMS is a recurring process as a whole, in most of the types of organizations mentioned above, steps 1 and 2 recur on a

longer cycle than steps 3,4,5 and 6. This is mainly because the establishment of a security policy and the definition of the ISMS scope are more often management and (to a certain extent) strategic issues while the Risk Management process is in the end an “everyday” operational concern.

2.3. RISK ASSESSMENT AND MANAGEMENT

Risk Management and Risk Assessment are major components of Information Security Management (ISM). Although they are widely known, a wide range of definitions of Risk Management and Risk Assessment are found in the relevant literature [ISO13335-2], [NIST], [ENISA Regulation]. Here a consolidated view of Risk Management and Risk Assessment is presented.

For the sake of this discussion, two approaches to presenting Risk Management and Risk Assessment, mainly based on OCTAVE [OCTAVE] and ISO 13335-2 [ISO13335-2] will be considered. Nevertheless, when necessary, structural elements that emanate from other perceptions of Risk Management and Risk Assessment are also used (e.g. consideration of Risk Management and Risk Assessment as counterparts of Information Security Management System, as parts of wider operational processes, etc.).

It seems to be generally accepted by Information Security experts, that Risk Assessment is part of the Risk Management process. After initialization, Risk Management is a recurrent activity that deals with the analysis, planning, implementation, control and monitoring of implemented measurements and the enforced security policy.

On the contrary, Risk Assessment **is executed at discrete time points** (e.g. once a year, on demand, etc.) and – until the performance of the next assessment - provides a temporary view of assessed risks and while parameterizing the entire Risk Management process.

This view of the relationship of Risk Management to Risk Assessment is depicted in the following figure as adopted from OCTAVE.

Figure 3 RM and ISMS (source: ENISA)

It is worth mentioning, that in this figure both Risk Management and Risk Assessment are presented **as processes**, that is, as sequences of activities (e.g. arrows in figure above).

Various standards and good practices exist for the establishment of these processes (e.g. through structuring, adaptation, re-configuration etc.). In practice, organizations tend to generate their own instantiations of these methods, in a form most suitable for a given organizational structure, business area or sector.

In doing so, national or international standards (or combination of those) are taken as a basis, whereas existing security mechanisms, policies and/or infrastructure are adapted one-by-one. In this way, new good practices for a particular sector are created. Some representative examples of tailored methods/good practices are:

- a method based on a native national standard (e.g. [IT-Grund]);
- a method based on a native international standard (e.g. [ISO13335-2]);
- a method based on a de facto standard (e.g. [OCTAVE]);
- a method based on a sector standard (e.g. [SIZ-DE]);
- a method based on an individual basic protection profile for the IT-systems of an organization (e.g. [SIZ-PP]);
- adoption of an already existing risk analysis of similar systems (e.g. based on an existing Protection Profiles according to Common Criteria [CC]).

In practice, combinations of the above examples are very common.

The assumption is made, that the Risk Management life-cycle presented in the figure (i.e. plan, implement, monitor, control, identify, assess), refers solely to **risks**. Similar activities that might be necessary within the Information Security Management process are considered to apply to **operational** aspects related to the implementation and control of security measurements.

Even although organizations tend to use a single method for Risk Management, multiple methods are typically be used in parallel for Risk Assessment. This is because different Risk Assessment methods might be necessary, depending on the nature of the assessed system (e.g. structure, criticality, complexity, importance, etc.).

2.3.1 Risk Management Process

It is worth to recall that the process of Risk Management, even in the case of an iterative process, consists of different parts:

- the analysis (Risk Assessment) and
- treatment of risks (Risk Management / treatment), and
- a method (just in case) to handle the Risk Assessment process.

Risk Management process

Figure 4 Components of the Risk Management process

As seen in Figure 1 Levels of formality to approach Risk Management, a methodology will be used for risk analysis in most of the cases where higher formality is needed. This is due to the fact that **standards and best practices suggest what to do but they do not define how to do it.**

As seen in Figure 6 Relationships between the RM principles, framework and process (source: ISO 31000), there is an example of Risk Management Process model used by the ISO standardization organization.

This framework, which serves all types of risks, will be applied to the risks derived from the use of information systems.

The implementation of a Risk Assessment and Management process, needs:

- To select the methodology for the process.
- To select the tool to support the methodology.
- To specify the methodology and the tool used inside the Corporate Policy that will be applied to the process.

2.3.2 Establishing the Context, Scope and Criteria

The purpose of establishing the scope, context and criteria is to adapt the process of risk management, to allow an effective risk assessment and an appropriate treatment of the risk. The scope, context and criteria involve defining the scope of the process, and understanding the external and internal contexts.

2.3.3 Risk Assessment

Risk assessment is the overall process of risk identification, risk analysis and risk evaluation.

Figure 5 Risk Assessment sub-phases: identification, analysis and evaluation

As one of the main parts of the Risk Management process, Risk Assessment is the process to analyze our situation with respect to the specific risks.

It is a complex activity that requires a methodical execution to carry it out effectively.

By methodical, I mean:

- Manage it as a project with tasks, deliveries and control points.
- Adjusted to a methodology that guides us and allows us to explain and compare results.

Due to this reason, most of the organizations are using a combination of standards, best practices and specific formal methodologies. Latest are really needed to define a consistent Risk Assessment process along the time. Some of this methodologies will be defined along the doctoral thesis.

2.3.3.1. Risk Identification

The organization should identify **sources of risk, areas of impacts, events** (including **changes** in circumstances) and **their causes** and their potential consequences.

The aim of this step is to generate a comprehensive list of risks based on those events that might create, enhance, prevent, degrade, accelerate or delay the achievement of objectives.

It is important to identify the risks associated with not pursuing an opportunity.

Comprehensive identification is critical, because a risk that is not identified at this stage will not be included in further analysis.

2.3.3.2. Risk Analysis

As described by ISO 31000, Risk analysis involves developing an understanding of the risk.

Risk analysis provides an input to risk evaluation and to decisions on whether risks need to be treated, and on the most appropriate risk treatment strategies and methods. Risk analysis can also provide an input into making decisions where choices must be made and the options involve different types and levels of risk.

Risk analysis involves consideration of the causes and sources of risk, their positive and negative consequences, and the likelihood that those consequences can occur. **Factors that affect consequences and likelihood should be identified. Risk is analyzed by determining consequences and their likelihood, and other attributes of the risk. An event can have multiple consequences and can affect multiple objectives. Existing controls and their effectiveness and efficiency should also be taken into account.**

The way in which consequences and likelihood are expressed and the way in which they are combined to determine a level of risk should reflect the type of risk, the information available and the purpose for which the risk assessment output is to be used. These should all be consistent with the risk criteria. **It is also important to consider the interdependence of different risks and their sources.**

The confidence in determination of the level of risk and its sensitivity to preconditions and assumptions should be considered in the analysis, and communicated effectively to decision makers and, as appropriate, other stakeholders.

Factors such as divergence of opinion among experts, uncertainty, availability, quality, quantity and ongoing relevance of information, or limitations on modelling should be stated and can be highlighted.

Risk analysis can be undertaken with varying degrees of detail, depending on the risk, the purpose of the analysis, and the information, data and resources available.

Analysis can be qualitative, semi-quantitative or quantitative, or a combination of these, depending on the circumstances.

As a summary, it is a systematic process to estimate the magnitude of the risks to which a system or organization is exposed as ignorance would not allow decisions to be made with foundation.

2.3.3.3. Risk Evaluation

It is the process in which the estimated risk is checked against the organization's criteria to determine the importance of the risk. The importance of the risk is related to the consequences that can be perceived by the organization which put their objectives at risk.

ISO3100 defines its objective to assist in making decisions, based on the outcomes of risk analysis, about which risks need treatment and the priority for treatment implementation.

Risk evaluation involves comparing the level of risk found during the analysis process with risk criteria established when the context was considered. Based on this comparison, the need for treatment can be considered.

Decisions should take account of the wider context of the risk and include consideration of the tolerance of the risks borne by parties other than the organization that benefits from the risk. Decisions should be made in accordance with legal, regulatory and other requirements.

In some circumstances, the risk evaluation can lead to a decision to undertake further analysis. The risk evaluation can also lead to a decision not to treat the risk in any way other than maintaining existing controls. This decision will be influenced by the organization's risk attitude and the risk criteria that have been established.

2.3.4 Risk Management / Treatment

The second part is the risk treatment or risk management of those risks which require attention after the analysis and evaluation. It is related to the selection and implementation of safeguards to know, prevent, reduce or control the identified risks. Risk treatment involves selecting one or more options for modifying risks, and implementing those options. Once implemented, treatments provide or modify the controls.

Risk treatment involves a cyclical process of:

- assessing if a risk treatment is good enough;
- deciding if residual risk levels are something acceptable;
- if not acceptable, generating a new risk treatment; and
- assessing the effectiveness of that treatment.

Risk treatment options are not necessarily mutually exclusive or appropriate in all circumstances. The options can include the following:

- avoiding the risk;
- increasing or taking the risk in order to take advantage of an opportunity;
- removing the source of the risk;
- reducing the probability / likelihood;

- reducing the consequences;
- sharing the risk with another 3rd party or parties; and
- keeping the risk by an internal informed decision.

Selecting the most appropriate risk treatment option involves balancing the costs and efforts of implementation against the benefits derived, with regard to legal, regulatory, and other requirements such as social responsibility and the protection of the natural environment.

2.3.5 Conclusion

As a conclusion we could say that if we have a good knowledge of the risks we face in each place and at each moment, we can make the best decisions to have the objectives under control and fulfill the mission of the organization.

Risk Management is the foundation for security management. It allows to quantify the risk and demonstrate that it is under control.

- It is **necessary**, but not trivial.
- It is **laborious**, requires effort from several people and the coordination of people with each other and with government bodies.
- It is **recurrent**, never ends and should be the subject of a plan of excellence and continuous improvement.

As seen in Figure 1 Levels of formality to approach Risk Management, most of the organizations are using a combination of standards, best practices and **specific formal methodologies**. **Latest are really needed to define a consistent Risk Assessment process along the time.**

2.4. STANDARDS AND METHODOLOGIES

2.4.1 ISO 27001

ISO 27001 International Standard has been prepared to provide requirements for establishing, implementing, maintaining and continually improving an **information security management system**.

The adoption of an information security management system is a strategic decision for an organization. The establishment and implementation of an organization's information security management system is influenced by the organization's needs and objectives, security requirements, the organizational processes used and the size and structure of the organization. All of these influencing factors are expected to change over time.

The information security management system preserves the confidentiality, integrity and availability of information by applying a risk management process and gives confidence to interested parties that risks are adequately managed.

It is important that the information security management system is part of an integrated with the organization's processes and overall management structure and that information security is considered in the design of processes, information systems, and controls. It is expected that an information security management system implementation will be scaled in accordance with the needs of the organization.

ISO27001 International Standard can be used by internal and external parties to assess the organization's ability to meet the organization's own information security requirements.

The order in which requirements are presented in it does not reflect their importance or imply the order in which they are to be implemented. The list items are enumerated for reference purpose only.

[ISO/IEC27000:2018] describes the overview and the vocabulary of information security management systems, referencing the information security management system family of standards (including ISO/IEC27003, ISO/IEC 27004 and ISO/IEC27005), with related terms and definitions.

In the update of 27001:2013 the main changes were related to:

- Elimination of the reference to the continuous improvement process approach plan-do-check-act.
- General restructuring of chapters and subsections. The chapters are aligned with the "Annex SL" published by ISO / IEC so that all standards of management systems have the same structure.
- Greater emphasis on knowledge of the context of the organization and the needs of the stakeholders. This knowledge should be the essential entry point for the establishment of the Management System: definition of the scope, policy, establishment of objectives and risk analysis. In this sense, it is aligned with the ISO 31000 standard for risk management.
- The process of risk analysis is defined more generally. References to the identification of assets, threats and vulnerabilities have been eliminated. It is only necessary to identify risks (without specifying how) associated with the loss of confidentiality, integrity and availability, after analyzing the potential consequences and the probability to finally quantify the risk. Additionally, the risk owner must be identified.
- Regarding the selection of security controls for the treatment of risk, it is decided that the organizations select a framework of controls in case they do not want to follow Annex A / ISO 27002, although in any case they should be compared with the controls of Annex A to verify that no control is ignored.
- Greater emphasis is placed on the leadership of the Management in the Management System, not only from the point of view of a formal commitment, as specified in the previous version, but with the aim of avoiding management as a Technology Directorate of Security.
- Another novelty introduced is the redefinition of security objectives related to information security, as part of the Management System, giving it greater relevance compared to the previous version.
- Greater depth in the area of monitoring and measuring the ISMS.
- Regarding the documentary requirements, the list of mandatory documents has been eliminated, although in the body of the standard reference is made to different documentary requirements. On the other hand, the separation between documents and records is eliminated, being called simply "documented information".
- It only takes into account corrective measures, excluding preventive measures, which are nothing other than actions derived from risk management
- Changes in the security controls annex result in a more logical and updated structure to the current reality. It goes from 11 to 14 chapters and the total number of controls is reduced from 133 to 114. The aspects related to cryptography are separated from the chapter Development and Acquisition Software and becomes an independent chapter. The same applies to Relations with Suppliers, which in this case was diluted

among several chapters. On the other hand, the Communications and Operations chapter is divided into two different ones.

2.4.2 ISO 31000

[ISO/IEC31000:2018] International standard that includes guidelines for good practices in risk management. Developed by ISO (International Organization for Standardization). Its process is characterized by the identification, analysis, evaluation, treatment, communication and monitoring of any type of risk that affects the company.

Organizations of all types and sizes face internal and external factors and influences that make it uncertain whether and when they will achieve their objectives. The effect this uncertainty has on an organization's objectives is "risk".

All activities of an organization involve risk. Organizations manage risk by identifying it, analyzing it and then evaluating whether the risk should be modified by risk treatment in order to satisfy their risk criteria. Throughout this process, they communicate and consult with stakeholders and monitor and review the risk and the controls that are modifying the risk in order to ensure that no further risk treatment is required. ISO31000 is an International Standard which describes this systematic and logical process in detail.

While all organizations manage risk to some degree, this International Standard establishes a number of principles that need to be satisfied to make risk management effective. This International Standard recommends that organizations **develop, implement and continuously improve a framework** whose purpose is to integrate the process for managing risk into the **organization's overall governance, strategy and planning, management, reporting processes, policies, values and culture**.

Risk management can be applied to an entire organization, at its many areas and levels, at any time, as well as to specific functions, projects and activities.

Although the practice of risk management has been developed over time and within many sectors in order to meet diverse needs, the adoption of consistent processes within a comprehensive framework can help to ensure that risk is managed effectively, efficiently and coherently across an organization. The generic approach described in ISO31000 International Standard provides the principles and guidelines for managing any form of risk in a systematic, transparent and credible manner and within any scope and context.

Each specific sector or application of risk management brings with it, individual needs, audiences, perceptions and criteria. Therefore, a key feature of ISO31000 International Standard is the inclusion of "establishing the context" as an activity at the start of this generic risk management process. **Establishing the context** will capture the objectives of the organization, the environment in which it pursues those objectives, its stakeholders and the diversity of risk criteria – all of which will help reveal and assess the nature and complexity of its risks.

The relationship between the principles for managing risk, the framework in which it occurs and the risk management process described in this International Standard are shown in the Figure 6 Relationships between the RM principles, framework and process (source: ISO 31000).

Figure 6 Relationships between the RM principles, framework and process (source: ISO 31000)

When implemented and maintained in accordance with ISO31000 International Standard, the management of risk enables an organization to, for example:

- increase the likelihood of achieving objectives;
- encourage proactive management;
- be aware of the need to identify and treat risk throughout the organization;
- improve the identification of opportunities and threats;
- comply with relevant legal and regulatory requirements and international norms;
- improve mandatory and voluntary reporting;
- improve governance;
- improve stakeholder confidence and trust;
- establish a reliable basis for decision making and planning;
- improve controls;
- effectively allocate and use resources for risk treatment;
- improve operational effectiveness and efficiency;
- enhance health and safety performance, as well as environmental protection;
- improve loss prevention and incident management;
- minimize losses;
- improve organizational learning; and
- improve organizational resilience.

ISO31000 International Standard is intended to meet the needs of a wide range of stakeholders, including:

- those responsible for developing risk management policy within their organization;
- those accountable for ensuring that risk is effectively managed within the organization as a whole or within a specific area, project or activity;
- those who need to evaluate an organization's effectiveness in managing risk; and
- developers of standards, guides, procedures and codes of practice that, in whole or in part, set out how risk is to be managed within the specific context of these documents.

The current management practices and processes of many organizations include components of risk management, and many organizations have already adopted a formal risk management process for particular types of risk or circumstances. In such cases, an organization can decide to carry out a critical review of its existing practices and processes in the light of ISO31000 International Standard.

In it, the expressions “risk management” and “managing risk” are both used. In general terms, “risk management” refers to the architecture (principles, framework and process) for managing risks effectively, while “managing risk” refers to applying that architecture to particular risks.

ISO31000 International Standard provides principles and generic guidelines on risk management. It can be used by any public, private or community enterprise, association, group or individual. Therefore, it is not specific to any industry or sector.

It can also be applied throughout the life of an organization, and to a wide range of activities, including **strategies and decisions, operations, processes, functions, projects, products, services and assets**.

It can also be applied to any type of risk, whatever its nature, whether having positive or negative consequences.

Although it provides generic guidelines, it is not intended to promote uniformity of risk management across organizations. The design and implementation of risk management plans and frameworks will need to take into account the varying needs of a specific organization, its particular objectives, context, structure, operations, processes, functions, projects, products, services, or assets and specific practices employed.

It is intended that it can be utilized to harmonize risk management processes in existing and future standards. It provides a common approach in support of standards dealing with specific risks and/or sectors, and does not replace those standards.

ISO31000 International Standard **is not intended** for the purpose of certification.

2.4.3 OCTAVE

Methodology of analysis and risk management developed by CERT at Carnegie Mellon University (Software Engineering Institute) in 2001. Its acronym stands for "Operationally Critical Threat, Assesed and Vulnerability Evaluation". Study the risks based on three principles: Confidentiality, Integrity and Availability. This methodology is used by different government agencies such as the US Department of Defense. There are three versions of this methodology: the original version OCTAVE, the version for small companies OCTAVE-S and the simplified version of the tool OCTAVE-ALLEGRO.

2.4.4 CRAMM

It has international recognition and is developed by the British CCTA (Central Computer and Telecommunications Agency). Its development consists of identification and

valuation of assets, assessment of threats and vulnerabilities and selection of countermeasures.

2.4.5 MAGERIT

Methodology of analysis and risk management of the Information Systems elaborated by the Superior Council of Electronic Administration in Spain. The current version is 3 and is structured in 3 books: Book I - Method, Book II - Catalog of elements and Book III - Guide to techniques. Its support tool is PILAR.

2.4.6 MEHARI

Methodology of analysis and risk management developed by CLUSIF (Club de la Sécurité de l'Information Français) in 1995 and derives from the previous methodologies Melissa and Marion. The methodology has evolved providing a guide for the implementation of security in an entity throughout the life cycle. Evaluate risks based on availability, integrity and confidentiality criteria.

2.4.7 NIST800-30

Developed by the NIST (National Institute of Standards and Technology), an agency of the Department of Commerce of the United States. The publication of this guide dates from July 2002.

2.4.8 ITIL 4.0

Information Technology Infrastructure Library is a methodology that is based on the quality of the IT service. Developed, at the request of the Government of the United Kingdom, by the CCTA (Central Computer and Telecommunications Agency) that at the end of the 80 published "The Government Method for managing IT infrastructure", which would later be renamed ITIL v1. The most recent version of ITIL was launched in February 2019. It is the first major update of the ITIL framework since 2011.

2.4.9 CMMI 2.0

Capability Maturity Model Integration is a capacity and maturity model for the improvement and evaluation of processes in an organization. It was developed initially for the SW processes by the Carnegie Mellon University (CMU) for the Software Engineering Institute (SEI).

2.4.10 COBIT 5.0

Control objectives for Information and related Technology is a set of best practices for IT administration created by ISACA (Information System Audit and Control Association) and ITGI (IT Governance institute) published in 1996. In 2012, the latest edition of this framework was launched.

2.5. EXISTING METHODS

Existing industry best practices, as well as standards such as the aforementioned ISO/IEC 27005, are always relevant, and existing methods, techniques and tools for security risk management should comply with such established guidelines.

Considering the research challenges and the current research landscape, there is still a gap to close, and also a need for substantial industry liaison.

2.6. RISK MANAGEMENT RESEARCH CHALLENGES

Traditional methods for risk assessment are monolithic in the sense that systems are understood and analyzed as a whole. For large, compositional and dynamic service systems, such methods easily become too heavy and costly to apply, and existing analysis results rapidly become outdated. Understanding how to cope with this requires research in several directions.

There is a common understanding across the different areas of interest that the new developments in ICT technology and its applications are increasing the complexity of system and provide a challenge for managing this complexity and the related cyber risks. On the other hand, the state of play today, indicates that however several approaches and frameworks to risk management exists:

- the lack of awareness of decision makers,
- the lack of interoperability and standardized metrics,
- the costs-benefit ratio (especially for SMEs),
- the lack of statistical information for predictive approaches,
- the existing frameworks' outdated status,
- their static assessment model, as well as
- the perception of weakness and coverage not sufficient for the complex business and cybersecurity risks of today.

It is becoming a real and important disincentive and barrier for adoption either by larger or smaller organizations.

The complexity has been increased during recent years as threats also evolve along the time. Their very nature as well as their motivations, techniques and tactics are becoming polymorphic and more sophisticated day after day, making them unpredictable for any risk analyst.

Vulnerabilities in technology are found at a rate that renders current cybersecurity solutions and strategies rapidly obsolete. In addition, there are threats, such as APT, state-sponsored attacks or governmental surveillance programs, for which no effective solution exists, leaving our organizations and society dramatically exposed.

This ever changing and unpredictable nature of the threats and the technology on which the organization depends makes **expert judgment for risk assessment incomplete and inaccurate**.

As a consequence, this assessment should be continuously reviewed and verified against the updated state of the system under observation.

The system under observation has to be expanded, especially in the supply chain model, where organizations need to incorporate the risk level of their suppliers into their global picture. The security paradigm has changed, although traditional security approaches are based on the existence of a perimeter that needs to be protected, this paradigm does not hold anymore. We need to operate in a world where a perimeter is not there or has already been breached. Another important factor it is the ability to decide and execute effective courses of action in timely fashion in order to mitigate and remediate the impact of attacks, when occurring on real time. Historically, technical security controls (anti-malware, firewalls, IDS/IPS, etc.) have focused on prevention of compromise. However, it is now widely recognized that no matter how good your preventive controls are, they can only

reduce the number of and severity of security breaches and not eliminate them. The emphasis has now shifted to augmenting preventive controls with detection and remediation measures. Clearly, the time taken for detection, diagnosis, remediation planning, and action is critical in limiting the impact of an attack. In the future, we can expect the sophistication and speed of execution of attacks to increase, and the difficulty of formulating a timely response to become correspondingly more challenging. A capability for autonomous response will become essential, because there will simply not be enough time to have a man in the loop. However, an inappropriate response may be more damaging than the original attack, so that the controls need not only to be speedy, but also trustworthy. This implies that they must understand the limits of their authority and the consequences of their actions. To be trusted as well as trustworthy, they must be able to explain and justify their actions in retrospect. Of course, the attackers will attempt to evade the defenses, so that the defensive technology must be able to adapt dynamically to the attackers' tactics. Despite the automation, people must remain in ultimate control, being able to set and modify policies that govern the actions of the autonomous agents. Establishing effective means of man-machine co-operation will be a research challenge in itself. Current frameworks and methodologies, in the way they were conceived, cannot provide a solution for all these problems. We need novel / modern, simpler, disruptive, dynamic, multi-stakeholder, interoperable (based on formal sustainable models), standardized, predictive, reactive and holistic approaches capable of estimating and reducing the risk in real-time, feeding from real-time operational information and threat intelligence sources, and automating the risk assessment and management activities (especially detection and remediation ones) for a new perimeter paradigm as never before. At the same time the cyber risk surface increased at all levels, especially in Critical Infrastructure Protection, where the solution is too complex and significant to be left to an ad hoc collection of volunteers. On the other hand this opens to EU the opportunity to address this by an executive and collaborative approach between member states in order to keep a permanent and standardized protection of EU essential services.

That is to say, EU needs and has the opportunity to harnessing barriers to become enablers and opportunities.

2.6.1 Risk Aggregation

In order to accommodate to a modular software development process, as well as effectively handling the heterogeneous and compositional nature of service systems of today, there is a need to focus on a modular approach to the analysis of security risk and cost. Such an approach should also have the capacity to involve the perspective and requirements of several competing stakeholders such as service vendors, service providers and service consumers. When services are compositional, so are risks, and should therefore be understood, modelled, and analyzed as such. This requires methods for aggregating the global risk level through risk composition.

Another challenge is about providing support for human actors that act in one organization (al unit) so as to prevent them from making mistakes due to interoperability with another organization (al unit).

2.6.2 Evolution

The setting of dynamic and evolving services and software systems implies that also the security risks and the set of adequate mitigations and security mechanisms are dynamic and evolving.

Thus, in order to maintain a valid and up to date risk picture, there is a need to continuously reassess security risks and identify cost-efficient means for risk mitigation.

Such reassessment should not imply that systems need to be analyzed from scratch every time a change, such as a service substitution, occurs. Novel methods and techniques are needed to systematically handle change, which could include techniques for modular risk assessment and run-time risk assessment.

2.6.3 Legal Aspects and Business Rules

Many legal aspects are strongly related to information security, including privacy, data protection and contracts on service level agreements. While systems become increasingly cross-organizational and cross-national, the legal requirements become more complex to understand, including which jurisdictions that may be relevant.

Hence, legal risk assessment should be an integrated part of security risk assessment, so as to systematically assess the potential legal implications of security breaches.

Business rules also need to be complied with which include rules and policies of organization (all units) as well as internal and external contracts (which may not have legal implications but nevertheless non-compliance may harm the business).

For a holistic and complete approach, legal risks should be integrated in the organizations decision making process. It may enable the evaluation and comparison of alternative regulatory and non-regulatory responses to complex and interdependent risks and selecting among them. This process requires knowledge of the legal, economic and social factors, as well as knowledge of the business world in which legal teams operate. Risk-preventive, reactive and mitigation services, including process identification, empirical analysis, quantitative evaluation, cost analysis and dynamic support. The system's basic concept is to solve enterprise legal risk problems by means of management, and the basic principle is to describe the enterprise legal risks in the language of economics although its value should be added to the enterprise risk decision making. Research should provide medium and long-term, holistic and dynamic legal risk management solutions compatible and interoperable with other technical and business risks. At the same time, legal and contractual obligations could also facilitate the adoption of risks management practices by "hard to reach" organizations as an enabler.

2.6.4 Cost

Industry and service providers need to ensure properties such as security, compliance, privacy, trust and identity protection while making business. Without techniques to assess security cost and to ensure return on investment (RoI) in security, such properties may fail the competition with other business priorities. Novel techniques are needed that not only ensure RoI in security, but also clearly demonstrate it at a business level.

2.6.5 Requirements Engineering

Where industry knows what it needs from a system yet it cannot express the specifications thereof. In software engineering better understanding is needed of what such risks spawn from the inability of people to write down the knowledge that they have and how to mitigate them.

2.6.6 Risk Ownership

First, there is risk ownership (the term 'risk owner' is defined in ISO 31k as 'person or entity with the accountability and authority to manage a risk'). A common problem in risk

management is that risks often have no owner in the ISO sense, or if they do, they have insufficient authority (e.g. lacking the means) to manage this risk.

2.6.7 Methods to Reduce and Manage Systems Complexity

The limitations of existing risk management methodologies in terms of addressing the cascading effects from the interdependent threats, the change of the security perimeter paradigm, the increasing complexity and nature of threats and the need to manage a multi-stakeholder supply chain inside a global risk management picture between other different facts, justify the need of more research in reducing complexity, improving accuracy of impact calculations at the same time the availability of simpler tools, simpler interfaces supporting these processes allow cost-effective solutions in order to avoid more barriers to adoption.

2.6.8 Dynamic Risk Assessment and Management

In order to achieve a comprehensive and continuous situational awareness, we need novel, disruptive approaches capable of estimating the risk in real-time, feeding from real-time operational information and threat intelligence sources, and automating the risk assessment and management activities as much as possible. Dynamic risk management should take care about systems evolution for a sustainable assurance. Dynamic discovering, the way the system inference new topologies as well as the usage of new advanced multi-dimensional sensors of different nature (such as environmental sensors) as inputs to the risk analysis could improve the accuracy and efficiency of the management process.

For a broader impact, information sharing and the effective and automatic use of exchanged data tampering of field devices, roadside and infrastructure equipment along the value chain, should be considered.

2.6.9 Formal Interoperable Models to enable Comparisons and Compatibility between Multi-Disciplinary Environments

Interoperability and standardization of the way the risks are calculated.

Without this, comparing and interpreting results of two different approaches is not possible, undermining the objective evaluation of the performance and effectiveness of new solutions. This is especially relevant in the supply chain model, where organizations need to incorporate the risk level of their suppliers (including physical, human, the cyber layer, processes and services) into their global picture. There is a need to progress toward comparability of risk assessment results, whatever method is used. As we go to ever large scale interconnected systems, and the development of new risk management models and systems for cyber societies is necessary wide activities towards a holistic risk management framework.

2.6.10 Statistical and Predictive Risk Analysis

Considering that statistical risk methods do not work well with intentioned threats, new methods should be researched. Furthermore acknowledging that most common risk assessments models depend on past information to picture the current risk scenario, additional data models that helps to predict probabilities and impacts of threats in the EU could be extremely useful for example as a tool for remedial actions, investment prioritization or even for a risk externalization. For Cyber Insurance (as explained under Assurance

section), the capability to predict the current strength of the system is considered to be a strong component of any calculation. Thus security and corresponding risk metrics (as explained below) are crucial (as other quantitative aspects of security). The way this data can be shared effectively and re-used at scale it is a key factor to provide organizational and shared benefit in EU.

2.6.11 Autonomous Detection and Remediation by a Man-Machine Effective Cooperation

A capability for autonomous response will become essential to fight against cyber-attacks, because there will simply not be enough time to have a man in the loop. However, an inappropriate response may be more damaging than the original attack, so that the controls need not only to be speedy, but also trustworthy. This implies that they must understand the limits of their authority and the consequences of their actions. To be trusted as well as trustworthy, they must be able to explain and justify their actions in retrospect. Of course, the attackers will attempt to evade the defenses, so that the defensive technology must be able to adapt dynamically to the attackers' tactics not only as an active defense but also to know more about the attacker while in the attack. Despite the automation, people must remain in ultimate control, being able to set and modify policies that govern the actions of the autonomous agents. Establishing effective means of man-machine co-operation will be a research challenge in itself.

2.6.12 Integrated Risk Metrics and Indicators

In order to evaluate to what extent current metrics can be incorporated (and enhanced) into new solutions, there is a need to have a better comprehension of the metrics currently being used in traditional frameworks and methodologies. They are usually said to be based on the 'experience' of the designer, but this doesn't mean anything if the method is not justified based on solid criteria. Calculation methods should at least be auditable. Current metrics tend to focus on individual organizations and not take into account supply chains or dependencies across sectors and borders. It is also easy to focus on those things that are easy to measure and possibly ignore the real indicators of success or failure. The increasing scale and connectivity of cyber security issues means that organizations can no longer live in their own silos (and measure things that only affect them) and new approaches to identifying and setting realistic metrics should be considered e.g. that the security mechanisms are appropriate for the protection of the assets. This requires security mechanisms that fit the purpose and are able to allow security managers to trade-off between cost and risk. New security metrics frameworks able to be easily computed should be envisaged. These security metrics could be merged with risk analysis methods to decide the appropriate security controls to be put in place or even facilitate the risk externalization (i.e. Cyber Insurance).

2.6.13 Visual Decision Making Governance Frameworks

The increasing complexity force simpler interfaces for an effective man-machine governance framework. We lack of a coherent framework that puts all these pieces together and helps to identify gaps that need to be filled with further research. For example, with the aforementioned solutions it is not possible to effectively translate the risk level into business impact, moving the analysis from the operational/technical layer to the business layer. This achievement would significantly support and ease the decision-making process for risk owners. New techniques are also needed to enable more consistent and appropriate security decision making as well as allowing aggregation and composition of different pieces (Software and Hardware) without losing the control of risks either including all the value chain sensors or other factors like legal and economics.

2.6.14 New Incentives for Adoption of Risk Management Best Practices and Reducing Barriers (Especially for SME).

There is a strong belief, backed by feedback, that SMEs are not applying even basic cyber risk management methods or best practices. A research is needed to establish how to communicate with 'hard to reach' organizations and to incentivize the adoption of best practices due to the fact that still most damaged cyber-attacks against this kind of organizations are considered basic (social engineering, phishing, default passwords, patching) however all the efforts being done in awareness raising. Further research into the use and take-up of risk management methods and practices by SMEs is needed. Barriers already identified by NIS WG1 of NIS platform³ include the lack of awareness, the complexity of risk assessments, the imbalance between resources devoted to analysis and the benefits for the organization (usually seen as one-off static exercise rather than part of on-going activity within a governance framework that is maintained). In addition to this, the executive and decision makers' perception is all about cost and expense rather than preventing financial and material loss. Usually in smaller organizations, the lack of expertise, training, dedicated staff are also barriers. There is a need to look and research about potential incentives for take-up and maintenance, both within an organization and across supply chains.

2.6.15 Catalyst for Improving Impact of Research

- Cover the full spectrum of the value chain, interdependencies and domains.
- Encourage research to incentivize effective adoption of risk management practices by "hard to reach" and other SMEs organizations.
- Focus on automation under an effective man-machine cooperation against risks. • Encourage research on formal models, languages and ontologies to compare and interoperate between different frameworks and disciplines.
- Encourage research on statistical and predictive analysis based on new models for intentioned threats.
- Encourage research of integrated risk metrics, indicators and the cost factor of risk management.

2.6.16 Expected Outcome

- Develop agile risk management framework for companies and SMEs.
- Develop reliable risk metrics.
- Enable decisions makers to develop risk prediction models for cyber threats.

2.6.17 Timeline

This table lists the research topics from above that were identified by NIS Platform WG3 of European Commission⁴ but they were also re-validated by ECSO ⁵(European

³ <https://resilience.enisa.europa.eu/nis-platform>

⁴ <https://resilience.enisa.europa.eu/nis-platform/shared-documents/wg3-documents>

⁵ <https://ecs-org.eu/documents/ecs-cppp-sria.pdf>

Cybersecurity Organization) during the contractual public private partnership (cPPP). In the table, their research challenges are identified together with the corresponding time framework when those can be addressed:

Topic / Timeframe	Short (1-3)	Medium (3-5)	Long (5-8)
Methods to reduce and manage systems complexity	Methods and process for managing risk interdependencies	Simpler tools and interfaces available to support these processes	
Dynamic risk assessment and management	Automation of risk analysis	Advanced real time multi-dimensional sensing capabilities	Significant improvements on real time risk estimation
Statistical and predictive risk analysis	Theoretical foundations and supporting methods and tools for intended threats prediction		Statistical methods to estimate the current strength of the system against current and predictive risks
Autonomous detection and remediation by a man-machine effective cooperation		Effective means of man-machine co-operation	Pseudo-autonomous real-time reasoning systems for detection and remediation
Integrated risk metrics and indicators	Auditable calculation methods for risk metrics	Integrated KPI	
Visual decision making governance frameworks	New techniques for appropriate risk decision making	Integrated visual decision frameworks to support this new techniques	
Legal risk assessment and management		Legal risk semantic formal models	Comprehensive legal risk guidelines and interoperable standards approved and established in practice
Incentives for adoption of risk management best practices and	Research into the use and take-up of risk management methods		Lightweight certification and other effective models

Figure 7 Research challenges of managing cyber risk topic.

2.7. CONCLUSIONS

In this chapter, we analyzed the state of the art of Risk Management. It includes the analysis of the Information Security Management Systems (ISMS) (see section 2.2) as well as the analysis of Risk Assessment and Risk Management process and best practices (see section 2.3).

An overview of standards and methodologies is also analyzed (see section 2.4).

Despite the tremendous work, experience and standards until today, several research challenges are identified (see section 2.6). There is room for improvement, especially with regard to Dynamic Risk Management (DRM) as the main proposal of this thesis.

CHAPTER 3. CYBER THREAT INTELLIGENCE

3.1. INTRODUCTION

Cyber security is a complex and multifaceted problem domain and continues to become more so. Our dependence on complex technology continues to grow and, at the same time, the threat environment continues to grow and evolve in dynamic and daunting ways. Traditional approaches for cyber security, that focus inward on understanding and addressing vulnerabilities, weaknesses and configurations are necessary but insufficient. Effective defense against current and future threats also requires the addition of a balancing, outward focus, on understanding the adversary's behavior, capability, and intent. Only through a balanced understanding of both the adversary and ourselves can we hope to understand enough about the true nature of the threats we face to make intelligent defensive decisions.

Today's evolving threat environment also brings with it far more complex attack scenarios. Alongside commoditized threats, more advanced capabilities that were rare in the past are now commonplace. Adversary behavior is not solely focused on widespread, disruptive activity, such as the Storm worm outbreak of years gone by, but rather it often involves more targeted, lower-profile multi-stage attacks that aim to achieve specific tactical objectives and establish a persistent foothold into our enterprises.

This newer attack scenario, and how to defend against it, can be effectively understood from the defensive perspective of a "kill chain"⁶ showing the multiple steps in an attack. As shown below, the adversary's attack unfolds in a series of steps, ending with the attacker having an established foothold in the victim's network. This is the modus operandi of today's sophisticated advanced persistent threat, more commonly known as the APT. APT actors are typically assumed to be nation states but the same behaviors can also be exhibited by those engaged in conducting cyber-crime, financial threats, industrial espionage, hacktivism, and terrorism.

Figure 8. Kill Chain Phases

The APT's desire and apparent capability to persist and cause ongoing damage is motivating the need to move beyond today's traditional reactive approaches to cyber security and become more proactive. Responding to incidents after the exploit has already occurred is very costly, both in the effective impact and in the level of effort necessary to root out the adversary's established foothold. To be proactive, cyber defenders need to fundamentally change the nature of the game by stopping the adversary's advance, preferably before the exploit stage of the attack illustrated in the kill chain (that is, moving left of the hack). Moving left of the hack requires defenders to evolve from a defensive

⁶ <https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>

strategy based primarily on after-the-fact incident investigation and response to one driven by cyber threat intelligence.

Just as traditional intelligence ascertains an understanding of adversaries' capabilities, actions, and intent, the same value carries over to the cyber domain. Cyber intelligence seeks to understand and characterize things like: what sort of attack actions have occurred and are likely to occur; how can these actions be detected and recognized; how can they be mitigated; who are the relevant threat actors; what are they trying to achieve; what are their capabilities, in the form of tactics, techniques, and procedures (TTP) they have leveraged over time and are likely to leverage in the future; what sort of vulnerabilities, misconfigurations, or weaknesses are they likely to target; what actions have they taken in the past; etc.

A holistic understanding of the threat posed by the adversary enables more effective decision support, prioritization of courses of action and a potential opportunity to fundamentally affect the balance of power between the defender and the adversary.

“The effect of intelligence-driven CND [computer network defense] is a more resilient security posture. APT actors, by their nature, attempt intrusion after intrusion, adjusting their operations based on the success or failure of each attempt. In a kill chain model, just one mitigation breaks the chain and thwarts the adversary, therefore any repetition by the adversary is a liability that defenders must recognize and leverage.”

“Through this model, defenders can develop resilient mitigations against intruders and intelligently prioritize investments in new technology or processes.”

“If defenders implement countermeasures faster than adversaries evolve, it raises the costs an adversary must expend to achieve their objectives. This model shows, contrary to conventional wisdom, such aggressors have no inherent advantage over defenders.”

Hutchins, E.M. et al. [Hutchins10]

Cyber threat intelligence itself poses a challenge in that no organization in and of itself has access to an adequate scope of relevant information for accurate situational awareness of the threat landscape. The way to overcome this limitation is via sharing of relevant cyber threat information among trusted partners and communities. Through information sharing, each sharing partner can potentially achieve a more complete understanding of the threat landscape not only in the abstract but also at the instantiable level of what specifics they can look for to find the attacker. The threat that organization A is facing today may very well be one that organization B will face tomorrow. This is especially true when A and B both fall within the targeting scope of a given adversary's campaign. If organization A can share their personal instantiable knowledge of what they learned/saw about a given threat in the form of a cyber threat indicator, organization B may be able to take advantage of this knowledge to address the threat while it's attack on them is still left of the hack (that is, pre-exploit).

Given the evolving complexities of the threat landscape, the speed at which events occur, and the vast quantities of data involved in cyber threat intelligence and threat

information sharing, establishing automation to aid human analysis or execute defensive actions at machine-speed is a prerequisite for any effective approach. Automation will require a feed of quality information and most defensive capabilities will typically be built not from homogenous architectures but rather from a diverse set of differing products and systems. The combination of all of these factors will require standardized, structured threat information representations so the widest possible community of information sources can contribute and leverage information without knowing ahead of time who will be providing what information.

One of the challenges threat-sharing organizations face is the ability to structure cyber threat information, yet not lose the human judgment and control involved in sharing. In many cases, organizations have a desire to exchange information in a way that is both human-readable as well as machine-parse able.

This requirement is largely an artifact of many information sharing programs where organizations consume not just the data but also assess the data as part of an intelligence collection process. This intelligence process is largely driven by human intelligence analysts that are focused on types of analysis that are either inappropriate for automation or focused on making decisions that require a human-in-the-loop where the analyst directly benefits from reading threat information for situational awareness and context. In addition, because of the wide range in quality of the shared threat information, the intelligence analyst is often also assessing the fidelity based upon the sources and methods used to produce the threat information.

Given all of these factors, there exists a need for structured representations of threat information that are expressive, flexible, extensible, automatable and readable. This paper outlines a community-driven solution to this problem known as the Structured Threat Information eXpression, or STIX.

3.2. CURRENT APPROACHES

3.2.1 Indicators of Compromise (IoC)

A piece of information that can be used to search for or identify potentially compromised systems.

Here we introduce some of the main reasons why sharing IoC is recommended and interesting:

- Faster access to actionable security information, often peer / industry relevant.
- Causes the threat actors to change infrastructure more frequently.
- Builds trust relationships between organizations.
- Supports an Intelligence Driven security model.

Examples:

- IP Address / Domain Name
- URL
- File Hash
- Email Address
- X-Mailer
- HTTP User Agent
- File Mutex

3.2.2 OPENIOC

Open Indicators of Compromise⁷ by Mandiant. OpenIOC is an extensible XML schema for the description of technical characteristics that identify a known threat, an attacker's methodology, or other evidence of compromise.

- Pros:
 - Free (Apache 2 license)
 - Extensible, can be extended as needed
 - Free IOC Editor software to create OpenIOC indicators
- Full support in Mandiant products
- Cons:
 - Limited commercial adoption (outside of Mandiant)
 - Limited Network based IoC support
 - Generic Network String needed to cover many IoCs
 - Viewed as a "vendor" solution
- No support for describing Tactics, Techniques, and Procedures

3.2.3 IODEF

Incident Object Description Exchange Format. IETF Standard (RFC 5070⁸) Combined with RFC 5901 (Phishing) for IoC usage

The Incident Object Description Exchange Format (IODEF) defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents.

- Pros
 - Open Standard through IETF.
 - Vendor neutral in origin.
- Commercial adoption (ArcSight).
- Cons
 - Limited native IoC descriptive capability.
 - Requires RFC 5901 and / or extensions.
 - Requires other formats to describe TTPs or campaigns.
 - Designed to share Incident data, not IoCs.
- Dated (last updated in 2007) but being update as we speak.

3.2.4 STIX™

STIX™ is relatively new, but the practice of cyber threat information sharing, particularly indicators, is not. The information being managed and exchanged today is

⁷ <http://www.openioc.org/>

⁸ www.ietf.org/rfc/rfc5070.txt

typically very atomic, inconsistent, and very limited in sophistication and expressivity. Where standardized structures are used, they are typically focused on only an individual portion of the overall problem, do not integrate well with each other, or lack coherent flexibility. Many existing indicator sharing activities are human-to-human exchanges of unstructured or semi-structured descriptions of potential indicators, conducted via web-based portals or encrypted email. A more recent trend is the machine-to-machine transfer of relatively simple sets of indicator data fitting already well-known attack models. Efforts fitting this description include the Research and Education Networking Information Sharing and Analysis Center's (REN-ISAC) Security Event System and its Collective Intelligence Framework component, the state of Washington's Public Regional Information Security Event Management (PRISEM), the Department of Energy's Cyber Federated Model (CFM), and CERT.FI's and CERT.EE's AbuseHelper.

STIX, however, aims to extend indicator sharing to enable the management and widespread exchange of significantly more expressive sets of indicators as well as other full-spectrum cyber threat information.

Currently, automated management and exchange of cyber threat information is typically tied to specific security product lines, service offerings, or community-specific solutions. STIX™ will enable the sharing of comprehensive, rich, "high-fidelity" cyber threat information across organizational, community, and product/service boundaries.

Due to the current widespread adoption of STIX, we leverage from STIX™ as a de facto standard which at the same time we selected as one of the main building blocks for our contribution. Because of that, we will be describing STIX™ and its evolution in details in the following sections.

3.3. PYRAMID OF PAIN

IoC are mainly ephemeral if the threat actor wants to persist in its attack.

As in [Bianco14] the author explain how easy is for an attacker to persist if our protection is based on specific IoC (e.g. IP addresses, hashes, domain names, etc.). A simple change can be done by the attacker to change the IP, hash, domain, etc. but it is not easy to change the complete pattern or Techniques, Tactics and Procedures (TTP).

As a result, new proposals like the one we present in this work [Riesco19a] suggest the idea to work in the dimension of TTP as a combination of intelligence data, malicious behaviors and patterns to detect and better mitigate threats.

3.4. STIX™

STIX™ Structure Threat Information Exchange originally evolved out of discussions among the security operations and cyber threat intelligence experts on the IDXWG email list (established by members of US-CERT and CERT.org in 2010 to discuss automated data exchange for cyber incidents) regarding the development of a standardized representation for cyber threat indicators. Out of these discussions a rough structured threat information architecture diagram was created.

The original purpose of this architecture diagram was to clearly define the scope of what sorts of information should be included within a structured cyber threat indicator and what sorts of information should be defined in other related structures. This architecture diagram helped to clarify scope such that initial cuts at a structured representation for cyber threat indicators could be successfully drafted. As the concept and initial structure for cyber threat indicators matured, there was increasing interest from numerous parties in fleshing-out the rest of the structured threat information architecture. An XML Schema implementation of the full STIX™ architecture, incorporating the cyber threat indicator

representation among others, is the result of those discussions and the vehicle for current ongoing development of the STIX™ language among a broad and dynamically growing community.

STIX™ is a language, being developed in collaboration with any and all interested parties, for the specification, capture, characterization and communication of standardized cyber threat information. It does so in a structured fashion to support more effective cyber threat management processes and application of automation.

A variety of high-level cyber security use cases rely on such information including:

- Analyzing cyber threats
- Specifying indicator patterns for cyber threat
- Managing cyber threat response activities
- Sharing cyber threat information

STIX™ provides a common mechanism for addressing structured cyber threat information across and among this full range of use cases improving consistency, efficiency, interoperability, and overall situational awareness.

In addition, STIX™ provides a unifying architecture tying together a diverse set of cyber threat information including:

- Cyber Observables.
- Indicators.
- Incidents.
- Adversary Tactics, Techniques, and Procedures (including attack patterns, malware, exploits, kill chains, tools, infrastructure, victim targeting, etc.).
- Exploit Targets (e.g., vulnerabilities, weaknesses or configurations).
- Courses of Action (e.g., incident response or vulnerability/weakness remedies or mitigations).
- Cyber Attack Campaigns.
- Cyber Threat Actors.

To enable such an aggregate solution to be practical for any single use case STIX™ is both flexible and extensible. Existing standardized languages may be leveraged as optional extensions where appropriate and numerous flexibility mechanisms are designed into the language. In particular, almost everything in this definitively-structured language is optional such that any single use case could leverage only the portions of STIX™ that are relevant for it (from a single field to the entire language or anything in between) without being overwhelmed by the rest. Specific subsets of STIX™ capabilities can be defined; and agreed to beforehand in the form of profiles for use within sharing communities, by tools, etc.

3.4.1 Use Cases

STIX™ is targeted to support a range of core use cases involved in cyber threat management. Very simple overviews of these use cases are provided below:

Figure 9. Some STIX™ use cases

3.4.2 Analyzing Cyber Threats

A cyber threat analyst reviews structured and unstructured information regarding cyber threat activity from a variety of manual or automated input sources. The analyst seeks to understand the nature of relevant threats, identify them, and fully characterize them such that all of the relevant knowledge of the threat can be fully expressed and evolved over time. This relevant knowledge includes threat-related actions, behaviors, capabilities, intents, attributed actors, etc. From this understanding and characterization the analyst may then specify relevant threat indicator patterns, suggest courses of action for threat response activities, and/or share the information with other trusted parties. For example, in the case of a potential phishing attack, a cyber threat analyst may analyze and evaluate a suspected phishing email, analyze any email attachments and links to determine if they are malicious, determine if the email was sent to others, assess commonality of who/what is being targeted in the phishing attack, determine whether malicious attachments were opened or links followed, and keep a record of all analysis performed.

3.4.2.1. Specifying Indicator Patterns for Cyber Threats

A cyber threat analyst specifies measurable patterns representing the observable characteristics of specific cyber threats along with their threat context and relevant metadata for interpreting, handling, and applying the pattern and its matching results. This may be done manually or with the assistance of automated tooling and structured instantiable threat information. For example, in the case of a confirmed phishing attack, a cyber threat analyst may harvest the relevant set of observables (e.g., to or from addresses, actual source, subject, embedded URLs, type of attachments, specific attachment, etc.) from the performed analysis of the phishing email, identify the relevant TTPs exhibited in the phishing attack, perform kill chain correlation of the attack, assign appropriate confidence for the indicator, determine appropriate handling guidance, generate any relevant automated rule patterns for the indicator (e.g. Snort, YARA, OVAL, etc.), assign any suggested courses of action, and package it all up as a coherent record for sharing (UC4, below) and future reference.

3.4.2.2. Managing Cyber Threat Response Activities

Cyber decision makers and cyber operations personnel work together to prevent or detect cyber threat activity and to investigate and respond to any detected incidences of such activity. Preventative courses of action may be remedial in nature to mitigate vulnerabilities, weaknesses, or misconfigurations that may be targets of exploit. After detection and investigation of specific incidents, reactive courses of action may be pursued. For example, in the case of a confirmed phishing attack with defined indicators, cyber decision makers and cyber operations personnel work together to fully understand the effects of the phishing attack within the environment including malware installed or malware executed, to assess the cost and efficacy of potential courses of action, and to implement appropriate preventative or detective courses of action.

3.4.2.2.1 CYBER THREAT PREVENTION

Cyber decision makers evaluate potential preventative courses of action for identified relevant threats and select appropriate actions for implementation. Cyber operations personnel implement selected courses of action in order to prevent the occurrence of specific cyber threats whether through general prophylactic application of mitigations or through specific targeted mitigations initiated by predictive interpretation of leading indicators. For example, in the case of a confirmed phishing attack with defined indicators, a cyber decision maker may evaluate a suggested preventative course of action (e.g., implementing a blocking rule at the email gateway) defined within an indicator for the phishing attack, determine its relevant cost and risk, and decide whether or not to implement it. If it is decided to implement the suggested course of action, cyber operations personnel carry out the implementation.

3.4.2.2.2 CYBER THREAT DETECTION

Cyber operations personnel apply mechanisms (both automated and manual) to monitor and assess cyber operations in order to detect the occurrence of specific cyber threats whether in the past through historical evidence, currently ongoing through dynamic situational awareness, or a priori through predictive interpretation of leading indicators. This detection is typically via cyber threat indicator patterns. For example, in the case of a confirmed phishing attack with defined indicators, cyber operations personnel may harvest any specified observable patterns from defined indicators of the attack and apply them appropriately within the operational environment to detect any evidence of the phishing attack occurring.

3.4.2.2.3 INCIDENT RESPONSE

Cyber operations personnel respond to detections of potential cyber threats, investigate what has occurred or is occurring, attempt to identify and characterize the nature of the actual threat, and potentially carry out specific mitigating or corrective courses of action. For example, in the case of a confirmed phishing attack, cyber operations personnel may conduct investigative activities to determine whether the phishing attack was successful in carrying out negative effects within the target environment (e.g., was malware installed or run) and if so, attempt to characterize in detail those effects (e.g., which systems were affected by malware, what data was exfiltrated, etc.). Once the effects are understood, cyber operations personnel would implement appropriate mitigating or corrective courses of action (e.g. wipe and restore systems, block exfiltration channels, etc.).

3.4.3 Guiding Principles

In its approach to defining a structured representation for cyber threat information the STIX™ effort strives to adhere to and implement a core set of guiding principles that community consensus has deemed necessary. These principles are as follows:

3.4.3.1. Expressivity

In order to fully support the diversity of threat-relevant use cases within the cyber security domain STIX™ is targeted to provide aggregated expressive coverage across all of its targeted use cases rather than specifically targeting one or two. STIX™ is intended to provide full expressivity for all relevant information within the cyber threat domain.

3.4.3.2. Integrate Rather Than Duplicate

Wherever the scope of STIX™ encompasses structured information concepts for which adequate and available consensus standardized representations already exist, the default approach is to provide appropriate mechanisms to integrate these representations into the overall STIX™ architecture rather than attempt to unnecessarily duplicate them.

STIX™ directly leverages the following constituent schema:

- Cyber Observable eXpression (CybOX™)⁹

Version 1.1 of STIX™ provides loose-coupling mechanisms and default implementations for leveraging the following constituent schemas as appropriate:

- Common Attack Pattern Enumeration and Classification (CAPEC™)¹⁰
- Malware Attribute Enumeration and Characterization (MAEC™)¹¹
- Common Vulnerability Reporting Framework (CVRP)¹²
- OASIS Customer Information Quality (CIQ) xPRL¹³

3.4.3.3. Flexibility

In order to support a wide range of use cases and information of varying levels of fidelity, STIX™ is intentionally designed to offer as much flexibility as possible. STIX™ adheres to a policy of allowing users to employ any portions of the standardized representation that are relevant for a given context and avoids mandatory features wherever possible.

3.4.3.4. Extensibility

In order to support a range of use cases with potentially differing representation details and to ensure ease of community-driven refinement and evolution of the language, the STIX™ design intentionally builds in extension mechanisms for domain specific use, for localized use, for user-driven refinement and evolution, and for ease of centralized refinement and evolution.

3.4.3.5. Automatability

⁹ <http://docs.oasis-open.org/cti/cybox/v2.1.1/cybox-v2.1.1-part01-overview.html>

¹⁰ <https://capec.mitre.org/>

¹¹ <https://maecproject.github.io/>

¹² <https://oasis-open.github.io/csaf-documentation/>

¹³ <http://docs.oasis-open.org/ciq/v3.0/cs02/specs/ciq-specs-v3-cs2.html>

The STIX™ design approach intentionally seeks to maximize structure and consistency to support machine-processable automation.

3.4.3.6. Readability

The STIX™ design approach intentionally seeks for content structures to not only be machine-consumable and processable but also to, as much as possible, be human-readable. This human readability is necessary for clarity and comprehensibility during the early stages of development and adoption, and for sustained use in diverse environments going forward.

3.4.4 Architecture

The STIX™ architecture diagram below identifies the core cyber threat concepts as independent and reusable constructs and characterizes all of their interrelationships based on the inherent meaning and content of each. Connecting arrows between construct icons indicate relationships in the form of content elements within the construct at the root of the connecting arrow. That is of the conceptual type of the construct at the head of the connecting arrow and is suggested but not required to utilize the specific STIX™ implementation of that construct. The bracketed asterisk on each of the arrow labels implies that each relationship may exist zero to many times. The structured content of each construct is fleshed out in detail within the language implementation (currently in the form of an XML Schema). The eight core constructs—Observable, Indicator, Incident, TTP, ExploitTarget, CourseOfAction, Campaign and ThreatActor— along with a cross-cutting Data Markings construct are briefly characterized below:

Figure 10. Characterization of STIX: core constructs and main relationships.

3.4.5 STIX™ V1.1 Structure

3.4.5.1. Observables

Observables are the “base” construct within the STIX™ architecture. Observables are stateful properties or measurable events pertinent to the operation of computers and networks. Information about a file (name, hash, size, etc.), a registry key value, a service being started, or an HTTP request being sent are all simple examples of observables. STIX™ leverages CybOX for its representation of Observables.

CybOX is a language for encoding and communicating standardized high-fidelity information about cyber observables, whether dynamic events or stateful measures properties that are observable in the operational cyber domain. CybOX, like STIX, is not targeted at a single cyber security use case but rather is intended to be flexible enough to offer a common solution for all cyber security use cases requiring the ability to deal with cyber observables. It is also intended to be flexible enough to allow both the high-fidelity description of instances of cyber observables that have been measured in an operational context as well as more abstract patterns for potential observables that may be targets for observation and analysis a priori.

Figure 11. STIX™ Cyber Observable object

3.4.5.2. Indicators

Indicators convey specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context. They consist of one or more Observable patterns potentially mapped to a related TTP context and adorned with other relevant metadata on things like confidence in the indicator’s assertion, handling restrictions, valid time windows, likely impact, sightings of the indicator, structured test mechanisms for detection, related campaigns, suggested courses of action, related indicators, the source of the Indicator, etc.

Recognizing limitations in current standardized approaches of representation, STIX™ leverages community knowledge and best practices to define a new Indicator structure for representing Indicator information.

Figure 12. STIX™ Indicators object

3.4.5.3. Incidents

Incidents are discrete instances of Indicators affecting an organization along with information discovered or decided during an incident response investigation. They consist of data such as time-related information, parties involved, assets affected, impact assessment, related Indicators, related Observables, leveraged TTP, attributed Threat Actors, intended effects, nature of compromise, response Course of Action requested, response Course of Action taken, confidence in characterization, handling guidance, source of the Incident information, log of actions taken, etc.

Recognizing limitations in current standardized approaches of representation, STIX™ leverages community knowledge and best practices to define a new Incident structure for representing Incident information.

Figure 13. STIX™ Incidents object

3.4.5.4. Tactics, Techniques, and Procedures (TTP)

TTPs are representations of the behavior or modus operandi of cyber adversaries. It is a term taken from the traditional military sphere and is used to characterize what an adversary does and how they do it in increasing levels of detail. For instance, to give a

simple example, a tactic may be to use malware to steal credit card credentials. A related technique (at a lower level of detail) may be to send targeted emails to potential victims, which have documents attached containing malicious code which executes upon opening, captures credit card information from keystrokes, and uses http to communicate with a command and control server to transfer information. A related procedure (at a lower level of detail) may be to perform open source research to identify potentially gullible individuals, craft a convincing socially engineered email and document, create malware/exploit that will bypass current antivirus detection, establish a command and control server by registering a domain called mychasebank.org, and send mail to victims from a Gmail account called accounts-mychasebank@gmail.com.

TTPs consist of the specific adversary behavior (attack patterns, malware, exploits) exhibited, resources leveraged (tools, infrastructure, personas), information on the victims targeted (who, what or where), relevant Exploit Targets being targeted, intended effects, relevant kill chain phases, handling guidance, source of the TTP information, etc.

TTPs play a central role in cyber threat information and cyber threat intelligence. They are relevant for Indicators, Incidents, Campaigns, and Threat Actors. In addition, they hold a close relationship with Exploit Targets that characterize the specific targets that the TTPs seek to exploit.

Recognizing a lack of current standardized approaches, STIX™ leverages community knowledge and best practices to define a new TTP structure for representing TTP information.

However, portions of the TTP construct utilize defined extension points to enable leveraging of other existing standardized approaches for detailed characterization of things like behaviors in the form of attack patterns and malware and resources in the form of tools and infrastructure. CAPEC may be utilized for structured characterization of TTP attack patterns. MAEC may be utilized for structured characterization of TTP malware. CybOX is utilized for characterization of tools and infrastructure.

Figure 14. STIX™ TTP object

3.4.5.5. Campaigns

Campaigns are instances of Threat Actors pursuing an intent, as observed through sets of Incidents and/or TTP, potentially across organizations. In a structured sense, Campaigns may consist of the suspected intended effect of the adversary, the related TTP leveraged within the Campaign, the related Incidents believed to be part of the Campaign,

attribution to the Threat Actors believed responsible for the Campaign, other Campaigns believed related to the Campaign, confidence in the assertion of aggregated intent and characterization of the Campaign, activity taken in response to the Campaign, source of the Campaign information, handling guidance, etc.

Recognizing a lack of current standardized approaches, STIX™ leverages community knowledge and best practices to define a new Campaign structure for representing Campaign information.

Figure 15. STIX™ Campaign object

3.4.5.6. Threat Actors

Threat Actors are characterizations of malicious actors (or adversaries) representing a cyber-attack threat including presumed intent and historically observed behavior. In a structured sense, Threat Actors consist of a characterization of identity, suspected motivation, suspected intended effect, historically observed TTP used by the ThreatActor, historical Campaigns believed associated with the ThreatActor, other Threat Actors believed associated with the ThreatActor, handling guidance, confidence in the asserted characterization of the ThreatActor, source of the ThreatActor information, etc.

Recognizing a lack of current standardized approaches, STIX™ leverages community knowledge and best practices to define a new ThreatActor structure for representing ThreatActor information.

Figure 16. STIX™ Threat Actor object

3.4.5.7. Exploit Targets

Exploit Targets are vulnerabilities or weaknesses in software, systems, networks or configurations that are targeted for exploitation by the TTP of a ThreatActor. In a structured sense, Exploit Targets consist of vulnerability identifications or characterizations, weakness identifications or characterizations, configuration identifications or characterizations, potential Courses of Action, source of the ExploitTarget information, handling guidance, etc.

Recognizing a lack of current standardized approaches for generalized characterizations, STIX™ leverages community knowledge and best practices to define a new ExploitTarget structure for representing ExploitTarget information. However, portions of the ExploitTarget structure utilize defined extension points to enable leveraging of other existing standardized approaches for characterizing things like vulnerabilities, weaknesses, and configurations. The identifier constructs from the Common Vulnerabilities and Exposures (CVE®) and the Open Source Vulnerability Database (OSVDB) are utilized for identification of publicly disclosed vulnerabilities. The Common Vulnerability Reporting Framework (CVRF) format may be utilized for detailed structured characterization of vulnerabilities not identified in CVE or OSVDB including the potential for characterizing 0-day vulnerabilities. The identifier construct from the Common Weakness Enumeration (CWE™) is utilized for identification of weaknesses. The identifier construct from the Common Configuration Enumeration (CCE™) is utilized for identification of configuration issues.

Figure 17. STIX™ Exploit Target object

3.4.5.8. Courses of Action (CoA)

CoursesOfAction are specific measures to be taken to address threat whether they are corrective or preventative to address Exploit Targets, or responsive to counter or mitigate the potential impacts of Incidents. In a structured sense, COA consist of their relevant stage in cyber threat management (e.g., remedy of an ExploitTarget or response to an Incident), type of COA, description of COA, objective of the COA, structured representation of the COA (e.g., IPS rule or automated patch/remediation), the likely impact of the COA, the likely cost of the COA, the estimated efficacy of the COA, observable parameters for the COA, handling guidance, etc.

Recognizing a lack of current standardized approaches for generalized characterizations, STIX™ leverages community knowledge and best practices to define a new COA structure for representing Courses of Action information.

Figure 18. STIX™ Course of action object

3.4.5.9. Data Markings

A consistent requirement across many of the core STIX™ constructs is the ability to represent markings of the data to specify things like handling restrictions given the potentially sensitive nature of many forms of cyber threat information. This construct is not conveyed as a separate entity in the STIX™ architecture diagram but exists as a cross-cutting structure within all of the top-level constructs described above. There currently exists no broad consensus standardized approach for such markings but rather various approaches within differing communities and driven by different motivations and usage contexts. Rather than adopting a single marking approach (e.g., Traffic Light Protocol (TLP¹⁴) and attempting to force everyone else to accept it, STIX™ has taken a flexible and generic approach.

The Data Markings structure defined for STIX™ is flexible in two ways. First, instances are specified independent of the structures being tagged (pointing to their locations) rather than embedded everywhere which is typically less efficient and more difficult to update and refine. Second, it allows the definition and use of any data marking structure simply as an abstraction from a base type structure. This allows varying marking schemes to be leveraged and when combined with the independent specification described above it can also easily enable marking any given data structure with multiple different marking schemes for leveraging within different use cases or by different communities. The initial implementation of this Data Marking structure has been created utilizing XML Schema (XSD) and has the potential to be leveraged not only throughout an XML implementation of STIX™ but also for any other XML-based structured representation.

¹⁴ <http://www.us-cert.gov/tlp/>

Figure 19. STIX™ Data markings object

3.4.6 STIX™ V1.1 Implementations

The initial implementation for STIX™ is utilizing **XML Schema** as a ubiquitous, portable and structured mechanism for detailed discussion, collaboration and refinement among the communities involved. It also provides a practical structure for real-world prototyping and proof-of-concept implementations in structured threat information management and sharing. This sort of real-world usage of the language will be encouraged and supported through the development of various supporting utilities such as programmatic language bindings, representation translation transforms, APIs, etc. Only through appropriate levels of collaborative iteration among a relevant community of experts and vetted through real-world data and use cases can a practical and effective solution evolve.

The structure for the language is also planned to be abstracted into an implementation-independent specification. This will then enable other potential implementations to be derived including possibilities such as **semantic web (RDF/OWL)**, **JSON-centric**, ProtocolBuffers, etc.

3.4.7 STIX™ V2 Structure

STIX™ v2 is JSON-centric instead of XML of STIX™ v1.1. Although STIX™ v1.1 whitepaper was suggesting future orientation to use semantics by RDF/OWL or at least improving XML by JSON, they decided in this iteration to improve XML by the latter (JSON) without semantics (neither JSON-LD).

Our thesis is on the other hand providing using the other proposal mentioned above of semantic web (RDF/OWL).

We propose the other path, the semantics path by using semantics (OWL) as it will be one of our main contributions in this thesis to demonstrate how STIX™ 2 could fill the gap with regard to expressiveness, inference and reasoning.

STIX™ Objects categorize each piece of information with specific attributes to be populated. Chaining multiple objects together through relationships allow for easy or complex representations of CTI. Below is a list of what can be represented through STIX.

STIX™ 2 defines **twelve** STIX™ Domain Objects (SDOs) and **two** STIX™ Relationship Objects (SROs). In addition STIX™ 2 includes Cyber Observables as an integration of Cyber Observable eXpression (CybOX™) objects which are now called STIX™ Cyber Observables

STIX™ 2 improves from XML to JSON to facilitate better understanding beyond machines.

Figure 20. STIX™ 2 twelve objects and relationships

Figure 21. STIX™ 2 legend of relationship types between objects

3.4.7.1. Attack Pattern

A type of Tactics, Techniques, and Procedures (TTP) that describe ways that adversaries attempt to compromise targets. Attack Patterns are used to help categorize attacks, generalize specific attacks to the patterns that they follow, and provide detailed information about how attacks are performed. An example of an attack pattern is "spear phishing": a common type of attack where an attacker sends a carefully crafted e-mail message to a party with the intent of getting them to click a link or open an attachment to deliver malware. Attack Patterns can also be more specific; spear phishing as practiced by a particular threat actor (e.g., they might generally say that the target won a contest) can also be an Attack Pattern.

The Attack Pattern SDO contains textual descriptions of the pattern along with references to externally-defined taxonomies of attacks such as CAPEC [CAPEC]. Relationships from Attack Pattern can be used to relate it to what it targets (Vulnerabilities and Identities) and which tools and malware use it (Tool and Malware).

Figure 22. STIX™ 2 Attack pattern object and relationships

```

{
  "type": "attack-pattern",
  "id": "attack-pattern--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
  "modified": "2016-05-12T08:17:27.000Z",
  "name": "Spear Phishing",
  "description": "...",
  "external_references": [
    {
      "source_name": "capec",
      "external_id": "CAPEC-163"
    }
  ]
}

```

Example 1. A generic Attack pattern for spear phishing, referencing CAPEC

3.4.7.2. Campaign

A Campaign is a grouping of adversarial behaviors that describes a set of malicious activities or attacks (sometimes called waves) that occur over a period of time against a specific set of targets. Campaigns usually have well defined objectives and may be part of an Intrusion Set.

Campaigns are often attributed to an intrusion set and threat actors. The threat actors may reuse known infrastructure from the intrusion set or may set up new infrastructure specific for conducting that campaign.

Campaigns can be characterized by their objectives and the incidents they cause, people or resources they target, and the resources (infrastructure, intelligence, Malware, Tools, etc.) they use.

For example, a Campaign could be used to describe a crime syndicate's attack using a specific variant of malware and new C2 servers against the executives of ACME Bank

during the summer of 2016 in order to gain secret information about an upcoming merger with another bank.

Figure 23. STIX™ 2 Campaign object and relationships

```

{
  "type": "campaign",
  "id": "campaign--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:00.000Z",
  "modified": "2016-04-06T20:03:00.000Z",
  "name": "Green Group Attacks Against Finance",
  "description": "Campaign by Green Group against a series of targets in the financial services sector."
}
  
```

Example 2. A generic Campaign against Financial services.

3.4.7.3. Course of Action (CoA)

Note: The Course of Action object in STIX™ 2.0 is a stub. It is included to support basic use cases (such as sharing prose courses of action) but does not support the ability to represent automated courses of action or contain properties to represent metadata about courses of action. Future STIX™ 2 releases will expand it to include these capabilities

A Course of Action is an action taken either to prevent an attack or to respond to an attack that is in progress. It may describe technical, automatable responses (applying patches, reconfiguring firewalls) but can also describe higher level actions like employee training or policy changes. For example, a course of action to mitigate a vulnerability could describe applying the patch that fixes it.

The Course of Action SDO contains a textual description of the action; a reserved action property also serves as placeholder for future inclusion of machine automatable

courses of action. Relationships from the Course of Action can be used to link it to the Vulnerabilities or behaviors (Tool, Malware, and Attack Pattern) that it mitigates.

Figure 24. STIX™ 2 Course of action object and relationships

```
[
{
  "type": "course-of-action",
  "id": "course-of-action--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "name": "Add TCP port 80 Filter Rule to the existing Block UDP 1434 Filter",
  "description": "This is how to add a filter rule to block inbound access to TCP port 80 to the existing UDP 1434 filter ..."
},
{
  "type": "relationship",
  "id": "relationship--44298a74-ba52-4f0c-87a3-1824e67d7fad",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:07:10.000Z",
  "modified": "2016-04-06T20:07:10.000Z",
  "relationship_type": "mitigates",
  "source_ref": "course-of-action--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "target_ref": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b"
},
{
  "type": "malware",
  "id": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:07:09.000Z",
  "modified": "2016-04-06T20:07:09.000Z",
  "name": "Poison Ivy"
}
]
```

Example 3. A Course of Action (CoA) to mitigate a specific malware connection

3.4.7.4. Identity

Identities can represent actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, or groups (e.g., the finance sector).

The Identity SDO can capture basic identifying information, contact information, and the sectors that the Identity belongs to. Identity is used in STIX™ to represent, among other things, targets of attacks, information sources, object creators, and threat actor identities.

Figure 25. STIX™ 2 Identity object and relationships

```

{
  "type": "identity",
  "id": "identity--023d105b-752e-4e3c-941c-7d3f3cb15e9e",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:00.000Z",
  "modified": "2016-04-06T20:03:00.000Z",
  "name": "John Smith",
  "identity_class": "individual"
}
  
```

Example 4. An Identity name

3.4.7.5. Indicator

Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity. For example, an Indicator may be used to represent a set of malicious domains and use the STIX™ Patterning Language to specify these domains.

The Indicator SDO contains a simple textual description, the Kill Chain Phases that it detects behavior in, a time window for when the Indicator is valid or useful, and a required pattern property to capture a structured detection pattern. While each structured pattern language has different syntax and potentially different semantics, in general an Indicator is considered to have "matched" (or been "sighted") when the conditions specified in the structured pattern are satisfied in whatever context they are evaluated in.

Relationships from the Indicator can describe the malicious or suspicious behavior that it directly detects (Malware, Tool, and Attack Pattern) as well as the Campaigns, Intrusion Sets, and Threat Actors that it might indicate the presence of.

Figure 26. STIX™ 2 Indicator object and relationships

```
[
{
  "type": "indicator",
  "id": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
```

```

    "modified": "2016-04-06T20:03:48.000Z",
    "labels": ["malicious-activity"],
    "name": "Poison Ivy Malware",
    "description": "This file is part of Poison Ivy",
    "pattern": "[file:hashes.'SHA-256'
'4bac27393bdd9777ce02453256c5577cd02275510b2227f473d03f533924f877' ]",
    "valid_from": "2016-01-01T00:00:00Z"
  },
  {
    "type": "relationship",
    "id": "relationship--44298a74-ba52-4f0c-87a3-1824e67d7fad",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:06:37.000Z",
    "modified": "2016-04-06T20:06:37.000Z",
    "relationship_type": "indicates",
    "source_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "target_ref": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b"
  },
  {
    "type": "malware",
    "id": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
    "created": "2016-04-06T20:07:09.000Z",
    "modified": "2016-04-06T20:07:09.000Z",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "name": "Poison Ivy"
  }
]

```

Example 5. An Indicator of a specific file hash belonging to a malware.

3.4.7.6. Intrusion Set

An Intrusion Set is a grouped set of adversarial behaviors and resources with common properties that is believed to be orchestrated by a single organization. An Intrusion Set may capture multiple Campaigns or other activities that are all tied together by shared attributes indicating a common known or unknown Threat Actor. New activity can be attributed to an Intrusion Set even if the Threat Actors behind the attack are not known. Threat Actors can move from supporting one Intrusion Set to supporting another, or they may support multiple Intrusion Sets.

Where a Campaign is a set of attacks over a period of time against a specific set of targets to achieve some objective, an Intrusion Set is the entire attack package and may be used over a very long period of time in multiple Campaigns to achieve potentially multiple purposes.

While sometimes an Intrusion Set is not active, or changes focus, it is usually difficult to know if it has truly disappeared or ended. Analysts may have varying level of fidelity on attributing an Intrusion Set back to Threat Actors and may be able to only attribute it back to a nation state or perhaps back to an organization within that nation state.

Figure 27. STIX™ 2 Intrusion set object and relationships

```

{
  "type": "intrusion-set",
  "id": "intrusion-set--4e78f46f-a023-4e5f-bc24-71b3ca22ec29",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "name": "Bobcat Breakin",
  "description": "Incidents usually feature a shared TTP of a bobcat being released within the building containing network access, scaring users to leave their computers without locking them first. Still determining where the threat actors are getting the bobcats.",
  "aliases": ["Zookeeper"],
  "goals": ["acquisition-theft", "harassment", "damage"]
}
  
```

Example 6 An Intrusion-set example.

3.4.7.7. **Malware**

Note: The Malware object in STIX™ 2.0 is a stub. It is included to support basic use cases but is likely not useful for actual malware analysis or for including even simple malware instance data. Future versions of STIX™ 2 will expand it to include these capabilities.

Malware is a type of TTP that is also known as malicious code and malicious software, and refers to a program that is inserted into a system, usually covertly, with the

intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim. Malware such as viruses and worms are usually designed to perform these nefarious functions in such a way that users are unaware of them, at least initially.

The Malware SDO characterizes, identifies, and categorizes malware samples and families via a text description property. This provides detailed information about how the malware works and what it does. Relationships from Malware can capture what the malware targets (Vulnerability and Identity) and link it to another Malware SDO that it is a variant of.

Figure 28. STIX™ 2 Malware object and relationships

```
{
  "type": "malware",
  "id": "malware--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
  "modified": "2016-05-12T08:17:27.000Z",
  "name": "Cryptolocker",
  "description": "...",
  "labels": ["ransomware"]
}
```

Example 7. An example of how a malware is described.

3.4.7.8. Observed Data

Observed Data conveys information that was observed on systems and networks using the Cyber Observable specification defined in parts 3 and 4 of this specification. For example, Observed Data can capture the observation of an IP address, a network connection, a file, or a registry key. Observed Data is not an intelligence assertion, it is simply information: this file was seen, without any context for what it means.

Observed Data captures both a single observation of a single entity (file, network connection) as well as the aggregation of multiple observations of an entity. When the `number_observed` property is 1 the Observed Data is of a single entity. When the `number_observed` property is greater than 1, the observed data consists of several instances of an entity collected over the time window specified by the `first_observed` and `last_observed` properties. When used to collect aggregate data, it is likely that some fields in the Cyber Observable Object (e.g., timestamp fields) will be omitted because they would differ for each of the individual observations.

Observed Data may be used by itself (without relationships) to convey raw data collected from network and host-based detection tools. A firewall could emit a single Observed Data instance containing a single Network Traffic object for each connection it sees. The firewall could also aggregate data and instead send out an Observed Data instance every ten minutes with an IP address and an appropriate `number_observed` value to indicate the number of times that IP address was observed in that window.

Observed Data may also be related to other SDOs to represent raw data that is relevant to those objects. The Sighting object, which captures the sighting of an Indicator, Malware, or other SDO, uses Observed Data to represent the raw information that led to the creation of the Sighting (e.g., what was actually seen that suggested that a particular instance of malware was active).

Figure 29. STIX™ 2 Observed data object

```
{
  "type": "observed-data",
  "id": "observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T19:58:16.000Z",
  "modified": "2016-04-06T19:58:16.000Z",
  "first_observed": "2015-12-21T19:00:00Z",
  "last_observed": "2015-12-21T19:00:00Z",
  "number_observed": 50,
  "objects": {
    "0": {
      "type": "file",
      ...
    }
  }
}
```

Example 8. Observed data of a file object

3.4.7.9. Report

Reports are collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details. They are used to group related threat intelligence together so that it can be published as a comprehensive cyber threat story.

The Report SDO contains a list of references to SDOs and SROs (the CTI objects included in the report) along with a textual description and the name of the report.

For example, a threat report produced by ACME Defense Corp. discussing the Glass Gazelle campaign should be represented using Report. The Report itself would contain the narrative of the report while the Campaign SDO and any related SDOs (e.g., Indicators for the Campaign, Malware it uses, and the associated Relationships) would be referenced in the report contents.

Figure 30. STIX™ 2 Report object

```
{
  "type": "report",
  "id": "report--84e4d88f-44ea-4bcd-bbf3-b2c1c320bcb3",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
  "created": "2015-12-21T19:59:11.000Z",
  "modified": "2015-12-21T19:59:11.000Z",
  "name": "The Black Vine Cyberespionage Group",
  "description": "A simple report with an indicator and campaign",
  "published": "2016-01-20T17:00:00.000Z",
  "labels": ["campaign"],
  "object_refs": [
    "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
    "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
    "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a"
  ]
}
```

Example 9. A standalone Report; the consumer may or may not already have access to the referenced STIX™ Objects.

```
{
  "type": "bundle",
  "id": "bundle--44af6c39-c09b-49c5-9de2-394224b04982",
  "objects": [
    {
      "type": "identity",
      "id": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
      ...,
      "name": "Acme Cybersecurity Solutions"
    },
  ],
}
```

```

{
  "type": "report",
  "id": "report--84e4d88f-44ea-4bcd-bbf3-b2c1c320bcbd",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
  "created": "2015-12-21T19:59:11.000Z",
  "modified": "2016-05-21T19:59:11.000Z",
  "name": "The Black Vine Cyberespionage Group",
  "description": "A simple report with an indicator and campaign",
  "published": "2016-01-20T17:00:00Z",
  "labels": ["campaign"],
  "object_refs": [
    "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
    "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
    "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a"
  ]
},
{
  "type": "indicator",
  "id": "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
  "created": "2015-12-21T19:59:17.000Z",
  "modified": "2016-05-21T19:59:17.000Z",
  "name": "Some indicator",
  "labels": ["malicious-activity"],
  "pattern": "[ file:hashes.MD5 = '3773a88f65a5e780c8dff9cdc3a056f3' ]",
  "valid_from": "2015-12-21T19:59:17Z",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283"
},
{
  "type": "campaign",
  "id": "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
  "created": "2015-12-21T19:59:17.000Z",
  "modified": "2016-05-21T19:59:17.000Z",
  "name": "Some Campaign"
},
{
  "type": "relationship",
  "id": "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
  "created": "2015-12-21T19:59:17.000Z",
  "modified": "2015-12-21T19:59:17.000Z",
  "source_ref": "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
  "target_ref": "campaign--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
  "relationship_type": "indicates"
}
}

```

Example 10. A Bundle with a Report and the STIX™ Objects that are referred to by the Report.

3.4.7.10. Threat Actor

Threat Actors are actual individuals, groups, or organizations believed to be operating with malicious intent. A Threat Actor is not an Intrusion Set but may support or be affiliated with various Intrusion Sets, groups, or organizations over time.

Threat Actors leverage their resources, and possibly the resources of an Intrusion Set, to conduct attacks and run Campaigns against targets.

Threat Actors can be characterized by their motives, capabilities, goals, sophistication level, past activities, resources they have access to, and their role in the organization.

Figure 31. STIX™ 2 Threat Actor object and relationships

```
{
  "type": "threat-actor",
  "id": "threat-actor--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "labels": [ "crime-syndicate" ],
  "name": "Evil Org",
  "description": "The Evil Org threat actor group",
  "aliases": [ "Syndicate 1", "Evil Syndicate 99" ],
  "roles": "director",
  "goals": [ "Steal bank money", "Steal credit cards" ],
  "sophistication": "advanced",
  "resource_level": "team",
  "primary_motivation": "organizational-gain"
}
```

Example 11 An example of a Threat Actor description.

3.4.7.11. Tool

Tools are legitimate software that can be used by threat actors to perform attacks. Knowing how and when threat actors use such tools can be important for understanding how campaigns are executed. Unlike malware, these tools or software packages are often found on a system and have legitimate purposes for power users, system administrators, network administrators, or even normal users. Remote access tools (e.g., RDP) and

network scanning tools (e.g., Nmap) are examples of Tools that may be used by a Threat Actor during an attack.

The Tool SDO characterizes the properties of these software tools and can be used as a basis for making an assertion about how a Threat Actor uses them during an attack. It contains properties to name and describe the tool, a list of Kill Chain Phases the tool can be used to carry out, and the version of the tool.

This SDO MUST NOT be used to characterize malware. Further, Tool MUST NOT be used to characterise tools used as part of a course of action in response to an attack. Tools used during response activities can be included directly as part of a Course of Action SDO.

Figure 32. STIX™ 2 Tool object and relationships

```

{
  "type": "tool",
  "id": "tool--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "labels": [ "remote-access" ],
  "name": "VNC"
}

```

3.4.7.12. Vulnerability

A Vulnerability is "a mistake in software that can be directly used by a hacker to gain access to a system or network" [CVE¹⁵]. For example, if a piece of malware exploits CVE-2015-12345, a Malware object could be linked to a Vulnerability object that references CVE-2015-12345.

The Vulnerability SDO is primarily used to link to external definitions of vulnerabilities or to describe 0-day vulnerabilities that do not yet have an external definition. Typically, other SDOs assert relationships to Vulnerability objects when a specific vulnerability is targeted and exploited as part of malicious cyber activity. As such, Vulnerability objects can be used as a linkage to the asset management and compliance process.

Figure 33. STIX™ 2 Vulnerability object and relationships

```
{
  "type": "vulnerability",
  "id": "vulnerability--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
```

¹⁵ <http://cve.mitre.org/>

```
"modified": "2016-05-12T08:17:27.000Z",
"name": "CVE-2016-1234",
"external_references": [
  {
    "source_name": "cve",
    "external_id": "CVE-2016-1234"
  }
]
```

Example 13 An example of the description of Vulnerability CVE-2016-1234

3.4.7.13. Relationship

The Relationship object is used to link together two SDOs in order to describe how they are related to each other. If SDOs are considered "nodes" or "vertices" in the graph, the Relationship Objects (SROs) represent "edges".

STIX™ defines many relationship types to link together SDOs. These relationships are contained in the "Relationships" table under each SDO definition. Relationship types defined in the specification SHOULD be used to ensure consistency. An example of a specification-defined relationship is that an indicator indicates a campaign. That relationship type is listed in the Relationships section of the Indicator SDO definition.

STIX™ also allows relationships from any SDO to any SDO that have not been defined in this specification. These relationships MAY use the related-to relationship type or MAY use a custom relationship type. As an example, a user might want to link malware directly to a tool. They can do so using related-to to say that the Malware is related to the Tool but not describe how, or they could use delivered-by (a custom name they determined) to indicate more detail.

Note that some relationships in STIX™ may seem like "shortcuts". For example, an Indicator doesn't really detect a Campaign: it detects activity (Attack Patterns, Malware, etc.) that are often used by that campaign. While some analysts might want all of the source data and think that shortcuts are misleading, in many cases it's helpful to provide just the key points (shortcuts) and leave out the low-level details. In other cases, the low-level analysis may not be known or sharable, while the high-level analysis is. For these reasons, relationships that might appear to be "shortcuts" are not excluded from STIX.

Source	Type	Target	Source	Type	Target
attack-pattern	targets	vulnerability	intrusion-set	attributed-to	threat-actor
attack-pattern	targets	identity	intrusion-set	targets	identity
attack-pattern	uses	malware	intrusion-set	targets	vulnerability
attack-pattern	uses	tool	intrusion-set	uses	attack-pattern
campaign	attributed-to	intrusion-set	intrusion-set	uses	malware
campaign	attributed-to	threat-actor	intrusion-set	uses	tool
campaign	targets	identity	malware	targets	identity
campaign	targets	vulnerability	malware	targets	vulnerability
campaign	uses	attack-pattern	malware	uses	tool
campaign	uses	malware	malware	variant-of	malware
campaign	uses	tool	threat-actor	attributed-to	identity
course-of-action	mitigates	attack-pattern	threat-actor	impersonates	identity
course-of-action	mitigates	malware	threat-actor	targets	identity
course-of-action	mitigates	tool	threat-actor	targets	vulnerability
course-of-action	mitigates	vulnerability	threat-actor	uses	attack-pattern
indicator	indicates	attack-pattern	threat-actor	uses	malware
indicator	indicates	campaign	threat-actor	uses	tool
indicator	indicates	intrusion-set	tool	targets	identity
indicator	indicates	malware	tool	targets	vulnerability
indicator	indicates	threat-actor			
indicator	indicates	tool			

Figure 34. STIX™ 2 relationship summary table.

3.4.7.14. Sighting

A Sighting denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen. Sightings are used to track who and what are being targeted, how attacks are carried out, and to track trends in attack behavior.

The Sighting relationship object is a special type of SRO; it is a relationship that contains extra properties not present on the generic Relationship object. These extra properties are included to represent data specific to sighting relationships (e.g., count, representing how many times something was seen), but for other purposes a Sighting can be thought of as a Relationship with a name of "sighting-of". Sighting is captured as a relationship because you cannot have a sighting unless you have something that has been sighted. Sighting does not make sense without the relationship to what was sighted.

Sighting relationships relate three aspects of the sighting:

- What was sighted, such as the Indicator, Malware, Campaign, or other SDO (sighting_of_ref)
- Who sighted it and/or where it was sighted, represented as an Identity (where_sighted_refs) and

- What was actually seen on systems and networks, represented as Observed Data (observed_data_refs)

What was sighted is required; a sighting does not make sense unless you say what you saw. Who sighted it, where it was sighted, and what was actually seen are optional. In many cases it is not necessary to provide that level of detail in order to provide value.

Sightings are used whenever any SDO has been "seen". In some cases, the object creator wishes to convey very little information about the sighting; the details might be sensitive, but the fact that they saw a malware instance or threat actor could still be very useful. In other cases, providing the details may be helpful or even necessary; saying exactly which of the 1000 IP addresses in an indicator were sighted is helpful when tracking which of those IPs is still malicious.

Sighting is distinct from Observed Data in that Sighting is an intelligence assertion ("I saw this threat actor") while Observed Data is simply information ("I saw this file"). When you combine them by including the linked Observed Data (observed_data_refs) from a Sighting, you can say "I saw this file, and that makes me think I saw this threat actor". Although confidence is currently reserved, notionally confidence would be added to Sighting (the intelligence relationship) but not to Observed Data (the raw information).

```
[ {
  "type": "sighting",
  "id": "sighting--ee20065d-2555-424f-ad9e-0f8428623c75",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:08:31.000Z",
  "modified": "2016-04-06T20:08:31.000Z",
  "first_seen": "2015-12-21T19:00:00Z",
  "last_seen": "2015-12-21T19:00:00Z",
  "count": 50,
  "sighting_of_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "observed_data_refs": ["observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf"],
  "where_sighted_refs": ["identity--b67d30ff-02ac-498a-92f9-32f845f448ff"]
},
{
  "type": "observed-data",
  "id": "observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T19:58:16.000Z",
  "modified": "2016-04-06T19:58:16.000Z",
  "start": "2015-12-21T19:00:00Z",
  "stop": "2016-04-06T19:58:16Z",
  "count": 50,
  "objects": {
    "0": {
      "type": "file",
      ...
    }
  }
}
}]
```

Example 14. Sighting with observed data (what exactly was seen) and where it was seen.

3.4.7.15. Cyber Observables

The STIX™ 2.0 specification defines structured representations for observable objects and their properties in the cyber domain. These can be used to describe data in many different functional domains, including but not limited to:

- Malware characterization

- Intrusion detection
- Incident response & management
- Digital forensics

STIX™ Cyber Observables document the facts concerning what happened on a network or host, but not necessarily the who or when, and never the why. For example, information about a file that existed, a process that was observed running, or that network traffic occurred between two IPs can all be captured as Cyber Observable data.

STIX™ Cyber Observables are used by various STIX™ Domain Objects (SDOs) to provide additional context to the data that they characterize. The Observed Data SDO, for example, indicates that the raw data was observed at a particular time and by a particular party.

The Cyber Observable Objects chosen for inclusion in STIX™ 2.0 represent a minimally viable product (MVP) that fulfills basic consumer and producer requirements. Objects and properties **not included in STIX™ 2.0, but deemed necessary by the community, will be included in future releases.**

For more details about the STIX™ V2 cyber-observables see 0.
--

3.4.7.16. Patterning

In order to **detect a large proportion of malicious behavior in the course of defending our networks**, it is necessary to correlate telemetry from both host-based and network-based tools.

Before undertaking work on STIX™ Patterning, OASIS technical subcommittee made a thorough effort to evaluate whether there was already an existing patterning language that would support STIX™ use cases available as an open standard. In particular, they considered whether it would be possible to extend the syntax of Snort or Yara rather than create an entirely new language. This was eventually ruled out as unfeasible, both from a technical perspective as well as taking into consideration that from a licensing/IPR perspective, extending either of those languages under the auspices of OASIS would have been problematic.

Given that STIX™ Patterning exists to support STIX™ Indicators, consider what value Indicator-sharing provides: a mechanism for communicating how to find malicious code and/or threat actors, active within a given network. Among the essential tools widely deployed by defenders are SIEMs (or similar data processing platforms capable of consuming, correlating, and interrogating large volumes of network and host-based telemetry.) **These data processing platforms utilize proprietary query languages.** As development began on STIX™ Patterning, one of the principal design goals was to create an abstraction layer capable of serializing these proprietary correlation rules so as to enhance the overall value proposition of indicator-sharing.

In order to enhance detection of possibly malicious activity on networks and endpoints, **a standard language is needed to describe what to look for in a cyber environment.** The STIX™ Patterning language allows matching against timestamped Cyber Observable data (such as STIX™ Observed Data Objects) collected by a threat intelligence platform or other similar system so that other analytical tools and systems can be configured to react and handle incidents that might arise.

This first language release is focused on supporting a common set of use cases and therefore allows for the expression of an initial set of patterns that producers and consumers of STIX™ can utilize. **As more complex patterns are deemed necessary, the STIX™**

patterning language will be extended in future releases to improve its effectiveness as an automated detection/remediation method.

STIX™ Patterns are composed of multiple building blocks, ranging from simple key-value comparisons to more complex, context-sensitive expressions. The most fundamental building block is the Comparison Expression, which is a comparison between a single property of a Cyber Observable Object and a given constant using a Comparison Operator.

As a simple example, one might use the following Comparison Expression (contained within an Observation Expression) to match against an IPv4 address:

```
[ipv4-addr:value = '198.51.100.1/32']
```

Moving up a level of complexity, the next building block of a STIX™ Pattern is the Observation Expression, which consists of one or more Comparison Expressions joined by Boolean Operators and bounded by square brackets. An Observation Expression refines which set of Cyber Observable data (i.e., as part of an Observation) will match the pattern, by selecting the set that has the Cyber Observable Objects specified by the Comparison Expressions. An Observation Expression consisting of a single Comparison Expression is the most basic valid STIX™ Pattern. Building upon the previous example, one might construct an Observation Expression to match against multiple IPv4 addresses and an IPv6 address:

```
[ipv4-addr:value = '198.51.100.1/32' OR ipv4-addr:value = '203.0.113.33/32' OR ipv6-addr:value = '2001:0db8:dead:beef:dead:beef:dead:0001/128']
```

Observation Expressions may be followed by one or more Qualifiers, which allow for the expression of further restrictions on the set of data matching the pattern. Continuing with the above example, one might use a Qualifier to state that the IP addresses must be observed several times in repetition:

```
[ipv4-addr:value = '198.51.100.1/32' OR ipv4-addr:value = '203.0.113.33/32' OR ipv6-addr:value = '2001:0db8:dead:beef:dead:beef:dead:0001/128'] REPEAT 5 TIMES
```

The final, highest level building block of STIX™ Patterning combines two or more Object Expressions via Observation Operators, yielding a STIX™ Pattern capable of matching across multiple STIX™ Observed Data SDOs. Building further upon our previous example, one might use an Observation Operator to specify that an observation of a particular domain name must follow the observation of the IP addresses (note the use of parentheses to encapsulate the two Observation Expressions), along with a different Qualifier to state that both the IP address and domain name must be observed within a specific time window:

```
[ipv4-addr:value = '198.51.100.1/32' OR ipv4-addr:value = '203.0.113.33/32' OR ipv6-addr:value = '2001:0db8:dead:beef:dead:beef:dead:0001/128'] FOLLOWEDBY ([domain-name:value = 'example.com']) WITHIN 600 SECONDS
```

The diagram below depicts a truncated version of the various STIX™ Patterning components in the above example:

Figure 35. STIX™ 2 Patterning components

Pattern Expressions evaluate to true or false. They comprise one or more Observation Expressions joined by Observation Operators. Pattern Expressions are evaluated against a set of specific Observations. If one or more of those Observations match the Pattern Expression, then it evaluates to true. If no Observations match, the Pattern Expression evaluates to false.

For example, consider the following Pattern Expression:

```
[(type-a:property-j = 'W' AND type-a:property-k = 'X') OR (type-b:property-m = 'Y' AND type-b:property-n = 'Z')]
```

Applying those to specific matchings, hereby we introduce some concrete examples:

Example of Matching on Network Traffic with a particular destination:

```
[network-traffic:dst_ref.type = 'ipv4-addr' AND network-traffic:dst_ref.value = '203.0.113.33/32']
```

Example of Matching on a Domain Name with IPv4 Resolution:

```
[domain-name:value = 'www.5z8.info' AND domain-name:resolves_to_refs[*].value = '198.51.100.1/32']
```

Example of Matching on an Email Message with specific Sender and Subject:

```
[email-message:sender_ref.value = 'jdoe@example.com' AND email-message:subject = 'Conference Info']
```

Patterning is equivalent to a rule language trying to foster STIX™ taxonomy into the claimed expressivity and standardization. On the other hand, it is still semantic agnostic.

Rules are then applied for “matching”, but they are not applied to infer new knowledge which we consider a great opportunity, that is the main reason of the current thesis proposal to leverage STIX™ into its semantic version, based on OWL as well as to leverage patterning rules into SWRL (semantic web rule language).

3.4.7.17. Bundle

A Bundle is a collection of arbitrary STIX™ Objects and Marking Definitions grouped together in a single container. A Bundle **does not have any semantic meaning** and Objects are not considered related by virtue of being in the same Bundle.

Bundle is not a STIX™ Object, so it does not have any of the Common Properties other than the type and id properties. Bundle is not a STIX™ Object and must not have any relationships to it or from it

Bundle is transient and implementations should not assume that other implementations will treat it as a persistent object.

The JSON MTI serialization uses the JSON object when representing bundle.

3.4.7.18. Data Marking

Data markings represent restrictions, permissions, and other guidance for how data can be used and shared. For example, **data may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest.**

In STIX, data markings are specified using the marking-definition object. These definitions are applied to complete STIX™ Objects using object markings and to individual properties of STIX™ Objects via granular markings.

Some types of marking definitions or trust groups have rules about which markings override other markings or which markings can be additive to other markings. STIX™ v2 does not define rules for how multiple markings applied to the same object or property should be interpreted.

3.4.7.18.1 Marking Definition

The marking-definition object represents a specific marking. Data markings typically represent handling or sharing requirements for data, and are applied in the `object_marking_refs` and `granular_markings` properties on STIX™ Objects, which reference a list of IDs for marking-definition objects.

Two marking definition types are defined in this specification: TLP, to capture TLP markings, and Statement, to capture text marking statements. In addition, it is expected that the FIRST Information Exchange Policy (IEP) will be included in a future version once a machine-usable specification for it has been defined.

Unlike STIX™ Objects, Marking Definition objects cannot be versioned because it would allow for indirect changes to the markings on a STIX™ Object. For example, if a Statement marking is changed from "Reuse Allowed" to "Reuse Prohibited", all STIX™ Objects marked with that Statement marking would effectively have an updated marking without being updated themselves. Instead, a new Statement marking with the new text should be created and the marked objects updated to point to the new marking.

The JSON MTI serialization uses the JSON object type (RFC7159¹⁶) when representing marking-definition

STATEMENT MARKING OBJECT TYPE

The Statement marking type defines the representation of a textual marking statement (e.g., copyright, terms of use, etc.) in a definition. The value of the `definition_type` property must be `statement` when using this marking type. Statement markings are

¹⁶ <https://tools.ietf.org/html/rfc7159>

generally not machine-readable and this specification does not define any behavior or actions based on their values.

Content may be marked with multiple Statement marking types that do not override each other. In other words, the same content can be marked both with a statement saying "Copyright 2016" and a statement saying "Terms of use are ..." and both statements apply.

```
{
  "type": "marking-definition",
  "id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",
  "created": "2016-08-01T00:00:00.000Z",
  "definition_type": "statement",
  "definition": {
    "statement": "Copyright 2016, Example Corp"
  }
}
```

Example 15 Statement marking.

TLP MARKING OBJECT TYPE

The TLP marking type defines how you would represent a Traffic Light Protocol (TLP) marking in a definition property. The value of the definition_type property MUST be tlp when using this marking type.

```
{
  "type": "marking-definition",
  "id": "marking-definition--613f2e26-407d-48c7-9eca-b8e91df99dc9",
  "created": "2017-01-20T00:00:00.000Z",
  "definition_type": "tlp",
  "definition": {
    "tlp": "white"
  }
}
```

Example 16. TLP WHITE.

```
{
  "type": "marking-definition",
  "id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",
  "created": "2017-01-20T00:00:00.000Z",
  "definition_type": "tlp",
  "definition": {
    "tlp": "green"
  }
}
```

Example 17. TLP GREEN.

```
{
  "type": "marking-definition",
  "id": "marking-definition--f88d31f6-486f-44da-b317-01333bde0b82",
  "created": "2017-01-20T00:00:00.000Z",
  "definition_type": "tlp",
  "definition": {
    "tlp": "amber"
  }
}
```

Example 18. TLP AMBER.

```
{
  "type": "marking-definition",
  "id": "marking-definition--5e57c739-391a-4eb3-b6be-7d15ca92d5ed",
  "created": "2017-01-20T00:00:00.000Z",
  "definition_type": "tlp",
  "definition": {
    "tlp": "red"
  }
}
```

Example 19. TLP RED.

3.4.7.18.2 Object Markings

Object Markings apply data markings to an entire STIX™ Object or Marking Definition and all of its contents. Object Markings are specified as embedded relationships in the `object_marking_refs` property, which is an optional list of IDs for marking-definition objects. The referenced markings apply to that STIX™ Object or Marking Definition and all of its contents. Changes to the `object_marking_refs` property (and therefore the markings applied to the object) are treated the same as changes to any other properties on the object and follow the same rules for versioning.

```
{
  "type": "indicator",
  "id": "indicator--b346b4b3-f4b7-4235-b659-f985f65f0009",
  ...
  "object_marking_refs": ["marking-definition--089a6ecb-cc15-43cc-9494-767639779123"],
  ...
}
```

Example 20. This example marks the Indicator and all its properties with the Marking Definition referenced by the ID

3.4.7.18.3 Granular Markings

Whereas object markings apply to an entire STIX™ Object or Marking Definition and all its properties, granular markings allow data markings to be applied to individual portions of STIX™ Objects and Marking Definitions. Granular markings are specified in the `granular_markings` property, which is a list of granular-marking instances. Each of those instances contains a list of selectors to indicate what is marked and a reference to the marking-definition object to be applied. Granular markings can be used, for example, to indicate that the name property of an indicator should be handled as TLP: GREEN, the description property as TLP: AMBER, and the pattern property as TLP: RED.

```
{
  ...
  "granular_markings": [
    {
      "marking_ref": "marking-definition--089a6ecb-cc15-43cc-9494-767639779123",
      "selectors": ["description", "labels"]
    }
  ],
  "description": "Some description",
  "name": "Some name",
  "labels": ["first", "second"]
}
```

Example 21. Granular marking considering marking ref and selectors.

3.4.8 STIX™ V1.1 and V2 Comparison

3.4.8.1. One Standard

First, the OASIS Cyber Threat Intelligence (CTI) Technical Committee (TC) decided to merge the two specifications into one. Cyber Observable eXpression (CybOX™) objects are now called STIX™ Cyber Observables.

Now when you discuss “STIX”, you are talking about the one standard needed for sharing cyber threat intelligence!

3.4.8.2. JSON vs. XML

STIX™ 2.0 requires implementations to support JSON serialization, while STIX™ 1.x was defined using XML. Though both XML and JSON have benefits, the CTI TC determined that JSON was more lightweight, and sufficient to express the semantics of cyber threat intelligence information. It is simpler to use and increasingly preferred by developers.

Note: in the current thesis, we will demonstrate that OWL and SWRL will provide real and better semantics than current approach for STIX™ based on JSON.

3.4.8.3. Stix™ Domain Objects

All objects in STIX™ 2 are at the top-level, rather than being embedded in other objects. These objects are called STIX™ Domain Objects (SDO). Some object properties use a reference to another object's id directly (e.g., created_by_ref), but most relationships are expressed using the top-level Relationship object.

The generic TTP (tactics, techniques, procedures) and Exploit Target types from STIX™ 1.X have been split into separate top-level objects (Attack Pattern, Malware, Tool and Vulnerability) with specific purposes in STIX™ 2.

STIX 1 Sample Object

```
<stix:TTPs>
  <stix:TTP id="attack-pattern:ttp-01" xsi:type='ttp:TTPType'
    version="1.1">
    <ttp>Title>Initial Compromise</ttp>Title>
    <ttp:Behavior>
      <ttp:Attack_Patterns>
        <ttp:Attack_Pattern capec_id="CAPEC-163">
          <ttp:Description>Spear Phishing</ttp:Description>
        </ttp:Attack_Pattern>
      </ttp:Attack_Patterns>
    </ttp:Behavior>
  </stix:TTP>
</stix:TTPs>
<stix:TTPs>
  <stix:Kill_Chains>
    <stixCommon:Kill_Chain id="stix:TTP-02"
      name="mandiant-attack-lifecycle-model">
      <stixCommon:Kill_Chain_Phase name="initial-compromise"
        phase_id="stix:TTP-03"/>
    </stixCommon:Kill_Chain_Phase>
  </stix:Kill_Chains>
</stix:TTPs>
```

STIX 2 Sample SDO

```
{
  "type": "attack-pattern",
  "id": "attack-pattern--01",
  "created": "2015-05-15T09:11:12.515000Z",
  "modified": "2015-05-15T09:11:12.515000Z",
  "name": "Initial Compromise",
  "external_references": [
    {
      "source_name": "capec",
      "description": "spear phishing",
      "external_id": "CAPEC-163"
    }
  ],
  "kill_chain_phases": [
    {
      "kill_chain_name": "mandiant-attack-lifecycle-model",
      "phase_name": "initial-compromise"
    }
  ]
}
```

Figure 36. STIX™ V1.1 Object VS. STIX™ V2 Domain Object

3.4.8.4. Relationships at Top Level Objects

STIX™ 2.0 introduces a top-level Relationship object, which links two other top-level objects via a named relationship type. STIX™ 2 content can be thought of as a connected graph, where nodes are SDOs and edges are Relationship Objects. The STIX™ 2 specification suggests different named relationships, but content producers are able to define their own. In STIX™ 1.X relationships were “embedded” in other objects. The types of relationships supported was restricted by the STIX™ 1.X specification. Because STIX™ 1.X relationships themselves were not top-level objects, you could not express a relationship between two objects without changing one of them. In CTI, it is often desirable for others to assert a relationship. Using this new Relationship object, others, besides the original content creator, can add to the shared knowledge in an independent way.

Sample Relationship

Figure 37. STIX™ 2 Relationship Objects

3.4.8.5. Streamlined Model

Experience with STIX™ 1.X showed that a common set of features were widely used and well understood while many other features lacked shared understanding and had only limited, if any use at all. In addition, almost all properties of objects were optional. Overall, the breadth of STIX™ 1.x was an impediment to sharing intelligence, and necessitated a formal agreement among threat groups on what should be shared (i.e., profiles).

STIX™ 2 takes a different approach – many properties are required, and the number of objects and properties have been reduced to a core set of well understood features. Future releases of STIX™ will incorporate additional concepts as they are needed.

However, the need to incorporate concepts not yet in the specification is enabled through the ability to use custom properties and custom objects.

3.4.8.6. Data Markings

Data markings no longer use a serialization specific language, e.g., XPath. In STIX™ 2, there are two types of data markings: object marking – applicable to a whole object, and granular markings – applicable to a property or properties of an object. Data markings scope is only within the object where they are defined.

3.4.8.7. Indicator Pattern Language

Indicator patterns in STIX™ 1.x were expressed using XML syntax. This made all but the simplest patterns difficult to create and to understand. STIX™ 2.0 takes a different approach, specifying a language for patterns which is independent of the serialization language. Patterns written in the STIX™ patterning language are more compact and easier to read. Additionally, there is no confusion between patterns and observations, because a pattern is not a top-level object, but a property of an indicator object.

STIX 1 Indicator Example

```
<stix:Indicator id="example:indicator-01"
  timestamp="2017-02-09T12:11:11.415000+00:00"
  xsi:type='indicator:IndicatorType'>
  <indicator:Title>HTRAN Hop Point Accessor</indicator:Title>
</stix:Indicator>
<stix:TTPs>
<stix:Kill_Chains>
  <stixCommon:Kill_Chain id="stix:TTP-02"
    name="mandiant-attack-lifecycle-model">
  <stixCommon:Kill_Chain_Phase name="establish-foothold"
    phase_id="stix:TTP-03"/>
  </stix:Kill_Chains>
</stix:TTPs>
<indicator:Observable id="example:Observable-04">
<cybox:Object id="example:Object-05">
  <cybox:Properties xsi:type="AddressObj:AddressObjectType"
    category="ipv4-addr">
  <AddressObj:Address_Value condition="Equals">10.1.0.0/15
  </AddressObj:Address_Value>
</cybox:Object>
</indicator:Observable>
```

STIX 2 Indicator Example with Pattern

```
{
  "type": "indicator",
  "id": "indicator--01",
  "created": "2017-02-09T12:11:11.415000Z",
  "modified": "2017-02-09T12:11:11.415000Z",
  "name": "HTRAN Hop Point Accessor",
  "pattern": "[ipv4-addr:value =
    '10.1.0.0/15']",
  "labels": [ "malicious-activity" ],
  "valid_from": "2015-05-15T09:00:00.000000Z",
  "kill_chain_phases": [
    {
      "kill_chain_name":
        "mandiant-attack-lifecycle-model",
      "phase_name": "establish-foothold"
    }
  ]
}
```

Figure 38. STIX™ V1.1 Indicator vs. STIX™ 2 Indicator pattern

3.4.8.8. Coming Attractions

STIX™ 2.0 establishes a foundation for future development with the intent to add several new SDOs and develop support for other critical CTI concepts. A small number of minor revisions will be developed over the coming months to iteratively mature STIX™ 2. By establishing a foundation that supports the most critical CTI sharing use cases, the CTI TC hopes to gain critical feedback and implementation experience early while also allowing future development of STIX™ to be more focused on a smaller set of selected topics. Work on STIX™ 2.1 is already underway with concepts such as incidents, confidence, internationalization, and malware under active development. At the same time organizations are actively building capabilities that utilize STIX™ 2.0.

It is a widespread industry initiative to make it easier for people and tools to share threat intelligence.

It is then a quickly evolving, collaborative community-driven effort to define and develop a language to represent structured threat information. The STIX™ language is meant to convey the full range of cyber threat information and strives to be fully expressive, flexible, extensible, automatable, and as human-readable as possible. Though relatively new and still evolving, it is actively being adopted or considered for adoption by a wide range of cyber threat-related organizations and communities around the world. All interested parties are welcome to participate in evolving STIX™ as part of its open, collaborative community via the STIX™ web site, email discussion lists and other collaborative forums.

3.5. EXCHANGE OF THREAT INTELLIGENCE INFORMATION

Cybersecurity Information Sharing can be defined as the set of technologies, data formats, messaging protocols, policies, and frameworks that enable the exchange of

cybersecurity-related information between peers or within a community of entities. The Information Sharing Systems are secure platforms designed to efficiently and effectively distribute critical information about offenders, crimes, and cyber-attacks incidents, in order to enhance prevention and apprehension activities by law enforcement.

As far as cyber security is concerned, there exist many different standards and initiatives that attempt to provide solutions for cyber security information exchange. These tools have recently begun to attract interest of information security agencies and organizations, and are gaining more and more acceptance. One might say that Information Sharing has become a fundamental piece in the fight against cyber threats and as a means for building a comprehensive situational awareness. The information may relate to threat intelligence (actors, TTP, campaigns, etc.), malware, vulnerabilities, attack patterns, cyber observables, security incidents, etc.

As a summary, the exchange of information is as useful as the following statement:

“Just a Twitter distribution of a new attack reduces losses of a given attack by 40% in a rural area of Kenya” There are no other means to communicate...

(Serah Francis - Kenya)

3.5.1 The Importance of Information Exchange

In cyber security we've traditionally been focusing on identifying what we want to protect and then building defenses around them along the time. When adversaries breach defenses, organizations adapt themselves to prevent those breaches to occur, and, of course, as we adapt, our adversaries adapt as well.

While it is critical to ensure good enough defenses, it is known that it is not enough. Cyber Threat Intelligence (CTI) is used to better understand, predict and adapt to the behaviors of malicious actors whether they are criminal groups, activist or even nation states.

The collective ability to detect and defend against malicious activity by sharing information about adversaries and their behaviors is paramount.

It is a joint initiative where public and private organizations (including security vendors) should work together to find better ways to create, share and use cyber threat intelligence.

The problem is that using IoC to fight against unknown threats is ineffective. New emerging techniques, tactics and procedures (TTP) remain undetected to any vendor solution whose detection capabilities are based on known threats. As an example, if a specific threat is using a malicious infrastructure under an IP address and our defenses are filtering such IP address to protect us, the threat actor can easily change it frequently to evade our filters along the time. On the other hand, if we decide to fight against such threat still by updating each IP filtering rule along the time with frequent new IP addresses received by peers about the same threat, we could be receiving so much ephemeral and useless data that will only be valid during a short period of time. The same will happen with domains and hashes that can also be mutated easily and frequently by the attacker to remain undetected.

As described in [Bianco14], we will be able to really harm the attacker only if we have the ability to filter the attack by the TTP used.

In order to do that, we will need to have **behavioral and context-aware rules** which are independent of specific IoC. Once we are able to work in such dimension, we will not need to share and store dozens or hundreds of IP addresses, domains and or hashes but the TTP algorithm itself. The idea is equivalent to the usage of DGA¹⁷ (domain generation algorithm) versus the use of single domains, however, we propose to go beyond that. A TTP description can combine a DGA together with any other concepts in the same rule or algorithm as a more effective behavioral pattern. Once such an algorithm is shared, it can be more effective and efficient than sharing hundreds of related IoC if each of them is ephemeral.

At the same time, although several initiatives exist for cybersecurity information exchange, **there is a common understanding about the lack of enough incentives**. There are few original contributors of CTI data in proportion to the consumers of that data. The lack of trust, reliability, and the sensibility of the information to share as well as the type of data to be shared (IoC) are some of the factors that are discouraging.

3.5.2 OSINT: Open Source Intelligence

One of the most promising models for information exchange was Open Source Intelligence (OSINT). Several entities and individuals share cyber threat intelligence data openly along several platforms since lot of time ago. These platforms are based on sharing indicators of compromise (IoC). Most of them allows automatic interaction and processing of the IoC by using their different external interfaces or API (Application Program Interfaces).

Once an IoC is shared by a certain entity or individual, it can be used by anyone who has access to that open platform. The usage can also be done automatically by systems in order to protect the perimeter, as an example, if a Firewall knows a new malicious IP address shared as an IoC, it can be blocked for either incoming or outgoing network traffic. Several tools and plugins are available in order to facilitate that different systems can use and load external threat information in the format of IoC from these platforms.

This promising model based on open source intelligence has demonstrated that is not reliable [OSINTArmy12] [NATO01] but necessary. Organizations can also use deception techniques where they can intentionally or unintentionally add, delete, modify, or otherwise filter the information made to the general public. It is important to evaluate the reliability of open sources in order to distinguish objective, factual information; bias; or deception. Non authoritative sources lack reliability and trustworthiness and seldom stand apart from authoritative sources.

When evaluating sources of information to determine reliability and credibility we should consider:

- **Identity.** Who produced the information (for example a student, teacher, political organization, or reporter)?
- **Authority.** How much does the source know about the information?
- **Motive.** Why was the information published?
- **Access.** Did the source have direct access to the event or information?
- **Timeliness.** What is the date of the information?

¹⁷ <https://blog.malwarebytes.com/security-world/2016/12/explained-domain-generating-algorithm/>

- **Internal and external consistency.** Does the information contradict governmental policies among local citizens?

Most of these characteristics can be better resolved using blockchain built-in features as presented in this work.

Cyber threat intelligence information **should be reliable** by definition if we want actionable intelligence to run automatic security processes. If our systems are automatically loading IoC to build filter rules, we need to trust the entity or individual behind, that is the one who shared that IoC. If not, we can be loading incorrect IoC or even worse, we can help a malicious actor to success. As an example, if a threat actor shares, as malicious, some licit IP addresses or domain names, our systems loading those IoC will automatically block traffic to those nodes causing an outage unless a specific validation process take place. It is then clear that we need to trust OSINT sources [OSINTArmy12] [NATO01] as a necessary condition before loading those IoC in our systems. More than that, maybe it is not enough to trust the entity itself as some IoC shared by them could be part of a third party detection, we then need to trust each piece of data. Specific validation processes as well as sources and evidences' auditing, accounting or checks are recommended. As an example, time-stamping related data to when that malicious behavior was seen, who shared the data, when it was updated or modified or even removed is very interesting for a healthy and up to date system.

On the other hand, OSINT and in general threat information exchange set-up have an asymmetric balance between consumers and producers [Vishik13], that is, most of the entities or individuals are only consumers and only a few are producers. Some of the main reasons is because people is reluctant to share information if there are not enough incentives for them, they do not have the same knowledge about the value of artifacts offered or simply they do not want to share their own data or incidents as they could be giving a competitive advantage to their own competitors. Information exchange was mainly invented to help others by sharing our knowledge at the same time we improve our level of protection thanks to others who share their knowledge. It is clear that no one has enough knowledge to consider fighting alone against any cybersecurity threat. The entities or individuals who are usually sharing intelligence as quid pro quo, or even more; those which are more producers than consumers, are a minority. The incentive for them is mainly the reputation and the image that their brand get from the community along the time.

In [Vishik13] there is a clear understanding that introducing new economic incentives and business models in cybersecurity initiatives as part of an economic modeling would derive in greater societal benefits.

3.5.3 ISAC: Information Sharing and Analysis Centers

Information Sharing and Analysis Centers (ISAC) are non-profit organizations that provide a central resource for gathering information on cyber threats (in many cases to critical infrastructure) as well as allow two-way sharing of information between the private and the public sector about root causes, incidents and threats, as well as sharing experience, knowledge and analysis.

European legislations like the NIS Directive [EC16] and the Cybersecurity Act nourish the creation of sectoral ISACs and PPPs within the EU. The NIS Directive among others separates the operators of essential services in sectors and tasks the operators to implement requirements on incident reporting which is far from a voluntary and enriched approach. It is mainly an obligation to report incidents under certain conditions to their national CSIRTs that could have associated sanctions by competent authorities if reporting and incident handling was not properly done by the entity. This approach is very different from OSINT initiatives but also different from any expected win-win approach plenty of incentives like the ones presented in our work.

The creation of sectorial ISACs at national level could further assist with the implementation of these provisions. During the transposition of such European legislation to national law, these communities could be further informed and advised by policy makers.

3.5.4 Current Status of Information Sharing

There are several standards and proposals aimed at defining the technical aspects of information sharing.

Making Security Measurable (MSM) initiative led by MITRE presents a comprehensive architecture for cyber security measurement and management, where current standards are grouped into processes and mapped to the different knowledge areas, each of which refers to a process (put in parentheses): Asset definition (inventory); Configuration guidance (analysis); Vulnerability alerts (analysis); Threat alerts (analysis); risk/attack Indicators (intrusion detection); and incident Report (management). Next Table relates current MSM standards to these areas:

	CPE	OVAL	SWID	XCCDF	CCE	OCIL	OCSS	CVE	CWE	CVSS	CAPEC	CVRP	MAEC	CyBOX	IndEX	STIX	IODEF	CPE	CEE	RID	RID-T	CYBEX	CWSS
A	•	•	•															•					
C		•		•	•	•	•																
V		•						•	•	•		•											
T								•	•	•	•		•	•	•	•	•		•	•	•		
I	•							•					•	•	•	•	•	•	•	•	•		
R	•	•			•			•	•	•			•		•	•	•			•	•	•	•

Figure 39 Table of MSM (Making Security Measurable) vs. Areas [Hernandez-Ardieta13]

Other remarkable initiatives for Information Sharing are:

- TAXII™ (Trusted Automated eXchange of Indicator Information), which, using STIX language, defines a set of services and message exchanges that, when implemented, enable sharing of actionable cyber threat information across organizational, product line and service boundaries.
- TLP (Traffic Light Protocol) is a set of designations used to ensure that sensitive information is shared with the correct audience.
- CybOX™ (Cyber Observable eXpression) is intended to be a set of fundamental data types for communicating details of campaigns, threat sources, malware characteristics, and other cyber security events that are valuable for a large number of cyber-security use cases. Such use cases are, for example, event management/logging, malware characterization, intrusion detection/prevention, incident response, and digital forensics. CybOX's main objectives are the consistency and the interoperability across this wide range of use cases through providing a common structure and content types. As of this writing, CybOX is an evolving specification.
- STIX™ (Structured Threat Information eXpression) and MAEC (Malware Attribute Enumeration and Characterization) are two composite data structures that are built on top of CybOX. STIX conceptualizes and organizes different traits of threat information into independent and reusable constructs (data types). As of this writing,

STIX is an evolving specification. Currently, STIX is comprised of seven main constructs:

- **Campaign** is typically linked to a series of Incidents and can be linked to specific Threat Actors, since a Campaign typically employs specific TTP and individual Incident can be linked with common set of Indicators. **TTP** stands for tactics, techniques and procedures; it can be used to link a series of Incidents to a specific Campaign.
- **Threat Actor** can be linked to a set of **TTPs**. Incident can be linked to specific TTP used by specific Threat Actor. During incident response, a set of Indicators may be identified. Incident can keep a record of Course of Actions that were taken during incident response.
- **Indicator** is typically linked to a set of relevant cyber observables. An Indicator can be linked to Course of Actions, or to TTPs.
- **Course of Action** characterizes both preventive measures and response measures.
- **Exploit Target** characterizes vulnerabilities or weaknesses that were targeted by specific TTPs. Exploit Target may suggest potential Course of Action, e.g., configuration changes or software upgrade.
- **MAEC** specializes in communicating details of malware. MAEC conceptualizes malware at multiple levels of granularity – mechanism, behavior, action and its implementation – and bridges them through grouping primitives called bundle and package. Unlike traditional virus names that fail to convey deeper technical insight, MAEC tries to capture both behavior and mechanism of malware in a structured manner.

3.5.5 TAXII™: A Oasis Proposal for Information Exchange

As well as STIX™, OASIS has promoted TAXII™ as a de facto standard.

Trusted Automated Exchange of Intelligence Information (TAXII™) is an application protocol for exchanging CTI over HTTPS.

TAXII defines a RESTful API (a set of services and message exchanges) and a set of requirements for TAXII Clients and Servers. As depicted below, TAXII defines two primary services to support a variety of common sharing models:

- **Collection:** a collection is an interface to a logical repository of CTI objects provided by a TAXII Server that allows a producer to host a set of CTI data that can be requested by consumers: TAXII Clients and Servers exchange information in a request-response model.
- **Channel:** Maintained by a TAXII Server, a Channel allows producers to push data to many consumers and consumers to receive data from many producers: TAXII Clients exchange information with other TAXII Clients in subscribe model. Note: The TAXII 2.0 specification reserves the keywords required for Channels but does not specify Channel services. Channels and their services will be defined in a later version of TAXII.

Figure 40. TAXII collection and channels

Collections and Channels can be organized in different ways. For example, they can be grouped to support the needs of a particular trust group.

A TAXII server instance can support one or more API Roots. API Roots are logical groupings of TAXII Channels and Collections and can be thought of as instances of the TAXII API available at different URLs, where each API Root is the “root” URL of that particular instance of the TAXII API.

TAXII relies on existing protocols when possible. In particular, TAXII Servers are discovered within a network via DNS Service records (and/or by a Discovery Endpoint, described in the next section). In addition, TAXII uses HTTPS as the transport for all communications, and it uses HTTP for content negotiation and authentication.

TAXII was specifically designed to support the exchange of CTI represented in STIX, and support for exchanging STIX™ 2.0 content is mandatory to implement.

However, TAXII can also be used to share data in other formats. It is important to note that STIX™ and TAXII are independent standards: the structures and serializations of STIX™ do not rely on any specific transport mechanism, and TAXII can be used to transport non-STIX™ data.

TAXII design principles include minimizing operational changes needed for adoption; easy integration with existing sharing agreements, and support for all widely used threat sharing models: **hub-and-spoke, peer-to-peer, and source-subscriber**.

3.5.5.1. TAXII Discovery Methods

This specification defines two discovery methods. The first is a network level discovery that uses a DNS SRV record [RFC2782]. This DNS SRV record can be used to advertise the location of a TAXII Server within a network (e.g., so that TAXII-enabled security infrastructure can automatically locate an organization’s internal TAXII Server) or to the general Internet.

The second discovery method is a Discovery Endpoint (this specification uses the term Endpoint to identify a URL and an HTTP method with a defined request and response) that enables authorized clients to obtain information about a TAXII Server and get a list of API Roots.

3.5.5.2. TAXII API Roots

API Roots are logical groupings of TAXII Channels, Collections, and related functionality. A TAXII server instance can support one or more API Roots. API Roots can be thought of as instances of the TAXII API available at different URLs, where each API Root is the “root” URL of that particular instance of the TAXII API. Organizing the Channels

and Collections into API Roots allows for a division of content and access control by trust group or any other logical grouping. For example, a single TAXII Server could host multiple API Roots - one API Root for Channels and Collections used by Sharing Group A and another API Root for Channels and Collections used by Sharing Group B.

Each API Root contains a set of Endpoints that a TAXII Client contacts in order to interact with the TAXII Server. This interaction can take several forms:

- Server Discovery, as described above, can be used to learn about the API Roots hosted by a TAXII Server.
- Each API Root might support zero or more Collections. Interactions with Collections include discovering the type of CTI contained in that Collection, pushing new CTI to that Collection, and/or retrieving CTI from that Collection. Each piece of CTI content in a Collection is referred to as an Object.
- Each API Root might host zero or more Channels.
- Each API Root also allows TAXII Clients to check on the Status of certain types of requests to the TAXII Server. For example, if a TAXII Client submitted new CTI, a Status request can allow the Client to check on whether the new CTI was accepted.

Figure 41. TAXII 2.0 API Roots components and relationships

3.5.5.3. Endpoints

An Endpoint consists of a specific URL and HTTP Method on a TAXII Server that a TAXII Client can contact to engage in one, specific type of TAXII exchange. For example, each Collection on a TAXII Server has an Endpoint that can be used to get information about it; TAXII Clients can contact the Collection's Endpoint to request a description of that Collection. A separate Endpoint is used for the TAXII Client to collect a manifest of that Collection's Content. Yet another Endpoint is used to get objects from the Collection and, at the same URL, a POST can be used to add objects to the collection.

3.5.5.4. Collections

A TAXII Collection is an interface to a logical repository of CTI objects provided by a TAXII Server and is used by TAXII Clients to send information to the TAXII Server or

request information from the TAXII Server. A TAXII Server can host multiple Collections per API Root, and Collections are used to exchange information in a request–response manner.

Figure 42 illustrates how Collection based communications are used when a single TAXII Client makes a request to a TAXII Server and the TAXII Server fulfills that request with information available to the TAXII Server (nominally from a database).

Figure 42. TAXII collections and channels

3.5.5.5. Channels

A TAXII Channel is maintained by a TAXII Server and enables TAXII Clients to exchange information with other TAXII Clients in a publish-subscribe model. TAXII Clients can publish messages to Channels and subscribe to Channels to receive published messages. A TAXII Server may host multiple Channels per API Root.

Figure 42 illustrates how Channel communications are used when a single authorized TAXII Client sends a message to the TAXII Server, and that TAXII Server then distributes the message to all authorized TAXII Clients that are connected to the Channel. The arrows in the following diagrams represent data flow.

3.5.5.6. Authentication and Authorization

Access control to an instance of the TAXII API is specific to the sharing community, vendor, or product and is not defined by this specification.

Authentication and Authorization in TAXII is implemented as defined in [RFC7235], using the Authorization and WWW-Authenticate HTTP headers respectively.

HTTP Basic authentication, as defined in [RFC7617] is the mandatory to implement authentication scheme in TAXII. TAXII Servers and Clients are required to implement support for HTTP Basic, though other authentication schemes can also be supported. Implementers can allow operators to disable the use of HTTP Basic in their operations.

If the TAXII Server receives a request for any Endpoint that requires authentication, regardless of HTTP method, and either an acceptable Authorization header that grants the client access to that object is not sent with the request or the TAXII Server does not determine via alternate means that the client is authorized to access the resource, the TAXII Server responds with a HTTP 401 (Unauthorized) status code and a WWW-Authenticate HTTP header.

The WWW-Authenticate header contains one or more challenges, which define which authentication schemes are supported by the TAXII Server. The format of the WWW-Authenticate HTTP header and any challenges are defined in [RFC7235]. To ensure compatibility, it is recommended that any authentication schemes used in challenges be registered in the IANA Hypertext Transfer Protocol (HTTP) Authentication Scheme Registry [HTTP Auth].

A TAXII Server may omit objects, information, or optional fields from any response if the authenticated client is not authorized to receive them, so long as that omission does not violate this specification.

3.5.5.7. Summary of Endpoints, Methods and Resources

As a summary, here we define a summary of the Endpoints (URLs and HTTP Methods) defined by TAXII and the Resources they operate on:

Table 3 TAXII defined Endpoints (URLs and HTTP Methods) and the resources they operate on

URL	Methods	Resource Type (section 2)
Core Concepts		
/taxii/	GET	discovery
<api-root>	GET	api
<api-root>/status/<status-id>/	GET	status
Collections		
<api-root>/collections/	GET	collections
<api-root>/collections/<id>/	GET	collection
<api-root>/collections/<id>/objects/	GET, POST	object*
<api-root>/collections/<id>/objects/<object-id>/	GET	object*
<api-root>/collections/<id>/manifest/	GET	manifest
Channels		
<TBD in a future version>		

```

GET Request
GET .../collections/my-collection/objects/?added_after=2016-02-01T00:00:01.000Z HTTP/1.1
Range: items 0-49
Accept: application/vnd.oasis.stix+json; version=2.0

GET Response
HTTP/1.1 206 Partial Content
Content-Type: application/vnd.oasis.stix+json; version=2.0
X-TAXII-Date-Added-First=2016-02-21T05:01:01.000Z
X-TAXII-Date-Added-Last=2016-02-21T12:01:01.000Z
Content-Range: items 0-49/500

```

Example 22. Client makes a request for items 0-49 (50 items) and the server responds with 0-49.

3.5.5.8. Object Resource

This resource type is negotiated based on the media type. This specification does not define any form of content wrapper for objects. Instead, objects are the direct payload of HTTP messages.

When returning STIX™ 2 content (the Content-Type header contains application/vnd.oasis.stix+json; version=2.0) in a TAXII response, **the root object MUST be a STIX™ bundle.**

For example:

- A single indicator in response to a request for an indicator by ID is enclosed in a bundle.
- A list of campaigns returned from a Collection is enclosed in a bundle.
- An empty response with no STIX™ objects results in an empty bundle.

Definitions for media types other than STIX™ can be found in their respective specifications.

```
{
  "type": "bundle",
  ...,
  "indicators": [
    {
      "type": "indicator",
      "id": "indicator--252c7c11-daf2-42bd-843b-be65edca9f61",
      ...,
    }
  ]
}
```

Example 23. A returned Bundle example.

3.5.6 Research Challenges of Information Sharing

There are many requirements that Information Sharing Systems need to meet. In this section we briefly discuss some of the most important ones provided by NIS WG3 Research and Innovation Landscape deliverable and the ones provided by NIS WG2¹⁸ – Information sharing into NIS WG3:

- *Reusability.* There is a need of a complete security ontology that can be reused and extended, and the security community can trust.
- *Scalability.* There is a need of mechanisms that can automatically share large quantities of information.
- *Storage volume.* The ever growing volume of data stored on the Internet or included in specific products, such as vulnerability repositories, signatures used by anti-virus products, etc., make it difficult to access and eventually share the desired information.
- *Different sources.* The large number of different sources can lead to inconsistencies of the collected and about to be shared information due to the existence of erroneous data.
- *Confidentiality.* There are concerns regarding the confidentiality of exchanged data. Effective means by which redistribution can be satisfactorily controlled must be addressed.
- *Speed.* Cyber-attacks execute and succeed in computer time, so we need information sharing mechanisms that operate within the same order of magnitude. Contrary to this, current approaches for information sharing heavily depend on the human factor. We need to design new frameworks that minimize the human intervention during the information sharing processes. To achieve this, we need to

¹⁸<https://resilience.enisa.europa.eu/nis-platform/shared-documents/5th-plenary-meeting/wg2-chapter-3-presentation/view>

explore other relevant dimensions of the information sharing problem that have been less researched so far, notably those related to deciding on what to share, with whom, when, as well as reasoning about and adapting to the repercussions (risk) of sharing. To meet this in an automated manner we need contributions from different areas, such as graph analysis and trust/reputation. Decision-making support tools for specific scenarios where human intervention is required is also a potential field of research.

- *Flexibility.* The intrinsic flawed and highly vulnerable nature of technology leads us to conclude that absolute trust cannot be achieved in cyber space. We should not rely, in absolute terms, on any information independently of who is the source. This demolishes the requirement that current approaches claim in mandating full trust in the peers that are part of an information sharing community. This situation is aggravated by the fact that we cannot foresee who will have the knowledge needed to prevent or respond to certain incident. Consequently, static, rigid procedure-based information sharing communities will for sure perform ineffectively. We need new approaches where information sharing communities can be dynamically adapted (dynamic composition) based on context, trust, and need-to- share requirements.
- *Information sharing arrangement.* There are so many option: Take informed decision on the arrangement.
- *On sharing between sharing arrangements.* Mutually beneficial for both parties.
- *Creating widely accepted standards.* Specific to the level of sharing.
- *Supplier crises support.* Need for detailed and stringent service level agreements with suppliers with clearly delineated support models for crisis situations.
- *Realistic trust in sharing arrangements.* Trust is the base on which information sharing is built. To foster trust personal interaction is always the key: E.g. periodic exercises and training for expertise sharing
- *Create an ontology and landscape of information sharing.* Today, the diversity of arrangements in information sharing is not sufficiently addressed. Research is needed.

This topic is directly addressed by this thesis together with a proposal of a new incentive model based on blockchain to work over other research topics like Trust and arrangements.

- *Common approach for certain tasks.* Subsidiarity is a well-desired goal of all activities in Europe. The common interest within Europe and a strong position towards global suppliers needs some coordination, helpdesk or reference-function on EU level.

A framework with rules and support of all MS should be elaborated and put into action.

- *Create legal security for information sharing personnel.* Make in-depth legal studies to learn about challenges and limits to information sharing stemming from the law. Use-cases should be elaborated and applied to train information sharing personnel.
- *Provide citizen and SME best practice and victim support.* An often reoccurring incident should be well described, easy to find and foreseen with a solution in electronic form (blog / Database). Today everyone has its secret source.
- *Leverage existing experience from all level of information sharing.* Running information Sharing is a separate task and differs from “Security and Incident Information Sharing”.

As one of our use cases in the validation part of this thesis, we propose to share knowledge in the format of algorithms and any other knowledge which can be represented in a formal way.

- *Close the biggest gaps.* As a policy issue (non-technical), usually, information sharing is instituted for the benefit of the sharing partners, and not for the benefit of larger communities and non-specialized user groups.

As per our proposal, we propose a new incentive model but also the use of ontologies in order we can share algorithms that will be used as plug and play in case that ontology is used between the different peers within the sharing platform.

- *Strengthen security by clear ownership definition of vulnerabilities.* As another policy issue (non-technical), who is the owner of vulnerability and threat information, and who has to act upon it if they have information about vulnerabilities in their possession?
- *Firm screening & verification of shared information from untrusted sources.* It is the last policy issue proposed by NIS WG2 related to Information exchange.

We partially address this topic by using Blockchain in order to control whatever source is providing the contents. Blockchain provides the attribution to the one who owns the private key of the wallet.

In addition we also propose the usage of a marketplace, where rating (however it is optional), can help to verify or provide better screening or filtering by users in case the shared information is useless or even its sharing has associated any malicious intention.

3.5.7 Interesting Information to Share per Level

In most of the cases, while talking about Information Sharing in the cybersecurity arena, we all think about IoC (Indicators of Compromise) or any other technical data, however there can be several and different interesting contents to share depending on the sharing level.

Depending what our interests are, we would share different type of information. Even within the same interest, the type of information can be different depending on the level of the recipients (operational, tactical or strategic).

Some examples can be seen in the Figure 43 Examples of shared contents per sharing level (source: NIS WG2 – Information sharing) provided by NIS WG2¹⁹ of European Commission.

¹⁹ <https://resilience.enisa.europa.eu/nis-platform/shared-documents/5th-plenary-meeting/wg2-chapter-3-presentation/view>

Figure 43 Examples of shared contents per sharing level (source: NIS WG2 – Information sharing)

3.6. CONCLUSIONS

In this chapter, we analyze the state of the art of Cyber Threat Intelligence (CTI).

Different approaches that are used massively around the world are described (see section 3.2). Due to the fact that all current approaches to fight threats are based on known information, the value of the CTI data is usually ephemeral. The concept of Pyramid of Pain [Bianco14] is introduced (see section 3.3). As a promising de facto standard, Structured Threat Information Exchange STIX™ is analyzed in details. Because the specification has been changed along the thesis, we analyze and compare both specifications (v1.1 and v2) (see section 3.4). Despite the lack of semantic expressiveness, STIX™ will be selected for our thesis, not only because of it has been developed to give more expressiveness to previous specifications but also because it has tremendous support by the cybersecurity industry.

The collective ability to detect and defend against malicious activity by sharing information about adversaries and their behaviors is paramount. It is a joint initiative where public and private organizations (including security vendors) should work together to find better ways to create, share and use cyber threat intelligence. Information sharing between national stakeholders but even in cross country cases is one important aspect for cyber security. Knowledge on tackling cyber attacked, incident response, mitigation measures and preparatory controls can be shared between the relevant stakeholders.

Because our DRM Framework must exchange CTI information to provide as much dynamics and risk exposure context (and protection) as possible, we analyze the state of the art of Threat Information Exchange (see section 3.5). Despite there are several and different approaches until today, there are still several research challenges of information sharing today (see Table 1 and section 3.5.6). One of this issues is related **the lack of enough incentives** (there are few original contributors of CTI data in proportion to the consumers of that data). The lack of trust, reliability, and the sensibility of the information to

share as well as the type of data to be shared (IoC) are some of the factors that are discouraging.

CHAPTER 4. ONTOLOGIES AND SWRL TO REPRESENT KNOWLEDGE

4.1. INTRODUCTION

The knowledge contained in the World Wide Web is available in interlinked documents written in natural language. To make use of this knowledge, technologies such as natural language processing, information retrieval, data and knowledge mining must be applied. Semantic Web technologies follow an alternative approach by complementing web documents with explicit semantics based on formal knowledge representations, such as ontologies.

"[...] we are entering a technological age in which we will be able to interact with the richness of living information - not merely in the passive way that we have been accustomed to using books and libraries, but as active participants in an ongoing process, bringing something to it through our interaction with it, and not simply receiving something from it by our connection to it.

[Licklider68]

WWW (World Wide Web) was born at European Nuclear Research Center (CERN) in 1990 due to until that date, the only way to use external data was to connect and download from a remote server to a local storage. It implied different problems:

- Information access required expert knowledge.
- Information access was expensive.
- Information retrieval was very expensive.

From Web 1.0 to Web 2.0:

"Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as platform, and an attempt to understand the rules for success of that new platform"

Tom O'Reilly, 2003

Ontologies are a technique used in computer science in the world of knowledge management and artificial intelligence as a way to formally represent concepts, their meaning and relationships between them. One of the most accepted definitions of the concept of ontology is **"an explicit and formal specification of a shared conceptualization"**:

- It is explicit, because it expressly defines the concepts, properties, relationships, functions, taxonomies, axioms and restrictions or rules that compose it.
- It is formal, because it is specified by a language interpretable by machines.

- It is a conceptualization, because it is an abstract model that supposes a simplified view of the domain that one wants to represent, both of its structure and of what happens in it.
- It is shared, because the information has been previously agreed between different groups of experts.

If two agents, whether people or software agents, decide to communicate through a common ontology, these agents "commit" to it. The ontology shared by both agents must be exactly the same, since otherwise it may lead to so-called ontological imbalances, which are nothing more than misunderstandings caused by the use of different ontologies. To minimize the appearance of these imbalances, the representation of ontology in a formal language is essential, in addition to the fact that the ontology must be shared and, therefore, its development must be a process in which different people collaborate under consensus.

In this way, through different ontologies, intelligent agents could share and reuse knowledge, trying to solve problems of heterogeneity that arise between different specification languages and at different levels: use of different paradigms or formalisms, use of different dialects of the same paradigm, absence of normalization in the knowledge exchange and mismatch protocols at the semantic level in the knowledge bases.

The use of representation through ontologies has gained enormous relevance with the appearance of the Semantic Web, whose main objective is to organize the Web information and formalize it, so that it is interpretable intelligently by applications capable of extracting its meaning. The most typical ontologies for use on the Web, are formed by two elements, a taxonomy and a set of rules of inference:

- The taxonomy defines classes and subclasses of objects and relationships between those objects. It is also responsible for assigning properties to classes.
- The rules of inference add an additional useful capacity for the human user that makes use of the ontology. These rules allow to express restrictions and conditions on the objects of the classes defined by the taxonomy. The way to express restrictions and rules in an ontology language is through axioms. The axioms can be of several kinds, for example, subclass axioms, class equivalence axioms, property restrictions, etc.

In the specific case of the use of ontologies as a basis for CTI and DRM framework, such as the one proposed in this thesis, it has the great added value of being able to semantically integrate the different sources of information from which part to produce intelligence. For this purpose, the languages of knowledge representation or ontology definition languages are used, which will be analyzed later in this chapter.

In addition to the ability to represent knowledge, for CTI and DRM system is also essential the ability to represent reactions to events, so you need languages of specification of behavior that can be interpreted in conjunction with the concepts represented in the ontologies. A review of the most relevant language for representing behavior or defining security policies will be carried out in this chapter of the report.

4.2. ONTOLOGY DEFINITION LANGUAGES

There are numerous languages oriented to the definition of ontologies, which are classified into different paradigms according to their information specification patterns and the terminology used in each of them. The difference between one and the other lies in the elements that they incorporate and the use that is made of them.

The most common elements are the concepts or classes, the copies of these classes and their properties, the facets of the properties (cardinality, universal quantification, existence, etc.), the relationships that can be established between the

classes (inheritance, association, etc.) and the axioms. Below is a brief definition of each of these elements:

- **Concept (class):** refers to the ideas we want to formalize with the ontology. A concept can be anything from an object, such as a table, to something abstract as a complex reasoning process. The formalization of the concept will take into account the domain of application where we will use the ontology. The concepts are formalized through the use of classes.
- **Exemplary (individual):** concrete instance of a class that models a certain concept. It also receives the name of individual. Each specimen is characterized by a set of properties. The diversity of possible values for each of the properties allows distinguishing one instance of a class from others of the same class. Likewise, these properties have facets that allow different degrees of ownership to be provided, such as cardinality, universal quantification, existence, etc.
- **Relationship:** element that represents the interactions that occur between two concepts or classes of the domain. The relationships can be of different nature (inheritance, association, etc.). Relationships are very important because they allow the deduction of new knowledge. A type of special relationship is one that is used to identify specific elements of the ontology through its application on all or part of the elements of the same. This type of relationship is called a function.
- **Axioms:** are truths that do not need proof and that in the world of ontologies are applied on the relationships that meet the elements of an ontology.

The ontology definition languages are classified into three large blocks:

- **Traditional languages:** languages prior to the development of the Semantic Web based on first order logic and frames, among which stand out: KIF (Knowledge Interchange Format), Ontolingua, Loom, OKBC (Open Knowledge Base Connectivity), OCML (Operational Conceptual Modeling Language) and F-Logic (Frame Logic).
- **UML (Unified Modeling Language) and OCL (Object Constraint Language).** The first normalizes a notation and a set of diagrams for the engineering modeling processes of the SW, and due to its great dissemination it has become almost a standard in this field. The interest of using UML as an ontology language derives from its wide dissemination and maturity. To specify restrictions on the information defined by UML, the OCL language was proposed, which is a full-fledged language, characterized by its own syntax.
- **Languages of the Semantic Web:** languages based on markup languages (XML) among which are RDF, RDFS, DAML + OIL, OWL, OWL2, etc.

Due to the advantages of using languages that have not only a formal syntax, but also a formal semantics interpretable by the machines, we pay special attention to the OWL and OWL2 languages.

It is considered that the basis of all the languages of the Semantic Web is the XML language (Extensible Markup Language). XML provides the syntax for building structured documents, but it does not impose any semantic constraints, so we cannot consider it an ontology language. However, XML is a syntactic structuring language that allows univocally identifying the elements of documents through XML namespaces (unique URI and single relative reference), which allows to reuse the elements of documents throughout the Web, and makes possible the creation of unique documents (including the ontologies defined by the languages we will see) with distributed location.

The first ontology languages in this environment appeared with the aim of providing Meta information for the different information elements of web documents (information

referring to the language used, author, related documents, etc.). Today, there is a long way in terms of ontology languages in the Semantic Web, resulting in different proposals and standards by W3C.

The following figure shows the ontology definition languages of the Semantic Web as a stack.

Figure 44 Ontology stack (source: Wikipedia)

4.2.1 SHOE

Simple HTML Ontology Extensions (SHOE) is an ontology language that defines extensions to include annotations in HTML. Although it is based on HTML, there is also a serialization in XML. It was the first labeling language to design ontologies on the web. SHOE combines the characteristics of markup languages, knowledge representation, data logging and ontologies in an attempt to uniquely address the problems of semantics on the Web. The language supports acquisition of knowledge by increasing the web with labels that supply semantics. The basic structure of SHOE are the ontologies, which define the rules that tell us what type of assertions can be made and what kind of conclusions can be derived from those assertions, and the instances, which make the assertions based on those rules.

This language allows us to define classes and rules of inference but not negations or disjunctions, and many publishers, search engines, APIs, etc. were developed. Despite this, this project was abandoned with the development of OIL and DAML (more complete ontology definition languages based on XML).

4.2.2 RDF and RDF-Schema

Resource Description Framework (RDF) is a lightweight ontology specification language used to describe and exchange metadata on the World Wide Web developed by the W3C. Much of the body of knowledge of the Semantic Web is based on this language, since it is the first language that incorporates a formal semantics, which can be understood by machines.

The semantics of RDF are very simple, based on triplets (subject-predicate-object). Using the triplet, a relationship between a subject and an object is specified through a property. The subject is what is being described, the predicate is the property or relationship that one wishes to establish about the subject, and the object is the value of the property. Subject, property and object are identified by means of a URI (Uniform Resource Identifier). Objects, in addition to elements identified by URIs, can be literal, that is, values with their associated data type. Resource is any element that can be identified by a URI, for example, web pages, elements of an XML document, etc. The same resource can be used in some triplets as a subject, in others as an object, and in others even as property. The conjunction of triplets referred to a specific set of resources forms an RDF graph, can be understood as RDF ontology.

Figure 45 RDF Description (source: Dlib.org)

The fact of incorporating a formal semantics allows reasoning with language to infer information. However, the simple semantics of RDF limits this reasoning, so in the specific case of RDF the reasoning has little applicability. In addition, the triplet structure that uses this language is not sufficiently expressive.

Resource Description Framework Schema, RDFS, is an ontology definition language that emerges as an extension of RDF. The first version was published in April 1998 by W3C, publishing the current version in February 2004.

An RDFS file has a very similar syntax (based on XML) and the same structure as an RDF file, so in principle it could be considered as another RDF model. However, the semantics defined for this new RDF model is totally different, which makes RDFS a new language to specify other knowledge domains. The peculiarity of this new language is that the information models based on it can also make use of RDF, that is, both languages can coexist at the same level. This is because the RDFS semantics extends RDF.

RDFS allows you to define additional restrictions on resources and RDF properties. For example, you define the class resource and the type property, which allow you to define instances and the classes to which they belong. The subclassOf properties are also created, to specify class inheritance; range and domain to specify classes such as range and domain of a property; subpropertyOf to define specialization of properties, etc. Using these new

elements of the RDFS language, the RDF vocabularies are built, which we can consider as ontologies.

4.2.3 OIL and DAML+OIL

Ontology Inference Layer or Ontology Interchange Language is an ontology definition language belonging to the Semantic Web, which was promoted by the On-To-Knowledge of the European Union. Like most languages of the Semantic Web, it uses the XML language syntax. It is defined as an extension of RDFS, so it is compatible with it. It is based both on descriptive logic (declaration of axioms) and on frame-based systems (taxonomies of classes and attributes). It includes a meticulous and precise semantics to describe the meaning of terms (so it also allows to describe implicit information).

OIL is structured in several layers of sub-languages, among them the base layer, RDFS. Each additional layer adds functionality and complexity to the previous layer. In this way, agents (humans or machines) that only have the capacity to process expressions of a lower level layer, can partially understand ontologies expressed in any of the upper layers. This model of layers is observed in the following figure.

Figure 46 Language hierarchy [Fensel01]

The main drawback of OIL is the lack of expressivity to declare axioms, which is why DAML + OIL emerged as a result of the cooperation between OIL and DARPA, responsible for the definition of the DAML language (DARPA's Agent Markup Language) that was developed in mid of the year 2000 in order to extend the level of expression of RDF-Schema.

DAML + OIL is a language based on W3C standards, which unifies both languages by combining the most important features of each one of them. DAML + OIL inherits many of the characteristics of OIL but moves away from the model based on frames, focusing and enhancing the descriptive logic, DL (Description Logic). It is a more powerful language than RDF-Schema to express ontologies. In the latest revision of the language, this already has a valuable set of elements that allow to create ontologies and mark the information so that it is readable and understandable by the machines. It also functions as an information exchange format and has a greater capacity for expression and inference than its predecessors.

But DAML + OIL presents a series of shortcomings due to its great conceptual and usage complexity. In order to solve this complexity, the OWL language was developed.

4.2.4 OWL

Ontology Web Language [W3C-OWL] is the language of the Semantic Web most used today together with OWL2 to represent ontologies explicitly, that is to say, it allows defining the meaning of terms in vocabularies and the relationships between those terms. OWL has replaced DAML + OIL as the W3C standard for the representation of ontologies. Both languages are very similar both syntactically and semantically, but OWL is much more complete.

OWL is an object-oriented ontology language, formal semantics and abstract syntax. The abstract syntax defines what can and cannot have each element of information contained in the ontology, but in a way not suitable for human reading. For this reason, two syntaxes are proposed for the exchange of information that allow ontologies to be interpreted and exchanged between different tools and applications. Both require the support of the data types defined in the XML Schema (e.g. RDF/XML).

The syntax of OWL defines classes, properties of classes and instances or individuals of said classes, which constitute the information elements of the language. Each individual has certain values of the properties of the class to which they belong to. In addition, OWL allows defining axioms that are applied to these elements to create relationships between them, as well as modifying or restricting their extensive meaning (although the axioms cannot modify the intensive meaning, that is, the concept they represent).

Like OIL, OWL is structured in layers that differ in complexity and can be adapted to the needs of each user, to the level of expressivity required and to the different types of existing applications (search engines, agents, etc.). In relation to this, three versions of OWL have been developed:

- **OWL Lite:** smaller version, not compatible with all RDF / RDFS documents. It is defined as a subset of the total existing constructions for OWL, and also establishes restrictions on its use. It is designed for beginners or those who look for simplicity. Formally, its semantics can be considered as an extension of a subset of RDFS.
- **OWL DL (OWL Description Logic):** intended for those users who want maximum expressiveness but guaranteeing computational completeness (possibility of reaching conclusions based on existing information) and inference in finite time. It includes all constructions defined for OWL, but must be used with certain restrictions to achieve the mentioned properties. This makes it incompatible with documents that use the maximum expressiveness of RDF / RDFS, since its semantics is also an extension of a subset of the latter.
- **OWL Full:** wider version, intended for those users who want maximum expressiveness and the freedom of syntax of RDF. It is compatible with RDF / RDFS, being able to conceive its semantics as an extension of the latter, that is, it follows the same construction procedure on RDF and RDFS as RDFS in relation to RDF. Therefore, models based on OWL Full can freely use own constructions of RDF, RDFS and OWL. In exchange for this, there are no guarantees about completeness and finite time in the reasoning. This is due to the fact that the constructions that define the metamodel can be used in the models, which means a lot of flexibility but also a lot of freedom.

Regarding the expressiveness of the language, both the RDF / XML syntax and the OWL / XML syntax are based on the theory of the model [RDFMT01]. This theory is a technique to formally specify the semantics of a formal language. It assumes that language refers to a world, and describes the minimum conditions that a world must satisfy to assign an appropriate meaning to each expression of the language. Each concrete world to which this language can be referred to, is called interpretation. An interpretation is the conception of a world with different elements of different type and different meaning (vocabulary), among which the datatypes (types of data used) are included, and this interpretation may

or may not satisfy a certain ontology. The fundamental idea of the theory for the design of applications based on OWL ontologies, is that its approach is totally descriptive: the sentences of the ontologies do not determine the existence or non-existence of the elements, simply they describe them. That is, it must be assumed that there is a world (interpretation) and our ontology simply talks about it. It is an open-world approach, in which any document can import another ontology and incorporate new sentences that complete the description of that world (consistent with the principle of reuse and distribution of resources on the Web).

In short, an ontology does not determine how the world is, but that world or interpretation may or may not satisfy the ontology, that is, it may be consistent with it or not. Given an OWL ontology, there can be many interpretations (different possible supposed worlds) that satisfy it, for which they must comply with a series of restrictions.

4.2.5 OWL2

OWL2 [W3C-OWL2], is an extension and revision of the OWL developed by the Web Ontology Working Group of W3C, whose first version was published in 2004. Like OWL, its objective is to facilitate the development of ontologies and exchange of information through the Web, to make web content more accessible and interpretable by machines. The following figure shows the main components of the structure of OWL2 and the relationship between them.

Figure 47 OWL2 Structure

The ellipse of the center represents the ontology itself, be it an abstract structure or an RDF graph. In the upper part, the different syntax formats for the exchange of information with the ontology are observed. In the lower part, we observe the two models of formal semantics supported, which define the meaning of ontologies in OWL2.

OWL2 extends OWL in both semantics and syntax, and its degree of expression is much greater. Regarding the syntax, OWL2 mainly uses RDF / XML as information exchange syntax, just like OWL, and this must be supported by all tools that handle OWL2 ontologies. In addition to RDF / XML, it supports other formats such as OWL2 Manchester, as seen in the following table:

Table 4 OWL2 syntax formats supported

Name of Syntax	Specification	Status	Purpose
RDF/XML	Mapping to RDF Graphs, RDF/XML	Mandatory	Interchange (can be written and read by all conformant OWL 2 software)
OWL/XML	XML Serialization	Optional	Easier to process using XML tools
Functional Syntax	Structural Specification	Optional	Easier to see the formal structure of ontologies
Manchester Syntax	Manchester Syntax	Optional	Easier to read/write DL Ontologies
Turtle	Mapping to RDF Graphs, Turtle	Optional, Not from OWL-WG	Easier to read/write RDF triples

In terms of semantics, two models are defined: direct semantics and semantics based on RDF. Both provide two alternative ways of giving meaning to ontologies and are used by reasoners and other tools, among other things, check the consistency of classes, subsumption or to make inquiries about individuals. There is a theorem that allows to relate both semantics establishing relationships and equivalences between them.

OWL2 has three different profiles, also called fragments or sublanguages. Each profile is a reduced version of OWL2 whose objective is to improve the reasoning efficiency in different application scenarios. The profiles are independent of each other, and the choice of one or the other, or of OWL2 with direct semantics or OWL2 with semantics based on RDF, depends on the structure of the ontology and the reasoning tasks that are to be achieved. The profiles are:

- **OWL2 EL (Existential Language):** profile widely used in ontologies with a large number of properties and classes. This profile is a subset of OWL2 that allows basic reasoning tasks using polynomial time algorithms without losing expressiveness. The reasoning time is proportional in a polynomial way to the size of the ontology. The scalability of these dedicated algorithms has been proven with ontologies of many kinds and properties.
- **OWL2 QL (Query Language):** profile aimed at ontologies that use large amounts of instances or individuals, where responding to queries about data from those instances in an efficient way is the most important reasoning tasks. In addition, polynomial time algorithms can be used to check the consistency of the ontology and the subsumption of classes. Queries can be formulated using the standard Query Language. On the other hand, the expressive force of this profile is more limited than that of the previous profile.
- **OWL2 RL (Rule Language):** profile aimed at applications that require scalable reasoning but without sacrificing too much expressive power. This profile is oriented to applications that need a balance between full expressivity and efficiency, as well as applications that need more expressiveness. The reasoning can be implemented by reasoning engines based on rules. The consistency of the ontology, the subsumption and satisfaction of the classes, the verification of instances, and the

resolution of solving joint queries is solved using polynomial time-based reasoning algorithms with respect to the size of the ontology.

As previously indicated, OWL2 extends to OWL, so it has a very similar structure (the use of RDF / XML as the main syntax, the semantics used, the theorem that relates both semantics, etc.). In addition, there is full compatibility between OWL and OWL2, so all OWL ontologies are still valid in OWL2. However, there are certain differences between both languages:

- OWL2 adds new functionalities that allow to increase the degree of expressiveness, as for example, it includes keys, support of a greater number of data types, richer data ranges, cardinality restrictions, properties of asymmetry, reflexivity and disjunction, and improves the notation capacity.
- OWL2 defines new exchange syntax and there is a greater number of reasoners that support it.
- Both languages are based on descriptive logic (DL), which allows describing elements and their relationships, and whose objective is to give the language the greatest possible expressiveness without losing decision. But while OWL is based on SHOIN (D), OWL2 is based on DL SROIQ (D), with greater expressive power.

4.2.6 Discussion

The ontology definition languages allow the uniform representation of conceptual domains. With these languages one obtains the expressivity and facility for the representation of the concepts that taxonomies lack. In addition, they favor the reusability of ontologies, since any previously created ontology can be referenced and reused from another ontology, thus reducing the time for the development of new ontologies. In this section, the main languages of existing ontologies have been analyzed, paying special attention to those belonging to the Semantic Web.

The choice of one or another language for the formalization of the resources of the present research work is a complex task that must be carried out in a precise way, since the effectiveness of the system will depend to a certain extent on the chosen language. In the choice of a language should be taken into account some features of it: level of expression that is achieved with the language, mechanisms of inferences associated with it, tools that support it and exchange between applications. From the analysis carried out, the following conclusions have been drawn, bearing in mind that OWL includes, in principle, both OWL and OWL2:

- Due to the great impetus and importance of the Semantic Web, it is considered convenient to choose a language of the so-called "Semantic Web Languages", so that SHOE would be, at first, discarded, since it is a language prior to it.
- OWL and RDFS are the most widespread languages to describe ontologies.
- OIL and DAML + OIL have evolved to give rise to OWL, so we could consider them as disused languages to replace OWL. Therefore, they would be discarded.
- RDF is the language on which the rest are built (RDFS, OIL, DAML + OIL, OWL). In addition, its semantics have little structure forcing a large number of assertions to capture information similar to that already captured by RDFS or OWL.
- OWL has more expressive power than RDF.
- The syntax of OWL is RDF / XML, which allows the automatic incorporation of vocabularies for the description of data like RDFS or Dublin Core.

- OWL is designed to be used when the information contained in the documents needs to be processed by programs or applications, and not just be presented to human beings.
- OWL has more functionalities to express the meaning and semantics than XML, RDF and RDFS, in addition to allowing to represent Web content interpretable by machine, unlike the others.
- OWL has reasonably contrasted reasoning languages such as KAON2 or SWRL.
- Most tools and applications of the Semantic Web support OWL.

From the conclusions drawn, it is considered that OWL and OWL2 are the best option as resource specification languages for use in this research work. Although the structure of both languages is very similar, there are a number of advantages of OWL2 over OWL, as previously indicated: OWL2 allows to achieve a higher level of expressivity than OWL (provides greater support of data types, has a greater number of axioms, includes cardinality restrictions and properties of asymmetry, reflexivity and disjunction, etc.) and the number of semantic reasoners that support it is greater. Within the versions of OWL2, the choice will depend on the type of reasoning required and the level of expressiveness.

However, the ontologies defined in OWL are fully compatible with OWL2, so the use of OWL is also valid to specify the proposed ontology.

4.3. BEHAVIORAL LANGUAGES SPECIFICATION

In the previous section devoted to ontology specification languages, a brief study was carried out on the most relevant ontology languages and their main characteristics. The use of these languages as formal languages for the representation of knowledge and specification of information in ICT systems, brings advantages such as orientation to objects, the reuse of information, the availability of tools, easy extensibility, etc. However, in addition to the ability to represent knowledge, we need the ability to represent reactions to events, so we need languages to specify the behavior in conjunction with the concepts represented in the ontologies.

Therefore, the possibility to go beyond the representation of knowledge (security restrictions, rules and policies that allow the definition of the behavior of the entities involved) has been valued. In this way, the behavior could also be expressed in a formal way.

The way to express restrictions is through axioms. The axioms can be of several kinds, for example, subclass axioms, class equivalence axioms, axioms of equality or inequality of instances, intersection or union of classes, restrictions on properties, etc. In the Semantic Web, the OWL2 language incorporates many of these types of axioms. For this reason, the integration of logical restriction specification languages is proposed to meet the following requirements:

- Create a level of rules above the ontologies, semantically integrated in a coherent way. This level allows you to create rules and make reasoning.
- Improve the ability of ontologies to perform queries on individuals (request individuals or specimens that meet certain conditions).
- Data integration. The majority of the data reside in relational databases. The ontologies do not, since they need to replicate the information in the language of ontologies. Therefore, an implementation of an ontology language based on a logical program can allow a closer interaction with the elements of the databases.

- Creation of Semantic Web Services: while ontologies are useful to represent hierarchical classifications of services, their inputs and outputs; the rules are useful to represent policies or relationships between preconditions and post-conditions.

OWL2 allows expressing certain restrictions including relationships using property chains, functionality that is not supported in OWL, but still cannot express relationships between individuals referenced by properties, among other restrictions. Therefore, the use of rules or policies languages based on first order logic (FOL, First Order Logic) is required, which allows to express more logical constraints than the ontology definition languages. These languages, in addition to specifying behavior, allow the definition of security policies or rules.

Some of these languages of specification of logical restrictions of the Semantic Web are SWRL, KaOS, REI and OWL-Services. These languages have in common that they are based on OWL as an ontology specification language, so they have inherited a series of OWL characteristics, such as the orientation of objects or the possibility of distributing ontologies.

In this section we try to perform an analysis of different languages defining rules, restrictions and behavior, highlighting its main characteristics and objectives.

4.3.1 RULEML

RuleML is a standard tag language for the definition, publication and exchange of rules in the Semantic Web. It is based on XML / RDF, and its core is the Datalog sublanguage (intersection of SQL and Prolog). RuleML uses the XML syntax for defining rules.

RuleML allows the representation of knowledge, combining facts and rules. The rules that can be defined with RuleML are of several types:

- Production rules: implications (if - then).
- Reactive rules: rules that perform actions when an event is fulfilled or when some certain circumstances (event-condition -action).
- Integrity rules: statements that must be fulfilled in any state of the system.
- Derivation rules: rules to define derived concepts from others (implication-inference).
- Transformation rules: rules created from a caller, a condition and a transformation.

The main advantages of RuleML are the following:

- It allows to define different types of rules.
- It is supported by a large part of the inference engines and applications that handle rules, independently of the inference mechanism used by them.
- It allows the exchange of rules between applications.

But it also has a great disadvantage: it is not intended for use combined with OWL or OWL2, also it does not support bindings of variables.

4.3.2 SWRL

Semantic Web Rule Language, SWRL [W3C-SWRL], is a Semantic Web rule specification language, based on OWL and RuleML, whose objective is to increase the expressiveness of OWL and OWL2 axioms regarding the specification of restrictions and rules. SWRL emerges as an integration of OWL DL and OWL Lite with the Unary / Binary Datalog sub-languages of RuleML.

SWRL extends the set of OWL axioms, thus allowing to define rules of greater expressivity. It allows you to include axioms of rules, also called Horn clauses. It provides a theoretical-semantic model to include rules in OWL / OWL2 ontologies. In addition, as information exchange syntax, an extension of the OWL syntax based on XML can be used directly, or an extension of the OWL-RDF / XML syntax.

4.3.2.1. Language Elements

The formal semantics of SWRL is an extension of the OWL DL independent of RDF / RDFS. Basically, SWRL introduces a new type of axioms called as rules. Each rule is formed by an antecedent (body) and a consequent (head), and each of them is formed by the conjunction of different atoms. The main elements are defined below.

▪ **RULE AXIOM:**

A rule axiom consists of an antecedent (body) and a consequent (head), each of which is composed of a set (can be empty) of atoms. In the SWRL classical logic notation syntax, a rule has the following form:

Equation 1 SWRL syntax: body and head

$$\text{Antecedent (body)} \rightarrow \text{Consequent (head)}$$

Informally, a rule can be interpreted as follows: provided that the conditions specified in the antecedent are verified, then the conditions specified in the consequent must also be verified.

Suppose we want to define a rule that states that the hasParent and hasBrother properties imply hasUncle, that is, we want to define the hasUncle property. The rule would be specified in the following way, if a syntax that is understandable by humans is used:

Equation 2 SWRL example rule to guess and uncle if knowing brothers of your parents

$$\text{hasParent}(?x1, ?x2) \wedge \text{hasBrother}(?x2, ?x3) \Rightarrow \text{hasUncle}(?x1, ?x3)$$

In the following rule, the antecedent states that there are two instances (? X1 and ? X2) that are related through the hasParent property (? X1 has a parent that is ? X2), and also the instance ?X2 has a sibling ?X3, related through the hasBrother property. The rule specifies that as long as the antecedent is met, the consequent is fulfilled, which implies that ?X3 is uncle of ?X1 (hasUncle property). Therefore, if a copy "Father" has a copy "Son" and another copy "Brother", it implies that the copy "Brother" is an uncle of the copy "Son" (they are related through the hasUncle property).

A SWRL rule therefore has the form of an implication relationship between the head and the body. In the consequent only variables that have appeared in the antecedent can appear. This condition, called safety condition, is very important, since it requires that only conclusions about information already present can be drawn.

The SWRL specification offers an abstract syntax that extends the OWL abstract syntax to include this new relationship in the ontology language. The XML marks that allow describing these rules include:

- <ruleml: imp>: Is the element that allows to relate the body of the rule with the head.
- <ruleml: _body>: Is the element that lists the atoms of the body of the rule.
- <ruleml: _head>: Is the element that lists the atoms of the head of the rule.
- <ruleml: var>: Allows to define the variables on which to evaluate the rules.

- `<swrlx: individualPropertyAtom>`: It allows to define atoms referred to concrete properties. It is also possible to define atoms referring to classes, ranges of data, valued properties, or functions of a mathematical type, strings of characters or dates.

As you can see, many of these marks are not defined within the SWRL namespace, but rather from RuleML, a previously defined rule language that has been taken as the basis of the SWRL definition.

SWRL contributes above all to the definition of atoms and their integration within an ontology written in OWL / OWL2. These can contain a conceptual representation of the knowledge required in OWL, and associated rules of behavior in SWRL, which are interpreted and validated by an expert system at runtime.

▪ **ATOMS:**

Atoms are predicated with the form `C (x)`, `P (x, y)`, `sameAs (x, y)`, `differentFrom (x, y)`, or `builtIn (r, x, ...)` where `C` is a class or data type in OWL / OWL2, `P` is a property in OWL / OWL2, `r` is a built-in relation (pre-programmed), and `x` and `y` can be both variables, such as OWL / OWL2 specimens, and data type values. Therefore variables can be used to build atoms. These are equivalent to the universal quantifier, and can be replaced by copies.

In SWRL there are different types of atoms, which can express among other things:

- belonging to a specimen (a variable can be used in its place) to a class extension,
- belonging to a literal to a datatype listed in OWL DL,
- relation between two specimens of type object through a property of type `ObjectProperty`,
- relation between an instance of type object (in the position of subject) and a literal (in the object position) through a property of type `DatatypeProperty`, or
- equality and inequality between two copies.

▪ **BUILT-INS:**

A very important element of the SWRL language are the so-called built-ins, which are SWRL elements that allow different operations to be carried out on different types of XML Schema data:

- **Comparison**: equality, inequality, greater than, less than ... The atom is satisfied if the comparison also does.
- **Mathematical operations**: addition, subtraction, product, division, sine, cosine, etc. The atom is satisfied if the elements of the operation (all, including the result) satisfy it.
- **Logical negation**: it is satisfied if one argument is equal to the logical negation of another.
- **Operations with character strings**: concatenation, chain length, etc.
- **Operations with dates, time and temporary durations**.
- **Operations with URIs**.
- **Operations with lists**.

The elements of the atoms on which the operations are applied must be variable, copies or literals.

4.3.2.2. Formal Interpretation of Its Semantics

The formal interpretation of the SWRL semantics is as follows:

- An interpretation satisfies an OWL / OWL2 + SWRL ontology if it satisfies the OWL / OWL2 ontology according to its formal semantics and also satisfies the SWRL rules.
- An interpretation satisfies a SWRL rule if for every link that satisfies the antecedent the consequent is also satisfied. Given an OWL / OWL2 interpretation, a link is an extension of it that is constructed by replacing a variable with an instance.
- The definitions of consistency and inference are coherent with those of OWL / OWL2.
- If the antecedent is empty, it is interpreted as trivially true: the consequent must always be fulfilled (unconditional facts). Many of these facts are best expressed by facts in OWL / OWL2, since they are explicit.
- If the consequent is empty, it is interpreted as trivially false, that is, that the antecedent must never be fulfilled.

Given an ontology consistent with a series of axioms. If we add SWRL rules about this ontology, new information is being added implicitly, in such a way that:

- If this implicit information conflicts with the information of the interpretation of interest, the ontology resulting from the combination of OWL with SWRL is no longer consistent.
- If this new information does not conflict with the pre-existing one, the new ontology satisfies the interpretation of interest and is consistent. In addition, two things can happen:
 - That this new information is not really new, coinciding with other information defined by OWL explicitly. In this case, SWRL serves to verify that certain axioms or restrictions are met.
 - That this new information does not coincide with explicit OWL information, but that it is telling us additional things. This new information is not represented explicitly in the ontology, it is not expressed in axioms of ontologies as such, and it is expressed implicitly: the interpretation must satisfy these implicit restrictions, even though the ontology does not reflect them explicitly.

As a conclusion, we can say that OWL / OWL2 allow to express information explicitly, while SWRL does it implicitly, and the only way to make it explicit is through consultations and inferences.

4.3.2.3. Expressivity and other Characteristics

Some of the rules that can be built with SWRL can also be created with OWL, but the scope of expressivity of SWRL is much greater, since it allows defining constraints that are not possible with OWL or RDF. Some of the most important characteristics of SWRL are:

- SWRL allows defining complex conditions to be fulfilled in the antecedent by using built-ins, the AND operator, and the atoms referred to concrete properties, to classes and relationships among them, to ranges of data, to properties, to mathematical functions, strings of characters or dates. The use of variables in atoms allows defining constraints that are not possible in RDF or OWL.

- First order logical formulas: in first-order logic, an implication clause can include the use of basic operators both in the antecedent and in the consequent: OR, AND, NOT, and other derivatives such as XOR, etc. This allows the definition of arbitrary formulas with any composition of expressions (e.g.: A OR (B AND NOT C) AND (D OR C)). **However, SWRL does not include NOT or OR operators for the definition of atoms, which limits their expressivity.** Within the W3C there is a proposal to extend SWRL to be able to define this type of first order logic formulas called SWRL-FOL, but at the moment it is not available.

In any case, through certain transformations, it is possible to achieve a degree of expressiveness almost as high as that of the first order logical rules. In the case of the condition of the SWRL rule, there is the possibility of expressing a rule with a logical formula of first order in the antecedent by means of several SWRL rules following certain guidelines.

The main advantages of the SWRL language are:

- High abstraction level.
- Rules language recommended by the Semantic Web community.
- Compatibility following the OWL XML and RDF syntax.
- Increase the OWL expression capacity to define rules and restrictions. Conditional rules (Horn clauses). SWRL allows defining complex conditions to be fulfilled in the antecedent of the rules, through the built-ins, the AND operator and the use of atoms. The use of variables in atoms allows defining constraints that are not possible in RDF or in OWL.
- Easily integrated into ontology editors such as Protégé. There are tools, such as a Protégé plugin that allow you to introduce the rules systems and reflect them in information that can be recovered, used and stored, as well as defining the ontology itself.
- Allows you to define more complex and expressive rules than other languages.
- High integration level with ontologies defined in OWL. It does not require translation between the elements represented in the ontology and the concepts handled by the language. You can define rules directly on elements of the information model represented by OWL ontologies.
- It is not limited to deontic logic (policies related to duty and rules), it allows the definition of all kinds of rules.

However, it also has certain drawbacks:

- Limited to OWL classes and binary predicates.
- It does not include NOT or OR operators for the definition of atoms, which limits the expressiveness of language. Yes, it allows AND. However, through some transformations, it is possible to achieve a degree of expressiveness almost as high as that of the first-order logical rules, dividing the rule with a logical formula of first order in the antecedent by means of several SWRL rules.
- It does not allow to specify meta-information about the rules (priorities, scope of application, etc.), useful to manage the rules remotely and for policy analysis and conflict resolution.
- It does not allow to represent events explicitly.

Due to its limitations in terms of expressiveness to specify security policies, they arise other approaches based on ontologies to define security policies.

4.3.3 OWL-Services

OWL-Services, is an ontology of services whose objective is to formally describe web services. It is intended that users and software agents are able to discover, invoke, compose and monitor web resources that offer particular services with a series of determined properties. In addition, it allows them to do so with a high degree of automation. Thanks to this formalization it is possible to achieve this desired automation in the different interaction processes between user agents and service providers. OWL-S provides a vocabulary in OWL suitable for these descriptions, and this vocabulary can be used in conjunction with the OWL mechanisms. OWL-Services is a web services specification language, with formal syntax and semantics built on the formal semantics of OWL.

The objectives that are intended to be achieved with the development and use of OWL-Services are the following:

- Automatic discovery of web services: automatic location of services that, thanks to their capabilities, allow satisfying a series of needs. For this, it is necessary to formally describe the services offered and their capabilities, so that agents can interpret it and assess whether or not they respond to the needs of the user. OWL-Services allows such a description.
- Automatic invocation of web services: the automatic invocation requires a formal description of how it should be carried out, which could be through remote procedure calls. OWL-Services allows you to describe the invoke primitives, in addition to allowing the user agent to automatically read the description of the inputs and outputs of the web services and invoke the service.
- Composition and automatic joint operation of web services: In a website where several services are available, it would be interesting to perform complex tasks that involve the coordinated invocation of several web services. This requires describing the prerequisites and consequences of the application of services. OWL-Services provides vocabulary suitable for this.

4.3.3.1. Language Elements

The OWL-Services ontology defines three top-level classes: ServiceProfile (describes the profile of the service, what the service provides to clients), ServiceModel (describes the model of the process, how the service is used, what a client should do to request and use the service) and ServiceGrounding (describes how it interacts with the service, describing for example, the format of the messages, details of communication and transport protocols, number of ports, service primitives ...). Each of the classes is broken down into a series of components.

The service profile is described in terms of input and output parameters, preconditions, results and service category. The concrete processes of each service do so through input and output parameters, preconditions, results and participants. To represent parameters, you can use variables, using an extension of the SWRL variables, and to describe preconditions and results, you use logical formulas or expressions that use the input and output variables as parameters. These expressions can be represented in an integrated manner in the ontology by means of some standard language or format such as KIF, but it is also possible to use simply RDF literals, so that the content thereof is transparent for the OWL interpreters. There are numerous proposals to represent formulas, but it should be noted that OWL really does not adapt too well to this purpose, since its objective is to describe in a more static way concepts and relationships, not formulas that detail processes.

The processes, once defined, are composed to form compound services through different mechanisms: sequence, concurrency, concurrency with synchronized sweep, choice alternatives, if-then-else structures, and loops.

4.3.3.2. Expressivity and other Characteristics

In the world of the Semantic Web, the aim is to achieve broader access not only to content, but also to web services. The objective of OWL-S is to automate this access to web services or resources, by formally defining both the services and the processes that compose them. OWL-S constitutes a metamodel or web services specification language, with formal syntax and semantics built on the formal semantics of OWL.

OWL-Services allows describing atomic or simple services and complex services. The descriptive capacity of OWL-Services allows to describe processes and flows as complex as those described by procedural languages. However, these are descriptions of services, not executable code.

By integrating SWRL conditions as preconditions and postconditions, OWL-S allows a powerful semantic description of policies.

4.3.4 Other Behavioral Languages' Specifications

4.3.4.1. SWSL

SWSL (Semantic Web Services Language): logic-based language whose objective is to specify formal characterizations of concepts and descriptions of web services. It is not a simple rule definition language. It is formed by two sublanguages:

- SWSL-FOL: logic language of the first order, used to define the formal specification of the service's ontology. It is structured in several independent levels, each of which provides new features that increase the power of the language. The levels are not organized on the basis of expressive power and computational complexity, as in OWL.
- SWSL-Rule: language based on rules, which can be used as a specification language and as an implementation language. This language is divided into several independent levels organized as SWSL-FOL, and is based on Horn clauses, in the form "head: - body". The layers or levels in which the sublanguage is divided are: Mon LT (allows disjunction in the body and conjunction and implication in the head of the rules), NAF (allows denial in the body), Nommon LT (allows quantification and involvement in the body), Corteous (introduces classical negation and prioritization of rules), HiLog (adds a certain degree of metaprogramming, allowing variables that extend from symbols of predicates, formulas or functions), Frame (introduces object orientation) and Equality (introduces the predicate of equality).

SWSL-FOL and SWSL-Rule cannot be used together, so SWSL serves as a bridge between them. SWSL-Rule contains syntactically all the connectivity with the first-order logic that SWSL-FOL provides, although semantically they are incompatible.

This language has the advantage of allowing the creation and processing of Web services using semantics based on ontologies and is similar to other systems of rules and inferences such as Prolog, in terms of defining rules. On the contrary, it is a language.

In the development phase (extension is expected through the incorporation of if-then-else statements, restrictions, etc.) and very complex for new users.

4.3.4.2. WSML

WSML (Web Service Modeling Language): rules representation language with "head: - body" form, focused on semantic web services, and compatible with ontologies written in OWL. It is formed by several sublanguages that differ mainly in the logical formalism on which they are based: descriptive logic, first order logic or programming languages based on logic. These sublanguages are:

- WSMLCore: corresponds to the intersection between descriptive logic and Horn clauses, without function symbols and equality predicate, covering only a part of OWL.
- WSMLDL: extends the previous one, expanding the compatibility with OWL.
- WSMLFlight: extends WSMLCore, adding functionalities for logical programming, such as restrictions, etc. It also incorporates a rules language (including predicates of equality and negation), allowing to carry out reasoning on the defined rules.
- WSMLRule: extends the previous one, improving the ability to define rules.
- WSMLFull: groups all the previous variants.

WSML is compatible with the OWL language, allows the exchange of rules with implementations of existing rules, its XML syntax is based on RuleML and solves certain limitations of SWRL. The big drawback is that it is limited to the field of Web services.

4.3.4.3. Jess

It is a rule engine and script language written entirely in Java. It is a small, light rule engine and one of the fastest. Initially, Jess was created as a Java copy of CLIPS, but currently has many features that differentiate them.

Jess can be used as a rule description language, which allows both the definition of the rules and the inference from them, but also supports their use as a general-purpose programming language. Among its main features as a rule language, it is a powerful scripting language, which gives the user access to all Java APIs, and which is characterized by a language whose operating philosophy is to act in response to inputs.

It presents a series of advantages: it is multiplatform, with the ability to function as a rule engine in combination with the Protégé tool through the Jess Tab plugin, it is widely used in the development of expert systems and is easy to integrate into any application. However, it has two major drawbacks: it is a slow language, and not adapted to languages such as OWL, SWRL, etc.

4.3.4.4. R2ML

R2ML: rules language based on XML with support for the definition of integrity, reactive, production and derivation rules, whose objective is to facilitate the exchange of rules between different languages and tools. In addition to the exchange of rules between different languages and tools, it provides the ability to publish rules, and obtain results through inference to any system.

Basically, R2ML involves the integration of OCL, OWL, SWRL and RuleML. Currently, R2ML rules can be translated automatically into a set of languages: F-Logic, Jess, RuleML, Jena, Drools, SWRL, XMI, OCL.

4.3.4.5. KAoS

Knowledge-able Agent-oriented System, KAoS, is a proposal of the IHMC (Institute for Human and Machine Cognition) of the University of West Florida, which emerges as a framework for modeling policy-based services. The main objective and purpose of KAoS is to control in a flexible way the behavior of the different agents that intervene in distributed

and open environments, and in multi-agent platforms, such as the use of web services of the Semantic Web. In addition, KAoS aims to respond to the challenges posed by the requirements emerging from semantic applications for infrastructures, especially in the area of security and trust management.

KAoS proposes a complete environment to define and implement policies that specify how the different agents participating in the indicated applications and in other types of distributed systems can and should act.

The philosophy and general concepts that characterize KAoS fit into the policy-based management paradigm.

4.3.4.6. REI

The REI policy definition language, emerges at the University of Maryland with a purpose similar to KAoS, to flexibly control the behavior of the different agents involved in distributed and open environments, especially the security.

The REI policy specification framework provides:

- A model of policy specification independent of the domain, which can be specified in Prolog predicates or in an ontology. This model also allows expressing policies, also expressing metainformation about them (how they are remotely managed) and useful information for policy analysis and conflict resolution, through use case management and what-if analysis.
- An engine of inferences that allows to reason with the policies, determining their applicability and making analysis of conflicts and queries.

4.3.5 Discussion

In certain systems based on ontologies, such as actionable cyber threat intelligence or a dynamic risk system, the ability to represent behaviors, patterns and reactions to events is very important.

Through the use of these languages there is a solution, among other things, to the problem of the expressivity of ontology languages such as OWL, since they allow the definition of axioms that provide more information about the modeled domain and facilitate the work of inference.

In this section, the most relevant rules and policies languages have been defined and analyzed briefly, highlighting their main characteristics. A behavior definition language must meet the following requirements:

- It must be compatible with the languages and formats used in the application environment. In the case in question, in the previous section the choice of OWL2 was indicated as the ontology definition language used in this doctoral thesis, so the rules language used must be fully compatible with ontologies written in OWL2.
- It must be expressive.
- It must be flexible, that is, it must allow to introduce new concepts and define new rules of behavior.
- Must allow the formal specification (understandable by machines) of the rules despite the complexity or high level of abstraction that may involve.
- It must allow its implementation, that is, it must allow the application of the rules to be reflected in the behavior of the DRM Framework system.

- Must have well-defined interfaces, unambiguous that are open to implementation, and scalable.
- From the analysis carried out, regarding the level of satisfaction of the previous requirements, the following conclusions can be drawn:
- RuleML is not a language designed to be combined with OWL, so it is discarded as a rule definition language. In addition, it is the base of SWRL, so the latter extends its main features and functionalities.
- The SWSL, WSML and OWL-Services languages are languages oriented to semantic web services, so they are not completely adapted to the needs that the definition of defined DRM system rules requires.
- Jess is very powerful as a rule engine. As a rule description language it is not adapted to OWL or OWL2, so this language would be discarded. However, there is a plug-in for Protégé 3.3.1 that allows to integrate the Jess rules engine with an ontology defined in OWL DL.
- R2ML is a language that allows you to exchange rules between different languages and tools. It could be considered as a "translator" of rules defined in different languages. However, it cannot be considered as a rule language as such, since none of the rule engines or semantic reasoners use this language.

SWRL, KAoS and REI, can satisfy the most of the requirements but they are languages with relevant differences, especially with regard to the definition of the elements of the language and the level of expressiveness. These differences make it possible to compare them based on the following characteristics:

- Level of interoperability and level of integration with ontologies, understanding as integration the capacity of the language to define rules or policies directly on the ontology domains.
- Level of expressiveness: it is desirable that they have sufficient expressiveness to express restrictions and logical relations on the data and objects of the information models.
- Usability level, considering this as the existence of tools for the specification of policies or rules, the existence of reasoning engines compatible with languages, etc.
- Easy to use.

In terms of level of integration with the OWL ontologies, SWRL emerges with the aim of extending the semantics of OWL, but it is not conceived as a simple OWL patch, but in its definition one of the main requirements has been the complete integration with OWL. In addition, both allow to handle a semantically integrated information model, so SWRL rules can be defined directly on elements of the information model represented in OWL / OWL2 ontologies.

On the other hand, KAoS and REI can also work importing ontologies of any type of domain in OWL, but for that, they need to translate them to the concepts handled by these languages, such as actors, entities, etc., which implies a greater degree of indirectness in the integration, and presents greater difficulties for the representation of low level policies in very specific domains.

Therefore, in terms of integration level, the SWRL language has a direct integration with OWL / OWL2, unlike REI and KAoS that require a prior mapping to the concepts handled by the language (entities, actors, roles, etc.). In addition, SWRL is not limited to policies with deontic logic such as REI and KAoS (policies related to duty and norms: authorizations, denials, delegations, etc.), and allows expressing other types of conditions and relationships on the elements represented in the ontology.

Regarding the level of expressiveness, the languages must have sufficient expressiveness to express restrictions of the value of the attributes (belonging to a range or a discrete set of values), relations between the values of the different qualitative attributes mainly and quantitative also, and preconditions and postconditions of the methods in case they exist.

In the case of the SWRL language, this chapter indicates that SWRL exceeds the limitations of OWL as far as the definition of restrictions is concerned. SWRL allows to express restrictions in the form of rule axioms thanks to the use of variables, restrictions that are not possible by OWL or OWL2 characteristics.

KAoS is a language based on OWL that exploits its characteristics. From the point of view of expressiveness this means that it has limitations when defining certain types of policies, especially those that require the definition of variables, or those that contain parametric restrictions. On the other hand, KAoS is a language rich in auxiliary elements (priorities, condition and action status, time stamps, etc.), and distinguishes between constraints of obligation and conditions.

For its part, REI is a language that, like SWRL, allows to specify restrictions that require the definition of variables, a feature that is not possible in OWL. In addition REI also allows to specify auxiliary elements in the definition of policies.

In terms of usability level, SWRL has a wider scope in the Semantic Web environment than REI and KAoS, which makes it supported by many environments and tools. Despite not offering a policy forcing model or a policy engine of its own, the SWRL language is supported by most of the existing semantic reasoners, as will be seen in the following section.

The REI environment includes a policy engine that reasons about policy definitions, which accepts definitions in both the REI and RDF-S language, consistent with the REI ontology, but like SWRL, does not offer a policy enforcement model. The inference engine is limited to reasoning about policies and answering queries.

KAoS has a more complete policy environment, which includes a graphical interface for editing policies (KPAT), a policy repository with conflict resolution and consultation, and a policy enforcement mechanism.

In terms of ease of use, SWRL is a rule definition language and not one of policies such as REI or KAoS. The rules languages are much simpler than those used to represent policies, because in the latter, much more elements are necessary.

In addition, as indicated and will be seen in the later section, the SWRL language is supported by most of the existing semantic reasoners.

4.4. SEMANTIC REASONERS

A semantic reasoner also known as a reasoning engine, rule engine or simply reasoner, is a software component capable of inferring logical consequences from a set of facts or axioms in a knowledge base. The concept of semantic reasoner encompasses the inference engine, since it provides a set of mechanisms with which to work. The inference rules are specified by policy definition languages, as seen in the previous section.

In the Semantic Web, there are different reasoners capable of inferring additional knowledge. The main difference between them is the language of the rules they support and the reasoning mechanism used, in addition to other functional characteristics that determine the performance and efficiency of each reasoner. Some of them use predicates of first-order logic to infer new facts, others are probabilistic reasoners.

Reasoners mainly use one of the following types of reasoning:

- **Deductive, progressive or forward chaining (FC) reasoning:** reasoning also called fact-directed chain, based on the modus ponens (evidence, symptoms, data, conclusions, hypotheses), whose input is the basis of facts, basis of rules or objectives. In this method, it is based on data available on the basis of facts, and from it more data is extracted using inference rules until a goal is reached. An inference engine that uses forward chaining searches for the inference rules until it finds one in which the antecedent is fulfilled (using the starting data); execute the rule and infer the consequent. It consists in applying rules to the knowledge of the factual basis to obtain new results.

This type of reasoning is characterized because it is an intuitive deduction, in addition to facilitating the formalization of knowledge when making a natural use of it, so it can be used in an exploratory way. On the contrary, it presents a series of problems:

- The search is not located in the objective, and a combinatorial explosion occurs, that is, deduces facts not related to the solution. Handles few data and many possible conclusions which makes it unspecific (triggers "all" possible rules).
 - Inference engines that use this type of reasoning usually use the RETE algorithm, a pattern recognition algorithm used to infer knowledge from a fact base, which sacrifices memory to increase the processing speed which increases the speed of the rule engines that use it.
- **Inductive, regressive or backward chaining (BC):** reasoning also known as goal-directed chaining, based on an inductive method in which, starting from the initial hypothesis, the chain of reasoning is reconstructed in the reverse order of the facts (conclusions, hypotheses data, evidence, symptoms), whose input is the basis of facts, the basis of rules and objectives. This type of reasoning is characterized in that each step implies new sub-objectives or hypotheses to be validated.

As it is extracted from the previous sequence, this type of reasoning is based on a list of goals or hypotheses and goes from the consequent to the antecedent, checking if there are data available on the basis of facts that support any consequent. An inference engine that uses this method analyzes inference rules until it finds one that has a consequent that corresponds to a desired goal. If the antecedent of the rule is met, it is added to the list of goals. It must be confirmed with data that said antecedent is fulfilled.

The main advantage of this type of reasoning is that only what is necessary to solve the problem is considered, but, as a counterpart, the solution of the a priori problem must be known. It handles a lot of information but little is relevant, and knowing the solution is more specific than the forward chaining.

- **Mixed, hybrid chain:** mechanism in which parts of the chain of reasoning that leads from the facts to the objectives are constructed deductively and others inductively. The change of strategy is carried out through meta-rules. This type of reasoning avoids the combinatorial explosion of deductive reasoning on the one hand, and on the other, improves the efficiency of inductive reasoning when there are no clear objectives.

At this point it should be clarified that the chaining address is exclusively related to the process of search and selection of rules, not the direction in which they are executed. That is, the rules are always executed "forward", if the antecedent is true, the consequent is executed.

This section includes a review of the main semantic reasoners available, indicating their main functional characteristics, **paying special attention to those that support SWRL, since it is the rule language used to specify the rules of this research work.**

4.4.1 Bossam

Bossam is an inference engine used in the Semantic Web, based on the RETE algorithm and which uses forward-chaining as a reasoning algorithm. In the last years several versions of Bossam have been developed, being each of them better and more functional than its predecessor.

The main features and functionalities of Bossam are described below.

4.4.1.1. Standard Reasoning Services

Bossam supports reasoning inference about different types of data using the forward chaining reasoning algorithm.

It has the ability to add additional expressivity, regardless of the expressiveness of the policy specification language used to define the rules and the language in which the ontology is defined, since it supports the following standard reasoning services:

- Ability to reference URIs as symbols.
- Syntax of second order logic.
- Disjunctions in the antecedent and conjunctions in the consequent.
- Support for denial of faults and classic negation.

4.4.1.2. Supported Languages and Formats

Bossam supports inference about different types of data and languages:

- RDF / RDFS (syntax RDF / XML or N3): Bossam can read and carry out reasoning on ontologies written in RDF / XML or N3. The support in N3 is limited to the vocabulary OWL and RDFS, since Bossam does not understand the formula N3.
- OWL (syntax RDF / XML or N3): Bossam supports reasoning about documents written in OWL.
- Buchingae rules: Bossam includes a non-flagged rules language, called Buchingae.
- SWRL and RuleML: Bossam supports reasoning about rules written in SWRL and RuleML. The support level of each of these languages is as follows:
 - RuleML: Supports fragments corresponding to URCDATALOG + <Naf> + <Or> - <Slot>
 - SWRL / XML: Bossam supports rules defined in SWRL / XML except those with descriptions of arbitrary classes that contain Atoms. Some SWRL built-ins are supported [Horrocks04].
 - SWRL / RDF: Bossam supports reasoning with most documents that contain SWRL rules in their RDF format, including those with arbitrary classes that contain Atoms. As in the previous case, it supports some SWRL built-ins, such as String, Comparison and Math built-ins.

4.4.1.3. Reasoner Access Interfaces

Bossam provides many different forms of interaction with the reasoner:

- Access through a command line interface.
- A programmable API for independent Java applications.
- Through a Web interface.

4.4.1.4. Other Characteristics

In addition to the main functions described in the previous points, Bossam presents a series of additional features:

- Support for answering questions, queries defined in a specific language. You can use rules to ask for data included in the basis of facts of the reasoner and get a set of bindings that contain the results of the query made.
- It is possible to call Java objects from the antecedent or the consequent of the rules using the java method based on URI. This allows you to mix Java objects with the rules and ontologies used for inference.
- Conflict resolution based on priorities: it is possible to assign priorities to the rules and be able to resolve conflicts if they occur based on that priority.
- Total Embedding of Bossam in Java applications: creation of reasoner, loading of documents in OWL and SWRL, inference ...
- It is possible to add new additional knowledge to the fact base of the reasoner that is not initially in the loaded ontology.
- It is possible to ask questions about the results of the inference.
- Allows to perform incremental reasoning.
- Allows you to define additional namespaces.

4.4.2 Pellet

Pellet is an OWL DL open source semantic reasoner based on Java, developed and supported commercially by Clark & Parsia LLC. It was originally developed at the Mindswap Laboratory in Maryland.

Pellet is based on algorithms of descriptive logic (DL), which allows it to support the expressiveness of OWL DL entirety, including nominal reasoning (numbered classes). In addition, it provides standard and advanced reasoning services for OWL ontologies, and incorporates several optimization techniques described in the DL literature, in addition to novel nominal optimizations, answer to compound questions, and incremental reasoning. Pellet can be used in conjunction with Jena libraries or with the OWL-API library. Among the main functions of Pellet are:

- Validate ontologies.
- Check consistency of ontologies.
- Classification of classes.
- Respond to SPARQL issues.

Pellet applications are numerous, given the wide range of features it provides. Pellet is the default reasoner of Swoop, an ontology editor. In addition, it is widely used in the discovery and composition of web services.

Pellet is in a state of continuous development, so that new versions are developed and implemented every day.

The main characteristics and functions of the last stable version of this reasoner are presented below.

4.4.2.1. Standard Reasoning Services

Reasoner that supports reasoning inference about different types of data using both forward chaining and backward chaining reasoning algorithm. It provides the basic inference services of DL reasoners allowing:

- **Consistency check:** service that ensures that an ontology does not contain conflicting facts.
- **Satisfaction concept:** determines if it is possible for a class to have instances, that is, if a class is unsatisfactory, and an instance of this class is defined, this will cause the ontology to become inconsistent.
- **Classification:** allows to create the complete hierarchy of classes, by evaluating the relationships between subclasses. This hierarchy can be used to formulate queries such as: all subclasses of a concept, infer new subclasses of a concept, etc.
- **Realization or instantiation of the concepts of the hierarchy:** it allows to find the most specific classes to which an individual belongs. The realization can only be carried out after classification, since the types are defined according to the hierarchy.
- **Debugging of ontologies:** Pellet supports both detection of unsatisfactory concepts in an ontology, as diagnosis and error resolution (for example, indicate why the error occurs or how the dependencies between classes influence the propagation of the error). To do this, Pellet has mechanisms that accurately locate the axioms that cause inconsistencies and the relationship between unsatisfactory concepts. The first of the tasks is an easy task, but the second is not usually provided frequently.

In addition to these services, Pellet supports reasoning taking into account the full expressiveness of OWL-DL, and it also allows reasoning while considering some features of OWL-Full.

4.4.2.2. Supported Languages and Formats

- **OWL-DL:** Pellet is based on "tableaux" algorithms (descriptive logic algorithms, DL), which allows it to support the expressiveness of the descriptive logic of OWL-DL in its entirety, including nominal reasoning (classes listed). Pellet supports all the features proposed in OWL 1.1, with the exception of n-ary data types.
- **OWL-Full:** Pellet supports reasoning about the inverse functionality properties of the OWL-Full data types.
- **OWL 2:** all versions after Pellet 1.5.2 have OWL2 support, language that includes limited cardinality constraints, axioms of complex subproperties, restrictions of local reflexivity, reflexive, thoughtless, symmetric and anti-symmetric properties, disjunctive properties, assertions of negative properties, share vocabulary among individuals, classes and properties, and ranges of user-defined data. The support for reasoning about OWL2 constructions has been improving both in compatibility with OWL2 as well as robustness, as new versions of Pellet 2 were developed.
- **XML-S:** supports the reasoning on all internal XML Schema data types (numeric types, strings, and date-time types), in addition to any user-defined data type that extends the numeric types or the date type / time

- SPARQL / SPARQL-DL: from the first versions of Pellet, this reasoner includes an ABox question engine, which supports answers to a conjunction of questions. In Pellet 1.x SPARQL was used for the formulation of this type of questions that allowed queries of type SELECT, CONSTRUCT and ASK but not DESCRIBE, OPTIONAL or FILTER. To answer these types of questions, Jena's SPARQL queries engine could be used in a Pellet inference model. However, the internal Pellet engine in versions less than 2.0 only supported ABox queries that did not use the owl predicate: sameAs or owl: differentFrom.

Since version 2.0, a new engine has been incorporated that can respond queries expressed in SPARQL-DL. This engine allows to answer mixed ABox / TBox questions, and supports special predicates. The SPARQL-DL engine is integrated into the Jena ARQ engine, which is why it deals with complex constructions of the SPARQL algebra, such as OPTIONAL, UNION and FILTER. Also, response with BGP (Basic Graph Patterns).

- SWRL: Pellet provides an implementation that allows DL-safe rules to be extended to OWL-DL. It includes a "parser" of rules in SWRL that allows to load and reason DL-safe rules encrypted in SWRL. In initial versions of Pellet, the reasoner's integration (using the indicated parser) with SWRL DL-safe rules was very basic, supporting only SWRL's built-in comparison functions. This integration improved significantly from version 1.5.2 to version 2.0, version from which Pellet supports all the built-ins functions of the SWRL specification including the built-ins for URIs, except those related to RDF: List.

For its part, the DL-safe implementation is useful for small or medium size ontologies. Thanks to the parser, the SWRL files can be loaded directly into Pellet and the rules will be analyzed and processed without problem. To treat SWRL rules you can also use the OWLAPI or Jena interfaces.

On the other hand, for the first-order logic a theorem tester is needed to specify axioms. With FOL expressivity is provided but reasoning is lost, so the management of a large knowledge base and axioms is not computationally treatable. Due to this circumstance its use in a large environment is not recommended.

4.4.2.3. Reasoner Access Interfaces

Pellet provides many different ways to access the reasoner's skills:

- Access from command line through CLI (Command Line Interface), included in the reasoner's distribution package. The CLI interface allows users to easily and intuitively access all Pellet features (new or existing). The uniform GNU-Style interface provides a set of subcommands that can be used to control consistency, classification, realization, questions / answers, extraction of inferences made, and module extraction. In addition, the CLI interface accepts multiple ontologies as input, so it is not necessary to join them manually or create a new ontology that imports them all, as it happened in previous versions of said interface.
- A programmable API that can be used in an isolated application.
- Through the interfaces of Jena and Manchester OWL-API, which allow to connect Pellet with them.
- Through direct integration with the SWOOP ontology editor.
- Through the DIG protocol, which allows different clients, such as Protégé, use Pellet.
- Through direct integration with the Protégé ontology editor.

4.4.2.4. Other Characteristics

In addition to the main functions described in the previous points, Pellet presents a series of additional features, which have been evolving and improving throughout all its versions:

- Analysis and repair of ontologies: Pellet incorporates a set of heuristics that can detect OWL Full ontologies that can be expressed as OWL DL ontologies, and can repair them properly.
- Integration of Pellint: Pellint is a tool for "spinning" ontologies integrated in Pellet that can be executed directly from the Pellet command line interface (CLI). Pellint detects modeling styles in OWL ontologies that degrade the reasoner's performance. It provides functionality both at the RDF level and at the OWL level.
- Classifier optimized for OWL2-EL: Pellet has a classifier optimized for OWL2-EL. Pellet automatically detects if an ontology corresponds to the expressivity of OWL2-EL and if so uses the optimized classifier. This functionality increases the speed and improves the memory usage in the classification task.
- Extractor module: Pellet incorporates an extractor module, which allows to extract a subset of an ontology for a set of given conditions. This module extracted from the ontology will be much smaller, therefore, it will be more understandable by humans and more manageable and processable by tools.
- Reasoning and incremental classification: Pellet allows to update the classification results according to the changes that occur in the ontology. The incremental classifier uses Pellet to calculate the initial class hierarchy of the ontology, but once this is done, when the ontology is updated only the parts involved are recalculated using the classifier directly. This characteristic is very important, since it makes it possible for Pellet to reason about knowledge bases that change, that are updated continuously. Pellet supports incremental classification and incremental consistency control.
- Customizable inference engine: Pellet provides a customizable inference engine that can be used to export inferences from the reasoner quickly. This extractor can be configured to select which type of inferences should be extracted, which gives users more control in the inference process.
- The Pellet Time API and its internal uses allow users to interrupt the reasoning process by defining their own timeouts. Users can define a timeout for basic reasoning services and Pellet will stop the reasoning process when this time has ended.

In addition, the developers of Pellet work constantly to improve the performance of the reasoner, both in reasoning time and in memory usage. For the former, one of the most significant changes was the improvement of the consistency control efficiency, the main service in the reasoning process, especially significant in the management of ontologies with a high number of instances. In the case of memory, the objective is to reduce the requirements of memory imposed by Pellet, for which the internal structure of Pellet has been modified.

4.4.3 Other Semantic Reasoners

4.4.3.1. Hoolet

Hoolet: semantic reasoner that handles SWRL rules and ontologies defined in OWL-DL, whose working principle is first-order logic. The ontology is translated into a collection of axioms in an obvious way based on OWL semantics. This collection of axioms is then passed to a component that is responsible for testing the logic of first order to check the

consistency of the axioms. It cannot be said that Hoolet is an effective reasoner, as simple as the proposal is, it is very unlikely to make its scalability feasible and feasible. However, Hoolet provides a useful tool to use in small illustrative examples.

Hoolet is implemented using the OWL API to parse and process OWL, and the Vampire test engine for reasoning tasks. An implementation of a prototype of Hoolet is available, consisting of a graphical user interface that allows loading ontologies and rule sets, together with the reasoner, whose latest version dates from 2004. This prototype is very simple and is only developed for the Linux operating system.

4.4.3.2. Fact/Fact+

Fast Classification of Terminologies, FaCT, is a classifier based on descriptive logic (DL) used to test the satisfaction of logic models. It includes two reasoners, one for the SHF logic and the other for the SHIQ logic. Both reasoners use solid algorithms.

The general characteristics of FaCT are:

- Its expressive logic (in particular, the reasoner SHIQ): SHIQ is sufficiently expressive to be used as reasoner of DLR logic, and therefore, to reason with databases.
- Their ability to reason with arbitrary knowledge bases: for example, the knowledge bases that contain general concepts of inclusion of axioms.
- Its optimized implementation of tables: this implementation will become standard for descriptive logic systems.
- Its client-server architecture based on CORBA (standard architecture for distributed object systems).

In terms of its implementation and system requirements, FaCT is written in Common Lisp (programming language used in the world of artificial intelligence), and has been successfully executed in several open source commercial environments, such as GNU. The source code is available under the GNU public license, so FaCT can be executed on any system where Lisp is available. Executable code is also available for Linux and Windows OS. In addition, a FaCT server running on a host can be used on any system that has access network to the server, via its CORBA interface.

On the other hand, FaCT ++ is the new generation of the reasoner FaCT OWL-DL, described in the previous paragraphs. FaCT ++ uses the solid FaCT algorithms, but with a different internal architecture. In addition, C ++ has been used for the implementation of FaCT ++, in order to create a more efficient software tool and maximize portability. In addition to the differences indicated, new optimizations and some added features have been introduced.

FaCT ++ has been made public under the GNU license and you can download both the binary file and the source code.

It does not support the SWRL rules language, which is a great inconvenience for its possible use as a DRM ontology based reasoning engine, object of this research work.

4.4.3.3. KAON2

KAON2, is a semantic reasoner and manager of ontologies aimed at business applications, which supports OWL DL, part of SWRL and F-Logic. It is jointly developed by the following institutions: the IPE (Information Process Engineering), the AIFB institute (Applied Informatics and Formal Description Methods) of the University of Karlsruhe and the information management group (IMG) of the University of Manchester.

KAON2 is a successor to the KAON project, from which it mainly differs in the ontology definition language it supports: KAON is used as a proprietary extension of RDFS, while KAON2 is based on OWL-DL and F-Logic. Despite being a continuation of KAON, KAON2 is a new system not compatible with KAON.

The latest stable version of KAON2 dates from June 2008. This version is freely distributed to university students and for non-commercial academic use (research laboratories that are not considered universities). However, there is another commercial version of KAON2, called OntoBroker OWL.

4.4.3.4. Racer Pro

RACER (Renamed Abox and Concept Expression Reasoner), is a semantic web reasoner and information store, based on solid and fully optimized algorithms. RacerPro is the name of commercial software.

RacerPro has its origins in the area of descriptive logic. Since the logic of descriptions provides a basis for the standardization of ontology languages in the context of the semantic web, RacerPro can also be used as a system to manage ontologies of the semantic web based on OWL and as a reasoning engine for the inference of knowledge from them.

RacerPro is implemented by Common Lisp and is available as a server program to search or for research purposes, on Linux, Windows and Mac OS X. RacerPro client programs can easily connect to the RacerPro server through sockets-based TCP / IP interface. A client interface for Java is available.

RacerPro has many applications, among them, applications that require queries to large databases. When working with a large amount of information and consultation, you cannot ignore the results obtained, so it is necessary to develop storage systems that provide persistence to ontologies and query languages, which are efficient and scalable. In addition, due to its scalability, its use extends to several fields: Semantic Web, Electronic Business, medicine and biomedicine, Vision based on knowledge and natural language processing, knowledge engineering, process engineering or software engineering.

Like the rest of semantic reasoners, RacerPro is in a state of continuous development, so that every day new versions are developed and implemented.

4.4.4 Discussion

In this section we have presented and analyzed some of the most widely used semantic reasoners today, software components that allow inferring additional knowledge. The OWL2 language has the capacity to infer certain additional information that is not explicitly contained in the definition of the ontology, but this inference of new knowledge is still limited. For this reason, semantic reasoners allow to increase said capacity of reasoning, beyond the intrinsic capacity of OWL2.

When comparing and classifying semantic reasoners, their main characteristics must be analyzed, grouping them into three dimensions:

- Reasoning characteristics: reasoning algorithm used, expressiveness, incremental classification capacity, external knowledge capacity, consistency control, rules support, SWRL support level and reasoning ability.
- Usability: OWL API interface, Jena API support, Protégé plugin, implementation language, usage platforms, user license and if documentation is available.
- Performance and efficiency: for those that support SWRL, it is important to compare the loading time of the ontology and the total inference time.

The choice of a reasoner or another depends on the specific requirements of each programmer, who will choose the one that best suits the needs of the system, taking into account the reasoning characteristics, usability and performance of each one, whose analysis is provided.

In the specific case of CTI and DRA/DRM proposed ontologies, the reasoner must satisfy three fundamental requirements, previously indicated: SWRL support and SWRL built-ins in special Comparison built-ins and Date, Time and Duration Built-ins; ability to introduce external knowledge to the facts base of the reasoner and incremental classification, and optimal inference speed and load. In addition, it is desirable that the reasoner have as many interfaces as possible (OWL API, Jena, etc.), and whose license to use is public and open source.

FaCT ++ does not support the SWRL language, so its use is ruled out. On the other hand, Hoolet has SWRL support but does not allow the use of built-ins in SWRL rules, nor does it have the capacity to introduce external knowledge or perform incremental classification, in addition to not having a developed and proven implementation of this reasoner, but only of a simple prototype available for Linux, whose latest version dates from 2004. Therefore, its use in the proposed initiative would also be ruled out.

On the other hand, KAON2 and RacerPro do not support the built-ins of SWRL and present problems when loading ontologies with SWRL rules, so they would be discarded.

Bossam, Pellet and newer versions of RacerPro, perform incremental classification, and support SWRL and its most common built-ins. Making a comparison between the three, one could conclude that Pellet is the best option for the CTI and DRA/DRM based ontologies, since it has a higher level of support of SWRL built-ins than Bossam and RacerPro, has the capacity to introduce external knowledge, have greater usability (OWL API, Jena, direct plugin in Protégé, usable in all platforms (Windows, Linux, MAC OS X) ...), and is developed in Java code. In addition, its inference and load times are much lower than those of Bossam, and it has an open source free license, unlike RacerPro.

This chapter describes different formal ontology representation languages (OWL, RDF, DAML + OIL, etc.), several policy specification languages or rules in the Semantic Web (SWRL, REI, KAoS, SWSL, etc.) and existing semantic reasoners that allow obtaining new information based on facts and conditions.

When specifying knowledge or information, you can use formal languages, such as those described in the chapter, or normal languages. The benefits of using formal semantics are:

- Definition and precise clarification of the semantics of language.
- Possibility to computationally reason with the information elements and extract conclusions, which allows:
 - Infer new information not explicitly established.
 - Check the consistency of the defined knowledge models, that is, that the information elements are coherent with the restrictions.
 - Verify the correctness of the information models, that is, that the use made of the language is coherent with its semantics.
 - And all this, with the support of tools that can interpret in a rigorous way the meaning of the defined information models.

The advantages that are expected to be obtained for the specification of the behavior of a CTI and or DRA/DRM through the application of behavioral representation techniques through ontologies are:

- Ease of application, since the policies will be defined in the same language of ontologies, interpretable by the same semantic manager with an engine of inferences and rules
- Expressiveness improvements for the specification of the different types of policies: obligation, authorization, etc.
- Advantages in terms of flexibility and extensibility of definitions and the possibility of introducing policy changes dynamically
- Reasoning possibilities on behavioral specifications that improve the current treatment of policies: detection of conflicts, inconsistencies of information, etc.
- Possibility of working from the same management system with ontologies of different levels of abstraction: level of network and systems management, level of service management, level of business management.

Therefore, ontologies are a mechanism that provides a shared and consensual understanding of knowledge within a domain. This understanding can be communicated between people or between heterogeneous systems.

4.5. CONCLUSIONS

In this chapter, the state of the art of Ontologies, Behavioral Languages and Semantic Reasoners have been presented.

Several Ontology definition languages were analyzed and compared (see section 4.2). At the same time, several Behavioral languages to represent patterns or reactions to knowledge were analyzed (see section 4.3). In the last section, an analysis of the different alternatives with regard to semantic reasoners was also considered (see section 4.4).

Despite the great benefits of most of the options, there are very few options that can contribute for the definition and implementation of our DRM Framework. That is the case of:

- OWL2.
- SWRL.
- Pellet reasoner.

A combination of the three of them will be used for the whole DRM Framework as described in the next chapters.

CHAPTER 5. GLOBAL ARCHITECTURE PROPOSED

This chapter motivates and introduces the proposed architecture.

The architecture, based on ontologies and SWRL, will allow the connection of two security domains: the domain of risk (analysis and risk management) and the domain of threats (cyber threat intelligence, CTI). This introduces a new hybrid model with the possibility to inference new knowledge, which can also be shared through the Intelligence Information sharing module.

5.1. INTRODUCTION

We present the Cyber Threat Intelligence (CTI) model considered in this work also as an evolution of current offering in the market today. The main purpose of threat intelligence is to have as much sophisticated knowledge as possible about unknown threats either for having better detection capabilities or having better prevention ones. For example, an emerging threat, which is affecting similar entities, would be able to affect our organization with high probability if we share the same threat actors' motivation. Detecting this type of complex connections by inference and expressiveness, will allow us to provide enhanced prevention and detection services.

Automation is needed due to the evolution in the number of threats, services to protect, lack of resources and complexity of each organization. Some of the most relevant and common cybersecurity needs with regard to automation are related to:

- Enrichment of Threat Information Context.
- Security Event Detection.
- Security Incident Detection and Prevention.
- Incident Triage by Severity (Incident Handling).
- Information Sharing Control (why, what, when and how).
- Risk Assessment and Risk Management.

In addition to this, incidents and risks are usually managed by different teams due to the specificity of each domain (operation, tactical or strategic levels), and their tools are in most of the cases not integrated as they should. Threat Intelligence data is sometimes connected to Incident Handling, however none of them are connected to enterprise Risk Management by default.

Each domain has its specificity also with regard to the concepts they manage. There are different initiatives either in Cyber Threat Intelligence (CTI) [Obrst12] [Syed16] or Risk Management (RM) [Fenz08] [Herzog07] [Ekelhart07] [Singapogu12] trying to define clear taxonomies of each domain, but all of them in an isolated way.

They are only targeting one domain at the same time and in most of the cases it is a partial approach. Querying approaches [Gao18a] [Gao18b] are suggesting semantics but they are semantic-agnostic nor using standards.

The most relevant and advanced products in CTI are started to use open standards like STIX™, that will allow them to share threat intelligence information based on a common taxonomy that is accepted by the majority of the Industry as a standard format. STIX™ is offering real benefits by its taxonomy of cybersecurity threat domain concepts, but also defining relationships between their objects.

However, some concepts inside STIX™ cannot be represented yet, due to their limitations to describe more complex concepts like TTP (Tactics, Techniques and

Procedures) [OasisSTIX-TTP], Campaigns [OasisSTIX-Campaign] or Incidents [OasisSTIX-Incidents].

STIX™ version 2 has evolved into a more integrated approach (e.g., cyber observables are represented inside) as well as it uses JSON instead of XML (used in STIX™ v1.1) for better automation with current product offering. It still lacks full expressiveness, however v1.1 white paper [OasisSTIX1] suggested RDF/OWL as a potential future solution for it.

Despite this evolution, we need formal models to describe the meaning of, even complex concepts that cannot be described by today. At the same time, formal models would enable machines to also understand those concepts and relationships for a better and more effective automation [W3C-Ontology] [W3C-Inference].

On the other hand, Risk Assessment (RA) and Risk Management (RM) domains are using a different taxonomy to describe the type of risks, asset dependencies but also the appropriate controls (safeguards or countermeasures).

Another relevant factor that could be one of the most important reasons for not having any effective integration between RA/RM and CTI Frameworks yet, is that both domains are using different timing.

RA/RM are usually using Deming cycle (Plan, Do, Check, Act) as an improvement lifecycle that is not real time, which is different from the reality of Cyber Threat Intelligence and Incident Handling as they have to be at least NRT (near real time) or if possible, RT (real time).

Not having NRT/RT Risk Assessment and Risk Management nowadays mean, that our management is not well informed of their exposure levels on time, as well as investments are based usually on annual external consultant's estimations (e.g. standard threat probability estimations) not considering the reality of our organization's threat landscape. Risk calculations are then only taking place few times per year. Mostly, one per cycle.

Dynamic Risk Assessment (DRA) and Dynamic Risk Management (DRM) are new concepts to describe the real need to have knowledge of our risk exposure level close to NRT/RT.

There are multiple benefits of information exchange in cybersecurity, like threat prevention and detection, between others. On the other hand, there are also several unsuccessful voluntary sharing initiatives, which have been studied by multiple authors [NIST16] [Vishik13] [Tosh18] [Skopik16] [deFuentes17] [Ring14]. A summary of all the following problems can be seen in the Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions.

Nowadays, there are multiple information sharing platforms running, with millions of IoC being shared every day, and thousands of users [Sauerwein17] [MISP16]. One of the most promising models for information exchange was Open Source Intelligence (OSINT). Several entities and individuals share cyber threat intelligence data openly, along several platforms, since a lot of time ago. This promising model, based on open source intelligence, has demonstrated that is not reliable [OSINTArmy12] [NATO01] but necessary. Organizations can also use deception techniques, where they can intentionally or unintentionally add, delete, modify, or otherwise filter the information made to the general public. It is important to evaluate the reliability of open sources, in order to distinguish their objectives, factual information; bias; or deception. Non authoritative sources lack reliability and trustworthiness and seldom stand apart from authoritative sources.

When evaluating sources of information to determine reliability and credibility, we should consider:

- **Identity.** Who produced the information (for example a student, teacher, political organization, or reporter)?
- **Authority.** How much does the source know about the information?
- **Motive.** Why was the information published?
- **Access.** Did the source have direct access to the event or information?
- **Timeliness.** What is the date of the information?
- **Internal and external consistency.** Does the information contradict any policies?

Cyber threat intelligence information should be reliable by definition, if we want actionable intelligence, that is, the possibility to run automatic security processes. Organizations will only automate those processes using reliable information, if not, their processes will not be reliable at all. More automation is needed due to the trend in the number of systems to protect, but also due to the increasing complexity of our networks. If our systems are automatically loading IoC data to build filter rules, we need to trust the entity or individual behind, that is, the one who shared that IoC. If not, we can be loading incorrect IoC or even worse, we can help a malicious actor to success. It is then clear that we need to trust OSINT sources [OSINTArmy12] [NATO01], as a necessary condition, before loading those IoC in our systems. More than that, maybe it is not enough to trust the entity itself, as some IoC shared by them could be part of a third party detection, we then need to trust each piece of data. Specific validation processes, as well as sources and evidences' auditing, accounting or checks, are recommended. As an example, time-stamping related data to when that malicious behavior was seen, who shared the data, when it was updated, modified or even removed, is very interesting for a healthy and up to date system.

These platforms are based on sharing indicators of compromise (IoC) [Sauerwein17]. Once an IoC is shared by a certain entity or individual, it can be used by anyone who has access to that open platform. The usage can also be automated, for example, to protect the perimeter. If a firewall receives a new malicious IP address, it can block either the incoming or outgoing network traffic related to that IP. Several tools and plugins are available to facilitate such integrations.

Despite these integrations, all the static approaches, based on signatures, have demonstrated that they do not match the dynamic nature of new generation of threats [Bianco14], [Riesco19a], [Tounsi18]. Using IoC to fight against unknown threats, is then ineffective. New emerging techniques, tactics and procedures (TTP) remain undetected to any vendor solution, whose detection capabilities are based on known threats. Contextual or behavioral aware rules, are the only plausible solution to fight unknown threats. Patterning, algorithms and rules, are the only way to detect suspicious behaviors beyond a detection strategy, based on signatures (e.g. hashes).

Vendors are usually focused on data collection instead of analysis [Sauerwein17]. They are providing unmanageable volume of (big) threat data [Tounsi18], however, having threat Intel data is not having real intelligence. Real intelligence is associated to analyzed and processed threat Intel data. This new approach is known as advanced intelligence. Advanced intelligence might also be represented, as complex structures like a TTP (Techniques, Tactics and Procedures) [OasisSTIX-TTP]. If we were able to work in such dimension, we will not need to share and store dozens or hundreds of IP addresses, domains and or hashes but the TTP algorithm itself. The idea is equivalent to the usage of DGA (domain generation algorithm) versus the use of single domains, however we go beyond that in our proposal. A TTP description can combine a DGA together with any other concepts in the same rule or algorithm as a more effective behavioral pattern. Once such an algorithm is shared, it can be more effective than sharing hundreds of related IoC, if each of them is ephemeral. As a result, new proposals suggest the idea to work in the dimension

of TTP as a combination of intelligence data, malicious behaviors and patterns to detect them.

Defining TTP rules is not easy nowadays as there is still room for improvement with regard to the lack of algorithm expressivity [Riesco19a] [OasisSTIX-TTP] [OasisSTIX1]. CTI data taxonomies nowadays, lack of semantic, unambiguous and universal understanding. However, initiatives like STIX™ are being considered as a de facto standard and it is widespread adopted by the industry. It is probably one of the most promising standard for it. What is clear is that sharing a TTP, will be more effective than sharing an isolated IoC. Once a taxonomy or standard is agreed among a huge number of stakeholders, an efficient and effective knowledge transfer will be possible between them. Without an agreed taxonomy, it would be nearly impossible to automate detection without the need of customized parsers. We propose to use a semantic enhanced version of STIX™ to solve that expressivity issue.

5.2. METHODOLOGY

The architecture proposed will be based on the following methodology:

- **General analysis of the state of the art:** An analysis will be made of **Risk analysis and management methodologies**.
 - Intelligence standards on threats.
 - Computer security Ontologies.
- **Benchmarking analysis of reference entities:** A benchmarking analysis will be conducted on different intelligence models of different international reference entities, belonging to different sectors:
 - A national security entity.
 - A financial sector leader.
 - A (our) national CSIRT.
- **SWOT analysis of the pilot entity:** in particular, factors such as the following will be analyzed:
 - Use of standards
 - Sensor network
 - Intelligence Model
 - Rules and algorithms
 - Incident management
 - Model Analysis and Risk Management
 - Models of exchange of threats
 - Anonymization, access control, authentication and authorization
 - Exploitation, visualization, simulation.
- **Evaluation of alternatives:** The advantages of using ontologies, rule languages and a reasoner are analyzed. They become the main building blocks of the proposal to infer new and actionable intelligence. As an example, different types of processes can be enhanced like the triage, detection and enrichment of information.
- **Building blocks:** they are the following:
 - Core Components:

- (Dynamic) CTI and Risk Data-Set:
 - CTI Ontology
 - Risk Ontology
- (Dynamic) Risk and CTI Intelligence Knowledge.
- Dynamic Risk Management:
 - Dynamic Risk Assessment (DRA)
 - Dynamic Risk Management (DRM)
- Information Sharing Module
- Reasoner.
- Decision Making Level.
 - Technical
 - Tactical
 - Strategic
- External Components:
 - N-Dimensional Sensors.
 - Knowledge Base
 - Visualization and Simulation Engine.
 - Anonymization.
- **Formalization:** proposed ontologies as a formal model.
- **Verification and validation:** intelligence inference, risk re-assessment and intelligence sharing.

5.3. EVALUATION OF ALTERNATIVES

This chapter proposes the use of semantic ontologies and SWRL rule language, as a formal model, to define new type of systems either to exchange or to infer new knowledge.

The proposal will provide solutions to all problems described in Table 1. The thesis, proposes several original contributions, among them:

- **Ontologies for "cyber threat intelligence" and "risk analysis and management" domains:** a hybrid ontology will formally model both domains, it will establish enriched relationships between them as well as there will be new knowledge inferred by the reasoner. It will take care of the relationships between threats and their influence on certain risk levels, regardless of the methodology used. The formal model also provides unequivocal understanding by either people or machines, as a necessary condition, to create "actionable" intelligence.
- **Enrichment, contextualization, generation and inference of new intelligence about risks and cyber threats, by using SWRL rules:** when using unambiguous concepts, any data that we receive from our own sensors or from external sources, will have an unambiguous meaning. This will automatically allow the establishment of all relationships between individuals, as well as a better contextualization, better discovering of new dependencies and enhanced or advanced semantic search, using semantic graphs. This will therefore allow us to infer new intelligence and make it actionable under the concept of *"adaptive intelligence: intelligence that not only responds to changes, but also creates changes"*. As an example, the system

could automatically find new relationships that allow new alerts to be triggered automatically, without human intervention.

- **Dynamic risk analysis:** through ontologies and SWRL rules, we will allow an entity to semantically and dynamically address the different phases of its risk analysis process, related to its own information security management process (ISMS):
 - **Asset inventory:** enabling the discovery of new related assets with certain services and keep an adequate asset inventory control.
 - **Identification and discovery of dependencies among them.**
 - **Asset valuation**
 - **Dynamic identification and inventory of threats.**
 - **Evaluation of threats.**
 - **Calculation of Impact and Potential Risk.**
- **Dynamic risk treatment:** through ontologies and SWRL rules we will allow an entity to address semantically and dynamically, the different phases in the treatment of risks:
 - Time horizon:
 - Potential state without safeguards.
 - Potential state with current safeguards.
 - Potential state with new safeguards.
 - Optimal selection of new safeguards
 - Risk map:
 - Impact.
 - Risk.
 - Aggregate impact.
 - Aggregate risk.
- Selection of optimal strategy: (with simulation capability)
 - Transfer risk.
 - Mitigate risk.
 - Monitoring.
- **Intelligence Exchange in the form of SWRL rules beyond IoC,** since the rules take the form of potentially interchangeable algorithms, which are potentially more effective than the exchange of IoC. As an example, if a DGA (Domain Generation Algorithm) is known, there will be no need to exchange all the possible combinations of domains. In our case, once a SWRL rule, is received as new knowledge by an entity B, it can be better protected. If it is equivalent to the entity A attacked, we will transfer effective knowledge if we share the detection algorithm instead some of the IoC used for the first attack. Due to the fact that the attacker can change easily the IoC [Bianco14], the use of context or behavioral pattern rules are the only plausible solutions to fight against these type of threats.
- **New incentives for a new exchange of knowledge:** we also propose to create a system based on Blockchain, to overcome several barriers seen in Table 1 with regard to information sharing. A new "Marketplace" of intelligence information and a token (as its digital representation), is proposed for example to exchange high rated

SWRL rules (e.g. to detect a new type of APT threat (advanced persistent threat)). This type of SWRL rules will represent advanced intelligence rules, also called TTP (Techniques, Tactics and Procedures).

Summarizing, the main advantages of using ontologies in the present architecture are:

- They allow the sharing of information and establish their relationships without ambiguities, a matter that is not trivial in any domain, especially in the domain of information security or cybersecurity.
- They allow this information to be understood both by people and by machines / applications, which is essential in information exchange systems and dynamic risk management systems given the high volume and the importance of the time factor to react on threats. The systems and applications must understand the language, the concepts and the relationships between them.
- They allow to reuse knowledge, in our case, we reuse intelligence knowledge about cyber-threats (intelligence ontology on cyber-threats) from the ontology of risk analysis and management.
- They formalize the knowledge in one or several concrete domains, in such a way that they facilitate the understanding by all the actors and systems or applications.
- It allows the reasoning and automatic inference of knowledge, this being a current problem in the case that concerns us.
- It allows establishing a meeting point between different methodologies and information models, such as the case of risk analysis and management where there are several methodologies and standards.
- Etc.

Ontologies offer great advantages, especially thanks to their formalization and great expressiveness in multiple fields, but also in the field of computer security. They are especially useful in those domains where there are different information models, different formats and methodologies.

Thus, concepts that, a priori, are equal, could be understood as different concepts. This can complicate the integration and interoperability of the systems that handle this type of information. If the information can be helpful to avoid a cyber-attack or a major damage, there must not be problems with regard to the re-usability, interpretation and action ability of the information. Nobody would understand that in a case of a national security threat with risk to people.

Thus, through the use of ontologies, we could establish initial "pairing" mechanisms towards the concepts of ontology and, due to its unambiguous meaning, the risk of erring in parsimony due to misinterpretation of the meaning of the concept would be minimal. At the same time, once the parsing was done, the concept would inherit the degree of expressivity of the ontology, allowing the systems and applications to interpret their meaning unequivocally.

Our two domains are:

- **Domain of Cyber Threat Intelligence:** where there is a great variety of data formats, absence of standards in its syntax and numerous proprietary implementations of manufacturers. The use of ontologies would allow us to integrate all this variety of formats and make them compatible with each other for those concepts that are equivalent or have the same meaning. The systems deployed in the current architectures could interoperate with each other but for this, they will need this formal model that has not existed to date. There are proposed standards

such as STIX™ (Structured Threat Information eXchange) of OASIS (advanced open standards for the information society) that has evolved a lot in the last year, going from a version 1.1 to a version 2.0 with a lot of changes and modifications. It goes from using XML format to JSON but, most importantly, it identifies the minimum necessary fields that are of interest to the numerous promoter entities and the relationships between the different concepts. However, its implementation is not addressed from the point of view of a formal model, nor does it specify specific fields due to its lack of expressivity, such as the TTP (Techniques, Tactics and Procedures), which are the most relevant when it comes to coping to a threat. Once the TTPs are identified, the attackers should have to completely change their attack method, the exploits or vulnerabilities used, if they want to continue insisting on achieving their goals [Bianco14]. On the other hand, how threats have been managed so far, based on commitment indicators (IOC) such as IP addresses, domains, hashes, etc. they are completely ineffective since it is trivial, to attack from an IP, domain or with different hash for example for an attacker. It is unfeasible to block threats based on these ephemeral or lower-valued indicators, since all possible modifications to those values (IP, domains, hashes, etc.) would have to be available, which is impossible (their variability or mutation capacity tends to infinite).

- **Domain of Risk Management:** where there is also a wide variety of standards, methodologies, frameworks, concepts and syntax, even in some cases, there are specific organizations forced to use some of them, as the case of the National Security Scheme (ENS) and MAGERIT® by the Spanish administration. The need for coexistence and interoperability between the different options is key since entities, not only incorporate information security management systems (ISMS), but also other quality management systems, and should not replicate or maintain parallel management models for each one of the accreditations or management systems. On the other hand, risk management analysis models are fundamentally static models, with periodic measurement and where the fundamental variables to decide investments depend on expert judgment. Variables such as the probability of materialization of a certain threat in a particular organization, have ranks that oscillate from occurrence of none, 1 or several times a year and due to its complexity, it is greatly simplified but leaving a loose approach to a serious problem that implies dedicate resources for nothing despicable. Similarly, the value judgment of the external expert is not connected with the reality of the organization, that is, with real, current or past incidents, so that, if two entities are not equal, why does the expert consultant apply equivalent probabilities for the same types of threats using simplistic approaches without a scientific basis if this can often lead to investments? The proposal of the present doctoral thesis includes an evolution of the probability in such a way that the judgment of the expert is weighted dynamically. In this way, it adjusts or weights, reducing to a large extent the subjective burden of the consultant and adapting variables such as probability to the reality of each company at the same time that it is endowed with a scientific basis.

In addition to the representation of concepts, ontologies will allow us to represent their behavior. For a system based on the continuous change of the level of threat, and thus the level of risk, it is also necessary to be able to represent the knowledge of the situation (cyber situational awareness), but the ability to represent the knowledge of the reaction is also needed, especially in certain events. This is where the definition of rules, restrictions and policies is extremely useful.

For the specification of the behavior of the system, rules expressed in SWRL will be used. SWRL has been chosen for various reasons, among them:

- Direct integration with OWL / OWL2.

- Greater capacity for the representation of behavior, including the ability to define all kinds of more general rules. OWL instead requires certain predefined restrictions.
- Greater level of expressivity allowing variables to be applied to the whole set of concepts, and to any part of the rule (background and consequent).
- Supported by most of the semantic reasoners.
- Simplicity regarding policy definition languages.

In summary, in this Doctoral Thesis **we will use OWL2 for the definition of knowledge and SWRL to define behavior** throughout all the different modules.

All components can be divided in two different types:

- **Core components:** relevant building blocks, which are addressed by the thesis.
- **Other components:** partially or not addressed by the thesis.

5.4. RELATED WORK

Since 2007 different authors have been investigating the usage of ontologies for risk domain. It is the case of Herzog et al. [Herzog07] where authors define a generic security domain ontology specified in OWL which covers most of the aspects of an information security domain. It provides a detailed vocabulary as well as it supports reasoning capabilities. It is built on classical risk assessment concepts: asset class, threat class, vulnerability class, countermeasure class, security goal class, defense strategy class. Authors provide with some detailed subclasses and relationships between them.

Fenz, S. et al. In [Ekelhart07] contribute with ontologies for a quantitative risk analysis in which authors visualizes the damage caused by specific threats, outage costs and the recovery time. Running the program with added safeguards shows their benefits and offers objective data for decision making: which safeguards to implement and to avoid installing countermeasures that are not cost-effective. Authors thus justify the need to have a security ontology to clarify the meaning and interdependence of unambiguous IT-security relevant terms which then can be used to facilitate qualitative risk analysis and decision processes.

Fenz, S. In [Fenz08] contributes with ontologies to define IT Security Metrics. Author plans to align it with ISO 27004 standards and to apply it in real world audit scenarios as well as to go further in the degree of automation.

Fenz, S. et. al. In [Goluch08] integrates an ontological information security concept in risk-aware business process Management. The ontology is based on NIST and authors provides threat, vulnerability and control sub-ontologies. Authors propose to improve and extend the threat classification in order to consider the threat for human life as the main priority in case of any risk. In addition to this, supplemental information to be considered by the ontology can provide valuable and essential details for decision makers.

More recent work like Vergara et al. [Vergara10] is used to propose a model based on ontologies to integrate and share alerts between different Security Information Management Systems.

In Mateos, V. et al. [Mateos10], the authors propose an Automated Intrusion Response System (AIRS) based on ontologies that is to use reasoning to select optimum responses. The system will infer the optimum responses at network level (e.g. Intrusion Prevention Systems). This time, they work at network domain.

Obrst, L. et al. [Obrst12] introduce a proposed ontology for Cybersecurity, especially as an extension of MAEC (Malware Attribute and Enumeration Characterization). Authors use as a reference the Diamond Model of Malicious Activity.

Singapogu, S. et al. [Singapogu12] describe a proposed ontology for making enterprise risk assessment by supporting the IT Security Risk analysis process.

Erbacher, R.F. [Erbacher15] developed a packet-centric ontology named PACO which allowed them to represent and capture the atomic elements of network communication, i.e., packets and sequences of packets. It is a proposed model as a basic for more holistic approaches.

Syed, Z. et al. [Syed16] worked on an integration between STIX™ and ontologies for Situational Awareness which is a very interesting approach. They demonstrated the benefits for different use cases (vulnerabilities associated with PDF readers, suggestion of similar SW, etc.) as a very interesting contribution for example to check the impact of changing vendors.

With regard to querying the data, some approaches [Gao18a] [Gao18b] suggest semantic but they are still semantic-agnostic nor using standards.

Meszaros, J. et al. in [Meszaros17] propose a framework for online service cybersecurity risk management applied to a large enterprise. The risk model is providing simplicity to manage either by providers or consumer's viewpoints. It is also aligned with standards [ISO/IEC27000:2018] [ISO/IEC31000:2018].

One of the most recent as well as interesting works it is the work of Qamar S. et al. [Qamar17]. Authors implemented the ontology version of STIX™ (version 1.X), together with CVE and Network ontologies to build STIX™ analyzer, a framework to perform data-driven analytics for threat intelligence and information sharing. It is based on known shared and threat data from threat repositories. One of its main use cases is working on attribution. Authors also provide a way to simulate and calculate some risks based on exposure levels. It is not focused on leveraging an organization threat intelligence data having a near real time detection, protection and risk management.

We propose a complete dynamic risk management framework compatible with any widespread risk assessment and management standards. We also provide the needed expressivity and granularity for risk management frameworks like for example considering the differences between the likelihood of threats based on knowledge, access rights and behaviors of each of our users. Our proposal is also focused on behavioral detection of new and still unknown threats (when IOCs are still not available in any intelligence repository or feed) and complex TTP behaviors. We also work with the new version of STIX™ 2.0 which has several improvements from version 1.X, like an integration between CyBox™ and STIX™.

Poolsappasit, N. et al. in [Poolsappasit12] proposed a very interesting Dynamic Risk Management model based on Bayesian Attack Graphs, using conditional probabilities to encode the contribution of different security conditions during system compromise. They estimate an organization security risk from different vulnerability exploitations based on the metrics defined in the Common Vulnerability Scoring System (CVSS) [CVSS].

Mozzaquatro et al. in [Mozzaquatro18] provide an interesting approach based on some of our building blocks like OWL, SWRL and a reasoner for detecting, identifying and classifying vulnerabilities (or bad configurations) of IoT devices. They propose 3 layers (design, run time and an integration layer). The reasoner is used to propose specific measures to improve vulnerabilities or bad configurations (e.g., if a WEP configuration is detected then it is best to use WPA2). At the same time authors use signature matching sensors and IMDEF format. However it is not oriented to a risk assessment or management framework like our work, it can potentially be connected with our framework to benefit from it. At the same time, our proposal goes beyond IoT, vulnerabilities and IMDEF. We implemented the whole STIX™ spec draft creating its ontology version to cover as much types of threats as possible leveraging from that standard draft. We also use STIX™ along the whole system (sensors work, understand and deliver STIX™ format data). We also

propose a fully integrated domain of STIX™ threat and the risk domain. In our case, using SWRL and the reasoner we are able to use it for detection at sensors level but we are also able to detect malicious patterns without knowledge of specific signatures. We create rules based on SWRL following a certain pattern to make reasoning proposals within risk management and CTI domains once those patterns have been detected. As a future research direction, authors in [Mozzaquatro18] propose the usage of artificial intelligence and Bayesian networks to overcome the limited detection capabilities of signature-based matching sensors. As demonstrated in our work, we consider that ontologies, SWRL and reasoners can be used together with STIX™ to handle the needed expressiveness to detect patterns of unknown IOCs.

Despite all references, **no one is providing a solution to have a near real time dynamic risk framework based on dynamic threat detection.** Even all of them are based on threat data (mostly IOCs) but they are not based on patterns or behaviors. Semantics has been used in some of the references provided even in very recent works as a good solution for the current lack of expressiveness.

Both domains need to be connected (cybersecurity threat intelligence and risk domains) completely and the connection should be based on standards to enable risk and threat information sharing.

By using and developing the entire STIX™ new version 2 in OWL format, we are promoting the widespread industry driven taxonomy for threats but in its semantic version to overcome their limitations with regard to expressiveness [OasisSTIX1] [OasisSTIX-TTP] [OasisSTIX-Campaign] [OasisSTIX-Incidents].

Inference [W3C-Inference] by our Pellet reasoner [W3C-Pellet] will enable automatic threat and risk data discovery. At the same time, the usage of SWRL [W3C-SWRL] rules, will also enable the usage of extended and enriched algorithms.

As an example, we could provide the automation needed to detect unknown threats based on patterns like detecting any technique, tactic and procedures (TTP) [OasisSTIX-TTP] with enough expressivity. This kind of detection techniques will go beyond current approaches based on known IoC (indicators of compromise). At the same time a detection is done, the system will infer risk re-calculations dynamically. SWRL rules will be able to give solution to any type of algorithm, it can be a cybersecurity threat intelligence algorithm, a risk domain algorithm, or a mixed CTI-risk one.

By using SWRL [W3C-SWRL] we will also simplify the creation of either threat or risk algorithms especially for non-technical people. One day training on domain ontologies and the usage of SWRL will be enough. Until today, those who create algorithms needed development skills [Zhang17].

Figure 48 Ontology representation of some instances of our integrated CTI and DRM Framework

Information exchange was mainly invented to help others, by sharing our knowledge, at the same time we improve our level of protection, thanks to others who share their knowledge. It is clear that no one has enough knowledge to consider fighting alone against any cybersecurity threat.

Wagner, C. et al. in [MISP16] presented MISP, the Malware Information Sharing Platform which has been one of the most used open source platforms to share threat intelligence among several stakeholders since 2016. They created a centralized platform which can be instantiated by anyone due to its open source license. They observed from 20 to 40 new instances deployed each day since its first release. MISP allows the exchange of IoC (hashes, hostnames, IP addresses, URL, etc.) mainly. The high volume of data been shared since the beginning (about 100K md5 hashes a month) demonstrated the real need and interest of the end users. On the other hand, authors propose to work over the quality of the data, as the data that is usually shared, is not always reliable. Due to they work in the IoC domain, they suggested to that new ways to handle such volume of data in the future, like data mining techniques, will be needed. In the beginning they were facing potential issues of reliability of the data, storage and processing but also they detected that a formal model or taxonomy was needed as new concepts like bank accounts were needing a data model representation. They also decided to simplify things by using tags at the beginning which lead them to a bad and complex situation, so they realized that a taxonomy was a good idea, however their taxonomy is just a triple tag to overcome that issue which still lacks of the needed expressivity.

In 2019, the French national cybersecurity agency (ANSSI), the CERT-EU and the Luatix non-profit organization are trying to develop an open source platform named OpenCTI [OpenCTI], allowing organizations to manage their cyber threat intelligence knowledge and observables. It has been created in order to structure, store, organize and visualize technical and non-technical information about cyber threats. They are using GraphQL (Graph Query Language) and created a knowledge schema based on STIX v2. Despite it is a promising open source option to handle and visualize the organization's STIX data, it is neither semantic agnostic nor using standards. Despite it is still under

development, it promises different connectors with MISP [MISP], TheHive and MITRE ATT&CK [MITRE-ATTCK].

Sauerwein, C. et al. in [Sauerwein17] analyzed 22 threat intelligence sharing platforms. They compared them following a methodology that included 2 workshops with security experts from the industry. Their key findings are really interesting, among them:

- There is no a common definition of threat intelligence sharing platform.
- STIX™ is the facto standard for describing threat intelligence.
- Platforms primarily focus on sharing of IoC.
- The majority of platforms are closed source.
- Most platforms focus on data collection instead of analysis.
- Trust issues between users and platform providers are mostly neglected.
- Academic and commercial interest in threat intelligence sharing increases.
- Many manual tasks make the user the bottleneck.

As a result, authors indicate that the threat intelligence sharing, research and practice lacks a consistent definition. They also indicate that threat intelligence sharing is comparable to data warehousing and it does not provide real intelligence. We agree with authors in all these conclusions.

De Fuentes, JM. et al. in [deFuentes17] indicate that privacy is paramount to foster cooperation, particularly when insecure infrastructures are used to support sharing. They propose PRACIS, a protocol that provides privacy-preserving and aggregatable cybersecurity information sharing. PRACIS provides these properties by leveraging existing format-preserving and homomorphic encryption techniques and adapting them to the particularities of standard message formats such as STIX™. Authors anyway provide a new layer of security thinking that the network is untrusted.

We propose a network based on blockchain precisely to overcome the lack of trust in the network. Moreover, the system, running as a dApp or smart contract instanced in the blockchain, will be ready for external calls either by each different user role or from other smart contracts (e.g. other CTI Marketplaces). There is no human intervention after it is deployed in the blockchain. As a good practice, the source code is usually appearing along the smart contract instance, which give more trust to the users (the ABI²⁰), equivalent to the running code, is checked to confirm if it matches with the source code).

Tounsi, W. et al. in [Tounsi18] suggest that the static approach of traditional security, based on heuristic and signatures, does not match the dynamic nature of new generation of threats that are known to be evasive, resilient and complex. Organizations need to gather and share real-time cyber threat information and to transform it to threat intelligence in order to prevent attacks or at least execute timely disaster recovery. Authors explains why there is reluctance among organizations to share threat intelligence. They provide sharing strategies based on trust and anonymity, reducing the risks of a business leak. They also show in their work why having a standardized representation of threat information can improve its quality. We support such statements.

Leszczyna, R. et al. in [Leszczyna19] indicate that a fine grained situational awareness linked to information sharing is needed. They propose different topologies, ranging from a decentralized peer to peer, to a confederated topology as seen in Figure 49 and Figure 50. Authors propose a framework to enhance current threat intelligence services by providing data at all levels however they propose to integrate all the data at the same

²⁰ <https://solidity.readthedocs.io/en/develop/abi-spec.html>

time especially in the context of critical infrastructure protection. They consider features like data anonymization and sanitization and the use of SIEM (security information event manager) however they still work in the IoC domain.

Figure 49 Information sharing topologies: a) Decentralized peer-to-peer sharing b) Centralized sharing through an instance.

In addition to this, their decentralized topology approach is really not a decentralized platform but several independent instances enabling any bilateral interconnection between peers. We propose a real decentralized infrastructure over a decentralized topology by definition of any blockchain network. Our decentralized application will be running into all network nodes at the same time as seen in Figure 53. Any interaction can be done automatically between any peers and the network itself will give trust to the users thanks to the blockchain built-in cryptography.

OSINT and in general threat information exchange set-up have an asymmetric balance between consumers and producers: most of the entities or individuals are only consumers and only a few are producers. The incentive for producers is mainly the reputation and the image that their brand get from the community along the time.

Figure 50 Confederated information sharing topology a) mixed with either decentralized P-to-P b1), b2) or centralized topologies c1), c2) and c3)

Visik, C. et al. in [Vishik13], characterize the main reasons about why people is reluctant to share information if there are not enough incentives for them. Authors suggest that users do not have the same knowledge about the value of artifacts offered or simply they do not want to share their own data or incidents, as they could be giving a competitive advantage to their own competitors. Authors provide a clear understanding that introducing new economic incentives and business models in cybersecurity initiatives, as part of an economic modeling, would derive in greater societal benefits. As suggested, we propose new economic incentives based on the Ether crypto currency and the creation of a digital asset (token) which value will be associated to the use and interest over the information sharing service (our CTI Marketplace exchange).

Tosh, D. et al. in [Tosh18] propose a cybersecurity information exchange framework (CYBEX) based on a gaming exercise. Authors used an evolutionary analysis in order to derive conditional constraints on the concept of "participation cost", a driving parameter for attracting firms to join and transact cyber threat intelligence (CTI) with other firms. There are some interesting conclusions in the study like that a simple incentive, which is the reduction of the participating cost (negative cost vs positive cost), is enough to continue sharing along the time. On the other hand, when the cost is positive, the strategy of each organization depends on how many other members are contributing. If participation is less than 65 percent then organizations stop to share. As a conclusion, any external motivation associated with economic incentives will foster the participation. They used an evolutionary cost model using a gaming theory. We leverage from it towards something more appeal beyond discounts, we keep cost very low using a decentralized application (dApp) but at the same time we introduce incomes to CTI data producers as well as a CTI token which represent the digital asset of the information. Taking into account that the value of CTI Data nowadays is one of the most valuable information for the organizations (to better protect themselves), we consider that it has a promising value in the market.

Skopik et al. in [Skopik16] suggest that due to networks have grown to a scale and complexity, and have reached a degree of interconnectedness, their protection can often only be guaranteed and financed as shared efforts. Consequently, authors suggest that new paradigms are required for detecting contemporary attacks and mitigating their effects. Many attack detection tasks are performed by each individual organizations, and there is little cross-organizational information sharing. However, information sharing is a crucial step to acquiring a thorough understanding of large-scale cyber-attack situations, and is therefore seen as one of the key concepts to protect future networks. Authors also propose improvements in standardization and legal aspects as open challenges as well. Our proposal will address most of these topics.

Riesco et al. [Riesco19a] propose to work in risk domain dynamically to keep risks under an acceptance level along the time. In order to do that, authors propose to leverage cyber threat intelligence for a Dynamic Risk Management (DRM) framework. One of the building blocks, is the exchange of knowledge, which can be understood by semantic reasoners [W3C-Reasoner] and humans at the same time. The exchange proposed goes beyond exchanging indicators of compromise (IoC) in the format of intelligence algorithms using a combination of standards like STIX™ v2 [OasisSTIX2], formatted as OWL [W3C-OWL], ontologies [W3C-Ontology] and semantic web rule language (SWRL) [W3C-SWRL]. By using this combination of standards, authors are able to represent context behavioral patterns like the representation of techniques, tactics and procedures (TTP) [OasisSTIX-TTP]. Authors propose to exchange this type of algorithms instead of simple IoC in order to provide and share the real knowledge of detection. By using a mix of standards, authors can overcome the expressivity barriers to define TTP in the format of algorithms. It also enables the usage of semantic reasoners [W3C-Reasoner] to infer new knowledge. Authors propose to share TTP algorithms instead of IoC, as a new way to share more effective knowledge, less ephemeral and more anonymized. In that case, an organization receiving a TTP algorithm could make detection by itself, despite potential changes of the IoC used

by the attacker. By using ontologies (unambiguous understanding) any organization could reuse those algorithms to replicate the source capacity of detecting behavioral patterns. It enables a more efficient information sharing system than standard IoC sharing as an algorithm might include all possible combinations of its variables. Authors also propose that when an attack is being received by a victim, the Dynamic Risk Framework of the organization would update its risk level automatically by the use of a reasoner. They propose that new incentive models still have to be developed as a future research direction. A TTP rule sharing model (instead of sharing IoC) anonymizes the sharing of specific IoC. It can be understood as a way to improve the number of producers (especially for those entities or individuals which were still reluctant to share without anonymizing data, because an algorithm does not contain sensitive data), however it is probably not enough incentive yet.

We have then, implemented this specific approach in our work to demonstrate that if we combine this approach together with the blockchain, we would be able to contribute to most of the open challenges and limitations of existing solutions (as seen in Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions, and Table 2 Solutions provided by our proposal to all open challenges and limitations identified in Table 1.

In [Nath16] the authors propose the use of Blockchain for fighting insurance fraud. The authors propose the creation of a marketplace to share intelligence data about fraud of users, however it lacks of empirical deployment, incentive model as well as calculations related to cost. On the other hand it is handling intelligence which is different from cyber threat intelligence data or algorithms, which is our use case.

There is a very interesting approach in [Polyswarm] that is implementing a bug bounty program using a decentralized Marketplace. It is used to connect different experts with potential customers which are willing to pay for professional cybersecurity services. They are also creating a token, named Nectar, in order to use it for exchanging value. Customers can raise a kind of auction for any interested expert or send a direct contract to check if a specific file is malicious or not. In our case, we propose to go beyond IoC or bug bounties into threat intelligence and risk intelligence exchange of algorithms to really share knowledge providing capabilities to consumers. Their model is still based on IoC (hashed) and is specific oriented to antivirus. At the same time, we have a mix usage of CTI token and Ether as incentives as well as we have the ability to foster and incentivize the sharing of CTI Data rules either long (token value) or short term (cash in Ether). Different roles are created and new incentive ways by including CTI investors, donors and even CTI Data producers which can get special investment offers while sharing new CTI Data rules.

Another interesting example [Graf18], where Blockchain is being proposed as a cost effective storage of intelligence data, does provide storage and sharing features for the life-cycle state of incidents. Authors suggest a Blockchain-based solution for life-cycle management for automatic classification of cyber security documents according to their expected threat level. It is clear that blockchain provides several possibilities like we demonstrate in our work.

5.5. APPROACH AND RESULTS

Our approach is based on OWL Ontologies [W3C-OWL] and Semantic Web Rule Language (SWRL) [W3C-SWRL] as seen for example in Figure 65 and Figure 98 respectively. It provides a coherent and integrated solution to the needed expressiveness of such concepts, rules but also to inference [W3C-Inference] new knowledge. They could also resolve the lack of interoperability between existing risk management frameworks and methodologies under a common language and a common understanding in order to expand risk context into a global and more realistic picture.

Ontologies [W3C-Ontology] are an explicit and formal way to represent concepts, their meaning and their relationships. Explicit because it defines concepts as classes, properties, relationships, functions, taxonomies, axioms as well as rules and restrictions. It is formal because it is defined by a language, which is interpretable by machines. It is a conceptualization, because it is an abstract model for a simplified view of the domain to be represented (e.g., structure). It is also shared by the community for consensus.

By connecting both domains (Risk and Cyber Threat Intelligence) semantically, we demonstrate the ability to dynamically assess organizational risks based on Near Real Time (NRT) threat data.

We propose a model based on Layers (Data, Business Logic, Services / Applications and Visualization) as well as to integrate both domains and ontologies to approach both domains simultaneously.

In order to test our proposal, we selected a specific international recognized organization in the cybersecurity arena. Its current static Risk Assessment and Management (RA/RM) approach has been improved by our model into a Dynamic Risk framework which is successfully reacting and responding to real time threats.

We have simulated a well-known Watering hole attack from July-August 2017 targeting our selected organization (however it was not the target of the real attack last year) in order to test how our model provides with a more effective way to prevent, detect and respond against a Watering hole TTP (Techniques, Tactics and Procedures). Once the TTP is detected, security events are created. Those security events would trigger specific risk re-assessments dynamically but also specific risk management actions at either tactical or strategic levels, like the exchange of information between different partners.

Now that we are able to model any TTP with enough expressiveness, we would be able to model any TTP like those of [Alrwais16].

We also approached the issues of the Information Sharing presented in Table 1, based on Blockchain, as the Intelligence Sharing is a building block of our DRM Framework.

5.6. CONTRIBUTIONS

Here we describe specific contributions provided by the global architecture of the DRM Framework.

5.6.1 Architecture and Formal Model for CTI and Dynamic Risk Management

In this work we propose the use of semantic web ontologies, semantic web rule language (SWRL), semantic reasoners and an OWL version of STIX™ to solve most of the issues related to expressiveness, effectiveness, efficiency, anonymity, interoperability, automation and inference.

We propose an evolution of STIX™ v2.0 JSON [OasisSTIX2] into a semantic STIX™ 2.0 OWL (ontology) version and associated SWRL (semantic web rule language) rules as a formal model of a CTI Framework to solve above needs.

We also introduce a new semantic DRA/DRM OWL (ontology) and associated SWRL rules, leveraging CTI, as a formal model of a DRA/DRM Framework to solve all problems described above.

We implemented and used the international standard ISO2700X family [ISO/IEC27005:2008] as well as the ISO31000 [ISO/IEC31000:2018] to evaluate our proposal. Because there are different reference standard models, the creation of a formal model would also enable us to have interoperability between themselves.

Our semantic, formal model, integrates cyber threat intelligence and risk management domains. In addition to this, the architecture proposed will cover all above needs, of different domains, at the same time.

With regard to timing (NRT/RT), our data model will be working as a synchronized or dynamic (all in one) data model. Any change over CTI data will produce probable changes on the risk level of our organization, depending on the specific impact, relationship and properties of the data.

In addition to this, new knowledge can be inferred by the use of a semantic reasoner. It uses axioms in the format of SWRL rules (as seen in Figure 51 Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX v2). As an example, we will cover above needs related to enrichment, event detection, incident detection, prevention and incident triage.

We will focus our main contributions to provide:

- Dynamic risk assessment.
- Dynamic risk management.
- Dynamic information sharing.

Data: W3C semantic ontology: Classes, properties, variables and individuals. CTI Data will be under an OWL version of STIX™

Result: The SWRL (algorithm) is loaded as an axiom into the semantic reasoner. The reasoner will infer new knowledge over CTI Data (detect)

SWRL syntax **begin**

```
Antecedent(Body) → Consequent(Head);
/* When all conditions in the "Antecedent" are met, all conditions at the
"Consequent" are also met */;
Example;
hasparent(?x,?y) AND hasbrother(?y,?z) → hasuncle(?x,?z);
/* If your parents, have brothers; all of them will be your uncles */;
/* Variables are represented by "?" */;
```

Example in STIX™ **begin**

```
while (stix2:IPv4Addr = ?ip and stix2:belongsToRefs(?ip, ?ref) and
stix2:DomainName(?ref)) do
    stix2 : DomainName(?ref) and stix2 : resolvesToRefs(?ref, ?ip);
/* The reasoner will infer DNS matches from RDNS (Reverse DNS) */;
/* Note: We use "while" as pseudocode as the axiom is always active */;
```

Figure 51 Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX v2

In summary, we will provide several benefits to the state of the art, among them:

- The semantic expressiveness by using semantic ontologies. They enable better understanding of data to either humans or reasoners. It will enable the combination of high level and more complex algorithms. As an example, we can create algorithms with variables from different domains.
- Enhanced and more dynamic situational awareness. By combining STIX™ v2 objects with Risk Management concepts in the same rule, we would be able to share the rule as new Risk Intelligence knowledge. It will represent the knowledge of a dynamic risk framework (e.g. algorithms handling risk levels associated to real time CTI data). This will enable the needed expressiveness to have better situational awareness connected to near real time data. Reasoners will be able to act

dynamically as they would understand the data. More complex algorithms can be represented like TTP [Riesco19a].

- The interoperability and the automation will be provided by the reasoner and the use of ontologies. The reasoner will also provide consistency over the whole data set.

5.6.2 An Enhanced Exchange of Knowledge

In this section we will cover the rest of the open challenges seen in Table 1. We propose a solution to exchange rules, based on semantic web ontologies, semantic web language (SWRL), semantic reasoners and an OWL version of STIX™. It might contribute to solve most of the issues related to expressiveness, effectiveness, efficiency, anonymity, interoperability, automation and inference.

Figure 52 Proposal of a new model of "Entity within the DRM Framework"

For that, we propose that all entities need to be enhanced peers, working with the same ontologies, as seen in Figure 52. An enhanced entity has the following building blocks:

- **A semantic reasoner**, as the key element of all entities. Each reasoner of each entity will handle the inbound and outbound traffic as well as all internal transactions (within the same entity).
 - Inbound traffic: the reasoner will decide what CTI Data is interesting to retrieve. Then it will use it to infer new knowledge.
 - Outbound traffic: the reasoner will decide what CTI Data is interesting to share, after checking that the data is eligible (green or white TLP). It will what

data will be also interesting to be shared (because its interest but also because its potential value in the system).

- Internal transactions: the reasoner will handle all CTI and Risk Data together making new relationships to infer new knowledge dynamically.
- **A semantic data set**, which represents all the data known by each entity. Data coming from external sources like the CTI Marketplace will be used to infer new knowledge.
 - The CTI Data is always under semantic web format (OWL, SWRL). The (semantic) info shared must be stored within the Blockchain. Each entity will have their own CTI knowledge base (because of their knowledge and interest).
 - Risk data imports CTI Data. It enables the combination of both domains within the same SWRL rule, as an example to provide enriched knowledge.
 - The CTI Data can be IoC (as usual) but we propose to support algorithms in the format of SWRL (beyond IoC) as a much better solution (anonymization, less ephemeral knowledge transfer, etc.).
 - The Traffic Light Protocol (TLP), understood as the restrictions on data redistribution, can also be handled and used by the reasoner.
- **Dynamic Risk and CTI Intelligence module**: it is the advanced intelligence, the knowledge of analyzed and processed data beyond the data itself.
- **Risk intelligence**: risk advanced knowledge, in the format of algorithms, rules or even inferred data. As an example, a rule to define new investments needed or awareness training sessions depending on the dynamics of threats. That rule can be interesting for another entity to improve the automation of their information security management systems.
- **CTI intelligence**: CTI advanced knowledge, again, in the format of algorithms, rules or even inferred data (e.g. possible attribution of a cyber-attack to a threat actor). These types of reasoning axioms can be interesting to other peer.
- **Pseudo-automated decision making level**: thanks to the contribution of the reasoner, it can be useful to infer, classify and automate the knowledge for each of the decision making levels.
 - Strategic: e.g. recommended investments or partnerships, to reduce risks. This can be modeled in algorithms by using this enhanced version of the language.
 - Tactical: e.g. dynamic situational awareness and tactics (when to share, what to share, what must be under TLP, at what level, etc.). In this case, the reasoner can handle optimal investments decisions in the CTI token as another example.
 - Technical: e.g. detection and behavioral patterns, updating security policies (e.g. password length).
- **Role of each transaction**: All entities can (and probably will) work under different roles (CTI Data producer, CTI Data consumer, Investor, Donor); however only one can be the owner of each smart contract. The reasoner will also contribute to any decision to be taken related to each transaction (e.g. how much to invest, in case of an investment is recommended).

- **Wallet and connection to the blockchain:** All entities 1...N, are connected to the same blockchain through their own Wallets using either public or private Ethereum nodes.

As we propose to exchange algorithms, or rules, beyond IoC, some examples of SWRL algorithms can be seen in the pseudocode of Equation 3 and Equation 4.

Equation 3 Example of the pseudocode of a simple SWRL rule handling an OWL ontology

Data: W3C semantic OWL ontology: Classes, properties, variables and individuals.
Result: The SWRL (algorithm) is loaded as an axiom into the semantic reasoner. The reasoner will infer new knowledge over the OWL data

SWRL syntax begin

```

    Antecedent(Body)  $\rightarrow$  Consequent(Head);
    /* When all conditions in the "Antecedent" are met, all conditions at the
       "Consequent" are also met */;
```

Example begin

```

    hasparent(?x,?y) AND hasbrother(?y,?z)  $\rightarrow$  hasuncle(?x,?z);
    // If your parents, have brothers; all of them will be your uncles;
    // Variables are represented by "?";
```

Example in pseudocode begin

```

    while (ont1:hasparent(?x,?y) and ont1:hasbrother(?y,?z)) do
        ont1:hasuncle(?x,?z);
        /* The reasoner will infer all "uncles" of all "people" in the data
           set */;
```

/* Note: "ont1" is the prefix of the ontology used. Different ontologies can be used within the same SWRL rule */;

```

    // Note: We use "while" as pseudocode, as the axiom is always active;
```

Equation 4 Example of the pseudocode of a simple CTI rule / algorithm in the format of SWRL and an OWL version of STIX™ v2

Data: CTI Data under an OWL version of STIX™
Result: The reasoner will infer new knowledge over CTI Data

OWL STIX™ example begin

```

    stix2:IPv4Addr=?ip AND stix2:belongsToRefs(?ip,?ref) AND
    stix2:DomainName(?ref)  $\rightarrow$  stix2:DomainName(?ref) AND
    stix2:resolvesToRefs(?ref,?ip);
```

OWL STIX™ example in pseudocode begin

```

    while (stix2:IPv4Addr=?ip and stix2:belongsToRefs(?ip,?ref) and
    stix2:DomainName(?ref)) do
        stix2:DomainName(?ref) and stix2:resolvesToRefs(?ref,?ip);
        /* The reasoner will infer all "DNS matches" from "RDNS (Reverse DNS)"
           in the data set */;
```

/* Note: We use "while" as pseudocode as the axiom is always active;

In order to demonstrate the potential of this model, we provide a piece of a more complex CTI algorithm, which is using an OWL/XML syntax. "Stix2" prefix, seen in "abbreviatedIRI", is the OWL version of the current STIX™ v2 object. "drm" prefix belongs to a different ontology related to dynamic risk management, which is interoperable within the stix2 ontology.

Equation 5 Example of a CTI rule or algorithm in the format of SWRL and STIX™ v2 to define a TTP detection

```

<DLSafeRule>
  <Annotation> <AnnotationProperty abbreviatedIRI="rdfs:label"/>
    <Literal>ThreatIntelligence#1 Security Event Dropper Detection</Literal>
  </Annotation>
  <Body>
    <ClassAtom>
      <Class abbreviatedIRI="stix2:NetworkTraffic"/>
      <Variable IRI="urn:swrl:var#nt"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty abbreviatedIRI="stix2:dstPayloadRef"/>
      <Variable IRI="urn:swrl:var#nt"/>
      <Variable IRI="urn:swrl:var#pl"/>
    </ObjectPropertyAtom>
    <ClassAtom>
      <Class abbreviatedIRI="stix2:Artifact"/>
      <Variable IRI="urn:swrl:var#pl"/>
    </ClassAtom>
    [<DataPropertyAtom>
      <DataProperty abbreviatedIRI="stix2:mimeType"/>
      <Variable IRI="urn:swrl:var#pl"/>
      <Literal>javascript</Literal>
    </DataPropertyAtom>
    [...]
    <ClassAtom>
      <Class abbreviatedIRI="stix2:URL"/>
      <Variable IRI="urn:swrl:var#red"/>
    </ClassAtom>
    [...]
    <DataPropertyAtom>
      <DataProperty abbreviatedIRI="stix2:extensions"/>
      <Variable IRI="urn:swrl:var#pl2"/>
      <Literal>windows-pebinary-ext</Literal>
    </DataPropertyAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class abbreviatedIRI="drm:SecurityEvents"/>
      <Variable IRI="urn:swrl:var#x"/>
    </ClassAtom>
    [...]
    <Literal>Dropper behavior of Malicious Windows Executable</Literal>
  </DataPropertyAtom>
</Head>
</DLSafeRule>

```

5.6.3 A Pure Decentralized Infrastructure Based On Blockchain for Information Exchange

As seen in Figure 49 and Figure 50, there are different types of topologies used nowadays, however, none of them are based on a pure decentralized infrastructure as we understand. All instances or platforms are managed by a centralized or unique entity at the end. We understand that authors in [Lesczyna19], are really talking about a peer-to-peer interaction between different user's instances, instead of a pure decentralized platform. Suggested platforms by authors are not decentralized, at least in a pure way. Their users are using either a specific centralized platform (provided by a third party) or their own

platforms. Trust is needed between peers as it is not provided by the network. Confederated instances' administrators belong to specific entities. Administrators have then the possibility to change the behavior of the system without the user consent anytime. Trust, as seen in Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions, is one of the main pain points that we are addressing in our work.

Figure 53 Proposal of P-to-P pure decentralized Intelligence Sharing Network based on blockchain

We propose a new model, based on a pure decentralized architecture, based on the Blockchain Decentralized Network. As seen in Figure 53, the model also includes an enhanced version of the participants or peers (*Entity 1..N*), as they are using semantic web ontologies combined with STIX™. The administrator concept does not longer exist as it will be a decentralized application (dApp).

The main components of each entity can be seen in Figure 52. Each entity will have a reasoner to handle its own data, reasoning about when to share, what can be shared, which investment can be done. At the same time, it will work the other way around, that is, what information is useless to spend cash (or tokens) on it. In the next subsection we will give more details about the different building blocks of the enhanced entities.

The new architecture of our model is addressing most of the problems seen in Table 1. It has some specificities, among them:

- **Trust** is provided by the network itself:
- **Integrity:** Once the application is deployed, it runs autonomously without any admin or human intervention in all the nodes (nodes run EVM: Ethereum Virtual Machines).
- **Validations:** all transactions are validated within the network using cryptographic algorithms.
- **Immutable register:** There is a transparent and immutable register. The network use Merkle trees and other advanced cryptographic features to protect it.
- **Code transparency:** The application does what it has been coded. The running code (ABI) can be checked within the source code. High level of transparency is then provided.

- **Availability:** is provided by the nodes. Nodes are running by network incentives. All transactions within the blockchain have economic incentives for nodes. There is no need to run an owned infrastructure so there are not ongoing costs.
- **PKI:** users are using wallets which represent the public keys in a PKI infrastructure. Private keys will then be needed to execute any transactions and only those private keys will be allowed to make transactions with those wallets.
- **Time-stamping** of each transaction. In CTI, one of the issues is related to the reliability of the information [OSINTArmy12]. All transactions will be time-stamped, within the immutable register of Blockchain, to keep track of who shared what and when.
- We will create a **token**, which is a crypto-feature which inherits all above characteristics to provide trust.
- There can be multiple decentralized **applications interacting between themselves**, all of them will be running in the network (in all of the nodes).
- The **interoperability** between decentralized applications (which is inherent to the system), can be understood as a confederated topology where different CTI Marketplaces can coexist and interact between themselves.
- **There can be malicious nodes** as noted in the figure. For simplicity, we do not consider those which try to create their own blockchain. We will consider, as malicious, those nodes that will try to get as much profitability from the system but also from the rest of users as possible.
- It has **very low cost** (just gas), as there is no infrastructure needed to deploy our information sharing application. Costs will be associated to transactions fees.
- **Investments and payments are proposed as economic incentives.** As described by authors in [Tosh18], just a simple reduction of the fee, was enough to provide incentives to users. In our case, they will have incomes, the more they share (with quality), the more it will be consumed. Each time the CTI Data shared is consumed, the producer of that data will receive incentives. There is also a new opportunity for non-cybersecurity experts, to invest in the CTI token, as it becomes a trading opportunity within the cybersecurity arena.

In addition to this, the use of Ethereum Blockchain, will allow us to provide economic incentives to better incentivize the users. It might help us to improve the dynamics of the information sharing as well.

We will use the built-in feature to create "tokens", by creating the first CTI token as a digital asset to represent de CTI Data, data which is considered very valuable nowadays. We will use the ERC20 [ERC20] standard to enable the interoperability between tokens. We could then be exchanging our CTI tokens with any other digital asset someday (e.g. gold or any other stock options).

5.7. CORE COMPONENTS

This section will define the general architecture where the reasoning has a key role, as seen in Figure 54.

Figure 54. General outline of the DRM Framework Architecture

5.7.1 (Dynamic) CTI and Risk Data-Set

We propose that all data has to be modeled with our ontologies. We will use a semantic (OWL) version of STIX™ for CTI Data and a new ontology for Risk data. Both are part of our contribution. We propose sensors to also work with those ontologies, especially with the CTI ontology. The same with the knowledge base. Dynamic Risk Ontology (DRM.owl) will import STIX2.owl ontology to have a hybrid, and an interoperable ontology. It will enhance the establishments of all type of relationships between both domains. New hybrid SWRL rules will be possible.

Figure 55 Dynamic Risk and CTI Data-set

5.7.2 (Dynamic) Risk and CTI Intelligence Knowledge

At the same time, if applying the reasoner to the CTI and Risk data, we will get new inferred knowledge, which is the advanced intelligence. There will be algorithms, as axioms, in SWRL, which will provide knowledge to the reasoner. There will be also relationships between classes (e.g. inverse, disjoint, etc.) that will help the reasoner to infer new knowledge either for CTI or Risk domains. We propose that reasoning layer is part of the business logic layer. It might help to decision making and even information sharing modules. In our thesis, we also propose the use of the reasoner to classify specific data to be share but also to decide the role of the new model for information sharing (CTI producer, consumer, investor, etc.)

5.7.3 Dynamic Risk Management

This module is based on 2 modules:

- Dynamic Risk Assessment (DRA)
- Dynamic Risk Management (DRM)

Figure 56 Dynamic Risk Management: DRA and DRM related modules

5.7.3.1. Dynamic Risk Assessment (DRA)

The proposed dynamic risk analysis system or Dynamic Risk Assessment (DRA), is subdivided into various subsystems and processes such as:

- **"Dynamic Risk Inference"**: responsible for the inference of new risk intelligence. It is also responsible for calculating the context of the organization including the discovery of assets and dependencies through a dynamic and iterative approach. It is therefore based on a process of continuous semantic reasoning based on the input of new information from the previous process or subsystem. At the same time, it is responsible for calculating and re-calculating the value of the assets following the variables I (integrity), C (confidentiality), A (authenticity), T (traceability), D (availability), the latter also being de-logged in RTO (maximum time for service interruption), RPO (target time for service recovery). It can also identify new risks in any dimension (Operational (Ops), Tactical, Strategic, Legal, Compliance (compliance), etc.) automatically.
- **"Dynamic Risk Assessment"**: responsible for the risk analysis of our organization. It dynamically reconsiders the input, like the inferred risk intelligence that requires continuous monitoring. To make the assessments, it considers the safeguards and controls that are already part of the organization. The safeguards or countermeasures (denominated C1... CN) would modify the levels of impact or probability. The system will also consider the security events (coming from CTI data, either internal or external through intelligence sharing) to calculate increments of impact or probability variables (associated to that specific context). As a result, the residual risk will be re-calculated dynamically in case of changes of our context.

Equation 6 Simple Risk Equation depending on impact and probability

$$R = I + P - C1 + E1$$

where:

- R = represents the risk.
- I = represents the impact.
- P = represents the probability of occurrence.

- C1 = represents the safeguards or countermeasures.
- E1 = represents the security events.

The probability of occurrence P , being a variable of high importance, is usually defined by expert consultants using an overly simplistic and subjective approach, lacking a scientific basis based on near real time data or even historical data about the entity in question so it is useless. For this reason, the present doctoral thesis tries to evolve from the scientific point of view, the definition of this variable based not only on expert opinion but also on knowledge about the organization real context since a generalist projection is not considered sufficient for a variable which depend on different decision making as for example in the priority when allocating budget, investments and resources.

5.7.3.2. Dynamic Risk Management (DRM)

The proposed dynamic risk management and treatment analysis system (DRM is subdivided into different subsystems and processes, such as:

- **Inference of Optimal Countermeasure Actions or "Dynamic Risk Countermeasure Action Inference"**: responsible for semantically inferring the optimal response in terms of countermeasures to mitigate risks. The answers can be of different types, ranging from a new Training to specific users, launching an information sharing information, setting up new signatures of IDS (Intrusion Detection System), etc. to the change of security policies (e.g. improve complexity of specific user passwords or activating a 2nd factor).

Depending on the value resulting from the risk:

- Extreme (8-10): the risk is **unacceptable**.
- High (6-7): the risk is **undesirable**.
- Medium (5): the risk is **tolerable**.
- Low (2-3-4): the risk is **acceptable**.
- **"Dynamic Risk Treatment Strategy Inference"**: responsible for semantically inferring the optimal strategy for the treatment of risks. Depending on the value of the subsystem or process mentioned above, the recommended strategies may be one of the following options:
 - **Mitigation strategy**: recommended for cases where the risk is High or Extreme. It has several possibilities:
 - Reduce the risk.
 - Avoid the risk.
 - Share the risk
 - Transfer the risk.
 - **Research strategy**: recommended for cases where medium level risk. It has several possibilities:
 - Take the risk.
 - Reduce the risk.
 - **Monitoring strategy**: recommended for cases where the risk is Low level. In this case normally the risks will be assumed.

Likewise, the strategies must take into account the potential actions to be carried out in their different dimensions and impacts:

- Image and reputation.
- Fulfillment.
- Security.
- Cost and budget.
- Operation.

In addition to the standard strategies discussed above, there is another type of "Special Actions" such as:

- Attack.
- Deception.
- Deterrent.
- Share information (information sharing).
- Awareness (awareness)
- Etc.

The execution of a specific strategy could impact other variables or even generate cascading effects, which is why, in this case, we recommend the use of a simulation and visual or graphical system. In this way, the simultaneous use of semantic inference with simulation or forecasting systems will allow us to know in advance what type of actions may influence, to a greater or lesser extent, an improvement and therefore a lower level of exposure to risk in our organization. Our DRM Framework will react when certain data is changed or added, so we can test or simulate future projections to check dependencies or benefits. The reasoner will help to understand specific reactions.

- **"Dynamic Risk Management Response"**: responsible for executing totally or partially the strategies selected in the previous subsystem. The use of semantic technologies will allow us to execute pseudo-automated responses.

Some examples:

- Change of a security policy
- Re-prioritization of SW patch "queues".
- Transfer of risk to a third party or "partner".
- New rules or signatures (IOC) for network devices.
- Sharing of risk information.
- Alert notification.
- Launching a decoy or counter-intelligence measures.
- Coordinated and distributed response from the sensor network.
- Scaled to the address on the level of exposure to risk when it exceeds a threshold by attaching visualization system to support the decision.

5.7.3.3. Feasibility of a New Dynamic Risk Equation

By using our framework, we propose an improvement of common risk equations to leverage Risk Assessment into something more dynamic (depending on time t and security events) and affordable with enough flexibility and granularity like:

Equation 7 New dynamic risk equation

$$R_{i,t} = P_{i,t} + I_{i,t} - \Sigma(C1_{i,t}, C2_{i,t}, \dots, CN_{i,t})$$

$$R_t = \Sigma R_{i,t}$$

being:

$$P_{i,t} = P_{i,t-1} + \Sigma(EP1_{i,t}, EP2_{i,t}, \dots, EPN_{i,t})$$

$$I_{i,t} = I_{i,t-1} + \Sigma(EI1_{i,t}, EI2_{i,t}, \dots, EIN_{i,t})$$

where:

- $P_{i,t}$ is the probability of threat i at time t ,
- $I_{i,t}$ is the impact of threat i at time t ,
- $CN_{i,t}$ is the decreasing value of the Impact (Severity) I or Probability P of threat i at time t due to Countermeasure N ,
- $EPN_{i,t}$ is the increasing value of the Probability P of threat i at time t due to the Security Event N ,
- $EIN_{i,t}$ is the increasing value of the Impact (Severity) I of threat i at time t due to the Security Event N ,
- $R_{i,t}$ is the Residual Risk associated to threat i at time t ,
- $\Sigma R_{i,t}$ is the sum of all type of Residual Risks associated to threats $i=1 \dots Z$, at time t ,
- R_t is the Total Residual Risk at time t of all type of threats $i=1 \dots Z$,

5.7.4 Dynamic Risk and CTI Intelligence Information Sharing Module

It is responsible for processing dynamic data and algorithms received from any sensor (external or internal) that provides or exchanges information (eg, threats, risks, incident management, change management processes, audit results, rules and algorithms, etc.). It also processes information from third parties, both partners or business partners and suppliers that are part of our supply chain. The type of information is diverse and heterogeneous, ranging from information about cyber threats to legal compliance information through risk information in rules format, based on SWRL, which will formally define behaviors such as those that allow certain detections. All the information will be valuable to be able to have a greater knowledge about the context and thanks to it, of our level of risk. The data on threats and risks will not only correspond to current data but also work on a database (BBDD) of knowledge with historical information. The information must be compatible and interoperable with de facto standards such as STIX™ and TAXII™ that are used extensively to structure and exchange information about threats. The ontologies of the present doctoral thesis will be the basis for the normalization of the data and to provide unambiguous meaning as well as to define the relationships between the datasets. Finally, rules based on SWRL will formally define behavior and knowledge in intelligence about cyber-threats. We propose the use of Blockchain and new incentive models to overcome the issues shown in Table 1 at least to count with an optimal intelligence data contribution for our DRM Framework.

For more information of the Information Exchange proposal, see CHAPTER 8.

5.7.5 Reasoner

The reasoner is one of the main parts of the whole architecture as it provides data consistency but also knowledge inference, between other benefits, at all levels (technical, tactical and strategic).

It will work with axioms in the format of SWRL rules to also represent behaviors.

Figure 57 The reasoner: a key building block of the whole architecture

For more information of the (Pellet) reasoner, see 4.4.2 and CHAPTER 7.

5.7.6 Decision Making Level

5.7.6.1. Technical

Decisions at technical level are usually near real time (NRT) but also taken by Technical staff. They are handling incidents or security events but those are usually disconnected from the ISMS (Information Security Managed Systems) of each organization. As a result, Situational Awareness applications are not using near real time data. At the same time, several risk investment decisions are not considering the level of real threat exposure of our organization. They are usually considering subjective expert theories which become ineffective. Our proposal integrates both domains to have NRT Risk Management (DRM).

Figure 58 Decision Making level

5.7.6.2. Tactical

At Tactical level, we can now implement Tactical decisions like Information Sharing of related peers, the delivery of algorithms beyond of IOC, or event investment decisions in new type of information sharing initiatives, like the one presented in this thesis. The decisions could be automatically triggered by the input of CTI reliable data dynamically.

5.7.6.3. Strategic

Strategic decisions will now be taken with as much relevant (and NRT) information as possible, and decisions or recommendations can be taken automatically by the reasoner. The trigger will also be the connection to NRT data coming from CTI dynamically. There will be graphical (semantic graphs) plugins to foster the representation of the data along the time as a NRT Situational Awareness approach. There are also opportunities for the reasoner, to implement a first level of decision making learning from axioms provided as knowledge experience. This knowledge can also be uploaded by non-technical experts in the format of SWRL which are not so complex, by definition.

5.8. EXTERNAL COMPONENTS

5.8.1 N-Dimensional Sensoring System

The sensors themselves, although their development is not part of the present doctoral thesis, are mentioned at the enunciative level with the aim of familiarizing the reader with the type of sensors or characteristics they should have:

- **Internal sensors:** are those sensors deployed by the organization in both internal networks and external networks (e.g. VPS) but always under the control of the organization's administration.
 - Advantages:

- More detailed information about the events.
- Greater reliability of the information, since it has access to the logs directly (e.g. Timestamp).
- Greater flexibility and adaptability, since it would allow, for example, to provide them with greater intelligence by expanding their business logic.
- Greater integration capacity (for example using the combination of several different sensors) or the use of standards.
- Possibility of having information in real time.
- Disadvantages:
 - Cost of Operation and maintenance.
 - Need to have resources trained in a wide variety of collection and deployment (IT) technologies in addition to the analysts themselves.
 - Need to have, in your case (e.g. traffic), with your own networks (e.g. the style of ISP).
- **External sensors or "feeders":** those sensors deployed by partners or suppliers and therefore their control and administration will be carried out by third parties. They will commonly be called "feeders" and the information they will share will be limited. They can have cost or be free (OSINT or through agreements / collaboration agreements). In our architecture, we propose a new model based on blockchain to provide new incentives. We will implement the Information Sharing dApp but not the sensors.
 - Advantages:
 - It does not require having resources trained for Operation, Maintenance in a wide variety of collection and deployment (IT) technologies.
 - It does not require having, in your case (e.g. traffic), with your own networks (e.g. the ISP style).
 - Disadvantages:
 - Price (in the case of non-OSINT sources or through collaboration agreements)
 - Limited information and usually with different or without context depending on the feed on the events.
 - Less reliable information, since not having access to logs directly (e.g. Timestamp) and not having detailed information, the interpretation is difficult to perform for use from actionable or pseudo-automated intelligence systems.
 - Less flexibility and adaptability, since for example to provide them with greater intelligence by extending their business logic, it must be the decision of the supplier or partner, which is not guaranteed in almost any case.
 - Reduced integration capacity (for example trying to use the combination of several sensors from several different providers or mixing internal and external) if there is no API or, if applicable, the data is not served in real time or NRT (near real time) or for example,

the type of threat information does not follow de facto standards (e.g. STIX™).

- Etc.

Likewise, the sensors can belong to different types of dimensions due to their nature (sensors of natural origin, physical sensors (e.g. presence, access control, etc.), vulnerability sensors, sentiment sensors applied to personnel (HR), monitoring sensors on user devices, sensors on perimeter network devices, etc.)

A key issue in intelligence models is that the data on which the algorithms usually work are ephemeral data, of greater volatility, less significance and they usually lack of context. For example, IOCs (indicators of commitment) are, for example, "hashes" with information on the footprint or signature of a specific device (e.g. malware) or a specific IP address. If a simple modification of a binary character implies that its "hash" or footprint / signature can be modified by almost 50%, it is not very effective to fight against threats that can so easily evade our detection systems.

The same happens with IP addresses, even if the change takes the attacker a little more time, this will be little more and again he will be able to evade our systems of detection and protection against threats.

In this way, the concept of "pyramid of pain" arises (see section 3.3), which proposes that our indicators and therefore our intelligence data about cyber threats, can be used to model the concept of TTP (Tactics, Techniques and Procedures), that is, the techniques, tactics and procedures of the attackers. In this way, for example, we will have information on what exploit they use, what combination of infrastructure to disturb the malware and what techniques (e.g. spear phishing) they are using, and once a TTP is known and modeled, our detection systems will be better prepared, they will be more robust and above all, it will make the attacker who wants to continue in his endeavor much more difficult, since he will force him to modify his modus operandi, a matter that is not trivial (e.g., they do not always have access to exploits, alternative Oday vulnerabilities or the possibility of change your techniques or attack vector).

Theoretically, the intelligence community on cyber-threats agrees with this approach and in some forums it is commonly known as "pyramid of pain" to change the way in which the indicators of commitment are worked to be really effective.

The present doctoral thesis, proposes a practical solution to this theoretical approach, since until now there are no possibilities of having a semantic representation of this knowledge and their formal relationship between the different concepts associated with a given TTP.

This approach is one of the original contributions of the present doctoral thesis, we will use ontologies and rule-based languages such as SWRL for the formal modeling of TTPs, making detection systems much more effective and practical, while at the same time complicating even more to the attackers who want to insist on their efforts once we have modeled their TTP in its different variants. The different variants, if they want to be effective, must be modified to a great extent, which is quite complex and will make things much more difficult for the attackers.

Figure 59 Pyramid of pain [Bianco14]

As an example, two examples for the "tools" levels are discussed (tools) and TTP:

- **Tools:** the attacker is forced to change or rewrite, re-invent new methods to do the same in another way that is effective again (e.g. Yara Rules)
- **TTPs:** the attacker is forced to change completely the method - you cannot do the same and you have to spend not at all despicable time if you want to insist on that attack (Options: Abandon or start from scratch) (ex: attack vector based on spear-phishing with malicious attachment that exploits a vulnerability, exploits / vulnerability used, methods of extracting cached passwords, etc.)

One characteristic that all these sensors must have to gain in efficiency and interoperability, will be that of having standard data models about threats. In the present doctoral thesis the use of working with the STIX™ standard which has been adapted to an ontology in its version 2.0 is proposed.

For the use of this doctoral thesis, therefore, the data models of the different sensors should be adapted to this standard format, for which, at times, it will be necessary to parse the data.

5.8.2 Knowledge Database

The knowledge database will allow greater precision in the calculation of the probability of occurrence of certain threats, a variable that directly influences the calculation of risk and, therefore, in the decision making about the realization of certain investments for the mitigation of certain risks.

The present doctoral thesis therefore proposes to evolve the judgment of the expert in the assignment of said variable to a system with greater scientific rigor and that is based on historical and NRT CTI Data.

The BBDD will allow adjusting the expert value judgment about the probability of occurrence of a given threat based on historic data.

5.8.3 Visualization and Simulation Engine (Business Intelligence)

Although it is not the aim of this doctoral thesis to emphasize that interactive visualization, based on the language of the semantic web, would allow greater knowledge and expressiveness about the possible impacts that certain measures would have on our organization (e.g.. the application of new controls).

The possibility of simulating certain security measures and policies in the future, supported by ontologies, would avoid problems (e.g. unexpected impacts by having control over the degree of dependence on the assets) as well as help in the decision-making processes on the more effective measures to achieve our objectives.

When decisions involve investments, as we see in the case of risk analysis and management systems, being able to count on simulation environments is considered a competitive advantage.

Also, the fact that our governing bodies know firsthand and in real time the level of risk exposure of our organization and this is calculated dynamically, will deliver value to all layers of the organization (operational, tactical and strategic level) in the terms that are expected and understandable.

The use of ontologies, SWRL language and semantic reasoners, will nevertheless allow us to unequivocally know the relationships, the meaning of the inferences, and the enrichment of the information, as well as the navigability through graphs on the set of data, making it even easier (an example is seen in Figure 65).

5.8.4 Anonymization, Authentication and Authorization System

In order to comply with not only legal measures (e.g. GDPR / LOPD) but also comply with confidentiality agreements (NDA), specific anonymization filters will be applied to certain information before being shared with third parties.

The TLP (Traffic Light Protocol) code will be used because it is widely used internationally.

Likewise, the principle of rationalizing information to be shared will be used, not sharing information that is not strictly necessary.

For this, although not objective of the present doctoral thesis, security controls on data based on policies and attributes will be applied. In this way you can limit access to certain information (e.g. a column) within the same table if you do not have the appropriate level of authorization.

For the calculation of the probability of occurrence of a certain threat, as mentioned above, we will need certain information about the materialized threats (incidents), although it is not relevant to know the entity or detailed information about the threat (even if they are). We only need to know if the entity is of the same type as our entity and the type of classification of the threat, thus being able to anonymize the rest of the information received from the sources.

Finally, and with the objective of anonymizing potential sensitive data, the use of STIX™, in its OWL ontology version (contribution of the present doctoral thesis), will allow the anonymization by using Data Markings.

5.9. INTEGRATED AND LAYERED ARCHITECTURE

Here we introduce our proposal as an integrated and layered architecture:

5.9.1 Semantic Data Model

Here we created two main OWL ontologies (OWL - Web Ontology Language) for the two domains:

- STIX2.owl for all Threat Intelligence Data in STIX™ v2.0 [OasisSTIX2] format,

- DRM.owl for Dynamic Risk Data.

Both are connected as DRM.owl directly imports and extends STIX2.owl. OWL DL (Description Logic) is designed to provide the maximum expressiveness as possible while retaining computational completeness (either Φ or $\neg\Phi$ holds), decidability (there is an effective procedure to determine whether Φ is derivable or not), and the availability of practical reasoning algorithms.

STI2.owl is a contribution to help STIX™ v2.0 [OasisSTIX2] to solve their current problems with regard to the representation of more complex concepts like TTP (Techniques, Tactics and Procedures) [OasisSTIX-TTP], Campaigns [OasisSTIX-Campaign] or Incidents [OasisSTIX-Incidents] but also to make automation easier by using semantic reasoners. We followed all OASIS open standard specification of its version 2.0 translating the whole standard, that is to say, all requirements and restrictions from the STIX™ specifications are now ontology Classes, Property Objects, Datatypes and Axioms in OWL format.

DRM.owl imports STIX2.owl leveraging CTI into a more comprehensive and meaningful DRA/DRM architecture. Taxonomies and domains are fully integrated: concepts are related between themselves, relationships are formally established between all types in order to help us to solve initial challenges but also providing us the capability to use a reasoner (this time we used Pellet incremental reasoner) [W3C-Reasoner]. Datatypes are formally defined as well.

We then have a graph and meaningful data model to connect a specific IOC, threat or security event to an asset. We also have the possibility to calculate associated risks around each service, based on dependencies, as seen in Figure 65.

The framework presented is a formal model which enables the representation of any CTI or DRM context despite its complexity and previous limitations like the ones still not solved in STIX™ XML and JSON format [OasisSTIX-TTP] [OasisSTIX-Incidents] [OasisSTIX-Campaign] evolving threat intelligence Data until today into a real TTP meaningful patterns and representation (graphs beyond connected IOC).

Due to our contribution of a complete DRM framework leveraging a complete version of STIX™ v2.0 ontology, any STIX™ related data can be parsed into our ontology.

In our work, all the relevant data for our use case at all levels (operational, tactic, strategic) were parsed into our DRM ontology as a central data storage. All SWRL rules handle or evaluate these data. On the other hand, we limited to STIX™ the different types of data to be collected as future sensors will probably provide data in a format compatible with this taxonomy. We have then focused our contribution to the processing of data but not about its collection.

5.9.2 Semantic Business Logic (Reasoning) Model

Now that all Data (threat intelligence and risk data) are based on OWL and relationships are formally established, we propose to use SWRL (Semantic Web Rule Language) which will enable us to express rules as well as logic combining OWL DL, OWL Lite and RuleML (Rule Markup Language).

Rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

We will use SWRL to create semantic algorithms and rules that will use semantic data to provide value added and to cover all use cases defined above, like:

- **Enrichment of Threat Information Context**, for example by using the SWRL rule of Figure 60 we can represent a simple DNS Data enrichment to make relationships between IP and Domain concepts. By knowing an IP address belonging to a DomainName, we create a reverse relationship, that is, the DomainName to IP is the "inverseOf" IP to DomainName:

Name
Enrichment#1 Auto IPv4 assignment by Domain match (DNS)
Comment
Status
Ok
stix2:IPv4Addr(?ip) ^ stix2:belongsToRefs(?ip, ?ref) ^ stix2:DomainName(?ref) -> stix2:DomainName(?ref) ^ stix2:resolvesToRefs(?ref, ?ip)

Figure 60 SWRL DNS Enrichment

The rule has an antecedent (body) indicating that all IP addresses in IPv4 format belonging to a specific Domain name, will then create an inverse relationship in the consequent (head), that is to say, the same Domain names will then resolve back to those IP addresses. This is a DNS versus Reverse DNS behavior.

Anyway, SWRL rules are quite easy to understand and to create by non-experience users. A 4 h briefing about SWRL and the namespace domains used (e.g., OWL ontologies: classes, properties and datatypes of both domains) should be enough to start creating business oriented SWRL rules. Until today, complex rules had to be created by programmers with high SW programming skills [Zhang17]. We propose SWRL rules to solve this problem and to enable any person (operator (Operational), manager (Tactical) or director (Strategic)) to create its own rules.

There are also other types of rules like:

- **Enrichment of Risk Assessment and Risk Management Context.**
- **Automation of Risk Management** like the automation of Risk Level Classification, as a previous step to decision making.
- **Etc.**

For more information, several types of business logic rules based on SWRL can be seen in CHAPTER 7 and ANNEX D.

5.9.3 Semantic Services / Application Model

Here we provide services and applications for both domains. We can also give answer to more complex scenarios and use cases in the Services / Application Layer by using SWRL expressivity as well.

As an example, we propose a rule for **CTI Security Event Detection (see section D.IX.I)**. By using a more sophisticated SWRL rule, we can represent a security event detection based on a TTP (which now can be represented without semantic limitations [OasisSTIX1] [OasisSTIX-TTP] [OasisSTIX-Campaign] [OasisSTIX-Incidents]). Its business logic is based on a specific network traffic pattern. The pattern, once it matches, will increase dynamically the risk level of those risks of type Deliberated Malicious SW Distribution (as seen in section D.V.III).

There are also other types of rules like:

- **Security Incident Detection** and Prevention.
- **Incident Triage by Severity.**
- **Information Sharing Application.**
- **Risk management action at Tactical level.**
- **Risk management action at Strategic level.**

Because our data, business logic and services / applications are based on semantic OWL data, SWRL and reasoners; our **semantic visualization model** could benefit from enriched semantic graphs. We have different options by using Protégé tool created by Stanford.

- OntoGraph plugin to render interactive graphs of our integrated formal model (CTI + DRA/DRM Classes, Properties and Datatypes). It also includes individual instances (threat intelligence and risk related data) as shown in Figure 65. Equivalent plugins could be VOWL and OntoViz.
- SQWRL query language to query our data model or instances. An example of query is shown in Figure 61 and results of that query are shown in Figure 83. When consequent belongs to a query, "sqwrl:select" is used.

Name
Dynamic Risk ReAssessment after a Security Event of type Dropper by a less experienced User
Comment
The case of Droppers behaviour when end user is unexperienced the probability of risk/damage increases
Status
Ok
<code>drm:SecurityEvents(?se) ^ stix2:type(?se, "Dropper behaviour of Malicious Windows Executable") ^ stix2:srcRef(?se, ?i) ^ drm:PersonalComputer(?pc) ^ stix2:hasCyberObservable(?pc, ?i) ^ stix2:isUsedBy(?pc, ?u) ^ drm:Users(?u) ^ drm:hasAccessTo(?u, ?data) ^ drm:Data(?data) ^ drm:ResidualRisk(?rr) ^ drm:threatens(?rr, ?data) ^ drm:evaluates(?rr, ?ev) ^ drm:Risk(?ev) ^ drm:damages(?ev, drm:AdobeFlashPlayerPlugin) ^ stix2:type(?rr, "Deliberated Malicious SW Distribution Risk") ^ drm:probability(?rr, ?p) ^ drm:potentialRisk(?rr, ?pr) ^ drm:actualRisk(?rr, ?ar) ^ swrlb:add(?newp, ?p, 1) ^ swrlb:add(?newpr, ?pr, 1) ^ swrlb:add(?newar, ?ar, 1) ^ stix2:isIncreasedBy(?rr, ?se) -> sqwrl:select(?u, ?rr, ?p, ?newp, ?pr, ?newpr, ?ar, ?newar)</code>

Figure 61 SQWRL query example

For more details, several types of Services / Applications rules based on SWRL can be seen in CHAPTER 7 and ANNEX D.

5.10. CONCLUSIONS

In this chapter, the Global Architecture of the DRM Framework has been presented as the main contribution of the thesis. All related work is analyzed to better focus the proposal.

Its associated contributions were also introduced:

- The (new layered and networked) architecture and the formal model for CTI and DRM (see 5.6.1).
- An enhanced exchange of knowledge (intelligence) (see 5.6.2).
- A pure decentralized infrastructure for Intelligence sharing based on Blockchain (see 5.6.3).

In addition to this, the whole DRM Framework (seen in Figure 54) is described in details for each of the modules or components:

- Core components (see section 5.7).
- External components (see section 5.8).

Last but not least, the layered and integrated architecture is described in details:

- Semantic Data layer (see section 5.9.1).
- Semantic Business logic layer (see section 5.9.2).
- Semantic Services / Application layer (see section 5.9.3).

As a result, the DRM Framework will improve current RA/RM and Information sharing legacy approaches by using ontologies and associated technologies, including the Ethereum Blockchain. A summary of all the solutions provided by the DRM Framework are presented in Table 2.

CHAPTER 6. ONTOLOGIES PROPOSAL

6.1. INTRODUCTION

In this work, we propose a mix of 3 standards to overcome all semantic expressiveness and limitations of STIX™ [OasisSTIX-TTP] [OasisSTIX-Campaign] [OasisSTIX-Incidents]:

- STIX™ [OasisSTIX1] as an Industry driven standard,
- OWL [W3C-OWL]
- SWRL [W3C-SWRL]

We consider that if our proposal is based on standards, it will be easier to implement and deploy by several organizations in the future.

Semantic ontologies [W3C-Ontology], supported by scientific language researchers as well as the World Wide Web Consortium (W3C), is a standard solution to create formal models, defining concepts, domains and relationships (even complex ones) guaranteeing unequivocal meaning.

They are considered building blocks for semantic inference [W3C-Inference], which is a mechanism to discover new relationships, to automatically analyze the content of the data and to manage knowledge. These inference based techniques are also important in discovering possible inconsistencies in the (integrated) data. The role of ontologies is to help data integration when, for example, ambiguities may exist on the terms used in the different data sets, or when a bit of extra knowledge may lead to the discovery of new relationships. Consider, for example, the application of ontologies in the field of health care. Medical professionals use them to represent knowledge about symptoms, diseases, and treatments. Pharmaceutical companies use them to represent information about drugs, dosages, and allergies. Combining this knowledge from the medical and pharmaceutical communities with patient data, enables a whole range of intelligent applications such as decision support tools that search for possible treatments, systems that monitor drug efficacy and possible side effects, and tools that support epidemiological research.

We selected the OWL Web Ontology Language [W3C-OWL] which is designed for the use of applications that need to process the content of information instead of presenting information to humans. OWL facilitates a greater machine capacity of interpretation of Web content than the one supported by XML, RDF, and RDF Schema (RDF-S). It is because it provides additional vocabulary along with formal semantics. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

Semantic web needed a separate language due to the nature of its applications. Interoperability is one of the primary goals of the Semantic Web and there is a significant interest in its standardization.

One of the key steps to rule interoperability on the Web is SWRL [W3C-SWRL] which was designed to be the rule language of the Semantic Web. SWRL is based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Ontology Web Language, the Unary/Binary Datalog (Datalog is a query and rule language for deductive databases that syntactically is a subset of Prolog) and Sublanguages of the Rule Markup Language.

SWRL permits users to write horn-like rules expressed in terms of OWL concepts in order to reason about OWL individuals. The rules can be used to get new knowledge from already existing OWL knowledge bases. The SWRL Specification does not impose restrictions on how reasoning should be performed with SWRL rules. Thus, developers are

free to use a variety of rule engines to reason with the SWRL rules stored in an OWL knowledge base.

SWRL [W3C-SWRL] increases the OWL expression ability to define rules and restrictions. Conditional rules (Horn clauses). SWRL allows defining complex conditions to be fulfilled in the antecedent of the rules, through the built-ins, the AND operator and the use of atoms. The use of variables in atoms allows defining constraints that are not possible in RDF or in OWL.

Taking into account that we developed the complete ontology version of STIX™ v2.0 as well as a complete DRM ontology there are several options to create enriched antecedent (body) and enriched consequent (head) rules as seen in this work.

Reasoners [W3C-Reasoner] are tools that can perform automatic and continuous reasoning tasks like inferencing, deriving new facts from existing ontologies and guaranteeing data consistency. They are based on RDF, OWL [W3C-OWL] or some rule engine like SWRL [W3C-SWRL].

We selected the Pellet Incremental reasoner [W3C-Pellet] for our work, based on Ontology Web Language (OWL) and SWRL to manage all our cybersecurity threat intelligence and risk data.

6.2. METHODOLOGY USED

There is no standard methodology to be followed for the development of an ontology, since it depends on the domain to be modeled and the context in which it is built, on the behavior to be specified and on the approach of the author of the ontology. There are many proposals for methodologies used to design and build ontologies. The most relevant are listed below:

- Methodology used to build the EO (Enterprise Ontology) ontology ([Uschold95] and [Uschold96]): This methodology is a scheme consisting of four steps, (1) Identify the purpose and scope of the ontology, (2) Build the ontology (Capture knowledge, Codify knowledge and Integrate knowledge), (3) Evaluate the ontology and (4) Document the ontology. In each of the steps, a set of guides or design recommendations are considered, which must be taken into account in each step of the method. This methodology is the basis of many of the methods proposed and used today.
- Methodology used in the development of the TOVE ontology (TOronto Virtual Enterprise) [Grüniger]: methodology used to build the ontology on business modeling processes, consisting of seven steps, (1) Capture motivational scenarios, (2) Formulate informal competence questions, (3) Specify the terminology of the ontology, (4) Formulate the list of competence questions formally, (5) Specify axioms and definitions for the terms, (6) Evaluate the ontology and (7) Define the conditions under which the solutions to the questions are complete.
- Unified methodology proposed by Uschold [Uschold96]: unified method that combines the best steps of EO and TOVE methodologies. It consists of four steps, (1) Define the purpose or objective of the ontology, (2) Decide the level of formalism of the ontology, (3) Define concepts and relationships between them and (4) Identify formal terms from the set of informal terms. In this last phase, the ontology design cycle is evaluated or reviewed, where the ontology is compared to the requirements of the users or companies.
- Methodology proposed by Sugumaran and Storey [Sugumaran02]: the authors propose a method based on heuristics for the development and creation of ontologies. First, we must (1) Identify the basic terms. Next, (2) Identify the relationships between the terms identified, phase in which the authors of the

methodology specify three types of relationships, generalization ("is-a" relationship), synonyms and association. In addition, in this second step, the relationships between ontologies are also taken into account and analyzed, in order to improve and enrich the ontology. The third step consists of (3) Identifying the basic constraints, that is, the relationship between terms and relationships, such as "one term or relationship depends on another", "one term or relationship must occur before another", etc. Finally, (4) Define high-level restrictions, such as domain restrictions and cross-domain dependencies.

- Methodology ([Fernández-López97], [Fernández-López99] and [Gómez-Pérez96]): methodology developed by the UPM (Polytechnic University of Madrid), which allows the construction of ontologies at the level of knowledge, composed of seven steps: (1) Specification of the objective (specify the domain and scope of the ontology and build a glossary of terms), (2) Conceptualization (build concept taxonomies and specify relationship diagrams), (3) Acquisition of knowledge, (4) Integration (analyze other ontologies for possible integration), (5) Implementation (build the ontology in formal language), (6) Evaluation and (7) Documentation.
- Stanford University Methodology [Noy01]: guide to create ontologies. It consists of seven steps:
 - Determine the domain and scope or scope of the ontology,
 - Consider reusing affected ontologies,
 - List the important terms in the ontology,
 - Define classes and class hierarchy,
 - Define the properties of the classes,
 - Define the characteristics of the properties such as cardinality, domain and rank, and finally,
 - Create instances of each class and fill in their properties.
- Methodology for the development of knowledge management based on ontologies [Staab01]: the authors propose a methodology consisting of five phases, (1) Feasibility study, (2) Preliminary development of the ontology (requirements specification, definition of the objective of the ontology, user identification, applications that could use the ontology, etc.), (3) Refinement and formal development, (4) Evaluation, and (5) Maintenance and evolution.

Regardless of the methodology used, every ontology must meet a series of requirements and properties [Gruber95]:

- It must be clear and objective: the meaning of the domain it represents must be specified objectively.
- It must be consistent and consistent: the inferences generated must be consistent with the ontology's own definitions. In addition, axiom definitions must be logically consistent.
- It must be complete: the ontology must contain the necessary and sufficient conditions for its complete specification, and not only the necessary conditions.
- It must be extensible, that is, to support the use of shared vocabulary: it must allow to be able to add new uses or axioms to the ontology, without having to review the existing definitions.
- You must comply with the principle of ontological distinction, that is, your classes must be disjoint.

- It must have modularity: it must be easily separable in independent modules.
- You should follow a standardization of attributes and properties as much as possible.

Despite Methodology is probably the most mature methodology, we have selected the Stanford University Methodology for both ontologies proposed, due to the fact that we start from STIX™ v2 taxonomy as a baseline, which is already conceptualized. This will also simplify the process.

6.3. TOOLS USED

6.3.1 OWL2

OWL and OWL2 are the most expressive languages. Although the structure of both languages is very similar, there are a number of advantages of OWL2 over OWL like OWL2 allows to reach a higher level of expressiveness than OWL (provides greater support for data types, has a greater number of axioms, includes cardinality restrictions and properties of asymmetry, reflexivity and disjunction, etc.) and the number of semantic reasoners that support it is greater as well. We will use OWL2 due this reasons.

6.3.2 Protégé

Protégé [Protégé] is a free and open source ontology editor, which provides tools for building domain models and knowledge-based applications through ontologies. Protégé consists of a set of structures that allow to create, visualize and manipulate ontologies defined in several representation formats, among them: RDF, OWL and OWL2.

Protégé presents a flexible architecture which facilitates the configuration and extension of the tool through plugins that extend its functionality, such as the Pellet reasoner plugin to support SWRL reasoning and inference, SWRLTab to support the definition of SWRL algorithms, and several ontology visualization plugins such as Ontograp and OWLViz.

For the creation of the proposed ontology, the following versions were used:

- Protégé: v5.0 to v5.5
- Protégé plugins:
- Pellet reasoner: v2.2.0
- SWRLTab: v2.0.5
- Ontograp: 2.0.3
- OWLViz: 5.0.3

6.4. CYBER THREAT INTELLIGENCE ONTOLOGY SPECIFICATION

6.4.1 Domain and Scope

The domain and scope selected for this ontology, is the Cyber Threat Intelligence domain under STIX™ in its version 2. From now on, we will use the naming convention “STIX2.owl” to talk about this specific ontology.

6.4.2 Reusing Affected Ontologies

The ontology is new, and it is using as a baseline the full version of the STIX™ v2 taxonomy from OASIS, in its last version.

During this thesis, there was a big update and a major release distribution by OASIS of a new version of STIX™. It required to create a complete new ontology and to completely migrate from v1.1 (see section 3.4.5) to v2 (see section 3.4.7), which was a big change (an effort), not only due to there was no possibility to reuse existing v1.1 ontologies, but also because v2 was completely re-shaped and merged with other taxonomies (e.g. CyBOX) as seen in STIX™ subsection.

6.4.3 The Important Terms

Important terms are also defined by the taxonomy in section 3.4.7.

6.4.4 Define Classes, Class Hierarchy, the Properties of the Classes and the Characteristics of the Properties

In the following sections, a brief description of each Class is provided.

For more information, the detailed Cyber Threat Intelligence Ontology (STIX2.owl) is described in ANNEX B.

A graphical representation of some of most of the classes of the CTI ontology (STIX2.owl) and their relationships is seen in Figure 62.

Figure 62 CTI Ontology representation (STIX2.owl)

6.4.4.1. ATTACK PATTERN

"Attack Patterns are a type of TTP that describe ways that adversaries attempt to compromise targets.

This class has the following relationships with other classes:

- createdByRef some Identity
- uses some Malware
- uses some Tool
- targets some Identity
- targets some Vulnerability
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.2. BUNDLE

"A Bundle is a collection of arbitrary STIX Objects and Marking Definitions grouped together in a single container.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.3. CAMPAIGN

"A Campaign is a grouping of adversary behavior that describes a set of malicious activities or attacks that occur over a period of time against a specific set of targets."

This class has the following relationships with other classes:

- attributedTo some IntrusionSet
- attributedTo some ThreatActor
- createdByRef some Identity
- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)
- targets some Identity
- targets some Vulnerability

- uses some AttackPattern
- uses some Malware
- uses some Tool
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.4. COURSE OF ACTION

"A Course of Action is an action taken either to prevent an attack or to respond to an attack that is in progress."

This class has the following relationships with other classes:

- createdByRef some Identity
- mitigates some AttackPattern
- mitigates some Malware
- mitigates some Tool
- mitigates some Vulnerability
- hasCyberObservable CyberObservable
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.5. CYBER OBSERVABLES

There are multiple types of CyberObservables. Each type is represented as a subclass.

They are:

- ARTIFACTS.
- AUTONOMOUS SYSTEM.
- DIRECTORY.
- DOMAIN NAME.
- EMAIL ADDRESS.
- EMAIL MESSAGE.
- FILE.
- IPV4 ADDRESS.
- IPV6 ADDRESS.
- MAC-ADDR.
- MUTEX.
- NETWORK-TRAFFIC.
- PROCESS.
- SOFTWARE.
- URL.

- USER ACCOUNT.
- WINDOWS REGISTRY KEY.
- X509 CERTIFICATE.

Depending on the specific subclasses, there are different relationships with other objects. Some examples:

- Common relationships:
 - observableOf Asset (relationship with DRM.owl ontology)
- Artifact:
 - relatedTo URL
 - redirection Network-Traffic
- IPv6Addr:
 - resolvesToRefs some MACAddr (IPV6Address)
 - belongToRefs some AutonomousSystem (IPV6Address)
- File:
 - isExecutedBy Users (relationship with DRM ontology)
- Etc.

6.4.4.6. DICTIONARY

“A dictionary captures an arbitrary set of key/value pairs”.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.7. IDENTITY

“Identities can represent actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, or groups (e.g., the finance sector).

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- hasAttributionOf some ThreatActor
- isImpersonatedBy some ThreatActor
- isTargetedBy some AttackPattern
- isTargetedBy some Campaign
- isTargetedBy some IntrusionSet
- isTargetedBy some Malware
- isTargetedBy some ThreatActor
- isTargetedBy some Tool

This class has no subclasses.

6.4.4.8. INCIDENT

"Reserved object" by STIX™ v2. SecurityEvents will be defined within DRM ontology, at least to work with events (which are a potential incident). This class has no further development on purpose, as **it is reserved**.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.9. INDICATOR

"Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity."

This class has the following relationships with other classes:

- createdByRef some Identity
- owl:Thing or (objectRefs some MarkingDefinition)
- indicates some AttackPattern
- indicates some Campaign
- indicates some IntrusionSet
- indicates some Malware
- indicates some ThreatActor
- indicates some Tool
- hasCyberObservable CyberObservable

This class has no subclasses.

6.4.4.10. INFRASTRUCTURE

"Reserved object" by STIX™ v2. This class has no further development on purpose, as it is reserved.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.11. INTRUSION SET

"An Intrusion Set is a grouped set of adversary behavior and resources with common properties that is believed to be orchestrated by a single organization."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- attributedTo some ThreatActor
- createdByRef some Identity
- hasAttributionOf some Campaign

- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some AttackPattern
- uses some Malware
- uses some Tool

This class has no subclasses.

6.4.4.12. KILL CHAIN PHASE

"The Kill Chain Phase represents a phase in a kill chain, which describes the various phases. This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.4.4.13. MALWARE

"Malware is a type of TTP that is also known as malicious code and malicious software, refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware
- hasCyberObservable CyberObservable

This class has multiple subclasses:

- ADWARE
- BACKDOOR
- BOT

- DDOS
- DROPPER
- EXPLOITKIT
- KEYLOGGER
- RANSOMWARE
- REMOTEACCESSTROJAN
- RESOURCE EXPLOITATION
- ROGUE SECURITY SOFTWARE
- ROOTKIT
- SCREEN CAPTURE
- SPYWARE
- TROJAN
- VIRUS
- WORM

6.4.4.14. MARKING DEFINITION

"The marking-definition object represents a specific marking. Data markings represent restrictions, permissions, and other guidance for how data can be used and shared. For example, data may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest".

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has two subclasses:

- TLP
- STATEMENT

6.4.4.15. OBSERVED DATA

"Observed data conveys information that was observed on systems and networks, such as log data or network traffic, using the Cyber Observable specification."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

6.4.4.16. REPORT

"Reports are collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

6.4.4.17. SIGHTING

"A Sighting denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition
- sightingOfRef some owl:Thing
- observedDataRefs some ObservedData
- whereSightedRefs some Identity

This class has no subclasses.

6.4.4.18. THREAT ACTOR

"Threat Actors are actual individuals, groups, or organizations believed to be operating with malicious intent."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- attributedTo some Identity
- createdByRef some Identity
- hasAttributionOf some Campaign
- hasAttributionOf some IntrusionSet
- impersonates some Identity
- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some AttackPattern
- uses some Malware
- uses some Tool

This class has no subclasses.

6.4.4.19. TOOL

"Tools are legitimate software that can be used by threat actors to perform attacks".

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some Malware
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability

This class has no subclasses.

6.4.4.20. VULNERABILITY

"A Vulnerability is a mistake in software that can be directly used by a hacker to gain access to a system or network."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isMitigatedBy some CourseOfAction
- isTargetedBy some AttackPattern
- isTargetedBy some Campaign
- isTargetedBy some IntrusionSet
- isTargetedBy some Malware
- isTargetedBy some ThreatActor
- isTargetedBy some Tool
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

6.4.5 Create Instances of each Class and fill in their Properties

For the validation related to the use case defined in 9.3.1, all the needed instances and properties were created.

As an example, see Figure 63 Example of some CyberObservable instances (STIX2.owl ontology):

Figure 63 Example of some CyberObservable instances (STIX2.owl ontology)

6.5. DYNAMIC RISK MANAGEMENT ONTOLOGY SPECIFICATION

6.5.1 Domain and Scope

The domain selected for this ontology, is the Risk Management domain, including their sub phases of either Risk Assessment (RA) or Risk Management (RM) processes.

The scope, will be at least the current scope of the ISMS (Information Security Management System) of any organization. Beyond that, the ontology proposed leverages the Cyber Threat Intelligence domain to get a Dynamic Risk Management framework.

It is then an OWL representation of a Cybersecurity DRA/DRM (Dynamic Risk Assessment and Dynamic Risk Management) model valid for any organization and potentially compatible with any risk assessment & management process of any information security risk management (ISMS) standard.

From now on, we will use the naming convention “DRM.owl” to talk about this specific ontology, which interconnected with STIX2.owl thanks a direct “import” operation. As one of the main contribution of this thesis, it allows the interoperability and enrichment of both domains dynamically.

6.5.2 Reusing Affected Ontologies

Despite the ontology was created from scratch, it uses few generic concepts that came from different Risk Management ontologies or contributions as seen in related work section. Those general concepts were also extended with specific concepts and relationships to connect it with CTI Ontology.

DRM.owl uses STIX2.owl as an imported ontology. During this thesis, there was a big update and a major release distribution by OASIS of a new version of STIX™. It required to create a complete new STIX2.owl ontology and to completely migrate from its v1.1 (see section 3.4.5) to v2 (see section 3.4.7). Thanks to the import feature, we replaced in DRM.owl the previous STIX ontology import (of v1.1) by a new import of STIX2.owl ontology. As it is not automatic, there were some tasks duplicated to validate that the new ontology was perfectly replacing the previous version.

6.5.3 The Important Terms

Important terms are also defined by the taxonomy in subsection 2.3.

6.5.4 Define Classes, Class Hierarchy, the Properties of the Classes and the Characteristics of the Properties

In the following sections, a brief description of each Class is provided.

For more information, the detailed Dynamic Risk Management Ontology (DRM.owl) is described in ANNEX C.

A graphical representation of most of the classes of the DRM ontology (DRM.owl) and their relationships is seen in Figure 64.

Figure 64 DRM ontology overview (DRM.owl imports STIX2.owl)

6.5.4.1. ASSET

"They are the different (valuable) assets integrated within and dependent of a business process."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- hasCyberObservable some CyberObservable
- hasVulnerability some Vulnerability
- isExposedTo some Threat
- isProtectedBy some Safeguards

This class has several subclasses:

- Data
 - ClassifiedData
 - Equivalent to "File" (from STIX2.owl)
 - PersonalDataProtection
- Hardware
 - BusinessRecovery
 - Computer
 - IoT
 - PersonalComputer
 - Server
 - MainBackboneHW
 - MobileDevices
 - NetworkDevice
 - Modem
 - Router
 - Switch
 - VirtualMachines
- Networks
 - CellCommunications
 - Internet
 - ISDN
 - LAN
 - PSTN
 - SatelliteCommunications
 - VLAN
 - VPN

- WLAN
- xDSL
- Processes : Equivalent to “Process” (STIX2.owl)
 - RelatedProcesses
- RealEstate
 - Building
 - Floor
 - Room
- ServiceDelivery
 - SelfProvided
 - ThirdPartyProvided
- Software
 - AcquiredSW
 - DevelopedSW
- Users
 - ExternalUsers
 - Contractors
 - InternalUsers

6.5.4.2. ASSET VALUATION

"The value of the assets of the organization based on an added value assessment."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only RiskScope
- evaluates some Asset

6.5.4.3. CONTEXT

"It is the context (internal or external) including dependencies where our business processes are being deployed."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity

It has multiple subclasses:

- ExternalContext
 - ComplianceContext
 - LegalContext
 - PoliticalContext

- InternalContext
 - RelatedAssets
 - StrategicAlignment
 - StrategicIndicator
 - StrategicObjective

6.5.4.4. SAFEGUARDS

"Safeguards to minimize the impact, or the probability, of a specific threat."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- mitigates some RiskImpact
- mitigates some RiskProbability
- mitigates some Vulnerability
- protects some Asset
- relatedTo some CourseOfAction

It has multiple subclasses:

- Control
 - NonStandardControl
 - StandardControl
 - ISO2700XControl
 - AccessControl
 - AccessControlBusinessRequisitesControl
 - AccessPolicyControl
 - NetworkAccessPolicyControl
 - AccessControlToSystemsandApplications
 - InformationAccessRestrictionControl
 - LoginSecureProcessesControl
 - PasswordManagementControl
 - ProgramSourceCodeAccessControl
 - RootPrivilegeUsedToolsControl
 - EndUserAccessManagementControl
 - PrivilegeManagementControl
 - ProvisionOfUserAccessControl
 - UserAccessRightsCancellationControl
 - UserAccessRightsRevisionControl
 - UserAuthenticationSecretInformationManagementControl

- UserRegistrationandCancellationControl
 - UserResponsibilityControl
 - AuthenticationSecretInformationUsageControl
- AssetManagementControl
 - AssetResponsibilityControl
 - AssetAcceptableUsageControl
 - AssetInventoryControl
 - AssetOwnershipControl
 - AssetReturnControl
 - InformationClassificationControl
 - InformationClassifyingControl
 - InformationHandlingControl
 - InformationLabelingControl
 - MediaHandlingControl
 - PhysicalSupportInTransitControl
 - RemovableSupportManagementControl
 - SupportRemovalControl
- BusinessContinuityControl
 - InformationSecurityContinuityControl
 - CheckandMaintenanceOfContinuityPlanControl
 - InformationSecurityContinuityImplementationControl
 - InformationSecurityContinuityPlanControl
 - RedundancyControl
 - BackupFacilityAvailableControl
- ComplianceControl
 - InformationSecurityRevision
 - InformationSecurityIndependentRevision
 - OrganizationPoliciesandGuidelinesCompliance
 - TechnicalComplianceAuditControl
 - LegalRequirementsComplianceControl
 - ApplicableLawIdentificationControl
 - CryptographicControlsRegulation
 - IntellectualPropertyRightsControl
 - OrganizationRegistersProtectionControl
 - PrivacyandDataProtectionControl
- CryptoControl
 - CryptographicControl

- CryptographicControlUsagePolicy
 - KeysManagementControl
- HRRelatedSecurityControl
 - HRRelatedSecurityAfterChangeJobControl
 - HRRelatedSecurityResponsibilitiesAfterQuittingJobControl
 - HRRelatedSecurityBeforeJobControl
 - HRRelatedSecurityBackgroundResearchControl
 - HRRelatedSecurityTermsandConditionsControl
 - HRRelatedSecurityDuringJobControl
 - HRRelatedSecurityAwarenessEducationandCybercapabilitiesControl
 - HRRelatedSecurityDisciplinaryProcessControl
 - HRRelatedSecurityManagementResponsibilitiesControl
- IncidentHandlingControl
 - IncidentHandlingandImprovementsControl
 - EvidenceCollectionControl
 - IncidentHandlingProceduresandResponsibilitiesControl
 - IncidentKnowledgeBaseControl
 - InformationSecurityEventAssessmentandDecisionMakingControl
 - InformationSecurityEventNotificationControl
 - InformationSecurityProceduresandResponsibilitiesControl
 - SecurityIncidentResponseControl
 - SecurityWeakPointNotificationControl
- InformationSecurityOrganizationControl
 - InformationSecurityInternalOrganizationControl
 - InformationSecurityContactWithAuthoritiesControl
 - InformationSecurityContactWithInterestGroupsControl
 - InformationSecurityInProjectManagementControl
 - InformationSecurityRolesandResponsibilitiesControl
 - InformationSecurityTaskSegregationControl
 - InformationSecurityMobileUsersControl
 - InformationSecurityHomeworkingControl
 - InformationSecurityMobileDevicesPolicyControl

- InformationSecurityPoliciesControl
 - InformationSecurityManagementGuidelines
 - InformationSecurityPolicy
 - InformationSecurityPolicyRevision
- IT_Acquisition_O&M_Control
 - DevOpsControl
 - ApplicationTestAfterOSChangesControl
 - OutsourcedSWDevelopmentControl
 - SecureCodingIntegrationIntoIDEControl
 - SecureCodingPolicyControl
 - SecureEngineeringDesignControl
 - SecurityFunctionalTestControl
 - SWPackagesChangingRestrictionsControl
 - SystemChangeManagementProcedureControl
 - SystemsAcceptanceTestControl
 - ITSecurityRequirementControl
 - ApplicationServicesSecurityInPublicNetworksControl
 - ITSecurityRequirementandSpecificationAnalysisControl
 - TransactionProtectionInApplicationServicesControl
 - TestDataControl
 - TestDataProtectionControl
- NetworkSecurityControl
 - InformationSharingControl
 - eMessagingControl
 - InformationSharingAgreementsControl
 - InformationSharingProceduresandPoliciesControl
 - NonDisclosureAgreementControl
 - NetworkSecurityManagementControl
 - NetworkControl
 - NetworkSegmentationControl
 - NetworkServicesSecurityControl
- OperationSecurityControl
 - BackupControl
 - ITAuditControl
 - ITAuditingControl
 - MaliciousSWProtectionControl
 - ControlsAgainstMaliciousSW

- OperationProceduresandResponsibilityControl
 - CapabilityManagementControl
 - ChangeManagementControl
 - DevTestandOperationResourcesSegmentationControl
 - OperationProceduresDocumentationControl
- RegisterandSupervisionControl
 - AdminandOperationLogControl
 - ClockSynchronizationControl
 - EventLogControl
 - LogInformationProtectionControl
- SWControl
 - SWInstallationControl
- VulnerabilityManagementControl
 - SWInstallationRestrictionControl
 - TechnicalVulnerabilitiesControl
- PhysicalSecurityControl
 - DeviceSecurityControl
 - CleanScreenandWorkspacePolicyControl
 - CompanyOwnedDiscardedMaterialControl
 - EquipmentMaintenanceAreaControl
 - EquipmentSecurityOutsideCompanyPremisesControl
 - LocationandEquipmentProtectionControl
 - ReuseOrEliminationOfEquipmentControl
 - SupplyFacilitiesControl
 - UnattendedUserEquipmentControl
 - WiringSecurityControl
 - SecuredAreasControl
 - LoadingandUnloadingAreasControl
 - PhysicalEntranceControl
 - PhysicalPerimeterSecurityControl
 - ProtectionAgainstExternalandEnvironmentalThreatsControl
 - ResourcesandOfficesSecurityControl
 - SecureWorkSpaceAreaControl
- ProvidersRelationshipControl
 - ProvidersRelationshipSecurityControl
 - ICTSupplyChainControl

- InformationSecurityPolicyForSupplyChainControl
 - ThirdPartiesSecurityRequirementsForContractsControl
 - ProvidersSupplyControl
 - ThirdPartyProvidedServicesAuditControl
 - ThirdPartyProvidedServicesChangeManagementControl
- Countermeasures
 - Firecountermeasure
 - FireExtinguisher
 - CO2FireExtinguisher
 - HandFireExtinguisher
 - WaterFireExtinguisher
- Detector
 - FireDetector
- FaultForecasting
 - OrdinalEvaluation
 - ProbabilisticEvaluation
 - Modeling
 - OperationalTesting
- FaultPrevention
- FaultRemoval
 - Correction
 - Diagnosis
 - NonRegressionVerification
 - Verification
 - DynamicVerification
 - StaticVerification
- FaultTolerance
 - ErrorDetection
 - ConcurrentDetection
 - PreemptiveDetection
 - Recovery
 - ErrorHandling
 - FaultHandling

6.5.4.5. SECURITY EVENTS

"Security events (alerts) produced by detections of our semantic Cyber Threat Intelligence framework".

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- (dstRef some DomainName) or (dstRef some IPv4Addr) or (dstRef some IPv6Addr) or (dstRef some URL)
- (srcRef some IPv4Addr) or (srcRef some IPv6Addr)
- createdByRef some Identity
- relatedTo some NetworkTraffic

It has no subclasses.

6.5.4.6. INCIDENT

"Reserved object" by STIX™ v2, we also reserve it at DRM. SecurityEvents will be defined within the DRM ontology, at least to work with events (which are potential incidents). This class has no further development on purpose, as it is reserved.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

6.5.4.7. RISK MANAGEMENT

"Risk Management strategies".

It has multiple subclasses:

- RiskInvestigation
- RiskMitigation
- RiskMonitoring
- RiskResponse

6.5.4.8. THREAT

"A threat is anything (manmade or act of nature) that has the potential to cause harm (e.g. exploits a vulnerability)"

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- exploits some Vulnerability
- initiatedBy only ThreatActor
- threatens some Asset

It has multiple subclasses:

- ComplianceThreat
 - DataProtectionRisks
 - OtherLegalObligationRisks

- DeliberatedThreat
 - DeliberatedConfigFilesTampering
 - DeliberatedHWTampering
 - DeliberatedInformationDestruction
 - DeliberatedInformationLeak
 - DeliberatedInformationTampering
 - DeliberatedMaliciousSWDistribution
 - DeliberatedRegistersTampering
 - DeliberatedSWTampering
 - DeliberatedUnauthorizedAccess
 - DeviceTheft
- HumanThreat
 - InsiderThreats
- ICTThreat
 - ConfigurationError
 - DeliberatedConfigFilesTampering
 - DeliberatedHWTampering
 - DeliberatedInformationDestruction
 - DeliberatedInformationLeak
 - DeliberatedInformationTampering
 - DeliberatedMaliciousSWDistribution
 - DeliberatedRegistersTampering
 - DeliberatedSWTampering
 - DeliberatedUnauthorizedAccess
 - DenialOfService
 - DeviceLost
 - DeviceTheft
 - HumanResourcesNotAvailable
 - HWMaintenanceError
 - IdentityThief
 - MonitoringError
 - NetworkOutage
 - NonIntentionalAdminError
 - NonIntentionalInformationDestruction
 - NonIntentionalInformationLeak
 - NonIntentionalInformationTampering
 - NonIntentionalMaliciousSWDistribution

- NonIntentionalUserError
- PhysicalFault
- PowerOutage
- PrivilegeEscalation
- SocialEngineering
- SWMaintenanceError
- SWVulnerabilities
- UnexpectedUsage
- PhysicalThreat
 - Accidents
 - Fire
 - NaturalDisasters
 - Terrorism
- Quality&ProcessThreat
 - CorporateBrandImageDamages
 - DelayedDelivery
 - StakeholdersSatisfaction
 - TechnicalComplexity
 - Untrustworthy
 - UsersComplaints
- StrategicThreat
 - BadReputationThreat
 - EconomicLoss
 - PartnershipsIssues
 - StakeholdersIssues
 - StrategicPlanOnRisk
- UnintentionalThreat
 - ConfigurationError
 - DeviceLost
 - HWMaintenanceError
 - NonIntentionalAdminError
 - NonIntentionalInformationDestruction
 - NonIntentionalInformationLeak
 - NonIntentionalInformationTampering
 - NonIntentionalMaliciousSWDistribution
 - NonIntentionalUserError
 - PowerOutage

6.5.4.9. RISK SEVERITY

"Risk Severity: Cumulative Risk classification by Severity as a mix of Impact and Probability."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk

It has the following subclasses:

- ExtremeRiskSeverity
- HighRiskSeverity
- LowRiskSeverity
- MediumRiskSeverity

6.5.4.10. RISK SCOPE

"It represents the scope of the Dynamic Risk Assessment (DRA) and Dynamic Risks Management (DRM), that is to say, the different "services" or "projects" of an organization that are included in the scope of their risk assessment and risk management model."

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- dependsOn some Data
- hasContext some Context

It has no subclasses.

6.5.4.11. RISK

"Risks of our Organization"

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- (isGeneratedBy min 1 Threat) or (isGeneratedBy min 1 Vulnerability)
- belongToRefs min 1 RiskOwner
- createdByRef some Identity
- damages min 1 Asset
- hasAssessmentOf only RiskAssessment
- hasImpactOf only RiskImpact
- hasProbabilityOf only RiskProbability
- hasSeverityOf only RiskSeverity
- isGeneratedBy min 1 Vulnerability

- threatens only RiskScope

It has multiple subclasses (also related to ISO31000):
- ComplianceRisk
 - DataProtectionComplianceRisk
 - OtherLegalComplianceRisk
- InformationSecurityRisk
 - ConfigurationErrorRisk
 - DeliberatedConfigFilesTamperingRisk
 - DeliberatedHWTamperingRisk
 - DeliberatedInformationDestructionRisk
 - DeliberatedInformationLeakRisk
 - DeliberatedInformationTamperingRisk
 - DeliberatedMaliciousSWDistributionRisk
 - DeliberatedRegistersTamperingRisk
 - DeliberatedSWTamperingRisk
 - DeliberatedUnauthorizedAccessRisk
 - DenialOfServiceRisk
 - DeviceLostRisk
 - DeviceTheftRisk
 - HumanResourcesNotAvailableRisk
 - HWMaintenanceErrorRisk
 - IdentityThiefRisk
 - LogicalFailureRisk
 - MonitoringErrorRisk
 - NetworkOutageRisk
 - NonIntentionalAdminErrorRisk
 - NonIntentionalInformationDestructionRisk
 - NonIntentionalInformationLeakRisk
 - NonIntentionalInformationTamperingRisk
 - NonIntentionalMaliciousSWDistributionRisk
 - NonIntentionalUserErrorRisk
 - PhysicalFailureRisk
 - PowerOutageRisk
 - SocialEngineeringRisk
 - SWMaintenanceErrorRisk
 - SWVulnerabilitiesRisk
- PhysicalRisk

- AccidentRisk
- FireRisk
- FloodingRisk
- NaturalDisasterRisk
- TerrorismAttackRisk
- Quality&ProcessRisk
 - CorporateBrandImageDamageRisk
 - DelayedDeliveryRisk
 - StakeholdersSatisfactionRisk
 - TechnicalComplexityDerivedRisk
 - UntrustworthyRisk
 - UsersComplaintsRisk
- StrategicRisk
 - BadReputationRisk
 - EconomicLossRisk
 - PartnershipRisk
 - PressNegativeImpactRisk
 - StakeholdersRisk
 - StrategicObjectiveRisk
- OtherRisk

6.5.4.12. RISK IMPACT

"Risk Impact: individual Risk Classification by Impact severity"

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk

It has multiple subclasses:

- CriticalImpactRisk
- HighImpactRisk
- LowImpactRisk
- MediumImpactRisk
- VeryLowImpactRisk

6.5.4.13. RISK OWNER

"Owner of all risks associated with a specific RiskScope (service, project, tool, etc.)".

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

- createdByRef some Identity
- evaluates min 1 RiskAssessment
- isResponsibleOf min 1 Risk
- isResponsibleOf min 1 RiskScope

It has no subclasses.

6.5.4.14. RISK PROBABILITY

"Risk Probability: individual Risk Classification by probability"

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity

It has multiple subclasses:

- ExtremeRiskProbability
- HighRiskProbability
- LowRiskProbability
- MediumRiskProbability
- VeryLowRiskProbability

6.5.4.15. RISK ASSESSMENT

"Risk Assessment:

- Potential Risk (w/o safeguards)
- Residual Risk (considering safeguards)"

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk
- isIncreasedBy some SecurityEvents
- isMitigatedBy some Safeguards
- threatens only RiskScope

It has multiple subclasses:

- PotentialRisk
- PotentialTotalRisk
- ResidualRisk
- ResidualTotalRisk

6.5.5 Create Instances of Each Class and Fill in Their Properties

For the validation related to the use case defined in 9.3.1, all the needed instances and properties were created.

As an example, see Figure 65 Visualization of some DRM ontology instances:

Figure 65 Visualization of some DRM ontology instances

Due to the use of SWRL, automatic instances are also created by the reasoner. Those are marked by Protégé by assigning automatic ID to each of the instances. Their prefix will be "owl:" followed by 32 characters.

As an example, see Figure 66 Automatic instances of class RiskAssessment, created by the reasoner.

Figure 66 Automatic instances of class RiskAssessment, created by the reasoner

6.6. CONCLUSIONS

As a conclusion, we have enriched each of the domains by mixing them. We now can have as much expressivity as possible in both domains but also in the new super-domain, as all classes, their properties and relationships are part of it.

As an example, we can now have a better visibility of both domains in just one single view as seen in next Figure 67 Visual ontology graph of the super-domain (STIX/CTI and Risk domains). In the figure, a residual risk probability is increased due to different security events that are dynamically occurring around the scope and context of the risk assessment carried out by the organization.

Figure 67 Visual ontology graph of the super-domain (STIX/CTI and Risk domains)

CHAPTER 7. REASONING AND BEHAVIOR

7.1. INTRODUCTION

As described in section 4.4, section 5.7.5, section 5.9 as well as seen in Figure 57, we propose the use of a semantic reasoner [W3C-Reasoner] as one of the main building blocks of our architecture.

There are several reasons, among them:

- It supports the **inference of new knowledge** (e.g. advanced intelligence) about CTI and DRM data using both forward chaining and backward chaining reasoning algorithm.
- It provides **consistency check** for CTI and DRM, avoiding data conflicts.
- It provides **automatic classification**, creating a complete hierarchy of classes taking into account all subclasses and relationships. This hierarchy can be used to formulate queries such as: all subclasses of a concept, infer new subclasses of a concept, etc.
- It enables the **automatic instantiation by finding the most specific classes to which an individual belongs**. The realization can only be carried out after classification.
- It provides **ontology debugging services** by a diagnosis and an error resolution (the latter is not usually provided frequently).
- It supports **reasoning using the full expressiveness of OWL-DL**, and some features of OWL-Full.

We also propose SWRL to represent CTI and DRM behaviors as described in section 4.3.2.

7.2. USE OF REASONING FOR DYNAMIC RISK AND CTI INTELLIGENCE BEHAVIORAL DOMAINS

Our CTI and DRM model approach (based on ontologies, SWRL and a reasoner) can be understood as part of the concept of "**Adaptive Intelligence**".

Adaptive Intelligence is the intelligence that not only responds to changes, but also creates changes".

Thanks to the use of ontologies, SWRL and a semantic reasoner, where the data is understood either by humans or the system, our CTI and DRM framework will create changes automatically. The changes can be triggered by new knowledge as part of new data coming from external sources (e.g. information sharing) but also due to a new discovery or inference (e.g. new relationship, new context) within our current data-set.

The characteristics of semantic reasoners allow the provision of advanced services at different levels (Technical, Tactic and Strategic), among them:

- **Enrichment, contextualization, generation and inference of new behavioral intelligence** about risks and cyber threats, by using SWRL rules.

This will automatically allow:

- the establishment of all relationships between individuals,
- better contextualization,
- better discovery of new dependencies
- enhanced or advanced semantic search, using semantic graphs.
- **Dynamic risk analysis:** to semantically and dynamically address the different phases of a risk analysis process:
 - Asset inventory: enabling the discovery of new related assets with certain services and keep an adequate asset inventory control.
 - Identification and discovery of dependencies among them.
 - Asset valuation (inheritance and dependencies)
 - Dynamic identification and inventory of threats.
 - Evaluation of threats.
 - Calculation of Impact and Potential Risk.
- **Dynamic risk treatment:** to semantically and dynamically address the different phases of risk treatment process (with simulation capability):
 - Selection of the optimal strategy:
 - Transfer risk.
 - Mitigate risk.
 - Monitoring.
 - Selection of the optimal tactic:
 - E.g. Share the “Intelligence” with peers to prevent further attacks, honey tokens, deception techniques, etc.
 - Selection of the optimal technical response:
 - E.g. Countermeasures, open a security incident, security policies changes, etc.

In the following table, all features of section 5.9 are associated to the characteristics described above (see section 7.1).

Table 5 Table of characteristics and features provided by the reasoner within the DRM framework

Reasoner characteristics	Features provided by SWRL and the reasoner within the DRM Framework
Inference of new knowledge	Enrichment of Threat Information Context (e.g. TTP and more expressive context). Enrichment of Risk Assessment and Risk Management Context (e.g. Asset inventory, dependencies, automatic identification of potential threats, etc.). Automation of Risk Management (e.g. Inference of optimal tactic and strategy) Security Event Detection (e.g. detection of incidents based on a combination of events – behavioral or context-aware rules) Security Incident Detection (e.g. real incident based on events or alarms and the behavior or patterns followed)

Consistency check	Data consistency is needed to implement reliable actionable intelligence: processes and services. (e.g. consistency among STIX2.owl and DRM.owl ontologies)
Automatic classification	Automation of Threat Classification. Automation of Risk Management (e.g. selection of optimal strategy) Incident Management (e.g. Incident Triage.)
Automatic instantiation	Intelligence algorithms to classify unclassified or new data.
Ontology debugging	When new inconsistencies appear, a proper description of the root cause is very important.
Full expressiveness	To overcome the lack of expressiveness to represent more complex statements like TTP (Techniques, Tactics and Procedures) [OasisSTIX-TTP]

In addition to this, the contribution of the SWRL and the reasoner to the specific challenges shown in Table 1 (and based on Table 2), are the following:

Table 6 Solution provided to challenges of Table 1 by the use of SWRL and a Semantic Reasoner

ID	Solution provided by the use of a Semantic Reasoner	IDs of Table 1 solved
1	CTI and risk data expressiveness provided by W3C semantic web ontologies and SWRL	10, 14, 15
2	Sharing of CTI and Risk behavioral context aware rules or algorithms beyond IoC	2, 9, 11, 12, 13, 16
3	Automation by using semantic web reasoners (analysis beyond data collection)	10, 12, 13, 17, 21
4	Quality of the data as inconsistencies are automatically detected by reasoners	10, 15, 21
5	Inference (new knowledge) provided by the use of semantic web reasoners	1, 2, 6, 19
6	Reduced volume of data needed (e.g. algorithms vs. IoC)	8, 9, 11, 12, 13
7	Situational awareness linked to information exchange provided by ontologies	15, 16, 17
8	Information sharing at all levels (operational, tactical and strategic)	15, 16, 17, 18

7.3. TYPES OF REASONING RULES

In order to better understand the type of SWRL reasoning rules developed along the thesis, the following section provides an example per each type of rule.

For more detailed information and examples, see ANNEX D.

7.3.1 Asset Valuation Rules

In this type of SWRL rules, the reasoner will infer new knowledge about the value of our assets, if applying them over an updated asset inventory where the Risk Owners had already evaluated their services in scope.

The rules will implement the following schema of inheritance due to the usual Risk management dependency tree within the assets.

Figure 68 Asset dependency tree for asset valuation

The following SWRL rule will provide a data inherit automatic valuation coming from higher level of the hierarchy and taking into account the dependency tree.

Table 7 Example of an Asset Valuation rule about a "Data Inherit Valuation"

AssetValuation#1 Data Inherit Valuation
<p>The reasoner will infer the value of the Data, as an automatic asset valuation, due to its dependencies. As it is in the top of the dependency tree, it takes the valuation of the risk scope, done by the Risk Owner:</p> <p>$(\text{drm:AssetValuation}(?av) \wedge \text{drm:evaluates}(?av, ?rs) \wedge \text{drm:RiskScope}(?rs)).$</p>
<p> $\text{drm:recoveryPointObjective}(?av, ?rpo) \wedge \text{drm:recoveryTimeObjective}(?av, ?rto) \wedge$ $\text{drm:Data}(?data) \wedge \text{drm:dependsOn}(?rs, ?data) \wedge \text{drm:integrity}(?av, ?i) \wedge$ $\text{drm:confidentiality}(?av, ?c) \wedge \text{drm:evaluates}(?av, ?rs) \wedge \text{drm:AssetValuation}(?av) \wedge$ $\text{drm:availability}(?av, ?a) \wedge \text{drm:RiskScope}(?rs) \wedge \text{drm:authenticity}(?av, ?au) \wedge$ $\text{swrlx:makeOWLThing}(?x, ?av, ?data) \wedge \text{drm:accounting}(?av, ?ac)$ $\rightarrow$ $\text{drm:recoveryPointObjective}(?x, ?rpo) \wedge \text{drm:confidentiality}(?x, ?c) \wedge \text{drm:integrity}(?x, ?i) \wedge$ $\text{drm:accounting}(?x, ?ac) \wedge \text{drm:authenticity}(?x, ?au) \wedge \text{drm:evaluates}(?x, ?data) \wedge$ $\text{drm:availability}(?x, ?a) \wedge \text{drm:AssetValuation}(?x) \wedge \text{drm:recoveryTimeObjective}(?x, ?rto)$ </p>

7.3.2 Threat Inventory Rules

The reasoner will infer new knowledge with regard to any potential Threat that has to be considered. If applying the following SWRL rules (examples) over our data-set, the reasoner will create instances of different type of threats that are usually very difficult or complex, to be manually identified by the risk owners. It will create an automatic inventory of threats at different levels (strategic, tactic and technical), to contribute to the whole process. It will be an input of the Risk Inventory.

Table 8 Example of a Threat Inventory rule to automate the identification of Bad Reputation threats associated to Classified Data

ThreatInventory#1 Auto Bad Reputation Threat in Classified Data
The reasoner will infer new knowledge, in this case, it will create Bad Reputation threats associated to our Classified Data. It will consider that any data leak (in this case, especially associated to Classified data) will be a Bad Reputation Threat to our organization.
<pre> drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:ClassifiedData(?data) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?x, ?data) ^ drm:dependsOn(?rs, ?data) -> drm:impact(?x, "4.0"^^xsd:float) ^ drm:probability(?x, "2.0"^^xsd:float) ^ stix2:type(?x, "Threat Bad Reputation"^^rdf:PlainLiteral) ^ drm:threatens(?x, ?data) ^ drm:BadReputationThreat(?x) </pre>

7.3.3 Risk Inventory Rules

The reasoner will infer new knowledge with regard to any potential Risk that has to be considered. If applying the following SWRL rules (examples) over our data-set, the reasoner will create instances of different type of risks that are usually very difficult or complex, to be manually identified by the risk owners. It will create an automatic inventory of risks at different levels (strategic, tactic and technical), to contribute to the whole process.

Table 9 Example of a Risk Inventory rule to automate the identification of Bad Reputation Risks

RiskInventory#1 Auto Bad Reputation Risk
The reasoner will infer new classes of Bad Reputation Risk, associated to the "Bad Reputation Threats" and it will handle inheritance along the dependency tree.
<pre> drm:dependsOn(?rs, ?a) ^ drm:threatens(?x, ?a) ^ drm:impact(?x, ?i) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?r, ?x) ^ drm:probability(?x, ?p) ^ drm:BadReputationThreat(?x) -> stix2:type(?r, "Bad Reputation Risk"^^rdf:PlainLiteral) ^ drm:isGeneratedBy(?r, ?x) ^ drm:threatens(?r, ?rs) ^ drm:BadReputationRisk(?r) ^ drm:damages(?r, ?a) </pre>

7.3.4 Data Enrichment Rules

The reasoner will infer new knowledge about the enrichment of our data, if applying the following SWRL rules (examples) over our data-set.

Table 10 Example of an Enrichment rule to automate the enrichment of DNS related data by its RDNS.

Enrichment#1 Auto IPv4 assignment by Domain match (DNS)

The reasoner will infer a DNS value of a domain if a Reverse DNS connection is known.
<pre> stix2:IPv4Addr(?ip) ^ stix2:belongsToRefs(?ip, ?ref) ^ stix2:DomainName(?ref) -> stix2:DomainName(?ref) ^ stix2:resolvesToRefs(?ref, ?ip) </pre>

7.3.5 Risk Assessment Rules

The reasoner will infer new knowledge related to the Risk Assessment's process. It will make specific calculations by applying the following SWRL rules (examples) over our data-set.

Table 11 Example of a Risk Assessment rule to automate the assessment of Risks

RiskAssessment#1 Auto Potential Risk Assessment
The reasoner will infer new instances of Potential Risk assessment classes, together with all their values, connections and dependencies.
<pre> drm:RiskScope(?rs) ^ stix2:type(?r, ?ty) ^ swrlb:add(?x, ?p, ?i) ^ drm:threatens(?r, ?rs) ^ drm:isGeneratedBy(?r, ?th) ^ drm:Risk(?r) ^ drm:impact(?th, ?i) ^ swrlx:makeOWLThing(?y, ?r) ^ drm:probability(?th, ?p) -> drm:impact(?y, ?i) ^ drm:threatens(?y, ?rs) ^ drm:evaluates(?y, ?r) ^ drm:potentialRisk(?y, ?x) ^ drm:PotentialRisk(?y) ^ drm:probability(?y, ?p) ^ stix2:type(?y, ?ty) </pre>

7.3.6 Risk Severity Rules

The reasoner will classify risks into their corresponding severity levels depending on their "actualRisk" value.

Table 12 Example of a Risk Severity rule to automatically classify any Extreme Risks

RiskSeverity#1 Extreme Risk Classification
The reasoner will infer (and classify), that all Risks between 8 and 10 are considered of Extreme Risk Severity.
<pre> swrlb:lessThan(?ar, 10) ^ drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 8) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ drm:evaluates(?rr, ?ris) -> drm:ExtremeRiskSeverity(?x) ^ stix2:type(?x, "Extreme risk-severity"^^rdf:PlainLiteral) ^ drm:evaluates(?x, ?ris) ^ stix2:value(?x, ?ar) </pre>

7.3.7 Security Policy Rules

The reasoner will infer new knowledge that can be applied into the Dynamic Risk Framework as a Security Policy.

Table 13 Example of a Security Policy rule to automate the detection, inventory and establishment of Data-Access-Control Policy

SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)
<p>The reasoner will infer new knowledge, as a Security Policy, and it will automatically apply it to all the data-set. In this case, the example show how easy is, for our framework, to know what users have access to what data, depending on the data required access level and the level access of each user. This SWRL rule will create enriched connections and dependencies that will be very useful to estimate new type of risks for our dynamic model.</p>
<pre> drm:Users(?u) ^ drm:Data(?data) ^ drm:hasAccessLevel(?u, ?a) ^ swrlb:greaterThanOrEqual(?a, ?r) ^ drm:requiredAccessLevel(?data, ?r) -> drm:hasAccessTo(?u, ?data) </pre>

7.3.8 Risk Management Rules

The reasoner will infer new knowledge with regard to the strategy to follow depending on the Residual Risk value of each risk. It will help to implement the specific policy of each organization. In our case:

Risk are classified in the following scale (from 1 to 10):

- Extreme (8 <= extreme <=10): risk is unacceptable,
- High (6 <= high < 8): risk is undesirable,
- Medium (5 <= medium < 6): risk is tolerable,
- Low (< 5): risk is acceptable.

A Mitigation strategy is recommended for Extreme and High risks with different possibilities:

- reduce,
- to avoid,
- to share,
- to transfer the risk.

An Investigation strategy is recommended for Medium risks with different possibilities: to assume or to reduce the risk.

A Monitoring strategy is recommended for Low risks where usually risks are assumed.

Table 14 Example of a Risk Management Mitigation Strategy for Extreme Risk Severity

RiskManagement#1 Mitigation Strategy for Extreme Risk Severity
--

The reasoner will infer (and classify), that all Extreme Risk Severity's instances will have to be mitigated. It will create new instances of the class Risk Mitigation Strategy associated to each extreme risk.

```

stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ swrlx:makeOWLThing(?x,
?er) ^ drm:ExtremeRiskSeverity(?er)
->
drm:RiskMitigation(?x) ^ stix2:type(?x, "Risk Mitigation
Strategy"^^rdf:PlainLiteral) ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex)

```

7.3.9 Threat Intelligence Rules

The reasoner will infer new knowledge due to semantic Threat Intelligence rules, which will be applied into the Dynamic Risk Framework to automate and provide better Detection and Prevention services.

By using the expressiveness provided by SWRL, improved TTP detection algorithms are now feasible, like the example shown below.

Table 15 Example of a Threat Intelligence rule to automate the detection of a Security Event

ThreatIntelligence#1 Security Event Dropper Detection

The reasoner will infer new knowledge, in this case, a security event will be created. The SWRL rule represents a "TTP Detection" algorithm of a dropper behavior. Our contribution (based on an OWL version of STIX, SWRL and the reasoner) allows the creation of context-aware and behavioral rules independent of IoC (Indicators of Compromise). In this case, a malicious windows executable being dropped after a sucesion of javascript redirections is noticed. The security event will have associated all the cyberobservables and will trigger a re-assessment of the risk.

```

stix2:NetworkTraffic(?nt) ^ stix2:dstPayloadRef(?nt, ?pl) ^ stix2:Artifact(?pl) ^
stix2:mimeType(?pl, "javascript"^^rdf:PlainLiteral) ^ stix2:redirection(?pl, ?red) ^
stix2:URL(?red) ^ stix2:NetworkTraffic(?nt2) ^ stix2:dstRef(?nt2, ?red) ^
stix2:dstPayloadRef(?nt2, ?pl2) ^ stix2:srcRef(?nt2, ?sr) ^ stix2:extensions(?pl2,
"windows-pebinary-ext"^^rdf:PlainLiteral) ^ stix2:name(?pl2, ?nm) ^
swrlx:makeOWLThing(?x, ?nt2)
->
drm:SecurityEvents(?x) ^ stix2:type(?x, "security-event"^^rdf:PlainLiteral) ^
stix2:srcRef(?x, ?sr) ^ stix2:type(?x, "Dropper behavior of Malicious Windows
Executable"^^rdf:PlainLiteral) ^ stix2:dstRef(?x, ?red) ^ stix2:relatedTo(?x, ?nt2)

```

7.4. CONCLUSIONS

As a result, reasoning can be used for detection, prevention and also to response processes. At the same time, reasoning can work at all levels (Technical, Tactical and Strategic).

Here we propose some other examples of behaviors than can be handled by the Dynamic Risk Management framework:

- Security Policy changes (e.g., blocking traffic, increase password robustness, etc.).
- Patching remediation and reprioritization queues.
- Risk transfer to a third party.
- New rules or signature updates for networking devices.
- Risk Information Sharing.
- Alert notification.
- Launch honeypot and counterintelligence (deception) measures.
- Distributed topology based on software defined networks (as a joint) response (e.g., sensing, monitoring, or even taking offensive actions).
- Risk exposure escalation to management including visualization to support decision making based on historical action and responses.
- Etc.

CHAPTER 8. INFORMATION EXCHANGE PROPOSAL

8.1. INTRODUCTION

In order to better prevent or mitigate any attack, the timing to receive new knowledge is always critical. Fast sharing is important then, however it is not enough [Tounsi18] [Ring14]. The gained situational awareness of the DRM Framework, as a way to understand and make decisions at strategic and tactical levels, needs to be linked to the information sharing initiatives [Riesco19a], [Leszczyna19]. New knowledge must be applied to reduce our risk exposure along the time. Despite the great number of information sharing initiatives, all of them are at technical level. As a result, CTI is usually not synchronizing threat Intel data with the information security management system (ISMS) dynamically as suggested by authors in [Riesco19a]. If our framework is based on this feature to give dynamics, we have to address all well-known open challenges and limitations of information sharing initiatives (see Table 1) within our framework.

Users are reluctance to participate in cyber incident information sharing beyond cyber threat intelligence sharing [Tounsi18] [Ring14]. These are bad news because several organizations have the same vulnerabilities, or have in common, the same threat actor's motivation. If certain incident is shared, several entities or users could be activated to prevent a potential attack.

It is well known, that in practice, there are usually more consumers than producers of CTI Data. In some cases it is because most of the users, are usually reluctance to share sensitive CTI data [NIST16], [Riesco19a], [Tounsi18], and [Riesco19b]. In other cases it is because the lack of enough incentives [Vishik13], [Ring14]. There are few original contributors of CTI data in proportion to the consumers of that data. The lack of trust, reliability, the sensibility of the information to share as well as the type of data to be shared (IoC), are some of the factors that are discouraging.

In some cases, the users are worried about their reputation risks in case of a sensitive information leakage. The anonymization of certain data is proposed [NIST16] as a preventive solution, but it also has trade-offs. The problem is how to provide actionable intelligence, to protect another entity from a similar attack, if it is based on an anonymized version of the real information. In other cases, users are also worried to give any kind of indirect competitive advantage to their own competitors, as they cannot control the redistribution of such data. There are some initiatives to address this issue like the Traffic Light Protocol (TLP)²¹, between others.

Trust issues between users and platform providers are mostly neglected [Sauerwein17], [Ring14]. Trust is the key. No one will share relevant information to anyone, without it. Trust must exist between peers, between users and the sharing infrastructure [Leszczyna19], [deFuentes17], and also between the user and the platform administrator. The majority of platforms are closed source [Sauerwein17], which is reducing trust.

In order to have more CTI data producers joining the system, it must be trustworthy and attractive. In general, cybersecurity initiatives lack of associated business models [Vishik13] [Tosh18]. Economic incentives can be a good solution, if those are designed in a win-win approach (e.g. honest approach). We propose an innovative model to introduce such economic incentives. They are some kind of evolutionary or dynamic incentives like the use of a new CTI token.

There are usually high associated costs, but there are also misapprehended costs [Skopik16] [Ring14]. There are different ways to build an information sharing topology:

²¹ <https://www.us-cert.gov/tlp>

centralized, decentralized (as seen in Figure 49), confederated or even there can be a mixed information sharing approach (as seen in Figure 50). At the end, all of them are instances managed by an administrator, which is in charge of its operation and management. Cost are proportional to the number of instances but all of them have associated ongoing costs, which is far from our proposal.

The blockchain will provide us with several benefits, however it still has some limitations associated to the novelty of technology. It uses an EVM (Ethereum Virtual Machine) which still far from a Turing complete machine. The compilers limitations and the limitations of its coding language (e.g. solidity), are still not good enough. There are important limitations, among them, the support of non-integer values, as well as the confidentiality and privacy of specific data. As an example, a privacy declared variable (private storage), could be read by using certain hacking techniques nowadays. Other limitations are related to the storage capacity of their blocks, however there are certain possible workarounds that will be discussed later.

Despite the blockchain limitations, we will use some of its great benefits, among them, the accounting, identity management, availability and especially, the tokens, as a built-in feature to represent any digital asset that will enable new incentives for CTI sharing.

In cyber security, we have traditionally been focusing on identifying what we want to protect, and then building defenses around them along the time. When adversaries breach defenses, organizations adapt themselves to prevent those breaches to occur, and, of course, as we adapt, our adversaries adapt as well.

Cyber threat intelligence can take several forms including detailed information about the malware, indicators of compromise (IoC) or specific techniques that malicious actors use to steal information. By having this knowledge, we can update our defenses against the threats. There are many possible sources of cyber threat intelligence, such as historic incidents, open source intelligence (OSINT) [OSINTArmy12], any threat feed, ISACs (information sharing and analysis centers) or even government threats sharing programs.

The collective ability to detect and defend against malicious activity, by sharing information about adversaries and their behaviors, is paramount.

It is a joint initiative where public and private organizations (including security vendors) should work together to find better ways to create, share and use cyber threat intelligence.

European legislations like the NIS Directive [EC16] and the Cybersecurity Act nourish the creation of sectoral ISACs and PPPs within the EU. The NIS Directive, among others, separates the operators of essential services in sectors and tasks the operators to implement requirements on incident reporting which is far from a voluntary and enriched approach. It is mainly an obligation to report incidents under certain conditions to their national CSIRTs that could have associated sanctions by competent authorities, if reporting and incident handling was not properly done by the entity. Obligations and sanctions, came (usually) after a period of unsuccessful voluntary sharing. This approach is very different from OSINT initiatives but also different from any expected win-win approach, plenty of incentives, like the ones presented in our work.

The creation of sectoral ISACs at national level could further assist with the implementation of these provisions. During the transposition of such European legislation to national law, these communities could be further informed and advised by policy makers.

Information sharing between national stakeholders but even in cross country cases is one important aspect for cyber security. Knowledge on tackling cyber attacked, incident response, mitigation measures and preparatory controls are recommended to be shared between the relevant stakeholders.

On the other hand, unsuccessful voluntary sharing has several and different root causes. Several studies have been analyzing why people is often reluctant to share [NIST16] [Vishik13] [Tosh18] [Skopik16] [deFuentes17] [Ring14].

In Table 1, an inventory of the open challenges and limits of existing solutions in information sharing nowadays is seen. The table includes references from the bibliography to support each concept. All the open challenges can be grouped into the following groups:

- The lack of trust (infrastructure, admin and peers)
- The lack of incentives (business cases) provided to all roles simultaneously.
- The asymmetry between consumers and producers.
- The reliability and accuracy of CTI data.
- The lack of semantics (unambiguous data) to exchange knowledge (beyond single pieces of data).
- The effectiveness and efficiency of platforms (automation).

We propose a solution, combining the use of semantic web ontologies, STIX™ and Ethereum Blockchain, in order to cover all open challenges at the same time. As an example, we will use built-in features provided by the blockchain, like the accounting, identity management, availability and the possibility to create, and to run decentralized applications (smart contracts). In our case, the whole system will be running as a smart contract to handle all data and interactions.

We provide also incentives based on other relevant feature: the tokens. It is a great built-in feature of Ethereum, created to provide a standard way to developers for creating any digital representation of any asset. By definition, tokens are interchangeable between them. A token can represent anything from a physical object, like gold, to a native currency. It can also represent any financial instrument like stocks and bonds.

We created the first digital asset (the CTI Token) to represent Cyber Threat Intelligence data, as a digital asset. The token provides great benefits comparing to its physical representation, it can be operated automatically between smart contracts and, if it is a standard token (e.g. ERC20 standard), and it is interoperable by definition with any other standard token. As an example, you would be able to exchange stock options and CTI tokens, or even gold by their digital representation within Ethereum. This incentive will attract investors, which are often reluctant to invest in cybersecurity products due to the difficulties to estimate their return on investments (ROI) and potential exits. We will use Montecarlo simulations to define key parameters for the proposed system dynamics. We also evaluated the benefits and limitations of our model by an experimentation.

8.2. RELATED WORK

Information exchange was mainly invented to help others, by sharing our knowledge, at the same time we can improve our level of protection. It is clear, that no one is ready to fight alone against cybersecurity threats.

Wagner, C. et al. in [MISP16] presented MISP, the Malware Information Sharing Platform, which has been one of the most open sourced platforms used since 2016. Authors observed, from 20 to 40 new instances deployed each day, since its first release. MISP allows the exchange of Indicators of Compromise (IOC: hashes, hostnames, IP addresses, URL, etc.). The size of the data shared since the beginning (about 100K md5 hashes a month), has demonstrated the real demand of the end users. On the other hand, authors propose new research directions with regard to the quality of the data, as it is usually not reliable. Authors also suggest that new ways to handle such volume of (big) data in the future, might be needed. They suggest also, to work in the development of a formal model

or taxonomy to overcome the issues of their triple tag approach, as it still lacks enough expressivity.

Sauerwein, C. et al. in [Sauerwein17], analyzed 22 threat intelligence sharing platforms. Their key findings are really interesting, among them:

- There is no a common definition of threat intelligence sharing platform.
- STIX™ is the facto standard for describing threat intelligence.
- Platforms primarily focus on sharing of IoC.
- The majority of platforms are closed source.
- Most platforms focus on data collection instead of analysis.
- Trust issues between users and platform providers are mostly neglected.
- Academic and commercial interest in threat intelligence sharing increases.
- Many manual tasks make the user the bottleneck.

As a result, the authors indicate that the threat intelligence sharing lacks a consistent definition. They also indicate that threat intelligence sharing is comparable to data warehousing, and it does not provide real intelligence. We agree with authors in all these conclusions.

De Fuentes, JM. et al. in [deFuentes17], indicate that privacy is paramount to foster cooperation, particularly when insecure infrastructures are used to support sharing. They propose PRACIS, a protocol that provides privacy-preserving and aggregatable cybersecurity information sharing. PRACIS provides these properties by leveraging existing format-preserving and homomorphic encryption techniques and adapting them to the particularities of standard message formats such as STIX™. Authors anyway provide a new layer of security, when the network is untrusted.

We propose a network based on blockchain precisely to overcome the lack of trust in the network, in the administrator of the platform and in the peers. Moreover, the system is running as a smart contract (or a dApp), which is instanced in the blockchain. It will be able to receive external calls either by peers or from other smart contracts (e.g. other CTI Marketplaces). There will be no human intervention (administrator), after it is deployed in the blockchain. As a good practice, the source code is usually appearing along the smart contract instance, which give more trust to the users.

Tounsi, W. et al. in [Tounsi18], suggest that the static approach of traditional security, based on heuristic and signatures, does not match the dynamic nature of new generation of threats, that are known to be evasive, resilient and complex. Organizations need to gather and share real-time cyber threat information, as well as to transform it into threat intelligence. Authors explain why there is reluctance among organizations, to share threat intelligence data. They provide sharing strategies based on anonymity, in order to reduce the risks in case of a data leak. They also show in their work, why having a standardized representation of threat information, can improve its quality. We support such statement as well.

Leszczyna, R. et al. in [Leszczyna19], indicate that there is a need to link a fine grained situational awareness to information sharing data. They propose different topologies, ranging from a decentralized peer to peer, to a confederated topology as seen in Figure 49 and Figure 50. Authors propose a framework, to enhance current threat intelligence services by providing data at all levels. They propose to integrate all the data at the same time, especially in the context of critical infrastructure protection. They still work in the IoC domain which under our opinion, can be improved by our work.

In addition to this, we consider that their decentralized topology approach is really not a decentralized platform. It is a peer-to-peer network of different instances. We propose a pure decentralized infrastructure by using the blockchain. Our decentralized application will be running into all network nodes at the same time as seen in Figure 53.

Visik, C. et al. in [Vishik13], characterize the main reasons about why people is reluctant to share information, if there are not enough incentives for them. Authors suggest that users do not have the same knowledge about the value of artifacts offered, or simply they do not want to share their own data or incidents, as they could be giving a competitive advantage to their own competitors. Authors provide a clear understanding that introducing new economic incentives and business models in cybersecurity initiatives, as part of an economic modeling, would derive in greater societal benefits. As suggested by authors, we propose new economic incentives in our work.

Tosh, D. et al. in [Tosh18], propose a cybersecurity information exchange framework (CYBEX), based on a gaming exercise. Authors used an evolutionary analysis of the "participation cost", a driving parameter for attracting firms to join and transact cyber threat intelligence (CTI) with other firms. There are some interesting conclusions in the study like, for example, a simple incentive can be the reduction of the participation cost (negative cost vs positive cost). On the other hand, when the cost is positive, the strategy of each organization depends on how many members are still contributing. If the participation is less than the 65 percent, then the organizations stop to share. As a conclusion, any external motivation associated to economic incentives will foster the participation of peers. They used an evolutionary cost model using a gaming theory.

Skopik et al. in [Skopik16], suggest that due to networks have grown to a scale and complexity, and have reached a degree of interconnectedness, their protection can often only be guaranteed and financed as shared efforts. Consequently, authors suggest that new paradigms are required, for detecting contemporary attacks and mitigating their effects. Many attack detection tasks are still performed only by each individual organization. Authors indicate that information sharing is a crucial step, to acquiring a thorough understanding of large-scale cyber-attack situations, and is therefore seen as one of the key concepts to protect future networks. Authors also propose improvements in standardization and legal aspects, as open challenges. Our proposal will address most of these topics.

Riesco et al. [Riesco19a], propose to work in risk domain dynamically to keep risks under an acceptance level along the time. In order to do that, authors propose to leverage cyber threat intelligence for a Dynamic Risk Management (DRM) framework. One of the building blocks, is having an enhanced or advance intelligence knowledge, which can be understood by semantic reasoners [W3C-Reasoner] and humans, at the same time. The intelligence goes beyond indicators of compromise (IoC) in the format of intelligence algorithms using a combination of standards like STIX™ v2 [OasisSTIX2] formatted as OWL [W3C-OWL] ontologies [W3C-Ontology] and semantic web rule language (SWRL) [W3C-SWRL]. By using this combination of standards, authors are able to represent context behavioral patterns like the representation of techniques, tactics and procedures (TTP) [OasisSTIX-TTP]. Authors propose as future research direction, to exchange this type of algorithms beyond simple IoC in order to provide and share the real knowledge (how to detect). By using a mix of standards, authors can overcome the expressivity barriers to define TTP in the format of algorithms. It also enables the usage of semantic reasoners [W3C-Reasoner] to infer new knowledge. They propose that new incentive models still have to be developed, as a future research direction. We implemented this specific approach in our work, to demonstrate their benefits.

In [Nath16], the authors propose the use of Blockchain for fighting insurance fraud. The authors propose the creation of a marketplace to share intelligence data about fraud of users, however it lacks of empirical deployment, an incentive model, as well as calculations

related to cost. On the other hand, it is handling fraud intelligence which is different from cyber threat intelligence.

There is a very interesting approach, in [Polyswarm], which is implementing a bug bounty program using a decentralized Marketplace. It is used to connect different experts with potential customers which are willing to pay for professional cybersecurity services. They are also creating a token, named Nectar, in order to use it for exchanging value. Customers can raise a kind of auction for any interested expert or send a direct contract to check if a specific file is malicious or not. In our case, we propose to go beyond IoC or bug bounties into threat intelligence and risk intelligence exchange. Their model is still based on IoC (hashed) and is specific oriented to antivirus.

Another interesting example [Graf18], where Blockchain is being proposed as a cost effective storage of intelligence data, does provide storage and sharing features for the life-cycle state of incidents. Authors suggest a Blockchain-based solution for life-cycle management for automatic classification of cyber security documents according to their expected threat level. It is clear that blockchain provides several possibilities like we demonstrate in our work.

8.3. APPROACH AND RESULTS

We also propose to exchange more effective knowledge, that is to say, to exchange TTP rules or algorithms (e.g. detection algorithm) beyond the exchange of IoC.

The objective of the work is to provide a coherent solution for all of the open challenges in Cyber Threat Intelligence sharing, seen in Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions, at the same time. The detailed characteristics of our all-in-one solution, are presented in the Table 2.

We propose an innovative Cyber Threat Intelligence (CTI) Exchange model, based on the combination of semantic web, STIX™ and the Ethereum Blockchain. It provides new type of incentives to all roles involved. Current approaches do not provide enough incentives for producers, as a result, most of the entities involved are just consumers.

Our proposal also provides incentives to engage new type of roles like the investors. We propose different type of economic incentives:

- A new CTI token (under the ERC20 [ERC20] standard), which represents the threat and risk intelligence data, as a digital asset. The creation of a new token will facilitate the interoperability with any other standard token within Ethereum blockchain.
- Crypto currencies (cash in Ether currency) to support the use of taxes in the exchange of knowledge (pay per use).

In our work, we also propose to share enhanced knowledge in the format of semantic algorithms or rules (beyond IoC). It uses semantic variables in an OWL [W3C-OWL] version of STIX™ v2 format, to support complex representation with more expressivity. Rules are provided in the format of semantic rule language rules (SWRL [W3C-SWRL]), as proposed by Riesco et al. in [Riesco19a] [Riesco19b]. The use of ontologies, by definition, enables the interoperability and unambiguity of concepts. At the same time, it allows the usage of semantic reasoners to infer new knowledge, which might help to bring the needed automation keeping the data consistency (reliability). It will be also more effective with regard to the number of data to be shared, because the value of the algorithm is greater than just a specific IoC. It is also less ephemeral [Bianco14].

We run different experiments and simulations to demonstrate benefits and limitations of our model. As a result, the proposed model brings several benefits to the state of play, among them:

- **Accountability and trusted sources:** who (private key) really shared / consumed / invested what and when.
- **Availability and low cost:** creating a decentralized marketplace application in Ethereum Blockchain with all the benefits of blockchain decentralization supported by several nodes, without the need of specific investments in an owned infrastructure.
- **Secure:** by using specific design-patterns and secure coding together with specific security enhancements provided by the Ethereum Blockchain. It is by definition a trusted network. It provides trust even between untrusted parties (e.g. crypto currency transaction).
- **Invest ready and attractive** by a **win-win** approach. It provides economic incentives to several roles simultaneously: creating a token named "CTI token" (crypto-token), which is interesting for investors and users. It will provides incentives. We made Montecarlo simulations to better select key parameters of our system.
- **Semantic expressiveness, effectiveness and efficiency by using a combination of standards:** the use of a combination of standards enables the reasoners to better contribute to the automation and decision making processes. The exchange of algorithms beyond IoC will provide better detection and prevention capabilities to all stakeholders involved. At the same time, it will contribute for the exchange of real knowledge, its expressivity, its consistency and its reliability. We propose the exchange of the analysis of the data (knowledge or advanced intelligence) instead of the exchange of the data itself (data).

8.4. ROLES

The main roles of the CTI Marketplace will be the following (all of them have associated functions for their type of interactions):

- **CTI Data producer:** an individual or a group which might be using a single Ethereum address to contribute or share (write) CTI data rules in the CTI Marketplace Smart contract.
- **CTI Data consumer:** an individual or a group which might be using a single Ethereum address to consume (read) CTI data rules from the CTI Marketplace Smart contract.
- **Investor:** an individual or a group which might be using a single Ethereum address to invest into the CTI token.
- **Donor:** an individual or a group which might be using a single Ethereum address to donate ether to support the CTI Marketplace.
- **Owner:** an individual or a group which might be using a single Ethereum address to launch each CTI Marketplace instance. In this case, the owner are the authors of this work.

Figure 69 Roles overlapping

Above roles can be combined under the same single Ethereum address. As an example, an organization could contribute with its own CTI data rules at the same time it consumes data from other organizations. It can also become an investor. As seen in Figure 69, we propose a paradigm shift, where we expect a significant increment in the number of active CTI Data producers. Most of them are today just consumers. Becoming CTI Producers, they will be able to also become investors. They will have special investment conditions thanks to the introduction of the Tax per transaction and the CTI Token.

8.5. MARKET GROWTH: MONTECARLO SIMULATION OF NEW INCENTIVES

In order to better design the system, we have made theoretical calculations as seen in Figure 70 and Figure 71. We used multiple Montecarlo simulations to forecast the potential market growth, in case of introducing our new incentive model. It helped us to better design and select which are the key variables of the model. In addition to this, it enabled us to evaluate it versus current legacy information sharing platforms (OSINT, Feeders and ISAC), taking into consideration the specificities of each role.

Figure 70 Montecarlo simulation of benefits with an evolutionary tax and CTI token value (p_c)

As a preliminary condition, we used the CTI Data volume of the top 20 types of data which are described in MISP [MISP16], as a baseline for all systems. In our case, as it is oriented to go beyond the exchange of IoC, it will have additional CTI Data, which are in the format of algorithms. At the same time, our system will be used to handle data from 3 different levels (technical, tactical or strategic), as seen in Figure 52. We then estimated the additional CTI data by using the inverse of the normal cumulative distribution. We adjusted its parameters with a 15 percent probability, a mean of 33 percent of the baseline volume of data and a standard deviation of 10,000.

Due to the fact that the investment by CTI producers is optional, we will consider that the volume of optional investments are an inverse of a lognormal distribution applied to the 50 percent of total data in the system. We are also considering that all the data is read at least a hundred times. We will consider the worst case scenario to read all the data (by CTI Data consumers), that is to say, each data is read in a different transaction (each transaction has an associated fee). We then applied the same condition for writing: each single piece of data uploaded to the system, will be uploaded in a single transaction.

We introduced a tax for CTI Data consumers each time they execute a read transaction, as they get important value back. For simplicity, we fixed the 50 percent of such tax will go for the CTI producers wallets and the 10 percent to the owner's wallet of the system. The number of CTI tokens provided to investors will be different depending on the role. To simplify our calculations, we used the worst case scenario in our Montecarlo simulations, that is, we used the ongoing incentive factor of 20 for CTI data producers and 10 for a standard investor. The smart contract will keep the remaining cash (the remaining 40 percent).

We will define the dynamic or evolutionary benefit B at time t of an entity x as seen in Equation 8.

Equation 8 Evolutionary benefit "B"

$$B_t(x, p_c) = I_t(x, p_c) - C_t(x, p_c)$$

where:

- p_c is the token prize at time t ,
- I_t is the income (gross benefit without cost) at time t ,
- C_t is the cost at time t .

The gross benefit or total income I_t of an Entity x , can be defined as seen in Equation 9:

Equation 9 Gross benefit or total income "I" of an entity x

$$I_t(x, p_c) = IC_t(x, p_c) + ITK_t(x, p_c)$$

where:

- IC_t is the income made by cash at time t ,
- ITK_t is the income made by the CTI token value at time t .

In our proposal, CTI Data producers and the owner, will receive cash IC_t , due to the introduction of taxes into each CTI read operation. At the same time, these payments will also represent variable costs VC_t for CTI Data consumers. CTI Data producers will also receive tokens ITK_t in case they decide to invest when uploading CTI Data to the system. We have an evolutionary token value p_c , depending on the balance within the smart contract. The smart contract will sum the cash of all the investments (made either by investors or CTI Data producers) to the 40 percent received from the applied taxes.

On the other hand, the total cost C_t of an entity x , can be defined as seen in Equation 10:

Equation 10 Total cost "C" of an entity x

$$C_t(x, p_c) = FC_t(x, p_c) + VC_t(x, p_c)$$

where:

- FC_t is the fixed cost at time t ,
- VC_t is the variable cost at time t .

To simplify our calculations, we propose zero fixed costs ($FC_t = 0$), due to the use of a decentralized infrastructure. Variable cost VC_t , will be associated to any transaction.

We have two types of variable costs: blockchain network fees (gas) and taxes. The gas used to write (store) CTI rules is much higher than the gas used to read (query) those rules. In order to calculate these type of variable costs, we implemented, deployed and evaluated a draft smart contract.

Equation 8, can then be divided into different equations as follows:

Equation 11 Evolutionary Benefit as a combination of cash and token prize

$$B_t(x, p_c) = BC_t(x, p_c) + BTK_t(x, p_c)$$

$$BC_t(x, p_c) = IC_t(x, p_c) - CC_t(x, p_c)$$

$$BTK_t(x, p_c) = ITK_t(x, p_c) - CTK_t(x, p_c)$$

$$I_t = IC_t(x, p_c) + ITK_t(x, p_c)$$

$$C_t = CC_t(x, p_c) + CTK_t(x, p_c)$$

where:

- BC_t is the cash benefit at time t ,
- BTK_t is the token benefit at time t ,
- IC_t is the cash income (cash gross benefit) at time t ,
- ITK_t is the cash income (token gross benefit) at time t ,
- CC_t is the cost paid in cash at time t ,
- CTK_t is the cost paid in tokens at time t .

We made a Montecarlo simulation (seen in Figure 70) to define the minimum tax value of our system equivalent to a breakeven calculation. In this case we look for a positive cash benefit $BC_t(x, p_c) > 0$ of the CTI Data producers.

Figure 71 Montecarlo simulation to compare our new model versus current models.

In order to study how our evolutionary incentives (especially tokens through investments), are also contributing to the dynamics of the system, we also used a Montecarlo simulation, as seen in Figure 71. In the figure, we see a comparison of our model versus other current approaches (OSINT, Feeders, ISAC), including benefits per type of role. We marked with discontinuous lines the OSINT, the Feeder and the ISAC values, as well as one selected from our model (the "New model 10 x Tax"). Our evolutionary model is anyway represented by different curves which are using different tax values as parameters. There are two types:

- "New model [1,5,10] x Tax": to represent the evolutionary benefit B at time t of a entity x (equivalent to $B_t(x, p_c)$),
- "New model [1,5,10] x Tax (only cash)": to represent the evolutionary cash benefit BC at time t of an entity x (equivalent to $BC_t(x, p_c)$).

As a result, the CTI token is especially contributing to the benefits of the CTI Data producers ($B_t(x, p_c)$ vs $BC_t(x, p_c)$). Furthermore, it is also contributing to the engagement of new key roles of any business: the investors. They will be able to invest in Cybersecurity CTI Data sharing, easily. Our proposal, then, provides new ways of getting value back from the CTI Data to several roles simultaneously.

8.6. CTI TOKEN

Despite the great number of ICO²² (initial coin offering), there are still several open discussions about the benefits and the economic model of tokens [Ravsan18]. Our proposal slightly differs from a typical ICO. In our proposal, we propose a CTI token to represent the value of the CTI Data, as its digital asset representation. We also propose a customized version of an ongoing ICO, because the very first time a new CTI producer joins the network with new CTI Data, it has a special offer to invest in CTI tokens. After that, they are still getting more tokens than standard investors, as a way to keep attractive incentives associated to the sharing of new CTI data.

As described by the author in [Ravsan18], the pricing service in a fixed token rate will correctly capture the proportionate relationship between demand and price, the greater the demand for the service, the more people will buy tokens, the price of tokens will increase, and so will the cash equivalent of the service; and vice versa. The network has honest but also malicious miners (as seen in Figure 53), each having different motivations. Malicious miners, want token price to depreciate, while honest miners and investors want token price to increase. The incentives should be modeled in such a way that no player will want to deviate from the honest conditional equilibrium. The author proposes a version of the Cobb-Douglas utility function to model the utility of a service, based on the quality but also the popularity of the service. In our case, we propose new parameters for the utility function to model our service as seen in Equation 12.

Equation 12 Cobb-Douglas utility function applied to our model

$$U(tp_t, rd_t) = (1 + E[tp_t]^\alpha) \cdot E[rd_t]^\beta$$

where $E[tp]$ is the expected volume of trusted experts contributing (trusted CTI producers) in the system and $E[rd]$ is the expected reliability and accuracy of the data (CTI Data).

Note that even when there is zero number of trusted experts ($E[tp] = 0; E[rd] > 0$) the customer still receives a non-zero utility from the service. Conversely, when the service is very popular among trusted producers and has high demand but zero reliable data ($E[tp] > 0; E[rd] = 0$) the utility from the service is zero despite its popularity.

Our proposal is similar like the one proposed by the author in [Ravsan18] about Investors. They are people who buy our tokens for the CTI market value and hold them in expectation of value appreciation. They are not really interested in the Threat Intelligence Data but its market growth (as seen in Figure 70 and Figure 71). We will also assume that there are N identical investors with equal wealth level w who are using a version of Markowitz's mean-variance formula (see Equation 13) to choose between investing in stocks and our CTI token. We will assume that all investors are risk neutral. Because the more investors a system has, the more cash we will have in our system.

We also used the inverse of the lognormal cumulative distribution for our Montecarlo simulations to simulate ongoing investments.

Equation 13 A version of Markowitz's mean-variance formula to choose between investing in stocks and our CTI token

$$\alpha_t = \frac{\mu_c - \mu_s}{\gamma \cdot \sigma_c^2}$$

²² <https://www.coingecko.com/en/ico>

where μ_c and μ_s are coin and stock returns respectively, and γ and σ_c are risk-aversion coefficient and coin volatility respectively.

Malicious miners are nodes who buy tokens and vote at quorum to force their version of blockchain. To simplify calculations, we do not consider worst types of malicious miners, which are those that fight to create their own blockchain. We are considering only those who want to maximize their profits within the system. That is to say, we consider those nodes that will try to act as major investors when the prize of the token p_c is as lowest as possible.

We decided to simplify the token prize equation suggested by Ciaian et al. in [Pavel16]. In our simulations we will use the Equation 14. Also for simplification, we will not limit the number of tokens NT as it might happen with proof of stake setups although it is a feasible alternative.

Equation 14 Token prize equation

$$p_c = \eta \cdot \frac{Bal}{\mu_s \cdot NT}$$

where:

- Bal is the balance of the smart contract (cash),
- $\mu_s = 1$ which is the opportunity interest rate (stock returns),
- NT is the volume of the token supply (number of minted tokens),
- $\eta = 1$ which is an exogenous stabilization parameter related to the token attractiveness.

Despite the simplification for our calculations, we expect that our model will improve the dynamics of the threat intelligence sharing as seen in Figure 71.

8.7. DECENTRALIZED ARCHITECTURE

Ethereum is a peer-to-peer network of nodes running Ethereum Virtual Machines (EVM) that stores a copy of all the data and code on the blockchain. Ethereum network is then a blockchain with dual use, it supports the integrity, reliability and availability of the transactions of Ether (ETH) crypto currency as well as it supports the integrity, reliability and availability of the execution of Smart contracts.

Decentralized applications (DApps) can be deployed as running code in all the Ethereum nodes at the same time under the name of Smart contracts. Ethereum nodes use, as Bitcoin (BTC), consensus algorithms to validate transactions however in Ethereum, transactions can go beyond money transfers. They can use transaction payloads to enable the possibility to execute applications. Transactions are now resolving read and write access to a running application which is running in all nodes at the same time, decentralized. Validation is then confirming that executions in certain DApps are validated by a consensus algorithm likewise that the same validation algorithms are validating money transfer.

All the network is optimized in such a way that each instruction or assembly call has associated a certain amount of gas, which is the equivalent to the use of resources needed (e.g. CPU) for each instruction. The idea is to have the most efficient running code, if not, a DApp could not finish each execution if it runs out of gas.

Our proposal to solve above problems and take suggested opportunities is based on the use of Ethereum Blockchain as a decentralized infrastructure to run our decentralized application (DApp) created for the purpose of this work. The DApp consists on a Smart contract that will handle the storage and exchange of CTI data in the form of a Marketplace

of rulesets (e.g. TTP detection algorithms) as described by authors in [Riesco19a] [Riesco19b].

An online marketplace is a type of e-commerce site where product or service information is provided by multiple third parties, whereas transactions are processed by the marketplace operator. Online marketplaces are the primary type of multichannel e-commerce and can be a way to streamline the production process. In our case, the product itself is the CTI Data ruleset (threat and risk intelligence algorithms).

The Smart contract provides new incentives to all type of users. Between them, a new ERC20 standard token is created, named CTI token. It represents a digital asset with special value and is related to the value of the CTI data ruleset. In addition to this, the Smart contract is coded in such a way that economic incentives are amplified to CTI Data producers. New roles are created as well, these are investors and donors.

Figure 72 Architecture design options: 1. Client-Blockchain (Serverless), 2. Server-Blockchain (Clientless), 3. Client-Blockchain-Server

In order to create the Marketplace within the Ethereum Blockchain there are different options and topologies as seen in Figure 72:

- Client-Blockchain (serverless)
- Server-Blockchain (clientless)
- Client-Blockchain-Server

With regard to the first option **Client-Blockchain (serverless)**, the client interacts directly with the blockchain. In order to do that, the client needs to connect to an Ethereum node via a web3 provider that will handle the connection. There are lightweight clients like Metamask²³ or Mist with their nodes.

If there are no clients available, a public new "Geth or Parity" node would have to be deployed as a gateway disabling personal API to manage accounts for security reasons. As an alternative, Infura²⁴ nodes are available for free as well. Transactions via Metamask

²³ <https://metamask.io/>

²⁴ <https://infura.io/>

will ask for approval immediately. If not using Metamask our application should be monitoring all smart contract's events to update the end user interface.

With regard to the second option **Server-Blockchain (clientless)**, the solution is to install a local Ethereum node in order to use its RPC JSON interface from the server application to execute the operations in the Blockchain. Its RPC JSON interface should not be accessible outside our application, otherwise anyone could access and steal our funds.

If we use public nodes, we could make transaction signatures offline. If nodes are not trusted to always deliver transactions to the network, we could mitigate this risk by delivering to more than one node at the same time.

Figure 73 Client-Blockchain-Server design option which might need enhanced synchronization.

Last option is the **Client-Blockchain-Server**, as seen in Figure 72, it requires more coordination between the different interactions from the client or the server within the Blockchain network. Observations are based on event listeners where an efficient management might be implemented. As the example presented, clients will be interested to listen only to their related (filtered) events however, the server will monitor all smart contracts related to the application. Messages from client to server are only recommended as informative, as it is better to wait until the blockchain confirmations. By using servers, our application will have the possibility to use several off chain backend connections to enriched applications (e.g. email, external storage, etc.).

We also take the opportunity to inherit an ERC20 token standard implementation to create a new token named CTI token. By that, we will contribute with more incentives around this digital asset. Investors, like they do in an ICO (Initial Coin offering) are then able to invest in CTI tokens.

For the purpose of this work, the **Option1: Client-Blockchain (serverless)** topology was selected due to simplicity and efficiency. Metamask is also used to connect to Ethereum Network as an injected web3 provider. At the same time it allows us to deploy a complete implementation of the CTI Marketplace to test our work and all related incentives per each role defined.

8.8. CONCLUSIONS

In this chapter, we motivate and detail the specific proposal of an enhanced version of the Information Exchange Peer-to-Peer Network, based on the Ethereum Blockchain.

A paradigm shift is proposed in order to improve the contribution of reliable information, by fostering new CTI Data producers coming into scene. There are several benefits, among them, the use of new incentives, a new business model and the sharing of intelligence (in the format of algorithms) beyond the exchange of ephemeral IoC [Riesco19b].

CHAPTER 9. VALIDATION

9.1. INTRODUCTION

In order to validate our proposal, we have created different use cases:

- **Use case 1 - Dynamic Risk Management While under Attack:** We implemented in our DRM Framework, an instance of the ISMS system for Risk Assessment and Management (RA/RM) of a leading organization (described in 9.2). We also instanced its CTI Data model in our DRM Framework in order to have the promised dynamic behavior. For that, data was transformed into OWL (its semantic version). We then selected a complex TTP of a well-known attack of type “Watering hole”, and introduced intelligence behavioral pattern detection (in the format of SWRL) to the reasoner. After executing the model to protect the organization from this complex attack, the system provides several benefits at all levels (Technical, Tactic and Strategic) [Riesco19a].
- **Use case 2 - Exchange of Threat and Risk Intelligence:** We implemented the proposed decentralized Intelligence Information Sharing module based on the Ethereum Blockchain to overcome open challenges and limitations of Table 1. Due to the fact that our DRM Framework needs as much reliable information from peers on time in order to have a better context of the risk exposure at all level, a simulation of how the paradigm shift provided are contributing with new incentives, was tested. In addition, all the benefits (e.g. exchange Tactic rules or detection algorithms beyond IoC, economic incentives, etc.) but also the limitations are identified in our prototype [Riesco19b].

9.2. ORGANIZATION UNDER STUDY: A NATIONAL CSIRT

Here we introduce the organization selected to test our work. We selected a National CSIRT (Computer Security Incident Response Team) as we consider it an international reference for its corporate maturity (risk management framework certified under ISO and under a National Cybersecurity Scheme) and the maturity of their CSIRT services (incident prevention and incident response services based on cyber threat intelligence services).

This public organization is responsible to provide nationwide preventive and incident response services affecting citizens and private sector audience, including Critical Infrastructures.

At the same time, this organization has its own SOC (Security Operation Center) to protect its corporate IT infrastructure. As described above, it is certified under ISO2700X Information Security Management standards but also under a new National Risk Management Standard which is a must for public entities like our selected organization.

Several attacks are targeting our organization (e.g., hacktivism, state-sponsored, etc.) dynamically. Security events are received and handled by its SOC however there is still not connection between those threats and its risk management framework.

9.2.1 Cyber Threat Intelligence Model

Given the different contributions and their applicability to a specific case on an entity of the CSIRT type. The real architecture of the cyber-threat intelligence model (CTI or Cyber Threat Intelligence) of this entity, on which the model of Analysis and Management of Dynamic Risks previously disconnected, would be applied, is introduced schematically to facilitate the understanding of the reader. The different components are described below:

- **Standards** (common feature of the data that should be standard to facilitate their interoperability): In this doctoral thesis will evolve the STIX™ standard to an ontology, as the original contribution of the present doctoral thesis.
- **Sensors** (sources of information): sources of information on threats. These, described in section 4.4.3, may be of various types and of two types:
 - Internal sensors.
 - External sensors.
- **Central Data Storage** (Big Data): storage (persistence) and processing of large volumes of cyber-intelligence information. At this point algorithms and rules will be applied, in our case SWRL rules and the data will use the STIX™ OWL v2.0 ontology (original contributions of the present doctoral thesis).
- **Knowledge database:** necessary historical of incidents for the calculation, among others, of the conditioned probability of the occurrence of certain threats based on the Bayes Theorem. In addition to this historical incident, could have other information such as historical decision making, or even the history of certain key indicators (KPI) issue that would greatly facilitate the processes of measurement, evaluation and continuous improvement. (Corresponds to module 4.4.4 and corresponds to another of the original contributions of this doctoral thesis)
- **Rules and algorithms + Analytics/Machine Learning.** They will be used primarily for:
 - triage processes (priority),
 - enrichment processes (generation of new intelligence directly by fixed rules or by inference), as well as for
 - incident detection and alerting processes.

In the present doctoral thesis, ontologies, SWRL rules and semantic reasoning will be used however, this will suppose a clear evolution of the current system of the organization whose current rules do not allow even more than simple rules, fixed, predefined and with logic of basic business, based on SIEM correlation algorithms (event management) without the possibility of semantic reasoning and inference. In addition to evolving the capacity of the system on the understanding of data and the ability to generate new knowledge by inference / reasoning, we will simultaneously connect and evolve two systems disconnected up to now in our organization, such as risk analysis and management domains and the domain of intelligence about cyber threats. In this way we will evolve the algorithms and rules of the different types (triage, enrichment and detection / alerts) since they will be able to expand their business logic through relations between both domains and these will not have to be fixed in detail in detail as before (SIEM) but may be the result of inference by semantic reasoning and SWRL rules I feel one of the main original contributions of the present doctoral thesis. For example, we can have an enrichment process in a domain such as risk analysis and management, and this domain may be linked to intelligence events about cyber threats. Also, following the example, we can have detection processes and alerts dynamically in the domain of risk analysis and management, also due to events in the domain of analysis and risk management as well as events of their linkage, through ontologies, with the Intelligence domain over cyber threats. Last but not least, the SWRL rules will give the possibility (together with Drools) to implement rules through a simple, conceptual syntax, and this can be done by knowledgeable people of the domains but without the need to have advanced technical knowledge, this being one of the main barriers in the capacities of the various processes (triage, enrichment and detection) of any type of organization, including ours despite its high level of qualification (e.g. the

technical profiles of Incident management cybersecurity, they do not have to be experts in SIEM rules and even less, in SCALA programming (this being the programming language used for this type of rules in the Big Data system)

- **Alerts management module.** The alerts will be part of the Algorithms and Rules module discussed above, although this module will be necessary for the management and maintenance of the Different alerts (operation and maintenance of the same, will include corrective and evolutionary maintenance).
- **Indicators/KPI y graph oriented** (visualization and simulation system discussed later in section 4.4.5 of business intelligence). Its main function will be to provide intelligence on cyber-threats and real-time knowledge of the level of risk exposure of our organization. The knowledge at the operational level will be offered directly to the technicians offering direct access to the data (web access and API) as is the case of our organization, although the information should be better structured and classified in more comprehensible concepts for the tactical levels and strategic, this being the main function of this module because this type of staff, with greater responsibility does not require or have advanced technical knowledge but has more business knowledge and have information in real time or close to real (NRT - Near Real Time) is basic for this type of profile.
- **Exchange:** corresponds to the bidirectional information exchange process and the anonymization process (TLP). We decided to include Exchange of information inside this doctoral thesis, as it is considered one of the main building blocks to trigger a risk level change along the time. As a bidirectional exchange within external partners. Here we also propose the use of ontologies (such as the evolution made by creating the OWL ontology of STIX™ in version 2.0) to go beyond the exchange of information based on IoC into the exchange of threat and risk intelligence in the format of algorithms (behavioral patterns). In addition to this we propose a new incentive model based on blockchain and a new crypto token named CTI.
- **TLP** (traffic light protocol): we propose to use this standard to simplify the access control to the specific information.
- **Monitoring module** (orchestration). Although outside the scope of the present doctoral thesis, the use of SWRL rules could be used associated with this module as a possible future line and by similarity to the processes of detection and triage discussed above that are addressed.

The different operators and stakeholders will be contributors and consumers of the information and will do so at 3 levels (operational (CERT), tactical (Responsible) and strategic (Managers)).

Figure 74 Intelligence Model of the leading organization (case study)

Here we describe the current CTI model of the selected organization based on layers (bottom-up)

With regard to the Data Layer, the organization has different international agreements with partners as well as different providers to gather valuable threat information affecting their customers.

Several IOC (Indicators of Compromise) are received from external sources with regard to different threats. Some examples are: malicious URL, domain, subdomains, IP addresses, bots, botnet servers, defacements, vulnerable assets, vulnerabilities, malware samples, hashes, SPAM emails, phishing campaigns.

The organization has its own threat data coming from their advanced sensors: honeypots (low, medium but also high interaction ones), Dark/Deep web monitoring systems, so on and so forth. Threat data information (internal and external) is parsed into its own data model through a SIEM (Security Information and Event Management) and it is persisted on a Big Data architecture (Apache Spark + Hadoop). SIEM vendor provides a restricted proprietary data model so there is a need to persist a parsed version into the Big Data under STIX™ format.

Each organization will have a different CTI Data set depending on its own interests. CTI Data will range from their own collection of data to external interested data from business partners/feeds. As an example, we consider a CTI Data set for our work that includes different types of data like:

- Incident handling-attacks received.
- Intelligence analyst data.
- Other internal data like strange patterns of traffic of specific users.
- External CTI data provided by third parties (e.g., supply chain, business partners, other CSIRT, equivalent entities within the sector, etc.) but in the same format (STIX™).
- Any data of any STIX™ object / concept.

The idea is to keep a coherent data set in a semantic version of STIX™ to be able to run SWRL rules. We are really open to any type of possibilities but limited to that taxonomy.

The Business Logic Layer is formed by algorithms and rules. Today, logic is widespread between sensors, security event information management (SIEM), as well as in the Big Data. It has mainly three problems: isolation (data not shared, API not available, proprietary language), limitations (static, predefined and not touring complete logic) and complexity [Zhang17] (SCALA language for Big Data, vendor specific for SIEM).

Services / Application Layer is where actions (e.g., alerts, notifications, etc.) are taken to cover all potential use cases (Incident Prevention, Incident Detection Notification, Incident Handling, Information sharing, KPI updates being triggered, etc.). It uses the business logic and data layers. As an example, the SIEM is used to open incident tickets automatically into their Incident Response ticketing system for its NRT/RT capability.

For example, the Information Sharing Application helps partners and stakeholders of our selected organization. TLP (traffic light protocol) tag is used to keep confidential information safe, it is equivalent to an ACL (Access Control List). TLP avoids cybersecurity information to be shared without consent at the same time we benefit the whole cybersecurity community when the info can be shared with a broader audience. Different levels will apply depending on the confidentiality of the information.

In case of the Visualization Layer, the organization is currently migrating into a Business Intelligence platform to guarantee effective visualization to the three different levels (operation, tactical and strategic levels). Nowadays visualization is provided only to operational level via web access to data storage (SIEM logger and Big Data). Tactical and Strategic levels have been approached partially, still without business intelligence, flexibility to deploy new algorithms and interactive drill down options to explore the data. This is something that we will have by definition by using ontologies (e.g., graphs). As data model is not completely standard yet (still using some vendor specific data model), tactical and strategic levels could experience some problems when new indicators need to be provided. In this case, there are situations where data meaning is not clear enough (e.g., compromised resources versus incidents). By using ontologies [W3C-Ontology] we will address these inconsistencies.

9.2.2 Risk Assessment and Management Model

Given the different contributions and their applicability to a case concrete on an entity of the CSIRT type. The actual architecture of the entity's risk analysis and management model is introduced schematically, to facilitate the reader's understanding, and is based on ISO27001 and ISO31000.

The risk analysis and management process is static, with an annual periodicity of review and continuous improvement based on the Deming cycle (PDCA - Plan Do Check Act).

The process, although supported on these standards, has undergone several interesting developments and summarized in a flowchart of steps as shown below:

Figure 75. Flowchart of Analysis and Risk Management phases of the entity under study (CSIRT).

The Different phases are defined below for a better understanding by the reader:

- **Phase 1: Communication and consultation.** It corresponds to the communication process with those responsible for the services object of scope of our analysis and risk management. In this phase, the necessary information will be taken through a questionnaire made to the RO (risk owner). Among the type of information, the RO must identify and analyze the expectations of the interested parties (internal and external) about the service, process, tool or project, addressing issues related to the risk in itself, its causes, its consequences (if they know each other), and the measures to be taken to treat it.
- **Phase 2: Establishment of the Context.** In this phase, the RO must establish the context in which the service, process, tool or project is defined, developed and deployed, identifying the external and internal objectives and parameters to be taken into account in risk management. Next, it will be evaluated according to the security dimensions, the parameters of the BIA (business impact assessment), the strategy and the quality.
- **Phase 3: Risk Assessment.** In this phase, the RO will indicate the applicability of the risks and will evaluate them taking into account the variables of impact and probability (scale of 1 to 5) to obtain the potential risk. The ISO 27002 controls are applied to the potential risk according to the current maturity levels, obtaining the current risk. Note that this phase will be aligned with our "Analysis" risk module and will be "dynamic", not static as in this case. In addition, subjectivities will be reduced by the introduction of Bayes' theorem for the calculation of said probability.
- **Phase 4: Treatment of Risks.** In this phase, given the current risk, the risk strategy will be applied to obtain the expected risk that will be obtained once the corresponding action plan has been implemented. Note that this phase will be aligned with our module "Management and Treatment" of risks and will also be "dynamic" unlike the current process, subjective and static (although revisable annually).
- **Phase 5: Monitoring and review.** The risk management process is subject to regular review and follow-up at this stage, at least once a year and whenever there are exceptional circumstances that require it. In our case, being dynamic, this phase will evolve to an NRT system (Near Real Time - close to real time).

Likewise, depending on the values obtained annually, the risks will be classified into 4 levels (according to a scale from 1 - lowest risk to 10 - highest risk):

- Extreme risk: $8 \leq \text{extreme} \leq 10$
- High risk: $6 \leq \text{high risk} < 8$
- Medium risk: $5 \leq \text{medium risk} < 6$
- Low risk: $1 \leq \text{low risk} < 5$

The different combinations of probability and impact are shown below:

			IMPACTO				
			Muy leve	Leve	Moderado	Grave	Crítico
			1	2	3	4	5
PROBABILIDAD	Muy baja (una vez cada 5 años)	1	2	3	4	5	6
	Baja (una vez cada 3 años)	2	3	4	5	6	7
	Posible (una vez cada 2 años)	3	4	5	6	7	8
	Alta (una vez al año)	4	5	6	7	8	9
	Muy alta (varias veces al año)	5	6	7	8	9	10

Figure 76. ISMS Risk categories of the entity under study

Risk Assessment, has limited scope as only a few (but business essential) services or projects are in the scope of the Risk Assessment process. Risk Owners usually give the service or project context by filling down a form and/or an interview. The scope includes the context that is to say, assets and dependencies either internal or external. The context also includes the compliance, legal, politics factors. Risk Owners have to evaluate each Service by different dimensions:

- Information Security: Confidentiality, Integrity, Accounting, Authenticity, Availability.
- Business Impact Assessment (Recovery Time Objective, Recovery Point Objective).
- Strategy.
- Satisfaction.

Dependencies are usually manually defined by risk owners which is considered a big limitation, as they are usually not experts on their corporate network topology to clearly define IT dependencies between assets (better known by IT staff). They evaluate the importance of each service with regard to different dimensions. That valuation is then considering all dependencies. As a result, in a top-down approach, all assets inherit the valuation from above. An asset which is providing support to two different services in scope, will have inheritance from both services. Services in risk assessment scope will depend on Data, which at the same time will depend on SW/HW and later will depend on their Users who will be using those assets.

Threat Inventory is taking place again by hand, identifying most relevant threats that can threaten our service / project. Those threats can harm the image of our organization sometimes, so those risks have also to be manually entered. After that, safeguards have to be manually identified as well. They could partially mitigate our risks depending on the threat and the countermeasure.

Risk should be a function of probability and impact as suggested by most of the methodologies and standards today.

The organization under study decided to follow the following formula where risk is a combination of probability and impact, reduced by availability of certain countermeasures:

Equation 15 Risk Equation of our Organization under study

$$R_i = P_i + I_i - C1_i - C2_i$$

where:

- R_i is the Residual Risk of threat i ,
- $C1_i$ is the decreasing value of the Impact (Severity) I or Probability P of threat i due to a Countermeasure 1 (specific control established in the organization),
- $C2_i$ is the decreasing value of the Impact (Severity) I or Probability P of threat i due to a new proposed Countermeasure 2 (specific new control to be established),
- P_i is the Probability P of threat i ,
- I_i is the Impact (Severity) I of threat i when it is materialized.

Threats could be of different types (Strategic, Compliance, Physical Security, IT Security, Quality and Process Management, Others). Because the assignment of each safeguard to each threat is not really done in the organization today (it had to be done by hand and high level of granularity), the formula is simplified in the organization under study by using average values. Depending on the new risk value after all new countermeasures are set-up, Risk are classified in the following scale (from 1 to 10):

- Extreme (8-10): risk is unacceptable,
- High (6-7): risk is undesirable,
- Medium (5): risk is tolerable,
- Low (1-2-3-4): risk is acceptable.

A Mitigation strategy is recommended for Extreme and High risks with different possibilities:

- to reduce,
- to avoid,
- to share,
- to transfer the risk.

An Investigation strategy is recommended for Medium risks with different possibilities: to assume or to reduce the risk.

A Monitoring strategy is recommended for Low risks where usually risks are assumed.

In addition to this, all strategies should take into account potential actions of different impact in different dimensions:

- Image and reputation.
- Compliance.
- Security.
- Budget and costs.
- Operations (Ops).

Other special response to risk strategies can also be considered where:

- Attack,
- Deception,
- Deterrent,
- Information sharing,
- Awareness,

are just some examples.

Summarizing, being the selected organization an International recognized CSIRT which is providing nationwide advanced cybersecurity services, it has its own risk as organization. Still CTI and RA/RM processes are isolated between themselves, the reason behind after some interviews is that they are supposed to belong to different domains and they are managed by different teams inside the organization (as usual in most organizations).

Data, business logic, services / applications and visualization from each of these domains are completely separated. RA/RM are using static tools (Excel, etc.) different from CTI tools but they are also using different taxonomies, level access and a different timing, when they shouldn't be isolated.

9.3. USE CASES

Here we describe the different use cases used to validate our work.

9.3.1 Use Case 1: Dynamic Risk Management While Under Attack

We selected a real attack from August 2017 that affected a popular Middle East news site. The attacks of type Watering hole are usually very sophisticated attacks as well as they are very difficult to detect.

This is the main reason for us to select this specific attack in this work. The difficulty to detect this type of attacks if our detection techniques are only based on knowns IOC (indicators of compromise), force us to deploy new detection techniques based on behavioral patterns to detect and fight the still unknown and dangerous attacks. Our proposal is based on standards and it adds so much expressiveness by using ontologies, so it will allow us to create any pattern detection rule in SWRL beyond watering hole attacks. It will help us to detect any other emerging unknown attack based on patterns. At the same time, all risk information will be updated dynamically after a dynamic re-assessment. It will be presented accordingly to tactical and strategic management but it will also may include any type of automatic responses. Here we propose a pseudo-automatic response, at least for offensive ones, as we consider that a human should take the ultimate decision in this type of attacks.

A Watering hole attack is a computer attack strategy, in which the victim is a particular group (organization, industry, or region). In this attack, the attacker guesses or observes which websites the group often uses and infects one or more of them with malware. Eventually, some member of the targeted group becomes infected.

Hacks looking for specific information may only attack users coming from a specific IP address (same request from different IPs might result in different responses), so malware will be only delivered to our victims. On the other hand standard responses are sent back to http requests coming from non-interesting users or entities (e.g., researchers investigating this attack from a different source IP address). This also makes the hacks

harder to detect and research. The name is derived from predators in the natural world, who wait for an opportunity to attack their prey near watering holes.

Our selected organization (a National CSIRT) did not suffer from such known attack however we will test our work into the organization by simulating the same Watering hole attack to our selected organization.

In our case, a top manager (with enough access rights to classified data) is usually reading a specific third party news website on a daily basis. Web surfing to these external systems is not part of the organization RA/RM scope. Then, despite all efforts at the perimeter and risk management countermeasures, a Watering hole attack could affect not only the victim data and its laptop but also any data accessible by the victim like the classified data.

We will then introduce the same traffic formatted in STIX™ and OWL. The reason for this is because this type of attacks are really by-passing the perimeter, being one of the most probable attacks against leading organizations in cybersecurity arena that are usually implementing good solutions at the perimeter. These attacks are targeting the people directly despite the perimeter, attackers try to have some interaction by the end user to help them to get into the system (e.g. like accepting a malicious installer from a trusted source like their usual news website)

We will use our integrated and semantic model to manage all related data in an integrated and effective way. Our objective it is to better protect our organization risks dynamically by improving our detection and prevention capabilities based on leveraged Cyber Threat Intelligence data.

All relevant information is managed by our model at all levels: data, business logic, services / applications, visualization and risk Assessment / risk Management. Another objective it is to facilitate the investigation by a new enriched data model.

For that, all CTI data is parsed into standards STIX™ and OWL (stix2.owl). Risk owners will define their services under scope and evaluate their importance as usual (in any other risk management framework like [ISO/IEC27005:2008]), however this time they will define that in DRM (DRM.owl) not in Excel or any other static file. The rest will be automatically done by our reasoner (e.g. asset dependencies, threats identification, inventory, inherited assessment, etc.)

Figure 77 Graphical description representing the TTP of the Watering hole attack

9.3.1.1. TTP of the Attack

On July the 8th, a news website was compromised. The target organization was used to have traffic to that news website from different internal (corporate) users. Since that date (firstSeen=2017-07-08 as timestamp in STIX™) a Malicious Javascript Artifact (CyberObservable of Class Artifact and mimetype="javascript" in STIX™) was injected in the homepage, affecting a specific company (e.g. our company) but not any other companies.

That was possible because of the business logic behind the Javascript. All users navigating through http network traffic (NetworkTraffic in STIX™) to that domain name (DomainName in STIX™) where loading the Javascript that was automatically redirecting to load another (second) Malicious Javascript. Our company users (and not other connections with a different source IP (srcRef in STIX™) were again redirected into a different Malicious URL where a Windows (exe) Installer (CyberObservable of Class URL in STIX™) was dropped as a fake Flash update. The rest of the users were redirected to the original Adobe Flash Update instead of the malicious one.

The attack started to get victims from our targeted organizations and were persisted at least until August the 1st, where the javascript was modified again to provide different payloads.

Once a victim executes such a fake update, its personal computer would be compromised, so threat actor would have access to the user data from its personal computer (e.g., stealing credentials, connecting to any other system from its computer, etc.).

A graphical representation of the attack TTP can be seen in Figure 77.

9.3.1.2. The Target and Motivation

The target is to get access to the organization "Classified Data". That means that the attackers are interested in those users with enough access level to any classified data. Attackers could have investigated specific user behavior like the news website they are usually reading. This information gathering is usually done by using social networks (e.g., continuous post in Twitter referencing this source for example). We will simulate the interest of the attackers around some Classified Data only accessible by specific internal (corporate) users. We will define a data property named hasAccessLevel in our ontology to define the access level of each user of our organization. That property will have a value between 1 and 5 (from low to high level access respectively). We can then use this variable in any SWRL rule.

9.3.2 Use Case 2: Exchange of Threat and Risk Intelligence

The objective of the work is to provide a coherent solution for all of the open challenges in Cyber Threat Intelligence sharing, seen in Table 1, at the same time. The detailed characteristics of our all-in-one solution, are presented in the Table 2. We propose an innovative Cyber Threat Intelligence (CTI) Exchange model, based on the combination of semantic web, STIX™ and the Ethereum Blockchain as seen in Figure 53. It provides new type of incentives to all roles involved. Current approaches do not provide enough incentives for producers, as a result, most of the entities involved are just consumers.

Our proposal also provides incentives to engage new type of roles like the investors. We propose different type of economic incentives:

- A new CTI token (under the ERC20 [ERC20] standard), which represents the threat and risk intelligence data, as a digital asset. The creation of a new token will facilitate the interoperability with any other standard token within the Ethereum blockchain

- Crypto currencies (cash in Ether currency) to support the use of taxes in the exchange of knowledge (pay per use).

In our work, we also propose to **share enhanced knowledge in the format of semantic algorithms or rules** (beyond IoC). It uses semantic variables in an OWL2 version of STIX™v2 format, to support complex representations with more expressivity.

Rules are provided in the format of semantic rule language rules, as proposed by Riesco et al. [Riesco19a] [Riesco19b]. The use of ontologies, by definition, enables the interoperability and unambiguity of concepts. At the same time, it allows the usage of semantic reasoners to infer new knowledge, which might help to bring the needed automation while keeping the data consistency (reliability). It will be also more effective with regard to the number of data to be shared, because the value of the algorithm is greater than just a specific IoC. It is also less ephemeral [Bianco14].

It is recommended to read CHAPTER 8, to better understand Use Case 2 prototype validation.

9.4. PROTOTYPE VALIDATION

In the following sections, both prototypes are described.

9.4.1 Use Case 1 Prototype Validation

We used Protégé Tool by Stanford [Protégé] to test our CTI/DRM Integrated Model with the proposed use case at section 9.3.1.1.

All data related to this TTP description will be under STIX™ and OWL. At the same time all DRM data will be under OWL. As a result, all data in our work is under a semantic data model.

9.4.1.1. Architecture

As described in section 5.9 we propose a layered approach with some characteristics:

- The whole architecture is based on a mix of standards (STIX™ [OasisSTIX1], OWL [W3C-OWL], SWRL [W3C-SWRL]) as a standard proposal oriented to have a quick and widespread adoption by the industry with the needed expressivity.
- STIX™ standard is used to handle all threat and traffic data (really any CTI data) but under OWL format (stix2.owl is the ontology that we have created for that).
- It is an integrated architecture. Our dynamic risk management framework is leveraging cyber threat intelligence data. For that, DRM.owl ontology imports STIX2.owl. There are relationships between objects and our reasoner together with SWRL will create new knowledge based on those relationships.
- SWRL is used for all business logic: algorithms and rules.
- It is a layered approach, a use case at service / application layer will need some business logic layer to access the data layer. The same happens with DRA/DRM or any other type of service (e.g., an alert).
- It is a Semantic approach, that means that a semantic reasoner (in our case, Pellet incremental reasoner [W3C-Reasoner]) will understand the meaning of data and all their relationships to infer new knowledge dynamically.
- Relevant data (e.g., events, traffic) collected for this work were necessarily parsed into our ontologies to demonstrate the benefits of using semantics. On the other

hand, we limited our data collection to STIX™ which is becoming a de facto standard. Future sensors will probably collect data under this new standard without the need to parse data (e.g., logs).

- The collection of all data used in a SWRL rule has not necessarily to be done on real time.
- We used a unique central data storage to implement and validate our work.
- The purpose of our work is to demonstrate the benefits to have all data under semantics and standards like STIX™ but it is neither the collection / parsing of the data nor the additional overhead in case of new type of sensors to collect STIX™ related data.
- Our data model is based on five types of concepts (entities):
 - Classes / SubClasses to represent the concepts, objects, families or classes of the ontology.
 - Object Properties to represent the relationships between objects.
 - Data properties to represent the variables, values or fields of each object.
 - Data types to represent the types of the data (float, integer, string, dateTimeStamp, custom, etc.).
 - Individuals to represent the "instances" of the different Classes.
- Because all data will be parsed into semantic OWL data, all the business logic, services / applications, as well as all the dynamic risk assessment and management models are implemented using SWRL rules and a reasoner. We selected Pellet Incremental reasoner [W3C-Reasoner] for this work to provide live semantic reasoning.
- As described earlier, we developed STIX2.owl (as the semantic and complete version of STIX™ v2.0 [OasisSTIX2]) and DRM.owl ontologies. By today, we have about 453 Classes and above 11565 axioms. We also developed different families of SWRL rules to support all levels of the Model presented in this work. 34 SWRL rules were created to test all of our implementation.
- Visualization Model is based in Protégé plugins, especially OntoGraph. With regard to querying the Data we used SQWRL as a standard query language by using its corresponding Tab in Protégé [Protégé].

9.4.1.2. Instances of Our Threatened Organization

With regard to the organization, we describe the set-up to test the behavior against the watering hole attack (use case) in the following subsections. Due to the motivation of the attack (access to classified data), we will describe in details those instances on risk as well as the corresponding relationships. This will help the reader to understand the attack vector.

Internal (corporate) users:

We use two main profiles of the organization, both belonging to class InternalUsers as SubClassOf Users:

Victim₀ will be the CFO user which is the Chief Financial officer of the organization. It hasAccessLevel 5 out of 5, so it would have access to all ClassifiedData in our company. However it will not have so much CybersecurityExperience (3 out of 5), so that characteristic will be taken into account for dynamic risk assessment as this type of users will likely

execute a fake SW Update with more probability than a cybersecurity expert (like the AdobeFlashUpdate described as part of the TTP).

Victim₁ will be the CybersecurityExpert user which is the Head of Cyber Threat Intelligence of the organization. It hasAccessLevel 3 out of 5, so it would not have access to ClassifiedData in the organization but it would have access to threat intelligence data and any other type of data equal or below its hasAccessLevel data type. It will have high CybersecurityExperience (5 out of 5), so that characteristic will be taken into account for dynamic risk assessment as this type of users will likely not execute a fake SW Update (like the AdobeFlashUpdate described as part of the TTP). Reject the execution (rejectExecution action in our model) is semantically equivalent as not executing the fake update. As part of its reaction, it could also create an IOC (indicator of compromise) after detecting and investigating its malicious behavior. It will be shared within the CTI community to avoid future attacks, this time, based on knowns (IOC data). Furthermore, it could also open an internal security investigation together with the CISO (Chief Information Security officer) and the quality team in charge of the information security certification based on ISO2700X.

Data:

It is a subClassOf Asset. We selected five specific instances of data of our organization, all belonging to class Data, but only two of them belonging to the subClass ClassifiedData.

There are five instances in our work:

- Data₁instance has a requiredAccessLevel of 3,
- Data₂instance has a requiredAccessLevel of 2,
- DataClassified₁ instance has a requiredAccessLevel of 5,
- DataClassified₂ instance has a requiredAccessLevel of 5,
- FileFlashPlayer instance of the malicious executable dropper.

Risk Scope, is the class representing all the services or projects in scope of the organization risk management certification.

There are different instances of risk scope (RiskScope concept in our ontology) representing all the different services or projects in scope of the risk management certification of the selected organization (based on ISO2700X). By using semantics, our framework will enable interoperability between any risk management framework or standard as all of them will use equivalent concepts.

For the testing of this work, we selected some relevant and interesting services of our organization, especially two services that dependsOn (property) ClassifiedData (class) named:

- GDPR-compliance instance of class RiskScope, which dependsOn DataClassified₁.
- CyberThreatINTEL-service instance of class RiskScope, which depends on DataClassified₂ and Data₂

Software is the class representing the installed SW on the PC Computers of users to access any data.

To validate this work, we selected the following SW instances:

- OperatingSystem instance of class Software representing the Operating System installed on the PC,
- Browser instance of class Software representing the Browser installed on the PC,
- AdobeFlashPlayerPlugin instance of class Software representing the Browser installed on the PC,

Personal Computer is the class representing the computers of the different users which will be used to access the data.

To validate this work, we selected just the following instances:

- PC-CFO instance of class PersonalComputer representing the PC of the CFO,
- PC-CybersecurityExpert instance of class PersonalComputer representing the PC of the Head of Cyber Threat Intelligence.

CyberObservable is the class representing all of the STIX™ Cyber Observables.

To validate this work, however we implemented the entire STIX™ v2 standard, our watering hole attack only needed instances belonging to the following subClasses of class CyberObservable:

- Artifacts class representing malicious javascript artifacts,
- Domain-Name/SubDomains class representing domains and subdomains,
- IPv4Addr class representing IP addresses,
- URL class representing URLs,
- Network-Traffic class representing the network traffic.

Safeguards represent the class representing the countermeasures of type safeguards. They represent the controls set up by the organization as for example in the Equation 15 and Equation 7.

To validate this work, we selected just the following safeguards' instances:

- LDAP-Group-and-Roles-Control instance of class Safeguards representing the access control based on LDAP.
- Antivirus instance of class Safeguards representing the Antivirus of the organization.
- Information Sharing control instances which are created automatically by the reasoner for the tactical scenario of Figure 90.
- HR Related Security Awareness Education and Cyber capabilities control instances which are created automatically by the reasoner for the strategic scenario of Figure 91.

AssetValuation, Threats, Risk, RiskAssessment, RiskSeverity, SecurityEvents classes are also used however their instances are created automatically by using SWRL rules. This is one of our main contributions as errors by hand are frequent in risk management frameworks nowadays. Organizations are also addressing risk management by simplifying relationships because until today they were difficult to consider if done by hand. Our reasoner could understand the meaning of our data, so does its relationships. When risk owners evaluate the service in scope, all related dependencies are considered. An automated dependency tree is created because of static or even inferred relationships. All their values are updated and inherited thanks to automatic Asset Valuation SWRL rule. The same happen when a data is of type ClassifiedData, the reasoner understands that different types of risks are associated to such data because it is classified. Associated risks to that data are identified automatically. Risks are ranging from Bad Reputation Risk (if someone leak that data) to Deliberated Malicious SW Distribution (if someone has access to that data but it does not have enough cybersecurity expertise) depending on the context around. In order to make this happen, rules should allow enough expressivity which is the main reason we use OWL as suggested by STIX™ white paper (RDF/OWL) [OasisSTIX1].

Dependencies are a key factor to define the different scenarios or contexts.

Some examples of SWRL rules have been provided during this work (some are Figure 60, Figure 98, Figure 99, Figure 101 and the ANNEX D). They implement most of the business logic however there are different families, some of them:

- **Asset valuation rules** to automatically calculate the asset value based on inheritance (risk frameworks usually evaluate the importance of the service and depending on asset dependency tree, there are cascading effects). This is done automatically to avoid errors (errors by hand are frequent) as well as to have a broader, consistent and a more standardized picture.
- **Enrichment rules** to enrich all our data model (e.g., for example filling down IP-Domain relationships (like a DNS / Reverse DNS). Once some data is missing, relationships between data would help the reasoner to infer the missed data. See Figure 60 for an example.
- **Threat inventory rules** to automatically identify threats depending on our topology. As an example, instances of class "Deliberated Malicious SW Distribution Threat" will be created around our company data when specific users that have access to that data have, at the same time, a cybersecurity experience below 3 (3 out of 5). See Figure 98 to see the SWRL rule.
- **Risk inventory rules** to make an automatic inventory of all risks. It uses identified threats to guess associated risks to them dynamically. It is important to have enough and accurate information for decision makers along the time.
- **Risk assessment rules** to make the assessment of each risk which depends on each threat. As an example, see Figure 81 for residual risk calculation.
- **Risk management rules** to decide the best strategy depending on the organization policy. An example can be seen in Figure 100 where we implemented the organization policy for high level risks in SWRL. In this case, any high level risk will have associated a mitigation strategy instance. The specific action to be performed could also be automated. Also in this category we created a tactical level rule see Figure 78 to implement a dynamic information sharing once an attack is detected. The whole SWRL detection sharing rule will be shared instead of IOC. On the other hand a strategic level rule was also created see Figure 79 to implement new knowledge for the reasoner. In this case new Awareness training programs are recommended for those users being threatened and at the same time they are not experts in cybersecurity.

Name	RiskManagement#6 Intelligence Information Sharing of specific pattern detection when attack is going-on
Comment	
Tactical Approach	
Status	
Ok	<pre> drm:SecurityEvents(?s) ^ stix2:type(?s, "Dropper behaviour of Malicious Windows Executable"^^rdf:PlainLiteral) ^ swrlx:createOWLThing(?x, ?s) -> drm:InformationSharingControl(?x) ^ stix2:name(?x, "Intelligence SWRL to detect Watering hole attacks pattern"^^rdf:PlainLiteral) ^ stix2:description(?x, "Decode by using Base64 (4oCcKHN0aXgyOk5ldHdvcmUcmFmZmljKD9udCkgXiBzdGl4Mjpkc3RQYXIsb2FkUmVmKD9udCwgP3BsKSBeIH0a XgyOkFydGlmYWN0KD9wbCkgXiBzdGl4MjptaW1VHlwZSg/cGwslCJqYXZhc2NyaXB0Il5ecmRmOlBsYWluTGIOZXJhbC kgXiBzdGl4MjpyZWRpcmVjdGlvbG/cGwslD9yZWQpIF4gc3RpeDI6VjJMKD9yZWQpIF4gc3RpeDI6TmV0d29ya1RyYWZ maWMoP250MikgXiBzdGl4Mjpkc3RSZWY0P250MiwgP3JlZCkgXiBzdGl4Mjpkc3RQYXIsb2FkUmVmKD9udDIsID9wbDI pIF4gc3RpeDI6c3JjUmVmKD9udDIsID9zcikgXiBzdGl4MjpleHRlbnNpb25zKD9wbDI6ICJ3aW5kb3dzLXBlymluYXJ5LW V4dCjEjXkZjPQbGFpbkxpdGVyYWwplF4gc3RpeDI6bmFtZSg/cGwylCA/bm0pIF4gc3dybHg6bWFRZU9XTRFoaW5k D94LCA/bnQyKSAtpiBkcm06U2VjdXJpdHlFdmVudHMoP3gpIF4gc3RpeDI6dHlwZSg/cEwgInNlY3VyaXR5LWV2ZW50Il 5ecmRmOlBsYWluTGIOZXJhbCkgXiBzdGl4MjpczcmNSZWY0P3gsID9zcikgXiBzdGl4Mjpo0eXBkID94LCAiRHJvcHBlciBiZW hhdmlvdXlgb2YgTWFSaWNpb3VzIFdpbmRvd3MgRXhIY3V0YWJsZSJeXnJkZjPQbGFpbkxpdGVyYWwplF4gc3RpeDI6ZH N0UmVmKD94LCA/cmVkKSBeIH0aXgyOnJlbgF0ZWRUbyg/eCwgP250Mikp4oCd")^^rdf:PlainLiteral] </pre>

Figure 78 SWRL Tactic rule to launch an Information Sharing of a “Detection rule”

Name
RiskManagement#5 New Awareness Training based on SecurityEvents targeting nonExpert users
Comment
Strategic approach
Status
Ok
<pre> drm:SecurityEvents(?s) ^ stix2:srcRef(?s, ?src) ^ stix2:observableOf(?src, ?pc) ^ drm:Computer(?pc) ^ stix2:isUsedBy(?pc, ?u) ^ drm:InternalUsers(?u) ^ drm:hasCybersecurityExperience(?u, ?ex) ^ swrlb:lessThanOrEqual(?ex, 3) ^ swrlx:createOWLThing(?x, ?u) -> drm:HRRelatedSecurityAwarenessEducationAndCybercapabilitiesControl(?x) ^ drm:protects(?x, ?u) </pre>

Figure 79 SWRL Strategic rule to select specific users under attack for an “Awareness Training”

- **Risk severity rules** to make another type of automated classification depending on the severity and the policy defined by the organization.
- **Security policy rules** for example to automatically make an inventory of users and their data access rights. Depending on hasAccessLevel property of each user and requiredAccessLevel property of each data, the reasoner will keep an updated version of access rights database, that is, who has (or could have) access to what. Thanks to that, the reasoner is able to associate risks in a granular a detailed way. (See Figure 80)
- **Threat intelligence rules** to detect Security Events instances like the SWRL rule shown in Figure 101. We created it to detect a Security Event dynamically once it fulfill the antecedent pattern of our rule. The algorithm will detect a malicious TTP pattern when any user from our organization is being redirected after another redirection to an URL which is dropping a (potential malicious) exe installer (in our case a Flake Flash Installer). By using this behavioral pattern rule, we are detecting a security event related to the TTP of the selected Watering hole attack. This security event will trigger a risk re-assessment to recalculate the new risk level automatically. The risk probability will be increased by this security event accordingly and dynamically. As a result, there would be a new instance of class high risk coming from an instance of medium risk due to its probability has changed. The risk is of type Deliberated Malicious SW Distribution. Now it is worse as there has been a security event. An event like this will trigger our reasoner to open or instance a new (proactive) security incident although the end user still has not executed the installer. Once the end user executes it, the proactive incident will become a reactive incident. Recommended actions will differ between proactive or reactive state but most of them could be automated by our framework.
- Etc.

Name
SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)
Comment
Status
Ok
<pre> drm:Users(?u) ^ drm:Data(?data) ^ drm:hasAccessLevel(?u, ?a) ^ swrlb:greaterThanOrEqual(?a, ?r) ^ drm:requiredAccessLevel(?data, ?r) -> drm:hasAccessTo(?u, ?data) </pre>

Figure 80 SWRL Enrichment rule of a Data Access Security Policy

Name
RiskAssessment#3 Auto Residual Risk for Deliberated Malicious SW Distribution
Comment
Status
Ok
swrlb:subtract(?ar, ?prisk, ?v) ^ drm:potentialRisk(?x, ?prisk) ^ stix2:type(?s, "control-against-malicious-sw") ^ stix2:value(?s, ?v) ^ drm:Safeguards(?s) ^ drm:PotentialRisk(?x) ^ stix2:type(?x, "Deliberated Malicious SW Distribution Risk") -> drm:ResidualRisk(?x) ^ drm:actualRisk(?x, ?ar) ^ stix2:isMitigatedBy(?x, ?s)

Figure 81 SWRL Auto Residual Risk Calculation for Deliberated Malicious SW Distribution Risks.

Figure 82 Screenshot in Protégé tool of the Security Event created by a TTP detection rule

9.4.1.3. Dynamic Risk Equation

We are finally able to implement the dynamic risk assessment equation (see Equation 7 and Equation 16) by using our model. We use SWRL rules to decrease values when a safeguard (CN) mitigates that risk. There are also rules to increase values when a security event (EPN or EIN) increases the same risk.

In our case:

Equation 16 Residual Risk associated to that threat i , at time $t=1$

$$R_{i_{t=1}} = P_{i_{t=1}} + I_{i_{t=1}} - \Sigma(C1_{i_{t=1}}, C2_{i_{t=1}}),$$

$$P_{i_{t=1}} = P_{i_{t=0}} + 1$$

$$I_{i_{t=1}} = I_{i_{t=0}} + 0$$

where:

- i is the "Threat of Deliberated Malicious SW Distribution",
- $P_{i_{t=1}}$ is the probability of that threat at time $t = 1$, that is "newp" of figure 12,

- $I_{i_{t=1}}$ is the impact of that threat at time $t = 1$, which is equal to the impact at $t = 0$ because the recent security event is only updating the probability,
- $C1_{i_{t=1}}$ is the decreasing value of the Impact (Severity) I or Probability P of that threat i at time $t = 1$ due to Countermeasure 1 which is the "Antivirus" (reducing 0.5 float),
- $C2_{i_{t=1}}$ is the decreasing value of the Impact (Severity) I or Probability P of that threat i at time $t = 1$ due to Countermeasure 2 which is the "LDAP" (reducing 0.0 float as it is not considered a countermeasure for this type of threats),
- $EPN_{i_{t=1}}$ which is equal to 1 because it is the increasing value of the Probability P of that threat i at time $t = 1$ due to the Security Event $N = 1$, in our case it is 1.0 float once the security event of a Dropper has been detected,
- $EIN_{i_{t=1}}$ which is equal to 0 because it is the increasing value of the Impact (Severity) I of that threat i at time t due to the Security Event $N = 1$, in our case it is 0.0 float, the security event updates the probability not the impact variable.
- $R_{i_{t=1}}$ is the Residual Risk associated to that threat i , at time $t = 1$, it is "newar" of figure 12,

The prototype shows how the probability changes after a suspicious pattern is detected as it increases the probability of a specific threat type.

New values are calculated and they can be seen in the Figure 83 after a Dropper has been detected inside Network Traffic of the CFO User. As an example:

Equation 17 New values after risk re-assessment

- $p = \text{previous probability} \rightarrow \text{newp} = \text{new probability}$
- $pr = \text{previous potential risk} \rightarrow \text{newpr} = \text{new potential risk}$
- $ar = \text{previous actual risk} \rightarrow \text{newar} = \text{new actual risk}.$

SQWRL Queries OWL 2 RL Dynamic Risk ReAssessment after a Security Event of type Dropper by a less experienced User							
u	rr	p	newp	pr	newpr	ar	newar
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:bbe0e299_00...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_etc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_etc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_etc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen70:dc0b56_etc...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float
drm:CFO	autogen2:d87b8bea_ff7...	"4.0"^^xsd:float	"5.0"^^xsd:float	"7.0"^^xsd:float	"8.0"^^xsd:float	"6.5"^^xsd:float	"7.5"^^xsd:float

Figure 83 Results of a querying rule (SQWRL) for a Risk recalculation

Risks Assessments will be updated dynamically, so does its Risk Management classification and their recommended treatment.

Until today, our selected organization is re-assessing risks once a year. Risks considered by our organization are also related to threats identified by hand by risk owners. Those are usually reflected in an Excel file or any other static file. By using our proposal we are providing to our organization the benefit to leverage their CTI into a dynamic risk management framework. The proposal is based on standards [OasisSTIX1] [W3C-OWL] [W3C-SWRL], then, any other organization who will be working under these standards will have the same benefit as well.

In our case, a risk that had been identified as a medium level risk by the organization is dynamically re-classified under high level risk once a threat is detected that will potentially be affecting a classified data. The reasoner will provide reasoning evidences for that, in our case, a specific user (CFO) with low level cybersecurity expertise but, with high level access to that classified data, is under attack. As this user has inherently more probability to accept this malicious fake installer, the dynamic risk framework will trigger a re-classification of the risk with that granularity (details) in that specific moment, before it becomes a real incident. It is then a proactive and preventive response with again, different options or alternatives.

Semantics would allow a pseudo-automated response execution were complex and/or special actions would require a balanced human supervision and interaction (HMI). This is our recommendation for all decision making actions related to cybersecurity domain.

9.4.2 Use Case 2 Prototype Validation

This section details specific building blocks that are considered interesting for reproducibility.

9.4.2.1. Environments

For the implementation we have developed an Ethereum Smart contract written in Solidity language which has been deployed into the public blockchain as a Decentralized application (DApp). We use Remix IDE²⁵ as a powerful online integrated development environment (IDE) to code and debug the decentralized application. We also use Metamask as a Wallet and as an injected web3 provider that is needed to connect to and interact with Ethereum network nodes.

On the other hand, we have deployed and tested our implementation in two different blockchain environments:

- **Development environment:** A local and private Ethereum blockchain by using Truffle-Ganache framework²⁶.
- **Integration environment:** A remote and public Ethereum blockchain by using one of the most used test blockchain networks named Ropsten²⁷.

The main-net (Production environment) is equivalent to Ropsten with regard to the consensus and behavior, on the other hand it has more nodes but it also needs real ether. We then decided that using Ropsten was enough to demonstrate the benefits of our work, by using the main-net, the benefits will be higher for end users as there are more nodes to provide better and faster confirmations. Ropsten anyway can support all the features and CTI data from our work.

9.4.2.2. Smart Contract (DApp) and the "CTI" Token

With regard to our Smart contract, we named it "CTI Marketplace". In order to follow coding best practices, we extended an IERC20 (ERC20 token interface [ERC20]). One of the benefits of using this ERC20 standard is the interoperability between tokens within Ethereum blockchain. Tokens are a way to represent any digital asset, once the interoperability is guaranteed within the Ethereum blockchain, there could be exchange

²⁵ <https://remix.ethereum.org/>

²⁶ <https://truffleframework.com/ganache>

²⁷ <https://ropsten.etherscan.io/>

between different tokens if they follow the same standard implementation like the one we provide in our work.

A specific token for the purpose of this work named CTI was created, the acronym of "Cyber Threat Intelligence" token. The token is created the very first time the contract is deployed. For that, we include its definition inside the **constructor** function as seen in the pseudocode of Equation 18.

Equation 18 Pseudocode of Constructor and token initialization

Data: Constructor function executed just once: while our smart contract is being deployed
Result: Initializing variables
 properties;
 $public \leftarrow visibility$
 initialization;

```

owner = msg.sender /* Wallet address of the source of the transaction who
                    deployed the smart contract (address)                */;
symbol = "CTI" // New token symbol (string);
tokenDescription = "Cyber Threat Intelligence Token" /* Description of the new
token (string)                                          */;

```

A token enables the possibility to invest in it, it is equivalent to a digital asset, in our case a digital "Cyber Threat Intelligence" token asset, which value might be getting more and more interest in the market. More interest means more value, like any other asset in the market.

In the Figure 84, a sequence diagram is shown to describe the process, messages and transactions between the browser, the Metamask Wallet, Ethereum nodes, the Smart contract itself and the CTI token, which is initialized as an ERC20 standard token.

Figure 84 Sequence diagram of the CTI Marketplace Smart contract deployment by its Owner (Wallet 1).

Another coding best practice implemented, is the use of safe math libraries²⁸ written in solidity, to implement math operations with safety checks that revert on error.

Our CTI Marketplace smart contract, will store CTI data which will be exchanged under certain conditions. It will also provide different incentives to each role. This time, we will implement the use case of sharing algorithms (e.g. TTP detection rules) beyond individual indicators of compromise (IoC) as proposed by authors in [Riesco19a] [Riesco19b]. To store that information, we decided to use **mappings** which have some

²⁸<https://github.com/OpenZeppelin/openzeppelin-solidity/blob/master/contracts/math/SafeMath.sol>

similarities to a key-value data store. At the same time, we decided to use a **struct**. Part of its data model is seen in the pseudocode of Equation 19.

Equation 19 Pseudocode of the data set structure to store CTI Data Rules in the Blockchain

```

Data: CTI Data from CTI Data producers
Result: Mapping between wallet addresses of each entity and their CTI Data (CTIRule)
properties;
addressId ← address // address of each entity;
entity ← string // name of each entity;
rating ← uint // rating of the entity;
ruleList ← string[] // CTI Data rules;
ruleName ← string[] // List of rules description;
ruleConfidence ← uint[] // List of rules confidence;
initialization;
CTIStructs → address → CTIRule // mapping addresses and rules;
balance ← uint256 // economic balance;

```

In this case, each Ethereum blockchain address will belong to an individual or an organization. At the end, the address is the public key of a standard asymmetric encryption system, where each private key associated to each public key is only known by each of the owners. Each organization has a struct with its name, its rating by users or consumers about the quality of the entity as a CTI source, and at the same time, it will have an array of rules represented by **string[] ruleList** where each rule is representing an algorithm (e.g. TTP detection algorithm).

We used a string format to store each rule as it will be able to represent not only a SWRL rule but also any other less expressive rule language. We suggest the usage of SWRL rules formed by an antecedent and consequent together with CTI and DRM ontologies presented by authors in [Riesco19a] to have actionable intelligence once the rule is downloaded (e.g. downloading a rule means a rule which is read or listed by calling the CTI Marketplace smart contract functions). If two organizations use the same semantic ontology [W3C-Ontology], CTI concepts will be the same, and rules (using those concepts) will be working as plug and play, as they will be understood by their semantic reasoners [W3C-Reasoner]. Downloading a CTI Data rule means learning new knowledge by our semantic reasoner. CTI ontology provided by authors in [Riesco19a] was based on STIX™ v2.0 [OasisSTIX2] as a de facto standard to work in cyber threat intelligence domain.

Each rule will have associated a **ruleName** which is the name or description of the rule (e.g. describing the objective of the rule) as well as a **ruleConfidence** representing the confidence of the rule which is related to the declared quality of each of the rules provided by the organization. Being the rules an array of strings, names and confidence of rules will also be arrays of string and unsigned integers respectively. Depending on the real confidence experienced by consumers using those rules, the **rating** of the organization will be higher or lower as consumers will have the opportunity to vote.

Equation 20 Pseudocode of the solidity function to share CTI Data by CTI Data producers

```

Data: Arrays of new CTI Data shared by CTI Data producers
Result: New CTI Data added to the storage structure of each entity, ready for reading
properties;
payable ← public // the function can receive Ether by anyone;
addressId ← address // address of each entity;
entity ← string // name of each entity;
rating ← uint // rating of the entity;
CTI Data addition (writing into storage);
for i ← 0 to NewCTIData.length do
    ruleList ← newString[] // Adding new CTI Data rules;
    ruleName ← newString[] // Adding new CTI Data rules description;
    ruleConfidence ← newUint[] // Adding new CTI Data rules confidence;
Optional investment;
if is a new entity then // only for new entities
    // Mint of tokens with special investment conditions;
    mint(msg.sender, msg.value * 1000 * NewCTIData.length);
    emitUpdateStatus('A new user has shared first bulk of CTI rules');
else // known entities
    // Mint of tokens with standard investment conditions;
    mint(msg.sender, msg.value * 20 * NewCTIData.length);
    emitUpdateStatus('An existing user has shared a new CTI rule');

```

9.4.2.3. Business Model and Economic Incentives

Wei is the minimum value of *Ether*. Gas is measured in *Gwei* so the ratio between the three of them is shown in Equation 21.

Equation 21 Wei - Ether - Gwei conversion ratio

$$1 \text{ Wei} = 10^{-18} \text{ Ether} = 10^{-9} \text{ Gwei}$$

As seen in the pseudocode of Equation 20, when a CTI Data producer is inserting a new rule in its mapped struct, (equivalent to share a new CTI data rule), it has the option to attach to the transaction any Ether (cash) to get special investor's conditions. A real transaction mined and executed in Ropsten blockchain, which is inserting 2 new rules simultaneously, is shown in Figure 85. In this case, the CTI Data producer is attaching 10 Wei of cash in the same transaction. We used 10 Wei in our experimentation, as a reduced value that will not interfere in the calculation of the network cost. The final tax value, to be used in the production environment, will need to be guessed and fine-tuned by using the Montecarlo simulations (as seen in Figure 70). As paying is optional, special investment conditions are applied (20 tokens/each new rule). It is then receiving back 400 CTI ERC-20 standard tokens = 10 Wei * 20 tokens/each new rule * 2 rules added.

Figure 85 Capture of a running-example where the system automatically mint and send 400 CTI tokens.

In this case, depending if it is the first time it shares or not, (there is a check for that condition in Equation 20, they will get different incentives. If it is the first time sharing, it will get 1000 tokens per each Wei sent with the transaction. From that moment, it will get 20 tokens (not 1000 but double of a standard investor which is getting 10) per each Wei sent associated to any new sharing.

As sending cash is not mandatory, being an investor (although it is optional), is a new incentive to CTI producers as well. This is one of the new contributions, as new incentives, of our work. CTI tokens will be created (using *mint* function from IERC20) in case the CTI Data producer takes this opportunity.

The function named **insertBulkEntityRules** (which pseudocode is seen in Equation 20, is defined as public as well as a payable function. The former characteristic of the function allows the reception of external calls from anywhere. The latter characteristic enables the reception of payments within the function call (for investments).

The mint of tokens uses an inherited function from the ERC20 standard named *mint* as seen in [ERC20]. Its usage can be seen in our Equation 20 and Equation 23.

With regard to **CTI Data consumer** role, this role has several functions available to consume CTI Data rules. All of them are related to the possibility to read from the key-value data store of our Smart contract.

In an example, knowing the (wallet) address of a CTI Data producer we will use the function seen in the pseudocode of Equation 22, to list its CTI Data rules. In order to do that, the CTI Data consumer is forced to send a minimum of 10 Wei of cash in order some taxes and benefits (as incentives for owner and CTI Data producer respectively), are applied.

Equation 22 Pseudocode of the solidity function used by CTI Data consumers to get access to CTI Data of a specific producer

```

Data: Input parameters: the address of the CTI Data producer AND payment of cash (msg.value > 10 Wei)
Result: A CTI Data consumer read CTI Data from a specific producer storage
properties;
payable ← public // the function can receive Ether by anyone;
ifValue ← modifier // filter to require a minimum of 10 Wei;
CTI Data read (reading from storage);
if msg.value > 10 Wei then // requisite
    emitUpdateStatus("These are the rules of the selected entity");
    ruleList[address] → OutputString[] // Reading CTI Data rules;
    ruleName[address] → OutputString[] // Reading CTI Data rules description;
    ruleConfidence[address] → OutputUint[] // Reading CTI Data rules confidence;
else // error is thrown
    | revert
Economic Incentives;
owner.transfer(msg.value/10); // 10 percent of taxes are sent to owner;
a.transfer(msg.value/2); // 50 percent of taxes are sent to CTI Data producer;

```

The sequence diagram shown in Figure 86 gives an overview of the process followed to list all CTI Data rules of a specific CTI Data producer. There are taxes that will be automatically transferred (in Ether) to the Owner and the CTI Data producer, taxes paid by the CTI Data consumer. Owner and CTI Data producer will receive 10 and 50 percent of the value (msg.value) of the transaction respectively. The remaining 40 percent of cash sent will be accumulated within the Smart contract in order to boost the value of the CTI Token.

Figure 86 Sequence diagram of a CTI Data consumer listing CTI data rules of a specific producer.

The role of **Investor** is available to anyone who sends cash to our Smart contract. If the cash is sent associated to a CTI Data producer transaction, it can get specific investment conditions as seen in the pseudocode of the Equation 20. If it is just a simple payment to our Smart contract by a standard investor, we implemented the fallback function to handle that payment converting it into an investment.

The fallback function (**function()**) without parameters is usually at the end of the solidity code of a smart contract. It will create an automatic investment in our CTI token on behalf of the sender address in case it sent cash directly to our Smart contract. The pseudocode can be shown in Equation 23.

Equation 23 Pseudocode of the solidity "fallback" function for investments in the "CTI"

```

Data: Input parameters: payment of cash (msg.value > 10 Wei)
Result: An Investor receive back a number of CTI tokens proportional to the investment made
properties;
payable ← public // the function can receive Ether by anyone;
ifValue ← modifier // filter to require a minimum of 10 Wei;
Investment (getting CTI tokens from the smart contract);
if msg.value > 10 Wei then // requisite
|   tokens = msg.value * 10 // 10 tokens per 1 Wei paid;
|   mint(msg.sender,tokens) // tokens are created and automatically sent back to Investor;
else // error is thrown
|   revert
Economic Incentives;
balance ← msg.value;
// the smart contract itself is receiving and keeping the cash of the transaction;
// CTI tokens are interoperable under ERC20 standard. ;

```

Figure 87 Sequence diagram of two different type of investments: single investor (Wallet 4) and a CTI Data producer (Wallet 2)

In Figure 87, a sequence diagram is shown to describe two different investment transactions:

- **Single Investor (Wallet 4)** is sending cash to the CTI Marketplace Smart contract which fallback function will handle the payment as an investment. Cash will be converted in tokens by 10 times its value as seen in the pseudocode of Equation 23.
- **CTI Data producer (Wallet 2)** is sending cash associated to a new exchange of CTI Data rules to the CTI Marketplace Smart contract. This time the function insertBulkCTIrules will handle the payment as an investment. Cash will be converted in tokens by 20 times its value (as it is a known CTI Data producer already).

Donor role will need to use a specific function to avoid an investment. The payable function to just make a donation is named **depositFunds()**, with no input parameter. In this case, the **Owner** of the Smart contract (the one who created and deployed it into the blockchain), will receive a donation for the CTI Marketplace.

In order to test our implementation, we deployed the CTI Marketplace in two different environments. As indicated earlier, the development environment is a Truffle-Ganache instance running locally and the integration environment is the Ropsten, which is one of the main (and more used) Ethereum Decentralized Test Networks, due to its similarity to the main net.

We implemented specific conditions along the code to better incentivize the CTI Marketplace's dynamics, by improving the appeal to contribute and to exchange new knowledge in the format of CTI Data rules. As described in the introduction, nowadays there is an asymmetric balance between consumers and producers, consuming data is far more frequent than producing or sharing it. Our work is focused on providing new ways to model this kind of CTI sharing by providing new incentives to have more producers.

As seen for example in the pseudocode of the Equation 22, there are specific cash transfers stated as **owner.transfer** and **a.transfer** (where "owner" is the address who deployed the CTI Marketplace in the blockchain, and "a" is the address of the CTI Data producer). Each of these transfers are implementing automatic tax payments, when the function is called or invoked from a CTI Data Consumer. A real transaction receiving those tax payments is shown in Figure 88.

Etherscan is an online service that provides access to all the immutable registers of the blockchain. In the figure, we see the hash of the transaction, if it was successful, how many confirmations validated the transaction, in what block the transaction was handled, the time-stamp of the transaction, the source and the destination together with the amount transferred. The most interesting part is that the destination is an identified as a contract, not a wallet owned by a person. Then the contract, which is running in the blockchain, will handle all the business logic automatically when someone interacts with it. In our case, we see that the contract automatically ordered 2 payments or transfers, 1 Wei to the owner address and 5 Wei to the CTI Data producer which data has been read.

The screenshot shows the Etherscan interface for a transaction on the Ropsten Testnet Network. The transaction is successful and has 12 block confirmations. The transaction hash is 0x02dcd87bf71edbd4a93721ee76855573a1d4b485ea6fa6480674e2aa285bb00. The transaction is from address 0x71763709da2488f75bc2db5d194769d801e97fa8 to a contract address 0x763d10c63395e3e0521b28705333aa5ae394bf58. The transaction value is 10 wei (\$0.00) and the transaction fee is 0.000544919 Ether (\$0.000000).

Field	Value
Transaction Hash:	0x02dcd87bf71edbd4a93721ee76855573a1d4b485ea6fa6480674e2aa285bb00
Status:	Success
Block:	5609736 (12 Block Confirmations)
Timestamp:	3 mins ago (May-16-2019 02:33:31 PM +UTC)
From:	0x71763709da2488f75bc2db5d194769d801e97fa8
To:	Contract 0x763d10c63395e3e0521b28705333aa5ae394bf58 ↳ TRANSFER 1 wei From 0x763d10c63395e3e0521... To → 0xa09d098e88b64d415c8... ↳ TRANSFER 5 wei From 0x763d10c63395e3e0521... To → 0x71763709da2488f75bc...
Value:	10 wei (\$0.00)
Transaction Fee:	0.000544919 Ether (\$0.000000)

Figure 88 Capture of the transaction of a CTI Data consumer reading Intelligence algorithms. Associated transfers are benefits to the owner and the CTI Data producer respectively.

In this case, we decided to implement taxes to the function call, for that, we implemented a **modifier** which is a solidity specific function to force additional conditions to be met before the function is executed.

The condition is to send a minimum of 10 Wei when calling this function by a CTI Data consumer. The function will not work without it.

Then, we use this modifier named **ifValue** as part of all functions that are related to CTI Data consumer role. Our work is then contributing with new incentives for CTI Data producers as any (read only) access to their data will provide them with 40 percent of the cash used in the query. At the same time, they get special investment conditions while sharing new knowledge. At least 4 Wei will be then automatically transferred to the Wallet of the CTI Data producer when someone is reading its CTI Data rules.

In addition to this, the owner of the CTI Marketplace will receive 10 percent of the cash used. Then the remaining 40 percent will be kept within the CTI Marketplace Smart contract itself. The idea is to manage the token value along the time in order its value can be maximized and linked to the balance of the Smart contract.

Again, investors using the fallback will need to send a transfer equal or above 10 wei (because ifValue modifier is used). Remember that investors will get an amount of CTI tokens 10 times the amount invested. On the other hand, a CTI Data producer could send a value below 10 Wei when contributing, as the investment is optional, when sharing new knowledge. Moreover, producers are getting at least 20 times the amount invested, as a special condition for those who are really sharing CTI knowledge.

With regard to donations, there is no minimum value associated with them. Any donation will be received, however this time, no tokens will be generated related to donations. Another modifier named **ifOwner** is used in some functions, to protect the balance of our CTI Marketplace Smart contract. Only the owner will have rights to withdraw cash from it.

9.5. RESULTS

9.5.1 Use Case 1 Results

In our case, as the Victim₁ (cybersecurity expert) does not have access to any classified data, its instance will not have access or dependencies from this type of data neither DataClassified₁ nor DataClassified₂. In our model, we can confirm the users having access to classified data anytime by querying the ontology dataset using SQWRL for example. We can check it out also by using any graph visualization plugin expanding all relationships around any classified data's instances.

The selected watering hole attack was motivated due to the interest of the threat actor to access this type of data, but in our organization only the CFO has access to it. Then, the CFO (Victim₀) is classified automatically by our framework as a potential victim of this type of attack. Different type of risks related to any unauthorized access to classified data will be created by the framework automatically due to the nature of the data (e.g., Bad Reputation Risk when classified data is accessed and leaked, Data Protection Risk, Corporate Bad image, etc.). In addition to this, once the framework detects that there is an end user which has low cybersecurity experience with access to this classified data, the framework will make automatic connections (relationships) between both type of risks (risk of unauthorized access to classified data and risk of deliberated malicious SW distribution

to the user who has access to that data). This type of connections are possible due to different reasons:

- Our model, based on ontologies, SWRL and STIX™, is able to provide enough expressivity to any type of rule that will be understood by the reasoner.
- A SWRL rule is creating automatic dependencies between Classified Data and the CFO. The rule acts as an access control security policy rule as seen in Figure 80. The rule understands that if CFO hasAccessLevel of 5 (out of 5), it will have then access to all data, including all classified data.
- A SWRL rule is automatically detecting a potential risk of unauthorized access to classified data due to a potential deliberated malicious SW distribution threat (as an attack vector) associated to the CFO user. This user has high probability to install a malicious SW based on its low experience on cybersecurity. In the selected attack, once the attacker infects the CFO by using a watering hole pattern, there will be an identity theft granting access to any classified data. This complex TTP pattern is now possible to be written as a threat detection algorithm by using our semantic framework. (See the SWRL rule in Figure 98 as an example)
- However the CFO becomes our main target as victim due to the TTP of this attack, other relevant staff with similar access level from the organization could also be a potential target of the ThreatActor. Then, our GDPR-compliance service in scope of our DRA/DRM will identify this dependency as well. On the other hand, the CyberThreatIntel-service is also dependent of classified data as well as non-classified data. This service will have the same type of risk associated to it. In this case, the reasoner will explain that the risk comes from the probability to get access to Data_Classified_2 in case the CFO is being hacked by a deliberated malicious SW distribution threat.
- Based on the same security events, our reasoner initiates the sharing of the SWRL rule to detect such TTP pattern within other offices and partners. It is a shift of paradigm because IoC are not shared (they are simple to be changed by attacker) but the intelligence algorithm itself. Knowledge about how to detect specific patterns is shared. This is an implementation of a tactical level risk management action. See SWRL rule at Figure 78 and created instances at Figure 90.
- Also because of the same detection, our reasoner proposes other action at strategic level. In this case, specific Awareness training for workforce capacity building is selecting only the users being threatened which at the same time has low cybersecurity experience. See SWRL rule at Figure 79 and created instances at Figure 91.

Figure 89 Screenshot of Protégé tool [Protégé] of Dynamic Risk- Reassessments.

DRA/DRM is calculated properly as expected and risks are classified into different Risk Management strategies depending on the severity.

Victim₀ and Victim₁ browsed the website the day before it was compromised, but no security alerts were received (instanced). The rules were created but that malicious behavior was inexistent.

The day after, when the press site was compromised, both users, as usual, started navigating to the infected domain but after the homepage was loaded, an injected javascript started to load different artifacts from different URLs making different redirections. After that, a payload was dropping a fake (exe) AdobeFlashPlayer installer as expected in our work to both users.

Once our SWRL rule detected that traffic, it created a new instance of type SecurityEvent class classifying that behavior into a security event of type "Dropper behavior of Malicious Windows Executable", again for both users. The rule is then a pattern-like to detect the TTP of our Watering hole attack.

The SWRL TTP-like rule was designed by using a combination of sequential http request and response traffic analysis while being automatically redirected by malicious javascript until an URL is dropping a fake windows installer.

All security events triggered were using our proposed semantic architecture. Then, it was very easy to follow all new or established relationships. As an example, knowing the LAN IPv4Addr (CyberObservable) originating the http request connection, we could not only identify the user behind that connection but we were also able to identify all related Risks (and only those related to this specific threat) to make a dynamic re-assessment (DRA). Risks related to other type of threats were kept unchanged as they did not have relationships within these type of security events.

In our work, however we had security events triggered from both users' network traffic (both were navigating and experiencing the same attack), our system considered that the Cybersecurity Expert was likely not going to execute the strange (fake) Flash update or any other strange update due to its cybersecurity experience. Risk re-assessment only took place for those risks which were belonging to those users whose experience in cybersecurity were less than 3 (out of 5). The model was then able to accommodate such kind of specificities. More than that, if our CFO improves its cybersecurity experience in the coming future, once the value is updated in its end user instance, our framework will adapt itself to the new context. Some risks will then be removed as the same SWRL rules will consider that some past risks are no longer justified. The reasoner will identify inconsistencies over the data anytime.

We are then moving forward from the original (static) risk assessment approach used by the organization until today (see Equation 15) to a more feasible, meaningful, more effective, realistic, complete and dynamic approach (by implementing the Equation 16).

For this evaluation, we limited the implementation mostly to "Deliberated Malicious SW Distribution Risks" but also to other type of non IT related Risks, like Strategic Risks of type "Bad Reputation Risks" because their relationships in case of potential data leak when unauthorized access to classified data take place.

Again, due to this semantic approach, it is easy to follow risks relationships from a service / project to the users and vice versa, even to connect its network-traffic to a specific service in risk dynamically.

In our example, IP 192.168.1.10 belongs to our CFO user. In case it executes the fake Flash Update, a new rule would be able for example to create an automated incident response with all the related context (security event, user potentially infected, dateTimeStamp, Services Affected, etc.). Without a malware analysis yet, or without more cyber threat intelligence data, the incident could not have still the highest severity score (we still do not know if the dropped file is a Malware and their motivation) however it is a malicious pattern detected. But, when the binary file is identified as malware, the incident severity would then get the highest score as classified data could have been compromised already by using stolen credentials.

One of the main challenges in cybersecurity it is to work against unknown or emerging threats (e.g., APT) having real time visibility about our risk exposure along the time. Another challenge is to clearly define when a security event becomes an incident (preventive or reactive). By using our framework (Figure 89) we have all the needed expressiveness to better know what is really happening along the time, in this case, we know that there is a risk automatically identified of type deliberate malicious SW distribution which has been mitigated by two safeguards but at the same time, it was increased by different security events. We perfectly know the connection of this risk to the affected assets and services, and all information is consistent. We can query our model to know more about all the relationships and reasoner conclusions but we can also use interactive graphs to see all the relationships as shown in Figure 65.

Apart from operational level, our framework is able to work at tactical and strategic levels as seen in Figure 90 and Figure 91.

As seen in Figure 61 and Figure 83, our Dynamic Risk Assessment approach would also allow risks to be updated along the time due to special Security Events. The one shown in Figure 89, is an example of an event that will increase the risk once it is detected. Risks can be increased by changes on two factors or variables: impact or probability. In this case, the probability will change as something strange is close to become an incident related to that type of risk.

As seen in Figure 78, specific algorithms have been implemented to help the reasoner to propose specific **tactics**. In our case, once a security event is detected of a specific type, the reasoner will create an instance to share the detection algorithm as a risk intelligence sharing practice. The algorithm encoded in base64 will be shared to other offices or partners once an attack has been detected in our office. We are then transferring knowledge and detection capabilities by transferring the algorithm or SWRL rule inside another SWRL rule. They will be able to detect the same pattern once the SWRL rule shared is loaded in their systems even if attack IoC have changed. The only condition is that the TTP pattern remains the same as our rule is an algorithm to detect a TTP pattern. There will be different SWRL rules to detect different TTP patterns. The instance created automatically (prefix owl followed by a hash, means automatically generated instances) can be seen in Figure 90 together with the contents of the instance, in this case, it includes the base64 encoding version of rule seen in Figure 101.

Figure 90 Screenshot in Protégé tool of the instance created dynamically of class Information Sharing Control which at the same time is including the base64 encoded version of the SWRL attack detection TTP.

At the same time, in Figure 79, specific algorithms have been implemented to help the reasoner to propose specific **strategies** about HR Awareness Trainings. In our case, once a security event is detected of a specific type, the reasoner will create an instance of recommended awareness training to specific users which are receiving real time attacks and can be vulnerable to those attacks due to their cybersecurity experience. As seen in Figure 91 a new instance of the suggested control class (HR Related Security Awareness Education and Cyber capabilities control) is created dynamically considering the specific internal user that is recommended for the training.

Figure 91 Screenshot in Protégé tool of the instance created dynamically of class HR Related Security Awareness Education and Cyber capabilities control as recommended training just for the CFO by the reasoner due to real time security events.

9.5.2 Use Case 2 Results

9.5.2.1. Limitations of Our Proposal

We identified two types of limitations of our proposal:

- **Limitations of the proposed model**, based on the exchange of algorithms: limitations due to the novelty of the proposed model.
- **Technical limitations**: technical limitations encountered in our implementation, as well as other potential technical limitations.

With regard to the **limitations of the proposed model**, we realized that current versions of STIX™ patterning language still lacks of enough expressivity to define any kind of TTP. The proposed solution, could improve its expressivity. Then, in case that the STIX™ draft evolves into a more expressive standard language someday, it will become the main CTI de facto standard. Once that expressivity milestone is achieved, all stakeholders within the Cyber Threat Intelligence Exchange must use the same ontology or taxonomy.

The technical limitations are also subdivided in two subtypes:

- Technical limitations to support algorithms.
- Technical limitations within the blockchain CTI Exchange.

The limitations due to the novelty of the proposed model based on the exchange of algorithms have associated **technical limitations to support algorithms**. Once a new algorithm is received by a CTI Data consumer, their security devices, which are working in the IoC domain today, must be upgraded or replaced by new technologies which will be compatible with evolved standards. Reasoners today can be a perfect mate to fight against cyber threats, but they will need to interoperate with any security device under a common language.

In order to test the **technical limitations within the blockchain CTI Exchange** of our proposal, we decided to write (store) several CTI rules in bulk operations of different sizes, to check its effectiveness, its functionality, its limitations as well as the cost to store this type of CTI Data in the blockchain. On the other hand, we decided to read (query) several CTI rules in the same way, comparing the differences between writing and reading within the blockchain.

9.5.2.2. Technical Limitations within the Blockchain CTI Exchange

We deployed the same CTI Marketplace Smart contract in both environments (development and integration) to run tests. The main difference between them is the Gas Block Limit within Ropsten (8M). This limitation is agreed within the network by miners/nodes. On the other hand we had no such limitation in our development environment (Gas Block Limit it was above 672M by default).

transact to CTI_Marketplace.insertBulkEntityRules pending ...

<https://ropsten.etherscan.io/tx/0x37eb6303cf6f4076fe275b1d86ec8e8b14308e82290a0abec2d2ff588707f652>

[block:5611571 txIndex:25] from:0xdbb...f0325
to:CTI_Marketplace.insertBulkEntityRules(string[],string[],string,uint256[]) 0x763...4bf58
value:10 wei data:0x9ff...00001 logs:2 hash:0x37e...7f652

status	0x1 Transaction mined and execution succeed
transaction hash	0x37eb6303cf6f4076fe275b1d86ec8e8b14308e82290a0abec2d2ff588707f652
from	0xdbb99a3d73d0b3308850844bf8df6a0d46cf0325
to	CTI_Marketplace.insertBulkEntityRules(string[],string[],string,uint256[]) 0x763d10c63395e3e0521b28705333aa5ae394bf58
gas	456738 gas
transaction cost	456738 gas
hash	0x37eb6303cf6f4076fe275b1d86ec8e8b14308e82290a0abec2d2ff588707f652
input	0x9ff...00001
decoded input	{ "string[] CTI_rule": ["stix2:DomainName(?ref) ^ stix2:resolvesToRefs(?ref, ?ip) ^ stix2:IPv4Addr(?ip) -> stix2:IPv4Addr(?ip) ^ stix2:belongsToRefs(?ip, ?ref)",] }

Figure 92 Capture of Remix IDE of a transaction in Ropsten mined and executed successfully to share a CTI enrichment algorithm.

With regard to the gas used to write (store) CTI rules in the CTI Marketplace, it follows a linear growing pattern along the number of CTI rules included in the bulk transactions as seen in Figure 93. At the same time, the amount of gas used to store CTI rules is very high, especially when more than 30 or 40 rules are included in the same transaction. At this point we find the first limitation due to the Gas Block Limit of Ropsten (integration environment). We do not have this limitation at the development environment. Above 50 CTI Rules, the gas needed is above Ropsten Block limit of 8M so it will through an exception out of Development environment due to the transaction runs out of gas.

Figure 93 Graph of gas used to write (store) Intelligence algorithms in the Marketplace in bulk operations

If we set up a limitation programmatically, we can avoid running out of gas. The CTI Data producer will have a limitation to share more than 40 CTI rule in the same transaction however it will be able to store hundreds of CTI rules if using different transactions (each of them limited to 40).

We used the gas station²⁹ to calculate the cost of storage of our Marketplace which results are presented in Figure 94. As a result, each CTI rule will cost us approximately 4 euro cents. We have to differentiate the efficiency between an IoC (indicator of compromise) to a CTI rule as described by authors in [Riesco19a]. In principle less rules could detect more (matches by detecting patterns) than having individual IoC (e.g. DGA algorithm versus a specific domain). It means that we will need much less additional space to store rules than single IoC [Riesco19b].

²⁹ <https://ethgasstation.info/index.php>

Figure 94 Graph of gas cost to write (store) Intelligence algorithms in the Marketplace and the cost per rule in bulk operations

On the other hand, how much a company is willing to pay to demonstrate its authorship while sharing a rule to detect a malicious TTP pattern?. Using the blockchain, there are inherent benefits like using PKI, immutable registers, etc.

Beyond that, we think that CTI Data producers would be willing to invest 4 euro cents to share a rule that will generate incomes above that investment. Each time that rule is read by any CTI Data consumer, it will generate to its source direct economic benefits (50 percent of cash on taxes in the same transaction). More than that, we think that CTI Data producers would also be willing to invest in CTI tokens (due to the opportunity to get special investment conditions while sharing new CTI Data) as its value is expected to be growing if the Marketplace usage is also growing along the time. The Gas Block Limit of Ropsten but also the main-net's suggest that the **maximum number of rules is 40 CTI rules per transaction**. That transaction will cost about 0,007669 Ether (equivalent to 1,43 euros today).

Figure 95 Graph of gas cost to read (query) Intelligence algorithms in the Marketplace and the cost per rule in bulk operations

With regard to reading that data (role of CTI Data consumers), Figure 95 shows that the situation is very different. Although it also follows a linear growing pattern, the cost to write is near 21 times the cost to read. The minimum cost to read a single CTI rule is about 0,0019 euros when the minimum cost to write a single CTI rule was about 4 euro cents (0,04 euros). Our implementation requires a tax payment to read, so remember that it is not just a simple Ethereum "call" but a transaction due to the used modifiers.

As either the price of gas or the price of an Ether are variables, results can be higher (gas price or ETH price increase) or cheaper (gas price or ETH price decrease). At the time of this work the price of Ether was 186,08 euros and the gas cost was 1 Gwei (see Equation 21).

In Figure 96, we introduce a comparison between read and write operations from a single rule to a bulk insert of 700 rules in the same transaction.

Figure 96 Graph comparing the gas used between READ (querying) and WRITE (storing) Intelligence algorithms in bulk operations within the CTI Blockchain based Marketplace.

9.5.2.3. Benefits of the Proposal

CTI Rules (as seen in Figure 92) are written in STIX™ v2 [OasisSTIX2], OWL [W3C-OWL] ontologies [W3C-Ontology] and SWRL semantic rule language [W3C-SWRL] as proposed by authors in [Riesco19a].

Then, the exchange of this type of information (algorithms and rules), in the format described, will provide benefits [Riesco19b] like:

- **Reasoning and inference:** data and algorithms will be understood by semantic reasoners (and humans) [W3C-Reasoner], to infer new knowledge.
- **Encouraging participation of more producers:** it will avoid specific, ephemeral and static IoC to be shared, instead, the data will represent patterns in the form of rules (with variables).
- **Effective and efficient actionable intelligence:** it will allow a more efficient and effective sharing of knowledge (e.g. just one algorithm representing a pattern, can represent multiple combinations of IoC values of different families). Better algorithms, will bring greater benefits.

At the same time, despite the limitations detected in the number of bulk CTI rules that can be stored in a single transaction (see Figure 93 and Figure 94) or the number of CTI rules that can be read (see Figure 95) in the same transaction, there are great advantages and problems resolved by our approach, compared to legacy CTI exchange centralized models. Among them:

- **Identity and trusted sources:** the usage of crypto-wallets addresses, where only their private key owners can update, use, send or approve transactions, is giving a level of identity and trust on the source that goes beyond nicknames, user/password or even double factor authentication approaches. In addition to this, the blockchain

network functioning gives trust itself by default to untrusted sources, it combines the openness of the internet with the security of cryptography to give everyone a faster, safer way to verify key information and establish trust. As an example, transactions are verified by the network nodes to give trust of the whole transaction. Transactions are, in our example, calls or functions within a software program, a smart contract.

- **Authority:** sharing rules or algorithms beyond just sharing detections (IoC). It allows a better understanding on the level of knowledge of the source. Rating provided by consumers (in the form of a blockchain smart contract transaction) will be used as a de facto way to provide feedback.
- **Time-stamping:** time stamping feature can provided by the blockchain.
- **Internal and external consistency:** we use an ontology version of STIX™ to foster standardization as it is widespread in the CTI industry. By using an ontology, it would be easy for a semantic reasoner to understand the data, as an example, it will know if forbidden data is used by the CTI rule (e.g. TLP - Traffic Light Protocol restrictions).
- **Accounting:** what was shared, when it was shared and by whom. It is clear that blockchain technology will keep this immutable register which is very interesting for accounting and audit purposes.
- **Availability:** the decentralization provided by the blockchain guarantees as much availability as possible. Due to the fact that processing and validations are giving incentives to network nodes, it is very difficult to have any network outage. In case of a centralized infrastructure or even cloud computing, there are ongoing cost and investments needed that will never be cheaper than a decentralized infrastructure like blockchain.
- **Low cost:** a decentralized smart contract will have associated cost to real usage of the blockchain (gas used in mining calls, taxes applied to list CTI rules, etc.). No additional costs or investments are needed. On the other hand, the more usage of the CTI marketplace service, the more value and incentives are added to the system (CTI token value and balance). The value of the CTI token will be higher and more interesting for all stakeholders involved.
- **Attractive and Invest ready:** special conditions and incentives apply to have more CTI Data producers in order to solve the asymmetric approach. As suggested by [Vishik13], some economic incentives, as part of a business model, are associated to this proposal. Our proposal enables any wallet to invest in the CTI token which is based on ERC20 standard. Being cyber threat intelligence a growing valuable asset within cybersecurity international community, its digital representation or digital asset will also follow the same path. As an ICO (initial coin offering), our CTI token can be available for investment to anyone interested in it.
- **Secure:** some very interesting security features are inherited from the blockchain (e.g. Merkle trees, PKI, hashes, Proof of Work, Proof of Stake, Verification, Immutable registers, Transparent register, secure coding, modifiers, etc.).

In order to have a better understanding about how our specific contributions are addressing all of the open challenges today seen in Table 1 Threat Intelligence Sharing: open challenges and limitations of existing solutions, we created the Table 2 Solutions provided by our proposal to all open challenges and limitations identified in Table 1. The last column indicates the ID reference of the Table 1.

In the Ethereum blockchain the confidentiality of data and transactions is not guaranteed. As an example, the use of variables declared as private within the smart contract are still visible by using certain techniques.

Most of the consumers, which share the same motivation for the attacker, are potential new victims. All of them might be interested to update their defenses with new knowledge to improve their detection, prevention and response capabilities.

9.5.2.4. Discussion

As a result, however there is a cost to store and read CTI Data from the blockchain, it could be cheaper than deploying and maintaining a dedicated infrastructure. It will also have better availability by definition of a decentralized application.

The cost are now related to the real usage of the CTI Data, which is very interesting, as there are not ongoing cost for maintenance because the use of a decentralized infrastructure. This setup, provides interesting and economic incentives to every role involved, either short (by cash) or long term (by CTI tokens). Consumers, which are willing to pay, will pay for less ephemeral intelligence and higher value added (knowledge in the form of TTP detection algorithms beyond IoC). On the other hand, the grade of volatility and variability of either ETH or gas prices, could affect the behavior of users, fortunately an increment on the price does not mean a decrement of usage. Anyway, the CTI token could keep its value despite changes over the ETH prize.

The limitation of the storage per transaction and the needed gas for this type of (write) transactions can be improved if using a mixed approach (on and off chain), like for example using IPFS³⁰, Filecoin³¹ or Storj³².

In [Zheng18], authors propose a compression ratio of 0.0817 if using an IPFS-based blockchain data storage model to solve the problem of bitcoin ledger size. It had more than 200GB which was an issue for any node to join the network.

Intelligence of any Dynamic Risk Framework (as seen in [Riesco19a] [Riesco19b]), should be updated along the time. Any detection rule but also any mitigation, tactic o strategy rule that can be modeled in the format of a SWRL rule, would be part and eligible for our system CTI Data.

On the other hand, our proposal provides new economic incentives as suggested by [Vishik13], [Riesco19b] that will help to reduce the asymmetry between producers and consumers today. CTI token value would be increased within the usage of the CTI Data. The value added to the Marketplace, if this type of knowledge is shared, will increase the value of the CTI token. More incentives to all will be possible only if the CTI data has very high quality. The different parameters (e.g. tax, token prize equation, etc.) were modeled to simulate the potential market growth. New simulations are recommended to better adjust the different key parameters to better address potential requirements by the investors or any other role.

With regard to security, there is still room for improvement in some specific but important topics like confidentiality or privacy. Nothing in the Ethereum blockchain is private. The keyword private, is merely an artificial construct of the Solidity language. Web3's `getStorageAt(...)` as seen in [EthereumStorage] can be used to read anything from (private) storage. It can be tricky to read what you want though, since several optimization rules and techniques, are used to compact the storage as much as possible.

³⁰ <https://ipfs.io/>

³¹ <https://filecoin.io/filecoin.pdf>

³² <https://storj.io/whitepaper/>

CHAPTER 10. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

10.1. CONCLUSIONS

Today, risk assessment (RA) and management (RM) are mostly manual processes performed once per year by different experts based on their personal opinions. On the other hand, any entity is exposed to cybersecurity threats every day. Unfortunately these threats are not taken into consideration dynamically into the organization risk calculation. Risk exposure level calculation, countermeasure's responses, projects or related investment plans should be updated dynamically and proportionally to the threat level and risk exposure of each organization along the time. There should not be treated as a static annual review process (by auditors).

We developed a formal model based on standards to connect real time threats to risk calculation and risk management processes which also provide better automation, enrichment, detection capabilities and simplicity by using standards STIX™ [OasisSTIX1], OWL [W3C-OWL], a reasoner [W3C-Reasoner] and SWRL [W3C-SWRL].

This work present the first practical DRA/DRM approach applied to up to date threat and risk processes of an international reference entity, a national CSIRT.

We have selected a real publicly known attack for the implementation, as we consider it a good example to test our proposal on leading organizations that could easily be impacted by this type of attack due to its nature. This type of attacks are very difficult to detect even by leading organizations. We implemented behavioral pattern rules in SWRL to detect and update our risk level exposure accordingly with so much expressivity to understand what is really happening either by humans or machines. At the same time we demonstrated how a specific security event could trigger different actions beyond the operational level, like the tactical and strategic levels. In our case, at tactical level, the same attack produces an automatic risk intelligence sharing (share of TTP detection algorithm but not specific IoC) as a tactic to avoid bigger impact of a potential campaign against other remote offices or partners even if the IoC are changed during the attack. As a future research direction, there is a need to improve incentives for intelligence sharing (IoC or algorithms). At strategic level, specific awareness training sessions were identified to those victims involved in the attack which at the same time has poor cybersecurity knowledge.

The Cybersecurity Threat Intelligence (CTI) data is considered, one of the most valuable assets of any organization, to better detect, prevent and response to cybersecurity threats on time. Its value is related to the quality, understood as the timing (when and how fast is available), the reliability and accuracy of the data. Because of that, there is a very high demand of CTI data, however, there is a limited size of providers, compared to the demand size. Entities are using different taxonomies without enough expressivity to define complex relationships, which are needed to create context-aware or behavioral rules. STIX™ is a promising standard but it still lacks of semantics. Furthermore, users are reluctant to share. Trust is one of the main reasons behind, but there are much more reasons (see Table 1).

As one of the main building blocks of our proposal, we propose an enhanced version of the Information sharing module. To have a better context, we need to have as much reliable intelligence as possible. It is impossible to fight alone against all threats, so we propose the use of a Peer to Peer Decentralized "Intelligence Sharing" Network that, at the same time, must address all open challenges and existing limitations of legacy approaches.

This paper presents a new model for this kind of P-t-P modules, to provide Cybersecurity Intelligence Exchange, based on Blockchain. It provides new economic incentives to all roles involved (to overcome one of the main issues as of today), as well as

an enhanced version of the peers. In order to operate, share and consume semantic advanced intelligence automatically, the semantic reasoner of the DRM Framework is considered a key and a powerful building block, also for this tasks. It will contribute to all transactions by its reasoning capabilities, ranging from the role used in a specific transaction, to the inference of new knowledge at any level (e.g. attribution), or even a simple TLP eligibility check before a new data is shared.

A new CTI token is also provided, as a digital representation of the CTI Data asset. It will add more value (benefit and market growth), especially for CTI Data producers, but also to Investors. We made some Montecarlo simulations as part of our proposal, to calculate and optimize certain key parameters. As a result, the system is more dynamic and it provides more value back, to more stakeholders, than current approaches (OSINT, Feeders, and ISAC). We evaluated the dynamics of our model through an experimentation. It allowed us to validate the benefits but also it helped us to detect potential limitations, especially with regard to storage, querying and processing transactions costs.

As a result, our contribution is providing a coherent solution at the same time to all of the open challenges described in this work (see Table 1 and Table 2).

10.2. DISSEMINATION OF RESULTS

There are several positive outcomes related to the contributions of this thesis. Some of them have been published already:

- Riesco, R., Villagr, V.A. **Leveraging cyber threat intelligence for a dynamic risk framework**. In Int. J. Inf. Secur. (2019)
DOI: <https://doi.org/10.1007/s10207-019-00433-2>
- Riesco, R. et al. **Cybersecurity Threat Intelligence Knowledge Exchange based on blockchain**. In Telecommunication Systems (2019)
DOI: <https://doi.org/10.1007/s11235-019-00613-4>
- Irene Romero, Raul Riesco, Francisco Barea, Victor A. Villagr. **Arquitectura de gestin dinmica de riesgos basada en ontologas y reglas de comportamiento**. Actas XV Reunin Espaola sobre Criptologa y Seguridad de la informacin (RECSI). Granada (Espaa). Octubre 2018. ISBN: 978-84-09-02463-6. p86-91
URI: <https://nesq.ugr.es/recsi2018/docs/ActasXVRECSI.pdf>
- Fernando Monje, Cristina Galvn, Raul Riesco, Vctor A. Villagr. **Sistema de Clculo de Riesgo Dinmico en Dominios Administrativos Basado en Ontologas**. Actas de las V Jornadas Nacionales de Investigacin en Ciberseguridad (JNIC 2019). Cceres (Espaa). 5-7 Junio 2019. ISBN 978-84-09-12121-2
URI: <http://dehesa.unex.es/handle/10662/9443 p177-184>
- Vctor A. Villagr, Mario Vega-Barbas, Xavier Larriva-Novo, Manuel lvarez-Campana, Ral Riesco. **Enhancing Cyber Situation Awareness by Using Visual Analytics of Heterogeneous Remote Cybersecurity Sensors Data**. Remote Sensing (2019). Confirmation pending.

10.3. FUTURE RESEARCH DIRECTION

At the same time, we identified some future research directions and limitations that were encountered during the evaluation of our implementation:

- With regard to the ontologies, a new version of STIX™ (v2.1) is coming soon. A deep analysis considering all potential changes, must be considered in the STIX2.owl ontology.
- All modules can be improved in terms of scope but also in terms of efficiency.
- The ontology must be adapted (reduced or simplified) to the resources of the reasoner needed in production environments. New testing and decisions must be taken to adjust the DRM Framework.
- New intelligence (types and or algorithms) should be developed and tested against most of the attack types [MITRE-ATTCK] and / or TTPs. A special case of intelligence, like guessing potential attribution of cyber-attacks, is considered one of the most interesting and challenging.
- Specific parses are needed to overcome the lack of semantic support by HW/SW appliances today.
- As an alternative to current Protégé plugins, new interactive Visualization modules should be analyzed or developed.
- The use of web services and / or endpoints (based on OWL-S) can be interesting to research.
- In the networked P-to-P information sharing solution, we were focused in the utility function's parameters seen in Equation 12. Those were especially related to the incentives to have more CTI Data producers and the reliability of the data. We made experiments and simulations to analyze some key parameters, like the breakeven for this role (e.g. the limit to get favorable economic conditions). However, the model and all their variables, should be improved and fine-tuned, running additional simulations, adding special profitability requirements of investors and its dependency with the profitability of producers and any other type of roles. The impact of malicious nodes has to be also introduced in our research.
- With regard to the CTI token value (see Equation 14), we depend on the value of Ether today, which is continuously fluctuating due to the trading of crypto currencies. We will work to give more stability alternatives to the CTI token, as well as considering the impact of the limitation in the number of tokens NT (as proof-of-stake) along the time.
- There are also technical limitations in our model. The technology and standards should evolve together to work in this specific enhanced version of the peers and the network. Some research about the potential impact, dependencies and transition cost are recommended. We propose to research in potential transition and coexistence scenarios.
- In addition to this, most of the security limitations encountered are inherent to the blockchain technology itself, like all the confidentiality and privacy issues detected. Some workarounds can be implemented nowadays however they will provide less efficient coding as well as higher storing or processing costs. New research will be focused to keep the efficiency of the model, while providing more confidentiality and privacy by default.

REFERENCES

A

- [Alrwais16] Sumayah Alrwais, Kan Yuan, Eihal Alowaisheq, Xiaojing Liao, Alina Oprea, XiaoFeng Wang, and Zhou Li. 2016. ***Catching predators at watering holes: finding and understanding strategically compromised websites***. In Proceedings of the 32nd Annual Conference on Computer Security Applications (2016).

B

- [Beckers12] K. Beckers, S. Faßbender, J.-C. Küster and H. Schmidt. ***A pattern-based method for identifying and analyzing laws***. In Requirements Engineering: Foundation for Software Quality (REFSQ'12), LNCS 7195, pp. 256–262. Springer, 2012
- [Bendovschi15] A. Bendovschi. ***Cyber-Attacks –Trends, Patterns and Security Countermeasures***. In 7th International Conference on Financial Criminology (2015).
- [Bianco14] Bianco, D., ***The Pyramid of Pain***.
<http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html> (2014).

C

- [CVSS] M. Schiffman. ***Common Vulnerability Scoring System (CVSS)***, In <http://www.first.org/cvss/cvss-guide> (2011).

D

- [deFuentes17] De Fuentes, JM., González-Manzano, L., Tapiador, J., Peris-Lopez, P. ***PRACIS: privacy-preserving and aggregatable cybersecurity information sharing***. In Comput Secur. 2017;69:127-141.
<https://doi.org/10.1016/j.cose.2016.12.011>

E

- [EC16] European Commission and European Parliament. ***NIS Directive***.
<http://data.europa.eu/eli/dir/2016/1148/oj>
- [Ekelhart07] Ekelhart, A., Fenz, S., Klemen, M.; Weippl, E. ***Security Ontologies: Improving Quantitative Risk Analysis***. In Proceedings of the 40th Hawaii International Conference on System Sciences (2007).
- [Erbacher15] Erbacher, R.F. ***Ontology-based Adaptive Systems of Cyber Defense***. In Semantic Technology for Intelligence, Defense and Security Conference, Fairfax, VA, (2015).
- [ERC20] ERC20 token. ***IEIP20 - ERC20 standard token***.
<https://eips.ethereum.org/EIPS/eip-20>
- [EthereumStorage] Ethereum contract storage. ***How to read Ethereum contract storage***.

<https://medium.com/aigang-network/how-to-read-ethereum-contract-storage-44252c8af925>

F

- [Fenz08] S.Fenz. **Ontology-based Generation of IT-Security Metrics**. In Proceedings of the 41st Hawaii International Conference on System Sciences (2008).
- [Fensel01] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness and P. F. Patel-Schneider. **OIL: an ontology infrastructure for the Semantic Web**. In *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 38-45, March-April 2001. DOI: <https://doi.org/10.1109/5254.920598>
- [Fernández-López97] M. Fernández-López, A. Gómez-Perez, N. Juristo. **Methontology: From ontological art towards ontological engineering**. Proceedings of the Spring Symposium Series, pp. 33-40. 1997
- [Fernández-López99] M. Fernández-López. **Overview of Methodologies for Building Ontologies**. In IJCAI99 Workshop on Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends, Stockholm. 1999
- [Fireeye19] Fireeye Mandiant services | special report **M-Trends 2019 (2019)**
<https://www.fireeye.com/current-threats/annual-threat-report/mtrends.html>

G

- [Gao18a] Gao, Peng et al. **AIQL: Enabling Efficient Attack Investigation from System Monitoring Data**. In USENIX Annual Technical Conference (2018).
- [Gao18b] Gao, Peng et al. **SAQL: A Stream-based Query System for Real-Time Abnormal System Behavior Detection**. In USENIX Security Symposium (2018).
- [Goluch08] G. Goluch, A. Ekelhart, S. Fenz, S. Jakoubi, S.Tjoa and T. M. **Integration of an Ontological Information Security Concept in Risk-Aware Business Process Management**. In Proceedings of the 41st Hawaii International Conference on System Sciences (2008).
- [Gómez-Pérez96] A. Gómez-Pérez. **A framework to verify knowledge sharing technology**. Expert Systems with Application, Vol. 11 (4), pp. 519-529. 1996
- [Graf18] Graf, R., King, R. **Neural network and blockchain based technique for cyber threat intelligence and situational awareness**. In International Conference on Cyber Conflict, CYCON (2018)
DOI: <https://doi.org/10.23919/CYCON.2018.8405028>
- [Gruber95] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. Padua workshop on Formal Ontology, 1993. International Journal of Human-Computer Studies, Vol. 43 (4-5), pp. 907-928. 1995
- [Grüninger] M. Grüninger, M. S. Fox. **Methodology for the Design and Evaluation of Ontologies**. Proceedings of the Workshop on Basic Ontological Issues in knowledge Sharing. IJCAI (1995)

H

- [Hernandez-Ardieta13] Jorge L. Hernandez-Ardieta, Juan E. Tapiador and Guillermo Suarez-Tangil. **Information Sharing Models for Cooperative Cyber Defence**. 2013 5th

International Conference on Cyber Conflict (CyCon'13). K. Podins, J. Stinissen, M. Maybaum (Eds.) 2013 © NATO CCD COE Publications, Tallinn.

- [Herzog07] A. Herzog, N. Shahmehri, C. Duma. **An ontology for information security**. In International Journal of Information Security and Privacy (2007).
- [Hutchins10] Hutchins, Eric Michael et al. **Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains** (2010).

I

- [ISC2-18] (ISC)2. **Cybersecurity Skills Gap**.
<https://www.isc2.org/Research/Workforce-Study>
- [ISO/IEC27000:2018] ISO/IEC 27000:2018, **Information technology -- security techniques -- information security management systems** -- overview and vocabulary (2018).
- [ISO/IEC27005:2008] ISO/IEC 27005:2008, **Information technology - Security techniques and information security risk management** (2008).
- [ISO/IEC31000:2018] ISO 31000:2018, **Risk Management - Guidelines** (2018).

L

- [Leszczyna19] Leszczyna, R., Wróbel, M.R. **Threat intelligence platform for the energy sector**. In Wiley online library (2019)
DOI: <https://doi.org/10.1002/spe.2705>
- [Licklider68] J.C.R. Licklider, Robert Taylor. **The Computer as a Communication Device**. Science and Technology 76, pp. 21-31 April 1968.
- [LopezDeVergara03] J. E. López de Vergara. **Especificación de Modelos de Información de Gestión de Red Integrada Mediante el Uso de Ontologías y Técnicas de Representación del Conocimiento**. Tesis doctoral (PhD Thesis), Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid (UPM). 2003

M

- [Mateos10] V. Mateos, V. A. Villagrà, F. Romero. **Ontologies-Based Automated Intrusion Response System**. In Computational Intelligence in Security for Information Systems 2010, pp. 99-106 (2010).
- [Meszaros17] J. Meszaros y A. Buchalceva. **Introducing OSSF: A Framework for Online Service Cybersecurity Risk Management**. In Computers and Security. vol.65, pp. 300-313 (2017).
- [MISP] MISP Open Source Platform
<https://www.misp-project.org/>
- [MISP16] Wagner, C., Dulaunoy, A., Wagener, G., and Iklody, A. **MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform**. In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security (WISCS '16). ACM, New York, NY, USA, 49-56.
DOI: <https://doi.org/10.1145/2994539.2994542>
- [MITRE-ATTCK] MITRE ATT&CK
<https://attack.mitre.org/>

- [Mozzaquatro18] Mozzaquatro, Bruno Augusti et al. **An Ontology-Based Cybersecurity Framework for the Internet of Things**. In Sensors (Basel, Switzerland) vol. 18,9 3053. 12 Sep. (2018).

N

- [Nath16] Nath,I. **Data Exchange Platform to Fight Insurance Fraud on Blockchain**. In IEEE 16th International Conference on Data Mining Workshops (ICDMW) (2016)
DOI: <https://doi.org/10.1109/ICDMW.2016.0121>
- [NATO001] NATO OSINT Handbook. NATO Open source intelligence handbook.
https://www.academia.edu/4037348/NATO_Open_Source_Intelligence_Handbook
- [NIST16] NIST Guide to CTI sharing. **Guide to Cyber Threat Information Sharing**. In Special Publication 800-150 (2016).
<http://dx.doi.org/10.6028/NIST.SP.800-150>
- [Noy01] N. F. Noy, D. L. McGuinness. **Ontology Development 101: A Guide to Creating Your First Ontology**. Stanford Knowledge Systems Laboratory. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. 2001

O

- [OasisSTIX1] OASIS. **STIX™ White paper**. In :
https://stixproject.github.io/about/STIX_Whitepaper_v1.1.pdf
- [OasisSTIX2] OASIS. **STIX™ 2.0 specifications**. In:
<https://oasis-open.github.io/cti-documentation/>
- [OasisSTIX-Campaign] OASIS. **Campaigns by STIX™**. In:
<https://stixproject.github.io/getting-started/whitepaper/#campaigns>
- [OasisSTIX-Incidents] OASIS. **Incidents by STIX™**. In:
<https://stixproject.github.io/getting-started/whitepaper/#incidents>
- [OasisSTIX-TTP] OASIS. **TTP (Techniques, Tactics and Procedures” by STIX™**. In
<https://stixproject.github.io/getting-started/whitepaper/#tactics-techniques-and-procedures-ttp>
- [Obrst12] Obrst, L. et al. **MITRE - Developing an Ontology of the Cyber Security Domain**. In MITRE (2012).
- [OpenCTI] **Open CTI Platform**
<https://opencti-platform.github.io/docs/getting-started/introduction>
- [OSINTArmy12] OSINT. **ATP 22-2.29. Open source intelligence headquarters departments of the army**
<https://fas.org/irp/doddir/army/atp2-22-9.pdf>

P

- [Pavel16] Pavel Ciaian, Miroslava Rajcaniova, d'Artis Kancs (2016). **The economics of BitCoin price formation**. In Applied Economics, 48:19, 1799-1815.
DOI: <http://doi.org/10.1080/00036846.2015.1109038>
- [Polyswarm] Polyswarm. **Polyswarm Decentralized Threat Detection marketplace**.

<https://polyswarm.io>

- [Ponemon19] IBM and Ponemon Institute. **Cost of a Data Breach Report (2019)**
- [Poolsappasit12] Poolsappasit, N.; Dewri, R. Ray, I. **Dynamic Security Risk Management Using Bayesian Attack Graphs**. In IEEE Trans. Dependable Sec. Comput. 9 (1), 61-74 (2012).
- [Protégé] Stanford University. **Protégé**.
<https://protege.stanford.edu/>

Q

- [Qamar17] S. Qamar, Z. Anwar, M. Ashiqur Rahman, E. Al-Shaer, Bei-Tseng Chu. **Data-driven analytics for cyber-threat intelligence and information sharing**. In Computers and Security, vol.67 (2017).

R

- [Ravsan18] Ravsan, S.K. (2018). **Utility tokens: discussion, economic model and simulation in R**. In Hackernoon (html).
<https://hackernoon.com/utility-tokens-discussion-economic-model-and-simulation-in-r-798c0ff3d26c>
- [RDFMT01] W3C Working Draft (2001). **RDF Model Theory**.
<http://www.w3.org/TR/rdf-mt/>
- [Refsdal14] A. Refsdal, Ø. Rideng, B. Solhaug and K. Stølen. **Divide and conquer - Towards a notion of risk encapsulation**. In Advances in Engineering Secure Future Internet Services and Systems. LNCS 8431, Springer, 2014
- [Riesco19a] Riesco, R., Villagra, V.A. **Leveraging cyber threat intelligence for a dynamic risk framework**. In International Journal of Information Security (2019)
DOI <https://doi.org/10.1007/s10207-019-00433-2>
- [Riesco19b] Riesco, R., Larriva-Novo, X., Villagra, V.A. **Cybersecurity threat intelligence knowledge exchange base don blockchain**. In Telecommunication Systems.
DOI: <https://doi.org/10.1007/s11235-019-00613-4>
- [Ring14] Ring, T. **Threat intelligence: why people don't share**. In Comput Fraud Secur. 2014 (3):5-9.
[https://doi.org/10.1016/S1361-3723\(14\)70469-5](https://doi.org/10.1016/S1361-3723(14)70469-5)

S

- [Sauerwein17] Sauerwein, C. et al. **Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives**. In Wirtschaftsinformatik (2017)
<https://www.wi2017.ch/images/wi2017-0188.pdf>
- [Singapogu12] Singapogu, S. et al. **Security Ontologies for Modeling Enterprise Level Risk Assessment**. In 2012 Annual Computer Security Applications Conference, Orlando (2012).
- [Skopik16] Skopik, F, Settanni, G, Fiedler, R. **A problem shared is a problem halved: a survey on the dimensions of collective cyber defense through security information sharing**. In Comput Secur. 2016 ;60:154-176.
<https://doi.org/10.1016/j.cose.2016.04.003>

- [Smart contract] Smart Contracts. **Smart contracts in Ethereum**.
<http://ethdocs.org/en/latest/contracts-and-transactions/contracts.html>
- [Solhaug14] B. Solhaug and F. Seehusen. **Model-driven risk analysis of evolving critical infrastructures**. In Journal of Ambient Intelligence and Humanized Computing, 5(2):187-204, 2014
- [Staab01] R. Staab, R. Studer, H. P. Schnurr, Y. Sure. **Knowledge Processes and Ontologies**. IEEE Intelligent Systems, Vol. 16 (1), pp. 26-34. 2001
- [Sugumaran02] V. Sugumaran, V.C. Storey. **Ontologies for conceptual modeling: their creation, use and management**. In Data & Knowledge Engineering, Vol. 42 (3), pp. 251-271. 2002
- [Syed16] Syed, Z. et al. **UCO - Unified Cybersecurity Ontology**. In The Workshops of the Thirtieth AAAI Conference on Artificial Intelligence. Artificial Intelligence for Cyber Security: Technical Report WS-16-03 (2016)
- [Symantec19] Symantec. **Internet Security Threat Report 2019**.
<https://www.symantec.com/en/uk/security-center/threat-report>

T

- [Tosh18] Tosh, D., Sengupta, S., Kamhoua, CA., Kwiat, KA. **Establishing evolutionary game models for CYBER security information EXchange (CYBEX)**. In J Comput Syst Sci. (2018) ;98:27-52.
<https://doi.org/10.1016/j.jcss.2016.08.005>
- [Tounsi18] Tounsi, W., Rais, H. **A survey on technical threat intelligence in the age of sophisticated cyber attacks**. In Comput Secur. 2018;72:212-233.
DOI: <https://doi.org/10.1016/j.cose.2017.09.001>
- [Tran13] L. M. S. Tran, B. Solhaug and K. Stølen. **An approach to select cost-effective risk countermeasures**. In Proc. 27th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy (DBSec'13). LNCS 7964, pp. 266-273, Springer, 2013

U

- [Ulicny14] Ulicny B.E., Moskal J.J., Kokar M.M., Abe K., Smith J.K. (2014) **Inference and Ontologies**. In Kott A., Wang C., Erbacher R. (eds) Cyber Defense and Situational Awareness. Advances in Information Security, vol 62. Springer, Cham
- [Uma2013] M. Uma and G. Padmavathi. **A Survey on Various Cyber Attacks and Their Classification**. (2011)
- [Uschold95] M. Uschold, M. King. **Towards a Methodology for Building Ontologies**. Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing held in conjunction with IJCAI (1995)
- [Uschold96] M. Uschold, M. Grüninger. **Ontologies: Principles, Methods and Applications**. Knowledge Engineering Review, Vol 11 (2), pp. 93-155. 1996

V

- [Vergara10] JEL de Vergara et al. **A Semantic Web Approach to Share Alerts among Security Information Management Systems**. In Communications in Computer and Information Science 72:14-25 (2010).

- [Viehmann12] J. Viehmann. **Reusing risk analysis results – An extension for the CORAS risk analysis method.** In: Proc. 4th International Conference on Information Privacy, Security, Risk and Trust (PASSAT). pp. 742–751. IEEE, 2012
- [Vishik13] Vishik C., Sheldon F., Ott D. **Economic Incentives for Cybersecurity: Using Economics to Design Technologies Ready for Deployment.** In Reimer H., Pohlmann N., Schneider W. (eds) ISSE 2013 Securing Electronic Business Processes. Springer Vieweg, Wiesbaden (2013);133-147.
https://doi.org/10.1007/978-3-658-03371-2_12

W

- [W3C-Inference] W3C. **Inference.**
<https://www.w3.org/standards/semanticweb/inference>
- [W3C-Ontology] W3C. **Ontology.**
<https://www.w3.org/standards/semanticweb/ontology>
- [W3C-OWL] W3C. **OWL.**
<https://www.w3.org/OWL/>
- [W3C-OWL2] W3C. **OWL2**
<https://www.w3.org/TR/owl2-overview/>
- [W3C-Pellet] W3C. **Pellet reasoner.**
<https://www.w3.org/2001/sw/wiki/Pellet>
- [W3C-Reasoner] W3C. **Reasoner.**
<https://www.w3.org/2001/sw/wiki/Category:Reasoner>
- [W3C-SWRL] W3C. **SWRL Semantic Web Rule Language.**
<https://www.w3.org/Submission/SWRL/>

Z

- [Zhang17] J. Zhang, J. Yang and J. Li. **When Rule Engine Meets Big Data: Design and Implementation of a Distributed Rule Engine Using Spark.** In IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), San Francisco, CA (2017).
- [Zheng18] Zheng, Q., Li, Y., Chen, P. and Dong, X. **An Innovative IPFS-Based Storage Model for Blockchain.** In IEEE/WIC/ACM International Conference on Web Intelligence, WI (2018).
<https://doi.org/10.1109/WI.2018.000-8>

ANNEX A. CYBER-OBSERVABLES STIX™ V2

A.I. ARTIFACT

The Artifact Object permits capturing an array of bytes (8-bits), as a base64-encoded string, or linking to a file-like payload. The size of the base64-encoded data captured in the `payload_bin` property MUST be less than or equal to 10MB.

One of `payload_bin` or URL MUST be provided. It is incumbent on object creators to ensure that the URL is accessible for downstream consumers. If a URL is provided, then the hashes property MUST contain the hash of the URL contents.

```
{
  "0": {
    "type": "artifact",
    "mime_type": "image/jpeg",
    "payload_bin": "VBORw0KGgoAAAANSUgEUgAAADI== ..."
  }
}
```

Example 24. Basic image artifact.

A.II. AS - AUTONOMOUS SYSTEM

The AS object represents the properties of an Autonomous System (AS).

```
{
  "0": {
    "type": "autonomous-system",
    "number": 15139,
    "name": "Slime Industries",
    "rir": "ARIN"
  }
}
```

Example 25. Basic AS Object.

A.III. DIRECTORY

The Directory Object represents the properties common to a file system directory.

```
{
  "0": {
    "type": "directory",
    "path": "C:\\Windows\\System32"
  }
}
```

Example 26. Basic directory.

A.IV. DOMAIN NAME

The Domain Name represents the properties of a network domain name.

```
{{
  "0": {
    "type": "domain-name",
    "value": "example.com",
  }
}}
```

```

    "resolves_to_refs": [
      "1"
    ]
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  }
}

```

Example 27. Basic FQDN.

A.V. EMAIL ADDRESS

The Email Address Object represents a single email address.

```

{
  "0": {
    "type": "email-addr",
    "value": "john@example.com",
    "display_name": "John Doe"
  }
}

```

Example 28. Basic email address

A.VI. EMAIL MESSAGE

The Email Message Object represents an instance of an email message, corresponding to the internet message format described in [RFC5322] and related RFCs.

Header field values that have been encoded as described in section 2 of [RFC2047] MUST be decoded before inclusion in Email Message Object properties.

For example, “*this is some text*” MUST be used instead of “*=?iso-8859-1?q?this=20is=20some=20text?=*”. Any characters in the encoded value which cannot be decoded into Unicode SHOULD be replaced with the 'REPLACEMENT CHARACTER' (U+FFFD). If it is necessary to capture the header value as observed, this can be achieved by referencing an Artifact Object through the `raw_email_ref` property.

```

{
  "0": {
    "type": "email-addr",
    "value": "joe@example.com",
    "display_name": "Joe Smith"
  },
  "1": {
    "type": "email-addr",
    "value": "bob@example.com",
    "display_name": "Bob Smith"
  },
  "2": {
    "type": "email-message",
    "from_ref": "0",
    "to_refs": [
      "1"
    ],
    "is_multipart": false,
    "date": "2004-04-19T12:22:23.000Z",
    "subject": "Did you see this?",

```



```

"additional_header_fields": {
  "Reply-To": [
    "steve@example.com",
    "jane@example.com"
  ]
}
}
}
}

```

Example 29. Simple email message with additional header properties

A.VII. FILE OBJECT

The File Object represents the properties of a file. A File Object MUST contain at least one of hashes or name.

```

{
  "0": {
    "type": "directory",
    "path": "C:\\Windows\\System32"
  },
  "1": {
    "type": "file",
    "hashes": {
      "SHA-256": "ceafbfd424be2ca4a5f0402cae090dda2fb0526cf521b60b60077c0f622b285a"
    },
    "parent_directory_ref": "0",
    "name": "qwerty.dll"
  }
}

```

Example 30. Basic file with parent directory.

A.VII.I. ARCHIVE FILE EXTENSION

The Archive File extension specifies a default extension for capturing properties specific to archive files. The key for this extension when used in the extensions dictionary MUST be archive-ext.

```

{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "ceafbfd424be2ca4a5f0402cae090dda2fb0526cf521b60b60077c0f622b285a"
    }
  },
  "1": {
    "type": "file",
    "hashes": {
      "SHA-256": "19c549ec2628b989382f6b280cbd7bb836a0b461332c0fe53511ce7d584b89d3"
    }
  },
  "2": {
    "type": "file",
    "hashes": {
      "SHA-256": "0969de02ecf8a5f003e3f6d063d848c8a193aada092623f8ce408c15bcb5f038"
    }
  },
  "3": {

```

```

    "type": "file",
    "name": "foo.zip",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "mime_type": "application/zip",
    "extensions": {
      "archive-ext": {
        "contains_refs": [
          "0",
          "1",
          "2"
        ],
        "version": "5.0"
      }
    }
  }
}

```

Example 31. Basic unencrypted ZIP Archive.

A.VII.II. NTFS FILE EXTENSION

The NTFS file extension specifies a default extension for capturing properties specific to the storage of the file on the NTFS file system. The key for this extension when used in the extensions dictionary **MUST** be `ntfs-ext`. An object using the NTFS File Extension **MUST** contain at least one property from this extension.

```

{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "ntfs-ext": {
        "alternate_data_streams": [
          {
            "name": "second.stream",
            "size": 25536
          }
        ]
      }
    }
  }
}

```

Example 32. NTFS File with a single alternate datastream.

A.VII.III. PDF FILE EXTENSION

The PDF file extension specifies a default extension for capturing properties specific to PDF files. The key for this extension when used in the extensions dictionary **MUST** be `pdf-ext`. An object using the PDF File Extension **MUST** contain at least one property from this extension.

```

{
  "0": {
    "type": "file",
    "hashes": {

```

```

    "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
  },
  "extensions": {
    "pdf-ext": {
      "version": "1.7",
      "document_info_dict": {
        "Title": "Sample document",
        "Author": "Adobe Systems Incorporated",
        "Creator": "Adobe FrameMaker 5.5.3 for Power Macintosh",
        "Producer": "Acrobat Distiller 3.01 for Power Macintosh",
        "CreationDate": "20070412090123-02"
      },
      "pdfid0": "DFCE52BD827ECF765649852119D",
      "pdfid1": "57A1E0F9ED2AE523E313C"
    }
  }
}

```

Example 33. Basic PDF file.

A.VII.IV. RASTER IMAGE FILE EXTENSION

The Raster Image file extension specifies a default extension for capturing properties specific to raster image files. The key for this extension when used in the extensions dictionary MUST be raster-image-ext. An object using the Raster Image File Extension MUST contain at least one property from this extension.

```

{
  "0": {
    "type": "file",
    "name": "picture.jpg",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "extensions": {
      "raster-image-ext": {
        "exif_tags": {
          "Make": "Nikon",
          "Model": "D7000",
          "XResolution": 4928,
          "YResolution": 3264
        }
      }
    }
  }
}

```

Example 34. Simple image file with EXIF data.

A.VII.V. WINDOWS™ PE BINARY FILE EXTENSION

The Windows™ PE Binary File extension specifies a default extension for capturing properties specific to Windows portable executable (PE) files. The key for this extension when used in the extensions dictionary MUST be windows-pebinary-ext.

```

{
  "0": {
    "type": "file",
    "hashes": {

```

```

"SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
},
"extensions": {
  "windows-pebinary-ext": {
    "pe_type": "exe",
    "machine_hex": "014c",
    "number_of_sections": 4,
    "time_date_stamp": "2016-01-22T12:31:12Z",
    "pointer_to_symbol_table_hex": "74726144",
    "number_of_symbols": 4542568,
    "size_of_optional_header": 224,
    "characteristics_hex": "818f",
    "optional_header": {
      "magic_hex": "010b",
      "major_linker_version": 2,
      "minor_linker_version": 25,
      "size_of_code": 512,
      "size_of_initialized_data": 283648,
      "size_of_uninitialized_data": 0,
      "address_of_entry_point": 4096,
      "base_of_code": 4096,
      "base_of_data": 8192,
      "image_base": 14548992,
      "section_alignment": 4096,
      "file_alignment": 4096,
      "major_os_system_version": 1,
      "minor_os_system_version": 0,
      "major_image_version": 0,
      "minor_image_version": 0,
      "major_subsystem_version": 4,
      "minor_subsystem_version": 0,
      "win32_version_value_hex": "00",
      "size_of_image": 299008,
      "size_of_headers": 4096,
      "checksum_hex": "00",
      "subsystem_hex": "03",
      "dll_characteristics_hex": "00",
      "size_of_stack_reserve": 100000,
      "size_of_stack_commit": 8192,
      "size_of_heap_reserve": 100000,
      "size_of_heap_commit": 4096,
      "loader_flags_hex": "abdbffde",
      "number_of_rva_and_sizes": 3758087646
    },
    "sections": [
      {
        "name": "CODE",
        "entropy": 0.061089
      },
      {
        "name": "DATA",
        "entropy": 7.980693
      },
      {
        "name": "NicolasB",
        "entropy": 0.607433
      },
      {
        "name": ".idata",

```

```

    "entropy": 0.607433
  }
]
}
}
}
}
}

```

Example 35. EXE file.

A.VIII. IPV4 ADDRESS OBJECT

The IPv4 Address Object represents one or more IPv4 addresses expressed using CIDR notation.

```

{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.0/24"
  }
}

```

Example 36. IPv4 CIDR block.

A.IX. IPV6 ADDRESS OBJECT

The IPv6 Address Object represents one or more IPv6 addresses expressed using CIDR notation.

```

{
  "0": {
    "type": "ipv6-addr",
    "value": "2001:0db8::/96"
  }
}

```

Example 37. IPv6 CIDR block.

A.X. MAC ADDRESS OBJECT

The MAC Address Object represents a single Media Access Control (MAC) address.

```

{
  "0": {
    "type": "mac-addr",
    "value": "d2:fb:49:24:37:18"
  }
}

```

Example 38. Typical MAC address.

A.XI. MUTEX OBJECT

The Mutex Object represents the properties of a mutual exclusion (mutex) object.

```

{
  "0": {
    "type": "mutex",
    "name": "__CLEANSWEEP__"
  }
}

```

```
}
```

Example 39. Malware Mutex.

A.XII. NETWORK TRAFFIC OBJECT

The Network Traffic Object represents arbitrary network traffic that originates from a source and is addressed to a destination. The network traffic MAY or MAY NOT constitute a valid unicast, multicast, or broadcast network connection. This MAY also include traffic that is not established, such as a SYN flood.

To allow for use cases where a source or destination address may be sensitive and not suitable for sharing, such as addresses that are internal to an organization's network, the source and destination properties (src_ref and dst_ref, respectively) are defined as optional in the properties table below. However, a Network Traffic Object MUST contain the protocols property and at least one of the src_ref or dst_ref properties and SHOULD contain the src_port and dst_port properties.

```
{
  "0": {
    "type": "domain-name",
    "value": "example.com"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "ipv4",
      "tcp",
      "http"
    ]
  }
}
```

Example 40. Basic HTTP network traffic.

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.3"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "tcp"
    ]
  }
}
```

Example 41. Basic TCP network traffic.

A.XII.I. HTTP REQUEST EXCEPTION

The HTTP request extension specifies a default extension for capturing network traffic properties specific to HTTP requests. The key for this extension when used in the extensions dictionary MUST be **http-request-ext**.

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.53"
  },
  "1": {
    "type": "network-traffic",
    "dst_ref": "0",
    "protocols": [
      "tcp",
      "http"
    ],
    "extensions": {
      "http-request-ext": {
        "request_method": "get",
        "request_value": "/download.html",
        "request_version": "http/1.1",
        "request_header": {
          "Accept-Encoding": "gzip,deflate",
          "User-Agent": "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113",
          "Host": "www.example.com"
        }
      }
    }
  }
}
```

Example 42. Basic HTTP Request.

A.XII.II. ICMP EXTENSION

The ICMP extension specifies a default extension for capturing network traffic properties specific to ICMP. The key for this extension when used in the extensions dictionary MUST be **icmp-ext**.

```
{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.9"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "203.0.113.5"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "protocols": [
      "icmp"
    ],
    "extensions": {
      "icmp-ext": {
        "icmp_type_hex": "08",
        "icmp_code_hex": "00"
      }
    }
  }
}
```



```

    }
  }
}

```

Example 43. Basic ICMP traffic.

A.XII.III. NETWORK SOCKET EXTENSION

The Network Socket extension specifies a default extension for capturing network traffic properties associated with network sockets. The key for this extension when used in the extensions dictionary MUST be **socket-ext**.

```

{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.2"
  },
  "1": {
    "type": "network-traffic",
    "src_ref": "0",
    "src_port": 223,
    "protocols": [
      "ip",
      "tcp"
    ],
    "extensions": {
      "socket-ext": {
        "is_listening": true,
        "address_family": "AF_INET",
        "protocol_family": "PF_INET",
        "socket_type": "SOCK_STREAM"
      }
    }
  }
}

```

Example 44. Basic stream SOCKET.

A.XII.IV. TCP EXTENSION

The TCP extension specifies a default extension for capturing network traffic properties specific to TCP. The key for this extension when used in the extensions dictionary MUST be **tcp-ext**. An object using the TCP Extension MUST contain at least one property from this extension.

```

{
  "0": {
    "type": "ipv4-addr",
    "value": "198.51.100.5"
  },
  "1": {
    "type": "ipv4-addr",
    "value": "198.51.100.6"
  },
  "2": {
    "type": "network-traffic",
    "src_ref": "0",
    "dst_ref": "1",
    "src_port": 3372,

```

```

    "dst_port": 80,
    "protocols": [
        "tcp"
    ],
    "extensions": {
        "tcp-ext": {
            "src_flags_hex": "00000002"
        }
    }
}
}
}

```

Example 45. Basic TCP traffic.

A.XIII. PROCESS OBJECT

The Process Object represents common properties of an instance of a computer program as executed on an operating system. A Process Object MUST contain at least one property (other than type) from this object (or one of its extensions).

```

{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
  },
  "1": {
    "type": "process",
    "pid": 1221,
    "name": "gedit-bin",
    "created": "2016-01-20T14:11:25.55Z",
    "arguments": [
      "--new-window"
    ],
    "binary_ref": "0"
  }
}

```

Example 46. Basic process.

A.XIII.I. WINDOWS™ PROCESS EXTENSION

The Windows™ Process extension specifies a default extension for capturing properties specific to Windows™ processes. The key for this extension when used in the extensions dictionary MUST be windows-process-ext. An object using the Windows™ Process Extension MUST contain at least one property from this extension.

```

{
  "0": {
    "type": "process",
    "pid": 314,
    "name": "foobar.exe",
    "extensions": {
      "windows-process-ext": {
        "aslr_enabled": true,
        "dep_enabled": true,
        "priority": "HIGH_PRIORITY_CLASS",
        "owner_sid": "S-1-5-21-186985262-1144665072-74031268-1309"
      }
    }
  }
}

```

```
}
}
}
```

Example 47. Basic Windows process.

A.XIII.II. WINDOWS SERVICE EXTENSION

The Windows Service extension specifies a default extension for capturing properties specific to Windows services. The key for this extension when used in the extensions dictionary **MUST** be windows-service-ext.

```
{
  "0": {
    "type": "file",
    "hashes": {
      "SHA-256": "35a01331e9ad96f751278b891b6ea09699806faedfa237d40513d92ad1b7100f"
    },
    "name": "sirvizio.exe"
  },
  "1": {
    "type": "process",
    "pid": 2217,
    "name": "sirvizio",
    "command_line": "C:\\Windows\\System32\\sirvizio.exe /s",
    "binary_ref": "0",
    "extensions": {
      "windows-service-ext": {
        "service_name": "sirvizio",
        "display_name": "Sirvizio",
        "start_type": "SERVICE_AUTO_START",
        "service_type": "SERVICE_WIN32_OWN_PROCESS",
        "service_status": "SERVICE_RUNNING"
      }
    }
  }
}
```

Example 48. Basic Windows service.

A.XIV. SW OBJECT

The Software Object represents high-level properties associated with software, including software products.

```
{
  "0": {
    "type": "software",
    "name": "Word",
    "cpe": "cpe:2.3:a:microsoft:word:2000:*:*:*:*:*:*:*",
    "version": "2002",
    "vendor": "Microsoft"
  }
}
```

Example 49. Typical SW instance.

A.XV. URL OBJECT

The URL Object represents the properties of a uniform resource locator (URL).

```
{
  "0": {
    "type": "url",
    "value": "https://example.com/research/index.html"
  }
}
```

Example 50. Typical URL.

A.XVI. USER ACCOUNT OBJECT

The User Account Object represents an instance of any type of user account, including but not limited to operating system, device, messaging service, and social media platform accounts.

```
{
  "0": {
    "type": "user-account",
    "user_id": "1001",
    "account_login": "jdoe",
    "account_type": "unix",
    "display_name": "John Doe",
    "is_service_account": false,
    "is_privileged": false,
    "can_escalate_privs": true,
    "extensions": {
      "unix-account-ext": {
        "gid": 1001,
        "groups": ["wheel"],
        "home_dir": "/home/jdoe",
        "shell": "/bin/bash"
      }
    }
  }
}
```

Example 51. Basic UNIX account.

A.XVII. WINDOWS™ REGISTRY KEY FILE

The Registry Key Object represents the properties of a Windows registry key.

```
{
  "0": {
    "type": "windows-registry-key",
    "key": "HKEY_LOCAL_MACHINE\\System\\Foo\\Bar"
  }
}
Registry key with values
{
  "0": {
    "type": "windows-registry-key",
    "key": "hkey_local_machine\\system\\bar\\foo",
    "values": [
      {
        "name": "Foo",
        "data": "qwerty",
        "data_type": "REG_SZ"
      }
    ]
  }
}
```

```

{
  "name": "Bar",
  "data": "42",
  "data_type": "REG_DWORD"
}
]
}
}

```

Example 52. Simple registry key.

A.XVIII. X.509 CERTIFICATE OBJECT

The X.509 Certificate Object represents the properties of an X.509 certificate, as defined by ITU recommendation X.509 [X.509]. An X.509 Certificate Object **MUST** contain at least one property (other than type) from this object.

```

"0": {
  "type": "x509-certificate",
  "issuer": "C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc, OU=Certification Services Division, CN=Thawte Server CA/emailAddress=server-certs@thawte.com",
  "validity_not_before": "2016-03-12T12:00:00Z",
  "validity_not_after": "2016-08-21T12:00:00Z",
  "subject": "C=US, ST=Maryland, L=Pasadena, O=Brent Baccala, OU=FreeSoft, CN=www.freesoft.org/emailAddress=baccala@freesoft.org"
}
}

```

Example 53. Basic X.509 certificate.

ANNEX B. CYBER THREAT INTELLIGENCE ONTOLOGY DETAILS

B.I. ATTACK PATTERN

"Attack Patterns are a type of TTP that describe ways that adversaries attempt to compromise targets.

Attack Patterns are used to help categorize attacks, generalize specific attacks to the patterns that they follow, and provide detailed information about how attacks are performed. An example of an attack pattern is "spear phishing": a common type of attack where an attacker sends a carefully crafted e-mail message to a party with the intent of getting them to click a link or open an attachment to deliver malware. Attack Patterns can also be more specific; spear phishing as practiced by a particular threat actor (e.g., they might generally say that the target won a contest) can also be an Attack Pattern.

The Attack Pattern SDO contains textual descriptions of the pattern along with references to externally-defined taxonomies of attacks such as CAPEC [CAPEC]. Relationships from Attack Pattern can be used to relate it to what it targets (Vulnerabilities and Identities) and which tools and malware use it (Tool and Malware)."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- external_references some stix:list
- killChainPhases some stix:list
- name only xsd:string
- type only {"attack-pattern"^^xsd:string}

where:

Table 16 Properties of Attack pattern class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of property field MUST be attack-pattern
external_references (optional)	list of type external-reference	A list of external references which refer to non-STIX information. This property MAY be used to provide one or more Attack Pattern identifiers, such as a CAPEC ID. When specifying a CAPEC ID, the source_name property of the external reference MUST be set to capec and the external_id property MUST be formatted as CAPEC-[id].
name (required)	string	A name used to identify the Attack Pattern.
description (optional)	string	A description that provides more details and context about the Attack Pattern, potentially including its purpose and its key characteristics.
killChainPhases (optional)	list of type kill-chain-phase	The list of Kill Chain Phases for which this Attack Pattern is used.

This class has the following relationships with other classes:

- createdByRef some Identity
- uses some Malware
- uses some Tool
- targets some Identity
- targets some Vulnerability
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.II. BUNDLE

"A Bundle is a collection of arbitrary STIX Objects and Marking Definitions grouped together in a single container.

A Bundle does not have any semantic meaning and Objects are not considered related by virtue of being in the same Bundle.

Bundle is not STIX Object, so it does not have any of the Common Properties other than the type and id properties."

Every instance of the class is characterized by the following properties:

- type only {"bundle"^^xsd:string}
- id only xsd:string
- spec_version only xsd:string
- objects some owl:Thing

Table 17 Properties of Bundle class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The type property identifies the type of object. The value of this property MUST be bundle
id (required)	identifier	An identifier for this Bundle. The id property for the Bundle is designed to help tools that may need it for processing, but tools are not required to store or track it. Consuming tools should not rely on the presence of this property or the ability to refer to bundles by ID.
spec_version (required)	string	The version of the STIX specification used to represent the content in this Bundle. This enables non-TAXII transports or

		<p>other transports without their own content identification mechanisms to know the version of STIX content.</p> <p>The value of this property MUST be 2.0 for bundles containing STIX Objects defined in this specification.</p>
objects (optional)	list of type <STIX Object> or marking- definition	Specifies a set of one or more STIX Objects. Objects in this list MUST be a STIX Object (SDO, SRO or Custom Object) or a Marking Definition object.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.III. CAMPAIGN

"A Campaign is a grouping of adversary behavior that describes a set of malicious activities or attacks that occur over a period of time against a specific set of targets."

Every instance of the class is characterized by the following properties:

- aliases some stix:list
- description some xsd:string
- firstSeen some xsd:dateTimeStamp
- lastSeen some xsd:dateTimeStamp
- name only xsd:string
- objective some xsd:string
- type only {"campaign"^^xsd:string}

where:

Table 18 Properties of Campaign class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be campaign
name (required)	string	A name used to identify the Campaign.
description (optional)	string	A description that provides more details and context about the Campaign, potentially including its purpose and its key characteristics.
aliases (optional)	list of type string	Alternative names used to identify this Campaign
first_seen (optional)	timestamp	The time that this Campaign was first seen.

		This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are earlier than the first seen timestamp, the object may be updated to account for the new data.
last_seen (optional)	timestamp	The time that this Campaign was last seen. This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are later than the last seen timestamp, the object may be updated to account for the new data.
objective (optional)	string	This property defines the Campaign's primary goal, objective, desired outcome, or intended effect — what the Threat Actor hopes to accomplish with this Campaign.

This class has the following relationships with other classes:

- attributedTo some IntrusionSet
- attributedTo some ThreatActor
- createdByRef some Identity
- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)
- targets some Identity
- targets some Vulnerability
- uses some AttackPattern
- uses some Malware
- uses some Tool
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.IV. COURSE OF ACTION

"A Course of Action is an action taken either to prevent an attack or to respond to an attack that is in progress."

Every instance of the class is characterized by the following properties:

- action some stix:reserved
- description some xsd:string
- name only xsd:string
- type only {"course-of-action"^^xsd:string}

where:

Table 19 Properties of Course of Action (CoA) class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be course-of-action
name (required)	string	A name used to identify the Course of Action
description (optional)	string	A description that provides more details and context about the Course of Action, potentially including its purpose and its key characteristics.
action (reserved)	RESERVED	RESERVED – To capture structured/automated courses of action.

This class has the following relationships with other classes:

- createdByRef some Identity
- mitigates some AttackPattern
- mitigates some Malware
- mitigates some Tool
- mitigates some Vulnerability
- objectMarkingRefs some MarkingDefinition
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V. CYBER OBSERVABLE

There are multiple types of CyberObservables. Each type is represented as a subclass. They are the following:

B.V.I. ARTIFACT

"The Artifact Object permits capturing an array of bytes (8-bits), as a base64-encoded string, or linking to a file-like payload."

Every instance of the class is characterized by the following properties:

- hashes some stix:hashes
- mimeType some xsd:string}
- payloadBin some xsd:hexBinary
- type some {"artifact"^^xsd:string}

where:

Table 20 Properties of Artifact cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be artifact.

mime_type (optional)	string	The value of this property MUST be a valid MIME type as specified in the IANA Media Types registry [Media Types].
payload_bin (optional)	binary	Specifies the binary data contained in the artifact as a base64-encoded string. This property MUST NOT be present if url is provided.
url (optional)	string	The value of this property MUST be a valid URL that resolves to the unencoded content. This property MUST NOT be present if payload_bin is provided.
hashes (optional)	hashes	Specifies a dictionary of hashes for the contents of the url or the payload_bin. This MUST be provided when the url property is present.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)
- redirection some URL
- url some URL

This class has no subclasses.

B.V.II. AUTONOMOUS SYSTEM

"The AS object represents the properties of an Autonomous Systems (AS)."

Every instance of the class is characterized by the following properties:

- name some xsd:string
- number only xsd:integer
- rir some xsd:string
- type some {"autonomous-system"^^xsd:string}

where:

Table 21 Properties of Autonomous System (AS) cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be autonomous-system.
number (required)	integer	Specifies the number assigned to the AS. Such assignments are typically performed by a Regional Internet Registry (RIR).
name (optional)	string	Specifies the name of the AS.
rir (optional)	string	Specifies the name of the Regional Internet Registry (RIR) that assigned the number to the AS.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.III. DIRECTORY

"The Directory Object represents the properties common to a file system directory."

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- modified some xsd:dateTimeStamp
- path only xsd:string
- pathEnc some xsd:string
- type some {"directory"^^xsd:string}

where:

Table 22 Properties of Directory cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be directory.
path (required)	string	Specifies the path, as originally observed, to the directory on the file system.
path_enc (optional)	string	Specifies the observed encoding for the path. The value MUST be specified if the path is stored in a non-Unicode encoding. This value MUST be specified using the corresponding name from the 2013-12-20 revision of the IANA character set registry. If the preferred MIME name for a character set is defined, this value MUST be used; if it is not defined, then the Name value from the registry MUST be used instead.
created (optional)	timestamp	Specifies the date/time the directory was created.
modified (optional)	timestamp	Specifies the date/time the directory was last written to/modified.
accessed (optional)	timestamp	Specifies the date/time the directory was last accessed.
contains_refs (optional)	list of type object-ref	Specifies a list of references to other File and/or Directory Objects contained within the directory. The objects referenced in this list MUST be of type file or directory.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.IV. DOMAIN NAME

"The Domain Name represents the properties of a network domain name."

Every instance of the class is characterized by the following properties:

- resolvesToRefs some DomainName
- resolvesToRefs some IPv4Addr
- resolvesToRefs some IPv6Addr
- type some {"domain-name"^^xsd:string}
- value only xsd:string

where:

Table 23 Properties of Domain name cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be domain-name.
value (required)	string	Specifies the value of the domain name. The value of this property MUST conform to RFC1034, and each domain and sub-domain contained within the domain name MUST conform to RFC5890.
resolves_to_refs (optional)	list of type object-ref	Specifies a list of references to one or more IP addresses or domain names that the domain name resolves to. The objects referenced in this list MUST be of type ipv4-addr or ipv6-addr or domain-name (for cases such as CNAME records).

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has a subclass: Subdomain.

B.V.IV.1. SUBDOMAIN

Every instance of the class is characterized by the following properties:

- resolvesToRefs some DomainName
- resolvesToRefs some IPv4Addr
- resolvesToRefs some IPv6Addr
- subdomain min 1 DomainName

- type some {"domain-name"^^xsd:string}
- value only xsd:string

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of DomainName
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.V. EMAIL ADDRESS

"The Email Address Object represents a single email address."

Every instance of the class is characterized by the following properties:

- belongToRefs some UserAccount
- displayName some xsd:string
- type some {"email-addr"^^xsd:string}
- value only xsd:string

where:

Table 24 Properties of Email address cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be email-addr.
value (required)	string	Specifies a single email address. This MUST NOT include the display name. This property corresponds to the addr-spec construction in section 3.4 of RFC5322 , for example, jane.smith@example.com.
display_name (optional)	string	Specifies a single email display name, i.e., the name that is displayed to the human user of a mail application. This property corresponds to the display-name construction in section 3.4 of RFC5322, for example, Jane Smith.
belongs_to_ref (optional)	object-ref	Specifies the user account that the email address belongs to, as a reference to a User Account Object. The object referenced in this property MUST be of type user-account.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of DomainName
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.VI. EMAIL MESSAGE

"The Email Message Object represents an instance of an email message."

Every instance of the class is characterized by the following properties:

- additionalHeaderFields some stix:dictionary
- body some xsd:string
- bodyMultipart some stix:list
- bodyMultipart some stix:mimeType
- contentType some xsd:string
- isMultipart only xsd:boolean
- receivedLines some stix:list
- receivedLines some xsd:string
- subject some xsd:string
- type some {"email-message"^^xsd:string}

where:

Table 25 Properties of Email message cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be email-message.
is_multipart (required)	boolean	Indicates whether the email body contains multiple MIME parts.
date (optional)	timestamp	Specifies the date/time that the email message was sent.
content_type (optional)	string	Specifies the value of the "Content-Type" header of the email message.
from_ref (optional)	object-ref	Specifies the value of the "From:" header of the email message. The "From:" field specifies the author(s) of the message, that is, the mailbox(es) of the person(s) or system(s) responsible for the writing of the message. The object referenced in this property MUST be of type email-address.
sender_ref (optional)	object-ref	Specifies the value of the "Sender" field of the email message. The "Sender:" field specifies the mailbox of the agent responsible for the actual transmission of the message. The object referenced in this property MUST be of type email-address.
to_refs (optional)	list of type object-ref	Specifies the mailboxes that are "To:" recipients of the email message.

		The objects referenced in this list MUST be of type email-address.
cc_refs (optional)	list of type object-ref	Specifies the mailboxes that are "CC:" recipients of the email message. The objects referenced in this list MUST be of type email-address.
bcc_refs (optional)	list of type object-ref	Specifies the mailboxes that are "BCC:" recipients of the email message. As per RFC5322, this list may be empty, which should not be treated the same as the key being absent. The objects referenced in this list MUST be of type email-address.
subject (optional)	string	Specifies the subject of the email message.
received_lines (optional)	list of type string	Specifies one or more "Received" header fields that may be included in the email headers. List values MUST appear in the same order as present in the email message.
additional_header_fields (optional)	dictionary	Specifies any other header fields (except for date, received_lines, content_type, from_ref, sender_ref, to_refs, cc_refs, bcc_refs, and subject) found in the email message, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single header field or names/values of a header field that occurs more than once. Each dictionary key SHOULD be a case-preserved version of the header field name. For cases where a header field occurs exactly once, the corresponding value for the dictionary key MUST be a string. For cases where a header field occurs more than once, the corresponding value for the dictionary key MUST be a list of type string, where each string in the list represents a single value of the header field.
body (optional)	string	Specifies a string containing the email body. This property MUST NOT be used if is_multipart is true.
body_multipart (optional)	list of type mime-part-type	Specifies a list of the MIME parts that make up the email body. This property MUST NOT be used if is_multipart is false.
raw_email_ref (optional)	object-ref	Specifies the raw binary contents of the email message, including both the headers and body, as a reference to an Artifact Object. The object referenced in this property MUST be of type artifact.

This class has the following relationships with other classes:

- SubClass of CyberObservable

- owl:Thing or (objectRefs some MarkingDefinition)
- bccRefs some EmailAddress
- ccRefs some EmailAddress
- fromRef some EmailAddress
- rawEmailRef some Artifact
- senderRef some EmailAddress
- toRefs some EmailAddress.

This class has a subclass: MimePartType.

B.V.VI.1. MIMEPARTTYPE

Every instance of the class is characterized by the following properties:

- additionalHeaderFields some stix:dictionary
- body some xsd:string
- bodyMultipart some stix:list
- bodyMultipart some stix:mimePartType
- contentDisposition some xsd:string
- contentType some xsd:string
- isMultipart only xsd:boolean
- receivedLines some stix:list
- receivedLines some xsd:string
- subject some xsd:string
- type some {"email-message"^^xsd:string}
- type some {"mime-part-type"^^xsd:string}

where:

Table 26 Properties of Mime Part Type class (STIX2.owl)

Property Name	Type	Description
body (optional)	string	Specifies the contents of the MIME part if the content_type is not provided or starts with text/ (e.g., in the case of plain text or HTML email). For inclusion in this property, the contents MUST be decoded to Unicode. Note that the charset provided in content_type is for informational usage and not for decoding of this property.
body_raw_ref (optional)	object-ref	Specifies the contents of non-textual MIME parts, that is those whose content_type does not start with text/, as a reference to an Artifact Object or File Object. The object referenced in this property MUST be of type artifact or file. For use cases where

		conveying the actual data contained in the MIME part is of primary importance, artifact SHOULD be used. Otherwise, for use cases where conveying metadata about the file-like properties of the MIME part is of primary importance, file SHOULD be used.
content_type (optional)	string	Specifies the value of the “Content-Type” header field of the MIME part. Any additional “Content-Type” header field parameters such as charset SHOULD be included in this property. Example: text/html; charset=UTF-8
content_disposition (optional)	string	Specifies the value of the “Content-Disposition” header field of the MIME part.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of EmailMessage
- owl:Thing or (objectRefs some MarkingDefinition)
- bodyRawRef some Artifact
- bodyRawRef some File
- ccRefs some EmailAddress
- fromRef some EmailAddress
- rawEmailRef some Artifact
- senderRef some EmailAddress
- toRefs some EmailAddress

This class has no subclasses.

B.V.VII. FILE

"The File Object represents the properties of a file."

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string

- modified some xsd:dateTimeStamp
 - name some xsd:string
 - name_enc some xsd:string
 - size some xsd:integer
 - type some {"file"^^xsd:string}
- where:

Table 27 Properties of File cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be file.
extensions (optional)	dictionary	<p>The File Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p>ntfs-ext, raster-image-ext, pdf-ext, archive-ext, windows-pebinary-ext</p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
hashes (optional)	hashes	Specifies a dictionary of hashes for the file.
size (optional)	integer	Specifies the size of the file, in bytes. The value of this property MUST NOT be negative.
name (optional)	string	Specifies the name of the file.
name_enc (optional)	string	<p>Specifies the observed encoding for the name of the file. This value MUST be specified using the corresponding name from the 2013-12-20 revision of the IANA character set registry. If the value from the Preferred MIME Name column for a character set is defined, this value MUST be used; if it is not defined, then the value from the Name column in the registry MUST be used instead.</p> <p>This property allows for the capture of the original text encoding for the file name, which may be forensically relevant; for example, a file on an NTFS volume whose name was created using the windows-1251 encoding, commonly used for languages based on Cyrillic script.</p>
magic_number_hex (optional)	hex	Specifies the hexadecimal constant ("magic number") associated with a specific file format that corresponds to the file, if applicable.

mime_type (optional)	string	Specifies the MIME type name specified for the file, e.g., application/msword. Whenever feasible, this value SHOULD be one of the values defined in the Template column in the IANA media type registry. Maintaining a comprehensive universal catalog of all extant file types is obviously not possible. When specifying a MIME Type not included in the IANA registry, implementers should use their best judgement so as to facilitate interoperability.
created (optional)	timestamp	Specifies the date/time the file was created.
modified (optional)	timestamp	Specifies the date/time the file was last written to/modified.
accessed (optional)	timestamp	Specifies the date/time the file was last accessed.
parent_directory_ref (optional)	object-ref	Specifies the parent directory of the file, as a reference to a Directory Object. The object referenced in this property MUST be of type directory.
is_encrypted (optional)	boolean	Specifies whether the file is encrypted.
encryption_algorithm (optional)	open-vocab	Specifies the name of the encryption algorithm used to encrypt the file. This is an open vocabulary and values SHOULD come from the encryption-algo-ov vocabulary. This property MUST NOT be used if is_encrypted is false or not included.
decryption_key (optional)	string	Specifies the decryption key used to decrypt the file. This property MUST NOT be used if is_encrypted is false or not included.
contains_refs (optional)	list of type object-ref	Specifies a list of references to other Observable Objects contained within the file, such as another file that is appended to the end of the file, or an IP address that is contained somewhere in the the file. This is intended for use cases other than those targeted by the Archive extension.
content_ref (optional)	object-ref	Specifies the content of the file, represented as an Artifact Object. The object referenced in this property MUST be of type artifact.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory

This class has several subclasses representing File Extensions:

- Archive File Extension
- NTFS File Extension.
- PDF File Extension.
- Raster Image File Extension.
- Windows™ PE Binary File Extension.

B.V.VII.1. ARCHIVE FILE EXTENSION

"The Archive File extension specifies a default extension for capturing properties specific to archive files."

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name some xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"archive-ext"^^xsd:string}
- version some xsd:string
- comment some xsd:string

where:

Table 28 Properties of Archive File extension cyber observable class (STIX2.owl)

Property Name	Type	Description
contains_refs (required)	list of type object-ref	Specifies the files contained in the archive, as a reference to one or more other File Objects.

		The objects referenced in this list MUST be of type file.
version (optional)	string	Specifies the version of the archive type used in the archive file.
comment (optional)	string	Specifies a comment included as part of the archive file.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of File
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory
- contains_refs some File

This class has no subclasses.

B.V.VII.2. NTFS FILE EXTENSION

"The NTFS file extension specifies a default extension for capturing properties specific to the storage of the file on the NTFS file system."

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name some xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"ntfs-ext"^^xsd:string}
- sid some xsd:string

where:

Table 29 Properties of NTFS File extension cyber observable class (STIX2.owl)

Property Name	Type	Description
sid (optional)	string	Specifies the security ID (SID) value assigned to the file.
alternate_data_streams (optional)	list of type alternate-data-stream-type	Specifies a list of NTFS alternate data streams that exist for the file.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of File
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory
- alternate_data_streams some alternate-data-stream-type

This class has a subclass:

- alternate-data-stream-type

ALTERNATE-DATA-STREAM-TYPE

“The Alternate Data Stream type represents an NTFS alternate data stream”

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name only xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"ntfs-ext"^^xsd:string}
- sid some xsd:string

where:

Table 30 Properties of Alternate Data Stream type class (STIX2.owl)

Property Name	Type	Description
name (required)	string	Specifies the name of the alternate data stream.
hashes (optional)	hashes	Specifies a dictionary of hashes for the data contained in the alternate data stream.
size (optional)	integer	Specifies the size of the alternate data stream, in bytes. The value of this property MUST NOT be negative.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of File
- SubClass of NTFSFileExtension
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory
- alternate_data_streams some alternate-data-stream-type

This class has no subclasses.

B.V.VII.3. PDF FILE EXTENSION

“The PDF file extension specifies a default extension for capturing properties specific to PDF files”

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name only xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"pdf-ext"^^xsd:string}
- version some xsd:string

- `is_optimized` some `xsd:Boolean`
- `document_info_dict` some `xsd:string`
- `pdfid0` some `xsd:string`
- `pdfid1` some `xsd:string`

where:

Table 31 Properties of PDF File extension cyber observable class (STIX2.owl)

Property Name	Type	Description
version (optional)	string	Specifies the decimal version number of the string from the PDF header that specifies the version of the PDF specification to which the PDF file conforms. E.g., 1.4.
is_optimized (optional)	boolean	Specifies whether the PDF file has been optimized.
document_info_dict (optional)	dictionary	Specifies details of the PDF document information dictionary (DID), which includes properties like the document creation data and producer, as a dictionary. Each key in the dictionary SHOULD be a case-preserved version of the corresponding entry in the document information dictionary without the prepended forward slash, e.g., Title. The corresponding value for the key MUST be the value specified for the document information dictionary entry, as a string.
pdfid0 (optional)	string	Specifies the first file identifier found for the PDF file.
pdfid1 (optional)	string	Specifies the second file identifier found for the PDF file.

This class has the following relationships with other classes:

- SubClass of `CyberObservable`
- SubClass of `File`
- `owl:Thing` or (`objectRefs` some `MarkingDefinition`)
- `containsRefs` some `CyberObservable`
- `contentRef` some `Artifact`
- `parentDirectoryRef` some `Directory`

This class has no subclasses.

B.V.VII.4. RASTER IMAGE FILE EXTENSION

“The Raster Image file extension specifies a default extension for capturing properties specific to raster image files”.

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name only xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"raster-image-ext"^^xsd:string}
- image_height some xsd:integer
- image_width some xsd:integer
- bits_per_pixel some xsd:integer
- image_compression_algorithm some xsd:integer
- exif_tags some xsd:string

where:

Table 32 Properties of Raster Image File extension cyber observable class (STIX2.owl)

Property Name	Type	Description
image_height (optional)	integer	Specifies the height of the image in the image file, in pixels.
image_width (optional)	integer	Specifies the width of the image in the image file, in pixels.
bits_per_pixel (optional)	integer	Specifies the sum of bits used for each color channel in the image file, and thus the total number of pixels used for expressing the color depth of the image.
image_compression_algorithm (optional)	string	Specifies the name of the compression algorithm used to compress the image in the image file, if applicable.
exif_tags (optional)	dictionary	Specifies the set of EXIF tags found in the image file, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single EXIF tag. Accordingly, each dictionary

		key MUST be a case-preserved version of the EXIF tag name, e.g., XResolution. Each dictionary value MUST be either an integer (for int* EXIF datatypes) or a string (for all other EXIF datatypes).
--	--	---

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of File
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory

This class has no subclasses.

B.V.VII.5. WINDOWS™ PE BINARY FILE EXTENSION

“The Windows™ PE Binary File extension specifies a default extension for capturing properties specific to Windows portable executable (PE) files”.

Every instance of the class is characterized by the following properties:

- accessed some xsd:dateTimeStamp
- created some xsd:dateTimeStamp
- decryptionKey some xsd:string
- encryptionAlgorithm some stix:openVocab
- extensions some stix:dictionary
- hashes some stix:hashes
- isEncrypted some xsd:boolean
- magicNumberHex some xsd:hexBinary
- mimeType some xsd:string
- modified some xsd:dateTimeStamp
- name only xsd:string
- name_enc some xsd:string
- size some xsd:integer
- type some {"windows-pebinary-ext"^^xsd:string}
- pe_type only stix: windows-pebinary-type-ov
- imphash only stix:OpenVocab
- machine_hex some xsd:hex
- number_of_sections some xsd:integer
- time_date_stamp some xsd:dateTimeStamp
- pointer_to_symbol_table_hex some xsd:hex
- number_of_symbols some xsd:integer

- size_of_optional_header some xsd:integer
- characteristics_hex some xsd:hex
- file_header_hashes some xsd:hex
- optional_header some stix: windows-pe-optional-header-type
- sections some stix: windows-pe-section-type

where:

Table 33 Properties of Windows PE Binary File extension cyber observable class (STIX2.owl)

Property Name	Type	Description
pe_type (required)	open-vocab	Specifies the type of the PE binary. This is an open vocabulary and values SHOULD come from the windows-pebinary-type-ov vocabulary.
imphash (optional)	string	Specifies the special import hash, or 'imphash', calculated for the PE Binary based on its imported libraries and functions. For more information on the imphash algorithm, see the original article by Mandiant/FireEye ³³ .
machine_hex (optional)	hex	Specifies the type of target machine.
number_of_sections (optional)	integer	Specifies the number of sections in the PE binary, as a non-negative integer.
time_date_stamp (optional)	timestamp	Specifies the time when the PE binary was created. The timestamp value MUST be precise to the second.
pointer_to_symbol_table_hex (optional)	hex	Specifies the file offset of the COFF symbol table.
number_of_symbols (optional)	integer	Specifies the number of entries in the symbol table of the PE binary, as a non-negative integer.
size_of_optional_header (optional)	integer	Specifies the size of the optional header of the PE binary. The value of this property MUST NOT be negative.
characteristics_hex (optional)	hex	Specifies the flags that indicate the file's characteristics.
file_header_hashes (optional)	hashes	Specifies any hashes that were computed for the file header.
optional_header (optional)	windows-pe-	Specifies the PE optional header of the PE binary.

³³<https://www.fireeye.com/blog/threat-research/2014/01/tracking-malware-import-hashing.html>

	optional- header- type	
sections (optional)	list of type windows- pe- section- type	Specifies metadata about the sections in the PE file.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of File
- owl:Thing or (objectRefs some MarkingDefinition)
- containsRefs some CyberObservable
- contentRef some Artifact
- parentDirectoryRef some Directory

This class has the following subclasses:

WINDOWS-PEBINARY-TYPE-OV

An open vocabulary of Windows PE binary types:

Table 34 Windows PE Binary types class

Value	Description
exe	Specifies that the PE binary is an executable image (i.e., not an OBJ or DLL).
dll	Specifies that the PE binary is a dynamically linked library (DLL).
sys	Specifies that the PE binary is a device driver (SYS).

WINDOWS-PE-OPTIONAL-HEADER-TYPE

The Windows PE Optional Header type represents the properties of the PE optional header.

Table 35 Windows PE Optional header type class

Property Name	Type	Description
magic_hex (optional)	hex	Specifies the hex value that indicates the type of the PE binary.
major_linker_version (optional)	integer	Specifies the linker major version number.

minor_linker_version (optional)	integer	Specifies the linker minor version number.
size_of_code (optional)	integer	Specifies the size of the code (text) section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative.
size_of_initialized_data (optional)	integer	Specifies the size of the initialized data section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative.
size_of_uninitialized_data (optional)	integer	Specifies the size of the uninitialized data section. If there are multiple such sections, this refers to the sum of the sizes of each section. The value of this property MUST NOT be negative.
address_of_entry_point (optional)	integer	Specifies the address of the entry point relative to the image base when the executable is loaded into memory.
base_of_code (optional)	integer	Specifies the address that is relative to the image base of the beginning-of-code section when it is loaded into memory.
base_of_data (optional)	integer	Specifies the address that is relative to the image base of the beginning-of-data section when it is loaded into memory.
image_base (optional)	integer	Specifies the preferred address of the first byte of the image when loaded into memory.
section_alignment (optional)	integer	Specifies the alignment (in bytes) of PE sections when they are loaded into memory.
file_alignment (optional)	integer	Specifies the factor (in bytes) that is used to align the raw data of sections in the image file.
major_os_version (optional)	integer	Specifies the major version number of the required operating system.
minor_os_version (optional)	integer	Specifies the minor version number of the required operating system.
major_image_version (optional)	integer	Specifies the major version number of the image.
minor_image_version (optional)	integer	Specifies the minor version number of the image.

major_subsystem_version (optional)	integer	Specifies the major version number of the subsystem.
minor_subsystem_version (optional)	integer	Specifies the minor version number of the subsystem.
win32_version_value_hex (optional)	hex	Specifies the reserved win32 version value.
size_of_image (optional)	integer	Specifies the size of the image in bytes, including all headers, as the image is loaded in memory. The value of this property MUST NOT be negative.
size_of_headers (optional)	integer	Specifies the combined size of the MS-DOS, PE header, and section headers, rounded up to a multiple of the value specified in the file_alignment header. The value of this property MUST NOT be negative.
checksum_hex (optional)	hex	Specifies the checksum of the PE binary.
subsystem_hex (optional)	hex	Specifies the subsystem (e.g., GUI, device driver, etc.) that is required to run this image.
dll_characteristics_hex (optional)	hex	Specifies the flags that characterize the PE binary.
size_of_stack_reserve (optional)	integer	Specifies the size of the stack to reserve, in bytes. The value of this property MUST NOT be negative.
size_of_stack_commit (optional)	integer	Specifies the size of the stack to commit, in bytes. The value of this property MUST NOT be negative.
size_of_heap_reserve (optional)	integer	Specifies the size of the local heap space to reserve, in bytes.. The value of this property MUST NOT be negative.
size_of_heap_commit (optional)	integer	Specifies the size of the local heap space to commit, in bytes. The value of this property MUST NOT be negative.
loader_flags_hex (optional)	hex	Specifies the reserved loader flags.
number_of_rva_and_sizes (optional)	integer	Specifies the number of data-directory entries in the remainder of the optional header.
hashes (optional)	hashes	Specifies any hashes that were computed for the optional header.

WINDOWS-PE-SECTION-TYPE

The Windows PE Section type specifies metadata about a PE file section.

Table 36 Windows PE Section type class

Property Name	Type	Description
name (required)	string	Specifies the name of the section.
size (optional)	integer	Specifies the size of the section, in bytes. The value of this property MUST NOT be negative.
entropy (optional)	float	Specifies the calculated entropy for the section, as calculated using the Shannon algorithm ³⁴ . The size of each input character is defined as a byte, resulting in a possible range of 0 through 8.
hashes (optional)	hashes	Specifies any hashes computed over the section.

B.V.VIII. IPV4ADDR

"The IPv4 Address Object represents one or more IPv4 addresses expressed using CIDR notation."

Every instance of the class is characterized by the following properties:

- type some {"ipv4-addr"^^xsd:string}
- value only xsd:string

where:

Table 37 Properties of IPv4Addr cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be ipv4-addr.
value (required)	string	Specifies one or more IPv4 addresses expressed using CIDR notation. If a given IPv4 Address Object represents a single IPv4 address, the CIDR /32 suffix MAY be omitted. Example: 10.2.4.5/24
resolves_to_refs (optional)	list of type object-ref	Specifies a list of references to one or more Layer 2 Media Access Control (MAC) addresses that the IPv4 address resolves to. The objects referenced in this list MUST be of type mac-addr.
belongs_to_refs (optional)	list of type object-ref	Specifies a list of reference to one or more autonomous systems (AS) that the IPv4 address belongs to.

³⁴ https://en.wiktionary.org/wiki/Shannon_entropy

		The objects referenced in this list MUST be of type autonomous-system.
--	--	--

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)
- resolvesToRefs some MACAddr
- belongToRefs some AutonomousSystem

This class has no subclasses

B.V.IX. IPV6ADDR

"The IPv6 Address Object represents one or more IPv6 addresses expressed using CIDR notation."

Every instance of the class is characterized by the following properties:

- type some {"ipv6-addr"^^xsd:string}
- value only xsd:string

where:

Table 38 Properties of IPv6Addr cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be ipv6-addr.
value (required)	string	Specifies one or more IPv6 addresses expressed using CIDR notation. If a given IPv6 Address Object represents a single IPv6 address, the CIDR /128 suffix MAY be omitted.
resolves_to_refs (optional)	list of type object-ref	Specifies a list of references to one or more Layer 2 Media Access Control (MAC) addresses that the IPv6 address resolves to. The objects referenced in this list MUST be of type mac-addr.
belongs_to_refs (optional)	list of type object-ref	Specifies a list of reference to one or more autonomous systems (AS) that the IPv6 address belongs to. The objects referenced in this list MUST be of type autonomous-system.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)
- resolvesToRefs some MACAddr
- belongToRefs some AutonomousSystem

This class has no subclasses

B.V.X. MAC-ADDR

"The MAC Address Object represents a single Media Access Control (MAC) address."

Every instance of the class is characterized by the following properties:

- type some {"mac-addr"^^xsd:string}
- value only xsd:string

where:

Table 39 Properties of Mac-addr cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be mac-addr.
value (required)	string	Specifies a single MAC address. The MAC address value MUST be represented as a single colon-delimited, lowercase MAC-48 address, which MUST include leading zeros for each octet. Example: 00:00:ab:cd:ef:01

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XI. MUTEX

"The Mutex Object represents the properties of a mutual exclusion (mutex) object".

Every instance of the class is characterized by the following properties:

- name only xsd:string
- type some {"mutex"^^xsd:string}

where:

Table 40 Properties of MUTEX cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be mutex.
name (required)	string	Specifies the name of the mutex object.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XII. NETWORK TRAFFIC

"The Network Traffic Object represents arbitrary network traffic that originates from a source and is addressed to a destination."

Every instance of the class is characterized by the following properties:

- dstByteCount some xsd:integer
- dstPackets some xsd:integer
- dstPort some xsd:integer
- end some xsd:dateTimeStamp
- extensions some stix:dictionary
- ipfix some stix:dictionary
- isActive some xsd:boolean
- protocols some stix:list
- protocols some xsd:string
- srcByteCount some xsd:integer
- srcPackets some xsd:integer
- srcPort some xsd:integer
- start some xsd:dateTimeStamp
- type some {"network-traffic"^^xsd:string}

where:

Table 41 Properties of Network Traffic cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be network-traffic.
extensions (optional)	dictionary	The Network Traffic Object defines the following extensions. In addition to these, producers MAY create their own. http-request-ext, tcp-ext, icmp-ext, socket-ext Dictionary keys MUST identify the extension type by name. The corresponding dictionary values MUST contain the contents of the extension instance.
start (optional)	timestamp	Specifies the date/time the network traffic was initiated, if known.
end (optional)	timestamp	Specifies the date/time the network traffic ended, if known. If the is_active property is true, then the end property MUST NOT be included.
is_active (optional)	boolean	Indicates whether the network traffic is still ongoing.

src_ref (optional)	object-ref	<p>Specifies the source of the network traffic, as a reference to one or more Observable Objects.</p> <p>The objects referenced in this list MUST be of type ipv4-addr or ipv6-addr or mac-addr or domain-name (for cases where the IP address for a domain name is unknown).</p>
dst_ref (optional)	object-ref	<p>Specifies the destination of the network traffic, as a reference to one or more Observable Objects.</p> <p>The objects referenced in this list MUST be of type ipv4-addr or ipv6-addr or mac-addr or domain-name (for cases where the IP address for a domain name is unknown).</p>
src_port (optional)	integer	Specifies the source port used in the network traffic, as an integer. The port value MUST be in the range of 0 - 65535.
dst_port (optional)	integer	Specifies the destination port used in the network traffic, as an integer. The port value MUST be in the range of 0 - 65535.
protocols (required)	list of type string	<p>Specifies the protocols observed in the network traffic, along with their corresponding state.</p> <p>Protocols MUST be listed in low to high order, from outer to inner in terms of packet encapsulation. That is, the protocols in the outer level of the packet, such as IP, MUST be listed first.</p> <p>The protocol names SHOULD come from the service names defined in the Service Name column of the IANA Service Name and Port Number Registry [Port Numbers]. In cases where there is variance in the name of a network protocol not included in the IANA Registry, content producers should exercise their best judgement, and it is recommended that lowercase names be used for consistency with the IANA registry.</p> <p>Examples:</p> <p>ipv4, tcp, http</p> <p>ipv4, udp</p> <p>ipv6, tcp, http</p> <p>ipv6, tcp, ssl, https</p>
src_byte_count (optional)	integer	Specifies the number of bytes sent from the source to the destination.
dst_byte_count (optional)	integer	Specifies the number of bytes sent from the destination to the source.
src_packets (optional)	integer	Specifies the number of packets sent from the source to the destination.

dst_packets (optional)	integer	Specifies the number of packets sent destination to the source.
ipfix (optional)	dictionary	Specifies any IP Flow Information Export [IPFIX] data for the traffic, as a dictionary. Each key/value pair in the dictionary represents the name/value of a single IPFIX element. Accordingly, each dictionary key SHOULD be a case-preserved version of the IPFIX element name, e.g., octetDeltaCount. Each dictionary value MUST be either an integer or a string, as well as a valid IPFIX property.
src_payload_ref (optional)	object-ref	Specifies the bytes sent from the source to the destination. The object referenced in this property MUST be of type artifact.
dst_payload_ref (optional)	object-ref	Specifies the bytes sent from the destination to the source. The object referenced in this property MUST be of type artifact.
encapsulates_refs (optional)	list of type object-ref	Links to other network-traffic objects encapsulated by this network-traffic object. The objects referenced in this property MUST be of type network-traffic.
encapsulated_by_ref (optional)	object-ref	Links to another network-traffic object which encapsulates this object. The object referenced in this property MUST be of type network-traffic.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)
- dstPayloadRef some Artifact
- dstRef some DomainName
- dstRef some IPv4Addr
- dstRef some IPv6Addr
- dstRef some MACAddr
- encapsulatedByRef some NetworkTraffic
- encapsulatesRefs some NetworkTraffic
- srcPayloadRef some Artifact
- srcRef some DomainName
- srcRef some IPv4Addr
- srcRef some IPv6Addr
- srcRef some MACAddr

This class has the following subclasses:

- HTTP Request Extension.

- ICMP Extension.
- Network Socket Extension.
- TCP Extension.

B.V.XII.1. HTTP REQUEST EXTENSION

"The HTTP request extension specifies a default extension for capturing network traffic properties specific to HTTP requests."

Every instance of the class is characterized by the following properties:

- dstByteCount some xsd:integer
- dstPackets some xsd:integer
- dstPort some xsd:integer
- end some xsd:dateTimeStamp
- extensions some stix:dictionary
- ipfix some stix:dictionary
- isActive some xsd:boolean
- protocols some stix:list
- protocols some xsd:string
- srcByteCount some xsd:integer
- srcPackets some xsd:integer
- srcPort some xsd:integer
- start some xsd:dateTimeStamp
- type some {"network-traffic"^^xsd:string}
- request_method only xsd:string
- request_value only xsd:string
- request_version some xsd:string
- request_header some stix:dictionary
- message_body_length some xsd:integer
- message_body_data_ref some xsd:string
- extensions some {"http-request-ext"^^xsd:string}

Table 42 Properties of HTTP Header extension class

Property Name	Type	Description
request_method (required)	string	Specifies the HTTP method portion of the HTTP request line, as a lowercase string.
request_value (required)	string	Specifies the value (typically a resource path) portion of the HTTP request line.
request_version (optional)	string	Specifies the HTTP version portion of the HTTP request line, as a lowercase string.

request_header (optional)	dictionary	Specifies all of the HTTP header fields that may be found in the HTTP client request, as a dictionary. Each key in the dictionary MUST be the name of the header field and SHOULD preserve case, e.g., User-Agent. The corresponding value for each dictionary key MUST be a string.
message_body_length (optional)	integer	Specifies the length of the HTTP message body, if included, in bytes.
message_body_data_ref (optional)	object-ref	Specifies the data contained in the HTTP message body, if included. The object referenced in this property MUST be of type artifact.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of NetworkTraffic
- owl:Thing or (objectRefs some MarkingDefinition)
- dstPayloadRef some Artifact
- dstRef some DomainName
- dstRef some IPv4Addr
- dstRef some IPv6Addr
- dstRef some MACAddr
- encapsulatedByRef some NetworkTraffic
- encapsulatesRefs some NetworkTraffic
- srcPayloadRef some Artifact
- srcRef some DomainName
- srcRef some IPv4Addr
- srcRef some IPv6Addr
- srcRef some MACAddr

This class has no subclasses.

B.V.XII.2. ICMP EXTENSION

“The ICMP extension specifies a default extension for capturing network traffic properties specific to ICMP”

Every instance of the class is characterized by the following properties:

- dstByteCount some xsd:integer
- dstPackets some xsd:integer
- dstPort some xsd:integer
- end some xsd:dateTimeStamp
- extensions some stix:dictionary
- ipfix some stix:dictionary

- isActive some xsd:boolean
- protocols some stix:list
- protocols some xsd:string
- srcByteCount some xsd:integer
- srcPackets some xsd:integer
- srcPort some xsd:integer
- start some xsd:dateTimeStamp
- type some {"network-traffic"^^xsd:string}
- icmp_type_hex some xsd:hex
- icmp_code_hex some xsd:hex
- extensions some {"icmp-ext"^^xsd:string}

Table 43 Properties of ICMP extensión class

Property Name	Type	Description
icmp_type_hex (required)	hex	Specifies the ICMP type byte.
icmp_code_hex (required)	hex	Specifies the ICMP code byte.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of NetworkTraffic
- owl:Thing or (objectRefs some MarkingDefinition)
- dstPayloadRef some Artifact
- dstRef some DomainName
- dstRef some IPv4Addr
- dstRef some IPv6Addr
- dstRef some MACAddr
- encapsulatedByRef some NetworkTraffic
- encapsulatesRefs some NetworkTraffic
- srcPayloadRef some Artifact
- srcRef some DomainName
- srcRef some IPv4Addr
- srcRef some IPv6Addr
- srcRef some MACAddr

This class has no subclasses.

B.V.XII.3. NETWORK SOCKET EXTENSION

“The Network Socket extension specifies a default extension for capturing network traffic properties associated with network sockets”.

Every instance of the class is characterized by the following properties:

- dstByteCount some xsd:integer
- dstPackets some xsd:integer
- dstPort some xsd:integer
- end some xsd:dateTimeStamp
- extensions some stix:dictionary
- ipfix some stix:dictionary
- isActive some xsd:boolean
- protocols some stix:list
- protocols some xsd:string
- srcByteCount some xsd:integer
- srcPackets some xsd:integer
- srcPort some xsd:integer
- start some xsd:dateTimeStamp
- type some {"network-traffic"^^xsd:string}
- address_family some xsd:string
- is_blocking some xsd:boolean
- is_listening some xsd:Boolean
- protocol_family some stix:network-socket-protocol-family-enum
- options some stix:dictionary
- socket_type some stix: network-socket-type-enum
- socket_descriptor some xsd:integer
- socket_handle some xsd:integer
- extensions some {"socket-ext"^^xsd:string}

Table 44 Properties of Network Socket extension class

Property Name	Type	Description
address_family (required)	network-socket-address-family-enum	Specifies the address family (AF_*) that the socket is configured for.
is_blocking (optional)	boolean	Specifies whether the socket is in blocking mode.
is_listening (optional)	boolean	Specifies whether the socket is in listening mode.
protocol_family (optional)	network-socket-protocol-family-enum	Specifies the protocol family (PF_*) that the socket is configured for.

options (optional)	dictionary	Specifies any options (SO_*) that may be used by the socket, as a dictionary. Each key in the dictionary SHOULD be a case-preserved version of the option name, e.g., SO_ACCEPTCONN. Each key value in the dictionary MUST be the value for the corresponding options key.
socket_type (optional)	network-socket-type-enum	Specifies the type of the socket.
socket_descriptor (optional)	integer	Specifies the socket file descriptor value associated with the socket, as a non-negative integer.
socket_handle (optional)	integer	Specifies the handle or inode value associated with the socket.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of NetworkTraffic
- owl:Thing or (objectRefs some MarkingDefinition)
- dstPayloadRef some Artifact
- dstRef some DomainName
- dstRef some IPv4Addr
- dstRef some IPv6Addr
- dstRef some MACAddr
- encapsulatedByRef some NetworkTraffic
- encapsulatesRefs some NetworkTraffic
- srcPayloadRef some Artifact
- srcRef some DomainName
- srcRef some IPv4Addr
- srcRef some IPv6Addr
- srcRef some MACAddr

This class has no subclasses.

B.V.XII.4. TCP EXTENSION

“The TCP extension specifies a default extension for capturing network traffic properties specific to TCP”.

Every instance of the class is characterized by the following properties:

- dstByteCount some xsd:integer
- dstPackets some xsd:integer
- dstPort some xsd:integer
- end some xsd:dateTimeStamp

- extensions some stix:dictionary
- ipfix some stix:dictionary
- isActive some xsd:boolean
- protocols some stix:list
- protocols some xsd:string
- srcByteCount some xsd:integer
- srcPackets some xsd:integer
- srcPort some xsd:integer
- start some xsd:dateTimeStamp
- type some {"network-traffic"^^xsd:string}
- src_flags_hex some xsd:hex
- fst_flags_hex some xsd:hex
- extensions some {"tcp-ext"^^xsd:string}

Table 45 Properties of TCP extensión class

Property Name	Type	Description
src_flags_hex (optional)	hex	Specifies the source TCP flags, as the union of all TCP flags observed between the start of the traffic (as defined by the start property) and the end of the traffic (as defined by the end property). If the start and end times of the traffic are not specified, this property SHOULD be interpreted as the union of all TCP flags observed over the entirety of the network traffic being reported upon.
dst_flags_hex (optional)	hex	Specifies the destination TCP flags, as the union of all TCP flags observed between the start of the traffic (as defined by the start property) and the end of the traffic (as defined by the end property). If the start and end times of the traffic are not specified, this property SHOULD be interpreted as the union of all TCP flags observed over the entirety of the network traffic being reported upon.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of NetworkTraffic
- owl:Thing or (objectRefs some MarkingDefinition)
- dstPayloadRef some Artifact
- dstRef some DomainName
- dstRef some IPv4Addr
- dstRef some IPv6Addr
- dstRef some MACAddr
- encapsulatedByRef some NetworkTraffic
- encapsulatesRefs some NetworkTraffic
- srcPayloadRef some Artifact

- srcRef some DomainName
- srcRef some IPv4Addr
- srcRef some IPv6Addr
- srcRef some MACAddr

This class has no subclasses.

B.V.XIII. PROCESS

“The Process Object represents common properties of an instance of a computer program as executed on an operating system”.

Every instance of the class is characterized by the following properties:

- arguments some stix:list
- arguments some xsd:string
- commandLine some xsd:string
- created some xsd:dateTimeStamp
- cwd some xsd:string
- environmentVariables some stix:dictionary
- extensions some stix:dictionary
- isHidden some xsd:boolean
- name some xsd:string
- pid some xsd:integer
- type some {"process"^^xsd:string}

Table 46 Properties of Process cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be process.
extensions (optional)	dictionary	The Process Object defines the following extensions. In addition to these, producers MAY create their own. windows-process-ext, windows-service-ext Dictionary keys MUST identify the extension type by name. The corresponding dictionary values MUST contain the contents of the extension instance.
is_hidden (optional)	boolean	Specifies whether the process is hidden.
pid (optional)	integer	Specifies the Process ID, or PID, of the process.
name (optional)	string	Specifies the name of the process.
created (optional)	timestamp	Specifies the date/time at which the process was created.

cwd (optional)	string	Specifies the current working directory of the process.
arguments (optional)	list of type string	Specifies the list of arguments used in executing the process. Each argument MUST be captured separately as a string.
command_line (optional)	string	Specifies the full command line used in executing the process, including the process name (depending on the operating system).
environment_variables (optional)	dictionary	Specifies the list of environment variables associated with the process as a dictionary. Each key in the dictionary MUST be a case preserved version of the name of the environment variable, and each corresponding value MUST be the environment variable value as a string.
opened_connection_refs (optional)	list of type object-ref	Specifies the list of network connections opened by the process, as a reference to one or more Network Traffic Objects. The objects referenced in this list MUST be of type network-traffic.
creator_user_ref (optional)	object-ref	Specifies the user that created the process, as a reference to a User Account Object. The object referenced in this property MUST be of type user-account.
binary_ref (optional)	object-ref	Specifies the executable binary that was executed as the process, as a reference to a File Object. The object referenced in this property MUST be of type file.
parent_ref (optional)	object-ref	Specifies the other process that spawned (i.e. is the parent of) this one, as reference to a Process Object. The object referenced in this property MUST be of type process.
child_refs (optional)	list of type object-ref	Specifies the other processes that were spawned by (i.e. children of) this process, as a reference to one or more other Process Objects. The objects referenced in this list MUST be of type process.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- binaryRef some File
- childRefs some Process
- creatorUserRef some UserAccount
- openedConnectionRefs some NetworkTraffic

- owl:Thing or (objectRefs some MarkingDefinition)
- parentRef some Process

This class has two subclasses:

- Windows™ Process Extension
- Windows™ Service Extension

B.V.XIII.1. WINDOWS™ PROCESS EXTENSION

“The Windows Process extension specifies a default extension for capturing properties specific to Windows processes”.

Every instance of the class is characterized by the following properties:

- arguments some stix:list
- arguments some xsd:string
- commandLine some xsd:string
- created some xsd:dateTimeStamp
- cwd some xsd:string
- environmentVariables some stix:dictionary
- extensions some stix:dictionary
- isHidden some xsd:boolean
- name some xsd:string
- pid some xsd:integer
- type some {"process"^^xsd:string}
- extensions some {"windows-process-ext"^^xsd:string}
- aslr_enabled some xsd:boolean
- dep_enabled some xsd:boolean
- priority some xsd:string
- owner_sid some xsd:string
- window_title some xsd:string
- startup_info some stix:dictionary

Table 47 Properties of Windows process extension class

Property Name	Type	Description
aslr_enabled (optional)	boolean	Specifies whether Address Space Layout Randomization (ASLR) is enabled for the process.
dep_enabled (optional)	boolean	Specifies whether Data Execution Prevention (DEP) is enabled for the process.

priority (optional)	string	Specifies the current priority class of the process in Windows. This value SHOULD be a string that ends in _CLASS.
owner_sid (optional)	string	Specifies the Security ID (SID) value of the owner of the process.
window_title (optional)	string	Specifies the title of the main window of the process.
startup_info (optional)	dictionary	Specifies the STARTUP_INFO struct used by the process, as a dictionary. Each name/value pair in the struct MUST be represented as a key/value pair in the dictionary, where each key MUST be a case-preserved version of the original name. For example, given a name of "lpDesktop" the corresponding key would be lpDesktop.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of Process
- binaryRef some File
- childRefs some Process
- creatorUserRef some UserAccount
- openedConnectionRefs some NetworkTraffic
- owl:Thing or (objectRefs some MarkingDefinition)
- parentRef some Process

This class has no subclasses.

B.V.XIII.2. WINDOWS™ SERVICE EXTENSION

“The Windows Service extension specifies a default extension for capturing properties specific to Windows services.”

Every instance of the class is characterized by the following properties:

- arguments some stix:list
- arguments some xsd:string
- commandLine some xsd:string
- created some xsd:dateTimeStamp
- cwd some xsd:string
- environmentVariables some stix:dictionary
- extensions some stix:dictionary
- isHidden some xsd:boolean
- name some xsd:string
- pid some xsd:integer
- type some {"process"^^xsd:string}

- extensions some {"windows-service-ext"^^xsd:string}
- service_name only xsd:string
- descriptions some stix:list
- display_name some xsd:string
- group_name some xsd:string
- start_type some stix:windows-service-start-type-enum
- service_dll_refs some stix:list
- service_type some stix:windows-service-type-enum
- service_status some stix:windows-service-status-enum

Table 48 Properties of Windows Service extension class

Property Name	Type	Description
service_name (required)	string	Specifies the name of the service.
descriptions (optional)	list of type string	Specifies the descriptions defined for the service.
display_name (optional)	string	Specifies the displayed name of the service in Windows GUI controls.
group_name (optional)	string	Specifies the name of the load ordering group of which the service is a member.
start_type (optional)	windows-service-start-type-enum	Specifies the start options defined for the service.
service_dll_refs (optional)	list of type object-ref	Specifies the DLLs loaded by the service, as a reference to one or more File Objects. The objects referenced in this property MUST be of type file.
service_type (optional)	windows-service-type-enum	Specifies the type of the service.
service_status (optional)	windows-service-status-enum	Specifies the current status of the service.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of Process
- binaryRef some File
- childRefs some Process
- creatorUserRef some UserAccount
- openedConnectionRefs some NetworkTraffic

- owl:Thing or (objectRefs some MarkingDefinition)
- parentRef some Process

This class has no subclasses.

B.V.XIV. SOFTWARE

“The Software Object represents high-level properties associated with software, including software products”.

Every instance of the class is characterized by the following properties:

- cpe some xsd:string
- languages some stix:list
- languages some xsd:string
- name only xsd:string
- type some {"software"^^xsd:string}
- vendor some xsd:string
- version some xsd:string

Table 49 Properties of SW cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be software.
name (required)	string	Specifies the name of the software.
cpe (optional)	string	Specifies the Common Platform Enumeration (CPE) entry for the software, if available. The value for this property MUST be a CPE v2.3 entry from the official NVD CPE Dictionary. While the CPE dictionary does not contain entries for all software, whenever it does contain an identifier for a given instance of software, this property SHOULD be present.
languages (optional)	list of type string	Specifies the languages supported by the software. The value of each list member MUST be an ISO 639-2 language code ISO639-2.
vendor (optional)	string	Specifies the name of the vendor of the software.
version (optional)	string	Specifies the version of the software

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XV. URL

“The URL Object represents the properties of a uniform resource locator (URL)”.

Every instance of the class is characterized by the following properties:

- type some {"url"^^xsd:string}
- value only xsd:string

where:

Table 50 Properties of URL cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be url.
value (required)	string	Specifies the value of the URL. The value of this property MUST conform to RFC3986, more specifically section 1.1.3 with reference to the definition for "Uniform Resource Locator".

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XVI. USER ACCOUNT

“The User Account Object represents an instance of any type of user account, including but not limited to operating system, device, messaging service, and social media platform accounts”.

Every instance of the class is characterized by the following properties:

- accountCreated some xsd:dateTimeStamp
- accountExpires some xsd:dateTimeStamp
- accountFirstLogin some xsd:dateTimeStamp
- accountLastLogin some xsd:dateTimeStamp
- accountLogin some xsd:string
- accountType some stix:openVocab
- canEscalatePrivs some xsd:boolean
- displayName some xsd:string
- extensions some stix:dictionary
- isDisabled some xsd:boolean
- isPrivileged some xsd:boolean
- isServiceAccount some xsd:boolean
- passwordLastChanged some xsd:dateTimeStamp
- type some {"user-account"^^xsd:string}
- userId only xsd:string

Table 51 Properties of User Account cyber observable class (STIX2.owl)

Property Name	Type	Description
type (required)	string	The value of this property MUST be user-account.
extensions (optional)	dictionary	<p>The User Account Object defines the following extensions. In addition to these, producers MAY create their own.</p> <p>unix-account-ext</p> <p>Dictionary keys MUST identify the extension type by name.</p> <p>The corresponding dictionary values MUST contain the contents of the extension instance.</p>
user_id (required)	string	Specifies the identifier of the account. The format of the identifier depends on the system the user account is maintained in, and may be a numeric ID, a GUID, an account name, an email address, etc. The user_id property should be populated with whatever field is the unique identifier for the system the account is a member of. For example, for UNIX systems it would be populated with the UID.
account_login (optional)	string	<p>Specifies the account login string, used in cases where the user_id property specifies something other than what a user would type when they login.</p> <p>For example, in the case of a Unix account with user_id 0, the account_login might be "root".</p>
account_type (optional)	open-vocab	<p>Specifies the type of the account.</p> <p>This is an open vocabulary and values SHOULD come from the account-type-ov vocabulary.</p>
display_name (optional)	string	<p>Specifies the display name of the account, to be shown in user interfaces, if applicable.</p> <p>On Unix, this is equivalent to the GECOS field.</p>
is_service_account (optional)	boolean	Indicates that the account is associated with a network service or system process (daemon), not a specific individual.
is_privileged (optional)	boolean	Specifies that the account has elevated privileges (i.e., in the case of root on Unix or the Windows Administrator account).
can_escalate_privs (optional)	boolean	Specifies that the account has the ability to escalate privileges (i.e., in the case of sudo on Unix or a Windows Domain Admin account)
is_disabled (optional)	boolean	Specifies if the account is disabled.

account_created (optional)	timestamp	Specifies when the account was created.
account_expires (optional)	timestamp	Specifies the expiration date of the account.
password_last_changed (optional)	timestamp	Specifies when the account password was last changed.
account_first_login (optional)	timestamp	Specifies when the account was first accessed.
account_last_login (optional)	timestamp	Specifies when the account was last accessed

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XVII. WINDOWS REGISTRY KEY

"The Registry Key Object represents the properties of a Windows registry key."

Every instance of the class is characterized by the following properties:

- accountCreated some xsd:dateTimeStamp
- accountExpires some xsd:dateTimeStamp
- accountFirstLogin some xsd:dateTimeStamp
- accountLastLogin some xsd:dateTimeStamp
- accountLogin some xsd:string
- accountType some stix:openVocab
- canEscalatePrivs some xsd:boolean
- displayName some xsd:string
- extensions some stix:dictionary
- isDisabled some xsd:boolean
- isPrivileged some xsd:boolean
- isServiceAccount some xsd:boolean
- passwordLastChanged some xsd:dateTimeStamp
- type some {"user-account"^^xsd:string}
- userId only xsd:string

where:

Table 52 Properties of Windows Registry Key cyber observable class (STIX2.owl)

Property Name	Type	Description
---------------	------	-------------

type (required)	string	The value of this property MUST be windows-registry-key.
key (required)	string	Specifies the full registry key including the hive. The value of the key, including the hive portion, SHOULD be case-preserved. The hive portion of the key MUST be fully expanded and not truncated; e.g., HKEY_LOCAL_MACHINE must be used instead of HKLM.
values (optional)	list of type windows-registry-value-type	Specifies the values found under the registry key.
modified (optional)	timestamp	Specifies the last date/time that the registry key was modified.
creator_user_ref (optional)	object-ref	Specifies a reference to the user account (represented as a User Account Object) that created the registry key. The object referenced in this property MUST be of type user-account.
number_of_subkeys (optional)	integer	Specifies the number of subkeys contained under the registry key.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.V.XVIII. X509 CERTIFICATE

“The X.509 Certificate Object represents the properties of an X.509 certificate, as defined by ITU recommendation X.509 [X.509]”.

Every instance of the class is characterized by the following properties:

- hashes some stix:hashes
- isSelfSigned some xsd:boolean
- issuer some xsd:string
- serialNumber some xsd:string
- signatureAlgorithm some xsd:string
- subject some xsd:string
- subjectPublicKeyAlgorithm some xsd:string
- subjectPublicKeyExponent some xsd:integer
- subjectPublicKeyModulus some xsd:string
- type some {"x509-certificate"^^xsd:string}
- validityNotAfter some xsd:dateTimeStamp
- validityNotBefore some xsd:dateTimeStamp

- version some xsd:string
- x509V3Extensions some stix:x509V3ExtensionsType

where:

Table 53 properties of X509 Certificate cyber observable

Property Name	Type	Description
type (required)	string	The value of this property MUST be x509-certificate.
is_self_signed (optional)	boolean	Specifies whether the certificate is self-signed, i.e., whether it is signed by the same entity whose identity it certifies.
hashes (optional)	hashes	Specifies any hashes that were calculated for the entire contents of the certificate.
version (optional)	string	Specifies the version of the encoded certificate.
serial_number (optional)	string	Specifies the unique identifier for the certificate, as issued by a specific Certificate Authority.
signature_algorithm (optional)	string	Specifies the name of the algorithm used to sign the certificate.
issuer (optional)	string	Specifies the name of the Certificate Authority that issued the certificate.
validity_not_before (optional)	timestamp	Specifies the date on which the certificate validity period begins.
validity_not_after (optional)	timestamp	Specifies the date on which the certificate validity period ends.
subject (optional)	string	Specifies the name of the entity associated with the public key stored in the subject public key field of the certificate.
subject_public_key_algorithm (optional)	string	Specifies the name of the algorithm with which to encrypt data being sent to the subject.
subject_public_key_modulus (optional)	string	Specifies the modulus portion of the subject's public RSA key.
subject_public_key_exponent (optional)	integer	Specifies the exponent portion of the subject's public RSA key, as an integer.
x509_v3_extensions (optional)	x509-v3-extensions-type	Specifies any standard X.509 v3 extensions that may be used in the certificate.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- owl:Thing or (objectRefs some MarkingDefinition)

This class has one subclass:

- x509-v3-extensions-type

B.V.XVIII.1. X509-V3-EXTENSIONS-TYPE

“The X.509 v3 Extensions type captures properties associated with X.509 v3 extensions, which serve as a mechanism for specifying additional information such as alternative subject names”.

Every instance of the class is characterized by the following properties:

- hashes some stix:hashes
- isSelfSigned some xsd:boolean
- issuer some xsd:string
- serialNumber some xsd:string
- signatureAlgorithm some xsd:string
- subject some xsd:string
- subjectPublicKeyAlgorithm some xsd:string
- subjectPublicKeyExponent some xsd:integer
- subjectPublicKeyModulus some xsd:string
- type some {"x509-certificate"^^xsd:string}
- validityNotAfter some xsd:dateTimeStamp
- validityNotBefore some xsd:dateTimeStamp
- version some xsd:string
- x509V3Extensions some stix:x509V3ExtensionsType
- basic_constraints some xsd:string
- name_constraints some xsd:string
- policy_constraints some xsd:string
- key_usage some xsd:string
- extended_key_usage some xsd:string
- subject_key_identifier some xsd:string
- authority_key_identifier some xsd:string
- subject_alternative_name some xsd:string
- issuer_alternative_name some xsd:string
- subject_directory_attributes some xsd:string
- crl_distribution_points some xsd:string
- inhibit_any_policy some xsd:string
- private_key_usage_period_not_before some xsd:dateTimeStamp
- private_key_usage_period_not_after some xsd:dateTimeStamp

- certificate_policies some xsd:string
- policy_mappings some xsd:string

where:

Table 54 Properties of X509 v3 extension type

Property Name	Type	Description
basic_constraints (optional)	string	Specifies a multi-valued extension which indicates whether a certificate is a CA certificate. The first (mandatory) name is CA followed by TRUE or FALSE. If CA is TRUE then an optional pathlen name followed by a non-negative value can be included. Also equivalent to the object ID (OID) value of 2.5.29.19.
name_constraints (optional)	string	Specifies a namespace within which all subject names in subsequent certificates in a certification path MUST be located. Also equivalent to the object ID (OID) value of 2.5.29.30.
policy_constraints (optional)	string	Specifies any constraints on path validation for certificates issued to CAs. Also equivalent to the object ID (OID) value of 2.5.29.36.
key_usage (optional)	string	Specifies a multi-valued extension consisting of a list of names of the permitted key usages. Also equivalent to the object ID (OID) value of 2.5.29.15.
extended_key_usage (optional)	string	Specifies a list of usages indicating purposes for which the certificate public key can be used for. Also equivalent to the object ID (OID) value of 2.5.29.37.
subject_key_identifier (optional)	string	Specifies the identifier that provides a means of identifying certificates that contain a particular public key. Also equivalent to the object ID (OID) value of 2.5.29.14.
authority_key_identifier (optional)	string	Specifies the identifier that provides a means of identifying the public key corresponding to the private key used to sign a certificate. Also equivalent to the object ID (OID) value of 2.5.29.35.
subject_alternative_name (optional)	string	Specifies the additional identities to be bound to the subject of the certificate. Also equivalent to the object ID (OID) value of 2.5.29.17.
issuer_alternative_name (optional)	string	Specifies the additional identities to be bound to the issuer of the certificate. Also equivalent to the object ID (OID) value of 2.5.29.18.
subject_directory_attributes (optional)	string	Specifies the identification attributes (e.g., nationality) of the subject. Also equivalent to the object ID (OID) value of 2.5.29.9.

crl_distribution_points (optional)	string	Specifies how CRL information is obtained. Also equivalent to the object ID (OID) value of 2.5.29.31.
inhibit_any_policy (optional)	string	Specifies the number of additional certificates that may appear in the path before anyPolicy is no longer permitted. Also equivalent to the object ID (OID) value of 2.5.29.54.
private_key_usage_period_not_before (optional)	timestamp	Specifies the date on which the validity period begins for the private key, if it is different from the validity period of the certificate.
private_key_usage_period_not_after (optional)	timestamp	Specifies the date on which the validity period ends for the private key, if it is different from the validity period of the certificate.
certificate_policies (optional)	string	Specifies a sequence of one or more policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. Also equivalent to the object ID (OID) value of 2.5.29.32.
policy_mappings (optional)	string	Specifies one or more pairs of OIDs; each pair includes an issuerDomainPolicy and a subjectDomainPolicy. The pairing indicates whether the issuing CA considers its issuerDomainPolicy equivalent to the subject CA's subjectDomainPolicy. Also equivalent to the object ID (OID) value of 2.5.29.33.

This class has the following relationships with other classes:

- SubClass of CyberObservable
- SubClass of X509Certificate
- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.VI. DICTIONARY

"A dictionary captures an arbitrary set of key/value pairs. Dictionary keys must be unique in each dictionary, must be in ASCII, and are limited to the characters a-z (lowercase ASCII), A-Z (uppercase ASCII), numerals 0-9, hyphen (-), and underscore (_). dictionary keys should be no longer than 30 ASCII characters in length, must have a minimum length of 3 ASCII characters, must be no longer than 256 ASCII characters in length, and should be lowercase. Dictionary values must be valid property base types."

Every instance of the class is characterized by the following properties:

- type only xsd:string

Table 55 Properties of Dictionary

Property Name	Type	Description
type (required)	string	The value of this property MUST be dictionary
key (required)	string	Key for the value.

value (required)	string	Value for the key.
-------------------------	--------	--------------------

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.VII. IDENTITY

“Identities can represent actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, or groups (e.g., the finance sector).

The Identity SDO can capture basic identifying information, contact information, and the sectors that the Identity belongs to. Identity is used in STIX to represent, among other things, targets of attacks, information sources, object creators, and threat actor identities”

Every instance of the class is characterized by the following properties:

- contactInformation some xsd:string
- description some xsd:string
- identityClass only stix:openVocab
- labels some stix:list
- name only xsd:string
- sectors some stix:list
- type only {"identity"^^xsd:string}

Table 56 Properties of Identity

Property Name	Type	Description
type (required)	string	The value of this property MUST be identity
labels (optional)	list of type string	The list of roles that this Identity performs (e.g., CEO, Domain Administrators, Doctors, Hospital, or Retailer). No open vocabulary is yet defined for this property.
name (required)	string	The name of this Identity. When referring to a specific entity (e.g., an individual or organization), this property SHOULD contain the canonical name of the specific entity.
description (optional)	string	A description that provides more details and context about the Identity, potentially including its purpose and its key characteristics.
identity_class (required)	open-vocab	The type of entity that this Identity describes, e.g., an individual or organization. This is an open vocabulary and the values SHOULD come from the identity-class-ov vocabulary.

sectors (optional)	list of type open-vocab	The list of industry sectors that this Identity belongs to. This is an open vocabulary and values SHOULD come from the industry-sector-ov vocabulary.
contact_information (optional)	string	The contact information (e-mail, phone number, etc.) for this Identity. No format for this information is currently defined by this specification.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- hasAttributionOf some ThreatActor
- isImpersonatedBy some ThreatActor
- isTargetedBy some AttackPattern
- isTargetedBy some Campaign
- isTargetedBy some IntrusionSet
- isTargetedBy some Malware
- isTargetedBy some ThreatActor
- isTargetedBy some Tool

This class has no subclasses.

B.VIII. INCIDENT

"Reserved object" by STIX™ v2. SecurityEvents will be defined within DRM ontology, at least to work with events (which are a potential incident). This class has no further development on purpose, as **it is reserved**.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.IX. INDICATOR

"Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity."

In our case, we will use SWRL rules to define the patterns instead. The indicator is a class and there will be instances of that class when detections from SWRL rules have been found (e.g. security event).

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name some xsd:string
- pattern only xsd:string

- type only {"indicator"^^xsd:string}
- validFrom only xsd:dateTimeStamp
- validUntil some xsd:dateTimeStamp

Table 57 Properties of Indicator

Property Name	Type	Description
type (required)	string	The value of this property MUST be indicator
labels (required)	list of type open-vocab	This property is an Open Vocabulary that specifies the type of indicator. This is an open vocabulary and values SHOULD come from the indicator-label-ov vocabulary.
name (optional)	string	A name used to identify the Indicator.
description (optional)	string	A description that provides more details and context about the Indicator, potentially including its purpose and its key characteristics.
pattern (required)	string	The detection pattern for this Indicator is a STIX Pattern.
valid_from (required)	timestamp	The time from which this Indicator should be considered valuable intelligence.
valid_until (optional)	timestamp	The time at which this Indicator should no longer be considered valuable intelligence. If the valid_until property is omitted, then there is no constraint on the latest time for which the Indicator should be used.
kill_chain_phases (optional)	list of type kill-chain-phase	The kill chain phase(s) to which this Indicator corresponds.

This class has the following relationships with other classes:

- createdByRef some Identity
- owl:Thing or (objectRefs some MarkingDefinition)
- indicates some AttackPattern
- indicates some Campaign
- indicates some IntrusionSet
- indicates some Malware
- indicates some ThreatActor
- indicates some Tool

This class has no subclasses.

B.X. INFRASTRUCTURE

"Reserved object" by STIX™ v2. This class has no further development on purpose, as it is reserved.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.XI. INTRUSION SET

"An Intrusion Set is a grouped set of adversary behavior and resources with common properties that is believed to be orchestrated by a single organization."

Every instance of the class is characterized by the following properties:

- aliases some stix:list
- aliases some xsd:string
- description some xsd:string
- firstSeen some xsd:dateTimeStamp
- goals some stix:list
- goals some xsd:string
- lastSeen some xsd:dateTimeStamp
- name only xsd:string
- primaryMotivation some stix:openVocab
- resourceLevel some stix:openVocab
- secondaryMotivations some stix:list
- secondaryMotivations some stix:openVocab
- type only {"intrusion-set"^^xsd:string}

Table 58 Properties of Intrusion set

Property Name	Type	Description
type (required)	string	The value of this property MUST be intrusion-set.
name (required)	string	A name used to identify this Intrusion Set.
description (optional)	string	A description that provides more details and context about the Intrusion Set, potentially including its purpose and its key characteristics.
aliases (optional)	list of type string	Alternative names used to identify this Intrusion Set.
first_seen (optional)	timestamp	The time that this Intrusion Set was first seen. This property is a summary property of data from sightings and other data that may or may not be

		available in STIX. If new sightings are received that are earlier than the first seen timestamp, the object may be updated to account for the new data.
last_seen (optional)	timestamp	<p>The time that this Intrusion Set was last seen.</p> <p>This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are later than the last seen timestamp, the object may be updated to account for the new data.</p>
goals (optional)	list of type string	<p>The high level goals of this Intrusion Set, namely, what are they trying to do. For example, they may be motivated by personal gain, but their goal is to steal credit card numbers. To do this, they may execute specific Campaigns that have detailed objectives like compromising point of sale systems at a large retailer.</p> <p>Another example: to gain information about latest merger and IPO information from ACME Bank.</p>
resource_level (optional)	open-vocab	<p>This defines the organizational level at which this Intrusion Set typically works, which in turn determines the resources available to this Intrusion Set for use in an attack.</p> <p>This is an open vocabulary and values SHOULD come from the attack-resource-level-ov vocabulary.</p>
primary_motivation (optional)	open-vocab	<p>The primary reason, motivation, or purpose behind this Intrusion Set. The motivation is why the Intrusion Set wishes to achieve the goal (what they are trying to achieve).</p> <p>For example, an Intrusion Set with a goal to disrupt the finance sector in a country might be motivated by ideological hatred of capitalism.</p> <p>This is an open vocabulary and values SHOULD come from the attack-motivation-ov vocabulary.</p>
secondary_motivations (optional)	list of type open-vocab	<p>The secondary reasons, motivations, or purposes behind this Intrusion Set. These motivations can exist as an equal or near-equal cause to the primary motivation. However, it does not replace or necessarily magnify the primary motivation, but it might indicate additional context.</p> <p>This is an open vocabulary and values SHOULD come from the attack-motivation-ov vocabulary.</p>

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- attributedTo some ThreatActor
- createdByRef some Identity
- hasAttributionOf some Campaign
- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some AttackPattern
- uses some Malware
- uses some Tool

This class has no subclasses.

B.XII. KILL CHAIN PHASE

“The Kill Chain Phase represents a phase in a kill chain, which describes the various phases an attacker may undertake in order to achieve their objectives”.

Every instance of the class is characterized by the following properties:

- killChainName only xsd:string
- phaseName only xsd:string

Table 59 Properties of Kill Chain Phase

Property Name	Type	Description
kill_chain_name (required)	string	The name of the kill chain. The value of this property SHOULD be all lowercase (where lowercase is defined by the locality conventions) and SHOULD use dashes instead of spaces or underscores as word separators.
phase_name (required)	string	The name of the phase in the kill chain. The value of this property SHOULD be all lowercase (where lowercase is defined by the locality conventions) and SHOULD use dashes instead of spaces or underscores as word separators.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

B.XIII. MALWARE

"Malware is a type of TTP that is also known as malicious code and malicious software, refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

Table 60 Properties of Malware

Property Name	Type	Description
type (required)	string	The value of this property MUST be malware.
labels (required)	list of type open-vocab	The type of malware being described. This is an open vocabulary and values SHOULD come from the malware-label-ov vocabulary.
name (required)	string	A name used to identify the Malware sample.
description (optional)	string	A description that provides more details and context about the Malware, potentially including its purpose and its key characteristics.
kill_chain_phases (optional)	list of type kill-chain-phase	The list of Kill Chain Phases for which this Malware can be used.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has multiple subclasses:

- ADWARE

- BACKDOOR
- BOT
- DDOS
- DROPPER
- EXPLOITKIT
- KEYLOGGER
- RANSOMWARE
- REMOTEACCESSTROJAN
- RESOURCE EXPLOITATION
- ROGUE SECURITY SOFTWARE
- ROOTKIT
- SCREEN CAPTURE
- SPYWARE
- TROJAN
- VIRUS
- WORM

B.XIII.I. ADWARE

"Any software that is funded by advertising. Adware may also gather sensitive user information from a system."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity

- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.II. BACKDOOR

"A malicious program that allows an attacker to perform actions on a remote system, such as transferring files, acquiring passwords, or executing arbitrary commands".

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.III. BOT

"A program that resides on an infected system, communicating with and forming part of a botnet. The bot may be implanted by a worm or Trojan, which opens a backdoor. The bot then monitors the backdoor for further instructions."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list

- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.IV. DDOS

"A tool used to perform a distributed denial of service attack".

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity

- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.V. DROPPER

"A type of trojan that deposits an enclosed payload (generally, other malware) onto the target computer"

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.VI. EXPLOIT KIT

"A software toolkit to target common vulnerabilities."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string

- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.VII. KEYLOGGER

"A type of malware that surreptitiously monitors keystrokes and either records them for later retrieval or sends them back to a central collection point."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity

- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.VIII. RANSOMWARE

"A type of malware that encrypts files on a victim's system, demanding payment of ransom in return for the access codes required to unlock files"

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.IX. REMOTE ACCESS TROJAN

"A remote access trojan program (or RAT), is a trojan horse capable of controlling a machine through commands issued by a remote attacker."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string

- type only {"malware"^^xsd:string}
- This class has the following relationships with other classes:
- owl:Thing or (objectRefs some MarkingDefinition)
 - createdByRef some Identity
 - isIndicatedBy some Indicator
 - isMitigatedBy some CourseOfAction
 - isUsedBy some AttackPattern
 - isUsedBy some Campaign
 - isUsedBy some IntrusionSet
 - isUsedBy some ThreatActor
 - objectMarkingRefs some MarkingDefinition
 - targets some Identity
 - targets some Vulnerability
 - uses some Tool
 - variantOf some Malware

This class has no subclasses.

B.XIII.X. RESOURCE EXPLOITATION

"A type of malware that steals a system's resources (e.g., CPU cycles), such as a bitcoin miner."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity

- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XI. ROGUE SECURITY SOFTWARE

"A fake security product that demands money to clean phony infections."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XII. ROOTKIT

"A type of malware that hides its files or processes from normal methods of monitoring in order to conceal its presence and activities. Rootkits can operate at a number of levels, from the application level — simply replacing or adjusting the settings of system software to prevent the display of certain information — through hooking certain functions or inserting modules or drivers into the operating system kernel, to the deeper level of firmware or virtualization rootkits, which are activated before the operating system and thus even harder to detect while the system is running."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase

- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XIII. SCREEN CAPTURE

"A type of malware used to capture images from the target systems screen, used for exfiltration and command and control."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet

- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XIV. SPYWARE

"Software that gathers information on a user's system without their knowledge and sends it to another party. Spyware is generally used to track activities for the purpose of delivering advertising."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XV. TROJAN

"Any malicious computer program which is used to hack into a computer by misleading users of its true intent."

Every instance of the class is characterized by the following properties:

- description some xsd:string

- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XVI. VIRUS

"A malicious computer program that replicates by reproducing itself or infecting other programs by modifying them."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign

- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIII.XVII. WORM

"A self-replicating, self-contained program that usually executes itself without user intervention."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list
- labels only stix:list
- name only xsd:string
- type only {"malware"^^xsd:string}

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some Tool
- variantOf some Malware

This class has no subclasses.

B.XIV. MARKING DEFINITION

"The marking-definition object represents a specific marking."

Data markings represent restrictions, permissions, and other guidance for how data can be used and shared. For example, data may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest.

Some types of marking definitions or trust groups have rules about which markings override other markings or which markings can be additive to other markings. This specification does not define rules for how multiple markings applied to the same object or property should be interpreted.

Two marking definition types are defined in this specification: TLP, to capture TLP markings, and Statement, to capture text marking statements. In addition, it is expected that the FIRST Information Exchange Policy (IEP) will be included in a future version once a machine-usable specification for it has been defined".

Every instance of the class is characterized by the following properties:

- type only xsd:string
- id only xsd:string
- created_by_ref some xsd:identifier
- created only xsd:dateTimeStamp
- external_references some stix:list
- object_marking_refs some stix:list
- granular_markings some stix:list
- definition_type only stix:openVocab
- definition only <marking object> (e.g. TLP or Statement, as subclasses)

Table 61 Properties of Marking Definition

Property Name	Type	Description
type (required)	string	The type property identifies the type of object. The value of this property MUST be marking-definition.
id (required)	identifier	The id property universally and uniquely identifies this Marking Definition. Because the object type is part of the identifier, it is not possible for objects of different types to share the same id.
created_by_ref (optional)	identifier	The created_by_ref property specifies the ID of the identity object that describes the entity that created this Marking Definition. If this attribute is omitted, the source of this information is undefined. This may be used by object creators who wish to remain anonymous.
created (required)	timestamp	The created property represents the time at which the Marking Definition was created. The object creator can use the time it deems most appropriate as the time the object was created.

external_references (optional)	list of type external-reference	The external_references property specifies a list of external references which refers to non-STIX information. This property is used to provide one or more URLs, descriptions, or IDs to records in other systems.
object_marking_refs (optional)	list of type identifier	The object_marking_refs property specifies a list of IDs of marking-definitions that apply to this Marking Definition. This property MUST NOT contain any references to this Marking Definition object (i.e., it cannot contain any circular references). Though uncommon, in some cases marking definitions themselves may be marked with sharing or handling guidance.
granular_markings (optional)	list of type granular-marking	The granular_markings property specifies a list of granular markings applied to this. This property MUST NOT contain any references to this Marking Definition object (i.e., it cannot contain any circular references). Though uncommon, in some cases Marking Definitions themselves may be marked with sharing or handling guidance.
definition_type (required)	open-vocab	The definition_type property identifies the type of Marking Definition. The value of the definition_type property SHOULD be one of the types defined in the subsections below: statement or tlp
definition (required)	<marking object>	The definition property contains the marking object itself (e.g., the TLP).

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has two subclasses:

- TLP
- Statement

B.XIV.I.1. TLP

“The TLP marking type defines how you would represent a Traffic Light Protocol (TLP) marking in a definition property. The value of the definition_type property MUST be tlp when using this marking type”.

Every instance of the class is characterized by the following properties:

- type only xsd:string
- id only xsd:string
- created_by_ref some xsd:identifier
- created only xsd:dateTimeStamp

- external_references some stix:list
- object_marking_refs some stix:list
- granular_markings some stix:list
- definition_type only stix:openVocab
- definition only <marking object> (e.g. TLP or Statement, as subclasses)
- tlp only xsd:string

Table 62 Properties of TLP

Property Name	Type	Description
tlp (required)	string	The TLP level of the content marked by this marking definition, as defined in this section.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- SubClass of MarkingDefinition

This class has no subclasses.

B.XIV.I.2. STATEMENT

"The Statement marking type defines the representation of a textual marking statement (e.g., copyright, terms of use, etc.) in a definition. The value of the definition_type property MUST be statement when using this marking type. Statement markings are generally not machine-readable and this specification does not define any behavior or actions based on their values.

Content may be marked with multiple Statement marking types that do not override each other. In other words, the same content can be marked both with a statement saying "Copyright 2016" and a statement saying "Terms of use are ..." and both statements apply".

Every instance of the class is characterized by the following properties:

- type only xsd:string
- id only xsd:string
- created_by_ref some xsd:identifier
- created only xsd:dateTimeStamp
- external_references some stix:list
- object_marking_refs some stix:list
- granular_markings some stix:list
- definition_type only stix:openVocab
- definition only <marking object> (e.g. TLP or Statement, as subclasses)
- statement only xsd:string

Table 63 Properties of Statements

Property Name	Type	Description
---------------	------	-------------

statement (required)	string	A Statement (e.g., copyright, terms of use) applied to the content marked by this marking definition.
--------------------------------	--------	---

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- SubClass of MarkingDefinition

This class has no subclasses.

B.XV. OBSERVED DATA

"Observed data conveys information that was observed on systems and networks, such as log data or network traffic, using the Cyber Observable specification."

Every instance of the class is characterized by the following properties:

- firstObserved only xsd:dateTimeStamp
- lastObserved only xsd:dateTimeStamp
- numberObserved only xsd:integer
- objects only stix:observableObjects
- type only {"observed-data"^^xsd:string}

Table 64 Properties of Observed data

Property Name	Type	Description
type (required)	string	The value of this property MUST be observed-data.
first_observed (required)	timestamp	The beginning of the time window during which the data was observed.
last_observed (required)	timestamp	The end of the time window during which the data was observed.
number_observed (required)	integer	<p>The number of times the data represented in the objects property was observed. This MUST be an integer between 1 and 999,999,999 inclusive.</p> <p>If the number_observed property is greater than 1, the data contained in the objects property was observed multiple times. In these cases, object creators MAY omit properties of the Cyber Observable object (such as timestamps) that are specific to a single instance of that observed data.</p>
objects (required)	observable-objects	<p>A dictionary of Cyber Observable Objects representing the observation. The dictionary MUST contain at least one object.</p> <p>The Cyber Observable content MAY include multiple objects if those objects are related as part of a single observation. Multiple objects not related to each other via</p>

		<p>Cyber Observable Relationships MUST NOT be contained within the same Observed Data instance.</p> <p>For example, a Network Traffic object and two IPv4 Address objects related via the src_ref and dst_ref properties can be contained in the same Observed Data because they are all related and used to characterize that single entity. Two unrelated IPv4 address objects that just happened to be observed at the same time, however, must be represented in separate Observed Data instances.</p>
--	--	--

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

B.XVI. REPORT

"Reports are collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- labels only stix:openVocab
- name only xsd:string
- published only xsd:dateTimeStamp
- type only {"report"^^xsd:string}

Table 65 Properties of Report

Property Name	Type	Description
type (required)	string	The value of this property MUST be report
labels (required)	list of type open-vocab	<p>This property is an Open Vocabulary that specifies the primary subject of this report.</p> <p>This is an open vocabulary and values SHOULD come from the report-label-ov vocabulary.</p>
name (required)	string	A name used to identify the Report.
description (optional)	string	A description that provides more details and context about the Report, potentially including its purpose and its key characteristics.

published (required)	timestamp	<p>The date that this Report object was officially published by the creator of this report.</p> <p>The publication date (public release, legal release, etc.) may be different than the date the report was created or shared internally (the date in the created property).</p>
object_refs (required)	list of type identifier	Specifies the STIX Objects that are referred to by this Report.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

B.XVII. SIGHTING

"A Sighting denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen."

Every instance of the class is characterized by the following properties:

- count some xsd:integer
- firstSeen some xsd:dateTimeStamp
- lastSeen some xsd:dateTimeStamp
- type only {"sighting"^^xsd:string}
- summary some xsd:boolean

Table 66 Properties of Sighting

Property Name	Type	Description
type (required)	string	The value of this property MUST be sighting
first_seen (optional)	timestamp	The beginning of the time window during which the SDO referenced by the sighting_of_ref property was sighted.
last_seen (optional)	timestamp	The end of the time window during which the SDO referenced by the sighting_of_ref property was sighted.
count (optional)	integer	<p>This MUST be an integer between 0 and 999,999,999 inclusive and represents the number of times the SDO referenced by the sighting_of_ref property was sighted.</p> <p>Observed Data has a similar property called number_observed, which refers to the number of times the data was observed. These counts refer to different concepts and are distinct.</p>

		<p>For example, a single sighting of a DDoS bot might have many millions of observations of the network traffic that it generates. Thus, the Sighting count would be 1 (the bot was observed once) but the Observed Data number_observed would be much higher.</p> <p>As another example, a sighting with a count of 0 can be used to express that an indicator was not seen at all.</p>
sighting_of_ref (required)	identifier	<p>An ID reference to the SDO that was sighted (e.g., Indicator or Malware).</p> <p>For example, if this is a Sighting of an Indicator, that Indicator's ID would be the value of this property.</p> <p>This property MUST reference only an SDO or a Custom Object.</p>
observed_data_refs (optional)	list of type identifier	<p>A list of ID references to the Observed Data objects that contain the raw cyber data for this Sighting.</p> <p>For example, a Sighting of an Indicator with an IP address could include the Observed Data for the network connection that the Indicator was used to detect.</p> <p>This property MUST reference only Observed Data SDOs.</p>
where_sighted_refs (optional)	list of type identifier	<p>A list of ID references to the Identity (victim) objects of the entities that saw the sighting.</p> <p>Omitting the where_sighted_refs property does not imply that the sighting was seen by the object creator. To indicate that the sighting was seen by the object creator, an Identity representing the object creator should be listed in where_sighted_refs.</p> <p>This property MUST reference only Identity SDOs.</p>
summary (optional)	boolean	<p>The summary property indicates whether the Sighting should be considered summary data. Summary data is an aggregation of previous Sightings reports and should not be considered primary source data. Default value is false.</p>

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- objectMarkingRefs some MarkingDefinition
- sightingOfRef some owl:Thing
- observedDataRefs some ObservedData
- whereSightedRefs some Identity

This class has no subclasses.

B.XVIII. THREAT ACTOR

"Threat Actors are actual individuals, groups, or organizations believed to be operating with malicious intent."

Every instance of the class is characterized by the following properties:

- aliases some stix:list
- aliases some stix:openVocab
- description some xsd:string
- goals some stix:list
- goals some xsd:string
- labels only stix:list
- labels only stix:openVocab
- name only xsd:string
- personalMotivations some stix:list
- personalMotivations some stix:openVocab
- primaryMotivation some stix:openVocab
- resourceLevel some stix:openVocab
- roles some stix:list
- roles some stix:openVocab
- secondaryMotivations some stix:list
- secondaryMotivations some stix:openVocab
- sophistication some stix:openVocab
- type only {"threat-actor"^^xsd:string}

Table 67 Properties of Threat Actor

Property Name	Type	Description
type (required)	string	The value of this property MUST be threat-actor
labels (required)	list of type open-vocab	This property specifies the type of Threat Actor. This is an open vocabulary and values SHOULD come from the threat-actor-label-ov vocabulary.
name (required)	string	A name used to identify this Threat Actor or Threat Actor group.
description (optional)	string	A description that provides more details and context about the Threat Actor, potentially including its purpose and its key characteristics.

aliases (optional)	list of type string	A list of other names that this Threat Actor is believed to use.
roles (optional)	list of type open-vocab	A list of roles the Threat Actor plays. This is an open vocabulary and the values SHOULD come from the threat-actor-role-ov vocabulary.
goals (optional)	list of type string	The high level goals of this Threat Actor, namely, what are they trying to do. For example, they may be motivated by personal gain, but their goal is to steal credit card numbers. To do this, they may execute specific Campaigns that have detailed objectives like compromising point of sale systems at a large retailer.
sophistication (optional)	open-vocab	The skill, specific knowledge, special training, or expertise a Threat Actor must have to perform the attack. This is an open vocabulary and values SHOULD come from the threat-actor-sophistication-ov vocabulary.
resource_level (optional)	open-vocab	This defines the organizational level at which this Threat Actor typically works, which in turn determines the resources available to this Threat Actor for use in an attack. This attribute is linked to the sophistication property — a specific resource level implies that the Threat Actor has access to at least a specific sophistication level. This is an open vocabulary and values SHOULD come from the attack-resource-level-ov vocabulary.
primary_motivation (optional)	open-vocab	The primary reason, motivation, or purpose behind this Threat Actor. The motivation is why the Threat Actor wishes to achieve the goal (what they are trying to achieve). For example, a Threat Actor with a goal to disrupt the finance sector in a country might be motivated by ideological hatred of capitalism. This is an open vocabulary and values SHOULD come from the attack-motivation-ov vocabulary.
secondary_motivations (optional)	list of type open-vocab	The secondary reasons, motivations, or purposes behind this Threat Actor. These motivations can exist as an equal or near-equal cause to the primary motivation. However, it does not replace or necessarily magnify the primary motivation, but it might indicate additional context.

		This is an open vocabulary and values SHOULD come from the attack-motivation-ov vocabulary.
personal_motivations (optional)	list of type open- vocab	<p>The personal reasons, motivations, or purposes of the Threat Actor regardless of organizational goals.</p> <p>Personal motivation, which is independent of the organization's goals, describes what impels an individual to carry out an attack. Personal motivation may align with the organization's motivation—as is common with activists—but more often it supports personal goals. For example, an individual analyst may join a Data Miner corporation because his or her skills may align with the corporation's objectives. But the analyst most likely performs his or her daily work toward those objectives for personal reward in the form of a paycheck. The motivation of personal reward may be even stronger for Threat Actors who commit illegal acts, as it is more difficult for someone to cross that line purely for altruistic reasons.</p> <p>This is an open vocabulary and values SHOULD come from the attack-motivation-ov vocabulary.</p>

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- attributedTo some Identity
- createdByRef some Identity
- hasAttributionOf some Campaign
- hasAttributionOf some IntrusionSet
- impersonates some Identity
- isIndicatedBy some Indicator
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability
- uses some AttackPattern
- uses some Malware
- uses some Tool

This class has no subclasses.

B.XIX. TOOL

"Tools are legitimate software that can be used by threat actors to perform attacks".

Every instance of the class is characterized by the following properties:

- description some xsd:string
- killChainPhases some stix:killChainPhase
- killChainPhases some stix:list

- labels only stix:list
- name only xsd:string
- toolVersion some xsd:string
- type only {"tool"^^xsd:string}

Table 68 Properties of Tool

Property Name	Type	Description
type (required)	string	The value of this property MUST be tool
labels (required)	list of type open-vocab	The kind(s) of tool(s) being described. This is an open vocabulary and values SHOULD come from the tool-label-ov vocabulary.
name (required)	string	The name used to identify the Tool.
description (optional)	string	A description that provides more details and context about the Tool, potentially including its purpose and its key characteristics.
kill_chain_phases (optional)	list of type kill-chain-phase	The list of kill chain phases for which this Tool can be used.
tool_version (optional)	string	The version identifier associated with the Tool.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isIndicatedBy some Indicator
- isMitigatedBy some CourseOfAction
- isUsedBy some AttackPattern
- isUsedBy some Campaign
- isUsedBy some IntrusionSet
- isUsedBy some Malware
- isUsedBy some ThreatActor
- objectMarkingRefs some MarkingDefinition
- targets some Identity
- targets some Vulnerability

This class has no subclasses.

B.XX. VULNERABILITY

"A Vulnerability is a mistake in software that can be directly used by a hacker to gain access to a system or network."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- external_references some stix:externalReference
- external_references some stix:list
- name only xsd:string
- type only {"vulnerability"^^xsd:string}

Table 69 Properties of Vulnerability

Property Name	Type	Description
type (required)	string	The value of this property MUST be vulnerability
external_references (optional)	list of type external-reference	A list of external references which refer to non-STIX information. This property MAY be used to provide one or more Vulnerability identifiers, such as a CVE ID. When specifying a CVE ID, the source_name property of the external reference MUST be set to cve and the external_id property MUST be the exact CVE identifier.
name (required)	string	A name used to identify the Vulnerability.
description (optional)	string	A description that provides more details and context about the Vulnerability, potentially including its purpose and its key characteristics.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- isMitigatedBy some CourseOfAction
- isTargetedBy some AttackPattern
- isTargetedBy some Campaign
- isTargetedBy some IntrusionSet
- isTargetedBy some Malware
- isTargetedBy some ThreatActor
- isTargetedBy some Tool
- objectMarkingRefs some MarkingDefinition

This class has no subclasses.

ANNEX C. DYNAMIC RISK MANAGEMENT ONTOLOGY DETAILS

C.I. ASSET

"They are the different (valuable) assets integrated within and dependent of a business process."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- name only xsd:string
- number only xsd:integer
- type only xsd:string
- type some {"asset"}
- value only xsd:float

Table 70 Properties of Asset class (DRM.owl)

Property Name	Type	Description
name (required)	string	The name of the asset.
value (required)	float	The value of the asset.
number (required)	integer	The number of asset (reserved).
type (optional)	string	It has to include type = {"asset"^^xsd:string}.
description (optional)	string	Specifies the description of the asset.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- hasCyberObservable some CyberObservable
- hasVulnerability some Vulnerability
- isExposedTo some Threat
- isProtectedBy some Safeguards

This class has several subclasses:

- Data
 - ClassifiedData
 - Equivalent to "File" (from STIX2.owl)
 - PersonalDataProtection
- Hardware
 - BusinessRecovery

- Computer
 - IoT
 - PersonalComputer
 - Server
- MainBackboneHW
- MobileDevices
- NetworkDevice
 - Modem
 - Router
 - Switch
- VirtualMachines
- Networks
 - CellCommunications
 - Internet
 - ISDN
 - LAN
 - PSTN
 - SatelliteCommunications
 - VLAN
 - VPN
 - WLAN
 - xDSL
- Processes : Equivalent to “Process” (STIX2.owl)
 - RelatedProcesses
- RealEstate
 - Building
 - Floor
 - Room
- ServiceDelivery
 - SelfProvided
 - ThirdPartyProvided
- Software
 - AcquiredSW
 - DevelopedSW
- Users
 - ExternalUsers
 - Contractors

- InternalUsers

C.II. ASSET VALUATION

"The value of the assets of the organization based on an added value assessment."

Every instance of the class is characterized by the following properties:

- accounting only xsd:float
- authenticity only xsd:float
- availability only xsd:float
- confidentiality only xsd:float
- informationSecurity only xsd:float
- integrity only xsd:float
- number only xsd:integer
- recoveryPointObjective only xsd:float
- recoveryTimeObjective only xsd:float
- type only xsd:string
- type some {"asset-valuation"}
- value only xsd:float

Table 71 Properties of Asset Valuation class (DRM.owl)

Property Name	Type	Description
accounting (required)	float	Accounting assessment value of the asset.
authenticity (required)	float	Authenticity assessment value of the asset.
availability (required)	float	Availability assessment value of the asset.
confidentiality (required)	float	Confidentiality assessment value of the asset.
integrity (required)	float	Integrity assessment value of the asset.
informationSecurity (required)	float	Information Security assessment value of the asset (as a sum of other parameters).
number (required)	integer	The number of the assessment, in case it is reassessed within the same instance.
recoveryPointObjective (required)	float	RPO assessment value of the asset. It is related to the objective point (in days, minutes, seconds) to recover data from a disaster of data loss (e.g. backup frequency definition).
recoveryTimeObjective (required)	float	RTO assessment value of the asset. It is related to the objective time (in days, minutes, seconds) to recover the service from a disaster (e.g. once a backup is available, it

		is related to the time to restore that backup and the service).
value (required)	string	Specifies the valuation of asset.
type (optional)	string	It has to include type = {"asset-valuation"^^xsd:string}.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only RiskScope
- evaluates some Asset

C.III. CONTEXT

"It is the context (internal or external) including dependencies where our business processes are being deployed."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- name only xsd:string
- type only xsd:string
- type only {"context"}

Table 72 Properties of Context class (DRM.owl)

Property Name	Type	Description
type (required)	string	It has to include type = {"context"^^xsd:string}.
name (required)	string	Specifies the name of the context.
description (optional)	string	Specifies the description of the context.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity

It has multiple subclasses:

- ExternalContext
 - ComplianceContext
 - LegalContext
 - PoliticalContext
- InternalContext
 - RelatedAssets

- StrategicAlignment
 - StrategicIndicator
 - StrategicObjective

C.IV. SAFEGUARDS

"Safeguards to minimize the impact, or the probability, of a specific threat."

Every instance of the class is characterized by the following properties:

- actualControl only xsd:string
- description some xsd:string
- name only xsd:string
- number only xsd:integer
- type only xsd:string
- type some {"safeguards"}
- value only xsd:float

Table 73 Properties of Risk Management class (DRM.owl)

Property Name	Type	Description
actualControl (required)	string	The control (e.g. ISO 27002).
value (required)	float	The value of the reduction (mitigation), either Impact or Probability.
name (required)	string	Name of the safeguard.
number (required)	integer	Number of the safeguard (reserved).
description (optional)	string	Specifies the description of the safeguard.
type (optional)	string	It has to include type = {"safeguards"^^xsd:string}.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- mitigates some RiskImpact
- mitigates some RiskProbability
- mitigates some Vulnerability
- protects some Asset
- relatedTo some CourseOfAction

It has multiple subclasses:

- Control

- NonStandardControl
- StandardControl
 - ISO2700XControl
 - AccessControl
 - AccessControlBusinessRequisitesControl
 - AccessPolicyControl
 - NetworkAccessPolicyControl
 - AccessControlToSystemsandApplications
 - InformationAccessRestrictionControl
 - LoginSecureProcessesControl
 - PasswordManagementControl
 - ProgramSourceCodeAccessControl
 - RootPrivilegeUsedToolsControl
 - EndUserAccessManagementControl
 - PrivilegeManagementControl
 - ProvisionOfUserAccessControl
 - UserAccessRightsCancellationControl
 - UserAccessRightsRevisionControl
 - UserAuthenticationSecretInformationManagementControl
 - UserRegistrationandCancellationControl
 - UserResponsibilityControl
 - AuthenticationSecretInformationUsageControl
 - AssetManagementControl
 - AssetResponsibilityControl
 - AssetAcceptableUsageControl
 - AssetInventoryControl
 - AssetOwnershipControl
 - AssetReturnControl
 - InformationClassificationControl
 - InformationClassifyingControl
 - InformationHandlingControl
 - InformationLabelingControl
 - MediaHandlingControl
 - PhysicalSupportInTransitControl
 - RemovableSupportManagementControl
 - SupportRemovalControl

- BusinessContinuityControl
 - InformationSecurityContinuityControl
 - CheckandMaintenanceOfContinuityPlanControl
 - InformationSecurityContinuityImplementationControl
 - InformationSecurityContinuityPlanControl
 - RedundancyControl
 - BackupFacilityAvailableControl
- ComplianceControl
 - InformationSecurityRevision
 - InformationSecurityIndependentRevision
 - OrganizationPoliciesandGuidelinesCompliance
 - TechnicalComplianceAuditControl
 - LegalRequirementsComplianceControl
 - ApplicableLawIdentificationControl
 - CryptographicControlsRegulation
 - IntellectualPropertyRightsControl
 - OrganizationRegistersProtectionControl
 - PrivacyandDataProtectionControl
- CryptoControl
 - CryptographicControl
 - CryptographicControlUsagePolicy
 - KeysManagementControl
- HRRelatedSecurityControl
 - HRRelatedSecurityAfterChangeJobControl
 - HRRelatedSecurityResponsibilitiesAfterQuittingJobControl
 - HRRelatedSecurityBeforeJobControl
 - HRRelatedSecurityBackgroundResearchControl
 - HRRelatedSecurityTermsandConditionsControl
 - HRRelatedSecurityDuringJobControl
 - HRRelatedSecurityAwarenessEducationandCybercapabilitiesControl
 - HRRelatedSecurityDisciplinaryProcessControl
 - HRRelatedSecurityManagementResponsibilitiesControl
- IncidentHandlingControl
 - IncidentHandlingandImprovementsControl
 - EvidenceCollectionControl

- IncidentHandlingProceduresandResponsibilitiesControl
- IncidentKnowledgeBaseControl
- InformationSecurityEventAssessmentandDecisionMakingControl
- InformationSecurityEventNotificationControl
- InformationSecurityProceduresandResponsibilitiesControl
- SecurityIncidentResponseControl
- SecurityWeakPointNotificationControl
- InformationSecurityOrganizationControl
 - InformationSecurityInternalOrganizationControl
 - InformationSecurityContactWithAuthoritiesControl
 - InformationSecurityContactWithInterestGroupsControl
 - InformationSecurityInProjectManagementControl
 - InformationSecurityRolesandResponsibilitiesControl
 - InformationSecurityTaskSegregationControl
 - InformationSecurityMobileUsersControl
 - InformationSecurityHomeworkingControl
 - InformationSecurityMobileDevicesPolicyControl
- InformationSecurityPoliciesControl
 - InformationSecurityManagementGuidelines
 - InformationSecurityPolicy
 - InformationSecurityPolicyRevision
- IT_Acquisition_O&M_Control
 - DevOpsControl
 - ApplicationTestAfterOSChangesControl
 - OutsourcedSWDevelopmentControl
 - SecureCodingIntegrationIntoIDEControl
 - SecureCodingPolicyControl
 - SecureEngineeringDesignControl
 - SecurityFunctionalTestControl
 - SWPackagesChangingRestrictionsControl
 - SystemChangeManagementProcedureControl
 - SystemsAcceptanceTestControl
 - ITSecurityRequirementControl
 - ApplicationServicesSecurityInPublicNetworksControl

- ITSecurityRequirementandSpecificationAnalysisControl
 - TransactionProtectionInApplicationServicesControl
- TestDataControl
 - TestDataProtectionControl
- NetworkSecurityControl
 - InformationSharingControl
 - eMessagingControl
 - InformationSharingAgreementsControl
 - InformationSharingProceduresandPoliciesControl
 - NonDisclosureAgreementControl
 - NetworkSecurityManagementControl
 - NetworkControl
 - NetworkSegmentationControl
 - NetworkServicesSecurityControl
- OperationSecurityControl
 - BackupControl
 - ITAuditControl
 - ITAuditingControl
 - MaliciousSWProtectionControl
 - ControlsAgainstMaliciousSW
 - OperationProceduresandResponsibilityControl
 - CapabilityManagementControl
 - ChangeManagementControl
 - DevTestandOperationResourcesSegmentationControl
 - OperationProceduresDocumentationControl
 - RegisterandSupervisionControl
 - AdminandOperationLogControl
 - ClockSynchronizationControl
 - EventLogControl
 - LogInformationProtectionControl
 - SWControl
 - SWInstallationControl
 - VulnerabilityManagementControl
 - SWInstallationRestrictionControl
 - TechnicalVulnerabilitiesControl
- PhysicalSecurityControl

- DeviceSecurityControl
 - CleanScreenandWorkspacePolicyControl
 - CompanyOwnedDiscardedMaterialControl
 - EquipmentMaintenanceAreaControl
 - EquipmentSecurityOutsideCompanyPremisesControl
 - LocationandEquipmentProtectionControl
 - ReuseOrEliminationOfEquipmentControl
 - SupplyFacilitiesControl
 - UnattendedUserEquipmentControl
 - WiringSecurityControl
 - SecuredAreasControl
 - LoadingandUnloadingAreasControl
 - PhysicalEntranceControl
 - PhysicalPerimeterSecurityControl
 - ProtectionAgainstExternalandEnvironmentalThreatsControl
 - ResourcesandOfficesSecurityControl
 - SecureWorkSpaceAreaControl
- ProvidersRelationshipControl
 - ProvidersRelationshipSecurityControl
 - ICTSupplyChainControl
 - InformationSecurityPolicyForSupplyChainControl
 - ThirdPartiesSecurityRequirementsForContractsControl
 - ProvidersSupplyControl
 - ThirdPartyProvidedServicesAuditControl
 - ThirdPartyProvidedServicesChangeManagementControl
- Countermeasures
 - Firecountermeasure
 - FireExtinguisher
 - CO2FireExtinguisher
 - HandFireExtinguisher
 - WaterFireExtinguisher
- Detector
 - FireDetector
- FaultForecasting
 - OrdinalEvaluation

- ProbabilisticEvaluation
 - Modeling
 - OperationalTesting
- FaultPrevention
- FaultRemoval
 - Correction
 - Diagnosis
 - NonRegressionVerification
 - Verification
 - DynamicVerification
 - StaticVerification
- FaultTolerance
 - ErrorDetection
 - ConcurrentDetection
 - PreemptiveDetection
 - Recovery
 - ErrorHandling
 - FaultHandling

C.V. SECURITY EVENTS

"Security events (alerts) produced by detections of our semantic Cyber Threat Intelligence framework".

Every instance of the class is characterized by the following properties:

- firstSeen only xsd:dateTimeStamp
- lastSeen some xsd:dateTimeStamp
- type only xsd:string
- type some {"security-event"}

Table 74 Properties of Security Events class (DRM.owl)

Property Name	Type	Description
firstSeen (required)	dateTimeStamp	TimeStamp of the first time it was seen.
type (required)	string	Type = {"security-event"^^xsd:string}.
lastSeen (optional)	dateTimeStamp	TimeStamp of the last time it was seen.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- (dstRef some DomainName) or (dstRef some IPv4Addr) or (dstRef some IPv6Addr) or (dstRef some URL)

- (srcRef some IPv4Addr) or (srcRef some IPv6Addr)
- createdByRef some Identity
- relatedTo some NetworkTraffic

It has no subclasses.

C.VI. INCIDENT

"Reserved object" by STIX™ v2, we also reserve it at DRM. SecurityEvents will be defined within the DRM ontology, at least to work with events (which are potential incidents). This class has no further development on purpose, as it is reserved.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)

This class has no subclasses.

C.VII. RISK MANAGEMENT

"Risk Management strategies".

Every instance of the class is characterized by the following properties:

- actualControl only xsd:string
- actualRisk only xsd:float
- description some xsd:string
- impact only xsd:float
- name only xsd:string
- potentialRisk only xsd:float
- probability only xsd:float
- residualRisk only xsd:float
- type only xsd:string
- value only xsd:float

Table 75 Properties of Risk Management class (DRM.owl)

Property Name	Type	Description
actualControl (required)	string	Actual control (e.g. ISO 27002) associated to the risk management strategy.
actualRisk (required)	float	Actual residual risk value (considering safeguards).
impact (required)	float	Impact value of the risk.
name (required)	string	Name of the risk management strategy.
potentialRisk (required)	float	Potential risk value (without safeguards).
probability (required)	float	Probability value of the risk.

residualRisk (required)	float	Residual risk value of the risk.
description (optional)	string	Specifies the description of the risk management strategy.
type (optional)	string	It has to include type = {"risk-management"^^xsd:string}.
value (optional)	float	(Reserved)

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- manages some Risk

It has multiple subclasses:

- RiskInvestigation
- RiskMitigation
- RiskMonitoring
- RiskResponse

C.VIII. THREAT

"A threat is anything (manmade or act of nature) that has the potential to cause harm (e.g. exploits a vulnerability)"

Every instance of the class is characterized by the following properties:

- count only xsd:integer
- description some xsd:string
- firstSeen only xsd:dateTimeStamp
- killChainPhases min 1 xsd:string
- lastSeen only xsd:dateTimeStamp
- name only xsd:string
- primaryMotivation some xsd:string
- secondaryMotivations some xsd:string
- type only xsd:string
- type some {"threat"}

Table 76 Properties of Threat class (DRM.owl)

Property Name	Type	Description
count (required)	integer	Counting of the threat (reserved).
firstSeen (required)	dateTimeStamp	TimeStamp of the first time it was seen.

killChainPhases (required)	string	Killchain related phase/s (>= 1).
lastSeen (required)	dateTimeStamp	TimeStamp of the last time it was seen.
name (required)	string	Name of the threat.
type (required)	string	Type = {"threat"^^xsd:string}.
description (optional)	string	Specifies the description of the threat.
primaryMotivation (optional)	string	Main or primaty motivation of the threat.
secondaryMotivation (optional)	string	Secondary motivation of the threat.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- exploits some Vulnerability
- isInitiatedBy only ThreatActor
- threatens some Asset

It has multiple subclasses:

- ComplianceThreat
 - DataProtectionRisks
 - OtherLegalObligationRisks
- DeliberatedThreat
 - DeliberatedConfigFilesTampering
 - DeliberatedHWTampering
 - DeliberatedInformationDestruction
 - DeliberatedInformationLeak
 - DeliberatedInformationTampering
 - DeliberatedMaliciousSWDistribution
 - DeliberatedRegistersTampering
 - DeliberatedSWTampering
 - DeliberatedUnauthorizedAccess
 - DeviceTheft
- HumanThreat
 - InsiderThreats
- ICTThreat
 - ConfigurationError

- DeliberatedConfigFilesTampering
- DeliberatedHWTampering
- DeliberatedInformationDestruction
- DeliberatedInformationLeak
- DeliberatedInformationTampering
- DeliberatedMaliciousSWDistribution
- DeliberatedRegistersTampering
- DeliberatedSWTampering
- DeliberatedUnauthorizedAccess
- DenialOfService
- DeviceLost
- DeviceTheft
- HumanResourcesNotAvailable
- HWMaintenanceError
- IdentityThief
- MonitoringError
- NetworkOutage
- NonIntentionalAdminError
- NonIntentionalInformationDestruction
- NonIntentionalInformationLeak
- NonIntentionalInformationTampering
- NonIntentionalMaliciousSWDistribution
- NonIntentionalUserError
- PhysicalFault
- PowerOutage
- PrivilegeEscalation
- SocialEngineering
- SWMaintenanceError
- SWVulnerabilities
- UnexpectedUsage
- PhysicalThreat
 - Accidents
 - Fire
 - NaturalDisasters
 - Terrorism
- Quality&ProcessThreat
 - CorporateBrandImageDamages

- DelayedDelivery
- StakeholdersSatisfaction
- TechnicalComplexity
- Untrustworthy
- UsersComplaints
- StrategicThreat
 - BadReputationThreat
 - EconomicLoss
 - PartnershipsIssues
 - StakeholdersIssues
 - StrategicPlanOnRisk
- UnintentionalThreat
 - ConfigurationError
 - DeviceLost
 - HWMaintenanceError
 - NonIntentionalAdminError
 - NonIntentionalInformationDestruction
 - NonIntentionalInformationLeak
 - NonIntentionalInformationTampering
 - NonIntentionalMaliciousSWDistribution
 - NonIntentionalUserError
 - PowerOutage

C.IX. RISK SEVERITY

"Risk Severity:

Cumulative Risk classification by Severity as a mix of Impact and Probability."

Every instance of the class is characterized by the following properties:

- count only xsd:integer
- type only xsd:string
- value only xsd:float

Table 77 Properties of Risk severity class (DRM.owl)

Property Name	Type	Description
type (required)	string	It has to include type = {"risk-severity"^^xsd:string}.
count (required)	integer	Specifies the counting of the scope (reserved).

value (required)	float	Specifies the value of the scope.
-------------------------	-------	-----------------------------------

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk

It has the following subclasses:

- ExtremeRiskSeverity
- HighRiskSeverity
- LowRiskSeverity
- MediumRiskSeverity

C.X. RISK SCOPE

"It represents the scope of the Dynamic Risk Assessment (DRA) and Dynamic Risks Management (DRM), that is to say, the different "services" or "projects" of an organization that are included in the scope of their risk assessment and risk management model."

Every instance of the class is characterized by the following properties:

- description some xsd:string
- name only xsd:string
- type only xsd:string
- type some {"scope"}

Table 78 Properties of Risk scope class (DRM.owl)

Property Name	Type	Description
type (required)	string	It has to include type = {"scope"^^xsd:string}.
name (required)	string	Specifies the name of the scope.
Description (optional)	string	Specifies the description of the scope.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- dependsOn some Data
- hasContext some Context

It has no subclasses.

C.XI. RISK

"Risks of our Organization"

Every instance of the class is characterized by the following properties:

- description some xsd:string
- name only xsd:string
- type only xsd:string

Table 79 Properties of Risk class (DRM.owl)

Property Name	Type	Description
type (required)	string	It has to include type = {"context"^^xsd:string}.
name (required)	string	Specifies the name of the context.
Description (optional)	string	Specifies the description of the context.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- (isGeneratedBy min 1 Threat) or (isGeneratedBy min 1 Vulnerability)
- belongToRefs min 1 RiskOwner
- createdByRef some Identity
- damages min 1 Asset
- hasAssessmentOf only RiskAssessment
- hasImpactOf only RiskImpact
- hasProbabilityOf only RiskProbability
- hasSeverityOf only RiskSeverity
- isGeneratedBy min 1 Vulnerability
- threatens only RiskScope

It has multiple subclasses (also related to ISO31000):

- ComplianceRisk
 - DataProtectionComplianceRisk
 - OtherLegalComplianceRisk
- InformationSecurityRisk
 - ConfigurationErrorRisk
 - DeliberatedConfigFilesTamperingRisk
 - DeliberatedHWTamperingRisk
 - DeliberatedInformationDestructionRisk
 - DeliberatedInformationLeakRisk
 - DeliberatedInformationTamperingRisk
 - DeliberatedMaliciousSWDistributionRisk
 - DeliberatedRegistersTamperingRisk
 - DeliberatedSWTamperingRisk

- DeliberatedUnauthorizedAccessRisk
- DenialOfServiceRisk
- DeviceLostRisk
- DeviceTheftRisk
- HumanResourcesNotAvailableRisk
- HWMaintenanceErrorRisk
- IdentityThiefRisk
- LogicalFailureRisk
- MonitoringErrorRisk
- NetworkOutageRisk
- NonIntentionalAdminErrorRisk
- NonIntentionalInformationDestructionRisk
- NonIntentionalInformationLeakRisk
- NonIntentionalInformationTamperingRisk
- NonIntentionalMaliciousSWDistributionRisk
- NonIntentionalUserErrorRisk
- PhysicalFailureRisk
- PowerOutageRisk
- SocialEngineeringRisk
- SWMaintenanceErrorRisk
- SWVulnerabilitiesRisk
- PhysicalRisk
 - AccidentRisk
 - FireRisk
 - FloodingRisk
 - NaturalDisasterRisk
 - TerrorismAttackRisk
- Quality&ProcessRisk
 - CorporateBrandImageDamageRisk
 - DelayedDeliveryRisk
 - StakeholdersSatisfactionRisk
 - TechnicalComplexityDerivedRisk
 - UntrustworthyRisk
 - UsersComplaintsRisk
- StrategicRisk
 - BadReputationRisk
 - EconomicLossRisk

- PartnershipRisk
- PressNegativeImpactRisk
- StakeholdersRisk
- StrategicObjectiveRisk
- OtherRisk

C.XII. RISK IMPACT

"Risk Impact: individual Risk Classification by Impact severity"

Every instance of the class is characterized by the following properties:

- count only xsd:integer
- impact only xsd:float
- type only xsd:string
- type only {"risk-impact"}
- value only xsd:float

Table 80 Properties of Risk Impact class (DRM.owl)

Property Name	Type	Description
count (required)	integer	Number of re-assessments of the impact of the risk (reserved).
impact (required)	float	Impact assessment of the risk.
type (required)	string	It has to include type = {"risk-impact"}^xsd:string.
value (required)	float	Specifies the value of the risk impact.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk

It has multiple subclasses:

- CriticalImpactRisk
- HighImpactRisk
- LowImpactRisk
- MediumImpactRisk
- VeryLowImpactRisk

C.XIII. RISK OWNER

"Owner of all risks associated with a specific RiskScope (service, project, tool, etc.)".

Every instance of the class is characterized by the following properties:

- contactInformation only xsd:string

- name only xsd:string
- type only xsd:string
- type only {"risk-owner"}

Table 81 Properties of Risk Owner class (DRM.owl)

Property Name	Type	Description
contactInformation (required)	string	Contact details of the Risk Owner.
name (required)	string	Name of the risk owner.
type (required)	string	It has to include type = {"risk-owner"^^xsd:string}.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates min 1 RiskAssessment
- isResponsibleOf min 1 Risk
- isResponsibleOf min 1 RiskScope

It has no subclasses.

C.XIV. RISK PROBABILITY

"Risk Probability: individual Risk Classification by probability"

Every instance of the class is characterized by the following properties:

- count only xsd:integer
- probability only xsd:float
- type only xsd:string
- type only {"risk-probability"}
- value only xsd:float

Table 82 Properties of Risk Probability class (DRM.owl)

Property Name	Type	Description
count (required)	integer	Number of re-assessments of the probability of the risk (reserved).
probability (required)	float	Probability assessment of the risk.
type (required)	string	It has to include type = {"risk-probability"^^xsd:string}.
value (required)	float	Specifies the value of the risk probability.

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity

It has multiple subclasses:

- ExtremeRiskProbability
- HighRiskProbability
- LowRiskProbability
- MediumRiskProbability
- VeryLowRiskProbability

C.XV. RISK ASSESSMENT

"Risk Assessment:

- Potential Risk (w/o safeguards)
- Residual Risk (considering safeguards)"

Every instance of the class is characterized by the following properties:

- actualControl only xsd:string
- actualRisk only xsd:float
- count only xsd:integer
- impact only xsd:float
- potentialRisk only xsd:float
- probability only xsd:float
- type only xsd:string
- value only xsd:float

Table 83 Properties of Risk Assessment class (DRM.owl)

Property Name	Type	Description
actualControl (required)	string	Control (e.g. ISO 27002 related) to the risk.
actualRisk (required)	float	Actual Residual risk assessment (considering safeguards).
count (required)	integer	Number of re-assessments of the risk.
impact (required)	float	Impact assessment of the risk.
potentialRisk (required)	float	Potential risk assessment (without safeguards).
probability (required)	float	Probability assessment of the risk.
value (required)	float	Specifies the value of the risk assessment (reserved).

type (required)	string	It has to include type = {"risk-assessment"^^xsd:string}.
---------------------------	--------	---

This class has the following relationships with other classes:

- owl:Thing or (objectRefs some MarkingDefinition)
- createdByRef some Identity
- evaluates only Risk
- isIncreasedBy some SecurityEvents
- isMitigatedBy some Safeguards
- threatens only RiskScope

It has multiple subclasses:

- PotentialRisk
- PotentialTotalRisk
- ResidualRisk
- ResidualTotalRisk

ANNEX D. SWRL REASONING RULES DETAIL

D.I. Asset Valuation Rules

The reasoner will infer new knowledge about the value of our assets, if applying the following SWRL rules (examples) over an updated asset inventory.

The rules will implement the following schema of inheritance due to the usual Risk management dependency tree within the assets.

Figure 97 Asset dependency tree for asset valuation of our organization

D.I.I. Asset Valuation#1 Data Inherit Valuation

Table 84 AssetValuation#1 Data Inherit Valuation

AssetValuation#1 Data Inherit Valuation
<p>The reasoner will infer the value of the Data, as an automatic asset valuation, due to its dependencies. As it is in the top of the dependency tree, it takes the valuation of the risk scope, done by the Risk Owner:</p> <p>(drm:AssetValuation(?av) ^ drm:evaluates(?av, ?rs)^ drm:RiskScope(?rs)).</p>
<p>drm:recoveryPointObjective(?av, ?rpo) ^ drm:recoveryTimeObjective(?av, ?rto) ^ drm:Data(?data) ^ drm:dependsOn(?rs, ?data) ^ drm:integrity(?av, ?i) ^ drm:confidentiality(?av, ?c) ^ drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:availability(?av, ?a) ^ drm:RiskScope(?rs) ^ drm:authenticity(?av, ?au) ^ swrlx:makeOWLThing(?x, ?av, ?data) ^ drm:accounting(?av, ?ac) -> drm:recoveryPointObjective(?x, ?rpo) ^ drm:confidentiality(?x, ?c) ^ drm:integrity(?x, ?i) ^ drm:accounting(?x, ?ac) ^ drm:authenticity(?x, ?au) ^ drm:evaluates(?x, ?data) ^ drm:availability(?x, ?a) ^ drm:AssetValuation(?x) ^ drm:recoveryTimeObjective(?x, ?rto)</p>

D.I.II. Asset Valuation#2 Sw Inherit Valuation

Table 85 AssetValuation#2 SW Inherit Valuation

AssetValuation#2 SW Inherit Valuation
The reasoner will infer the value of the SW, as an automatic asset valuation, due to its dependency tree.
$\begin{aligned} & \text{stix2:Software(?sw)} \wedge \text{drm:Data(?data)} \wedge \text{drm:dependsOn(?rs, ?data)} \wedge \text{drm:integrity(?data, ?i)} \\ & \wedge \text{drm:dependsOn(?data, ?sw)} \wedge \text{drm:confidentiality(?data, ?c)} \wedge \text{drm:availability(?data, ?a)} \wedge \\ & \text{drm:evaluates(?av, ?rs)} \wedge \text{swrlx:makeOWLThing(?x, ?av)} \wedge \text{drm:recoveryPointObjective(?data, ?rpo)} \\ & \wedge \text{drm:AssetValuation(?av)} \wedge \text{drm:RiskScope(?rs)} \wedge \text{drm:accounting(?data, ?ac)} \wedge \\ & \text{drm:authenticity(?data, ?au)} \wedge \text{drm:recoveryTimeObjective(?data, ?rto)} \rightarrow \\ & \text{drm:recoveryPointObjective(?x, ?rpo)} \wedge \text{drm:confidentiality(?x, ?c)} \wedge \text{drm:evaluates(?x, ?sw)} \wedge \\ & \text{drm:integrity(?x, ?i)} \wedge \text{drm:accounting(?x, ?ac)} \wedge \text{drm:authenticity(?x, ?au)} \wedge \text{drm:availability(?x, ?a)} \\ & \wedge \text{drm:AssetValuation(?x)} \wedge \text{drm:recoveryTimeObjective(?x, ?rto)} \end{aligned}$

D.I.III. Asset Valuation#3 Hw Inherit Valuation

Table 86 AssetValuation#3 HW Inherit Valuation

AssetValuation#3 HW Inherit Valuation
The reasoner will infer the value of the HW, as an automatic asset valuation, due to its dependency tree.
$\begin{aligned} & \text{drm:Data(?data)} \wedge \text{drm:dependsOn(?rs, ?data)} \wedge \text{drm:integrity(?data, ?i)} \wedge \\ & \text{drm:confidentiality(?data, ?c)} \wedge \text{drm:Hardware(?hw)} \wedge \text{drm:availability(?data, ?a)} \wedge \\ & \text{drm:evaluates(?av, ?rs)} \wedge \text{swrlx:makeOWLThing(?x, ?av)} \wedge \text{drm:recoveryPointObjective(?data, ?rpo)} \\ & \wedge \text{drm:AssetValuation(?av)} \wedge \text{drm:RiskScope(?rs)} \wedge \text{drm:accounting(?data, ?ac)} \wedge \\ & \text{drm:authenticity(?data, ?au)} \wedge \text{drm:recoveryTimeObjective(?data, ?rto)} \wedge \\ & \text{drm:dependsOn(?data, ?hw)} \rightarrow \text{drm:recoveryPointObjective(?x, ?rpo)} \wedge \text{drm:confidentiality(?x, ?c)} \\ & \wedge \text{drm:integrity(?x, ?i)} \wedge \text{drm:accounting(?x, ?ac)} \wedge \text{drm:authenticity(?x, ?au)} \wedge \\ & \text{drm:availability(?x, ?a)} \wedge \text{drm:AssetValuation(?x)} \wedge \text{drm:recoveryTimeObjective(?x, ?rto)} \wedge \\ & \text{drm:evaluates(?x, ?hw)} \end{aligned}$

D.I.IV. Asset Valuation#4 Users Inherit Valuation

Table 87 AssetValuation#4 Users Inherit Valuation

AssetValuation#4 Users Inherit Valuation
The reasoner will infer the value of the Users, as an automatic asset valuation, due to its dependency tree.
$\begin{aligned} & \text{drm:Users(?u)} \wedge \text{drm:availability(?hw, ?a)} \wedge \text{drm:Data(?data)} \wedge \text{drm:dependsOn(?rs, ?data)} \wedge \\ & \text{drm:Hardware(?hw)} \wedge \text{drm:confidentiality(?hw, ?c)} \wedge \text{drm:integrity(?hw, ?i)} \wedge \text{drm:evaluates(?av, ?rs)} \\ & \wedge \text{swrlx:makeOWLThing(?x, ?av)} \wedge \text{drm:dependsOn(?hw, ?u)} \wedge \text{drm:AssetValuation(?av)} \wedge \\ & \text{drm:recoveryTimeObjective(?hw, ?rto)} \wedge \text{drm:RiskScope(?rs)} \wedge \\ & \text{drm:recoveryPointObjective(?hw, ?rpo)} \wedge \text{drm:accounting(?hw, ?ac)} \wedge \text{drm:dependsOn(?data, ?hw)} \\ & \wedge \text{drm:authenticity(?hw, ?au)} \rightarrow \text{drm:recoveryPointObjective(?x, ?rpo)} \wedge \\ & \text{drm:confidentiality(?x, ?c)} \wedge \text{drm:integrity(?x, ?i)} \wedge \text{drm:accounting(?x, ?ac)} \wedge \text{drm:evaluates(?x, ?u)} \\ & \wedge \text{drm:authenticity(?x, ?au)} \wedge \text{drm:availability(?x, ?a)} \wedge \text{drm:AssetValuation(?x)} \wedge \\ & \text{drm:recoveryTimeObjective(?x, ?rto)} \end{aligned}$

D.I.V. Asset Valuation#5 Networks Inherit Valuation

Table 88 AssetValuation#5 Networks Inherit Valuation

AssetValuation#5 Networks Inherit Valuation
The reasoner will infer the value of the Networks, as an automatic asset valuation, due to its dependency tree.
$\begin{aligned} & \text{drm:availability(?hw, ?a)} \wedge \text{drm:Data(?data)} \wedge \text{drm:Networks(?net)} \wedge \text{drm:dependsOn(?rs, ?data)} \wedge \text{drm:Hardware(?hw)} \wedge \text{drm:confidentiality(?hw, ?c)} \wedge \text{drm:integrity(?hw, ?i)} \wedge \\ & \text{drm:evaluates(?av, ?rs)} \wedge \text{drm:AssetValuation(?av)} \wedge \text{drm:recoveryTimeObjective(?hw, ?rto)} \wedge \\ & \text{swrlx:makeOWLThing(?x, ?net)} \wedge \text{drm:RiskScope(?rs)} \wedge \text{drm:recoveryPointObjective(?hw, ?rpo)} \wedge \text{drm:dependsOn(?hw, ?net)} \wedge \text{drm:accounting(?hw, ?ac)} \wedge \text{drm:dependsOn(?data, ?hw)} \\ & \wedge \text{drm:authenticity(?hw, ?au)} \rightarrow \text{drm:recoveryPointObjective(?x, ?rpo)} \wedge \text{drm:evaluates(?x, ?net)} \wedge \text{drm:confidentiality(?x, ?c)} \wedge \text{drm:integrity(?x, ?i)} \wedge \text{drm:accounting(?x, ?ac)} \wedge \\ & \text{drm:authenticity(?x, ?au)} \wedge \text{drm:availability(?x, ?a)} \wedge \text{drm:AssetValuation(?x)} \wedge \\ & \text{drm:recoveryTimeObjective(?x, ?rto)} \end{aligned}$

D.I.VI. Asset Valuation#6 Real Estate Inherit Valuation

Table 89 AssetValuation#6 Real Estate Inherit Valuation

AssetValuation#6 Real Estate Inherit Valuation
The reasoner will infer the value of the Real Estate, as an automatic asset valuation, due to its dependency tree.
$\begin{aligned} & \text{drm:availability(?hw, ?a)} \wedge \text{drm:Data(?data)} \wedge \text{drm:dependsOn(?rs, ?data)} \wedge \\ & \text{drm:Hardware(?hw)} \wedge \text{drm:confidentiality(?hw, ?c)} \wedge \text{drm:integrity(?hw, ?i)} \wedge \text{drm:evaluates(?av, ?rs)} \wedge \\ & \text{swrlx:makeOWLThing(?x, ?av)} \wedge \text{drm:AssetValuation(?av)} \wedge \text{drm:recoveryTimeObjective(?hw, ?rto)} \wedge \text{drm:RiskScope(?rs)} \wedge \\ & \text{drm:recoveryPointObjective(?hw, ?rpo)} \wedge \text{drm:RealEstate(?re)} \wedge \text{drm:accounting(?hw, ?ac)} \wedge \\ & \text{drm:dependsOn(?data, ?hw)} \wedge \text{drm:authenticity(?hw, ?au)} \wedge \text{drm:dependsOn(?hw, ?re)} \rightarrow \\ & \text{drm:recoveryPointObjective(?x, ?rpo)} \wedge \text{drm:evaluates(?x, ?re)} \wedge \text{drm:confidentiality(?x, ?c)} \wedge \\ & \text{drm:integrity(?x, ?i)} \wedge \text{drm:accounting(?x, ?ac)} \wedge \text{drm:authenticity(?x, ?au)} \wedge \text{drm:availability(?x, ?a)} \wedge \\ & \text{drm:AssetValuation(?x)} \wedge \text{drm:recoveryTimeObjective(?x, ?rto)} \end{aligned}$

D.II. Threat Inventory Rules

The reasoner will infer new knowledge with regard to any potential Threat that has to be considered. If applying the following SWRL rules (examples) over our data-set, the reasoner will create instances of different type of threats that are usually very difficult or complex, to be manually identified by the risk owners. It will create an automatic inventory of threats at different levels (strategic, tactic and technical), to contribute to the whole process. It will be an input of the Risk Inventory.

D.II.I. Threat Inventory#1 Auto Bad Reputation Threat In Classified Data

Table 90 ThreatInventory#1 Auto Bad Reputation Threat in Classified Data

ThreatInventory#1 Auto Bad Reputation Threat in Classified Data
The reasoner will infer new knowledge, in this case, it will create Bad Reputation threats associated to our Classified Data. It will consider that any data leak (in this case, especially associated to Classified data) will be a Bad Reputation Threat to our organization.


```

drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:ClassifiedData(?data) ^
drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?x, ?data) ^ drm:dependsOn(?rs, ?data)
->
  drm:impact(?x, "4.0"^^xsd:float) ^ drm:probability(?x, "2.0"^^xsd:float) ^ stix2:type(?x, "Threat
  Bad Reputation"^^rdf:PlainLiteral) ^ drm:threatens(?x, ?data) ^ drm:BadReputationThreat(?x)

```

D.II.II. Threat Inventory#2 Auto Data Protection Threat

Table 91 ThreatInventory#2 Auto Data Protection Threat

ThreatInventory#2 Auto Data Protection Threat
<p>The reasoner will infer new knowledge, in this case, it will create “Threats against Data Protection” associated to our Classified Data. It will consider that any non-authorized access to the data (in this case, associated to Classified data) will be a threat to the compliance of the Data Protection Policy.</p>
<pre> drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:ClassifiedData(?data) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?x, ?data) ^ drm:dependsOn(?rs, ?data) -> stix2:type(?x, "Threat against Data Protection"^^rdf:PlainLiteral) ^ drm:impact(?x, "4.0"^^xsd:float) ^ drm:DataProtectionRisks(?x) ^ drm:probability(?x, "2.0"^^xsd:float) ^ drm:threatens(?x, ?data) </pre>

D.II.III. Threat Inventory#3 Auto Unintentional User Error Threat

Table 92 ThreatInventory#3 Auto Unintentional User Error Threat

ThreatInventory#3 Auto Unintentional User Error Threat
<p>The reasoner will infer new knowledge, in this case, it will create “Threats of Unintentional User Error” associated to users with cybersecurity experience <=3. It will understand that unintentional user errors will be associated, especially, to this type of non-experienced users.</p>
<pre> :evaluates(?av, ?rs) ^ drm:Users(?u) ^ swrlx:makeOWLThing(?x, ?u) ^ swrlb:lessThanOrEqual(?e, 3) ^ drm:dependsOn(?hw, ?u) ^ drm:AssetValuation(?av) ^ drm:Data(?data) ^ drm:hasCybersecurityExperience(?u, ?e) ^ drm:RiskScope(?rs) ^ drm:dependsOn(?rs, ?data) ^ drm:Hardware(?hw) ^ drm:dependsOn(?data, ?hw) -> stix2:type(?x, "Threat Unintentional User Error"^^rdf:PlainLiteral) ^ drm:NonIntentionalUserError(?x) ^ drm:threatens(?x, ?u) ^ drm:impact(?x, "3.0"^^xsd:float) ^ drm:probability(?x, "3.0"^^xsd:float) </pre>

D.II.IV. Threat Inventory#4 Auto Sw Vulnerabilities Threat

Table 93 ThreatInventory#4 Auto SW Vulnerabilities Threat

ThreatInventory#4 Auto SW Vulnerabilities Threat
<p>The reasoner will infer new knowledge, in this case, it will create “Threats of SW Vulnerabilities” associated to the use of SW.</p>

```

drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:Software(?sw) ^ drm:Data(?data) ^
drm:RiskScope(?rs) ^ drm:dependsOn(?rs, ?data) ^ drm:dependsOn(?data, ?sw) ^
swrlx:makeOWLThing(?x, ?sw)
->
drm:impact(?x, "2.0"^^xsd:float) ^ drm:threatens(?x, ?sw) ^ drm:probability(?x,
"4.0"^^xsd:float) ^ stix2:type(?x, "Threat SW Vulnerabilities"^^rdf:PlainLiteral) ^
drm:SWVulnerabilities(?x)

```

D.II.V. Threat Inventory#5 Auto Deliberated Malicious Sw Distribution Threat

Table 94 ThreatInventory#5 Auto Deliberated Malicious SW Distribution Threat

ThreatInventory#5 Auto Deliberated Malicious SW Distribution Threat

The reasoner will infer new knowledge, in this case, it will create Deliberated Malicious SW Distribution threats associated to users with low cybersecurity experience (≤ 3) and their use of SW to handle specific Data.

```

drm:evaluates(?av, ?rs) ^ drm:Users(?u) ^ swrlb:lessThanOrEqual(?e, 3) ^
drm:AssetValuation(?av) ^ drm:Software(?sw) ^ drm:Data(?data) ^
drm:hasCybersecurityExperience(?u, ?e) ^ drm:RiskScope(?rs) ^ drm:dependsOn(?data, ?u)
^ drm:dependsOn(?rs, ?data) ^ drm:dependsOn(?data, ?sw) ^ swrlx:makeOWLThing(?x, ?sw)
->
drm:DeliberatedMaliciousSWDistribution(?x) ^ drm:threatens(?x, ?sw) ^ drm:probability(?x,
"4.0"^^xsd:float) ^ drm:impact(?x, "3.0"^^xsd:float) ^ stix2:type(?x, "Threat Deliberated
Malicious SW Distribution"^^rdf:PlainLiteral)

```

Name
ThreatInventory#5 Auto Deliberated Malicious SW Distribution Threat
Comment
Status
Ok
<pre> drm:evaluates(?av, ?rs) ^ drm:Users(?u) ^ swrlb:lessThanOrEqual(?e, 3) ^ drm:AssetValuation(?av) ^ drm:Software(?sw) ^ drm:Data(?data) ^ drm:hasCybersecurityExperience(?u, ?e) ^ drm:RiskScope(?rs) ^ drm:dependsOn(?data, ?u) ^ drm:dependsOn(?rs, ?data) ^ drm:dependsOn(?data, ?sw) ^ swrlx:makeOWLThing(?x, ?sw) -> drm:DeliberatedMaliciousSWDistribution(?x) ^ drm:threatens(?x, ?sw) ^ drm:probability(?x, "4.0"^^xsd:float) ^ drm:impact(?x, "3.0"^^xsd:float) ^ stix2:type(?x, "Threat Deliberated Malicious SW Distribution") </pre>

Figure 98 SWRL rule to make automatic inventory of deliberate malicious SW distribution threats when a service in scope of our risk management framework depends on data and later on SW (OS, Browser, AdobeFlashPlugin) but also on HW (Personal Computers)

D.II.VI. Threat Inventory#6 Auto Social Engineering Threat

Table 95 ThreatInventory#6 Auto Social Engineering Threat

ThreatInventory#6 Auto Social Engineering Threat

The reasoner will infer new knowledge, in this case, it will create Social Engineering threats associated to users with low cybersecurity experience (≤ 3) which, at the same time, they have access to Classified Data.
<pre> drm:evaluates(?av, ?rs) ^ drm:Users(?u) ^ swrlx:makeOWLThing(?x, ?u) ^ swrlb:lessThanOrEqual(?e, 3) ^ drm:AssetValuation(?av) ^ drm:ClassifiedData(?data) ^ drm:hasCybersecurityExperience(?u, ?e) ^ drm:RiskScope(?rs) ^ drm:dependsOn(?data, ?u) ^ drm:dependsOn(?rs, ?data) -> drm:SocialEngineering(?x) ^ drm:threatens(?x, ?u) ^ drm:impact(?x, "4.0"^^xsd:float) ^ stix2:type(?x, "Threat Social Engineering"^^rdf:PlainLiteral) ^ drm:probability(?x, "3.0"^^xsd:float) </pre>

D.II.VII. Threat Inventory#7 Auto Corporate Brand Image Damage Threat

Table 96 ThreatInventory#7 Auto Corporate Brand Image Damage Threat

ThreatInventory#7 Auto Corporate Brand Image Damage Threat
The reasoner will infer new knowledge, in this case, it will create Brand Image threats associated to our Classified Data. It will consider that any data leak (in this case, especially associated to Classified data) will be a Brand Image Threat to our organization.
<pre> drm:evaluates(?av, ?rs) ^ drm:AssetValuation(?av) ^ drm:ClassifiedData(?data) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?x, ?data) ^ drm:dependsOn(?rs, ?data) -> stix2:type(?x, "Threat for Corporate Brand Image"^^rdf:PlainLiteral) ^ drm:impact(?x, "4.0"^^xsd:float) ^ drm:CorporateBrandImageDamages(?x) ^ drm:probability(?x, "1.0"^^xsd:float) ^ drm:threatens(?x, ?data) </pre>

D.III. Risk Inventory Rules

The reasoner will infer new knowledge with regard to any potential Risk that has to be considered. If applying the following SWRL rules (examples) over our data-set, the reasoner will create instances of different type of risks that are usually very difficult or complex, to be manually identified by the risk owners. It will create an automatic inventory of risks at different levels (strategic, tactic and technical), to contribute to the whole process.

D.III.I. Risk Inventory#1 Auto Bad Reputation Risk

Table 97 RiskInventory#1 Auto Bad Reputation Risk

RiskInventory#1 Auto Bad Reputation Risk
The reasoner will infer new classes of Bad Reputation Risk, associated to the "Bad Reputation Threats" and it will handle inheritance along the dependency tree.
<pre> drm:dependsOn(?rs, ?a) ^ drm:threatens(?x, ?a) ^ drm:impact(?x, ?i) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?r, ?x) ^ drm:probability(?x, ?p) ^ drm:BadReputationThreat(?x) -> stix2:type(?r, "Bad Reputation Risk"^^rdf:PlainLiteral) ^ drm:isGeneratedBy(?r, ?x) ^ drm:threatens(?r, ?rs) ^ drm:BadReputationRisk(?r) ^ drm:damages(?r, ?a) </pre>

D.III.II. Risk Inventory#2 Auto Deliberated Malicious SW Distribution Risk

Table 98 RiskInventory#2 Auto Deliberated Malicious SW Distribution Risk

RiskInventory#2 Auto Deliberated Malicious SW Distribution Risk	
The reasoner will infer new classes of Deliberated Malicious SW Distribution Risk, associated to all the "Deliberated Malicious SW Distribution" Threats, and it will handle inheritance along the dependency tree.	
<pre>drm:dependsOn(?rs, ?a) ^ drm:threatens(?x, ?a) ^ drm:impact(?x, ?i) ^ drm:DeliberatedMaliciousSWDistribution(?x) ^ drm:RiskScope(?rs) ^ swrlx:makeOWLThing(?r, ?x) ^ drm:probability(?x, ?p) -> drm:DeliberatedMaliciousSWDistributionRisk(?r) ^ drm:isGeneratedBy(?r, ?x) ^ drm:threatens(?r, ?rs) ^ stix2:type(?r, "Deliberated Malicious SW Distribution Risk"^^rdf:PlainLiteral) ^ drm:damages(?r, ?a)</pre>	

D.IV. Data Enrichment Rules

The reasoner will infer new knowledge about the enrichment of our data, if applying the following SWRL rules (examples) over our data-set.

D.IV.I. Enrichment#1 Auto Ipv4 Assignment By Domain Match (DNS)

Table 99 Enrichment#1 Auto IPv4 assignment by Domain match (DNS)

Enrichment#1 Auto IPv4 assignment by Domain match (DNS)	
The reasoner will infer a DNS value of a domain if a Reverse DNS connection is known.	
<pre>stix2:IPv4Addr(?ip) ^ stix2:belongsToRefs(?ip, ?ref) ^ stix2:DomainName(?ref) -> stix2:DomainName(?ref) ^ stix2:resolvesToRefs(?ref, ?ip)</pre>	

D.IV.II. Enrichment#2 Auto Domain Assignment By Ipv4 Match (Reverse Dns)

Table 100 Enrichment#2 Auto Domain assignment by IPv4 match (Reverse DNS)

Enrichment#2 Auto Domain assignment by IPv4 match (Reverse DNS)	
The reasoner will infer a RDNS value of a domain if a DNS connection is known.	
<pre>stix2:DomainName(?ref) ^ stix2:resolvesToRefs(?ref, ?ip) ^ stix2:IPv4Addr(?ip) -> stix2:IPv4Addr(?ip) ^ stix2:belongsToRefs(?ip, ?ref)</pre>	

D.IV.III. Enrichment#3 Auto Hw-Dependencies By Data Access

Table 101 Enrichment#3 Auto HW-dependencies by Data Access

Enrichment#3 Auto HW-dependencies by Data Access	
--	--

The reasoner will infer a dependency of Data into a specific Personal Computer (PC) depending on the user access rights. If a specific user has access rights to a specific type of data (e.g. confidential data), and the user uses a specific PC, the data will have direct dependencies on the pc of that user.

This will be useful for our dynamic risk management framework. On one hand, it provides an enhanced, automated and enriched dependency tree: for simplicity, it is a complex task if it has to be manually done by any ISMS framework. On the other hand, it provides dynamic inter-connection between security events on that specific pc to the risk assessment of that data along the time.

drm:Users(?u) ^ drm:hasAccessTo(?u, ?data) ^ drm:Computer(?pc) ^ stix2:uses(?u, ?pc) -> drm:Data(?data) ^ drm:dependsOn(?data, ?pc)

D.V. Risk Assessment Rules

The reasoner will infer new knowledge related to the Risk Assessment's process. It will make specific calculations by applying the following SWRL rules (examples) over our data-set.

D.V.I. Risk Assessment#1 Auto Potential Risk Assessment

Table 102 RiskAssessment#1 Auto Potential Risk Assessment

RiskAssessment#1 Auto Potential Risk Assessment
The reasoner will infer new instances of Potential Risk assessment classes, together will all their values, connections and dependencies.
<pre> drm:RiskScope(?rs) ^ stix2:type(?r, ?ty) ^ swrlb:add(?x, ?p, ?i) ^ drm:threatens(?r, ?rs) ^ drm:isGeneratedBy(?r, ?th) ^ drm:Risk(?r) ^ drm:impact(?th, ?i) ^ swrlx:makeOWLThing(?y, ?r) ^ drm:probability(?th, ?p) -> drm:impact(?y, ?i) ^ drm:threatens(?y, ?rs) ^ drm:evaluates(?y, ?r) ^ drm:potentialRisk(?y, ?x) ^ drm:PotentialRisk(?y) ^ drm:probability(?y, ?p) ^ stix2:type(?y, ?ty) </pre>

D.V.II. Risk Assessment#2 Auto Residual Risk Calculation for Bad Reputation Risk

Table 103 RiskAssessment#2 Auto Residual Risk Calculation for Bad Reputation Risk

RiskAssessment#2 Auto Residual Risk Calculation for Bad Reputation Risk
The reasoner will infer the value of the Residual Risk for those Risks related to "Bad reputation". For that, it will take into account the improvement (or mitigation values) provided by the safeguards of type "privilege-management-control".
<pre> swrlb:subtract(?ar, ?prisk, ?v) ^ drm:threatens(?x, ?rs) ^ drm:Data(?data) ^ drm:potentialRisk(?x, ?prisk) ^ drm:dependsOn(?rs, ?data) ^ stix2:type(?x, "Bad Reputation Risk"^^rdf:PlainLiteral) ^ stix2:type(?s, "privilege-management-control"^^rdf:PlainLiteral) ^ stix2:value(?s, ?v) ^ drm:Safeguards(?s) ^ drm:PotentialRisk(?x) -> drm:ResidualRisk(?x) ^ drm:actualRisk(?x, ?ar) ^ stix2:isMitigatedBy(?x, ?s) </pre>

D.V.III. Risk Assessment#3 Auto Residual Risk for Deliberated Malicious Sw Distribution

Table 104 RiskAssessment#3 Auto Residual Risk for Deliberated Malicious SW Distribution

RiskAssessment#3 Auto Residual Risk for Deliberated Malicious SW Distribution
The reasoner will infer the value of the Residual Risk for those Risks related to “Deliberated Malicious SW Distribution”. For that, it will take into account the improvement (or mitigation values) provided by the safeguards of type “control-against-malicious-sw”.
<pre>swrlb:subtract(?ar, ?prisk, ?v) ^ drm:potentialRisk(?x, ?prisk) ^ stix2:type(?s, "control-against-malicious-sw"^^rdf:PlainLiteral) ^ stix2:value(?s, ?v) ^ drm:Safeguards(?s) ^ drm:PotentialRisk(?x) ^ stix2:type(?x, "Deliberated Malicious SW Distribution Risk"^^rdf:PlainLiteral) -> drm:ResidualRisk(?x) ^ drm:actualRisk(?x, ?ar) ^ stix2:isMitigatedBy(?x, ?s)</pre>

D.V.IV. Risk Assessment#4 Increasing Risks Due To Security

Note: It is specific for Adobe Flash Player, this time.

Table 105 RiskAssessment#4 Increasing Risks due to Security Events

RiskAssessment#4 Increasing Risks due to Security Events (SPECIFIC FOR ADOBE FLASH PLAYER THIS TIME)
The reasoner will infer the new value of a specific risk (risk re-assessment) due a specific security event associated to their dependent assets (and CyberObservables).
<pre>drm:SecurityEvents(?se) ^ stix2:type(?se, "Dropper behaviour of Malicious Windows Executable"^^rdf:PlainLiteral) ^ stix2:srcRef(?se, ?i) ^ drm:PersonalComputer(?pc) ^ stix2:hasCyberObservable(?pc, ?i) ^ stix2:isUsedBy(?pc, ?u) ^ drm:Users(?u) ^ drm:hasAccessTo(?u, ?data) ^ drm:Data(?data) ^ drm:ResidualRisk(?rr) ^ drm:threatens(?rr, ?data) ^ drm:evaluates(?rr, ?ev) ^ drm:Risk(?ev) ^ drm:damages(?ev, drm:AdobeFlashPlayerPlugin) ^ stix2:type(?rr, "Deliberated Malicious SW Distribution Risk"^^rdf:PlainLiteral) ^ drm:probability(?rr, ?p) ^ drm:potentialRisk(?rr, ?pr) ^ drm:actualRisk(?rr, ?ar) ^ swrlb:add(?newp, ?p, 1) ^ swrlb:add(?newpr, ?pr, 1) ^ swrlb:add(?newar, ?ar, 1) -> drm:ResidualRisk(?rr) ^ stix2:isIncreasedBy(?rr, ?se)</pre>

D.VI. Risk Severity Rules

The reasoner will classify risks into their corresponding severity levels depending on their “actualRisk” value.

D.VI.I. Risk Severity#1 Extreme Risk Classification

Table 106 RiskSeverity#1 Extreme Risk Classification

RiskSeverity#1 Extreme Risk Classification
The reasoner will infer (and classify), that all Risks between 8 and 10 are considered of Extreme Risk Severity.
<pre> swrlb:lessThan(?ar, 10) ^ drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 8) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ drm:evaluates(?rr, ?ris) -> drm:ExtremeRiskSeverity(?x) ^ stix2:type(?x, "Extreme risk- severity"^^rdf:PlainLiteral) ^ drm:evaluates(?x, ?ris) ^ stix2:value(?x, ?ar) </pre>

D.VI.II. Risk Severity#2 High Risk Classification

Table 107 RiskSeverity#2 High Risk Classification

RiskSeverity#2 High Risk Classification
The reasoner will infer (and classify), that all Risks between 6 and <8 are considered of High Risk Severity.
<pre> drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 6) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ drm:evaluates(?rr, ?ris) ^ swrlb:lessThan(?ar, 8) -> drm:evaluates(?x, ?ris) ^ stix2:type(?x, "High risk- severity"^^rdf:PlainLiteral) ^ drm:HighRiskSeverity(?x) ^ stix2:value(?x, ?ar) </pre>

Name	RiskSeverity#2 High Risk Classification
Comment	
Status	Ok
	<pre> drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 6) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ drm:evaluates(?rr, ?ris) ^ swrlb:lessThan(?ar, 8) -> drm:evaluates(?x, ?ris) ^ stix2:type(?x, "High risk-severity") ^ drm:HighRiskSeverity(?x) ^ stix2:value(?x, ?ar) </pre>

Figure 99 SWRL rule for auto classification of high severity risks

D.VI.III. Risk Severity#3 Medium Risk Classification

Table 108 RiskSeverity#3 Medium Risk Classification

RiskSeverity#3 Medium Risk Classification
The reasoner will infer (and classify), that all Risks between 5 and <6 are considered of Medium Risk Severity.


```

    drm:actualRisk(?rr, ?ar) ^ swrlb:greaterThanOrEqual(?ar, 5) ^
    drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ swrlb:lessThan(?ar, 6) ^
    drm:evaluates(?rr, ?ris)

    ->

    drm:MediumRiskSeverity(?x) ^ drm:evaluates(?x, ?ris) ^ stix2:value(?x, ?ar) ^
    stix2:type(?x, "Medium risk-severity"^^rdf:PlainLiteral)

```

D.VI.IV. Risk Severity#4 Low Risk Classification

Table 109 RiskSeverity#4 Low Risk Classification

RiskSeverity#4 Low Risk Classification
The reasoner will infer (and classify), that all Risks below < 5 are considered of Low Risk Severity.
<pre> drm:actualRisk(?rr, ?ar) ^ swrlb:lessThan(?ar, 5) ^ drm:ResidualRisk(?rr) ^ swrlx:makeOWLThing(?x, ?rr) ^ drm:evaluates(?rr, ?ris) -> drm:LowRiskSeverity(?x) ^ drm:actualRisk(?x, ?ar) ^ stix2:value(?x, ?ris) ^ stix2:type(?x, "Low risk-severity"^^rdf:PlainLiteral) </pre>

D.VII. Security Policy Rules

The reasoner will infer new knowledge that can be applied into the Dynamic Risk Framework as a Security Policy.

D.VII.I. Security Policy#1 Data-Access-Control Rule

Table 110 SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)

SecurityPolicy#1 Data-Access-Control Rule (automated compilation of allowed users to each data)
The reasoner will infer new knowledge, as a Security Policy, and it will automatically apply it to all the data-set. In this case, the example show how easy is, for our framework, to know what users have access to what data, depending on the data required access level and the level access of each user. This SWRL rule will create enriched connections and dependencies that will be very useful to estimate new type of risks for our dynamic model.
<pre> drm:Users(?u) ^ drm:Data(?data) ^ drm:hasAccessLevel(?u, ?a) ^ swrlb:greaterThanOrEqual(?a, ?r) ^ drm:requiredAccessLevel(?data, ?r) -> drm:hasAccessTo(?u, ?data) </pre>

D.VIII. Risk Management Rules

The reasoner will infer new knowledge with regard to the strategy to follow depending on the Residual Risk value of each risk. It will help to implement the specific policy of each organization. In our case:

Risk are classified in the following scale (from 1 to 10):

- Extreme (8 ≤ extreme ≤ 10): risk is unacceptable,
- High (6 ≤ high < 8): risk is undesirable,
- Medium (5 ≤ medium < 6): risk is tolerable,
- Low (< 5): risk is acceptable.

A Mitigation strategy is recommended for Extreme and High risks with different possibilities:

- reduce,
- to avoid,
- to share,
- to transfer the risk.

An Investigation strategy is recommended for Medium risks with different possibilities: to assume or to reduce the risk.

A Monitoring strategy is recommended for Low risks where usually risks are assumed.

D.VIII.I. Risk Management#1 Mitigation Strategy for Extreme Risk Severity

Table 111 RiskManagement#1 Mitigation Strategy for Extreme Risk Severity

RiskManagement#1 Mitigation Strategy for Extreme Risk Severity	
The reasoner will infer (and classify), that all Extreme Risk Severity's instances will have to be mitigated. It will create new instances of the class Risk Mitigation Strategy associated to each extreme risk.	
<pre> stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ swrlx:makeOWLThing(?x, ?er) ^ drm:ExtremeRiskSeverity(?er) -> drm:RiskMitigation(?x) ^ stix2:type(?x, "Risk Mitigation Strategy"^^rdf:PlainLiteral) ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex) </pre>	

D.VIII.II. Risk Management#2 Mitigation Strategy for High Risk Severity

Table 112 RiskManagement#2 Mitigation Strategy for High Risk Severity

RiskManagement#2 Mitigation Strategy for High Risk Severity	
---	--

The reasoner will infer (and classify), that all High Risk Severity's instances will have to be mitigated. It will create new instances of the class Risk Mitigation Strategy associated to each high risk.

```

    drm:HighRiskSeverity(?er) ^ stix2:value(?er, ?sev) ^ drm:evaluates(?er,
?risex) ^ swrlx:makeOWLThing(?x, ?er)
    ->
    drm:RiskMitigation(?x) ^ stix2:type(?x, "Risk Mitigation
Strategy"^^rdf:PlainLiteral) ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex)

```

Name
RiskManagement#2 Mitigation Strategy for High Risk Severity
Comment
Status
Ok
<pre> drm:HighRiskSeverity(?er) ^ stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ swrlx:makeOWLThing(?x, ?er) -> drm:RiskMitigation(?x) ^ stix2:type(?x, "Risk Mitigation Strategy") ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex) </pre>

Figure 100 SWRL rule used in risk management for automatic classification. Once a high severity risk is detected, it is automatically classified for a mitigation strategy because the risk is not acceptable for the organization as is

D.VIII.III. Risk Management#3 Investigation Strategy for Medium Risk Severity

Table 113 RiskManagement#3 Investigation Strategy for Medium Risk Severity

RiskManagement#3 Investigation Strategy for Medium Risk Severity
<p>The reasoner will infer (and classify), that all Medium Risk Severity's instances will have to be investigated. It will create new instances of the class Risk Investigation Strategy associated to each medium risk.</p>
<pre> stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ drm:MediumRiskSeverity(?er) ^ swrlx:makeOWLThing(?x, ?er) -> drm:RiskInvestigation(?x) ^ stix2:type(?x, "Risk Investigation Strategy"^^rdf:PlainLiteral) ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex) </pre>

D.VIII.IV. Risk Management#4 Monitoring Strategy for Low Risk Severity

Table 114 RiskManagement#4 monitoring Strategy for Low Risk Severity

RiskManagement#4 Monitoring Strategy for Low Risk Severity
--

The reasoner will infer (and classify), that all Low Risk Severity's instances will have to be monitored. It will create new instances of the class Risk Monitoring Strategy associated to each low risk.

```

stix2:value(?er, ?sev) ^ drm:evaluates(?er, ?risex) ^ swrlx:makeOWLThing(?x,
?er) ^ drm:LowRiskSeverity(?er)
->
stix2:type(?x, "Risk Monitoring Strategy"^^rdf:PlainLiteral) ^
drm:RiskMonitoring(?x) ^ stix2:value(?x, ?sev) ^ drm:manages(?x, ?risex)

```

D.IX. Threat Intelligence Rules

The reasoner will infer new knowledge due to semantic Threat Intelligence rules, which will be applied into the Dynamic Risk Framework to automate and provide better Detection and Prevention services.

By using the expressiveness provided by SWRL, improved TTP detection algorithms are now feasible, like the example shown below.

D.IX.I. Threat Intelligence#1 Security Event Dropper Detection (TTP)

Table 115 ThreatIntelligence#1 Security Event Dropper Detection

ThreatIntelligence#1 Security Event Dropper Detection

The reasoner will infer new knowledge, in this case, a security event will be created. The SWRL rule represents a "TTP Detection" algorithm of a dropper behavior. Our contribution (based on an OWL version of STIX, SWRL and the reasoner) allows the creation of context-aware and behavioral rules independent of IoC (Indicators of Compromise). In this case, a malicious windows executable being dropped after a sucesion of javascript redirections is noticed. The security event will have associated all the cyberobservables and will trigger a re-assessment of the risk.

```

stix2:NetworkTraffic(?nt) ^ stix2:dstPayloadRef(?nt, ?pl) ^ stix2:Artifact(?pl) ^
stix2:mimeType(?pl, "javascript"^^rdf:PlainLiteral) ^ stix2:redirection(?pl, ?red) ^
stix2:URL(?red) ^ stix2:NetworkTraffic(?nt2) ^ stix2:dstRef(?nt2, ?red) ^
stix2:dstPayloadRef(?nt2, ?pl2) ^ stix2:srcRef(?nt2, ?sr) ^ stix2:extensions(?pl2,
"windows-pebinary-ext"^^rdf:PlainLiteral) ^ stix2:name(?pl2, ?nm) ^
swrlx:makeOWLThing(?x, ?nt2)
->
drm:SecurityEvents(?x) ^ stix2:type(?x, "security-event"^^rdf:PlainLiteral) ^
stix2:srcRef(?x, ?sr) ^ stix2:type(?x, "Dropper behaviour of Malicious Windows
Executable"^^rdf:PlainLiteral) ^ stix2:dstRef(?x, ?red) ^ stix2:relatedTo(?x, ?nt2)

```

Name
ThreatIntelligence#1 Security Event Dropper Detection
Comment
Status
Ok
<pre> stix2:NetworkTraffic(?nt) ^ stix2:dstPayloadRef(?nt, ?pl) ^ stix2:Artifact(?pl) ^ stix2:mimeType(?pl, "javascript") ^ stix2:redirection(?pl, ?red) ^ stix2:URL(?red) ^ stix2:NetworkTraffic(?nt2) ^ stix2:dstRef(?nt2, ?red) ^ stix2:dstPayloadRef(?nt2, ?pl2) ^ stix2:srcRef(?nt2, ?sr) ^ stix2:extensions(?pl2, "windows-pebinary-ext") ^ stix2:name(?pl2, ?nm) ^ swrlx:makeOWLThing(?x, ?nt2) -> drm:SecurityEvents(?x) ^ stix2:type(?x, "security-event") ^ stix2:srcRef(?x, ?sr) ^ stix2:type(?x, "Dropper behaviour of Malicious Windows Executable") ^ stix2:dstRef(?x, ?red) ^ stix2:relatedTo(?x, ?nt2) </pre>

Figure 101 Pattern based SWRL rule to detect a Security Event that will be triggered by a dropper executable delivered automatically after a network traffic redirection of an injected Javascript takes place. It has not specific IOC, all individuals are variables

ANNEX E. LIST OF ABBREVIATIONS AND ACRONYMS

C

CMMI	<i>Capability Maturity Model Integration</i>
COBIT	<i>Control Objectives for Information and related Technology</i>
CRAMM	<i>CCTA Risk Analysis and Management Method</i>
CTI	<i>Cyber Threat Intelligence</i>
CVE	<i>Common Vulnerabilities and Exposures</i>
CVSS	<i>Common Vulnerability Scoring System</i>

D

DAML	DARPA Agent Markup Language
DAML+OIL	DARPA Agent Markup Language + Ontology Inference Layer
DL	Description Logic
DRA	Dynamic Risk Assessment
DRM	Dynamic Risk Management

E

ERC20	Ethereum Request for Comments used for Ethereum Standard Token
EVM	Ethereum Virtual Machine

F

FaCT	Fast Classification of Terminologies
-------------	--------------------------------------

H

HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol

I

IoC	Indicator of Compromise
IODEF	Incident Object Description Exchange Format
ISMS	Information Security Management System
ISO	International Organization for Standardization
ITIL	Information Technology Infrastructure Library

J

JSON Javascript Object Notation

K

KAoS Knowledge-able Agent-oriented System

M

MAGERIT Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información

MEHARI MEthod for Harmonized Analysis of Risk

MISP Malware Information Sharing Platform

N

NATO North Atlantic Treaty Organization

NIS Network and Information Security

NIST National Institute of Standards and Technology

O

OASIS Organization for the Advancement of Structured Information Standards

OCTAVE Operationally Critical Threat, Assesed and Vulnerability Evaluation

OPENIOC Open Framework for Sharing IoC

OSINT Open Source Intelligence

OWL Web Ontology Language

OWL-S OWL for Services

R

RACER Renamed Abox and Concept Expression Reasoner

RDF Resource Description Framework

RDFS RDF Schema

RFC Request for Comments

RULEML Rule Markup Language

S

SHOE Simple HTML Ontology Extensions

STIX Structure Threat Information Exchange

SWRL Semantic Web Rule Language

SWSL Semantic Web Services Language

T

TAXII	Trusted Automated eXchange of Indicator Information
TCP	Transmission Control Protocol

U

UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

W

W3C	World Wide Web Consortium
------------	---------------------------

X

XML	eXtensible Markup Language
------------	----------------------------