UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

CONTRIBUCIONES A LA GESTIÓN DE INCIDENTES DE CIBERSEGURIDAD Y PREPARACIÓN FORENSE EN REDES DEFINIDAS POR SOFTWARE

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

María Belén Jiménez Amoroso
Ingeniera en Sistemas mención Telemática

Madrid, 2024

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Doctorado en Ingeniería de Sistemas Telemáticos

CONTRIBUCIONES A LA GESTIÓN DE INCIDENTES DE CIBERSEGURIDAD Y PREPARACIÓN FORENSE EN REDES DEFINIDAS POR SOFTWARE

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

María Belén Jiménez Amoroso
Ingeniera en Sistemas mención Telemática

Bajo la supervisión de:
Dr. David Fernández Cambronero (Director)

Madrid, 2024

Título: CONTRIBUCIONES A LA GESTIÓN DE INCIDENTES DE CIBERSEGURIDAD Y PREPARACIÓN FORENSE EN REDES DEFINIDAS POR SOFTWARE

Autor: María Belén Jiménez Amoroso

Programa de Doctorado: Ingeniería de Sistemas Telemáticos

Dirección de Tesis:

Dr. David Fernández Cambronero, Profesor Titular de la E.T.S.I.T, Universidad Politécnica de Madrid (Director)

Revisores Externos:

Tribunal de Tesis:

Fecha de Defensa de Tesis:

El trabajo de investigación presentado en esta tesis doctoral fue apoyado por la República del Ecuador a través de SENESCYT - Secretaría de Educación Superior, Ciencia, Tecnología e Innovación del Ecuador; por el Ministerio Español de Ciencia e Innovación en el marco del proyecto ECTICS (PID2019-105257RB-C21) y por el Ministerio Español de Ciencia, Innovación y Universidades en el marco del proyecto Go2Edge (RED2018-102585-T).

Al esfuerzo y la constancia

Agradecimientos

Muchos pensamos que el llegar a la meta es lo más importante, pero en realidad lo mejor de alcanzarla es el camino que trazamos, pues cuando vamos avanzando nos enfrentamos a varias situaciones que ponen a prueba nuestra capacidad de resolución de problemas. Esa capacidad muchas veces se ve opacada por los miedos, que de no ser por Dios y por quienes han confiado en nosotros no seríamos capaces de derribarlos. ¡A todos quienes no me han dejado caer, gracias por estar siempre ahí!

Al ser más importante durante todo este tiempo, Dios, porque sin Él nada soy.

A mi esposo, amigo y confidente; Jorge Eduardo, gracias por haber sido mi más grande apoyo en este recorrido. Tu has sido ese "Pepe grillo" que todos necesitamos en nuestro paso terrenal y el motor que me ha impulsado a seguir adelante. Gracias por tanta comprensión, por todos los sacrificios que solo tú y yo conocemos, por escucharme durante horas y ser mi guía, por tu valioso tiempo, por tus consejos, por tus regaños totalmente justificados, por secar mis lágrimas tras cada momento de debilidad, por ayudarme a superar mis miedos, por las risas cómplices, por los buenos momentos, por tu conocimiento, por ayudarme a conseguir mis objetivos. En otras palabras, por ser incondicional. ¡Gracias por tanto, amor!

A mis padres; Lily y Patricio, quienes me han brindado su apoyo, su amor, sus palabras de motivación y aliento, por darme la fortaleza y por tener predisposición para hablar cuando necesitaba un consejo o favor a lo lejos. Siempre se ingeniaban la manera para darme una mano y hacerme las cosas más fáciles. Gracias papitos, siempre reconoceré que han sido mis ángeles terrenales.

A mis hermanos Gabriel y David, y a mi tía Elsita, porque a pesar de la distancia nunca me han dejado sola y siempre me han visto con ojos de cariño. Ustedes a través de sus conversaciones y de las fotos de los más chiquitos de la familia; tal vez sin saberlo, me devolvían la energía cuando más lo necesitaba.

A mis suegros, a mis cuñadas y a doña Lelia, gracias por estar siempre pendientes de mi progreso, por enviarme mensajes, fotos y por las conversaciones que han ayudado a que este camino sea más ligero.

A mis primos y amigos que a pesar de la distancia siguen en contacto siempre con la actitud positiva, Cesitar, Verito, Pauly, Estefy, Lily, Edwin, Alexita, y especialmente Rita, por ser una amiga maravillosa y por enviarme audios que han llenado de esperanza mi corazón.

A las personas que han llegado a mi vida durante estos últimos años y que me han alegrado con su presencia, Sandrita, Ricardo, Esther y David. A Marcelo y Érica, gracias por su amistad, por su ayuda y por volverse parte de mi entorno.

A mi tutor y supervisor; Dr. David Fernández, gracias por el acompañamiento en este período de formación. Gracias por la confianza depositada en mí, por ser un guía que comparte el conocimiento, por la paciencia, la tolerancia y sobre todo gracias por el buen trato.

A todos los integrantes del grupo GIROS, por la integración y su valioso tiempo para

escucharme y hacer comentarios que mejoraban notablemente mi desempeño.

Gracias profundas a mi país y a la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación del Ecuador y a la Universidad Politécnica de Madrid por el apoyo y auspicio.

Abstract

This doctoral thesis addresses challenges related to cybersecurity management in SDN environments, considering forensic readiness processes and incident response. In order to comprehend the evolution of cybersecurity in this paradigm, the first contribution details the SDN architecture, identifying its cybersecurity issues and analyzing solutions using the STRIDE methodology. The vulnerability of SDN deployments to malicious actions is highlighted, emphasizing the urgent need for adopting a holistic cybersecurity management approach that considers both technical and organizational aspects, as a cybersecurity incident can transcend the technical realm to affect other environments, including the legal one. Given this, there is a significant research opportunity in cybersecurity incident management with a forensic focus.

Exploring the realm of forensic and incident response, key terms are introduced, and related works on SDN are analyzed. Unaddressed challenges are highlighted, acknowledging the lack of synergy between forensic processes and incident response, with significant implications for organizations. It is recognized that inadequate management of these concepts could affect the finding of the root cause of a cybercrime, impacting everything from service quality to organizational reputation. Therefore, the need for frameworks or models that integrate cybersecurity incident management and forensic preparation processes is emphasized.

In this regard, the next contribution of this doctoral thesis is the proposal of a framework for managing cybersecurity in SDN environments, integrating incident management and forensic readiness processes. Additionally, an architecture supporting the applicability of the framework is proposed, detailing event information sources, functional components, and interactions in forensic and incident response processes.

The following contributions are framed within two key models focused on data filtering, acquisition, and treatment, and the preservation of information integrity regarding a cybersecurity event in SDN environments. For the first model, artificial intelligence is explored with various techniques and algorithms, completing data extraction, transformation, and loading (ETL) processes of network traffic. As for the second model, private or permissioned distributed ledger technologies are chosen to provide a secure and transparent environment for recording and auditing transactions, ensuring information integrity, and enabling controlled and authorized access.

The validation of the proposals is carried out in controlled environments, where test scenarios are designed, and weaknesses exploited in terms of cybersecurity are highlighted. Subsequently, the proposals are evaluated on virtualization platforms, presenting the results obtained regarding the performance of each model, according to their respective evaluation parameters.

Finally, the conclusions of the doctoral thesis are presented, summarizing the contributions and providing a comprehensive analysis of the research's relevance. Recommendations for future research and improvement actions based on the findings are also included. Additionally, the dissemination of the research is detailed, listing specialized journals and academic conferences where the results have been presented.

Resumen

Esta tesis doctoral aborda desafíos relacionados con la gestión de la ciberseguridad en entornos SDN considerando los procesos de preparación forense y de respuesta a incidentes. A fin de comprender la evolución de ciberseguridad en este paradigma, como primera contribución se detalla la arquitectura SDN, identificando sus problemas de ciberseguridad y analizando soluciones con la metodología STRIDE. Se destaca la vulnerabilidad de los despliegues SDN ante acciones maliciosas, por lo que urge la adopción de una visión holística de gestión de ciberseguridad que considere aspectos técnicos y organizacionales, puesto que un incidente de ciberseguridad puede trascender el ámbito técnico para afectar otros entornos, incluido el jurídico. Visto esto, se observa una gran oportunidad de investigación en el manejo de incidentes de ciberseguridad con un enfoque forense.

Profundizando en el mundo forense y de respuesta a incidentes, se introducen los términos clave y se analizan los trabajos relacionados con las SDN. Se destacan desafíos no abordados, reconociendo la falta de sinergia entre procesos forenses y de respuesta a incidentes, con implicaciones significativas para las organizaciones. Se reconoce que una gestión inadecuada de estos conceptos podría afectar el hallazgo de la causa raíz de un cibercrimen, impactando desde la calidad de los servicios hasta la reputación organizacional. Por lo que, se enfatiza la necesidad de frameworks o modelos que integren la gestión de incidentes de ciberseguridad y los procesos de preparación forense.

En este sentido, la siguiente contribución de esta tesis doctoral es la propuesta de un framework para gestionar la ciberseguridad en entornos SDN, integrando la gestión de incidentes y los procesos de preparación forense. Además, se propone una arquitectura que respalda la aplicabilidad del framework, detallando las fuentes de información de eventos, los componentes funcionales y las interacciones en los procesos forenses y de respuesta a incidentes.

Las siguientes contribuciones se enmarcan en dos modelos clave centrados en el filtrado, adquisición y tratamiento de datos; y la preservación de la integridad de la información relativa a un evento de ciberseguridad en entornos SDN. Para el primer modelo, se explora la inteligencia artificial con diversas técnicas y algoritmos, completando procesos de extracción, transformación y carga de datos (ETL) del tráfico de red. En cuanto al segundo modelo, se opta por tecnologías de ledger distribuidos privados o permisionados, que proporcionan un entorno seguro y transparente para registrar y auditar transacciones, garantizando la integridad de la información y permitiendo un acceso controlado y autorizado.

La validación de las propuestas se lleva a cabo en ambientes controlados, donde se esquematizan los escenarios de prueba y se destacan las debilidades explotadas en términos de ciberseguridad. Posteriormente, se procede a evaluar las propuestas en plataformas de virtualización, presentando los resultados obtenidos en cuanto al rendimiento de cada modelo, de acuerdo con sus respectivos parámetros de evaluación.

Finalmente, se presentan las conclusiones de la tesis doctoral, resumiendo las contribuciones y ofreciendo un análisis global sobre la relevancia de la investigación. También se incluyen recomendaciones para futuras investigaciones y acciones de mejora basadas en los hallazgos

obtenidos. Además, se detalla la diseminación de la investigación, enumerando las revistas especializadas y las conferencias académicas donde se han presentado los resultados.

Índice general

Índice de figuras

Índice de Tablas

Abreviaturas y acrónimos

AAA Authentication, Authorization and Accounting

ACL Access Control List

AE Autoencoder

AI Artificial Intelligence

AIDS Anomaly-based Intrusion Detection System

ANN Artificial Neural Network

API Application Programming Interface

ARP Address Resolution Protocol

BDDP Broadcast Domain Discovery Protocol

BGP Border Gateway Protocol

BGRU Bidirectional Gated Recurrent Unit

CAP Consistency, Availability, Partition Tolerance

CAPEX Capital Expenditure

CFG Control-Flow Graph

CPS Cyber Physical System

CNN Convolutional Neural Network

CSP Cloud Service Providers

DAC Discretionary Access Control

DDoS Distributed Denial of Service

DESO Digital Evidences Semantic Ontology

DFIR Digital Forensics and Incident Response

DFS Distributed Firewall Service

DWT Discrete Wavelet Transform

DLBS Distributed Load Balancer Service

DNS Domain Name System

DoS Denial of Service

DP Data Plane

DTLS Datagram Transport Layer Security

EAP Extensible Authentication Protocol

EAPoL Extensible Authentication Protocol over LAN

ECC Elliptic Curve Cryptography

ENISA European Union Agency for Cybersecurity

EU European Union

EWMA Exponentially Weighted Moving Average

E/WI East/West Interface

FE Feature Extraction

FIFO First-In First-Out

FN False Negative

FP False Positive

FS Feature Selection

FSM Finite State Machines

FPGA Field-Programmable Gate Array

GRU-RNN Gated Recurrent Unit Recurrent Neural Network

HTS Host Tracking Service

IBC Identity-Based Cryptography

ICS Industrial Control System

IDS Intrusion Detection System

IETF Internet Engineering Task Force

IoE Internet of Everything

IIoT Industrial Internet of Things

IoT Internet of Things

IPS Intrusion Prevention System

ISO International Organization for Standardization

ISOMAP Isometric Mapping

KNN K-Nearest Neighbors

KPCA Kernel Principal Component Analysis

LDA Linear Discriminant Analysis

LDS Link Discovery Service

LLDP Link Layer Discovery Protocol

LLE Locally Linear Embedding

LR Logistic Regression

LRU Least Recent Used

LSI Latent Semantic Indexing

LSTM Long Short-Term Memory

MAC Mandatory Access Control

MAC Media Access Control Address

MDS Multi-Dimensional Scaling

MiM Man-in-the-Middle

MTU Maximum Transfer Unit

MLP Multi-Layer Perceptron

NB Naive Bayes

NAT Network Address Translation

NBI Northbound Interface

NIST National Institute of Standards and Technology

NFV Network Function Virtualization

NMDA Network Management Datastore Architecture

NN Neural Network

NSS Network Security Services

ODL OpenDaylight

OF OpenFlow

OFDP OpenFlow Discovery Protocol

ONF Open Networking Foundation

ONOS Open Network Operating System

OPEX Operational Expenditures

OVSDB Open vSwitch Database

P2P Peer-to-Peer

PCA Principal Component Analysis

PoW Proof of Work

PoS Proof of Stake

QoS Quality of Service

QoE Quality of Experience

RF Random Forest

R2L Remote to Local

RADIUS Remote Authentication Dial-In User Service

RBAC Role-Based Access Control

RBM Restricted Boltzmann Machine

ReLU Rectified Linear Unit

REST Representational State Transfer

RFE Recursive Feature Elimination

RNN Recurrent Neural Networks

RQ Research Question

SBI Southbound Interface

SDN Software-Defined Networking

SIDS Signature-based Intrusion Detection System

SLA Service Level Agreement

SOM Self Organizing Maps

SSBG Security-Sensitive Behavior Graphs

SSL Secure Socket Layer

SVM Support Vector Machine

Tanh Tangente Hiperbólica

TCAM Ternary Content-Addressable Memory

TCP Transmission Control Protocol

TIET Thapar Institute of Engineering Technology

TP True Positive

TLS Transport Layer Security

TN True Negative

TO Telecommunications Operator

U2R User to Root

UDP User Datagram Protocol

WAN Wide Area Network

XFSM eXtensible Finite State Machines

XXE XML External Entity

Capítulo 1

INTRODUCCIÓN

1.1 Contexto y Motivación

El paradigma SDN se encuentra en constante evolución y crecimiento, a tal punto que está transformando el mundo de las telecomunicaciones, pues ha llegado a ser usado como tecnología habilitadora en el despliegue de nuevas propuestas como 5G, 6G o el Internet del Todo (Beshley et al., 2022). De hecho, se ha observado que en la actualidad un 64 % de los centros de datos utilizan SDN para transportar datos dentro de las empresas, que el 58 % de los despliegues de redes de área extendida (WAN) aprovechan las ventajas de SDN para garantizar la entrega de servicios, y que un 40 % de redes de acceso trabajan con SDN para mejorar la gestión y propagación de políticas de la red (Cisco, 2020). Tal ha sido su impacto en nuevos despliegues, que hasta el año 2025 se espera un incremento de hasta en un 19 % del uso de estas redes (BusinessWire, 2020).

En gran medida los principales impulsadores del crecimiento del uso de SDN son los proveedores de servicios de red y operadores de telecomunicaciones, quienes han visto la innovación que las SDN representan en relación con las redes tradicionales, por lo cual han decidido invertir en la automatización de sus infraestructuras de red (BusinessWire, 2020). Esto ha ocasionado que los requerimientos que el mundo de la tecnología demanda de este paradigma sean cada vez más estrictos para contar con despliegues más dinámicos, flexibles y sobre todo seguros.

La arquitectura SDN se basa en el desacoplamiento del plano de datos del de control, que tiene como actor principal al controlador. El controlador es la parte más sensible de toda la arquitectura ya que es el responsable de la gestión centralizada de la red, permitiendo la implementación de políticas de red dinámicas, la automatización de tareas y la orquestación de recursos de red. El controlador interactúa con el plano de datos a través de la interfaz sur, mientras que para las aplicaciones usa la interfaz norte. Adicionalmente, existe la interfaz este-oeste que es la responsable de interconectar controladores distribuidos (CISCO, 2019; Open Networking Foundation, 2018b).

Esta arquitectura supone una serie de ventajas relativas a la administración y la proyección de crecimiento de la infraestructura, dado que con control centralizado se puede tener un panorama ampliado de la red, sus servicios y las necesidades respecto a los recursos. También

puede ofrecer un control flexible sobre los flujos de datos por lo que es adaptable a varias topologías de red, simplificando el diseño de hardware de las funciones de red físicas. Al mismo tiempo, al ser un paradigma con tendencia al uso de código abierto, ofrece un abanico de posibilidades con relación a las aplicaciones, las cuales pueden ser desarrolladas incluso por terceras partes, características que al final se traducen en un beneficio económico.

En contraste con las ventajas previamente mencionadas, se deben considerar los principales desafíos de ciberseguridad presentes en la arquitectura SDN. Es así que la Open Networking Foundation (ONF) ha emitido un documento que destaca las preocupaciones relacionadas con el control centralizado, la dicotomía inherente a la programabilidad de la red, las dificultades para integrar protocolos heredados como DNS, NAT o BGP en la arquitectura SDN sin un análisis previo de compatibilidad, y los problemas de confianza en la interconexión entre distintos dominios (Open Networking Foundation, 2015b). Este tema se torna preocupante, especialmente tomando en cuenta el incremento de ciberataques de hasta el 51 % en relación al año 2019, con una clara tendencia ascendente (Check Point, 2022). Dicha tendencia inquieta tanto a los actores del sector de las telecomunicaciones como a la Agencia de la Unión Europea para la Ciberseguridad (ENISA, por sus siglas en inglés), ya que se reconoce el riesgo de pérdidas económicas significativas debido a la falta de normativas y políticas de seguridad efectivas para hacer frente a los ataques (Prieto et al., 2023). En consecuencia, los desafíos en materia de ciberseguridad plantean incertidumbre para las organizaciones que utilizan la arquitectura SDN, ya que su exposición a diversas vulnerabilidades inherentes y del entorno, las convierte en blancos potenciales para futuros ataques cibernéticos.

Esto es sumamente relevante porque cuando se desencadena un ataque dentro de una organización, existe la posibilidad de una pérdida parcial o total de los servicios proporcionados y en el peor de los casos, incluso se podría comprometer el control sobre la información de los clientes o usuarios de la red, lo que puede derivar en acciones jurídicas, por lo que surgen necesidades tanto a nivel técnico como organizacional. Desde una perspectiva técnica, el personal involucrado se esfuerza por mitigar el daño en la red y restaurar los servicios lo más rápido posible. Mientras que, desde una perspectiva organizacional, la magnitud del evento de ciberseguridad podría resultar en una violación de los acuerdos de nivel de servicio (SLA, por sus siglas en inglés), pérdidas o afectaciones económicas y, en última instancia, podría conducir a implicaciones legales. Es por ello que, tras un episodio de vulnerabilidad o ataque a la SDN, resulta crucial comprender los factores desencadenantes que provocaron el incidente para poder atender tanto los requerimientos técnicos comprendiendo los problemas subyacentes en la red, así como los organizaciones relativos a vinculación jurídica. En este sentido, se podrían articular de manera integral tanto las necesidades técnicas como las organizacionales manteniendo un adecuado manejo de los incidentes desde una perspectiva forense.

Las ciencias forenses digitales han tenido un despunte en los últimos años, ya que a través de su interacción con otras disciplinas han logrado: afrontar problemas operacionales, recuperar datos, identificar violaciones de políticas y auditoría, y fortalecer los procesos del ciclo de vida de respuesta a incidentes. En consecuencia, la información obtenida de una apropiada preparación forense, permitiría reconstruir los escenarios de ataque y de este modo mejorar las políticas de ciberseguridad (S. Khan; Gani; Wahab et al., 2016). Paralelamente, utilizando la misma información antes referida, se podría crear un mapa de evidencia para usarla en

procesos legales vinculados con actividades delictivas en entornos digitales, con lo cual se emitiría una respuesta legalmente sólida y efectiva ante actos de orientación jurídica derivados de problemas de ciberseguridad.

Dado que tanto los procesos de preparación forense digital como los de respuesta a incidentes de ciberseguridad comparten un interés común en la información de un evento, lo óptimo sería abordarlos simultáneamente para mitigar los daños y recopilar evidencia digital consolidada, en el caso de necesatarlo. Sin embargo, diversos desafíos han llevado a una situación en la que ambos conceptos, en la práctica, sean tratados de manera independiente. Esta separación puede entenderse en cierta medida, ya que las organizaciones que enfrentan un incidente de ciberseguridad a menudo carecen de pautas estructuradas para ejecutar los procesos y experimentan un alto nivel de estrés, lo que puede llevarlas a posponer o incluso a omitir los procesos forenses y centrarse exclusivamente en remediar las fallas (Singh et al., 2022).

Así mismo, esta falta de sinergia entre los procesos de preparación forense y los del ciclo de vida de respuesta a incidentes genera un conflicto sobre la gestión de la información de los eventos de ciberseguridad, lo cual lleva a cada uno de los interesados a actuar voluntariamente en la obtención de dicha información. Esto puede resultar desfavorecedor para una organización, pues si se obtienen datos de fuentes no relacionadas, con diferentes características y ventanas de tiempo ajustados a perspectivas individuales, se puede dilatar el tiempo para el hallazgo del origen del problema de ciberseguridad. Por consiguiente, se crean ambigüedades sobre el escenario del incidente, lo cual opaca la adopción de medidas de ciberseguridad y aumenta el riesgo de arrojar una suposición de culpabilidad distorsionada, teniendo serias repercusiones sobre la calidad de los servicios que se entregan y por ende en la reputación de la organización (Singh et al., 2022).

Las discordancias antes mencionadas son aún más evidentes en entornos SDN. La razón detrás de esto radica justamente en la naturaleza dinámica presente en el desacoplamiento de las capas y al ambiente altamente programable de las SDN, lo que plantea un desafío significativo en la identificación de fuentes de información y recopilación de datos cuando se produce un evento de ciberseguridad. Como resultado, la adopción de un enfoque combinado que integre conceptos de ciencia forense digital y respuesta a incidentes en entornos SDN puede retardarse. La capacidad para llevar a cabo investigaciones forenses efectivas y para responder rápidamente a incidentes de ciberseguridad en estos entornos requiere una comprensión sólida de cómo se gestionan y controlan las SDN, así como la identificación de las fuentes de información pertinentes.

De igual modo se debe considerar los retos inherentes a la implementación de los procesos de preparación forense en las SDN, tales como la seguridad de los archivos de registros de actividad, el desempeño de la red cuando se usan mecanismos digitales forenses, la posible suplantación de las fuentes de registros de actividad, la sincronización de dichos registros cuando se cuentan con varios controladores SDN, el costo beneficio de la implementación de una preparación forense digital, la falta de personal con conocimientos especializados del área forense digital, la falta de políticas en estos ambientes, el rendimiento de procesamiento y almacenamiento, el almacenamiento de evidencias sin alineación a los requerimientos de retención y la falta de un adecuado mecanismo de divulgación de la información para que se encuentre disponible para su consumo, conforme las necesidades de cada interesado y

entorno (S. Khan; Gani; Wahab et al., 2016; Karie et al., 2021). Mientras que en lo que refiere a los desafíos en la implementación de procesos de respuesta a incidentes, las principales preocupaciones se enmarcan en el tratamiento de los eventos de manera individual, la capacidad de reacción ante un incidente, el tiempo invertido en el análisis de la información y el riesgo en el manejo de los falsos positivos (Salfati et al., 2022).

Superar estos desafíos es fundamental para garantizar despliegues SDN más seguros y aprovechar plenamente su potencial en términos de eficiencia y escalabilidad. Esto por supuesto implica generar propuestas enmarcadas en frameworks, modelos, metodologías, arquitecturas y mecanismos que aborden conceptos forenses combinado con manejo de eventos, considerando las particularidades de las SDN y de esta manera brinden una respuesta efectiva a los incidentes de ciberseguridad en estos entornos altamente dinámicos.

Tomando en cuenta lo expuesto, la motivación de esta tesis doctoral se origina en un profundo interés por fortalecer la gestión de la ciberseguridad de las SDN, poniendo especial énfasis en la integración de procesos de preparación forense con los de respuesta a incidentes. Esto desde luego, tiene un impulso subyacente que se fundamenta en la optimización de la información cuando ocurre un incidente de ciberseguridad, con el propósito de que los datos resultantes sean aprovechados tanto en contextos técnicos como legales. En consecuencia, la meta es lograr una mejora continua en estas redes, asegurando un manejo eficiente de la información cuando se presenta un evento de ciberseguridad manteniendo una correcta vinculación de procesos forenses y de respuesta a incidentes.

1.2 Objetivos

La sección anterior resume los principales problemas de ciberseguridad detectados en la arquitectura SDN, derivados del desacoplamiento de sus planos. De igual manera se expone la necesidad de cubrir aspectos fundamentales organizacionales no técnicos, entre ellos, los de índole jurídica. En este contexto, se identifica una gestión fragmentada de la entrega de evidencias y la resolución de incidentes de ciberseguridad, en el ámbito de las SDN. En respuesta a esta problemática, surge la necesidad de abordar de manera holística los conceptos de gestión de incidentes, integrándolos con los procesos forenses. En consecuencia, el objetivo principal de esta tesis doctoral es contribuir al avance del conocimiento y a la mejora de la gestión tanto de los procesos forenses digitales como del tratamiento de incidentes en las SDN. Para alcanzar el objetivo principal se han establecido los siguientes objetivos específicos:

1. Realizar un análisis exhaustivo del estado del arte que abarque de manera estructurada el conocimiento existente respecto a la ciberseguridad en las SDN, tratando las principales preocupaciones que requieren atención por parte de la Academia, las propuestas existentes hasta el momento para sortear los inconvenientes y la presentación de los desafíos que merecen atención.
2. Proponer un framework que conjugue la ciencia forense digital con la gestión de procesos de respuesta a incidentes de ciberseguridad en SDN, considerando normativas enmarcadas por organismos internacionales como la Organización Internacional de Normalización (ISO, por sus siglas en inglés) y el Instituto Nacional de Estándares y Tecnología (NIST,

por sus siglas en inglés).

3. Desarrollar una arquitectura pragmática basada en el framework a proponer, en la cual se detallen los componentes, módulos y mecanismos funcionales, las fuentes de información y los roles de los interesados durante los procesos de respuesta a incidentes y los procesos de preparación forense, así como las interacciones existentes.
4. Proponer un modelo para el de filtrado de eventos de ciberseguridad basado en tecnologías de inteligencia artificial. Esto comprende, entre otras cosas, generar un conjunto de datos propietario obtenido del tráfico de la interfaz sur, haciendo uso del proceso ETL (*extract*, *transform*, *load*).
5. Proponer un modelo para preservar la integridad de la información relevante de un evento de ciberseguridad de implementaciones SDN, basado en tecnologías de ledger distribuidos permisionados .
6. Definir un ambiente de pruebas para validar los modelos propuestos en la presente tesis doctoral a través del despliegue de entornos virtualizados controlados con herramientas de código libre usados tanto en la industria como en la Academia.

1.3 Metodología de Trabajo

La investigación llevada a cabo en la presente tesis doctoral consta de dos enfoques fundamentales: documental y experimental. El primero se basa en la descripción detallada de la documentación recopilada dentro del contexto de ciberseguridad de las SDN. Para ello, se revisaron exhaustivamente los fundamentos teóricos de las SDN y su evolución dentro del ámbito de ciberseguridad. De igual manera, paulatinamente, se fue profundizando en la documentación relativa a la ciencia forense y respuesta a incidentes en el contexto SDN. A partir de dicha revisión, surgieron las preguntas de investigación, mismas que derivaron en la búsqueda de propuestas desarrolladas tanto por la Academia como por la industria, a fin de identificar claramente los problemas y retos que no han sido tratados. Posteriormente, se establecieron las estrategias de búsqueda y criterios de selección para encaminar las preguntas de la investigación (RQ, por sus siglas en inglés). Estos hallazgos permitieron definir un conjunto de propuestas que contribuyen al actual estado del arte y que son descritas en detalle en los capítulos 3, 4 y 5 .

En lo que refiere al enfoque experimental, en esta tesis doctoral se validan y discuten las propuestas planteadas como parte de las contribuciones presentadas, proporcionando escenarios y generando ambientes controlados de prueba. De este modo, se despliega una serie de recursos tecnológicos que permite observar el comportamiento de las SDN bajo determinados escenarios que pueden afectar su ciberseguridad, así como el accionar práctico de las propuestas.

1.3.1 Enfoque documental

Como se explicó anteriormente, la presente investigación tiene un componente documental. En este sentido, esta tesis doctoral, se apalanca en una revisión sistemática de la literatura que expone el estado del arte. Dicha revisión sistemática, permite identificar evaluar e

interpretar los trabajos de diferentes investigadores, académicos, la industria y profesionales de un determinado campo, que permiten conducir una pregunta de investigación específica, imparcialmente (Keele, 2007). En la Figura 1.1, se representan gráficamente las fases seguidas para llevar a cabo el proceso de revisión sistemática de la literatura en esta tesis doctoral.

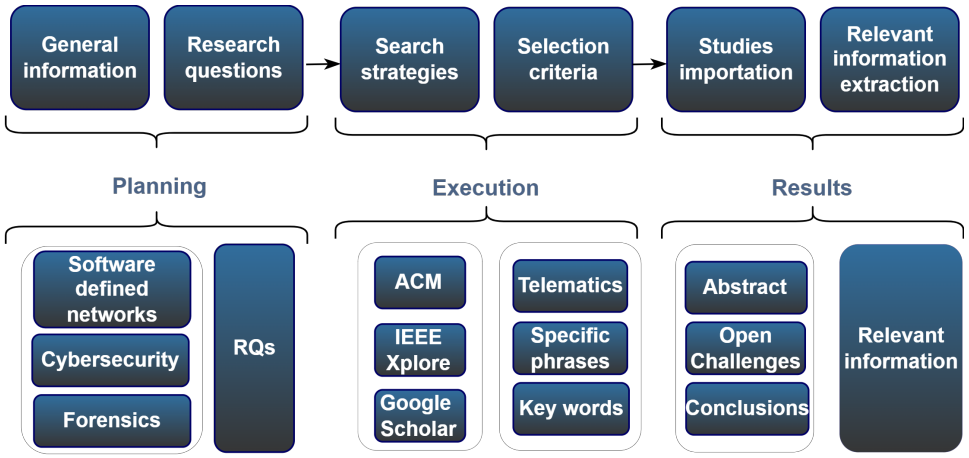

Figura 1.1: Fases de la revisión sistemática de la literatura

Objetivo de la revisión sistemática y preguntas de investigación

El principal objetivo de la revisión sistemática de la literatura llevada a cabo es proporcionar una visión completa y actualizada de los conocimientos, incluyendo la identificación de patrones, tendencias, incongruencias y vacíos relacionados con la ciberseguridad de las SDN. Esta revisión por lo tanto nos permite cubrir de mejor manera las preguntas de investigación planteadas en esta tesis doctoral. En este sentido, a continuación, se detallan claramente las preguntas de investigación formuladas.

RQ1: ¿Qué desafíos de ciberseguridad en SDN aún no han sido tratados?

RQ2: ¿Cómo aportan las ciencias forenses digitales a una gestión de ciberseguridad en SDN?

RQ3: ¿Cuáles son los puntos críticos que necesitan ser optimizados para mejorar la gestión de incidentes de ciberseguridad y preparación forense en SDN?

Estrategias de búsqueda y criterios de selección

Dado que la organización coherente del conocimiento dentro del análisis del estado del arte se considera una herramienta importante para guiar la futura investigación, en la presente tesis doctoral se establecieron estrategias de búsqueda y criterios de selección de los recursos de información, que son detallados a continuación.

Por una parte, para encontrar información relevante relacionada con los fundamentos del paradigma SDN y el manejo de la ciberseguridad en este tipo de redes, se utilizaron herramientas de apoyo que han permitido extraer documentación de bases de datos, motores de búsqueda o bibliotecas digitales, tales como: IEEE Xplore, Scopus, Springer, ACM Digital Library, Google Scholar, entre otras. Con el inicio de esta búsqueda también se pudo aprovechar información

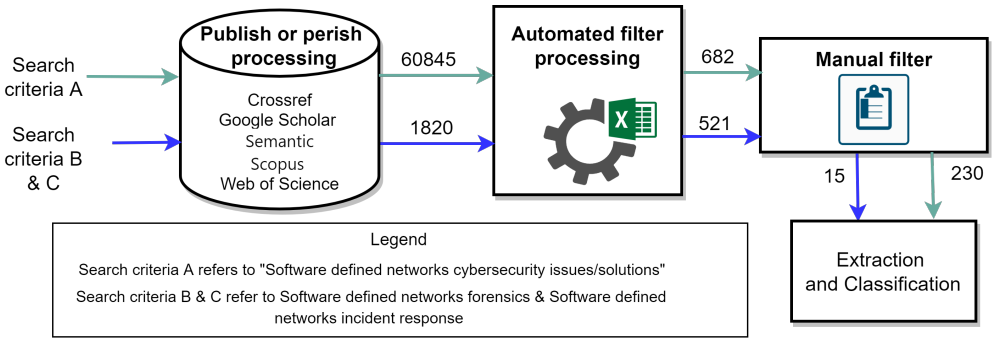

Figura 1.2: Proceso de selección de artículos

expuesta por los diferentes entes de estandarización, consorcios y organismos de la industria de telecomunicaciones, entre los cuales destaca la ONF.

Vista la amplitud del concepto tanto de SDN como de ciberseguridad, el siguiente paso fue realizar búsquedas específicas sobre problemas y soluciones de ciberseguridad para la arquitectura SDN. Posteriormente, dado que el objetivo principal de esta tesis doctoral es contribuir al avance del conocimiento y a la mejora de la gestión tanto de los procesos forenses digitales como del tratamiento de incidentes en las SDN, se realizó la búsqueda de esta temática, basada en palabras clave o frases específicas.

De este modo, la estrategia de búsqueda exploró documentación compuesta por términos como: "*Software defined networks cybersecurity issues*", "*Software defined networks cybersecurity solutions*", "*Software defined networks forensics*", "*Software defined networks incident response*". Posterior a las consultas, se obtuvieron varios recursos publicados en revistas y congresos. De la información provista, en una primer análisis se descartaron documentos que no se centraban en problemas o soluciones de la arquitectura SDN. Adicionalmente, en un segundo análisis, se han descartado artículos que no estaban relacionados con modelos de gestión de ciberseguridad desde enfoques forenses o de gestión de ciberseguridad para la arquitectura SDN.

Posteriormente, a partir de la información provista por estos recursos, se consideraron, mayoritariamente, artículos de revistas de alto impacto y congresos publicados entre los años 2016 y 2024. No obstante, tampoco se ha descartado información relevante de años pasados, considerando su influencia en el estado del arte. Principalmente, se han utilizado recursos cuyo resumen inicial, introducción, oportunidades de investigación y conclusiones han revelado una gran calidad de la información. También se han considerado fuertemente la reputación académica de los autores y la relevancia de los estudios, basados en las citaciones que han tenido y por quienes han sido citados.

Importación de estudios y extracción de información

Una vez que se han establecido las estrategias de búsqueda y criterios de selección de la documentación se realizó el proceso de importación de estudios y extracción de información,

usando herramientas como Publish or Perish [1]. En la Figura 1.2 se detalla el proceso de revisión de los artículos, documentos y entregables que han sido seleccionados.

1.3.2 Enfoque experimental

El enfoque experimental de esta tesis doctoral se basa en la observación minuciosa de muestras de datos y comportamientos de la SDN en situaciones de ciberseguridad en conjunto con un proceso iterativo de prueba y error, principalmente durante el análisis de herramientas de código libre y la exploración activa de nuevas tecnologías. Este enfoque se emplea para validar y debatir las propuestas presentadas, así como para generar escenarios de prueba controlados.

Para ello, se despliegan recursos tecnológicos que permiten simular condiciones específicas que podrían afectar la seguridad de las SDN. Estos escenarios controlados proporcionan un entorno propicio para realizar experimentos y recopilar datos relevantes. Esto implica la ejecución de diferentes configuraciones y condiciones, así como la observación detallada de los resultados obtenidos. Se analizan métricas clave relacionadas con la ciberseguridad de las SDN, como la detección de intrusiones, la resistencia a ataques cibernéticos y la integridad de los datos durante fases de almacenamiento. A través de este enfoque, se busca identificar vulnerabilidades, mejorar el diseño de la propuesta y la implementación de procesos forenses y de respuesta a incidentes, con el objetivo de fortalecer la seguridad y robustez de SDN en entornos prácticos.

1.4 Estructura de la tesis

Esta tesis doctoral está estructurada por capítulos, los cuales presentan los conceptos teóricos, las propuestas y sus evaluaciones; así también los principales hallazgos realizados a lo largo del trabajo de investigación. A continuación, se resume la estructura de la memoria de la presente tesis.

Capítulo 1, *Introducción*

Este capítulo contextualiza el área de investigación desarrollada, detallando los antecedentes de la problemática existente en los temas relativos a la ciberseguridad en la arquitectura de las SDN y el manejo de ésta desde un enfoque forense, los objetivos que se desean alcanzar y las contribuciones realizadas. De igual manera, se describe la metodología seguida durante la investigación con el fin de articular de mejor manera el conocimiento en el ámbito de ciberseguridad de las SDN con enfoque forense.

Capítulo 2, *Estado del Arte*

Este capítulo ofrece una contribución detallada sobre la ciberseguridad de las SDN. Inicia con la definición del paradigma, la presentación de los varios problemas de ciberseguridad de su arquitectura y las principales soluciones propuestas, resumiéndolo a través de tablas que incluyen las categorías de la metodología de modelado de amenazas *STRIDE*. También se realiza una discusión y se detallan los retos abiertos detectados, conducidos desde lo general

[1] https://harzing.com/resources/publish-or-perish

a lo particular.De este modo, se llega a determinar la necesidad de una visión holística para gestión de los incientes de ciberseguridad considerando la disciplina forense digital.

Capítulo 3, *Framework y Arquitectura para la Preparación Forense y Respuesta a Incidentes de Ciberseguridad en SDN*

Este capítulo aborda dos contribuciones de la tesis doctoral. Comienza con una introducción que establece el contexto general, seguido de una exploración de los fundamentos en el ámbito forense y de respuesta a incidentes. Además, se examinan los avances relativos a la ciencia forense y de respuesta a incidentes en el contexto específico de las SDN, destacando las limitaciones y desafíos aún no resueltos y que son relevantes para la tesis. Posteriormente, se presenta un framework integral diseñado para la gestión de la ciberseguridad en SDN. Este framework aborda aspectos cruciales como la preparación forense digital y los procesos de respuesta a incidentes. Asimismo, se proporciona una descripción detallada de la arquitectura que acompaña a este framework, incluyendo las fuentes de información de eventos de ciberseguridad, los componentes funcionales, y los diversos interesados e interacciones implicados en el proceso.

Capítulo 4, *Modelos Prevalentes*

En este capítulo se describen los modelos propuestos tanto para el filtrado, adquisición y tratamiento de datos, como para la preservación de la integridad de la información relativa a un evento de ciberseguridad en SDN. Para ambos modelos se detallan las estrategias y tecnologías consideradas.

Capítulo 5, *Validación de las Propuestas*

En este capítulo se pormenoriza el desarrollo técnico de las contribuciones presentadas en el capítulo 4. El capítulo contiene la explicación del ambiente para la validación y la evaluación de los modelos relativos a inteligencia de filtrado,adquisición y tratamiento de datos y preservación de evidencia. Además, se especifican las herramientas tecnológicas y paquetes de software utilizados para el despliegue de la generación de tráfico, tecnologías de ledger distribuidos permisionados y manejo de datos. De igual manera, se presentan los resultados respecto al desempeño de cada uno de los modelos conforme sus parámetros de evaluación.

Capítulo 6, *Conclusiones, Futuras Líneas de Investigación y Divulgación de Resultados*

En este capítulo se describen en forma de conclusión los resultados de la presente investigación, resaltando los hallazgos y beneficios. Así también se especifica el cumplimiento de las acciones para alcanzar los objetivos planteados al inicio de la investigación. Finalmente, se presentan los trabajos futuros respecto a los temas no cubiertos en la presente tesis doctoral. Del mismo modo, se detallan las publicaciones realizadas referentes a la presente tesis doctoral, en las cuales consto como primer autor y trabajos en los cuales he colaborado.

Capítulo 2

ESTADO DEL ARTE

2.1 Introducción

Las SDN han emergido como una solución innovadora para abordar una variedad de desafíos que enfrentan las redes tradicionales, como la escalabilidad, flexibilidad y programabilidad. Sin embargo, a medida que las organizaciones adoptan esta arquitectura para mejorar sus infraestructuras, es crucial examinar en detalle las vulnerabilidades que podrían comprometer su seguridad.

Comprender los inconvenientes de ciberseguridad es crucial por varias razones. En primer lugar, permite conocer los puntos débiles de la arquitectura, lo cual sirve como guía para proponer e implementar nuevas soluciones. Así también, al profundizar en las vulnerabilidades de las SDN, se promueve una mayor conciencia sobre los posibles vectores de ataque y las técnicas utilizadas por los ciberdelincuentes para comprometer la seguridad de estas redes. Además, entender las vulnerabilidades específicas puede ayudar a evaluar y mitigar los riesgos asociados con la implementación y operación de SDN.

Por estas razones, este capítulo se adentra en las problemáticas de seguridad identificadas en SDN. Para ello, inicialmente, se exploran los fundamentos de las SDN, en el cual se detalla su origen y arquitectura. Luego, se exponen las principales preocupaciones de seguridad de la arquitectura SDN y las diferentes soluciones para la confidencialidad, integridad y disponibilidad de los elementos que la componen. Esto incluye una revisión pormenorizada de las vulnerabilidades asociadas con cada plano, las interfaces y los controladores, como componente central. Finalmente, se efectúa una discusión que permite observar los retos abiertos con oportunidades de investigación, determinando la necesidad de un enfoque holístico para la gestión de la ciberseguridad en SDN.

2.2 SDN: Fundamentos

La conectividad digital desempeña un papel esencial en la humanidad, pues con ella se ha obtenido mayor disponibilidad de información, lo cual acelera el progreso, la eficiencia y la innovación en diversas esferas de la sociedad. Sin embargo, este avance conlleva desafíos

significativos para las redes de datos, ya que son responsables de proporcionar las capacidades que respaldan una sociedad hiperconectada. En la actualidad, una amplia gama de servicios se encuentra alojada en la nube, lo que impone una mayor demanda a las redes tradicionales, mismas que han ido presentando cada vez menos facilidades relativas a la flexibilidad, programabilidad y administración.

Estas facilidades son de gran interés, especialmente para organizaciones con ambiciones de expansión, como proveedores de servicios de red y operadores de telecomunicaciones, que necesitan adaptarse a un entorno en constante evolución. Por una parte, la flexibilidad se ha vuelto crucial en este panorama, ya que las redes deben ser capaces de adaptarse y escalar de manera eficiente para satisfacer a los pedidos del entorno. Por otra parte, la programabilidad es esencial para permitir la implementación de nuevas características y servicios de manera ágil, sin depender de cambios costosos en la infraestructura. Mientras que la administración eficaz es clave para garantizar un rendimiento constante y una alta disponibilidad, al tiempo que se minimizan los costos operativos.

Con estos antecedentes, se ha hecho evidente la necesidad de explorar tecnologías y enfoques innovadores que permitan a las redes de datos evolucionar, por lo cual se ha planteado el paradigma SDN, el cual pretende brindar soluciones alternativas a las actuales limitaciones de las redes tradicionales. SDN tiene su concepción desde hace aproximadamente dos décadas. Es así como, entre 1990 y 2000, a medida que las redes crecían en tamaño y complejidad, aparecieron las redes activas, lo que aportó algunas funciones programables. Posteriormente entre 2001 y 2007 se introdujo la idea de que las redes deberían ser capaces de adaptarse a las necesidades cambiantes de las aplicaciones. En el año 2008, un equipo de investigadores de Stanford presentó una arquitectura llamada “OpenFlow”, que determinaría la separación de los planos de datos y control en conmutadores de red mediante interfaces abiertas. Sin embargo, no fue hasta el 2011 que SDN comenzó a tener notoriedad en el mundo de las telecomunicaciones. Esto debido a que en ese año Google anunció que usaba SDN en su centro de datos, lo que generó interés en entornos empresariales (Feamster et al., 2014).

Actualmente la ONF es la organización sin fines de lucro que se dedica a promover y avanzar en la adopción de SDN y de código abierto, liderando varios proyectos concernientes a este paradigma. La ONF fue fundada en 2011 por un grupo de empresas líderes en la industria de las telecomunicaciones y la tecnología de la información. La ONF está conformado por alrededor de 130 miembros en los que destacan empresas como Facebook, Google, Deutsche Telekom, Microsoft, Yahoo!, fabricantes de equipamiento, entre otros (Open Networking Foundation, 2016).

De acuerdo con la ONF, la arquitectura SDN está compuesta de tres planos o capas: plano de datos, plano de control y plano de aplicación (Open Networking Foundation, 2014). El plano de control es el eje central de la arquitectura SDN, con el controlador como cerebro centralizado, este plano toma decisiones sobre el manejo de la red. El plano de datos agrupa todos los elementos de red (NE, por sus siglas en inglés), como conmutadores y enrutadores. Estos elementos reciben instrucciones del plano de control para ejecutar las acciones necesarias de conmutación y enrutamiento para que los paquetes de datos fluyan por la red. Por último, en el plano de aplicación residen todas las aplicaciones y servicios que utilizan la SDN, pudiendo ser estos: políticas de red, calidad del servicio (QoS, por sus siglas en inglés), balanceo de

carga, etc. La actualización e intercambio de información entre los planos (datos, control, aplicación) se realiza mediante las interfaces: sur (SBI), norte (NBI) y este/oeste (E/WI). La interfaz sur enlaza el plano de control con el de datos. La interfaz norte, vincula el plano de control con el de aplicaciones. Finalmente, la interfaz este/oeste, interconecta controladores distribuidos. En la Figura 2.1 se grafica lo explicado respecto a la arquitectura de SDN.

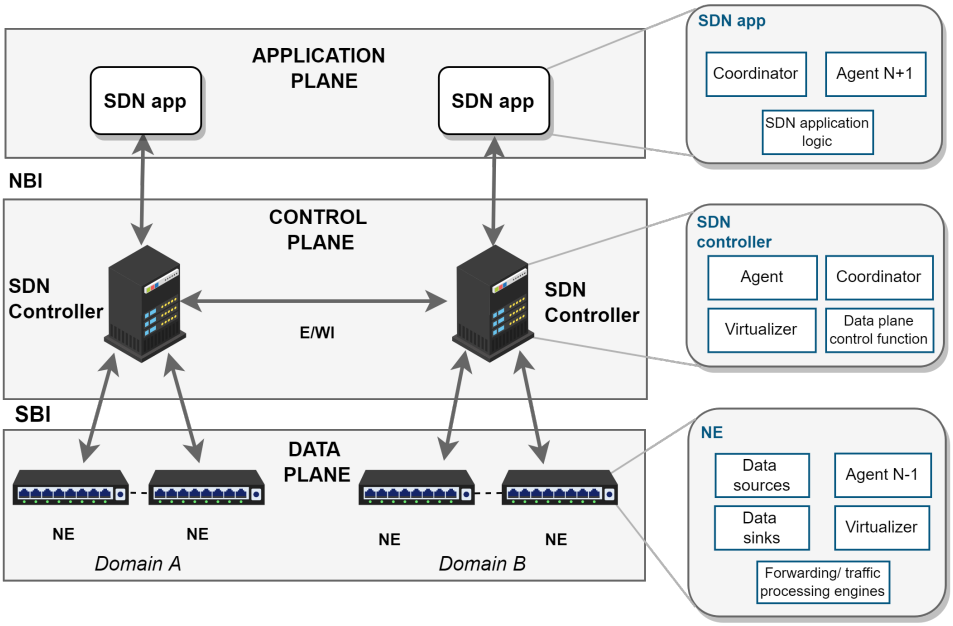

Figura 2.1: Arquitectura SDN. Adaptada de (Open Networking Foundation, 2014)

La idea fundamental detrás de esta arquitectura es separar la lógica de control, que reside en el controlador, de los elementos de red. Esto permite una gestión centralizada y programable de la red, lo que facilita la implementación de políticas de red más dinámicas. Asimismo, permite mantener una infinidad de aplicaciones o abstracciones de red que pueden ser desarrolladas por los proveedores de los controladores o por terceras partes.

En SDN la compartición de estas aplicaciones se realiza de manera más dinámica haciendo uso de nodos más simples, lo cual puede verse reflejado en la reducción de los costos de operación (Opex) y de gastos de capital (Capex), pues ya no es necesario contar con un fabricante de hardware o software especializado en una marca. Debido a las ventajas que presenta la arquitectura SDN, actualmente está siendo usada en el despliegue de servicios más rápidos, por ejemplo, en redes 5G y 6G, así como en otros paradigmas como el internet de las cosas (IoT) (Beshley et al., 2022). Sin embargo, la ONF reconoce que en una implementación a gran escala es necesario definir límites y políticas de seguridad en la compartición de información y recursos (Open Networking Foundation 2014), pues SDN al igual que muchos otros paradigmas tecnológicos presenta amenazas de ciberseguridad, varias de ellas debido a su arquitectura en capas.

De hecho, en un documento emitido por la ONF se enfatizan varias preocupaciones de ciberseguridad que están relacionados con la arquitectura SDN (Open Networking Foundation, 2015b), siendo estas:

- El control centralizado, dado que los atacantes tendrán un objetivo claro para desplegar sus acciones maliciosas, pudiendo manipular los servicios de red o tomar la administración de toda la SDN a través del controlador.
- La programabilidad, considerando que terceras partes son capaces de incluir sus aplicaciones lo cual abre la puerta a nuevos retos de seguridad desconocidos en redes tradicionales. Es así como los administradores de la red deberían tener mayor control sobre el aislamiento de tráfico y recursos entre varios inquilinos para evitar interferencias o uso indebido de recursos entre ellos. También se deberían tomar acciones sobre los procesos de autenticación y autorización durante el registro de aplicaciones en el controlador.
- El manejo de interfaces, dado que haciendo uso de los canales de comunicación entre los diferentes planos sin los debidos atributos de seguridad pueden existir acciones maliciosas que perjudiquen el desempeño de las SDN.
- La conexión entre dominios, ya que puede existir abusos por partes de los controladores interconectados si no se consideran los mecanismos para establecer relaciones de confianza y determinar el nivel de autorización.
- La integración de protocolos heredados, pues debido a la incipiencia de las SDN aún es necesario comprobar la compatibilidad y las capacidades de seguridad antes de implantar protocolos heredados, tales como NAT o BGP.

Como se observa estos son algunos de los inconvenientes de seguridad que presenta la arquitectura SDN. En los siguientes apartados se podrán observar de manera más detallada los inconvenientes, preocupaciones y vulnerabilidades relativos a las SDN.

2.3 CiberegurIdad en la arquitectura SDN: Aspectos clave y soluciones propuestas

A continuación, se revisan las principales preocupaciones de seguridad de la arquitectura SDN y las diferentes soluciones para la confidencialidad, integridad y disponibilidad de los elementos que componen la arquitectura SDN. Para realizar esta revisión, se han considerado todas las interfaces y planos que constituyen la arquitectura SDN, segmentándola en las siguientes agrupaciones:

- Plano de aplicación e interfaz norte
- Plano de control e interfaz este/oeste
- Plano de datos e interfaz sur (*stateless/stateful*)

Al finalizar la revisión de cada agrupación, se presentará una tabla resumen que incluye las categorías de la metodología de modelado de amenazas *STRIDE* (Microsoft, 2009), a través de la cual se analizará la contribución de cada propuesta. Cabe mencionar que se han elegido las categorías establecidas en la metodología *STRIDE* dada su madurez y adopción para evaluar aspectos de seguridad. Además, las categorías de esta metodología son ampliamente

reconocidas, ya que abarca los siguientes conceptos del mundo de la ciberseguridad: *spoofing, tampering, repudiation, information disclosure, denial of service* y *elevation of privileges*. A continuación, se explica en detalle cada una de las categorías:

Categoría		Consiste en
S	*Spoofing*	Suplantación de identidad de un usuario, o en este caso de un elemento o componente de la SDN
T	*Tampering*	Modificación de información con un propósito malicioso
R	*Repudiation*	Negación de haber realizado la acción sin que otras partes tengan forma de probar lo contrario
I	*Information disclosure*	Los datos pasan a estar disponibles para el usuario que se supone que no debería tenerlos
D	*Denial of service*	Sobrecargar un sistema o red con peticiones maliciosas, lo que resulta en una indisponibilidad de servicios para usuarios legítimos
E	*Elevation of privileges*	Un usuario sin privilegios obtiene de manera fraudulenta acceso privilegiado

Tabla 2.1: Descripción de la metodología *STRIDE*

2.3.1 Plano de aplicación e interfaz norte

El plano de aplicación proporciona un conjunto de programas esenciales (aplicaciones /abstracciones) para satisfacer las necesidades propias del sistema, con lo cual se puede generar o atender requerimientos del entorno SDN, a través del controlador. Actualmente, existe un gran espectro de aplicaciones, a saber: contrafuegos, algoritmos, políticas de enrutamiento, protocolos, etc.; las cuales pueden ser provistas a través del mismo desarrollador del controlador o por terceros.

El plano de aplicación está conformado principalmente por un agente, un coordinador y una lógica de aplicaciones SDN. Estos componentes permiten compartir o virtualizar los detalles de la red virtual que se exponen a las aplicaciones SDN, al tiempo que aíslan los servicios en ambientes multi-inquilino. En este punto, se torna necesario aclarar que el término de virtualización en las SDN aquí referido, no está anclado con la virtualización de funciones de red (NFV).

La comunicación entre la capa de aplicación y el controlador se efectúa mediante la NBI, la misma que no se encuentra estandarizada, por lo que cada controlador define su manera de comunicación. Dichas formas de comunicación en la NBI incluyen el uso de: *API RESTful*, lenguajes de programación o API especializadas (*ad-hoc*) (Vijay Tijare et al., 2016).

Hasta el momento, la *API REST* ha sido la más dominante en la industria en el desarrollo y uso de aplicaciones SDN, debido a que permite la comunicación entre aplicaciones y servicios web de manera eficiente. No obstante, también presenta ciertos desafíos en términos de seguridad relacionados con falta de autenticación y autorización, exposición de datos sensibles, inyección de *scripts* maliciosos, entre otros. Estas vulnerabilidades, podrían aprovecharse para facilitar ataques basados en la explotación de relaciones de confianza hacia el controlador y por consiguiente comprometer la integridad y seguridad de toda la red (Katt et al., 2018).

En la Figura 2.2 se grafican algunos inconvenientes de seguridad presentes en el plano de aplicación y la NBI.

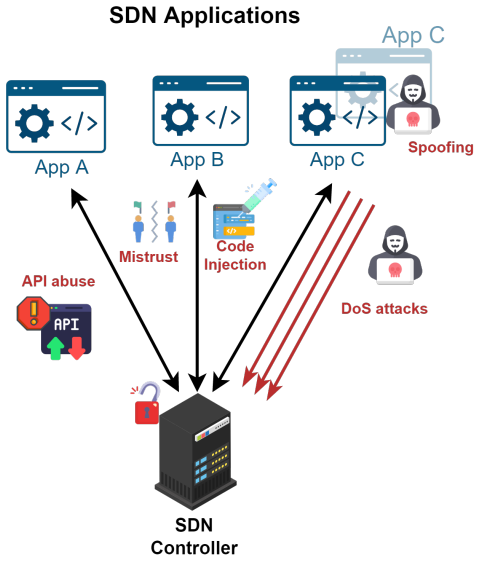

Figura 2.2: Inconvenientes de ciberseguridad asociados a la NBI y plano de aplicación

Debido a esto, la ONF no descarta la posibilidad de una estandarización, por lo que se encuentra trabajando en un API de código abierto para la NBI, que considere la participación de todos los interesados (desarrolladores de aplicaciones, proveedores de controladores, investigadores, etc.) y que a su vez contribuya de mejor manera en la generación de aplicaciones para las SDN (Raza et al., 2013).

Por su parte la Academia también ha tomado conciencia sobre estos inconvenientes en la NBI. De este modo los autores de (Du et al., 2018) proponen el uso del protocolo gRPC para la NBI. gRPC, es un protocolo de comunicación web basado en llamadas a procedimientos remotos (RPC) publicado por Google. Los autores mencionan que gRPC tiene ventajas significativas sobre REST. Por una parte, el procesamiento a gran velocidad en comparación con REST pues, gRPC puede procesar una llamada a procedimiento remoto con una sola conexión a través de la función de flujo multiplexado. Por otra parte, en términos de seguridad gRPC trabaja con HTTP/2.0 y este a su vez realiza la comunicación considerando TLS.

Así también, al existir aplicaciones SDN tan diversas y permitiéndose su implementación por terceros con gran variedad de reglas y métodos o APIs de comunicación, se pueden acarrear amenazas de seguridad, pues una aplicación maliciosa podría impersonalizar a una legítima, infiltrarse en el controlador de la red y ejecutar acciones que comprometan la configuración. Esto podría incluir la inserción de reglas falsas en el controlador, alterando drásticamente el comportamiento de toda la red (S. Y. Zhu et al., 2017).

Precisamente, respecto a este punto se han realizado algunos análisis de seguridad que identifican diversas vulnerabilidades que pueden dar cabida a ataques (P. Ahmad et al., 2018; Artmann et al., 2018; Bräuning et al., 2018; Chikhale et al., 2018; Jain et al., 2018; Sagare et al., 2018). Es así como en el análisis de un par de aplicaciones de encaminamiento y balanceo

de carga para tráfico en SDN, se pudieron detectar faltas asociadas con todas las categorías de *STRIDE*. Así también, los resultados de los análisis señalan que una aplicación SDN puede ser o volverse maliciosa dependiendo el componente de la aplicación que sea explotado (Sagare et al., 2018).

En este sentido, se pueden usar soluciones como *Indago* (Lee; Yoon et al., 2018), que realiza la detección de aplicaciones maliciosas de manera proactiva, mediante *Security-Sensitive Behavior Graphs* (SSBGs) y aprendizaje automático. De acuerdo con los autores su solución ofrece contramedidas para vectores de ataque relacionados a la manipulación de la información, impersonalización, asignación de permisos o divulgación de información o incluso que pueda llevar a una paralización del servicio de red. Generan un conjunto de datos propio extraído de los comportamientos de alrededor de 73 aplicaciones presentes en diversos controladores y entidades que disponibilizan información de aplicaciones malignas y benignas. Posterior a las pruebas realizadas obtuvieron un 96 % de exactitud en la tasa de detección.

De igual manera, existe *Shield* (Lee y Shin, 2016), un framework que analiza principalmente, el comportamiento de las aplicaciones mediante el *Control-Flow Graph* (CFG). Con esta propuesta los autores señalan que es posible identificar comportamientos maliciosos que podrían llevar a una modificación de parámetros internos en la red.

Algo a tomar en cuenta es que al tener múltiples aplicaciones provenientes de terceras partes se pueden presentar interferencias entre ellas, generando conflictos en las políticas de red o asignaciones de recursos indebidas a los diferentes clientes (Durairajan et al., 2014). Dichas interferencias pueden proceder de un ataque o pueden llevar a la ocurrencia de uno. Dados estos escenarios, existen propuestas como *MSAID* (Y. Li et al., 2019), que mediante algoritmos usados sobre el mismo código de las aplicaciones SDN es capaz de detectar interferencias. Así también se encuentra *SAIDE* (Hu; Yi et al., 2020), una propuesta para lograr la detección y la eliminación de las interferencias con el uso de modelos matemáticos.

Al mismo tiempo, la ausencia de mecanismos efectivos de control previo de autenticación y de control de accesos para las aplicaciones SDN, expone a la red a gestionar peticiones ilegítimas generando una elevación de consumo de los recursos y por ende el agotamiento de la red. Justamente Latah et al., 2020, realizaron pruebas de carga, estrés y ataques de DoS/DDoS desde la NBI, que llevaron a una disminución del rendimiento de los controladores analizados: POX (GitHub, 2020), *Ryu* (Ryu SDN Framework Community, 2017), *Floodligth* (Project Floodlight, 2016), ODL (Projects, 2019) y ONOS (Open Network Foundation, 2020).

Para dar solución a problemas de autorización en aplicaciones, se ha planteado tanto la utilización de controles de acceso de grano fino como de acceso de grano grueso. Por una parte, los controles de grano fino están relacionados con el monitoreo y control detallado, lo cual implica la supervisión del rendimiento de la red cuando se usa una aplicación, la evaluación del uso de ancho de banda por aplicación, monitoreo constante de la latencia, etc. Por otra parte, los controles de grano grueso se refieren a un nivel de supervisión más general, en lugar de centrarse en detalles técnicos específicos.

Los controles de grano grueso, dado que actúan en un nivel con menor detalle, pueden ser muy útiles en ambientes invariables donde el comportamiento habitual es casi predictivo. Dentro de esta categoría podemos hacer uso de mecanismos como RBAC, MAC, DAC, entre otros.

Mientras que los controles de acceso de grano fino, debido a que presentan mayor granularidad y detalle sobre los permisos de las aplicaciones, resultan ser muy útiles en ambientes con mayor dinamismo.

Considerando que por concepto la arquitectura SDN está diseñada para ser usada en ambientes multi-inquilino, multi-servicio, multi-proveedor o multi-dominio, el incorporar únicamente controles de acceso de grano grueso o muy rígidos, podría dar cabida a abusos en el uso de los permisos o a su vez ralentizar la transaccionalidad en la red. De hecho, los autores Ujcich et al., 2018, demuestran que uno de los mecanismos de control de acceso de grano grueso antes mencionado, no es suficiente para controlar ataques de integridad sobre el flujo de información, pues permite que se efectúe un ataque denominado *cross-app poisoning*, en el cual una aplicación accede al controlador y engaña a otras aplicaciones, para que éstas ejecuten acciones.

Tomando en cuenta lo indicado, las soluciones para problemas de autorización de aplicaciones SDN se están orientando a un control de accesos de grano fino. Mientras que, en lo referente a autenticación existen soluciones por cada controlador o con mecanismos propietarias independientes. Tomando en cuenta estos antecedentes, a continuación, se detallan los principales contribuciones en este dominio y que han aportado significativamente en el estado del arte. Es así como, los autores (Chang et al., 2019) proponen *MD-UCON*, un mecanismo de control de accesos para la interfaz norte, enfocado a escenarios multi-dominio. Esta propuesta usa mecanismos RBAC adaptados para ambientes dinámicos, aplicando un método *cross-domain role mapping* para soportar un control de accesos *cross-domain*. Los autores generaron esta propuesta, basados en un modelo denominado *UCON* (Park et al., 2002).

A través del uso de blockchain se ha propuesto a *BlockAS* (Hoang et al., 2019) para relacionar los controladores con las aplicaciones OF (participantes), usando identificadores en el proceso de autenticación y asignación de privilegios. Esta propuesta además de brindar autenticación, autorización y auditoría, cuenta con descentralización e inmutabilidad de la base de datos donde se almacena información de las aplicaciones, permisos y tokens. Inclusive permite el control de logs, en el cual se registra la actividad de todos los participantes, lo que le permite realizar un monitoreo.

Los autores Tseng; Pattaranantakul et al., 2017, plantean una solución independiente al controlador para evitar el abuso de permisos estáticos asignados a una API. Está compuesta de una extensión de seguridad en la interfaz norte, cuya función es reempaquetar los *built-in services* de dicha interfaz, para servir de "mediador" entre las aplicaciones y el controlador. Esta solución, además incluye un IDS específico del controlador, el cual guarda información sobre los permisos y registros de *accounting* de la aplicación y un motor de políticas de alto nivel, que predefine las políticas para cada aplicación. De este modo se pueden autenticar y autorizar las aplicaciones y adicionalmente verificar la legitimidad de las solicitudes de *accounting*, haciendo uso de autenticación basada en clave y en token.

Padekar et al., 2016, proponen el framework AEGIS, en el cual se mantiene un control de accesos dinámico, mediante la verificación y monitoreo del uso de la API en tiempo real. Los componentes de esta propuesta comprenden: *Data generator*, que identifica la lista de API a ser protegidas, para este propósito se realizan acciones de extracción de datos con el uso

de *Daikon*[1]. De igual manera se cuenta con un generador de reglas de seguridad, en el que se definen las reglas de acceso entre aplicaciones, API y sus entradas o salidas. Por último, cuenta con un motor de decisión, que utiliza *API hooking* para interceptar el comportamiento de las aplicaciones, aquí se revisan las entradas y salidas de cada API y se las contrasta con las reglas.

BEAM (Toshniwal et al., 2019), es una solución que asigna permisos a las aplicaciones de terceros. Está basado en el comportamiento de la red, el mismo que es tomado de métricas como *flow_injection_rate* o *packet_in_rate*, que después de ser analizadas por un IDS, incrementa o retira los permisos a las aplicaciones en tiempo de ejecución. *BEAM*, trabaja con los módulos: registration handler y policy engine. El primero, se encarga del registro y asignación de permisos iniciales a nuevas aplicaciones. El segundo, define políticas para incrementar o retirar los permisos de las aplicaciones y cuenta con dos bases de datos importantes: *policy store* y *mapping table*. BEAM, además usa módulos encargados de revisar la actividad de las aplicaciones apalancados en un IDS y los registros (*logs*).

Tseng; Naït-Abdesselam et al., 2018, preocupados por la inyección de aplicaciones maliciosas, DoS y abuso de la API, proponen una arquitectura denominada *SENAD*, que consta de controller agents, que basados en un modelo *publish/suscribe* permiten la interacción entre la capa de aplicación-controlador (APC) y la capa de datos-controlador (DPC). El *policy engine* brinda control de recursos de cada aplicación, así como el control de accesos para éstas. Application *sandbox* y *resource controller*, se encargan de aislar las aplicaciones y la entrega de recursos conforme lo requerido por el policy engine, respectivamente. Por último, cuenta con los módulos de autenticación y autorización, los cuales trabajan con la información intercambiada entre APC y DPC, manejando autenticación basada en contraseña y consultas de reglas al policy engine para la autorización.

Los autores Cui et al., 2018, proponen un enfoque denominado *application authentication system*, desplegado fuera del controlador, que permite autenticar, tener un historial de registros de operaciones no autorizadas y gestionar permisos de acceso, recursos, certificados de las aplicaciones, elementos de autorización, autenticación y cifrado, entre otros. De manera similar, en (Kim et al., 2017), se define una solución de seguridad como servicio, siendo una de sus prestaciones el mecanismo de autorización entre el controlador y las aplicaciones tomando como referencia la arquitectura del controlador *Floodligth*.

De igual manera, Banse et al., 2015 proponen un framework NBI basado en web y que está influenciado por REST. Dicho esto, se implementa una API denominada *REST-like* ya que es parcialmente *stateless*[2], es decir que en un punto guarda información que utiliza para el registro, la autenticación y la autorización de las aplicaciones; incorpora gestión de confianza, control de accesos y comunicación cifrada con el uso de certificados TLS. Tseng; Zonghua Zhang et al., 2016, se han basado en la propuesta anterior para exploran más funciones de *REST-like* que pueden ser aprovechadas para brindar seguridad a las aplicaciones.

Los autores Natanzi et al., 2018, presentan una arquitectura que únicamente brinda información a aplicaciones de terceros que sean confiables, para ello hace uso de la interfaz norte con

[1]http://swmath.org/software/4319
[2]https://restfulapi.net/statelessness/

la implementación de firma digital NSS y el algoritmo de cifrado NTRU. De igual manera, Hu; Zhen Zhang et al., 2021, proponen un framework que contiene, principalmente, dos módulos: *Permissions detection engine*, donde se identifica la legalidad de los permisos de las aplicaciones y *Registration authorization engine*, en que se efectúa tanto el registro como la autorización de la aplicación con el algoritmo NTRU para evitar ataques de *eavesdropping* o de *tampering*.

Haciendo uso de técnicas de *fuzzing*, Shou, 2021, propone un framework para probar las aplicaciones SDN *stateful*, mediante gráficos de propiedades. Logra su objetivo a través de la recuperación de los estados basándose en instantáneas.

En la Tabla 2.2 se resumen las soluciones propuestas a los principales preocupaciones de seguridad para la capa de aplicación y la interfaz norte. En la tabla se podrá encontrar la referencia de la contribución, los problemas que originaron la investigación, los principales mecanismos utilizados en el planteamiento de la solución y finalmente, una tabulación referenciada con las categorías de *STRIDE*.

Tabla 2.2: Resumen de contribuciones del plano de aplicación y la interfaz norte

Ref.	**Principales preocupaciones abarcadas**	**Principales planteamientos de solución**	**Análisis de contribución a la literatura**					
			S	**T**	**R**	**I**	**D**	**E**
(Lee; Yoon et al., 2018)	Aplicaciones maliciosas	SSBG +Aprendizaje automático	✓	✓		✓	✓	✓
(Lee y Shin, 2016)	Aplicaciones maliciosas	Control-Flow Graph (CFG)		✓				
(Chang et al., 2019)	Falta de control de accesos cross-domain en escenarios multidominio	RBAC modificado + método cross-domain role mapping						✓
(Hoang et al., 2019)	Modificación ilegal de información de aplicaciones Falta de control de la asignación de permisos	Blockchain	✓	✓	✓	✓	✓	✓
(Tseng; Pattaranantakul et al., 2017)	Abuso de API	Extensión de seguridad en la interfaz norte + IDS + motor de políticas	✓		✓	✓		✓
(Padekar et al., 2016)	Falta de control de acceso dinámico	API hooking + Generador de reglas de seguridad						✓
(Toshniwal et al., 2019)	Falta de permisos de acceso inicial Actualización de los permisos de las aplicaciones	Método basado en el comportamiento + IDS			✓			✓
(Tseng; Naït-Abdesselam et al., 2018)	Abuso de API, DoS, aplicaciones maliciosas	Modelo publish/subscribe	✓	✓			✓	✓
(Cui et al., 2018)	Falta de mecanismos de autenticación entre aplicaciones y controlador	Mecanismo independiente	✓		✓			✓
(Kim et al., 2017)	Falta de autenticación entre aplicaciones y controlador	Autenticación basada en Floodlight	✓					
(Banse et al., 2015) (Tseng; Zonghua Zhang et al., 2016)	Comunicación no cifrada entre aplicaciones y controlador Falta de gestión de la fiabilidad Falta de control de acceso	Uso de funciones “REST-like”	✓	✓	✓	✓		✓
(Natanzi et al., 2018)	Falta de autenticación de aplicaciones de terceros	Algoritmo de cifrado NTRU y firmas digitales NSS	✓	✓				
(Hu et al., 2021)	Aplicaciones maliciosas	Algoritmo de cifrado NTRU	✓	✓				
(Shou, 2021)	Aplicaciones stateful maliciosas	Fuzzing	✓	✓				

2.3.2 Plano de control e interfaz este/oeste

El control de las SDN se encuentra lógicamente centralizado en esta capa a través de “una caja negra” denominada controlador o *Network Operating System* (NOS), el cual gestiona requerimientos efectuados desde el plano de datos. El controlador cuenta con la información referente a la topología, la generación de recursos para las aplicaciones, las estadísticas e inventario, entre otros.

Aunque el controlador es una caja negra la ONF define algunos componentes, principalmente: un coordinador, un agente, un virtualizador y la función de control del plano de datos (DPCF). El coordinador es un componente funcional del controlador SDN para brindar gestión en ambientes de cliente-servidor. En SDN, la entidad que soporta la instancia del modelo de información y la política en la interfaz norte se denomina virtualizador. El virtualizador traduce y valida las solicitudes de cada cliente para la entrega de los recursos subyacentes y transmite los resultados a la DPCF y a la interfaz sur. El componente DPCF es el propietario de los recursos subordinados por lo tanto es capaz de gestionarlos. Un agente en un controlador SDN representa los recursos y acciones disponibles para un cliente o aplicación del controlador SDN (Open Networking Foundation, 2015a).

Actualmente existen más de treinta controladores disponibles, los cuales manejan sus propios lenguajes de programación e interfaces, muchos son de código abierto y otros son propietarios. A los controladores se los puede clasificar como de arquitectura centralizada y de arquitectura distribuida (Karakus et al., 2017a; Kreutz et al., 2015; L. Zhu et al., 2019).

Conforme la literatura existe una subclasificación de las arquitecturas distribuidas pudiendo ser estas planas o jerárquicas. Esta subclasificación está basada en las responsabilidades que posee cada controlador dentro de la arquitectura (L. Zhu et al., 2019). En las arquitecturas distribuidas planas, todos los controladores poseen las mismas aplicaciones y responsabilidades, mientras que en las arquitecturas distribuidas jerárquicas existe un controlador robusto (*root*) que maneja todas las aplicaciones y varios controladores con menos aplicaciones y por lo tanto menos responsabilidades (Karakus et al., 2017b). Sin embargo, el modo de comunicación entre los planos ya sea en arquitecturas distribuidas planas o jerárquicas, es igual. Por lo que, para temas de cibeseguridad, esta diferenciación es poco relevante, por lo tanto en el presente documento se referenciará a los controladores de manera genérica como centralizados o distribuidos.

Las arquitecturas centralizadas utilizan un único controlador para toda la red para facilitar su gestión. Por lo general, este tipo de arquitectura se utiliza en implementaciones en las que la demanda de rendimiento es baja. Sin embargo, depender de una única entidad de control podría crear un único punto álgido en la gestión de la red. Por ejemplo, en caso de picos de tráfico, tener un único controlador crearía cuellos de botella para las solicitudes entrantes, afectando así al tiempo de respuesta. Además, desde el punto de vista de la ciberseguridad, esta arquitectura puede ser más propensa a los ataques de denegación de servicios (Dhawan et al., 2015). Por otro lado, se cuenta con las arquitecturas distribuidas que hacen uso de múltiples controladores en redes multi-dominio o heterogéneas (Y. Zhang et al., 2018). Los controladores distribuidos, son usados en implementaciones a gran escala principalmente por operadores de telecomunicaciones, por ejemplo en redes de área extendida (WAN).

Con controladores distribuidos se busca mejorar las condiciones relativas a la escalabilidad, rendimiento, latencia y resiliencia de la red; siendo esta última condición muy útil ante la presencia de ataques de DoS. La resiliencia es manejada por los controladores mediante el uso de los mecanismos de tolerancia a fallas. De esta manera se observa que el controlador *Hyperflow* (Tootoonchian et al., 2010), hace uso de la propiedad de tolerancia de partición de *WheelFS system*. Por su parte *Ravana* (Katta; H. Zhang et al., 2015) y *Pane* (Ferguson et al., 2013), en el proceso de tolerancia a fallas y replicación usan *ZooKeeper* (Hunt et al., 2010).

Así también ONOS y ODL han trabajado con el algoritmo de concenso *Raft*, para elección de un controlador líder (Ongaro et al., 2014).

Aunque estos mecanismos de tolerancia son los más reconocidos, también existen propuestas como la emitida por los autores Macedo et al., 2016, en la cual se hace uso de métodos de anti-entropía a través del protocolo *Gossip*, logrando la detección de controladores con sobrecarga de tráfico malicioso para posteriormente elegir un controlador robusto que lidere el clustering ante ataques de DDoS en las SDN.

La comunicación entre controladores de arquitecturas distribuidas se efectúa mediante la interfaz este/oeste (E/WI), que al igual que la interfaz norte no está estandarizada, por lo que cada controlador propone una interfaz. Las interfaces orientadas al este, proporcionan comunicación entre los controladores SDN en diferentes dominios administrativos de ambientes distribuidos, mientras que las interfaces orientadas al oeste, interconectan redes tradicionales con arquitecturas (Latif et al., 2019). Aunque los autores Kreutz et al., 2015, indicaron que la interfaz este se encarga de conectar SDN a redes tradicionales y la interfaz oeste se encarga de interconectar SDN a SDN. Este punto en particular hace que la interfaz este/oeste sea de gran importancia.

Se ha observado que uno de los principales inconvenientes de ciberseguridad en la E/WI es la falta de verificación de procedencia de la información compartida entre los controladores durante el proceso de actualización de descubrimiento de topología, lo cual puede ser aprovechado por los atacantes para dificultar el normal desempeño de la red (S. Khan; Gani; Abdul Wahab et al., 2017). En este contexto, la *Internet Engineering Task Force* (IETF) inicialmente desarrolló *SDNi,* un protocolo diseñado para la comunicación en la E/WI (Yin et al., 2012). Aunque dicho protocolo actualmente se encuentra en estado expirado y a pesar de que presenta una vulnerabilidad que permite la inyección de código SQL (NIST, 2018a), continúa siendo relevante para la comunidad científica, pues se ha observado su uso en propuestas de interconexión en ambientes distribuidos (Allybokus et al., 2018; Benamrane et al., 2017; P. Lin et al., 2014; Phemius et al., 2014). Desafortunadamente, en las propuestas referidas no se han especificado factores de ciberseguridad.

En contrapartida, existen autores que si han considerado elementos de ciberseguridad. Es así como Yu et al., 2020, proponen una solución multi-controlador, haciendo uso de una capa encargada de concentrar el control de acceso y las decisiones de encaminamiento. Así también los autores Benamrane et al., 2017, implementan servicios de contrafuegos distribuidos (DFS), servicio de balanceador de carga distribuido (DLBS) y aseguramiento del canal mediante SSL. Por otra parte, hay soluciones para asegurar la comunicación en la E/WI que trabajan con *identity-based cryptography* (IBC) (J. H. Lam et al., 2015), o que brindan un canal protegido mediante un algoritmo de cifrado ECC (Natanzi et al., 2017).

Aún cuando en los controladores SDN se observan mecanismos para la recuperación ante fallas, los atacantes no dan tregua y presentan múltiples acciones para desestabilizar la red. Es por ello que existen entidades dedicadas al análisis de vulnerabilidades y ciberseguridad que tratan de orientar a la industria y a la Academia hacia la búsqueda de nuevas y mejores alternativas de ciberseguridad, pudiendo citar a *Red Hat, OWASP*[3] o a la Academia.

[3]https://owasp.org/

Tal es la aportación por parte de estas entidades que los autores Arbettu et al., 2016, realizan un análisis de ciberseguridad de los controladores: ODL, ONOS, *Rosemary* y *Ryu*, observando las medidas de ciberseguridad con las que cuenta cada uno. Posterior a este análisis, los autores determinaron que, si bien no existe controlador completamente seguro, los controladores que presentan mejores contramedidas ante ataques son ODL seguido de ONOS. Basándonos en esta premisa y debido a la considerable presencia de los controladores ONOS y ODL en el estado del arte, los siguientes párrafos tendrán un enfoque hacia ellos. Siendo así, a continuación se presentan los principales aspectos de ciberseguridad de ONOS y ODL.

ODL, tiene implementado módulos AAA y *secure network bootstrapping infrastructure* para evitar problemas de autenticación, repudio o impersonalización ya sea de usuarios o aplicaciones. Para evitar impersonalización en *host tracking*, ODL usa parámetros tales como *MAC address*, IP y *location address*, VLAN ID dentro del *device manage* en lugar de solo usar la *MAC address*. A través de la función de canal seguro unificado obtiene comunicación segura con soporte TLS/DTLS (Scott-Hayward, 2017).

ONOS, está provisto de un *Security Mode-ONOS* el cual tiene mecanismos de control de acceso de grano fino tanto para las aplicaciones como para los usuarios, protegiendo al controlador de ataques de elevación de privilegios (ONOS, 2020). También cuenta con un servicio de auditoría de ciberseguridad que evita el repudio. No refiere un mecanismo que evite ataques de impersonalización en el proceso de *host tracking*.

En nuevos estudios se identifican vulnerabilidades para ODL y ONOS relativas al consumo de recursos, autenticación, integridad, confidencialidad, y otras que aprovechan el diseño de *Network Management Datastore Architecture* (NMDA)[4] (Dixit et al., 2018; Secci et al., 2019). A continuación, se detallan de manera ampliada los principales inconvenientes encontrados.

Respecto a ODL, la mayoría de vulnerabilidades detectadas derivan en un incremento de uso de recursos y por consiguiente en un ataque de DoS (NIST, 2018e; NIST, 2017a; Bidaj et al., 2016). Igualmente, se exponen vulnerabilidades que permiten que se efectúen acciones de spoofing sobre la topología de la red (NIST, 2017f; NIST, 2017g). En cuanto a la integridad y la confidencialidad en ODL se han hallado vulnerabilidades derivadas de un *bug* de *Netconf TCP service* (NIST, 2014), que al momento se encuentra en reevaluación por parte de la NIST.

Dichas vulnerabilidades en conjunto con un ataque *XML eXternal entity* (XXE), puede permitir la inclusión de archivos locales y remotos (OWASP, 2020). A esto se agrega una falta de limpieza automática del caché después del cambio de contraseña en el módulo AAA de ODL, con lo cual un atacante podría aprovechar esta vulnerabilidad para modificar archivos o información del sistema (NIST, 2017i). Del mismo modo es preciso indicar que, aunque el mecanismo de *Defense4all* ya no tiene actividad, este fue quebrantado por parte de usuarios autenticados remotamente (NIST, 2017e). En lo concerniente a los clústers de ODL, se debe tener principal atención en el intercambio de mensajes entre las instancias dado que los mismos carecen de cifrado y autenticación en una de sus versiones (OpenDaylight, 2016).

Por otra parte, en ONOS se han identificado vulnerabilidades relacionadas con DoS. Una está enmarcada en una falta de límites en la asignación de los recursos y otra está vinculada al cierre

[4]https://www.rfc-editor.org/rfc/rfc8342.txt

inesperado en el componente OVSDB (NIST, 2017d; NIST, 2018c). Así también, un mal manejo de excepciones cuando se registran jumbo frames, puede llevar al apagado del controlador (NIST, 2017b). ONOS presenta vulnerabilidades que afectan la confidencialidad e integridad, relacionadas con XXE y que pueden ser explotadas usando *OpenConfig Terminal Device* o cuando no se usan mecanismos de autenticación (NIST, 2018d; NIST, 2018b). Adicionalmente, en la interfaz de usuario del controlador se han identificado vulnerabilidades relacionadas a autenticación, con lo cual es posible realizar acciones sobre la carga de aplicaciones o tener acceso a la información de la topología de la red (NIST, 2017h; NIST, 2017c). Últimamente, se descubrió una vulnerabilidad que compromete la integridad, confidencialidad y disponibilidad de la red debido al mal manejo de los caracteres de las comillas inversas que se pueden utilizar en un comando *shell*. (NIST, 2019).

Además de los problemas de seguridad mencionados, se deben considerar los ataques derivados de las vulnerabilidades de día cero, los cuales pueden llegar a afectar a los controladores y por ende a toda la arquitectura SDN. Estos ataques pueden ser tratados mediante la implementación de IDS. Existen dos tipos de IDS: *Signature-based Intrusion Detection System (SIDS)* y *Anomaly-based Intrusion Detection System* (AIDS) (Khraisat et al., 2019). De estos dos, SIDS es menos eficiente en la detección de vulnerabilidades de día cero, debido a que su funcionalidad es reactiva, es decir es necesario que se conozca el ataque para que éste sea incluido en la firma. Mientras que los AIDS tratan de brindar soluciones de manera proactiva con el uso de métodos basados en estadísticas o en conocimientos y en algoritmos de inteligencia artificial. En este sentido, a continuación se exponen algunas contribuciones presentes en el estado del arte para dar soluciones a problemas de ciberseguridad en controladores SDN.

Song et al., 2017 plantean un IDS mediante el uso de aprendizaje automático y enrutamiento reactivo, compuesto de tres subsistemas. El primero se encarga de eliminar información irrelevante mediante el uso de estadísticas, análisis de tráfico e ingeniería de características. Con los datos filtrados, el segundo módulo realiza de la identificación de intrusos, mediante *Random Forest* (RF). Simultáneamente, en el tercer módulo se trabaja sobre información imprecisa a fin de lograr mayor exactitud en el filtrado. Para probar su propuesta usaron el conjunto de datos KDD'99 obteniendo resultados oscilantes entre el 98 y 99 % de exactitud. Aunque no especifican el vector de ataque analizado, indican los ataques que contiene el conjunto de datos elegido: *User to Root* (U2R), *Remote to Local* (R2L), DoS y *probe*.

De igual manera, Garg et al., 2019, con el uso de aprendizaje profundo proponen un framework para detección de anomalías de tráfico multimedia en SDN. La propuesta consta de dos módulos, siendo el primero el encargado de la parte de seguridad. Dicho módulo está enfocado a varios vectores de ataque como *hijacking, spoofing, malware*,etc. y trabaja con *Restricted Boltzmann Machine* (RBM), para reducir la dimensionalidad y *Support Vector Machine* (SVM), para clasificar las características de los flujos. El proceso de clasificación es llevado a cabo en la capa de aplicación. Para su propuesta usan los conjuntos de datos KDD'99, CMU y el de *Thapar Institute of Engineering & Technology* (TIET)[5], que mayoritariamente registra tráfico HTTP, produciendo una exactitud de 99.02 %. Mientras que, el segundo módulo se encarga de la calidad de entrega de la información para proveer calidad de la experiencia (QoE).

[5]https://www.thapar.edu/

Malik et al., 2020, presentan un mecanismo para la detección de intrusos en la capa de control. Para ello, aplican técnicas híbridas de aprendizaje profundo como *Convolutional Neural Network* (CNN) para extraer de características de los datos en bruto y *Long Short-Term Memory* (LSTM) para evitar el problema de desaparición del gradiente, el cual se presenta en las secuencias de conjuntos de datos amplios. Utilizan el conjunto de datos CICIDS2017 y obtuvieron una exactitud de 98.6 %. Esta solución está orientada al control de ataques de tipo: *port scan, cross site scripting o botnet*.

Los autores Tang et al., 2019 plantearon un proceso de detección de anomalías que trabaja con tres módulos y que se apalanca con el algoritmo denominado *Gated Recurrent Unit Recurrent Neural Network* (GRU- RNN). Para evaluar su propuesta utilizaron características de los conjuntos de datos: NSL-KDD y CICIDS2017 alcanzando un 89 % de exactitud en la detección de ataques de DoS, R2L, U2R o *probe*.

Manteniendo el mismo esquema que la Tabla 2.2, en la Tabla 2.3 se encuentra un resumen de las principales contribuciones respecto a soluciones de ciberseguridad para el plano de control y la E/WI. Dado que los ataques suelen desplegarse desde los planos de datos o de aplicaciones, las soluciones para los distintos problemas que ponen en peligro al controlador son visibles en estos planos.

Tabla 2.3: Resumen de contribuciones del plano de control e interfaz este/oeste

Ref.	**Principales preocupaciones abarcadas**	**Principales planteamientos de solución**	**Análisis de contribución a la literatura**					
			S	**T**	**R**	**I**	**D**	**E**
(Macedo et al., 2016)	Ataques DDoS	Protocolo Gossip					✓	
(Yu et al., 2020)	Comunicación multi-controlador	Método ACL	✓		✓			✓
(Benamrane et al., 2017)	Comunicación multi-controlador	DFS, DLBS y SSL	✓	✓	✓		✓	✓
(J. H. Lam et al., 2015)	Comunicación multi-controlador	IBC	✓					✓
(Natanzi et al., 2017)	Comunicación multi-controlador	Algoritmo de cifrado ECC	✓	✓		✓		
(Song et al., 2017)	Falta de detección de intrusiones y anomalías	Aprendizaje automático + enrutamiento reactivo					✓	
(Garg et al., 2019)	Falta de detección de intrusiones y anomalías	Aprendizaje profundo	✓				✓	
(Malik et al., 2020)	Falta de detección de intrusiones y anomalías	Aprendizaje profundo	✓	✓		✓		✓
(Tang et al., 2019)	Falta de detección de intrusiones y anomalías	Aprendizaje profundo					✓	

2.3.3 Plano de datos e interfaz sur

El plano de datos es la capa de la arquitectura SDN donde se concentran todos los elementos de red o de reenvío, tales como conmutadores, enrutadores, etc. Este plano interactúa directamente con los clientes de una red y responde a cada una de las peticiones realizadas, basándose en las políticas emitidas por el controlador.

El plano de datos contiene a los elementos de red, los cuales a su vez están compuestos de fuentes de datos, un agente, un virtualizador, disipadores de datos y motores de procesamiento de gestión de tráfico (Open Networking Foundation, 2015a). En estas designaciones, el agente del plano de datos es la entidad que ejecuta las instrucciones emitidas por el controlador SDN.

El plano de datos y el plano de control se comunican mediante la SBI, usando los protocolos tales como OpenFlow (Open Networking Foundation, 2018a), OVSDB (IETF, 2013), OpFlex Cisco, 2014, NETCONF (IETF, 2011), ForCes (ForCES, 2014), entre otros. De estos, el más extendido y estudiado actualmente es OpenFlow, ya considerado como un estándar según la ONF (Open Networking Foundation, 2018a), por lo que en el la revisión del estado del arte se hará mayor énfasis en el manejo del mismo.

OpenFlow es una propuesta presentada en el año 2008 por un grupo de investigadores de Stanford. La propuesta, liderada por Nick McKeown, inicia haciéndose preguntas sencillas, que consideraban que no eran cubiertas dentro de su ambiente universitario. De este modo, se interrogaban: ¿Cómo gestionaban los administradores de red los requerimientos de los investigadores cuando se necesitaban realizar experimentos en los campus universitarios?, ¿Qué funciones debería tener una red?, o dentro de las porciones de cada red asignada ¿Cómo controlaban los investigadores que no existan interrupciones en ambientes multi-inquilino? Una vez que comprendieron las respuestas a estas interrogantes, exploraron posibles soluciones (McKeown et al., 2008).

En su intento por hallar una nueva función de conmutación que les permita extender la programabilidad en los equipos de los campus universitarios, se encontraban con algunos inconvenientes; por ejemplo, observaron que existía hermetismo por parte de los vendedores de equipos de red y que la lucha por mejorar las condiciones de las redes desde un enfoque de código abierto no sería aceptable por los gigantes del mercado. También observaron que existían otras propuestas para contar con redes programables como *GENI*, pero que su implementación o puesta en práctica era demasiado compleja para efectuarla a corto plazo. En este sentido, llegaron a la conclusión que era necesario tener un enfoque más ligero que pueda ser implementado en corto plazo (McKeown et al., 2008).

Más tarde, tras varios experimentos, identificaron similitudes en las funciones de las tablas de flujo, normalmente construidas a partir de TCAM, de algunos de los conmutadores y enrutadores ethernet y observaron que estas similitudes permitían manejar la operatividad de la red con aplicaciones, a saber: cortafuegos, NAT, QoS, entre otros. Una vez que tuvieron claro que la funcionalidad de la red radicaba en sus tablas de flujo, independientemente del hardware, propusieron el desacoplamiento del plano de control del plano de datos con el uso del canal de comunicación regulado mediante el protocolo abierto OpenFlow (McKeown et al., 2008).

Desde la perspectiva de conmutación, un equipo de red OpenFlow debe al menos contar con: **(A)** una tabla de flujo, **(B)** un canal seguro y **(C)** el protocolo OpenFlow. Subsecuentemente, una tabla de flujo debe contener al menos una acción por cada entrada de flujo, estas acciones pueden ser: **(a)** reenvío de los paquetes de este flujo a un puerto (enrutamiento de los paquetes), **(b)** encapsulamiento y reenvío de los paquetes de este flujo a un controlador. El paquete se envía al canal OpenFlow, donde se encapsula y se envía a un controlador, o **(c)** descarte de

paquetes, el cual puede utilizarse para romper el *broadcast* de descubrimiento de la red o como una medida de ciberseguridad al tener múltiples peticiones. A su vez las entradas de flujo deben tener tres campos: **(i)** un encabezado de paquete que define el flujo, por ejemplo: TCP o IP, **(ii)** la acción, que define cómo deben procesarse los paquetes **(a,b o c)**, y **(iii)** estadísticas, que registran el número de paquetes y bytes de cada flujo, y el tiempo transcurrido desde que el último paquete coincidió con el flujo (McKeown et al., 2008).

En resumen, OpenFlow emerge como una solución innovadora en redes de campus universitarios, que tenía como objetivo principal derribar las limitaciones de flexibilidad, crecimiento y administración centralizada de red, dando más libertades a los administradores de red y mayores facilidades de expansión y control de los segmentos de red asignados a los investigadores de los campus. Aunque existen varios protocolos de comunicación disponibles para este propósito, OpenFlow ya es reconocido como un estándar.

OpenFlow permite que un controlador agregue, actualice o elimine entradas en las tablas de flujo de los dispositivos de red controlados por software (como los conmutadores y enrutadores). En otras palabras, el controlador puede programar los elementos de red con políticas de red personalizadas y tomar decisiones de enrutamiento en tiempo real a través de OpenFlow (Open Networking Foundation, 2015a).

OpenFlow, por sí mismo, no posee mecanismos de ciberseguridad. La ciberseguridad puede establecerse de manera opcional como una conexión entre el conmutador OF y el controlador OF. Dicho esto, las conexiones principales pueden usar TLS o TCP simple y las conexiones auxiliares pueden usar TLS, DTLS, TCP o UDP (I. Ahmad et al., 2015; Open Networking Foundation, 2015a). La ONF recominda el uso de TLS a partir de la versión 1.2, aún cuando esta presenta vulnerabilidades (NIST, 2020b; NIST, 2020a).

Algunas investigaciones han determinado que aún se presentan inconvenientes de ciberseguridad en la SBI, así J. H. Lam et al., 2015 y Agborubere et al., 2018 mencionan la falta de certificados al momento de efectuar el *handshake* en el proceso de autenticación del lado del cliente y las falencias en la configuración del protocolo TLS, con lo cual se pueden tener ataques de MiM. (Benton et al., 2013), inclusive señalan que la falta de configuración de TLS aumenta el riesgo en los conmutadores, ya que al no tener autenticación y, en muchos casos, tener activo el "modo escucha", el atacante podría tener acceso a la información y reglas de reenvío. En este sentido, para mitigar el ataque de MiM en la SBI, los autores Agborubere et al., 2018 proponen una extensión al protocolo TLS. Mientras que los autores J. H. Lam et al., 2015; J. Lam et al., 2016 usan el protocolo IBC para asegurar las comunicaciones en la SBI y el plano de datos.

Otro aspecto relevante para tomar en cuenta en la SBI, es la saturación, que podría llevar a una falta de disponibilidad de la red. En este contexto, existen dos maneras que permiten minimizar la sobrecarga de señalización entre el plano de datos y el de control (X.-N. Nguyen et al., 2015). La primera es la delegación de responsabilidades y el poder de decisión al plano de datos transformándolo de *stateless* a *stateful*, lo cual será abordado más adelante. La segunda, contempla la configuración de los controladores y conmutadores para que las reglas de flujo sean manejadas en modo proactivo (Open Networking Foundation, 2015a). En lo que respecta a esta última opción, es preciso tomar en cuenta que, aunque el modo proactivo

sería capaz de reducir el tráfico entre el plano de datos y control, también podría saturar la memoria de los conmutadores, por lo que su uso es aconsejable en ambientes en los cuales se tenga conocimiento amplio de las peticiones de la red (Alsaeedi et al., 2019).

Como ya se indicó previamente en el plano de datos se concentran todos los elementos de red y a través de estos, se implementan todas las decisiones tomadas para responder a las peticiones de la red. Una de las acciones más importantes que involucra el plano de datos para responder a dichas peticiones, es el descubrimiento de la topología, su actualización y la toma de decisiones de reenvío de tráfico que se basan principalmente en dos servicios: el *link discovery service* (LDS) y el *host tracking service* (HTS) (Marin et al., 2019).

El LDS de manera genérica emplea el link layer discovery protocol (LLDP), para recopilar información de los conmutadores y link entre conmutadores (Congdon et al., 2004). En SDN que implementan OpenFlow, la obtención de información de links entre conmutadores OF se efectúa mediante OpenFlow discovery protocol (OFDP), mientras que para obtener información de los links entre conmutadores OF y conmutadores tradicionales se usa broadcast domain discovery protocol (BDDP) (Ochoa Aday et al., 2015). Tanto OFDP como BDDP son una adaptación del protocolo LLDP. Por su parte HTS mantiene la información de los hosts y su posicionamiento dentro de la red.

El proceso de estos servicios está enfocado en el *handshake* entre conmutador y controlador, que se resume a un intercambio de mensajes *packet_in* y *packet_out*. Los intervalos para efectuar dicho intercambio de mensajes entre entidades están determinados por cada controlador. Con la información provista de la actualización de topología, el controlador puede ejercer una correcta administración de los recursos de la red, permitiendo reencaminar el tráfico conforme las necesidades reales del entorno, discriminando caminos óptimos, en caso de requerirlo, y dando mejor calidad de servicio (QoS) a las aplicaciones dotadas por el plano de aplicación (Shu et al., 2016).

A pesar de la importancia de ambos servicios, se ha observado que el proceso de intercambio de paquetes para la actualización de la topología presenta problemas de ciberseguridad derivados de la carencia de mecanismos de protección en el HTS del controlador y la falta de adecuados mecanismos de autenticación de origen del paquete LLDP, el cual llega transformado al controlador en un mensaje *packet_in* legítimo, y por ende se lo trata como un requerimiento real. En este punto, un atacante (*host* o conmutador malicioso) puede lograr ataques de envenenamiento de la topología (T. H. Nguyen et al., 2016; T. H. Nguyen et al., 2017; Azzouni et al., 2017; S. Khan; Gani; Abdul Wahab et al., 2017), a saber: *host location hijacking attack* o *link fabrication attack* (Hong et al., 2015), con la finalidad de introducir información ilegítima para construir nuevas rutas y así desviar el tráfico con fines maliciosos. Inclusive, dentro del proceso de actualización de la topología de la red se pueden desencadenar otros ataques de tipo DoS (*LLDP flooding* o *packet injection attack*) (Deng et al., 2018), *topology tampering attacks (port probing, port amnesia)* (Skowyra et al., 2018), *repudiation* o MiM, pudiendo ser este último ejecutado mediante un *silent relay attack* (Shrivastava et al., 2018).

Como se puede observar existen varios inconvenientes de ciberseguridad en el proceso de descubrimiento de topología de la red. No obstante, no se puede dejar de lado a los ataques más comunes que no son exclusivos de SDN, los ataques de DoS y DDoS provenientes de

clientes finales (*host*). Estos ataques constituyen un gran reto para todo administrador de infraestructura, pues su identificación y mitigación dependen de la forma de ejecución.

Los ataques de DoS se ejecutan desde un mismo cliente por lo que su tratamiento puede ser más rápido a diferencia de los ataques de DDoS, que se realizan a través de múltiples clientes, generalmente *botnets*, lo cual hace que su identificación sea más compleja. Asimismo, se ha observado que los ataques de DoS pueden lanzarse en conjunto con ataques de impersonalización tales como *MAC address* o *IP address spoofing* (T. Y. Lin et al., 2020; T. Wang y H. Chen, 2017), o inclusive pueden desencadenarse tras un ataque por inferencia (Zhou et al., 2018). En la Figura 2.3 se aprecian algunos inconvenientes de ciberseguridad presentes en el plano de datos y la SBI.

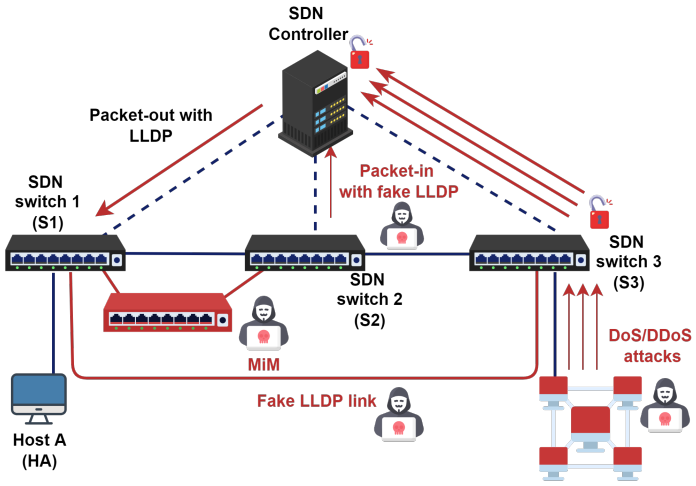

Figura 2.3: Inconvenientes de ciberseguridad asociados a la SBI y plano de datos

Para abarcar las deficiencias de ciberseguridad en el plano de datos de las SDN, muchos autores han propuesto varias soluciones que serán detalladas en esta tesis. Sin embargo, para clarificar la narrativa, este capítulo resalta la diferencia entre plano de datos sin estado (*stateless*) y plano de datos con estado (*stateful*). En un plano de datos *stateless* los elementos de red no guardan estados de la red, únicamente implementan las soluciones de las decisiones tomadas en el plano de control. Todas las acciones nuevas que deba efectuar el plano de datos *stateless* deben ser consultadas al controlador (Akyildiz et al., 2014).

No obstante, el plano de control es capaz de delegar funciones al plano de datos siempre que sea oportuno y necesario con el propósito de dinamizar el comportamiento de la red. Esta delegación le permite al plano de datos almacenar estados de la red y tomar ciertas acciones, convirtiendo así de un plano de datos *stateless* a un plano de datos *stateful* (Open Networking Foundation, 2014). En la Figura 2.4 se aprecia la diferencia existente entre ambientes *stateless* y *stateful*, incluyendo un enfoque relacionado a su seguridad.

De este modo, se estima la presencia de ataques de manera predominante en los núcleos de procesamiento en cada uno de los casos. Por lo que, en ambientes *stateless* (Figura 2.4a), los ataques pueden llegar a afectar mayoritariamente al controlador, sin descartar afectación a nivel de los elementos de red. Por otro lado, en un ambiente *stateful* (Figura 2.4b), aunque los ataques son más visibles en los elementos de red, también pueden extenderse al controlador.

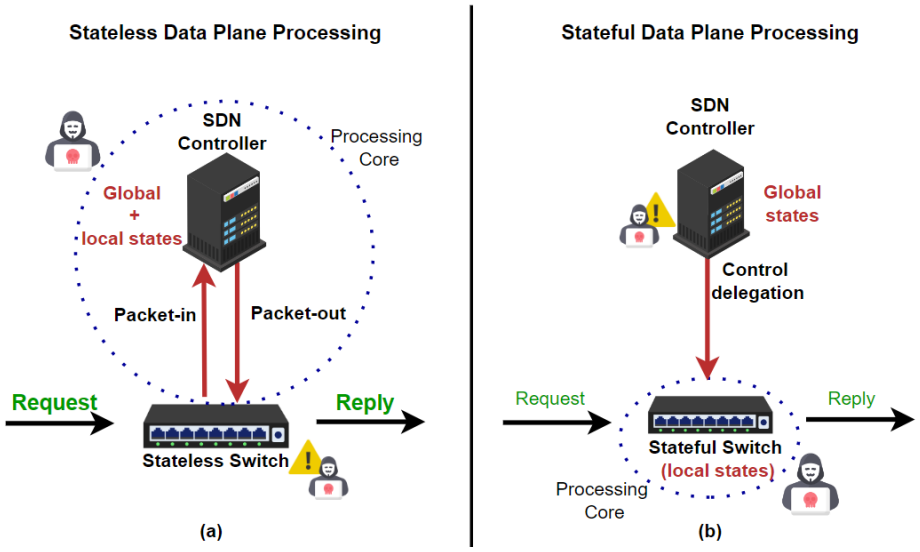

Figura 2.4: Plano de datos *stateless* y *stateful*. Adaptado de Capone et al., 2015; X. Zhang et al., 2021.

Plano de datos *stateless*

Dentro de un plano de datos *stateless* existen diversas soluciones para mitigar o detectar ataques en la topología de la red (*hijacking, link fabrication, packet injection attack*, etc.). A continuación se detallan algunas de ellas.

Dhawan et al., 2015, presentan *SPHINX*, un framework capaz de integrarse al controlador como un módulo o como una aplicación. Esta solución monitorea los mensajes OpenFlow entre controlador y conmutador para extraer los estados de la topología y reenvío de la red. De esta manera logra construir los gráficos de flujo, los cuales son contrastados con las políticas aprendidas o establecidas en el motor de políticas. Con lo cual, en caso de sospecha de una anomalía en la red se levanta una alarma.

TopoGuard (Hong et al., 2015), es una extensión para controladores que trabajan con OpenFlow para evitar *host location hijacking attack* y *link fabrication attack*. *TopoGuard* consta de cuatro módulos principales. *Port manager*, supervisa los mensajes OpenFlow de los puertos del conmutador. *Port property*, cuyo objetivo es almacenar información (precondición) que se usa para verificar la confiabilidad de la actualización de la topología. *Host prober*, es la contraparte en el proceso de verificación de confiabilidad (postcondición). Finalmente, *Topology update checker*, que efectúa el proceso de verificación de la información de la topología basado en precondiciones y postcondiones, valida el origen y la integridad de los paquetes LLDP a través del *keyed-hash message authentication code* HMAC/TLV. Además, en el caso de detectar una anomalía en el *internal link update*, bloquea el puerto y levanta una alerta. De este modo logra mitigar ataques de envenenamiento de la topología de red. Es por ello que varios controladores de código abierto trabajan con esta solución, pese a ser cuestionado en ciertos escenarios (Xiang et al., 2018).

Los autoes Shrivastava et al., 2018, indican que los *silent relay attacks* (ataques de tipo MiM), podrían inyectar links falsos entre conmutadores en el proceso de descubrimiento de la

topología. Adicionalmente, los autores generan una breve revisión a *TopoGuard*, señalando que, a pesar de sus controles, dicha solución no logra detectar *silent relay attacks*. Ante esto, se presenta una solución denominada *Silent Relay Detector*, cuya principal idea es forzar al atacante a comportarse de manera inusual. La solución detecta a un atacante que ejecuta un *silent relay* entre conmutadores, mediante la comparación del tamaño del *payload* los mensajes LLDP que envía, en relación con el tamaño de MTU permitido.

De igual manera Deng et al., 2018, demuestran que *TopoGuard* no puede controlar los ataques de inyección de paquetes ya que en su procedimiento no verifica la legitimidad del origen de la dirección MAC. Por ello, proponen una solución llamada *PacketChecker*, la cual se cataloga como un módulo para controladores que trabajan con OpenFlow. Este cuenta con dos partes importantes: *attack detection* y *attack solution*. El *attack detection* revisa los *packet_in* y extrae información que sirve para asociar el puerto del conmutador con la dirección MAC del host. De este modo si un paquete llega con diferente dirección MAC del mismo puerto del conmutador se lo envía al *attack solution* para que el conmutador lo descarte, dado que se considera un mensaje malicioso.

INSPECTOR (Alshra'a et al., 2019), es una solución basada en hardware para los ataques de inyección de paquetes. Esta solución se apalanca en un equipo añadido a la arquitectura de la red para autenticar el origen de los mensajes *packet_in* a través de una base de datos de host válidos. Así, si un host no es autenticado, su mensaje *packet_in* se descarta.

TopoGuard+ (Skowyra et al., 2018), es una versión mejorada de *TopoGuard*, en la cual se agregan dos módulos para detectar anomalías, ya sea en las interacciones (*control message monitor* (CMM)) o en las latencias (*link latency inspector* (LLI)) durante el proceso de intercambio de paquetes LLDP. Con esta propuesta se logra mitigar *port probing* y *port amnesia*, ataques de *tampering* que lograron evadir los mecanismos de *TopoGuard* y *SPHINX*.

De igual modo existen propuestas para fortalecer el protocolo OFDP, es el caso de sOFTDP (Azzouni et al., 2017). La principal diferencia entre OFDP y sOFTDP, es que sOFTDP usa los valores hash de las direcciones MAC en el proceso de intercambio LLDP a diferencia de OFDP que usa MAC pura. sOFTDP trabaja con un mecanismo detección de reenvío bidireccional en modo asíncrono para tener información de la conectividad. sOFTDP mantiene memoria de topología, con una base de datos de los links para de esta manera elegir la ruta más corta para el reenvío. Cuenta con grupos de *fast-failover*, que se encargan de revisar el estado de los puertos del conmutador y en caso de requerirlo generar un cambio a un enlace de respaldo. Por último, tiene reglas *LLDP drop*, que le permiten eliminar paquetes LLDP nocivos y *hashed LLDP content* para el envío de paquetes LLDP cifrados.

En un contexto distinto, para los ataques que afectan a la disponibilidad del servicio existen soluciones tanto de análisis de estadísticas de tráfico dentro de un tiempo determinado, como aquellas basadas en inteligencia artificial. Estas soluciones que aunque no reducen la cantidad de los ataques, pueden ayudar a detectarlos y contenerlos tempranamente para disminuir el impacto nocivo en los servicios de red. A continuación se presenan algunas soluciones.

DAISY (Imran et al., 2020), es una propuesta para detección y mitigación de ataques DoS que posee cuatro funciones principales. La función *data collection*, recolecta y almacena información de los mensajes *packet_in*. La función *threat detection*, a través de estadísticas

determina si existe o no un exceso de solicitudes por parte del host. En caso de existir un exceso, este tráfico es catalogado como sospechoso y la función *attack prevention* bloquea ese tráfico durante un intervalo corto de tiempo. Si el host continúa enviando solicitudes en la misma cantidad, el tráfico es considerado malicioso y se lo bloquea durante más tiempo. Finalmente, *threat value reduction* actualiza las reglas de flujo de bloqueo tras cada iteración del sistema.

FloodGuard (H. Wang; L. Xu et al., 2015), es una solución de seguridad que se encarga de un ataque de DoS que satura el plano de control desde el de datos. Usa dos módulos OF que son incluidos como una aplicación en el controlador, estos módulos son: *Proactive Flow Rule Analyzer* y *Packet Migration*. Este último está compuesto de dos submódulos: *Migration Agent* y *Data Plane Cache*. El momento que *FloodGuard* a través de *Migration Agent* detecta un ataque, redirecciona los *table-miss* al *data plane cache* y el sistema entra en modo de defensa, enviando los flujos benignos al controlador, sin recargarlo. En paralelo, *proactive flow rule analyzer* genera e instala las nuevas reglas de flujo directamente sobre el plano de datos y las mantiene actualizadas dinámicamente. Cuando se detecta el fin del ataque el *data plane cache* deja de recibir los flujos y se vuelve a un estado normal.

En (X. Huang et al., 2017), los autores formulan una solución basada en entropía haciendo uso de un *security gateway* y una *honey pot*. El *security gateway*, a través de los algoritmos de defensa y de filtrado, determina si existe o no un ataque de DDoS, si el ataque existe entonces se envían los paquetes a la *honey pot*, caso contrario se solicita reenvío de reglas al controlador para desplegarlos en los conmutadores.

Los autores Y. Zhang et al., 2018, exponen un ataque de saturación denominado *table-miss striking attack*. Este ataque se presenta cuando un ente malicioso aprende información sensible y confidencial del plano de control. Con esta información el atacante es capaz de generar patrones de tráfico y por ende suficiente comunicación en la SBI hasta saturarla. Ante esto, J. Xu et al., 2020, proponen una solución denominada *SDNGuardian*. Esta solución tiene cuatro módulos: *preprocessor, attack detector, traffic filtering, rule sweeper*. Los dos primeros módulos son los encargados de la identificación, la extracción y el almacenamiento de campos sensibles de un *packet_in*. Con la información extraída y el uso de algoritmos de entropía se determina si existe o no un ataque. Los dos últimos módulos tienen la responsabilidad de identificar los puertos del conmutador que están siendo atacados para limitar su velocidad y remover las reglas maliciosas de las tablas de flujo, conteniendo de esta forma el ataque.

SDNManager (T. Wang; H. Chen et al., 2018), es un planteamiento para la detección y mitigación de ataques DoS, que consta de cinco módulos. El primero, *Monitor*, recolecta los estados de la red y los convierte en variables para enviarlas al *forecast engine module* para generar un pronóstico de consumo de ancho de banda basado en estadísticas. El módulo *checker* valida si el ancho de banda utilizado es igual al pronosticado. En lo que respecta a los módulos de *updater*, y *storage service*, tienen la función de actualizar la red y almacenar las variables, respectivamente. En caso de detectar un tráfico anormal esta solución genera una penalización sobre el consumo del ancho de banda, otorgándole a un atacante menor prioridad.

Dado que a menudo las *botnets* usan comunicación P2P, los autores Su et al., 2018 plantean y

experimentan un módulo programable con aprendizaje automático que permite identificar tráfico P2P malicioso, compuesto de tres partes esenciales: *rule arbitrator* (controlador), *data-link bridges* (conmutadores OpenFlow) y *detection agent*. Esta solución permite, posterior a un proceso de duplicado de paquetes entrantes y reconocimiento de flujo, etiquetar cada flujo como P2P o aplicación P2P benigna, mediante modelos de aprendizaje. Una vez que se cuenta con esta información se efectúa una modificación en las tablas de flujo, de este modo se descartan los paquetes maliciosos.

También, se han identificado soluciones que abordan ataques de *botnets* del plano de datos, pero con la propuesta desde la capa de aplicación. De este modo, Maeda et al., 2019, propusieron la utilización de aprendizaje profundo basado en *Multi-Layer Perceptron* (MLP), en la capa de aplicación, para detección de *botnets* en SDN. Para probar su propuesta los autores utilizaron tres conjuntos de datos. Por una parte, para tomar el tráfico de las *botnets* usaron el conjunto de datos CTU-13, mientras que para el tráfico normal usaron un conjunto de datos propietario y el conjunto de datos ISOT. De dichos conjunto de datos extrajeron información de cabecera de los paquetes IP/TCP/UDP, principalmente. Una vez ejecutadas las pruebas, la solución alcanzó una exactitud de detección de 99.2 %.

También se ha observado que muchas *botnets* aprovechan las características de HTTP para generar ataques DDoS que se expresan a través de ataques de inundación HTTP GET. En este sentido, los autores Viet et al., 2017 proponen una solución que combina hardware y un mecanismo de entropía. El mecanismo denominado *per-URL counting*, trabaja con dos secciones, una de detección y otra de mitigación. En la sección de detección se valida la petición HTTP GET, a través de varios filtros, siendo uno de estos el de conteo. Si el contador supera un umbral, se considera un ataque y se ubica al *host* en una lista negra que es manejada por la sección de mitigación. Finalmente, todo el sistema es presentado en hardware *Field Programmable Gate Array* (FPGA).

Kumar et al., 2018; Mousavi et al., 2015; Sahoo; Puthal et al., 2018, presentan soluciones de detección temprana de ataques DDoS haciendo uso de entropía. Sin embargo, los autores D. Wu et al., 2018, señalan que las medidas de entropía ante ataques DDoS funcionan cuando el objetivo de ataque tiene un direccionamiento IP fijo pero cuando el ataque es a una IP aleatoria las medidas de mitigación se ven limitadas, dado que los umbrales no contemplan las posibles varianzas, por lo que plantean una solución que hace uso de *Principal Component Analysis* (PCA). De este modo, con la información recolectada se pueden tener nuevos modelos que permiten predecir el ataque.

De igual manera, en relación con los ataques de DDoS con objetivo aleatorio, Shohani et al., 2020, proponen una solución de detección temprana. La solución se compone de tres fases. En la primera se recopila la información de los conmutadores. La segunda usa algoritmos que calculan los umbrales mediante el modelo estadístico EWMA, con el cual es factible manejar la volatilidad del conjunto de datos y en la tercera fase se realiza la detección del ataque comparando los valores obtenidos de la fase anterior con las *table-miss* de los conmutadores.

Asimismo, en lo que respecta a detección temprana de DDoS se han presentado contramedidas que hacen uso de IDS, es el caso de la propuesta de los autores Badotra et al., 2021, en la cual

se usan reglas de *Snort* IDS[6] y a la vez generan las alarmas en caso de detectar un ataque.

Barki et al., 2016, presentan un IDS para detectar ataques de DDoS. La solución está conformada de dos módulos. El primero de ellos clasifica los comportamientos normales o anómalos de los hosts a través de firmas obtenidas de los algoritmos de aprendizaje automático. Si se detectan anomalías asociadas a un host, este se delega al segundo módulo, en el cual se determinará la legitimidad del host a través de un *handshake* de tres vías y en caso de que este proceso no se complete se puede incluir al host como atacante en una lista de control de accesos. Esta propuesta trabaja con un conjunto de datos obtenido de tráfico TCP del laboratorio de *Lawrence Berkley*[7] y red real sin mostrar valores de exactitud obtenidos en las pruebas.

C. Li et al., 2018, proponen una solución de detección y defensa de ataques de DDoS con aprendizaje profundo. Esta arquitectura que usa bidirectional *Recurrent Neural Network* (RNN) está pensada para detección de ataques DDoS en SDN. Para probar su propuesta se generan pruebas en distintos modelos LSTM con el conjunto de datos ISCX2012, alcanzando hasta un 99 % de exactitud.

Sahoo; Tripathy et al., 2020, emiten un framework para detectar y mitigar ataques DDoS mediante el uso de algoritmos de aprendizaje automático. Su primer módulo recibe la información de estadísticas de flujo desde los conmutadores en determinados intervalos de tiempo. Posteriormente, se obtienen las características de los flujos con el uso de *Kernel Principal Component Analysis* (KPCA). Las características extraídas son utilizadas en el módulo classifier que trabaja con el clasificador SVM con optimización de parámetros mediante genetic algorithm (GA), de esta manera identifica el tráfico malicioso del benigno. Para la propuesta usaron el conjunto de datos NSL-KDD, obteniendo una exactitud de 98.9 %.

Aujla et al., 2020, proponen una arquitectura como servicio basados en blockchain, para mitigar ataques de DDoS. Los autores detallan el procedimiento de su propuesta, que se resume en la validación de la identidad de un conmutador que solicita acceso a la red antes del inicio de la transmisión del flujo. En esta solución, las claves públicas y privadas se generan mediante *blockchain* y se comparten entre los conmutadores de la red. Así, cada vez que un dispositivo genera una solicitud en la red y tras un consenso de los demás miembros, las transacciones son aceptadas o rechazadas.

Los autores Makuvaza et al., 2021, presentan una solución para detección de ataques DDoS en SDN. Esta propuesta de detección trabaja entre los elementos de red y el controlador. Esta solución usa un modelo de aprendizaje profundo y un subconjunto de datos extraído manualmente del conjunto de datos CICIDS 2017. En este sentido, los autores usaron cuatro características como datos de entrada y probaron una red neuronal que consta de 128 perceptrones. Posterior a una evaluación, obtuvieron una exactitud de detección de 97.59 %.

Dehkordi et al., 2021, proponen una solución para detectar ataques de DDoS de bajo y alto volumen de tráfico. Cuenta con dos tipos de umbrales, estáticos a través de entropía y dinámicos a través de aprendizaje automático. En lo que respecta a los algoritmos de aprendizaje automático, los autores usaron *BayesNet*, J48, *RandomTree*, regresión logística y

[6]https://www.snort.org/
[7]https://www.lbl.gov/

REPTree, La propuesta trabajó con los conjuntos de datos: UNB- ISCX1 y CTU-132 y dado que se realizan pruebas parciales se obtienen resultados para cada conjunto de datos, llegando a un promedio de exactitud de 99.48 % .

Polat et al., 2020, analizan algoritmos SVM, *Naive Bayes* (NB), *Artificial Neural Network* (ANN), y *K-Nearest Neighbors* (KNN) para la detección de ataques DDoS. El planteamiento consta de dos fases. En la primera observan el comportamiento de los algoritmos sin selección de características y la segunda incorpora métodos de *filter, wrapper, embedded* para obtención de características de un conjunto de datos propietario. Mostrando mayor rendimiento en las pruebas realizadas con selección de características, y alcanzando exactitud de un 92,46 % para SVM, 97,15 % para KNN, 92,28 % para ANN y 95,7 % para NB.

Los autores Fouladi et al., 2022, presentan una solución para ataques DDoS basada en *Discrete Wavelet Transform* (DWT) para la obtención de características y redes neuronales con *autoencoder* (AE) para la clasificación del tráfico. La solución ha sido probada haciendo uso de conjuntos de datos MAWI y FIFA world cup. Indican que alcanzan una tasa de detección de 100 % mas no especifican la exactitud.

Singh et al., 2022, plantean un framework de detección y mitigación de ataques DDoS el cual realiza el proceso de detección a través de CNN. Previamente realizan un proceso de selección de características a través de una mejora del algoritmo de optimización *Rider* (IU-ROA). Con esta propuesta alcanzan una exactitud de 90.06 %, usando características del conjunto de datos KDD'99.

Gadallah et al., 2024, proponen una técnica para la detección de ataques DDoS usando AE con redes neuronales denominadas *Bidirectional Gated Recurrent Unit* (BGRU). Extrayendo cierta información de las estadísticas de tráfico, logran establecer la existencia de un ataque. Para ello, obtiene un subconjunto de datos del conjunto de datos NSL-KDD, logrando una exactitud de hasta un 99,91 %.

T. Wang y H. Chen, 2017, plantean una solución denominada *SGuard* para mitigar ataques de spoofing y de DoS. Está compuesta de tres módulos: access control, classification y data plane cache. El access control recopila información de los usuarios de la red y la contrasta con una ACL de *SGuard*, evitando ataques de *spoofing*. Mientras que el módulo de classification, discierne el tráfico anómalo del benigno, mediante *Self Organizing Maps* (SOM). Por su parte el módulo *data plane cache* se encarga de mantener los *table-miss packets* durante un ataque con lo cual evita el agotamiento de recursos en el plano de datos.

Mohammadi et al., 2017, abordan dos tipos de ataque: *spoofing route attack* y *resource exhaustation attack*. El primer ataque, se lo contrarresta mediante una función denominada *selective blocking* instalada sobre el conmutador OF. *Selective blocking* extrae información (IP, MAC) cuando recibe una nueva petición a la red. Esta información es contrastada con datos preexistentes y en caso de encontrar duplicación se comprende que existe suplantación y se genera un bloqueo al host malicioso. Para el segundo ataque, los autores usan un módulo de detección y prevención llamado *periodic monitoring* el cual valida la cantidad de tráfico enviado en un tiempo determinado. Si los valores sobrepasan un umbral se genera una alarma para que el controlador se encargue del bloqueo.

A continuación, se hace énfasis en las vulnerabilidades de las tablas de flujo de los conmutadores de redes que usan el protocolo OF, dado que pueden sufrir ataques de inferencia (Zhou et al., 2018). En este tipo de ataques los adversarios mediante minería de datos y con el apoyo de diferentes algoritmos son capaces de estimar, entre otras cosas, el tamaño de la tabla de flujo de los conmutadores OF. Con esta información el atacante es capaz de generar peticiones que saturen la memoria de los conmutadores, hasta llevarlos a una DoS.

Los autores Khorsandroo et al., 2018 consideran que los ataques por inferencia tienen sus propias complicaciones y retos por lo que desplegarlos en una red real sería extremadamente complejo. No obstante, proponen dos contramedidas para evitar estos ataques. La primera se basa en una técnica de aleatoriedad de los atributos de la red (Chavez et al., 2016) y la segunda, hace uso de limitación de tasa de paquetes OF combinado con reglas proactivas para mitigar ataques ICMP inferidos o limitación de tasa con un *proxy* para mitigar ataques TCP.

Otra preocupación latente referente a las tablas de flujo está relacionada con las sobrecargas que estas puedan tener. Dichas sobrecargas pueden o no provenir de una acción maliciosa, pero pueden influir negativamente en la disponibilidad de la red, por lo que es necesario mantener una administración adecuada de las memorias de los conmutadores para que respondan correctamente.

En este sentido, existen tres maneras para administrar la memoria de los conmutadores de forma eficiente: *rule eviction, rule compression*, y *rule split and distribution*. La primera y generalmente usada, busca el reemplazo de entradas antiguas y el uso de nuevas entradas en las tablas de flujo. Se apalanca en la utilización de algoritmos de sustitución de caché, por ejemplo *Least Recent Used* (LRU), *First-In First-Out* (FIFO), en el estado del flujo, y en el tiempo de espera en los conmutadores (*hard timeout/idle timeout*). La segunda manera, *rule compression/aggregation*, busca reducir la cantidad de reglas usando *wildcards* hasta ajustarse a la tabla de flujo. La compresión puede realizarse tanto a reglas de control de acceso como a reglas de reenvío. Por último, *rule split and distribution*, mantiene como idea principal la división de las reglas entre varios conmutadores a fin de que se encuentren distribuidos en la red (X.-N. Nguyen et al., 2015).

Dado que la administración de la memoria de los conmutadores se torna un punto sensible para un óptimo rendimiento de las SDN, algunos autores han considerado las técnicas antes mencionadas (Assefa et al., 2019; Karakus et al., 2017a). Sin embargo, son pocas las propuestas orientadas a un ámbito de seguridad. Así en (Xie et al., 2021), los autores proponen un mecanismo de detección y defensa contra ataques LDoS (*Low-rate DoS*), basado en análisis estadístico. Esta propuesta usa la técnica de reemplazo apoyado con el algoritmo LRU para la detección y para la mitigación de dichos ataques. Zhou et al., 2018, plantean un modelo de ataques de inferencia y dos contramedidas. El modelo de ataque usa algoritmos de sustitución FIFO y LRU. En lo que respecta a las contramedidas, la primera está basada en *routing aggregation* y la segunda en una arquitectura de tabla de flujo con dos niveles, el nivel uno conformado por TCAM y el nivel dos por SRAM.

Por otro lado, para ataques de impersonalización, protocolos como EAP, EAPoL y RADIUS son utilizados para autenticar y autorizar a los clientes de la red antes que accedan a la misma. Bajo esta premisa Nife et al., 2018 y Benzekki et al., 2016, han planteado soluciones

que trabajan con el protocolo EAP, en modo reactivo; o Matias et al., 2014, que logran su objetivo mediante *EAPoL-in-EAPoL encapsulation*, en modo proactivo.

Plano de datos *stateful*

La mayoría lenguajes de programación para SDN se han basado en OpenFlow 1.0, el cual tiene por defecto una configuración del plano de datos *stateless*, es decir que los conmutadores únicamente cumplen las reglas de reenvío que fueron emitidas por el controlador, lo que en ciertas ocasiones genera sobrecarga de trabajo para el controlador y latencia para peticiones de las abstracciones de red tales como contrafuegos, balanceadores de carga, entre otras (Sun et al., 2017).

En este sentido, se ha observado que es posible extender la programabilidad de aplicaciones de red al plano de datos, manteniendo la información de los estados locales en los conmutadores, para que puedan tomar el control de reenvío de paquetes sin hacer la consulta al controlador (plano de datos *stateful*), proporcionándole "mayor dinamismo" a la red.

No obstante, existen riesgos de ciberseguridad presentes en un plano de datos *stateful* de manera inherente a esta condición. Es así como al tener que almacenar en los *state tables* de los conmutadores toda la información que antes era manejada por el controlador, se puede llegar a una saturación de la memoria TCAM y a un alto consumo de CPU de los conmutadores, dejando exhausta a la red hasta obtener una auto denegación del servicio (Dargahi et al., 2017).

Para otorgar programabilidad al plano de datos existen propuestas como SDPA (Sun et al., 2017), que extiende el procesamiento *"match-action"* de OpenFlow y que propone una abstracción *"match-state-action"*. SNAP (Arashloo et al., 2016), que basa su programabilidad en una red abstracta y arreglos globales persistentes. O aquellas basadas en tablas XFSM (*eXtensible Finite State Machines*) para el procesamiento de flujos en los conmutadores tales como FAST (Moshref et al., 2014), OpenState (Bianchi; Bonola; Capone et al., 2014), OPP (Bianchi; Bonola; Pontarelli et al., 2016), y la más conocida P4 (Bosshart et al., 2014).

Apoyados en las soluciones de programabilidad antes descritas se han desarrollado múltiples abstracciones de red para los planos de datos *stateful*, a saber: contrafuegos (Caprolu et al., 2019; Krongbaramee et al., 2018), aplicaciones de gestión del tráfico (Cascone et al., 2015), balanceadores de carga (Katta; Hira et al., 2016), entre otros (Bonola et al., 2017).

De igual manera, bajo esta tendencia los autores (Yazdinejad et al., 2020), plantean la arquitectura *blockchain-enabled packet parser* (BPP) para detección de hasta cinco categorías de ataque, siendo la más representativa DoS. Esta solución fusiona *blockchain* y P4, y se implementa sobre FPGA. P4 personaliza el procesamiento de paquetes en el plano de datos mientras que *blockchain* examina el comportamiento de los paquetes. En caso de detección de un ataque, se reporta al controlador y se toman acciones bajo las políticas definidas.

T. Y. Lin et al., 2020, trabajan sobre *SYN flooding* y ataques de *spoofing* ARP. En la solución para el primer ataque, utilizan la proporción de paquetes SYN/ACK y ACK/FIN. Con ello, si la relación existente entre estos paquetes no coincide, la red tiene tráfico anómalo. Respecto al segundo ataque, los autores logran descartar los ARP repetidos mediante P4 cache, para

reducir la carga en el controlador.

Hwang et al., 2019, plantean un framework de ciberseguridad basado en P4 para reducir la sobrecarga de tráfico y la latencia. Está compuesto de dos partes primarias: *statefit app* y *statefit interpreter*. El primero está instalado como una aplicación en el controlador y se encarga, entre otras cosas, del análisis del tráfico. El segundo, actúa en los conmutadores P4. En este punto se procesa y filtra el tráfico conforme las políticas establecidas.

Los autores Lewis et al., 2019, proponen P4ID. Esta solución busca reducir el tráfico de la red generando un filtrado en los conmutadores P4 antes que los paquetes sean enviados a un IDS. P4ID está compuesto de dos partes importantes: *rule parser* y *P4 implementation*. El primero trabaja con las reglas de *Snort* IPS que son instaladas en los conmutadores P4 para que pueda efectuarse el *match-action* inicial de manera orgánica. Mientras que a través de P4 los autores son capaces de procesar paquetes *stateless* y *stateful*.

Ilha et al., 2020, presentan una propuesta para detectar y mitigar ataques DDoS en ambientes stateful con P4, para lo cual hacen uso de entropía en la detección, mientras que para la mitigación usan Finite-State Machine (FSM).

Pese a la ventaja que representa un plano de datos stateful respecto a la disminución de tráfico entre los planos de datos y control durante el procesamiento de peticiones, se ha observado que en este tipo de ambientes podría poner en riesgo a la consistencia de la red (Zaidi et al., 2018), ya que no se ha generado un mecanismo de autenticación entre los conmutadores por lo que estos en cualquier momento podrían ser impersonalizados por agentes maliciosos dando cabida a ataques. En este sentido, se presentó una propuesta para un *secure link discovery*, cuya idea es manejar paquetes LLDP protegidos a través de cifrado, descifrado y autenticación. La propuesta que se denomina P4-MACsec, protege los links entre conmutadores basados en P4 mediante la utilización del protocolo IEEE 802.1AE (MACsec). Esta solución contiene tres partes principales: *Packet switching* con aprendizaje de direccionamiento MAC, *secure link discovery* y despliegue automatizado de MACsec (Hauser et al., 2020).

Del mismo modo, los autores Xing; A. Chen et al., 2020, preocupados por el intercambio de estados entre conmutadores P4, proponen una solución de autenticación mediante firmas digitales. Esta solución crea una cadena *hash* adjunta a cada paquete que se transmite para el intercambio de estado. Aunque el controlador mediante clave pública es responsable de la parte final de la cadena, las operaciones de verificación de esta se efectúan en el plano de datos.

Xing; W. Wu et al., 2019, plantean una solución para los ataques de inundación en los enlaces de los planos de datos programables. La idea de esta propuesta es reencaminar el tráfico maligno por rutas disponibles, usando ofuscación de la topología. En esta propuesta, el atacante cree que su flujo está logrando su propósito malicioso, aunque realmente este está siendo descartado.

Los autores Musumeci; Ionata et al., 2020, sugieren una medida para la detección de ataques DDoS. Esta propuesta explota P4 y usan algoritmos de aprendizaje automático. A través de los conmutadores, se obtiene información del tráfico, la cual es clasificada con algoritmos de KNN, RF y SVM para conocer si existe o no un ataque. Utilizan conjuntos de datos

obtenidos de la escucha de tráfico. Los resultados mostraron hasta un 98 % de exactitud. Cabe mencionar que esta propuesta cuenta con una extensión en la cual se agregan redes neuronales en el proceso de clasificación (Musumeci; Fidanci et al., 2022). De igual manera realizan las pruebas con un conjunto de datos extraído de los conmutadores P4 y obtienen 98 % de exactitud.

En la Tabla 2.4 se resumen las soluciones a los problemas presentados en el plano de datos e interfaz sur.

Tabla 2.4: Resumen de contribuciones del plano de datos e interfaz sur

Ref.	Principales preocupaciones abarcadas	Principales planteamientos de solución	Análisis de contribución a la literatura					
			S	T	R	I	D	E
(Agborubere et al., 2018)	Ataque de MiM en el canal de comunicación	Extensión del protocolo TLS		✓			✓	
(J. H. Lam et al., 2015; J. Lam et al., 2016)	Ataque de MiM en el canal de comunicación	Protocolo IBC		✓			✓	
(Hong et al., 2015)	Envenenamiento de topología de red (host location hijacking y link fabrication)	Pre y post condiciones	✓	✓			✓	
(Dhawan et al., 2015)	Violación a la topología de red y ataques DoS	Flow Graphs		✓			✓	
(Shrivastava et al., 2018)	Inyección de enlaces falsos	Comprobación del tamaño de la carga útil LLDP		✓			✓	
(Skowyra et al., 2018)	Envenenamiento de topología de red (port probing y port amnesia)	Pre y post condiciones	✓	✓			✓	
(Azzouni et al., 2017)	Vulnerabilidad de OFDP	BFD	✓	✓		✓	✓	
(Deng et al., 2018)	Packet_in falsos	Asociación del puerto del conmutador con MAC del host	✓				✓	
(Alshra'a et al., 2019)	Falta de autenticación de mensajes packet_in	Implementación independiente en hardware	✓					
(Imran et al., 2020)	Ataques DoS	Estadística					✓	
(H. Wang; L. Xu et al., 2015)	Ataques DoS	Marco de defensa independiente del protocolo					✓	
(T. Wang y H. Chen, 2017)	Spoofing y ataques DoS	ACL + Aprendizaje automático	✓				✓	✓
(Mohammadi et al., 2017)	Spoofing route y ataques Dos	Estadística de tráfico	✓				✓	
(X. Huang et al., 2017)	Ataques DoS	Entropía					✓	
(J. Xu et al., 2020)	Ataques DoS	Entropía					✓	
(T. Wang; H. Chen et al., 2018)	Ataques DoS	Estadística de tráfico					✓	
(Maeda et al., 2019)	Botnets	Aprendizaje automático					✓	
(Su et al., 2018)	Falta de identificación de tráfico P2P	Aprendizaje automático					✓	
(Viet et al., 2017)	HTTP DDoS attacks	Entropía + hardware					✓	
(D. Wu et al., 2018)	Ataques DDoS	PCA					✓	
(Shohani et al., 2020)	Ataques DDoS	EWMA					✓	
(Badotra et al., 2021)	Ataques DDoS	Snort IDS					✓	
(Barki et al., 2016)	Ataques DDoS	Aprendizaje automático					✓	
(C. Li et al., 2018)	Ataques DDoS	Aprendizaje profundo					✓	
(Sahoo; Tripathy et al., 2020)	Ataques DDoS	KPCA+GA+ Aprendizaje automático					✓	
(Aujla et al., 2020)	Ataques DDoS	Blockchain					✓	✓
(Makuvaza et al., 2021)	Ataques DDoS	Aprendizaje profundo					✓	
(Dehkordi et al., 2021)	Ataques DDoS	Entropía + Aprendizaje automático					✓	
(Fouladi et al., 2022)	Ataques DDoS	AE + Aprendizaje profundo					✓	
(Singh et al., 2022)	Ataques DDoS	IU-ROA + Aprendizaje profundo					✓	
(Gadallah et al., 2024)	Ataques DDoS	AE + Aprendizaje profundo					✓	
(Khorsandroo et al., 2018)	Ataques a la interfaz	Limitación de tasa + Reglas proactivas + Proxy					✓	
(Xie et al., 2021)	Ataques DoS (LDoS)	Estadística / LRU					✓	
(Zhou et al., 2018)	Ataques a la interfaz	Agregación de rutas + TCAM+SRAM					✓	
(Benzekki et al., 2016) (Nife et al., 2018)	Falta de control de acceso de los clientes a la red	EAP+RADIUS						✓
(Matias et al., 2014)	Falta de control de acceso de los clientes a la red	EAPoL+ RADIUS	✓					✓
(Yazdinejad et al., 2020)	Ataques DoS	Blockchain + Hardware					✓	
(T. Y. Lin et al., 2020)	Inundación SYN y ataques de ARP spoofing	Relación de paquetes SYN/ACK y ACK/FIN/ P4 cache	✓				✓	
(Hwang et al., 2019)	Sobrecarga de tráfico Latencia	App+P4					✓	
(Lewis et al., 2019)	Sobrecarga de tráfico	Snort IPS + P4					✓	
(Ilha et al., 2020)	Ataques DDoS	P4+Entropía+FSM					✓	
(Hauser et al., 2020)	Falta de protección de enlaces entre conmutadores stateful	MACsec	✓	✓		✓		✓
(Xing; A. Chen et al., 2020)	Intercambio de estados entre conmutadores stateful	Firmas digitales	✓					✓
(Xing; W. Wu et al., 2019)	Ataques de link flooding a planos de datos stateful	Ofuscación de la topología					✓	
(Musumeci; Fidanci et al., 2022) (Musumeci; Ionata et al., 2020)	Ataques DDoS	Aprendizaje automático +P4					✓	

2.4 Discusión, retos abiertos y oportunidades de investigación

Posterior a una exhaustiva revisión del estado del arte en el ámbito de la ciberseguridad de las SDN, se ha observado que en los últimos años se han dedicado considerables esfuerzos, tanto por parte de la comunidad académica como de la industria, para abordar las vulnerabilidades existentes y atender las preocupaciones relacionadas con la ciberseguridad de estas redes. El propósito fundamental de los planteamientos ha sido minimizar el impacto de posibles ataques dirigidos hacia las SDN, lo que contribuye de manera significativa a fortalecer la integridad, confiabilidad y disponibilidad de sus despliegues.

Se ha dedicado un esfuerzo considerable para mejorar las características de las interfaces y componentes de SDN, con la contribución activa de organizaciones como la ONF. En este contexto, se ha logrado la estandarización de la comunicación en la SBI, aprovechando soluciones de código abierto, lo que representa un avance sustancial. No obstante, es importante reconocer que, a pesar de mantener la estandarización de SBI, aún existe la dependencia de protocolos adicionales como TLS para garantizar la integridad del canal de comunicación.

Por otro lado, persisten desafíos en relación con la estandarización de los canales de comunicación NBI y de la E/WI. Esto es especialmente relevante dada la sensibilidad asociada con la inclusión de aplicaciones sin medidas de ciberseguridad o a la interconexión de controladores provenientes de diferentes dominios administrativos. Los beneficios de estandarizar estas interfaces, particularmente en lo que respecta a la seguridad, se reflejarían en varios ámbitos. En primer lugar, esta estandarización reduciría los frentes de ataque actuales, que surgen de la diversidad de aplicaciones de terceros. En segundo lugar, las líneas de investigación se orientarían a reforzar las medidas de protección de la arquitectura bajo un esquema regulado. Por último, esto tendría un impacto económico en las implementaciones de SDN ya que se tornaría una preocupación menos para las organizaciones.

Respecto a la variedad de aplicaciones muchos de los autores abarcan los problemas de autorización mediante soluciones con control de acceso de grano fino, pues coinciden en que debido al comportamiento dinámico de los entornos SDN, la granularidad de los permisos de acceso es obligatoria (Padekar et al., 2016; Park et al., 2002; Toshniwal et al., 2019). Así mismo, dado que el principal problema es la relación de confianza entre las aplicaciones y el controlador, existe una oportunidad de investigación respecto a la generación de un repositorio estandarizado que valide los niveles de ciberseguridad de las aplicaciones de SDN.

En lo que respecta a la vinculación de controladores distribuidos, muchos autores señalan que existe una encrucijada entre resiliencia y consistencia (Bannour et al., 2018; Hanmer et al., 2018; Roohitavaf et al., 2019; Sakic et al., 2019; T. Zhang et al., 2016). Esto debido a que, de acuerdo con el *Brewer's theorem* o teorema CAP (*Consistency, Availability, Partition Tolerance*), en los ambientes distribuidos, posterior a una partición, no se puede asegurar por igual la disponibilidad y la consistencia. Es por ello que, un reto presente al momento de mantener controladores distribuidos es la mejora del desempeño de mecanismos de tolerancia a fallas, a fin de que sean capaces de brindar disponibilidad sin sacrificar la información de la topología circundante entre controladores tras la recuperación de una interrupción del

servicio.

Por último, en lo que refiere al plano de datos es importante mencionar que este se encuentra evolucionando y es capaz de responder a los requerimientos de la red sin consultar constantemente al controlador. Contradictoriamente, esto puede suponer una descontextualización del concepto original de SDN, pues ya sea habilitando la configuración para el manejo proactivo de las reglas de flujo o delegando funciones en el plano de datos, se crea un dilema entre reducir el tráfico en el canal de comunicación SBI o sobrecargar potencialmente los conmutadores. Los retos dentro de este enfoque están enmarcados principalmente en la evaluación de las técnicas de gestión de la memoria de los conmutadores y las opciones de programabilidad del plano de datos. Para ello se torna importante considerar el desempeño, limitaciones y los riesgos de ciberseguridad asociados a dicho plano, tales como inconsistencias en la topología de la red, divulgación de información, sobrecarga computacional, duplicación o eliminación de reglas de flujo, o sobrecarga de un nodo (Dargahi et al., 2017; X.-N. Nguyen et al., 2015; Rifai et al., 2017).

Tras una minuciosa comprensión al estado del arte, se evidencia que la mayoría de las propuestas abordan las soluciones desde una perspectiva fragmentada, tratando a las SDN como entidades o servicios aislados a la organización, lo cual ha llevado a subestimar, dentro de las propuestas, las posibles repercusiones derivadas de problemas de ciberseguridad. En este contexto, la ONF ha subrayado claramente que, para avanzar con las implementaciones de las SDN a gran escala, es imperativo abordar los desafíos de ciberseguridad de manera holística. Por lo tanto, es esencial no solo considerar las implicaciones técnicas de los problemas de ciberseguridad en las SDN, sino también evaluar cómo estos eventos pueden desencadenar resultados negativos en otros aspectos organizacionales.

En consecuencia, se ha notado que se ha subestimado la repercusión jurídica que puede acarrear un problema de ciberseguridad en los entornos SDN. Pues si existe un inconveniente o un ataque que resulte en la pérdida de información o incumplimiento de los SLA, sumado a la ausencia de medidas adecuadas para conocer y tratar el origen del problema, esto podría derivar en acciones legales de gran escala para las organizaciones, llegando inclusive a afectar su economía debido a que los problemas de ciberseguridad generan pérdidas financieras de proporciones considerables en la industria de las telecomunicaciones, ya sea debido al pago de rescates de información, al pago de costas procesales o al pago de los costos asociados para solventar los daños ocasionados.

Por lo que, en un contexto donde la ciberseguridad de las SDN ya no puede reducirse únicamente a un asunto técnico, sino una parte integral de la estrategia organizacional, se hace imperativa la necesidad de contar con guías y herramientas que permitan gestionar los recursos y la información cuando se desencadena un inconveniente de ciberseguridad, a fin de que tanto el eje organizacional como técnico solventen las preocupaciones que puedan presentarse.

Desde luego, esto debe abarcar un espectro que adopte perspectivas más avanzadas y proactivas. Para ello, lo ideal sería considerar la detección y tratamiento de incidentes de ciberseguridad de manera extensa, tomando en cuenta incluso las amenazas internas que puedan ser generadas por administradores maliciosos. Así también, se debería considerar la preservación de registros

con integridad, para utilizarlos en caso de enfrentar responsabilidades legales. Por lo que, se observa una gran oportunidad de investigación referente a la gestión integral de ciberseguridad de SDN, pues se hace evidente la falta de un framework, que vincule prácticas de preparación forense digital y procesos de respuesta a incidentes en este tipo de redes.

Tomando en cuenta estas observaciones, esta tesis doctoral trata los retos relacionados con la gestión de incidentes de ciberseguridad en combinación con los procesos de preparación forense digital en SDN. Desde una perspectiva de alto nivel, la gestión de incidentes de ciberseguridad consiste en planificar, implementar y supervisar medidas y políticas de seguridad, mientras que los procesos de preparación forense digital están vinculados con todos los pasos necesarios para obtener una evidencia.

Es por ello que esta tesis doctoral, brinda un enfoque integral de la gestión de ciberseguridad en SDN a través de la presentación de un framework que espera servir como instrumento y guía para entregar evidencia y al mismo tiempo responder a los incidentes de ciberseguridad, considerando la integridad, la confidencialidad y la disponibilidad de los elementos de la SDN y los datos asociados. De igual manera, para ampliar la visibilidad del framework, se ha contemplado la presentación de contribuciones adicionales enfocadas en mejorar su practicidad. Esto incluye la propuesta de una arquitectura que respalda de manera pragmática el enfoque conceptual, así como la introducción de dos modelos prevalentes para fortalecer su utilidad y aplicabilidad.

Dicho esto, se considera oportuno profundizar en el conocimiento de la ciencia forense digital y los conceptos relativos a los procesos de respuesta a incidentes aplicados a las SDN, en el siguiente capítulo. El fundamento de esta decisión radica en simplificar la comprensión y el análisis de los principios esenciales relacionados con el tema de investigación de esta tesis, al consolidarlos en un único lugar, de forma organizada y de fácil acceso.

2.5 Resumen del capítulo

En este capítulo se presentó la primera contribución de esta tesis, orientada al estado del arte, enfocándose en la comprensión de los desafíos de ciberseguridad en las SDN, destacando su importancia para guiar el desarrollo de las posteriores contribuciones. De esta manera, se exploraron las vulnerabilidades específicas de las SDN, aumentando la conciencia sobre posibles vectores de ataque y técnicas utilizadas por ciberdelincuentes. Se profundizó en los fundamentos de las SDN, su arquitectura y las principales preocupaciones de seguridad, incluyendo una revisión detallada en cada plano y los componentes clave, como los controladores. También se hace un análisis de las soluciones usando la metodología *STRIDE*. Se concluye con una discusión sobre los retos abiertos y oportunidades de investigación, enfatizando la necesidad de un enfoque integral para la gestión de la ciberseguridad en SDN que abarque ciencias forenses y gestión de incidentes.

Capítulo 3

FRAMEWORK Y ARQUITECTURA PARA LA PREPARACIÓN FORENSE Y RESPUESTA A INCIDENTES DE CIBERSEGURIDAD EN SDN

3.1 Introducción

La amplia gama de ventajas, como la flexibilidad y el dinamismo en la gestión, que ofrecen las SDN en comparación con las redes tradicionales, ha llevado a su adopción en varios servicios de telecomunicaciones. Sin embargo, el desacoplamiento de las capas y el control centralizado, que son sus características más destacadas, también puede ser su punto más vulnerable, ya que los atacantes se ven atraídos por esta condición, convirtiéndose en un objetivo potencial de ataques.

Cuando se desencadena un incidente de ciberseguridad, tanto los campos técnicos como organizacionales están preocupados en conocer la causa raíz del mismo. Por una parte, el área técnica necesita tener información que le permita dar soluciones a los problemas, mientras que las áreas organizacionales intentan tener evidencia que pueda ser usada, por ejemplo, en un ámbito legal. Para poder cubrir tanto las dudas técnicas como para asistir a las necesidades de un proceso investigativo, se puede recurrir a los procesos de respuesta a incidentes y a los procesos forenses, respectivamente. Sin embargo, el manejo de estos de manera individual puede llevar a la duplicidad de esfuerzos y mayor consumo de recursos.

A pesar de la estrecha relación entre los procesos de preparación forense y los de respuesta a incidentes de ciberseguridad, su integración dentro de las SDN ha enfrentado desafíos significativos. Estos desafíos están principalmente relacionados con la falta de modelos y arquitecturas que faciliten la integración efectiva de las áreas forenses y de respuesta a incidentes de ciberseguridad en este tipo de redes. Además, su integración implica grandes

esfuerzos en términos de implementación técnica.

En el presente capítulo se abordan estos inconvenientes, proponiendo como una de las contribuciones principales de esta tesis doctoral un framework que integra las ciencias digitales forenses y los procesos de respuesta a incidentes en SDN. Este framework, relaciona y optimiza de manera dinámica algunas fases de los procesos de preparación forense digital y del ciclo de vida de respuesta a incidentes propuestos por la ISO y el NIST, respectivamente.

En consecuencia, en este capítulo se realiza una revisión de la terminología que permitirá comprender con mayor profundidad las ciencias forenses y de respuesta a incidentes. Así también se describirán los principales desafíos existentes para su integración y se presentarán los trabajos relacionados a este particular mundo. Posteriormente se describen en detalle: **(i)** Un framework que conceptualiza la integración los procesos de respuesta a incidentes con los procesos de preparación forense. **(ii)** Una arquitectura extendida que respalda pragmáticamente la asociación entre los procesos de preparación forenses y de respuesta a incidentes en SDN, entregando la descripción pormenorizada de los componentes, interesados e interacciones pertinentes.

3.2 Fundamentos

Antes de entrar en los detalles de las propuestas de este capítulo es importante realizar un repaso de los principales conceptos utilizados en el mundo forense y de respuesta a incidentes a fin de familiarizar la terminología a ser usada.

3.2.1 Ciberataques, ciberdelincuencia, entidades y actos jurídicos

Un ciberataque se define como cualquier acción maliciosa para dañar o comprometer redes de datos, dispositivos, sistemas o información digital, explotando vulnerabilidades tecnológicas. Los ciberataques pueden tener diversos motivos, que incluyen extorsión, manipulación o solicitudes de recompensa económica a cambio de evitar el robo de datos, la interrupción de servicios, la destrucción de información o el acceso no autorizado a sistemas. Los ciberataques pueden ser desarrollados por actores tanto externos como internos. Los actores externos son grupos dedicados a la ciberdelincuencia de manera independiente. Mientras que en el caso de los actores internos se puede referenciar a administradores maliciosos o incluso cómplices de un delito que trabajan dentro de la organización.

Los ciberdelitos son conductas ilícitas o ilegales realizadas hacia o a través de redes y dispositivos tecnológicos. Estas conductas pueden ser contempladas dentro del derecho penal o civil como una falta grave bajo los sistemas jurídicos de cada país o región. Con el transcurrir de los tiempos y el tendiente crecimiento de uso de tecnología, se ha visto la necesidad de crear entidades que apoyen a los sistemas jurídicos y que permitan adoptar medidas regionales o estatales para reaccionar ante un incidente de ciberseguridad. Dado que cada región mantiene su propia visión respecto a un ciberdelito y que su análisis sería extremadamente extenso, en esta tesis doctoral, se considera el manejo de ciberdelitos desde un enfoque europeo.

Dicho esto, el Parlamento Europeo ha entregado la potestad de actuación sobre esta temática

a la Agencia de la Unión Europea para la Ciberseguridad (ENISA, por sus siglas en inglés) que fue establecida en 2004 y fortalecida por el Reglamento de Ciberseguridad de la Unión Europea (EU, por sus siglas en inglés). ENISA contribuye a la política cibernética regional que busca mejorar la confiabilidad de productos, servicios y procesos tecnológicos mediante esquemas de certificación para alcanzar un alto nivel común de ciberseguridad en todos los estados miembros y organismos de la Unión . En este sentido, han existido una serie de actos jurídicos (reglamentos y directivas) que comprenden la intervención de ENISA en el ámbito jurídico y en el operativo, relativos a las redes de datos e incidentes de ciberseguridad. A continuación, se detallan algunos de los actos jurídicos más representativos y afines a esta tesis doctoral.

Es así como en el año 2018, se emitió la Directiva (EU) 2018/1972, por la que se establece el Código Europeo de las comunicaciones electrónicas, entregando un marco jurídico que permite a cada miembro de la EU tomar las medidas para brindar protección de sus intereses esenciales de seguridad y permitir la investigación, la detección y el procesamiento de delitos en términos de seguridad cibernética (European Parliament and European Union Council, 2018).

Mientras que, en el año 2019, se emitió el Reglamento (EU) 2019/881, relativo a ENISA y a la certificación de la ciberseguridad de las tecnologías de la información y la comunicación, en el cual se le confieren responsabilidades y actuaciones sobre los actos jurídicos de la Unión para su contribución en la seguridad de las comunicaciones electrónicas, aportando conocimientos y asesoramiento en esta materia. Con esta premisa, se le ha dispuesto a ENISA apoyar a las instituciones y Estados miembros en la aplicación de políticas para cumplir con las leyes y regulaciones de seguridad cibernética de la Unión (European Parliament and European Union Council, 2019). Además, en el Reglamento (EU) 2019/881, se insta a ENISA a mantener un elevado nivel de conocimientos técnicos para ayudar a los Estados miembros de la EU a mejorar la capacidad y preparación para abordar incidentes relacionados con la seguridad de redes e información, evitando la duplicación de tareas y promoviendo las sinergias regionales. Posteriormente, el 14 de diciembre de 2022, se redacta la Directiva (EU) 2022/2555, mejorando reglamentos y directivas previas, para garantizar que todos los miembros de la Unión contengan un elevado nivel común en la aplicación de métodos de ciberresiliencia. En la Directiva se hace especial énfasis, entre otras cosas, a la gestión de incidentes de ciberseguridad de cada uno de los estados miembros (European Parliament and European Union Council, 2022).

Más tarde, a mediados del 2023 se ha propuesto una reforma al Reglamento (EU) 2019/88, en la cual ENISA tiene el mandato de aumentar la cooperación operativa de gestión de incidentes de ciberseguridad para los Estados miembros de la EU que lo deseen y apoyando en la coordinación de la EU en caso de crisis y ciberataques transfronterizos a gran escala. Cabe indicar que dicha reforma aún se encuentra en revisión.

De manera resumida, la EU ha invertido esfuerzos a nivel normativo de alta gerencia que le permite a la región hacer frente a los ciberataques, considerando su incremento y complejidad, en una sociedad cada vez más hiperconectada. Su enfoque se ha demostrado en la emisión de diferentes actos jurídicos para que cada miembro de la Unión sea capaz de reaccionar en caso de un incidente de ciberseguridad y de ser necesario poder contar con el apoyo de

diferentes organizaciones. Dichas organizaciones, tales como ENISA, claramente deberán tener acompañamiento público y privado para dar continuidad a los objetivos planteados en el marco europeo.

3.2.2 Procesos investigativos y evidencias

Un proceso de investigación se define como un conjunto metódico y organizado de pasos y procedimientos destinados a obtener respuestas a interrogantes específicas o a resolver problemáticas particulares, que pueden ser de cualquier índole, a saber, periodísticas, sociales, académicas, jurídicas, tecnológicas, científicas, etc. A menudo, las interrogantes de un proceso investigativo suelen estar basadas en el método *5W1H* que abarca las preguntas: *what?*, *when?*, *where?*, *who?*, *why?*, *how?*, que en su traducción al castellano es: ¿qué? - ¿qué ocurrió?, ¿cuándo? - ¿cuándo ocurrió?, ¿dónde? - ¿dónde ocurrió?, ¿quién? - ¿quién está involucrado?, ¿por qué? - ¿por qué ocurrió? y ¿cómo? - ¿cómo ocurrió?. Este método ha sido usado ampliamente en varios campos como el periodismo y la investigación, dada la sencillez y precisión de sus preguntas.

Los procesos investigativos son esenciales para garantizar la solidez de los resultados cuando existe un delito o crimen, es por ello que requiere la recopilación de recursos probatorios (evidencias), ya que estas constituirán la base sobre la que se tomarán decisiones importantes. Las evidencias son, por lo tanto, un compendio de información que puede proporcionar indicios entre la ocurrencia de un delito y su autor (Carrier et al., 2003).

El éxito de las investigaciones en el ámbito tecnológico depende en gran medida de la recopilación de una amplia gama de información digital. La ausencia de pruebas o la presentación de pruebas inválidas, puede ser perjudicial en el ámbito legal. Para evitar que las pruebas sean desestimadas en un tribunal, es imperativo que se presenten elementos que cumplan con los requisitos legales para la admisibilidad de evidencias.

Aun cuando la admisibilidad de una evidencia digital dentro de un proceso investigativo depende de la legislación de cada país, existen ciertos principios universales que deben cumplirse para su validez (Leroux, 2004). En este sentido, las evidencias digitales deben ser pertinentes, precisas, completas, consistentes, auténticas, transparentes, y haber tenido una correcta cadena de custodia (CoC, por sus siglas en inglés), para su recopilación, trasmisión y preservación (ENISA, 2017).

Una evidencia es pertinente cuando está directamente relacionada con la acción que se está investigando. Por lo tanto, debe arrojar luces sobre los hechos que están en disputa, por ejemplo, ¿qué ocurrió? o ¿quién está involucrado? Así también para considerar que una evidencia es pertinente ésta no debe ser confusa para quien la examina.

De manera semejante se evalúa la precisión y completitud de una evidencia con el fin de respaldar un argumento, lo que significa contar con elementos relevantes y que consideren un panorama dentro de un período de tiempo significativo. Lo cual puede ser de utilidad para cubrir interrogantes tales como: ¿por qué ocurrió?, ¿dónde ocurrió? o incluso ¿cómo ocurrió?

Por otro lado, se consideran características directamente relacionadas con la integridad, tales como la consistencia, la autenticidad, la transparencia y la vinculación de una CoC. De este

modo, se dice que una evidencia es consistente cuando la misma no presenta contradicciones significativas en los datos presentados, lo que aumenta su fiabilidad y utilidad en un proceso de investigación. En consonancia con esto, se considera la autenticidad, la cual asegura que la evidencia no ha sido alterada, manipulada o falsificada. Siguiendo esta línea y para cerrar el ciclo en la entrega de evidencia, la CoC y su correspondiente almacenamiento es de vital importancia para que una evidencia sea admitida en un proceso investigativo

3.2.3 Eventos e incidentes

Un evento es cualquier ocurrencia observable en un sistema o en este caso en la SDN. Los eventos pueden incluir a un administrador accediendo al controlador, la instalación de una aplicación SDN, la activación de un enlace, el encendido y apagado del controlador, entre otros.

Los eventos pueden o no ser un indicativo de un incidente de ciberguridad. De hecho, se considera un incidente de ciberseguridad a los eventos con consecuencias negativas, como interrupciones de los servicios ofrecidos por la SDN, accesos por fuerza bruta al controlador, uso no autorizado de privilegios, acceso no autorizado a datos sensibles, ejecuciones no requeridas de aplicaciones, agregación de nuevos enlaces sin que se haya aprobado una modificación a la topología, etc (NIST, 2012).

3.3 Ciencia forense digital y Gestión de respuesta a incidentes

3.3.1 Ciencia forense digital

Las ciencias forenses comprenden una serie de metodologías, procesos y procedimientos que utilizan conocimientos científicos y técnicos para recopilar y analizar información vinculada a un delito. Esta información se puede usar como evidencia que ayude a esclarecer procesos investigativos. En sus inicios, las actividades forenses eran llevadas a cabo por profesionales con formación genérica y su uso se limitaba únicamente al campo de criminalística física. No obstante, con el avance tecnológico y la complejidad creciente de ciertos delitos, se volvió necesario contar con especialistas en diversas disciplinas científicas para llevar a cabo investigaciones más detalladas en un mundo cibernético (Raghavan, 2013).

Es así como nace la ciencia forense digital, también llamada informática forense o *digital forensic* en inglés. Esta rama, relativamente nueva, se desarrolló en respuesta al crecimiento de la tecnología de la información y las comunicaciones. Tiene sus raíces en la necesidad de investigar delitos cibernéticos, fraudes electrónicos y otras actividades ilegales que involucran el uso de computadoras, redes de datos e información digital. Sin embargo, la ciencia forense digital no se limita a las investigaciones criminales, pues también puede ser usada para recuperar archivos perdidos y reconstruir información de equipos averiados. De igual manera, es muy útil cuando se trata de determinar el impacto de un evento de seguridad sobre los activos tecnológicos.

Los primeros pasos de la ciencia forense digital comenzaron en la década de los 70, cuando

los investigadores se enfocaron en la recuperación de datos de sistemas informáticos dañados. Posteriormente, en la década de los años 80 y 90, con la popularización de la conectividad, los investigadores desarrollaron técnicas y herramientas para dar seguimiento a actividades sospechosas en línea, lo cual incluía actividades de hacking. Desde el año 2000 a la actualidad, la ciencia forense digital ha ganado mayor presencia con el aparecimiento de organizaciones especializadas, inclusión de estándares y certificaciones en este campo (Raghavan, 2013). En el momento presente, la ciencia forense digital se enfoca en la obtención y conservación de la integridad de evidencia digital proveniente de diversas fuentes de información. Las fuentes de información en tecnología abarcan un gran espectro de elementos que pueden vincularse a hardware, software, datos electrónicos o tráfico de las redes de datos (INTERPOL, 2022).

La ciencia forense digital tiene varias ramas, una de ellas denominada ciencia forense de redes, que radica en el estudio y obtención de evidencias digitales de la red y su tráfico en movimiento (Zinge et al., 2018; Javed et al., 2022). Dado que la información que circula en las redes es volátil y extremadamente dinámica, se apalanca fuertemente en el monitoreo permanente del tráfico y la detección de intrusos. Por lo que, a través de la ciencia forense de redes se puede, entre otras cosas, identificar y corregir vulnerabilidades de seguridad en la red, y mejorar la asignación de los recursos durante un incidente de ciberseguridad.

En este punto es pertinente abarcar el concepto de preparación forense digital o *digital forensic readiness* en inglés, cuya definición está enmarcada en una actividad que permite tener evidencias digitales a partir de un enfoque organizacional. El objetivo de la preparación forense digital es garantizar que las operaciones y la infraestructura sean capaces de apoyar de manera efectiva una investigación, con lo cual se busca maximizar las capacidades de un entorno para el registro de actividades y recopilación de evidencia sólida de manera proactiva, minimizando el impacto sobre los recursos (económico, tiempo de desconexión de un servicio, etc.) durante el proceso forense digital (Elyas et al., 2015).

A lo largo de los años ha existido un uso indiferenciado de los términos de las ciencias digitales forenses y la preparación forense digital. A pesar de la sinonimia recurrente de estos términos, lo que se pretende con su aplicabilidad es asegurar que tanto las operaciones como la infraestructura estén en condiciones de brindar un respaldo completo a una investigación, haciendo que los registros probatorios entregados, cumplan con los principios universales de admisibilidad.

Es importante señalar que para conducir un proceso forense digital existen dos maneras: *live* y *post-mortem*. La primera, también conocida como proactiva o dinámica, se orienta a la adquisición de datos volátiles, es decir que procura obtener información mientras la fuente de información está activa. La segunda, también llamada reactiva o estática, analiza los datos almacenados; es decir, una vez que un incidente se ha producido y las fuentes de información no se encuentren operativas (Conrad et al., 2012; S. Khan; Gani; Wahab et al., 2016).

En la literatura, se han desarrollado diversos modelos que proporcionan directrices para llevar a cabo investigaciones digitales forenses. Estos modelos han evolucionado a lo largo del tiempo para adaptarse a las necesidades cambiantes de diferentes campos de aplicación y a los requisitos legales y sociales (Pollitt, 2007; Mohammmed et al., 2020).

A pesar de la diversidad de estos modelos, la ISO ha considerado varias normas estandarizadas

para la orientación sobre la obtención y tratamiento de las pruebas digitales. A continuación, se listan las normas requeridas para poder brindar lineamientos significativos en un proceso forense digital.

1. **ISO/IEC 27037:2012**, se refiere a la captura inicial de pruebas digitales y ofrece pautas relativas para las actividades de gestión de evidencia digital. Estas actividades abarcan la identificación, la recolección, la adquisición y la preservación de pruebas digitales (ISO, 2012).
2. **ISO/IEC 27041:2015**, se enfoca principalmente en asegurar la calidad de los procesos y herramientas forenses empleados en la investigación de evidencia digital, enmarcados en confiabilidad, credibilidad e integridad (ISO, 2015b).
3. **ISO/IEC 27042:2015**, se refiere a lo que ocurre una vez recogidas las pruebas digitales, es decir, su análisis e interpretación. De esta manera se pretende abordar la continuidad, validez, reproducibilidad y repetibilidad de la evidencia digital (ISO, 2015a).
4. **ISO/IEC 27043:2015**, proporciona directrices basadas en modelos para los procesos comunes de investigación que involucran evidencia digital (ISO, 2015c).

La Figura 3.1 ilustra la perspectiva del manejo de evidencias digitales en un proceso forense considerando las normas ISO.

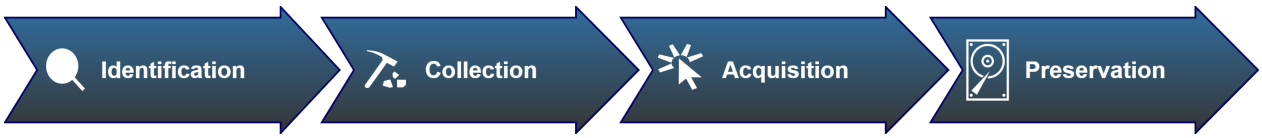

Figura 3.1: Actividades del manejo de evidencia digital conforme la ISO (ISO, 2012; ISO, 2015b; ISO, 2015a; ISO, 2015c)

3.3.2 Gestión de respuesta a incidentes

La gestión de respuesta a incidentes es un proceso esencial en términos de ciberseguridad pues a través de ella se puede garantizar resiliencia efectiva ante un problema de seguridad producido sobre los activos digitales de una organización (Schlette et al., 2021). La gestión eficaz de respuesta a incidentes es fundamental para mantener la continuidad de los servicios prestados por la organización, ya que permite contener posibles ataques, mitigar los daños causados por incidentes de seguridad y restaurar de manera expedita la operación normal de los sistemas y servicios afectados, minimizando de esta manera el impacto sobre los activos y por lo tanto reduciendo costos asociados.

A la par, una gestión adecuada de incidentes de ciberseguridad puede minimizar el impacto sobre la reputación y la imagen pública de una organización ya que al contener ataques o mitigar daños también se protege información crítica y se evita la pérdida de datos valiosos, sensibles y confidenciales, aumentando la confianza de los clientes. Así también, la gestión de incidentes de seguridad permite mantener un aprendizaje constante sobre las experiencias y esto impulsa a la mejora continua, dado que se pueden perfilar de manera más adecuada las políticas internas y procedimientos de seguridad. Esto influye notablemente en la capacidad

de reacción ante futuros eventos de ciberseguridad, pudiendo detectar amenazas y ataques en etapa temprana.

Aunque cada organización puede optar por seguir sus propios parámetros para direccionar la gestión de incidentes, el NIST, provee la guía de manejo de incidentes de seguridad informática que expone un marco integral para navegar por el complejo panorama de la respuesta a incidentes (NIST, 2012). El enfoque sistemático propuesto por el NIST engloba un ciclo representado en la Figura 3.2 compuesto de cuatro fases: preparación, seguida de detección y análisis, contención, erradicación y recuperación, y finalmente actividad posterior al incidente.

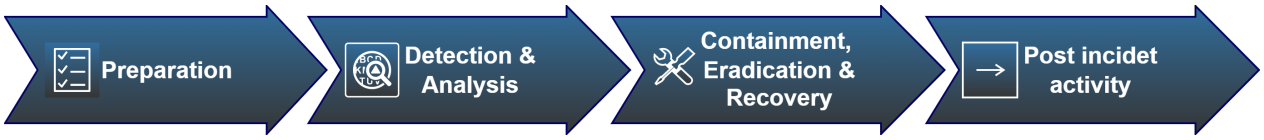

Figura 3.2: Ciclo de gestión de incidentes de acuerdo al NIST (NIST, 2012)

Se han identificado tres etapas claves dentro del ciclo de vida de la gestión de incidentes:

- **Pre-incidente:** En esta etapa se desarrollan los documentos, políticas, estrategias y procedimientos de ciberseguridad que se deberán seguir en caso de un incidente. Así también se describen los roles de los actores o interesados que intervendrán en la respuesta de un incidente. Simultáneamente, se implementan medidas preventivas en el entorno, lo cual incluye: monitoreo, adaptación de controles de acceso, concienciación y formación del personal sobre las mejores prácticas de seguridad.
- **Peri-incidente:** Durante esta etapa continúa el monitoreo de los sistemas o redes. Se activan todos los recursos para la respuesta a incidentes, lo cual incluye personal cualificado para solventar incidentes conforme los roles asignados, uso de infraestructura para contener el incidente, recopilación de pruebas para identificar la causa del incidente, iniciando los procesos investigativos.
- **Post-incidente:** En esta etapa continúa activo el monitoreo con las acciones de contención y se suman las medidas de erradicación y recuperación. De igual manera, se generan reportes o notificaciones a los interesados para tomar nuevas medidas de seguridad que mejoren los niveles de protección.

3.3.3 Integración de procesos de ciencias forenses y respuesta a incidentes

Conforme la información que antecede se observa que existe una gran relación entre los procesos forenses y los procesos de respuesta a incidentes. Esto es debido a que, desde un punto de vista de ciberseguridad, ambos están interesados en la información que permite conocer los orígenes de un incidente, lo cual resulta beneficioso tanto para la parte técnica como para la parte investigativa. Teniendo en mente esta perspectiva, a lo largo de varios años, se ha observado un compromiso significativo tanto por parte de entidades especializadas como de distinguidos académicos en un intento de combinar ambas nociones.

Es así como el NIST a través de una guía presentada en el año 2006, detalló los pasos técnicos para integrar técnicas forenses dentro de los procesos de respuesta a incidentes. En dicha guía se expone una visión estrictamente tecnológica utilizando los procesos forenses para proveer restauración a servicios de red o computacionales (Kent et al., 2006). Durante la revisión de la documentación se observan varias notas que señalan que la guía ha sido desarrollada fuera del contexto legal por lo que es necesario considerar el cumplimiento de los requerimientos universales de admisibilidad de evidencia. Ésta, sin embargo, es una referencia importante en el camino que se está trazando para la integración de ambos conceptos.

Más recientemente, en el año 2017, Johansen, 2017 presenta un libro denominado "*Digital Forensics and Incident Response*", definiéndolo como una manera inteligente de hacer frente a los ataques de ciberseguridad. En el primer capítulo se relata el desarrollo del proceso de respuesta a incidentes, destacando la necesidad de considerar más detalles para asegurar su aplicabilidad. Luego, en el segundo capítulo, se narra el proceso forense digital, incluyendo la adquisición de pruebas y la utilización de un laboratorio forense, la seguridad física de las pruebas y las herramientas de hardware y software que se pueden usar. Aunque en su primer capítulo, el autor deja una nota introductoria de la preparación forense digital, no se detecta una incorporación global de los conceptos. Los capítulos siguientes tratan sobre la colección de pruebas basadas en la red o en los hosts, los procedimientos técnicos de análisis, la elaboración de informes forenses y se examina el papel de la inteligencia de amenazas en la respuesta a incidentes. Cabe mencionar que, en el año 2020 se obtuvo la segunda edición del libro, manteniendo de manera bastante similar lo descrito en su primera edición.

Pero estos no han sido los únicos acercamientos que se han tenido de la fusión de estos dos conceptos. Se ha notado que existen varios campos de la ingeniería que han llegado a involucrarlos en sus propuestas. De este modo, el NIST, emite un documento denominado Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT) (NIST, 2022). El framework propuesto está compuesto de seis fases: rutina, identificación inicial, manejo del evento, análisis y respuesta a un ciberincidente, fin del ciberincidente y post incidente. Tomando en consideración la primera fase de su framework, se nota la preocupación existente por considerar especialmente información de sistemas de control industrial como PLC o equipamiento industrial.

DFIR también ha sido nombrado en el mundo de IoT, es así que los autores Itodo et al., 2021, han explorado los retos de la implementación de DFIR en estos ambientes. Para emitir sus observaciones respecto a los retos, primero mezclan los pasos de los procesos de forense digital y los de respuesta a incidentes, obteniendo una estructura compuesta por fases de preparación, identificación/colección/evaluación de evidencia, contención/erradicación/recuperación, adquisición de evidencia/preservación, examinación de evidencia/análisis, documentación/reporteo y lecciones aprendidas. Los autores conciben su labor como un punto de partida destinado a fomentar una mayor exploración de DFIR en el contexto de entornos IoT, pues argumentan que este estudio aún no ha sido exhaustivamente abordado.

Binnar et al., 2023, generan un análisis para IoT industrial (IIoT) usando procesos digitales forenses y respuesta a incidentes. Generan un modelo denominado DFIR para ambientes de control y adquisición de datos de supervisión (SCADA), teniendo como partida las siguientes etapas: preparación, detección, aislamiento, triaje, respuesta y reporte. A esto le suman

la categorización de las etapas DFIR para sistemas ciberfísicos (CPS) y para sistemas de control industrial (ICS). Es relevante destacar que a pesar de que usan el término DFIR, no se desarrolla claramente un modelo que sugiera su implementación en ambientes IoT. De hecho, en su documento se resalta la necesidad de contar con un DFIR estándar diseñado específicamente para la seguridad de la IIoT dentro de los futuros procesos de fabricación 4.0.

En síntesis, se observa que paradigmas tecnológicos como IoT y disciplinas de ingeniería como la industria 4.0 están adoptando un enfoque unificado de los conceptos relacionados con la ciencia forense digital y la respuesta a incidentes, denominándolo como DFIR. Esto se debe al creciente interés en abordar los desafíos de ciberseguridad, responder eficazmente a los incidentes y mantener registros precisos de los eventos. Ante este panorama, se plantea la interrogante de si SDN, cuyo rol en el despliegue de estos paradigmas y disciplinas de ingeniería es cada vez más destacado, ha evaluado la integración unificada de los conceptos forenses y de respuesta a incidentes para abordar los desafíos de ciberseguridad que puedan surgir en los elementos de su propia arquitectura.

3.3.4 Presencia del mundo de ciencia forense digital y respuesta a incidentes en las SDN

A diferencia de otras ramas de ingeniería y paradigmas en los que ya existe un gran interés por consolidar terminología y practicidad de las ciencias forenses y los procesos de respuesta a incidentes, en las SDN estos campos son poco explorados y su divulgación ha sido tratada de manera individual e incluso mutuamente excluyente, teniendo su adopción en fase de desarrollo. En este sentido, en esta subsección se presentan algunas investigaciones emergentes que han ido allanando el camino para mejorar su aplicación.

Es así como los autores T. Zhang et al., 2016, proponen un marco conceptual forense basado en el concepto de servicio forense bajo demanda, para obtener, limpiar, almacenar e integrar varia información relacionada a un ciberdelito. El documento presenta la descripción de sus principales módulos, objetivos y requisitos de diseño. Dentro de las partes sugeridas se encuentran la adquisición de información, extracción de datos, fusión de información, detección de anomalías, alarma de seguridad y conservación de la evidencia. Los autores señalan que la extracción y fusión de datos está en fase investigativa, y como trabajo futuro planean desarrollar un prototipo de software.

Tras identificar ciertas limitaciones en los métodos de recolección de evidencia en procesos forenses, incluyendo el enfoque estático que no se ajusta a las cambiantes demandas de los entornos SDN, lo cual incide directamente en la confiabilidad de la evidencia en estas redes, los autores Munkhondya et al., 2020 presentan una propuesta de enfoque forense dinámico destinado a los entornos SDN. La propuesta utiliza procesos de identificación y recolección de evidencia tanto automatizados como adaptativos para que únicamente se recoja información asociada a un evento.

Como extensión de esta última propuesta, Lagrasse et al., 2020 proponen un marco de referencia forense con un mecanismo de recogida basado en disparadores IDS para mejorar la eficiencia en la recolección de la evidencia y de este modo reducir los requisitos de almacenamiento. La propuesta presentada se apalanca en estrategias de detección de ataques, usando las políticas

de Snort IDS. Después de analizar detenidamente su propuesta, los autores han reconocido que la implementación del IDS seleccionado en la infraestructura desplegada generó desafíos significativos en términos de escalabilidad.

Para garantizar la cadena de custodia de los datos recogidos, los autores Pourvahab et al., 2019 proponen una arquitectura forense para entornos SDN-IoT utilizando tecnología blockchain. De forma similar Duy et al., 2019, presentan un mecanismo basado en blockchain para asegurar los registros obtenidos de algunos elementos SDN.

Los autores Pandya et al., 2018, proponen un marco para la investigación forense en entornos SDN basados en OpenFlow. En su propuesta, la obtención de evidencias se realiza mediante la captura de paquetes desde la SBI y la adquisición de imágenes de memoria de los switches SDN. Para llevar a cabo la reducción de datos, emplean un método basado en ontología conocido como *Digital Evidences Semantic Ontology* (DESO).

Por otra parte, algunos autores proponen soluciones forenses orientadas al enfoque técnico, aunque sus propuestas no contemplan la entrega de pruebas para su utilización en un proceso legal. Así, H. Wang; Yang et al., 2018 presentan una propuesta para ofrecer diagnósticos para soluciones de seguridad apalancadas en el preprocesamiento de trazas de ejecución orientadas a eventos del plano de control y gráficos de transición de estados del plano de datos.

Mugitama et al., 2020 proponen un conjunto de procesos técnicos destinados a un contexto forense dentro de SDN. Este enfoque incluye el desarrollo de módulos específicos diseñados para recuperar evidencia de los registros del controlador, un aspecto que según los autores ha sido desestimado en la literatura. Tras llevar a cabo pruebas y evaluaciones, se señala que esta propuesta puede ser de gran utilidad para abordar incidentes de seguridad, como ataques DoS y envenenamiento de la topología de la red.

Tras revisar la literatura, se ha detectado que, en el contexto de SDN, los procesos forenses y de respuesta a incidentes se han abordado de manera separada. Por lo general, las soluciones se presentan de forma aislada, ya sea desde una perspectiva legal o técnica. Asimismo, se ha observado que, en la fase de preparación forense, algunos autores se limitan a considerar los registros de transacciones del controlador, pasando por alto otras fuentes de información valiosas. Además, la mayoría de las propuestas se centran exclusivamente en la adquisición y conservación de datos, descuidando el proceso crítico de examinación o corroboración. Esta omisión puede resultar en la entrega de información redundante y extender los tiempos de análisis. Así también, existen propuestas en las que se hacen uso de los componentes de la arquitectura SDN para proponer soluciones forenses en ámbitos como IoT o ambientes en la nube, sin que esto constituya una solución que cubra el ámbito legal y técnico de manera integral para las SDN (Pourvahab et al., 2019; Y. Khan et al., 2021).

Visto este escenario, nace una interrogante importante respecto a ¿Cuáles han sido los obstáculos que han dificultado la adopción holística de las ciencias forenses y los procesos de respuestas a incidentes en el ámbito de SDN?

En consecuencia, en el próximo apartado, se llevará a cabo una revisión detallada de los desafíos identificados que pudieron haber ralentizado la adopción de un marco conjunto de los procesos DFIR en SDN.

3.3.5 Retos identificados hacia la integración de un concepto DFIR en SDN

Para entender la falta de integración de procesos de preparación forense en comunión con los procesos de respuesta a incidentes en SDN, es esencial considerar diversos factores. En primer lugar, se debe llevar a cabo un análisis exhaustivo de los desafíos conceptuales. Luego, se deberá abordar los retos pragmáticos, que están estrechamente relacionados con la implementación de la unión de los procesos de preparación forense y de respuesta a incidentes en SDN. Finalmente, se debería examinar los desafíos de factores externos (sociales, económicos) asociados con la incorporación de procesos de preparación forense y de respuesta a incidentes en las SDN.

Retos conceptuales

Como se puede inferir a partir de lo abordado en la subsección 3.3.4, existe la necesidad de tomar conciencia o conocimiento acerca de la relevancia de las disciplinas forenses en el mundo de las SDN, dado que la falta de interés ha contribuido a la adopción de un enfoque fragmentado con los procesos de respuesta a incidentes. Asimismo, se ha observado cierta ambigüedad en cuanto a los alcances de los procesos de respuesta a incidentes y los procesos de preparación forense digital, lo que ha resultado desfavorecedor en la comprensión de un marco DFIR.

Al mismo tiempo, cabe destacar que SDN es un paradigma en evolución, lo que plantea un desafío significativo para llevar a cabo investigaciones forenses efectivas y para responder rápidamente a incidentes de ciberseguridad en estos entornos ya que se requiere un entendimiento sólido de cómo se gestionan y controlan las SDN. Esto involucra la identificación de fuentes de información y la recopilación de datos ante la presencia de incidentes de ciberseguridad. Lo mismo se aplica a la implementación de medidas de contención o remediación, dada la división de sus capas.

Retos pragmáticos

De igual modo se deben considerar los retos pragmáticos respecto a la vinculación de los conceptos forenses digitales y de respuestas a incidentes en SDN. Iniciando con la implementación de los procesos de preparación forense, pues existe principal preocupación en la cantidad de información que se debe recopilar, ya que cualquier desestimación o sesgo en la identificación para adquisición y el tratamiento de los datos en la entrega de evidencia comprometería la validez de los hallazgos y debilitaría la credibilidad de la investigación. Esto tambien generaría incertidumbre sobre la confiabilidad de las fuentes de información o la sincronización de los registros obtenidos cuando se cuentan con varios controladores SDN, lo que acarrearía consecuencias significativas en el ámbito legal y por consiguiente de la organización.

De igual manera existen desafíos respecto al desempeño de la red cuando se usan mecanismos forenses digitales, ya que dependiendo de los datos recolectados existe una preocupación constante sobre las capacidades de procesamiento de información y almacenamiento de evidencias. Uno de los retos que merece mucha atención y que por lo tanto debe ser correctamente abarcado es su retención considerando los requerimientos de admisibilidad. Esto debido que en muchos casos puede existir modificación de la evidencia, lo cual podría socavar significati-

vamente cualquier esfuerzo realizado para su obtención. De igual manera, se ha visto como una limitación a la falta de un adecuado mecanismo de divulgación de los hallazgos para que se puedan adoptar nuevas estrategias de seguridad y aplicarlo a las medidas correctivas (S. Khan; Gani; Wahab et al., 2016; Karie et al., 2021).

Mientras que en lo que refiere a los desafíos en la implementación de procesos de respuesta a incidentes, las principales preocupaciones se enmarcan en el tratamiento de los eventos de manera individual, la capacidad de reacción ante un incidente, el tiempo invertido en el análisis de la información o el riesgo en el manejo de los falsos positivos. Por otra parte, también existe una preocupación latente en la falta de asignación de roles en el accionar de cada participante involucrado en el proceso (NIST, 2012).

Retos del entorno

Dado que a menudo se desestiman los factores externos que pueden obstaculizar el desarrollo de una propuesta, en esta tesis doctoral se presta una atención especial a las limitaciones de naturaleza social y económica que podrían afectar la inclusión de un DFIR en SDN. Es así que desde una visión gerencial se ha notado que la falta de interés en la formación y desarrollo de personal altamente capacitado en el campo de la seguridad, torna mayor el desafio de la integración de DFIR en la arquitectura SDN. Esta cuestión merece una atención prioritaria, ya que la falta de recursos humanos con experiencia en SDN, así como en procesos forenses y de respuesta a incidentes, puede resultar en la adopción de estrategias de baja calidad o en la falta de alineación de las políticas de seguridad con las normativas jurídicas regionales (Luttgens et al., 2014).

Igualmente, es evidente que el desafío económico radica en la inversión inicial necesaria para la adquisición de equipos y licencias necesarias que garanticen el procesamiento y almacenamiento de información relativa a una evidencia, así como la capacidad de respuesta ante incidentes. Esta inversión puede representar un reto significativo para las organizaciones, a pesar de que la implementación de estas medidas puede traducirse en un ahorro en lugar de un gasto, especialmente cuando se presenta un incidente de gran escala.

Además, es importante considerar las posibles limitaciones sociales que involucran aspectos jurídicos. Estas limitaciones pueden desalentar tanto a la comunidad académica como a la industria a la hora de proponer soluciones integrales en el ámbito de seguridad y justicia. Esto, de manera contraproducente, podría dar lugar a especulaciones sobre la inclusión de un enfoque DFIR y sus potenciales beneficios.

En resumen, la falta de involucramiento, recursos y alineación con los marcos legales pueden ser las principales limitantes que han impedido la consideración holística de las ciencias forenses y de respuestas a incidentes en SDN. Abordar estos desafíos es esencial para avanzar hacia una mayor integración y eficacia en estos campos y para aprovechar plenamente el potencial de SDN, en términos de eficiencia y escalabilidad aun cuando se presente inconvenientes de seguridad. Dicho esto, en la siguiente sección se presenta una de las principales contribuciones de esta tesis doctoral y que intenta cubrir en mayor medida los retos aquí explicados.

3.4 Framework propuesto

Tomando en consideración varios retos que supone la vinculación de los procesos de la ciencia forense digital con los procesos de respuesta a incidentes, en esta tesis doctoral como una de las contribuciones principales se presenta un framework que relaciona, mejora y optimiza de manera dinámica algunas fases de la preparación forense digital y del ciclo de vida de respuesta a incidentes recomendados por la ISO/IEC 27037 (ISO, 2012) y el NIST SP 800-61r2 (NIST, 2012), respectivamente.

Este framework, cuya descripción de alto nivel se encuentra plasmada en la Figura 3.3, tiene como objetivo gestionar adecuadamente la ciberseguridad de las SDN, manejando la información relativa a un evento para entregar evidencias que cuenten con los principios universales de admisibilidad y al mismo tiempo aprovechar la información con la que se cuenta para proveer soluciones técnicas mediante la realimentación del ciclo de vida de la respuesta de incidentes de ciberseguridad en las SDN.

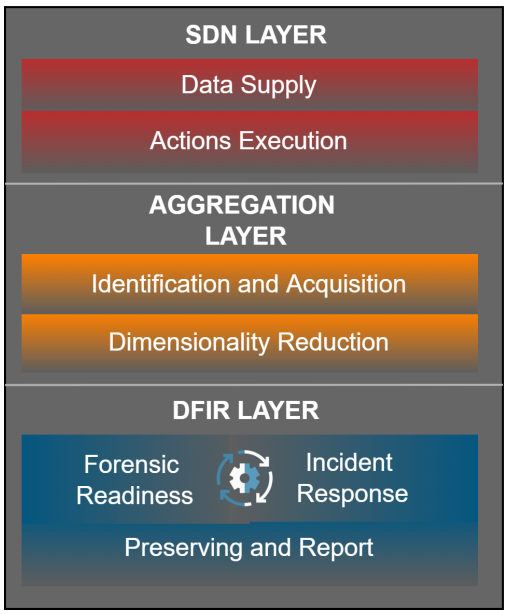

Figura 3.3: Descripción de alto nivel del framework propuesto

Este framework se organiza en capas, las cuales desempeñan funciones específicas en los procesos de la ciencia forense digital y de respuesta a incidentes. Esta estructura en capas aporta claridad a la explicación y simplifica la comprensión de este ámbito técnico, haciendo que el framework sea eficaz para gestionar incidentes de seguridad y las etapas de una preparación forense digital. A continuación, se describe cada una de las capas del framework.

3.4.1 Capa SDN

Esta capa abarca a todos los elementos de la SDN tales como controlador, aplicaciones, elementos de red e interfaces. Cada uno de estos elementos representa una fuente valiosa de información, de la cual se pueden obtener registros significativos para su evaluación en procesos

de investigación, tanto legales como técnicos. Sin embargo, generalmente se ha asociado al controlador como la única fuente de información relevante, considerando sus registros de actividades como evidencia concluyente de un evento de ciberseguridad.

Desafortunadamente, no se puede descartar que los registros del controlador SDN, como cualquier otro, tengan imprecisiones relacionadas con variación de tiempo en su recopilación o carencia de datos. Esto sin duda puede anular la utilidad de la evidencia dentro de un marco técnico, operacional o legal. Por lo que contar con los registros del controlador como única fuente de información para entrega de evidencia, significa disminuir la visión total del entorno y trabajar con un mayor grado de error en la reconstrucción del posible origen del evento, lo cual puede distorsionar la situación real del inconveniente de seguridad y conducir a un escenario diferente y/o erróneo.

Para superar estos inconvenientes y teniendo en cuenta que uno de los principios esenciales para garantizar la validez de la evidencia en un proceso forense radica en que ésta sea completa y consistente, resulta de vital importancia contar tanto con información primaria como con información secundaria, o también denominada corroborativa. Por una parte, la información primaria se refiere a los datos que se recopilan de la fuente principal, que habitualmente se considera la fuente más confiable. Por otra parte, la información secundaria o corroborativa se refiere a los datos que respaldan o confirman la veracidad y precisión de los argumentos de la información primaria. La información secundaria o corroborativa se obtiene de fuentes diferentes a la fuente de la información primaria (Walton et al., 2008).

En este sentido, se ha observado que, aunque los elementos SDN tienen establecidas sus propias tareas independientes, estos se encuentran relacionados entre sí. Por lo que cuando ocurre un incidente en la SDN se genera información encadenada como un "efecto dominó", que puede ser aprovechada (NIST, 2012). Este efecto, permite recuperar tanto la información primaria y la secundaria. De este modo es posible confirmar un incidente de seguridad cotejando datos entre los eventos de varias fuentes de información. Por ejemplo, se puede corroborar si una aplicación se activó bajo pedido de otra, observando los registros específicos de las aplicaciones y contrastándolos con información del controlador.

Debido a lo expuesto, para la consolidación de la evidencia además de trabajar con registros del controlador, este framework aprovecha la separación de los planos en la arquitectura SDN, usando cada elemento (controlador, canales de comunicación, aplicaciones, elementos de red) como una diferente fuente de información.

Las funciones principales de la capa SDN del framework propuesto son:

Provisión de datos: A pesar de que cada elemento SDN entrega datos con su propia denominación, por ejemplo logs, información de los flujos de tráfico, información obtenida de las APIs, etc., en este documento se hará referencia a ellos como datos crudos. Los datos crudos proporcionados por la capa SDN se clasifican como de control o de tráfico. Los datos crudos de control abarcan las acciones de gestión del entorno SDN, tales como inicio de sesión del controlador, activación de aplicaciones SDN, cambio de estado de la topología y aplicaciones, etc. Mientras que los datos crudos de tráfico se refieren al detalle de los flujos de la SDN.

Ejecución de acciones de contención y remediación: Por otro lado, como parte del proceso de respuesta a incidentes, se ejecutan acciones de contención en los elementos de la SDN tan pronto como se detecte un comportamiento inusual en la SDN o acciones de remediación cuando se apliquen nuevas políticas de seguridad.

3.4.2 Capa de agregación de datos

Esta capa es de vital importancia dentro del framework propuesto en esta tesis doctoral, dado que permite recopilar, articular o agregar datos asociados a un evento de manera que se pueda contar con la consistencia de una evidencia y se estructuren los posibles mapas de los eventos de ciberseguridad. En síntesis, esta capa permite reducir la complejidad de los datos crudos para facilitar el análisis y comprensión. Para ello, la capa de agregación debe realizar dos funciones relevantes: identificación y adquisición, y reducción de dimensionalidad.

Identificación y adquisición

Dentro de las SDN existe un gran dinamismo, lo cual genera abundante información que le permite al controlador mantener una visión detallada de las acciones desplegadas en su entorno. Como resultado, circulan varios paquetes, incluso cuando no hay solicitudes de tráfico específicas, que pueden ser o no vinculantes a un evento de seguridad. Entre los diferentes paquetes que atraviesan las SDN, existen aquellos de rutina, que desempeñan funciones para monitorear la salud del sistema, gestionar errores y recopilar estadísticas (Open Networking Foundation, 2015a). Por lo que, al colectar este tipo de información y otra de similares características de múltiples fuentes de información de manera continua, se podría generar una carga adicional en los recursos que gestionan, consolidan y almacenan dicha información, y en consecuencia, también se podría aumentar notablemente el tiempo requerido para llevar a cabo estos procesos y para el análisis de datos.

Esta situación plantea preocupaciones sobre la aplicabilidad efectiva de la ciencia forense digital y de gestión de respuesta a incidentes en las SDN. Además, el exceso de información no relevante o no vinculada a un incidente específico podría dificultar el hallazgo de la causa subyacente del problema. Visto esto, es necesario adoptar acciones o estrategias para llevar a cabo la adquisición de datos.

Considerando lo expuesto, este framework plantea una estrategia de identificación para llevar a cabo la adquisición de registros objetivos vinculantes a un potencial incidente de seguridad. La identificación desempeña un papel crucial para reconocer patrones de comportamiento, tener información de ataques previamente catalogados, permitir el reconocimiento de usuarios para su autenticación y autorización, clasificación de eventos de acuerdo con su importancia, ubicar posibles brechas de seguridad, entre otros.

De este modo, se propone la obtención de datos de manera selectiva, condicionada a la identificación de cambios sustanciales. Este enfoque prioriza la eficiencia y la optimización de recursos al dirigir la recolección exclusivamente hacia eventos o modificaciones que revistan importancia para generar evidencia en el planteamiento forense digital y para gestionar las acciones técnicas durante la respuesta a incidentes. En este punto, una vez que los datos crudos son obtenidos, reciben el nombre de datos colectados.

Reducción de dimensionalidad

Ya con los datos colectados es importante determinar la utilidad de estos mediante la revisión de la coherencia. Esta revisión se realiza a fin de evitar duplicaciones, carencia de datos o eliminación de información para establecer de manera clara un evento de ciberseguridad. Esto también ayuda a minimizar la cantidad de información, reteniendo únicamente aquella que sea relevante, lo cual facilita el análisis de los datos colectados tratando de mejorar la eficiencia computacional. De este modo, si un registro A contiene n entradas, pero de esas entradas solo x tienen vinculación a un evento e, el registro útil no será $A{=}n$ sino $A{=}\ n \mid x \rightarrow e$ (Kotsiantis et al., 2007).

Posteriormente, dado que es necesario mantener información en un formato uniforme para la debida comprensión dentro del hallazgo de la causa raíz, se estandarizan los datos colectados tanto en marcas de tiempo como en estructura de datos para facilitar su procesamiento.

3.4.3 Capa forense digital y respuesta a incidentes (DFIR)

El propósito fundamental de la capa DFIR del framework propuesto es ayudar a determinar la causa raíz de un incidente de ciberseguridad y las acciones llevadas a cabo, tratando de esclarecer dudas respecto a cómo se desarrollaron los eventos subyacentes o para respaldar o rebatir teorías sobre su autoría. Esta capa por lo tanto permite obtener evidencia digital y al mismo tiempo información para la reconstrucción del incidente basado en eventos. De este modo se presentan los hallazgos que pueden ser utilizados en procesos legales o para mejorar la adopción de políticas de seguridad para la SDN.

Dado que en este framework se propone la vinculación de los conceptos de ciencias digitales forenses y de gestión de respuesta a incidentes, ha sido necesario reorganizar y abstraer fases de ambos conceptos, proponiendo una subasignación de capas y la delimitación de los alcances de cada una. A continuación, se detallan las funciones de cada subcapa.

Subcapa de preparación forense

El objetivo principal de esta subcapa es crear una imagen panorámica y comprensible de eventos de ciberseguridad, analizando las relaciones existentes entre los datos colectados que permitan obtener evidencia sólida y generar hipótesis sobre la ocurrencia de un incidente de ciberseguridad. En este contexto, esta subcapa es la responsable de realizar las acciones de consolidación de evidencia y de correlación de eventos.

La consolidación de evidencia agrupa y organiza eventos individuales de ciberseguridad de la SDN de múltiples fuentes de información que fueron colectados por la capa de agregación. En este punto los eventos pueden agruparse de manera cronológica y por origen, de manera combinada. Es así que se agrupan los eventos de las diferentes fuentes de información a través de una reconstrucción de líneas de tiempo, con lo cual se puede verificar la evolución del incidente en relación con el tiempo (Bhandari et al., 2020). La intención es contar con la presentación de los hallazgos en un formato que facilite la toma de decisiones en el ámbito técnico o legal.

Adicionalmente, esta subcapa asume un rol fundamental al llevar a cabo la correlación

de eventos, desempeñando un papel significativo al explorar eventos individuales que, en muchas ocasiones, carecen de contexto y pueden haber ocurrido en situaciones específicas que no necesariamente indican un incidente de ciberseguridad. Por consiguiente, el uso de eventos aislados podría no resultar fructífero al intentar formular hipótesis sobre un incidente. Es por esto que la correlación de eventos se emplea para analizar detenidamente estos eventos individuales, estableciendo conexiones entre ellos, con el propósito de construir una representación comprensiva del problema de ciberseguridad y, de este modo, desarrollar una hipótesis al respecto. Por ejemplo, se puede correlacionar la activación de una aplicación SDN o accesos remotos no autorizados al controlador con el inicio de actividades sospechosas.

Posteriormente, tanto las evidencias como las posibles hipótesis son enviadas a la subcapa de preservación y reporte para que los interesados adopten o descarten nuevas políticas de seguridad.

Subcapa de respuesta a incidentes de ciberseguridad

Cuando se ha identificado un incidente de ciberseguridad en la SDN, es esencial tomar medidas para contenerlo o erradicarlo y finalmente recuperarse de los daños causados. Visto esto, el objetivo de esta subcapa es accionar los mecanismos técnicos necesarios para contener o erradicar los incidentes de ciberseguridad y finalmente, si el incidente ha producido daños en los servicios, activar los procesos de recuperación.

La fase de contención se activa cuando se confirma o se sospecha la existencia de un incidente. De este modo, es posible aislar elementos comprometidos. Bajo esta perspectiva, cuando la capa de agregación ha identificado posibles infracciones de seguridad, la fase de contención efectúa acciones temporales. Por ejemplo, cuando se identifican comportamientos inusuales en el tráfico de red, se desvía el tráfico de la SDN, con lo cual se pretende evitar que las afectaciones escalen hasta el total deterioro de la SDN o la suspensión total de los servicios.

Una vez que se ha logrado contener el incidente, en la etapa de erradicación se generan soluciones definitivas que permiten eliminar la causa raíz del incidente, basadas en las políticas de seguridad establecidas. Estas soluciones pueden abarcar la corrección de vulnerabilidades a través de instalación de parches o actualizaciones, revisar las listas de control de accesos (ACL, por sus siglas en inglés) y los parámetros de la red.

Las acciones de erradicación pueden involucrar a la capa SDN o a la de agregación. En la capa SDN se pueden realizar cambios en las acciones de reenvío de los conmutadores SDN, activación o desactivación de aplicaciones, etc. Mientras que en la capa de agregación se pueden definir nuevos parámetros de identificación estableciendo nuevos niveles de filtrado, apalancados en la visión global de la SDN. Esta fase, se encuentra en constante actualización conforme las políticas de seguridad.

Desafortunadamente, cuando un incidente de seguridad ha escalado a niveles en los cuales se han visto afectados elementos o servicios, es necesario iniciar el proceso de recuperación. En este proceso, se puede recurrir por ejemplo a restablecer los respaldos de las configuraciones del controlador o mantener una infraestructura paralela que sea capaz de soportar una falla.

Por otro lado, considerando que dentro de la investigación de un evento influye en igual

medida lo actuado por el atacante como por el responsable técnico que está trabajando sobre el evento, esta capa recolecta información referente a las acciones técnicas realizadas para contener o remediar un incidente de ciberseguridad, pues esto a más de ampliar el conocimiento futuro para generar políticas y estrategias de seguridad, permite descartar o reafirmar suposiciones sobre la existencia de operaciones tardías, maliciosas o incluso las omisiones por parte de los administradores en la resolución de un incidente de ciberseguridad (Matsumoto et al., 2014). La decisión de esta acción está basada en que no se puede descartar que existan administradores malintencionados que manipulen o eliminen registros vinculantes a un evento (Casey, 2002).

Subcapa de preservación y reporte

La capa de preservación y reporte es la responsable de salvaguardar eficientemente las evidencias que se desprenden de la subcapa de preparación forense, haciendo que se cumplan los lineamientos de integridad y disponibilidad. Adicionalmente, esta capa es la encargada de mantener los registros de las transacciones sobre las evidencias y hacer partícipes de lo que ocurre en el proceso de todos los interesados técnicos y no técnicos vinculados a la SDN. A partir de la información que se almacena en esta capa se generan reportes para cada interesado, conforme asignación por roles.

Una vez que se ha presentado la propuesta conceptual, en la sección subsiguiente se presenta una arquitectura que respalda de manera práctica lo expuesto en este apartado.

3.5 Arquitectura propuesta

Como parte de las contribuciones principales de esta tesis doctoral, se presenta una arquitectura que sustenta la aplicabilidad del framework propuesto anteriormente. La decisión de presentar esta contribución se basa en el reconocimiento de una necesidad común en el desarrollo de frameworks, donde la falta de una representación funcional suele ser una limitación a la hora de decidirse por la aplicabilidad de un marco referencial (Prieto et al., 2023). La introducción de esta arquitectura no solo aborda esta limitación, sino que también mejora significativamente la comprensión del funcionamiento y el valor práctico del framework propuesto.

La arquitectura propuesta se estructura en torno a una serie de componentes, motores, módulos, interesados e interacciones, diseñados con el fin de recopilar evidencia e información esencial para la gestión de incidentes de ciberseguridad. Esta arquitectura prevé satisfacer tanto los requisitos legales como técnicos de una organización, garantizando una respuesta integral y efectiva ante cualquier incidente que suceda en la SDN. La Figura 3.4 presenta una arquitectura que apalanca el modelo propuesto en esta tesis doctoral.

En términos generales, los componentes son elementos tecnológicos formados por módulos y mecanismos con funciones establecidas que interactúan entre sí y con las partes interesadas. Los interesados son quienes requieren información vinculante a un evento de ciberseguridad ya sea para fines legales o técnicos. Finalmente, todo se articula a través de interacciones entre ellos. En la Figura 3.4 se detalla el funcionamiento de cada una de las partes de esta arquitectura.

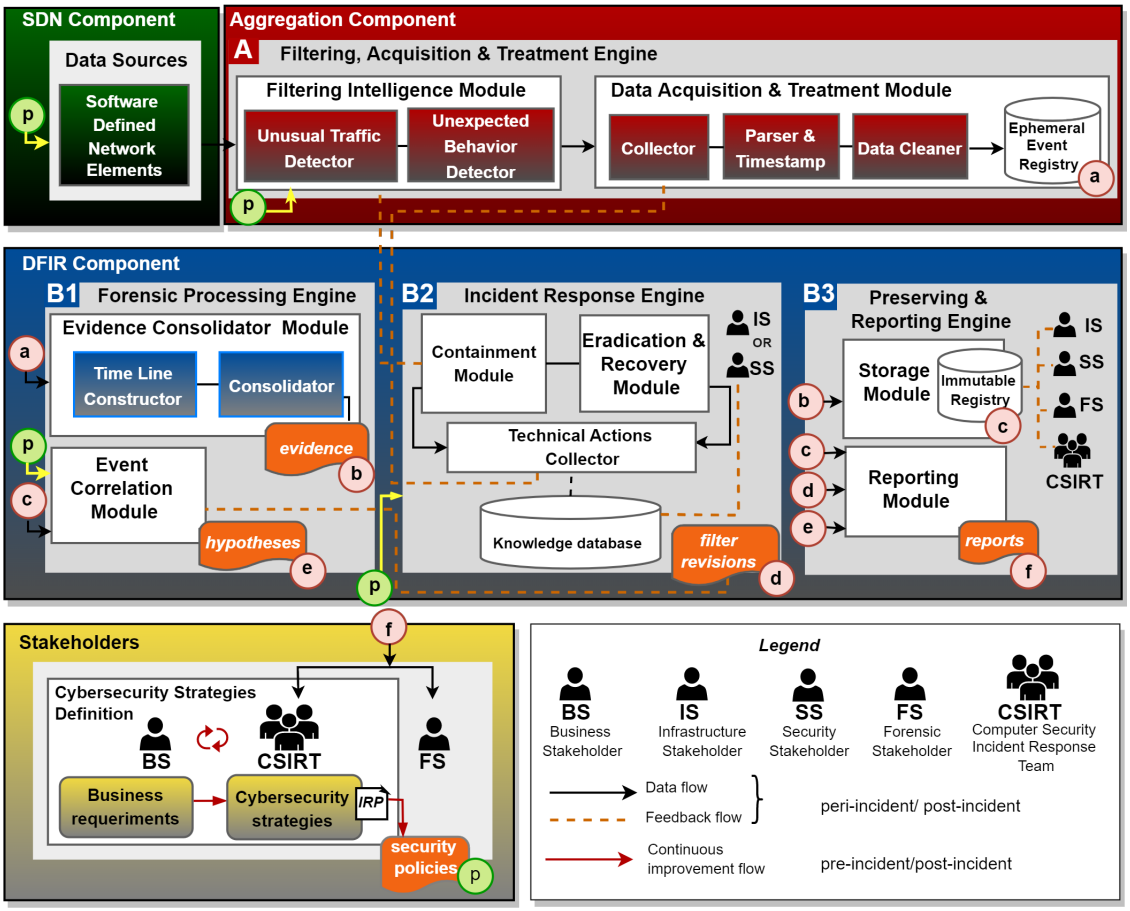

Figura 3.4: Arquitectura DFIR propuesta

3.5.1 Componentes

Los componentes descritos a continuación desempeñan un papel fundamental dentro de la arquitectura propuesta. El diseño de cada uno de los componentes está basado en la usabilidad, flexibilidad y adaptabilidad que requiere el framework a fin de cubrir las brechas pendientes en el ámbito de la ciencia forense y de respuesta a incidentes de ciberseguridad en SDN.

Componente SDN

El componente SDN está formado por el controlador, como una parte fundamental, las aplicaciones, los elementos de red y las interfaces asociadas. Sobre este componente se llevan a cabo las acciones técnicas que se han definido en las políticas de seguridad ⓟ. Este componente desempeña un papel crucial al proporcionar información al motor de filtrado, adquisición y tratamiento.

Componente de agregación

Este componente cuenta con el ***Motor de filtrado, adquisición y tratamiento***, el cual a su vez está conformado de dos módulos: Módulo de inteligencia de filtrado y módulo de

adquisición y tratamiento de datos, el cual incluye un registro de eventos efímero.

- **Módulo de inteligencia de filtrado:** Dado que la información para una evidencia debe ser, entre otras cosas, pertinente y precisa, este módulo es un disparador de inicio automático de la adquisición de datos crudos. Con esto se pretende recopilar únicamente datos relevantes y no información de rutina. Por lo expuesto, este módulo contiene dos mecanismos de detección:

  Detector de tráfico inusual: Este mecanismo identifica los comportamientos inusuales en el tráfico de la SDN, escuchando la información de los paquetes entrantes y evaluándolos conforme lo determinado en las políticas de seguridad ⓟ.

  Detector de comportamientos inesperados: Este mecanismo compara continuamente los estados de los elementos de la SDN. De este modo cada vez que existe una alteración en uno de los elementos SDN, se adquieren datos de todas las fuentes de información relacionadas al elemento.

Una vez que el módulo de inteligencia de filtrado ha identificado comportamientos inesperados de los elementos SDN, en la administración del controlador o se ha detectado un tráfico inusual, comienza la adquisición de los datos crudos. De igual manera, este módulo mantiene informado al motor de respuesta a incidentes sobre un evento, para que, en caso de ser necesario, se activen sus mecanismos.

- **Módulo de adquisición y tratamiento de datos:** Este módulo está conformado de tres mecanismos: Colector, intérprete y marcas de tiempo; y un depurador de datos.

  Colector: Este mecanismo es el encargado de recopilar todos los registros de las diferentes fuentes de información.

  Intérprete y marcas de tiempo: Es el encargado de controlar que todos los datos colectados y la información de las acciones técnicas provistas por el motor de respuesta a incidentes tengan una fecha, hora y formato estándar. En caso de que un registro llegue sin estos campos, este módulo los asignará. Al mismo tiempo este módulo estandariza los formatos de todos los datos colectados asignando un solo tipo para que sea de fácil manejo y establece un etiquetado a la información de las acciones técnicas a fin de poder distinguirlas.

  Depurador de datos: Se encarga de la limpieza de los datos, descartando registros que no aportan al proceso de preparación forense o de respuesta a incidentes. En este sentido, se discriminan registros que no contienen suficiente metadata, registros duplicados o aquellos que tienen exceso de ruido.

Una vez que se ha validado que todos los datos colectados se encuentran estandarizados y limpios se genera un almacenamiento local de manera temporal en el registro efímero de eventos, mientras son procesados ⓐ. El registro efímero mantendrá una configuración de sobreescritura, por lo que una vez que se haya comprobado que se ha consolidado una evidencia, los datos serán eliminados.

Componente DFIR

El componente DFIR es el encargado de gestionar los procesos de preparación forense y de respuesta a incidentes de ciberseguridad de la SDN. Este componente está conformado de tres motores: Motor de procesamiento forense, motor de respuesta a incidentes y motor de preservación y divulgación.

Motor de procesamiento forense: Está compuesto de dos módulos: Módulo consolidador de evidencia y módulo de correlación de eventos.

- **Módulo consolidador de evidencia:** Consume la información del registro efímero de eventos ⓐ y es el encargado de presentar datos como evidencia, para ello este módulo trabaja con dos mecanismos: Constructor de líneas temporales y consolidador.

  Constructor de líneas temporales: Desde una perspectiva forense, las líneas de tiempo arrojan luces sobre el inicio, duración y fin de una o varias acciones que pueden o no estar asociadas a un incidente de seguridad. De este modo, es posible ubicar todos los eventos que han ocurrido de manera cronológica; donde los eventos pueden ser la activación de un equipo, los inicios de sesión, eliminación de registros, entre otros (X. Lin, 2018). Conociendo que la arquitectura que se propone es modular y que las SDN permiten despliegues flexibles a través de controladores distribuidos físicamente en diferentes zonas horarias, el manejo de la temporalidad puede suponer un gran desafío. En este sentido, es imprescindible tener en cuenta la precisión de los relojes a través de una sincronización de cada uno de los componentes de esta arquitectura, ya que las pequeñas discrepancias pueden dificultar la creación de una secuencia cronológica exacta de las acciones.

  Consolidador: Genera un compendio de información sólida, haciendo uso de los datos desprendidos de todas las fuentes de información de la SDN. Este compendio está compuesto de las principales características de los datos y que ha sido construida en una línea de tiempo. En este punto dicha información es tratada como evidencia ⓑ, la cual es enviada al motor de preservación y divulgación.

- **Módulo de correlación de eventos:** Este módulo permite construir posibles hipótesis ⓔ sobre el origen del incidente de seguridad. Para ello, aprovecha los datos de los registros de las evidencias ⓒ y el histórico de comportamiento del filtrado ⓓ, tomado desde el motor de respuesta de incidentes. Este módulo considera las reglas de correlación que se definen en las políticas de seguridad ⓟ y que permiten identificar patrones y tendencias de la conducta de la red de manera global. Dichas reglas trabajan sobre las relaciones de dependencia e independencia condicional existentes entre los eventos de ciberseguridad.

Motor de respuesta a incidentes: Es el encargado de gestionar técnicamente los incidentes de ciberseguridad de la SDN. Se apalanca fuertemente en las políticas de seguridad ⓟ, las cuales están basadas en un plan de respuesta a incidentes (IRP, por sus siglas en inglés). Entre las múltiples medidas que se establece en un IRP, se encuentra presente un análisis de riesgos para clasificar los incidentes según su impacto. De esta manera, se consideran los procedimientos para contener o erradicar el incidente, delimitando lo tiempos para realizar

cada acción.

Es importante destacar que este motor de respuesta a incidentes se alinea con marcos de gestión establecidos, como ISO 27000, ITIL y COBIT. En este sentido, en esta propuesta por cada elemento de la SDN (controlador, elementos de red, aplicaciones e interfaces) se contempla la evaluación del riesgo, a fin de asignar un nivel de respuesta apropiado (*N, N+1*).

Este motor trabaja con un módulo de contención, un módulo de erradicación y recuperación, con un colector de las acciones técnicas y por último con una base de conocimiento, explicados a continuación.

- **Módulo de contención:** Este módulo gestiona acciones técnicas definidas en un nivel *N*, por cada tipo de incidente manteniendo comunicación constante con el módulo de inteligencia de filtrado y con la base de conocimientos. De igual manera retroalimenta constantemente al módulo de erradicación y recuperación. Por citar ejemplos de contención se tienen: desvío de tráfico, desactivación temporal de una aplicación SDN, restricción temporal de permisos de acceso al controlador.
- **Módulo de erradicación y recuperación:** Este módulo considera información del módulo de contención y reevalúa el impacto residual. Conforme el comportamiento o evolución del incidente se pueden tomar medidas técnicas de nivel *N+1*, tales como el bloqueo de tráfico proveniente de un segmento de red específico, desactivación permanente de puertos, desactivación permanente de aplicaciones, retiro de permisos de acceso al controlador. De igual manera este módulo es capaz de reestablecer configuraciones de respaldo.

Cada acción técnica realizada tanto en el módulo de contención como en el de erradicación y recuperación es captado por el colector de acciones técnicas, para que pueda ser consumido por el módulo de adquisición y tratamiento de datos y posteriormente se lo incluya en el proceso de preparación forense. Simultáneamente, se alimenta una *base de conocimiento*, la cual es utilizada por el personal técnico de infraestructura o de ciberseguridad con el fin de generar diferentes procesos de solución de problemas, con lo cual se intenta asegurar que se han seguido los protocolos necesarios para solventar un incidente.

En este motor también se analiza la efectividad de los filtros aplicados sobre el tráfico ⓓ. Para ello se toman algunos indicadores tales como valores estadísticos de la cantidad de tráfico que circuló en la red durante un periodo de tiempo definido, la cantidad de veces que el motor de respuestas se ha activado conforme los filtros establecidos y las estadísticas de salud de la red en fases pre-incidente y post-incidente. Así, los hallazgos se emiten al módulo de reporte para que los interesados puedan considerarlo en la toma de decisiones y afinamiento de las estrategias de seguridad en la SDN.

Motor de preservación y divulgación: Una de las preocupaciones más importantes de cualquier proceso forense es el cumplimiento de una adecuada CoC. Ésta inicia desde el momento de la identificación de fuentes de información y se extiende hasta el momento mismo de la entrega de la evidencia. Dado que, como se ha explicado, la evidencia debe mantener ciertos parámetros para ser considerada válida dentro de un proceso investigativo, esta propuesta considera como un factor importante el almacenamiento de la evidencia. Del

mismo modo dado que es necesario que cada interesado sea de la parte técnica o legal, tenga conocimiento de lo ocurrido durante un incidente de seguridad en esta arquitectura se prevé un módulo de reporte.

A continuación, se detallan los módulos de almacenamiento y de reporte que son parte del motor de preservación y divulgación.

- **Módulo de almacenamiento:** Este módulo está pensado en completar la CoC, garantizando la integridad, confiabilidad y disponibilidad de la evidencia. Para ello se recurre a la inmutabilidad y al registro descentralizado. La idea es mantener múltiples copias idénticas del registro de las evidencias, almacenados en diferentes ubicaciones, de este modo si un nodo falla o es comprometido, las evidencias aún están disponibles en otras localizaciones.
- **Módulo de reporte:** Este módulo se encarga de entregar los registros de las evidencias ⓒ y la información relativa a las posibles hipótesis del origen del incidente de ciberseguridad ⓔ a los interesados , considerando el estado de los filtros que se desprende del motor de respuesta a incidentes ⓓ. La presentación de los reportes hacia los interesados se realiza de una manera comprensible para que se puedan tomar las acciones necesarias sobre las políticas de seguridad ⓟ.

3.5.2 Interesados

Toda acción de gestión tiene relacionado un componente humano. En este caso las partes interesadas comprenden individuos y grupos que deben tener presencia durante los procesos de la arquitectura propuesta. En este sentido se han considerado los siguientes interesados:

Del negocio (BS): Conoce el giro del negocio y tiene claros los objetivos empresariales y el entorno gerencial de la organización. Tiene suficiente poder de decisión para aportar información cuando se desarrollan las políticas de seguridad. Tiene capacidad para disponer de diferentes recursos dentro de todo el proceso DFIR (económico, tecnológico, humano).

De la infraestructura tecnológica (IS): Es el personal técnico encargado de administrar la infraestructura tecnológica. En este caso, es quien tiene mayor conocimiento del componente SDN. Por lo que, conforme las estrategias de seguridad, es capaz de realizar determinadas acciones de configuración sobre los elementos SDN. Aporta conocimiento sobre los activos de información de la SDN en la evaluación de riesgos y elaboración del IRP, por consiguiente, evalúa las necesidades futuras de capacidad y los niveles de escalamiento de un incidente *(N, N+1)*. Sin embargo, no puede manipular el colector de acciones técnicas de manera autónoma.

De la seguridad tecnológica (SS): Comprende el personal técnico encargado de velar por la ciberseguridad de la SDN. También es capaz de proponer acciones de reajuste sobre los procesos de filtrado y reglas de correlación. Se encuentra activamente revisando que la base de conocimiento se mantenga actualizada. Al igual que el interesado de infraestructura tecnológica, este interesado tampoco puede manipular autónomamente el colector de acciones técnicas.

Del ámbito forense (FS): Es un interesado debe tener habilidades técnicas en el ámbito forense digital. Desempeña un papel crucial en la investigación de incidentes de ciberseguridad pues es quien recibe los reportes de los hallazgos, por lo que debe velar por el cumplimiento de la CoC de la evidencia hasta su presentación para que pueda ser validada en un juzgado. Dentro de la arquitectura propuesta, este interesado puede operar como un observador del proceso DFIR. De este modo, cuando se necesitan realizar trabajos de mantenimiento sobre el colector de acciones técnicas, por parte de los IS o de los SS, este interesado puede dar su criterio, generando un consenso en caso de discrepancias, asegurando que se sigan los debidos lineamientos y que no existan vicios dentro del proceso.

Equipo de respuesta a incidentes de seguridad informática(CSIRT): Es un grupo especializado de profesionales del área tecnológica encargado de gestionar y orquestar adecuadamente todas las acciones de los procesos de preparación forense digital y de respuesta a incidentes. También es la primera línea de contacto con el interesado del negocio. Alineado con las estrategias de seguridad, es el responsable inmediato de IRP, por lo que debe liderar la evaluación de riesgos.

Entre el BS y el CSIRT se establecen las estrategias de ciberseguridad que darán origen al IRP, ya mencionado. El IRP delinea las políticas de seguridad que se aplican durante todo el ciclo de vida de los procesos involucrados. De este modo, se definen los procedimientos y herramientas para responder en caso de un incidente, se establecen los parámetros de filtrado y los ajustes que puedan requerirse, se determinan las acciones a efectuar sobre los componente SDN y como se explicó anteriormente se puede analizar el impacto para adoptar medidas de acción ante un incidente de ciberseguridad.

3.5.3 Interacciones

La presente arquitectura involucra múltiples interacciones entre sus diversos componentes, motores, módulos e interesados. El primer conjunto de interacciones pertenece a la información cruda que llega desde el componente SDN al módulo de inteligencia de filtrado en componente de agregación y que puede ser derivada para el detector de tráfico inusual o al detector de comportamientos inesperados, dependiendo si es información de tráfico o información de control. En caso de detección, esta información es conducida al módulo de adquisición y tratamiento y paralelamente se genera una reacción en el módulo de contención del motor de respuesta a incidentes. Por último, este conjunto de interacciones finaliza en la entrega de datos limpios y estandarizados al motor de procesamiento forense.

Posteriormente, se presentan interacciones en el componente DFIR. En primer lugar, en este componente sus motores mantienen retroalimentación de los datos que se desprenden de una evidencia como se observa en la Figura 3.4. En el motor de procesamiento forense, los datos de evidencia son canalizados para el módulo de almacenamiento y posteriormente consumidos por el módulo de correlación de eventos. En lo que respecta al motor de respuesta a incidentes, éste mantiene interacción constante con el módulo de correlación, para mantener información de la efectividad de los filtros. Esta última información también es compartida al módulo de reporte que más tarde será emitida a los interesados. Ambos módulos del motor de respuesta

a incidentes interactúan con los interesados IS o SS, con la base de conocimiento y con el colector de acciones técnicas. Así también, se generan interacciones entre el colector de acciones técnicas con el módulo de adquisición y tratamiento de datos y la base de conocimiento.

La tercera área de interacción destaca por la eficaz comunicación entre el motor de procesamiento forense, el motor de respuesta a incidentes y el motor de preservación y divulgación. En este contexto, el último motor recibe información respecto a las evidencias, las hipótesis y las revisiones de los filtros, para ser transaccionada por los interesados o para generar reportes.

Finalmente, existen interacciones que están estrechamente relacionadas con la gestión, de este modo una vez que se reciben los reportes, es posible redefinir políticas de seguridad para la SDN. En este sentido, los BS interactúan directamente con el CSIRT para emitir estrategias y políticas de seguridad basados en los requerimientos del negocio. De igual manera, durante todo el proceso de preparación forense se interactúa con un FS, quien en algunos casos actúa como observador; por ejemplo, en el proceso de CoC o cuando se requiere dar mantenimiento al colector de acciones técnicas.

3.6 Resumen del capítulo

En este capítulo, se presentaron dos contribuciones importantes de esta tesis doctoral: un framework diseñado para gestionar la preparación forense y la respuesta a incidentes relacionados con las SDN, y una arquitectura basada en el framework propuesto.

Para el desarrollo del framework, se realizó un análisis exhaustivo de los desafíos actuales en estas disciplinas, tomando como referencia principal los estándares de la ISO y el NIST. Aunque estos organismos abordan algunos aspectos de manera individual, se identificó la necesidad de una integración orgánica de los conceptos para mejorar significativamente la gestión ciberseguridad en SDN.

Por otra parte la arquitectura está diseñada para complementar y ampliar los componentes de ingeniería de esta tesis. Esta arquitectura está basada en el framework propuesto y se detallan exhaustivamente sus componentes, partes interesadas e interacciones, con el objetivo de proporcionar una visión clara y completa de su funcionamiento subyacente.

Capítulo 4

MODELOS PREVALENTES

4.1 Introducción

En el capítulo anterior, se presentó el framework para la integración de procesos de preparación forense y de respuesta a incidentes para SDN. Además, se expuso el diseño arquitectónico que abarca una variedad de componentes, motores y módulos necesarios para lograr dicha integración de manera pragmática. No obstante, dada la extensión y complejidad del framework y de la arquitectura propuesta, se ha priorizado el desarrollo de dos modelos claves, siendo estos la inteligencia de filtrado con la adquisición y tratamiento de datos, y la preservación de la evidencia.

Es por ello que, en este capítulo se presentan dos contribuciones específicas que abordan los desafíos inherentes a la identificación de eventos, junto con la adquisición y tratamiento de datos y al almacenamiento de evidencia. Estas áreas han sido seleccionadas debido a los retos inherentes durante la selección de información relativa a un incidente de seguridad y la posibilidad de manipulación de la evidencia, lo que dejaría sin efecto todo el trabajo realizado.

4.2 Modelo de inteligencia de filtrado, adquisición y tratamiento de datos

Al abordar el ámbito de los procesos de preparación forense digital y de respuesta a incidentes en SDN, es inevitable enfrentarse al manejo de una amplia cantidad de información. Tomando en cuenta esta preocupación, la capacidad de identificar eficazmente comportamientos inusuales o eventos relevantes que pueden escalar a un incidente de ciberseguridad se convierte en un factor determinante durante una implementación.

Por lo expuesto, el framework presentado en el capítulo 3 subraya la importancia de aplicar estrategias de identificación para el filtrado, ya que esto permitiría obtener una selección cuidadosa de datos vinculados al incidente, ya sea para presentar evidencias en el ámbito jurídico o para descubrir la causa raíz de un problema de ciberseguridad y proporcionar una solución técnica, por lo que representa un punto de impacto crucial en el rendimiento

de toda la propuesta. Esto desde luego no solo mejora y acelera el proceso subsecuente de adquisición de datos, sino que también alivia la carga de datos superfluos que podría dificultar una investigación (Reith et al., 2002). Además, al desarrollar una adecuada identificación es posible centrarse en otras acciones críticas de la propuesta, como la entrega de evidencia, la implementación de soluciones técnicas y el almacenamiento de evidencia respetando la CoC.

En este sentido, en este apartado se detallan las estrategias que se plantean tanto para la identificación de los comportamientos inusuales de tráfico como para la identificación de los comportamientos inesperados de los elementos SDN. La función de estos identificadores, radica en servir como un desencadenante para la adquisición y tratamiento de datos que darán inicio tanto el proceso de preparación forense como el de respuesta a incidentes.

Dado que se trata de una propuesta de ingeniería, es importante presentar la terminología utilizada. A continuación, se entrega una descripción general de las tecnologías que se consideran para estos identificadores.

4.2.1 Terminología

Inteligencia artificial, aprendizaje automático y aprendizaje profundo

La inteligencia artificial (AI, por sus siglas en inglés) es un campo de la informática que se enfoca en desarrollar sistemas y tecnologías para realizar tareas que normalmente requerirían la presencia de la inteligencia humana. La AI se centra en la creación algoritmos capaces resolver problemas, comprender el lenguaje natural, reconocer patrones y tomar decisiones autónomas. Dado su potencial, ha sido ampliamente usado en campos de la ingeniería, la medicina, logística, finanzas, etc. Cuenta con varias subramas, siendo las más predominantes el aprendizaje automático, redes neuronales, procesamiento de lenguaje natural, visión por computadora, entre otros (Campesato, 2020). El aprendizaje automático (*machine learning*, en inglés), es una rama de la inteligencia artificial que trabaja con algoritmos y modelos basados en matemática y cálculo estadístico. El aprendizaje automático permite a las computadoras tomar decisiones en base a un entrenamiento previo con un conjunto de datos que contiene patrones con etiquetas conocidas. A través del análisis de estos datos iniciales, el algoritmo aprende a reconocer patrones y hacer predicciones o tomar decisiones sobre nuevos datos (Campesato, 2020).

El aprendizaje profundo (*deep learning*, en inglés) por su parte, es una rama del aprendizaje automático que usa redes neuronales, las cuales imitan la arquitectura y el funcionamiento de una red neuronal biológica del cerebro humano. Este tipo de aprendizaje es capaz de descifrar nueva información comparándola con la ya conocida, por lo tanto, puede entrenar algoritmos que se adapten automáticamente a los datos de entrada, lo cual permite mayor escalabilidad y precisión en la resolución de problemas complejos (Campesato, 2020).

Existen algunas ventajas que tiene el aprendizaje profundo sobre el aprendizaje automático. Por una parte, la autonomía que maneja el aprendizaje profundo es mayor que la del aprendizaje automático. Esto debido a que el aprendizaje automático requiere mayor intervención humana a diferencia del aprendizaje profundo que está pensado para autogestionarse. Por otra parte, el aprendizaje profundo, dada la complejidad simulada del cerebro humano, permite trabajar

con datos no estructurados y entregar resultados más precisos. Sin embargo, en el contexto del aprendizaje profundo, surgen dos cuestiones que generan incertidumbre al considerar su adopción. Este tipo de aprendizaje está concebido como una caja negra, por lo que se desconoce claramente su comportamiento interno. Además, se presenta el desafío del sobreajuste, que ocurre cuando la red neuronal no logra generalizar un modelo de inteligencia artificial debido a su tendencia a la memorización de datos en lugar de aprender de la experiencia. Esto por supuesto está estrechamente relacionado con el conjunto de datos que le es dado en el proceso de entrenamiento (Campesato, 2020).

Aún con todas estas ventajas y desventajas, en los últimos años ha habido un gran interés en el uso de estas tecnologías en el mundo forense digital y por supuesto en la resolución de incidentes de ciberseguridad. Esto se debe a que, al existir gran cantidad de información digital relacionada a un evento de ciberseguridad, los procedimientos actuales que dependen de la habilidad humana pueden llegar a tardar demasiado y pueden ser más propensos a errores y sesgo (Dunsin et al., 2023). Autores como Guarino, 2013 o Jarrett et al., 2021 reconocen el potencial de AI en las ciencias forenses digitales dado que mejora la precisión y consistencia de las investigaciones, lo cual incide sobre la confiabilidad de la evidencia. Afirman que, al automatizar y simplificar diversas acciones relacionadas con la identificación, el análisis de datos o el reconocimiento de patrones se puede establecer conexiones que pueden no ser discernibles mediante métodos humanos convencionales. Además, Jarrett está firmemente convencido de que la aplicación de AI en una investigación permite hallar respuestas a cuestiones de relevancia jurídica de manera más eficiente y económica. Mientras en lo que refiere a respuesta a incidentes, el incluir algoritmos de AI en los procesos de identificación de incidentes y procesamiento de datos se pueden reducir el número de falsos positivos, pudiendo limitar futuros incidentes dado el conocimiento de patrones y al igual que en los procesos forenses, se pueden mejorar los tiempos de respuesta para encontrar la causa raíz y brindar soluciones.

Máquinas de estado finito

Las máquinas de estado finito (FSM), son un modelo que representan una máquina hipotética con uno o más estados, donde solo uno puede estar activo. El cambio entre estados también llamados transiciones, ocurre en respuesta a eventos específicos. Un ejemplo cotidiano de una FSM es la operación de encender y apagar una luz mediante un interruptor, que transita entre los estados encendido y apagado. Cada estado en una FSM requiere entradas y salidas definidas, junto con transiciones que especifican cómo la máquina evoluciona.

4.2.2 Desarrollo del modelo de inteligencia de filtrado, adquisición y tratamiento de datos

Este modelo es un punto sensible de la arquitectura propuesta en esta tesis doctoral. Autores como Reith et al., 2002, argumentan que todas las fases de una investigación forense dependen en gran medida en la identificación de una violación de seguridad, por lo que se debe integrar mecanismos de detección. Dicho esto, se ha tornado imperativo delinear con precisión el enfoque de identificación que regirá el modelo propuesto. En este sentido, después una minuciosa

revisión de varias opciones como identificación basada en firmas, basada en heurística y basada en anomalías, se ha seleccionado esta última como estrategia a usarse para el desarrollo de este modelo. Esta estrategia se centra en la identificación de comportamientos inusuales que se apartan de los patrones establecidos como normales. Al observar de manera continua desviaciones significativas en el comportamiento habitual, se puede detectar posibles amenazas que podrían estar relacionadas con eventos de ciberseguridad.

Se ha tomado en cuenta este enfoque dado que es altamente útil en entornos dinámicos donde las amenazas pueden evolucionar constantemente, y como ya se ha explicado anteriormente, el despliegue de la arquitectura SDN es generalmente en entornos altamente dinámicos, por lo que en esta tesis doctoral se han considerado dos tipos de identificadores de anomalías: uno de tráfico inusual que trabaja directamente con el flujo de datos de la SBI y otro que identifica los comportamientos inesperados de los elementos SDN, el cual trabaja con información de las aplicaciones, la topología, los dispositivos, flujos y enlaces y de acceso al controlador. A continuación, se detalla el funcionamiento de cada uno de ellos.

Identificador de tráfico inusual

En virtud de la elección de una estrategia de identificación basada en anomalías, se ha requerido la definición del punto medio de normalidad, actuando como un estado de referencia que evalúa el comportamiento habitual del tráfico. Para ello, se han establecido umbrales de detección que permitan identificar patrones de comportamiento fuera de lo común. Los umbrales se clasifican principalmente en dos categorías: estáticos y dinámicos.

Por una parte, los umbrales estáticos tienen un valor fijo y no cambian automáticamente en función de las condiciones del entorno. Estos umbrales generalmente son usados en ambientes donde existe información preexistente y por lo cual es fácil establecerlos. Los umbrales estáticos pueden estar fundamentados en comportamientos históricos del tráfico, por ejemplo, la cantidad de paquetes entrantes y salientes en una hora determinada. Por granularidad por perfiles, lo cual está relacionado con la especificidad de los accesos por cada usuario. Y por frecuencia, por ejemplo, la cantidad de intentos de acceso a los servicios de red como en una VPN.

Por otra parte, los umbrales dinámicos, también llamados adaptativos, se ajustan en respuesta a patrones y cambios en el tráfico. A diferencia de los umbrales estáticos, propuestos por varios autores en el contexto de procesos de filtrado, la utilización de umbrales adaptativos tiene el potencial de minimizar el nivel de sesgo y asegurar la recolección de datos de manera más precisa al ajustarse dinámicamente a las condiciones cambiantes del entorno. Así también los umbrales adaptativos permiten maximizar la tasa de verdaderos positivos y disminuir la tasa de falsos positivos (Chae et al., 2019).

En este contexto, y considerando que la mayoría de despliegues de la arquitectura SDN se efectúa en entornos altamente dinámicos, la integración de umbrales estáticos en un proceso de filtrado llevaría a tener una red demasiado permisiva o restrictiva. Por lo que, el modelo planteado en esta tesis para identificar el tráfico en el canal de comunicación (SBI), incorpora umbrales adaptativos basados en esquemas predectivos que aplican algoritmos avanzados de inteligencia artificial.

Una vez explicado el principio elegido para el filtrado de tráfico a continuación se detalla el funcionamiento del mismo. De igual manera, en la Figura 4.1 se visualiza el detalle del identificador de tráfico inusual.

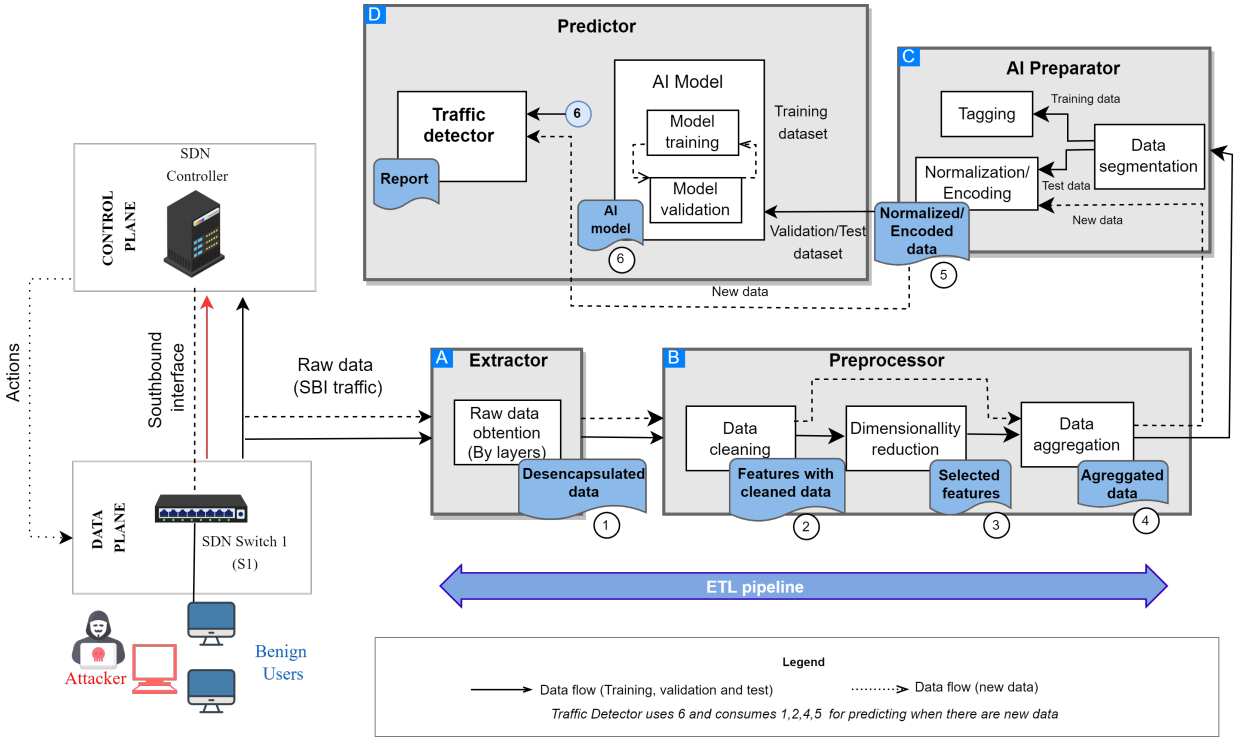

Figura 4.1: Identificador de tráfico inusual

El identificador de tráfico inusual está formado por varios componentes que manejan procesos interactuantes con los datos de la SBI, con los elementos de red y con el controlador. Los componentes de este modelo de detección son: Extractor de información de paquetes, preprocesador de los datos crudos, preparador para inteligencia artificial y predictor. El identificador de tráfico inusual tiene dos fases de operación:

① obtención del modelo de inteligencia artificial.

② predicción.

El primer componente del modelo denominado **extractor**, trabaja en las fases ① y ②, siendo el encargado de adquirir la información del tráfico de la SBI. En este componente el proceso de extracción comienza con la obtención de datos crudos a partir de la escucha activa del canal. En este proceso es necesario desencapsular la información iniciando por la capa principal (α) hacia las capas inferiores (β), generando repeticiones sucesivas hasta obtener todos los campos de cada paquete. *Refiérase al algoritmo 1.*

El segundo componente llamado **procesador**, se encarga de la transformación los datos crudos de tráfico (desencapsulados). En el proceso de transformación los datos crudos extraídos pueden reordenarse, transponerse, asociarse, sustituirse, eliminarse e incluso, a partir de ellos se pueden desprender cálculos para obtener nuevas características. Esto debido a que desde la perspectiva de inteligencia artificial, los datos crudos trabajan en arreglos de filas por

Algorithm 1 Unusual Traffic Detection (a and b components)

Require: Raw traffic data = sourcePacKet
Ensure: Aggregated Data
  Extractor Component phases ① & ②
  $a \leftarrow$ sourcePacKet
  while a **do**
    $\text{layer}_{\alpha} = a[\text{layers}][\text{layer}]$
    $\text{layer}_{\beta} = \text{layer}_{\alpha}[\text{sublayer}]$
    if (feature $\in \text{layer}_{\alpha}$) or (feature $\in \text{layer}_{\beta}$) **then**
      $\text{infoPK} \leftarrow \text{layer}_{\alpha,\beta}[\text{feature}]$
    end if
    $X \leftarrow \{\text{feature}, \text{infoPK}\}$
  end while
  Processor Component
  procedure CLEANING DATA(feature, infoPK, threshold) phases ① & ②
    $\text{threshold} \leftarrow N \in [0, 1]$
    for X **do**
      $\text{feature}' =$ drop (feature, threshold)
      $\text{infoPK}' = \text{drop}(\text{NaN}, \text{Null}, \text{infoPK}, \text{any})$
    end for
    $X \leftarrow \{\text{feature}', \text{infoPK}'_n\}$
  end procedure
  procedure DIMENSIONALITY REDUCTION phase ①
    for each $\text{feature}' \in \text{X}'$ **do**
      if (correlation $\wedge$ relevance) **then**
        Keep $\text{feature}'$ in X'
      else
        Drop $\text{feature}'$ in X'
      end if
    end for
    $\text{X}' \leftarrow \{\text{feature}', \text{infoPK}'\}$
  end procedure
  procedure DATA AGGREGATION phases ① & ②
    while X' **do**
  $\text{chunk} \leftarrow$ portion of infoPK'
      for each $\text{chunk} \in \text{X}'[\text{feature}']$ **do**
        $(\text{SD}(X'_i) = a \times 10^b)$, $\bar{X}'_i$ or $Mo(X'_i)$ on infoPK'
      end for
      $\text{X}' \leftarrow \{\text{feature}', \text{chunk of infoPK}'\}$
    end while
  end procedure

columnas, donde las columnas representan cada una de las características de los paquetes, mientras que las filas son cada uno de los registros de los paquetes. Para efectuar el proceso de transformación este componente utiliza subcomponentes de limpieza de datos, reducción de dimensionalidad y agregación de datos. *Refiérase al algoritmo 1.*

La *limpieza de datos*, que se usa tanto en la fase ① como en la ②, se enfoca en higienizar los datos crudos, duplicados y que contienen registros nulos o NaN, tomando en cuenta que al existir grandes volúmenes de registros de cada uno de los paquetes, lo más probable es que exista ruido o ausencia de datos, debido a posibles errores de transmisión o encapsulación

propia de los paquetes. La limpieza de datos, puede realizarse usando técnicas de eliminación de muestras, interpolación o imputación de valores. En este caso, se ha considerado una técnica de eliminación la cual descarta las características X o registros que contienen más de N probables datos inválidos (donde $N \in [0, 1]$).

Así mismo, en virtud de la diversidad de campos que conforman los paquetes de datos, se hace imperativo disponer de características precisas con el fin de optimizar el tiempo de entrenamiento y simplificar el modelo de AI. Con este propósito, se aborda la *reducción de la dimensionalidad*, una estrategia que incorpora dos métodos fundamentales como la selección y la extracción de características.

Por un lado, la selección de características es un método que devuelve un subconjunto de características relevantes y significativas de un conjunto de datos más grande, es decir, donde $X' \subseteq X$. El objetivo es mejorar la exactitud del modelo de AI eliminando las características irrelevantes, redundantes o ruidosas. Algunos autores señalan que la selección de características reduce el sobreajuste, disminuye el tiempo de cálculo involucrado para obtener el modelo de AI y se mejora la exactitud de las predicciones. La selección de características se apoya en las siguientes técnicas: *filter, wrapper, embedded, feature importance (FI)* o incluso técnicas híbridas.

Por otro lado, la extracción de características permite construir nuevas características que dependen del conjunto de características original, es decir, $X' = f(X)$. Para ello, se utiliza transformación algebraica y criterios de optimización que permiten encontrar el conjunto de características más distintivo, informativo y reducido para mejorar tanto la eficiencia del procesamiento como del almacenamiento de datos (Abdulazeez et al., 2020). Entre las técnicas más comunes para la extracción de características se encuentran los algoritmos lineales y no lineales, PCA, *Linear Discriminant Analysis* (LDA), DWT, *Multi-Dimensional Scaling* (MDS), *Isometric Mapping* (ISOMAP), *Locally Linear Embedding* (LLE), *Latent Semantic Indexing* (LSI) y *clustering*.

En este caso, se ha generado la *reducción de la dimensionalidad* usando técnicas híbridas que permiten calcular la medida de mínima redundancia-máxima relevancia (mRMR) y la medida de correlación de selección de características (CFS, por sus siglas en inglés). En este punto se torna importante señalar que la *reducción de la dimensionalidad* se efectúa únicamente durante la fase ①. Por lo que, una vez que se calcula la importancia de cada característica basados en su FI_i, se crea un conjunto de datos X'_i que contiene únicamente dichas características, obteniendo la siguiente expresión matemática:

$$FI_i = f(X_i, Y) - f(X_i', Y) \tag{4.1}$$

Donde:

FI_i es la ganancia de información de la característica i.

X_i es el conjunto de datos original que incluye la característica i.

Y es la variable objetivo.

$f(X_i, Y)$ es la exactitud del modelo utilizando todas las características.

X_i' es el conjunto de datos que excluye la característica i.

$f(X_i', Y)$ es la exactitud del modelo utilizando todas las características excepto la i.

Consecuentemente, teniendo en cuenta la naturaleza vasta y diversa del tráfico de red, y analizando que las observaciones individuales de los paquetes no proporcionan una comprensión exhaustiva del comportamiento usual o inusual de los flujos, se emplea un proceso de *agregación de datos* que involucra la agrupación de paquetes en bloques o ventanas. La agrupación de los paquetes implica la combinación de múltiples entradas de datos en un grupo o categoría. Por ejemplo: IP de origen y destino, puerto de origen y destino, timestamp, etc. Adicionalmente, por cada característica de cada uno de estos bloques de paquetes, se usan funciones estadísticas, tales como desviación estándar $SD(X_i') = a \times 10^b$, promedio $\bar{X}_i'$ o moda $Mo(X_i')$, dependiendo del valor significativo de cada campo, lo cual permite analizar el tráfico de red para reducir la complejidad y facilitar la identificación de patrones y anomalías.

En el componente denominado **preparador para AI**, se realiza la segmentación, codificación y normalización de los datos. La segmentación de los conjuntos de datos para entrenamiento, validación y prueba se ejecuta exclusivamente en la fase ①. No obstante, la codificación y normalización se llevan a cabo tanto en la fase ① como en la ②. *Refiérase al algoritmo 2*

Durante la segmentación de los datos, en el segmento usado para entrenamiento se etiqueta la información para que el modelo de AI genere una asociación entre los patrones observados y sus correspondientes valores representativos, los cuales son asignados según la naturaleza del problema que se aborda. En este contexto, al intentar discernir entre tráfico "usual" o "inusual", el modelo de AI se enfrenta a un problema específico de clasificación binaria. Los modelos de clasificación binaria se revelan como una solución eficaz para problemas donde la respuesta se limita a dos posibles resultados. Estos modelos se caracterizan por su capacidad para asignar instancias de datos a una de dos categorías distintas, las cuales tienden a ser numéricas (0, 1), pero también pueden ser categóricos (usual, inusual). En lo que respecta a los segmentos de validación y de prueba no usan ningún tipo de etiquetado. Por una parte el conjunto de validación permite ajustar los hiperparámetros del modelo de AI, a través de las continuas iteraciones. Por otra parte, el segmento de prueba se usa para la evaluación del rendimiento final del modelo de AI.

Abordando lo relativo a las diferencias existentes entre los datos numéricos y categóricos, se efectúa un proceso de *codificación*, la cual es utilizada para convertir variables categóricas en numéricas, facilitando así el manejo de los datos. Existen varias técnicas de codificación de variables categóricas a numéricas, estando entre las más comunes *one hot, dummy, ordinal, sum,*

Helmert, polynomial, Backward o binary (Björk, 2011). En este caso, el modelo propuesto utiliza codificación one hot, en la que se generan vectores binarios, limitados a contener exclusivamente valores 0 o 1, con una longitud igual al número total de categorías únicas. De este modo, a cada observación, se asigna el valor 1 a la columna que corresponde al valor presente, mientras que las demás columnas permanecen con un valor de 0. Por ejemplo, para referenciar los diferentes protocolos se generarán vectores tales como: TCP = [1, 0, 0], IP = [0, 1, 0], ICMP [0, 0, 1], etc.

De igual manera, a fin de asegurar que los datos de tráfico referenciados como usuales e inusuales tengan un impacto equitativo en el modelo de AI, se genera un ajuste de los valores para que estén en una escala común, a través de un proceso de normalización. Existen varias operaciones de normalización de datos, entre las que se pueden citar a centrado, escalamiento, descorrelación, estandarización o blanqueo (L. Huang et al., 2023). Para este caso particular se han usado operaciones combinadas de centrado y escalamiento, debido a la variabilidad de los datos. De este modo de obtiene la siguiente representación matemática:

$$X_{\text{normalizado}} = \frac{X - \bar{X}}{\sigma} \tag{4.2}$$

Donde:

X es el conjunto de datos original.

$\bar{X}$ es el promedio de cada uno de los valores del conjunto de datos.

σ Es la desviación estándar de los datos de X.

Por último, en el componente denominado **predictor** se define el modelo de AI para la detección y en el proceso de carga se trabaja con tres conjuntos de datos. El de entrenamiento, debidamente etiquetado con tráfico usual e inusual y dos adicionales para validación y test, donde los registros de tráfico se encuentren ubicados de manera aleatoria para que el modelo de AI sea capaz de predecir. *Refiérase al algoritmo 2*

Posteriormente, se extrae el modelo de AI probado y se lo utiliza en un subcomponente denominado **detector de tráfico**. Este subcomponente trabaja en conjunto con el controlador. De este modo, cuando ingresa nuevo tráfico y al detectarse comportamientos inusuales se emite un reporte para establecer las acciones sobre los elementos SDN conforme las políticas de seguridad. *Refiérase al algoritmo 2*

Considerando la amplia variedad de algoritmos de inteligencia artificial disponibles, resulta imperativo especificar aquellos que han sido elegidos para la presente propuesta, así como los motivos que respaldan dicha elección. En este contexto, se ha optado por integrar tanto algoritmos de aprendizaje automático, específicamente para abordar problemas de regresión relacionados con la reducción de dimensionalidad, así como de algoritmos de aprendizaje profundo, orientados en resolver la clasificación. Esta elección permite el desarrollo del identificador de tráfico inusual.

Se han seleccionado algoritmos de regresión para la reducción de dimensionalidad dado que se necesita encontrar una relación funcional entre las características de entrada y una variable

Algorithm 2 Unusual Traffic Detection (c and d components)

Require: Aggregated Data $X' \leftarrow \{\text{feature}', \text{chunk of infoPK}'\}$
Ensure: Model Detection
  AI Preparator Component
  procedure SEGMENTATION TAGGING ENCODING NORMALIZATION
  training/validation/test $\leftarrow$ %X′[chunk of infoPK′]
    if training **then**
      if (traffic type A) **then**
        chunk of infoPK'_n label =0
      else
        chunk of infoPK'_n label =1
      end if
    end if
    if (validation/test/new traffic) **then**
      random (chunk of infoPK′)
    end if
    for each chunk of [infoPK′] **do**
      encode unique value, Index 1 : 0
      scale X′
    end for
    $X' \leftarrow \{\text{feature}', \text{encoded infoPK}'\}$
    Dataset
  end procedure
  Predictor Component
  procedure AI MODELING
    select AI algorithm
    if (training/validation phase) **then**
      hyperparameter adjustments by rounds
    else if test phase **then**
      test predictions
    end if
    AI Model
  end procedure
  procedure DETECTION AND ACTIONS(new traffic data, AI model)
    model $\leftarrow$ AI model imported
    incomingPks $\leftarrow$ new traffic data
    while incomingPks **do**
      predictions in incomingPk
      if predictions = unusual traffic **then**
        send report to SDN controller
        controller takes actions
      else
        controller allows traffic
      end if
    end while
  end procedure

objetivo. En este caso, se requiere encontrar aquellas características de mínima redundancia y máxima relevancia que tengan correlación. Es así como para este fin se han considerado algoritmos de aprendizaje automático de bosques aleatorios (RF) y regresión logística (LR).

Los bosques aleatorios son un conjunto de árboles de decisión, es decir modelos que toman

decisiones basadas en varias condiciones, donde la salida final se determina por la combinación de las predicciones individuales de cada árbol (Dewi et al., 2019). Se han seleccionado los bosques aleatorios porque cada vez que se genera un árbol, se registra la disminución de la impureza (*Gini impurity*) causada por cada característica. De este modo, al promediar estas disminuciones en todos los árboles, se obtiene una puntuación de importancia para cada característica. Esto facilita la identificación de las características basadas en el puntaje obtenido y por consiguiente se obtienen las características más relevantes (Grömping, 2009).

Por otro lado, también se ha seleccionado regresión logística, que, aunque desde el enfoque estadístico es generalmente adoptada en problemas de clasificación, en este planteamiento se ha empleado como técnica combinada para eliminar las características de menor importancia usando *Recursive Feature Elimination* (RFE) que es un método que elimina recursivamente las características menos importantes hasta que se alcance el número deseado de características.

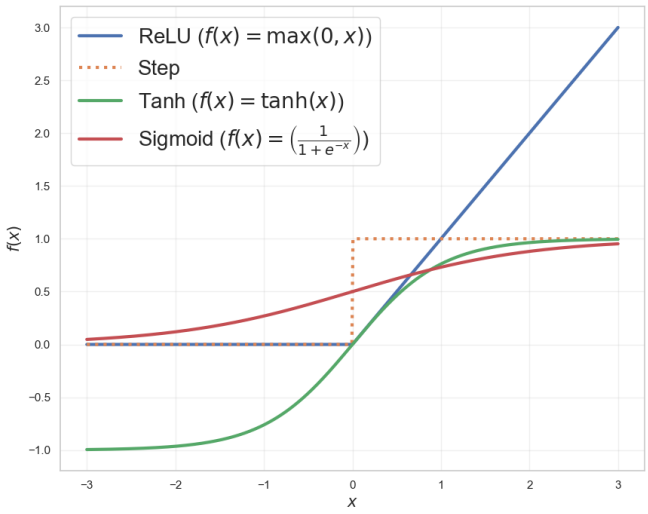

Figura 4.2: Funciones de activación

En el contexto de la clasificación entre tráfico usual e inusual, se ha optado por la implementación de redes neuronales (NN, por sus siglas en inglés). La elección de trabajar con NN se fundamenta en la diversidad inherente al tráfico de una red, caracterizado por datos complejos y variados. Así también, la adopción de NN se justifica por la capacidad de estos modelos para capturar representaciones profundas y complejas, lo que resulta esencial en la interpretación efectiva de patrones de tráfico. Esta elección se respalda en la necesidad de obtener resultados más precisos y realistas, especialmente en comparación con enfoques de aprendizaje automático convencionales.

En las NN cada neurona o también conocida como perceptrón recibe múltiples entradas, aplica pesos a cada una de esas entradas, calcula su suma ponderada y luego aplica una función no lineal de activación para producir la salida. Este proceso es manejado en capas de entrada, de salida y las capas ocultas, siendo las capas ocultas muy significativas, dado que cuando los datos pasan entre ellas, las características se recombinan y se reconstruyen de manera más

compleja (Jakhar et al., 2020).

Las entradas de la NN por lo tanto son cada una de las características o clases obtenidas del proceso de reducción de dimensionalidad. Por otro lado, los pesos, son parámetros ajustables que se utilizan para ponderar la importancia de las entradas de un perceptrón. Estos pesos se ajustan durante el proceso de entrenamiento de la red neuronal para minimizar la diferencia entre las salidas predichas y las salidas reales. El cálculo de la suma ponderada incluye la consideración de un sesgo, pues este permite entre otras cosas desplazar la función de activación de la neurona en el eje de salida, por ejemplo, cuando el resultado de suma ponderada sea cero. Por último, la función de activación permite determinar si la neurona debe activarse o no. Entre las funciones de activación más conocidas están: *Step, Rectified Linear Unit (ReLU), sigmoidal, Tangente Hiperbólica (Tanh)*, como se puede observar en la Figura 4.2, en donde se muestra el comportamiento de cada una de las funciones de activación y sus respectivas expresiones matemáticas. Así también en la Figura 4.3 se visualiza de mejor manera la estructura de un perceptrón y de una red neuronal en concordancia con esta tesis doctoral.

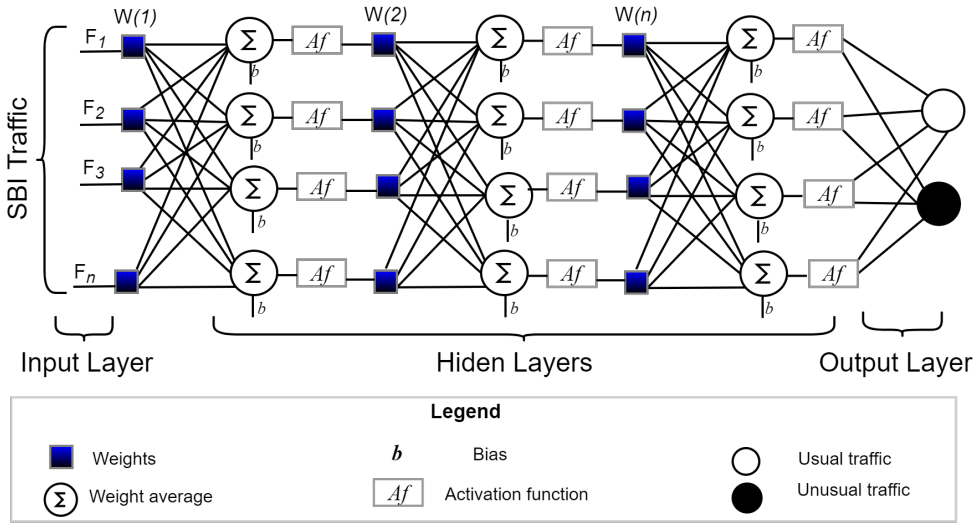

Figura 4.3: Estructura de un perceptrón y de una red neuronal

En este modelo se ha elegido trabajar con funciones de activación de manera híbrida, es decir por cada capa se ha elegido un tipo de función de activación. Esto dado que lo que se desea es evitar la linealidad del modelo de AI y permitir que se puedan interpretar caracteríticas cada vez más complejas. En el capítulo 5 , se explorará la configuración de cada componente de la red neuronal creada, junto con una explicación detallada de los valores seleccionados.

Identificador de comportamientos inesperados

Como se ha destacado en repetidas ocasiones, la arquitectura de SDN es desplegada en entornos intrínsecamente dinámicos, lo que implica que cada alteración en alguno de sus componentes conlleva la generación de un registro. Estos registros contienen información relevante que puede ser usada como información corroborativa. Ciertamente, al reconocer que

la información rutinaria puede resultar redundante, esta propuesta se centra únicamente en datos pertinentes a acciones que puedan estar asociados a un evento de ciberseguridad en relación con los elementos de la SDN, por lo que en este caso se considera información de las aplicaciones, la topología, los dispositivos, los flujos y enlaces; y los accesos al controlador.

El identificador de comportamientos inesperados está basado principalmente en el concepto de máquinas de estado finitas. Este identificador guarda el estado de cada elemento mencionado anteriormente y se mantiene en escucha activa, por consiguiente es el responsable de asignar los estados. Básicamente, actúa como un ente de corroboración que se encarga de comparar la información entre varias fuentes. De este modo, evalua la cadena de cada registro en relación a su longitud y temporalidad. En caso de detección de discrepancias en las fuentes de información, emite un reporte. Es así que, el proceso de identificación inicia cuando se encuentra una alteración del estado génesis *(q1)*. Posteriormente, los estados pueden pasar a un estado auxiliar *(q2)* o *(q3)*, que debe ser comprobado, y posteriormente pasar a un estado siguiente *(q3)* para recopilación de la información, trabajando de manera interactiva y continua.

Dicho esto, a continuación, se describen cada uno de los estados del identificador de comportamientos inesperados y sus respectivas transiciones.

- **Estado Génesis *(q0)*:** Hace referencia al inicio de operaciones de una SDN, en el cual se pueden apreciar las aplicaciones activadas e instaladas para la normal funcionamiento de la red, la topología inicial en la cual se observa la cantidad de dispositivos conectados, los enlaces iniciales, etc. Este estado es registrado como una instantánea.
- **Transición 1 (δ1):** Punto de detección de cambios en los elementos SDN. El estado génesis o idle pueden cambiar a medida que se presentan los requerimientos del entorno debido a la activación, instalación o desactivación de aplicaciones; modificaciones a nivel del plano de datos como la inclusión de nuevos elementos de red, lo cual asocia nuevos enlaces, reconfiguración del plano de control, etc.
- **Estado Auxiliar *(q1)*:** Se adopta este estado cuando se ha identificado un cambio en el estado génesis *(q0)*.
- **Transición 2 (δ2):** Punto de verificación del cambio. Si el cambio coincide en las otras fuentes de información, hay un cambio de estado al nivel siguiente *(q3)*. Por el contrario, si ocurre un cambio de estado en un elemento, pero en el proceso de comparación la información no coincide, hay un cambio de estado a un nuevo estado auxilar *(q2)*.
- **Nuevo estado auxilar *(q2)*:** Es el estado adquirido en caso de discrepancias en las diversas fuentes de información.
- **Transición 3 (δ3):** A pesar de que se asume que las fuentes de información son confiables, no se puede descartar discrepancias. En este punto, surge la incertidumbre sobre la veracidad de la información y se plantea una analogía con el problema de los generales bizantinos (Lamport, 1983). En este problema, la confiabilidad de las fuentes de información es desconocida, y se vuelve esencial tomar decisiones sobre cuál información debe ser considerada válida. Ante este desafío, se pueden aplicar diversas técnicas entre las cuales destacan el consenso por mayoría o votación, la verificación

cruzada, la evaluación de credibilidad de la fuente, entre otras. En este caso, se ha elegido un consenso por mayoría. De este modo, cuando un cambio esté presente en la mitad más uno de las fuentes de información se efectuará el cambio de estado de un nuevo estado auxiliar *(q2)* a estado siguiente *(q3)*.

- **Estado siguiente *(q3)*:** Es el estado adoptado cuando no hay discrepancia en las fuentes de información, o cuando ya ha existido un consenso.
- **Transición 4 (δ4):** Una vez que el identificador de comportamientos inesperados se encuentra en estado siguiente *(q3)* se puede generar una recuperación de datos que servirá a posterior para la consolidación de evidencias y para la construcción del trazado del origen del problema de ciberseguridad.
- **Estado Idle *(q4)*:** Es un estado de reposo o pausa. Si no existen cambios se mantiene en este estado. Sin embargo, si existen cambios se retoma nuevamente todo el proceso de evaluación.

En la Figura 4.4 se representa el funcionamiento del identificador de comportamientos inesperados de los elementos SDN.

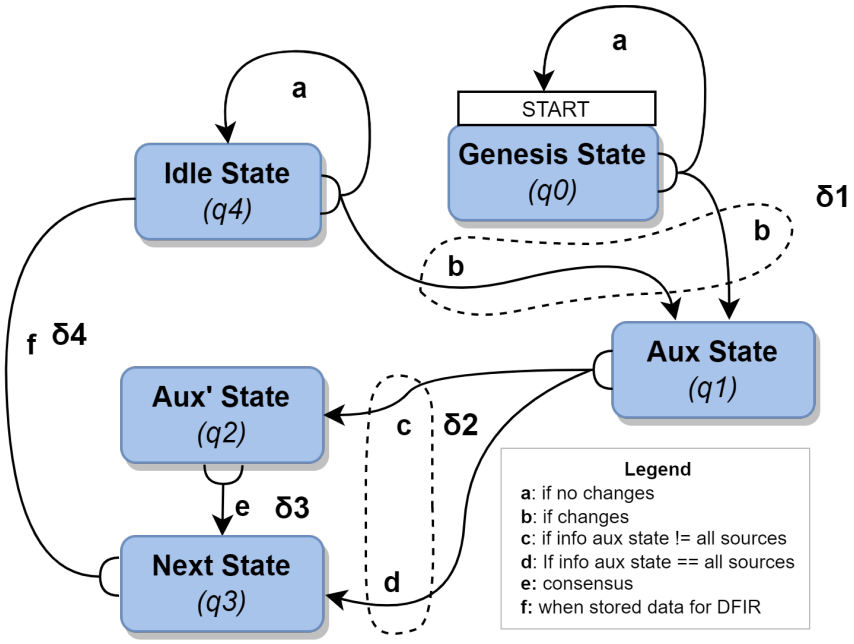

Figura 4.4: Funcionamiento del identificador de comportamientos inesperados

4.3 Modelo para la preservación de la evidencia

La integridad de la evidencia es un factor crítico en el contexto de una investigación forense ya que fortalece la credibilidad de un proceso legal. A lo largo del despliegue de los procesos forenses, se destaca la importancia de garantizar el almacenamiento de la evidencia en complemento con la CoC. La falta de cautela en el proceso de almacenamiento puede incrementar el sesgo o discriminación de la evidencia y comprometer la integridad y precisión

de los resultados forenses, generando incertidumbre en la determinación de responsabilidades. Por lo que la correcta preservación de evidencias no solo es fundamental para la validez de los procesos forenses, sino que constituye un eje ético para evitar resultados injustos o inexactos.

De igual manera es preciso destacar que un apropiado almacenamiento de la evidencia no solo asegura su integridad y confiabilidad en un entorno legal, sino que también toma relevancia desde el enfoque técnico ya que impulsa una utilización efectiva de la información durante la correlación de eventos de ciberseguridad. Este aspecto por lo tanto contribuye significativamente en la capacidad para identificar patrones y relaciones entre eventos agilizando los procesos de toma de decisiones respecto a las políticas y estrategias adoptadas. En consecuencia, se establece un sólido fundamento para la mejora continua en el ámbito de la ciberseguridad de las SDN.

Teniendo en cuenta la importancia del almacenamiento de la evidencia tanto desde el enfoque organizacional-legal como desde el técnico, se ha considerado como una contribución de esta tesis doctoral la presentación de un modelo de preservación de evidencia. En este sentido, en este apartado se detalla la descripción del modelo en cuestión. Para el efecto, se presentará la terminología utilizada y el detalle de la funcionalidad del modelo con las tecnologías que han sido consideradas.

4.3.1 Terminología

Sistemas de almacenamiento descentralizados, *blockchain* e inmutabilidad

Existen varias opciones para almacenamiento que van desde las bases de datos, almacenamiento en la nube hasta sistemas de almacenamiento centralizados o sistemas de almacenamiento descentralizados. Los últimos se sustentan en la premisa fundamental de que la toma de decisiones y el control no se encuentran concentrados en una única entidad o autoridad, sino que se distribuyen entre diversos participantes. Este enfoque descentralizado busca eliminar la dependencia de una entidad central, la manipulación de los registros y los fallos en un punto único.

Entre las tecnologías más conocidas para almacenamiento descentralizado se encuentran IPFS y *blockchain*, siendo esta última de gran utilidad cuando se requiere inmutabilidad sobre las transacciones.

La tecnología *blockchain*, basada en un modelo de comunicación *peer-to-peer* (P2P), utiliza una cadena de bloques para registrar y verificar de manera segura las transacciones entre sus participantes o nodos. Cada bloque contiene un conjunto de transacciones y está enlazado criptográficamente al bloque anterior, formando una cadena inmutable que proporciona un historial confiable de todos los registros. Los nodos o participantes de la red *blockchain* desempeñan roles equivalentes y se comunican directamente sin depender de una entidad central. Para lograr este propósito, los nodos llegan a un consenso sobre el estado compartido de la información mediante algoritmos específicos, estableciendo reglas sobre el estado compartido de la información y validando las transacciones (Pahlajani et al., 2019).

En este punto se torna importante hacer distinciones ente redes *blockchain* privadas o *permissioned* y públicas *permissionless*. Por una parte, en las redes *blockchain* públicas no

existen restricciones respecto a la participación, por lo que cualquier persona puede unirse a la red para transaccionar y participar en el proceso de consenso el cual involucra la resolución de problemas criptográficos. Los algoritmos de consenso más utilizados en estas redes blockchain son prueba de trabajo (PoW, por sus siglas en inglés) y prueba de participación (PoS, por sus siglas en inglés).

Por otra parte, en las *blockchain* privadas el acceso a la red es controlado y exclusivo para los participantes autorizados. Dichos participantes, por lo tanto, son los únicos que pueden transaccionar y acceder a la información almacenada. En este tipo de redes *blockchain* se tiene un mayor control sobre la confidencialidad de la información ya que se pueden definir reglas y permisos de participación. Los mecanismos de consenso en este tipo de redes *blockchain* son diferentes a los usados en las redes públicas. En las *blockchain* privadas se elige un nodo o conjunto de nodos para crear y difundir el siguiente bloque de la cadena y garantizar que la cadena almacenada en cada nodo sea coherente.

De igual manera, un término por acotar a las redes *blockchain* es el de los contratos inteligentes, siendo estos programas informáticos autoejecutables diseñados para facilitar, verificar o hacer cumplir acuerdos. Estos contratos contienen lógica programable que determina las condiciones y acciones específicas que deben llevarse a cabo cuando se cumplen ciertas condiciones. Al eliminar la necesidad de intermediarios, los contratos inteligentes ofrecen un medio transparente y confiable para automatizar procesos, como transacciones financieras o acuerdos contractuales (Pahlajani et al., 2019).

4.3.2 Desarrollo del modelo de preservación de evidencia

El modelo de preservación de evidencias constituye un elemento crucial en la arquitectura propuesta, siendo los registros de las evidencias un recurso fundamental e imprescindible para la continuidad del proceso DFIR. Por un lado, los registros de las evidencias representan la información efectiva que contribuye a los procesos investigativos desde una perspectiva legal. Por otro lado, estos registros desempeñan un papel crucial como insumo esencial para la reconstrucción de los posibles escenarios que condujeron a la ocurrencia del incidente. En vista de la importancia de los registros en mención, se torna imperativo brindarles el debido tratamiento durante el proceso de almacenamiento.

Dicho esto, este modelo tiene como finalidad recibir, dar formato y almacenar evidencias de manera segura en un registro inmutable. La decisión de incorporar la inmutabilidad se fundamenta en el reconocimiento de que en todos los sistemas existe la posibilidad no solo de ataques a los registros de evidencias que impacten sobre su autenticidad e integridad, sino también de la presencia de un administrador con intenciones maliciosas. En este caso para la propuesta se ha elegido trabajar con redes *blockchain*, ya que al replicar los datos en múltiples nodos, se reduce los riesgos de las debilidades antes mencionadas.

Considerando que la propuesta de esta tesis doctoral se basa en la preservación de evidencia dentro de una organización, y tras destacar las diferencias significativas entre los dos tipos de redes *blockchain*, es pertinente subrayar que el modelo planteado adopta el uso de redes *blockchain* privadas como medio para la conservación de la evidencia. En la Figura 4.5 se observa la representación en alto nivel del modelo de preservación de la evidencia.

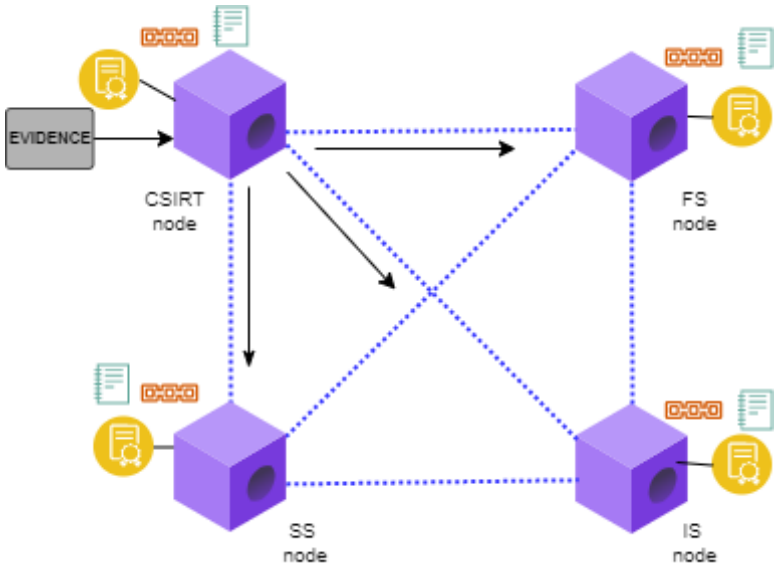

Figura 4.5: Modelo de preservación de evidencia

De igual manera, se torna importante aclarar que este modelo se enfoca en la utilización de la evidencia ya consolidada. La consolidación de la evidencia, al estar influenciada por factores de gobernanza y regulaciones locales, puede variar significativamente según la jurisdicción y el contexto. Por lo tanto, para los fines de esta explicación, se parte de la idea de que la evidencia necesaria ya ha sido consolidada.

En este sentido, el modelo de preservación está compuesto por tres componentes interrelacionados: **formateador de evidencia, nodos y registro inmutable.** La operación de los componentes de este modelo se describe mediante un diagrama de secuencia ilustrado en la Figura 4.6.

De este modo, el componente inicial, llamado **formateador de evidencia**, recibe la evidencia y realiza ajustes en su formato sin comprometer las características intrínsecas de la misma. Efectúa esta acción dado que los contratos inteligentes presentes en los nodos de la red son programados definiendo estructuras de datos particulares.

En relación con los **nodos** en este caso particular, se ha considerado la participación de la infraestructura de los siguientes interesados: IS, SS, FS y CSIRT, previamente definidos en la sección 3.5.2, las cuales desempeñarán el papel de nodos en la red *blockchain*. El propósito de esta definición es asegurar la trazabilidad integral de todas las transacciones para todos los participantes involucrados en el proceso de preparación forense y de respuesta a incidentes. De este modo, una vez que el primer componente ha dado formato a la evidencia, ésta es enviada a los nodos que componen la red descentralizada, quienes a su vez han acordado e instanciado un *contrato inteligente*. En este modelo, el nodo CSIRT será el encargado de proponer las transacciones para que estas sean puestas en consideración de los otros nodos hasta llegar a un consenso.

El *contrato inteligente* en este caso, cuenta con funciones de lectura y de escritura, lo que les permitirá a los nodos interactuar con el **registro inmutable**. De este modo, cada

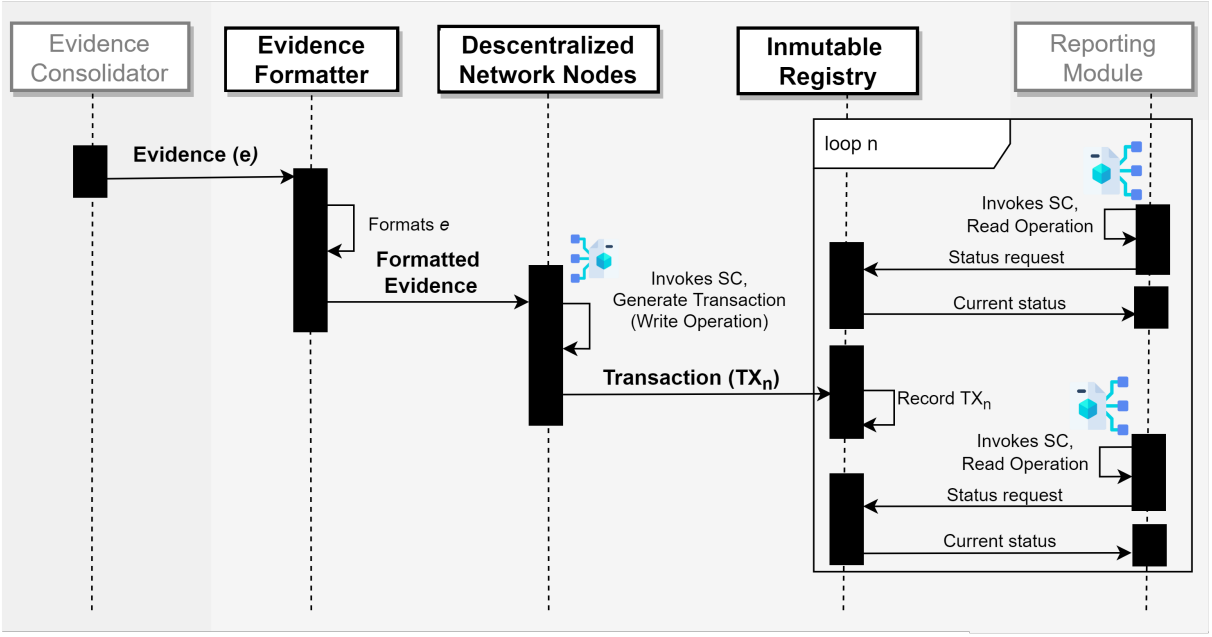

Figura 4.6: Diagrama de secuencia del modelo de preservación de evidencia

evidencia que llega al nodo es procesada generando una transacción como resultado de la función de escritura y que posteriormente será almacenada en un bloque dentro del **registro inmutable**. Finalmente, este modelo de preservación interactúa con el módulo de reporte de la arquitectura propuesta, el cual para obtener datos del **registro inmutable** ejecuta periódicamente operaciones de lectura con el fin de recuperar su estado.

4.4 Resumen del capítulo

En este capítulo se presentaron dos contribuciones específicas que abordan los desafíos relacionados con la identificación de eventos junto con la adquisición y tratamiento de datos, y el almacenamiento de evidencia en el contexto de DFIR. Se detallaron las estrategias para detectar comportamientos inusuales de tráfico y comportamientos inesperados de los elementos SDN, optando por la identificación de anomalías como estrategia principal, respaldada por el uso de inteligencia artificial y máquinas de estado finitas. Además, se introdujo un modelo para la preservación de evidencia, enfatizando la importancia de mantener su integridad para asegurar la validez de la cadena de custodia, lo cual es esencial en procesos investigativos judiciales. En este sentido, se propuso el empleo de redes Blockchain privadas como una solución para este propósito.

Capítulo 5

VALIDACIÓN DE LAS PROPUESTAS

5.1 Introducción

En este capítulo se detalla la implementación y validación de las contribuciones presentadas en esta tesis doctoral y que han sido previamente explicadas en el capítulo 4. Esta presentación es realizada con el fin de poner a prueba la viabilidad de cada una de las propuestas planteadas, en un ambiente que requiere principal atención en el contexto de la gestión de preparación forense y de respuesta a incidentes de seguridad en SDN.

Para el efecto, es preciso señalar que la consideración prioritaria para la validación de las propuestas recae en la protección del activo más crítico, que en este contexto es el controlador SDN. Por lo tanto, se ha definido un ambiente de pruebas que recopila en gran medida la realidad de muchas organizaciones respecto a la gestión de la ciberseguridad cuando existe un incidente que pueda afectar a este activo.

En este sentido, en la sección 5.2 se detalla un ambiente para la validación de las propuestas presentadas en el capítulo 4. Este ambiente recopila principalmente dos eventos: **(i)** ataque al controlador SDN y **(ii)** cambios en los elementos SDN. Para ello se presenta un escenario de ataque, describiendo el contexto, las debilidades encontradas y aprovechadas, el tipo de ataque a ejecutar, los actores involucrados y las asunciones. También se incluye la descripción de un escenario con cambios en los elementos SDN. La sección 5.3 describe el ambiente de implementación de los modelos de inteligencia de filtrado, adquisición y tratamiento de datos, y de preservación de evidencia. De igual manera, se especifican los paquetes de software utilizados para el despliegue tanto del escenario de ataque como de las propuestas efectuadas en esta tesis doctoral. Por último, en la sección 5.4 se entregan los resultados obtenidos de la evaluación de las propuestas de los modelos de inteligencia de filtrado, adquisición y tratamiento de datos, y de preservación de evidencia presentados en el capítulo 4.

5.2 Diseño de escenarios para validación

5.2.1 Escenario de ataque: Contexto, debilidades aprovechadas, tipo de ataque y supuestos

De manera general la gestión de una red se realiza a través de políticas, que en un entorno SDN se interpretan como reglas, las cuales se utilizan para tomar decisiones sobre cómo se deben dirigir los paquetes de datos en función de su contenido. Dichas reglas son gestionadas mediante las tablas de flujo de los conmutadores. De este modo, secuencialmente, cuando se recibe un paquete, su contenido es analizado y se aplica la primera regla de la tabla de flujo que coincida con dicho contenido. En caso de que ninguna regla se corresponda con el paquete, se aplicará una regla por defecto para determinar su destino en la red. En redes que hacen uso de Openflow se puede gestionar las tablas de flujo en dos modos: proactivo y reactivo.

En un modo proactivo, un controlador Openflow puede predefinir las acciones o instrucciones en las tablas de flujo de los conmutadores, antes que el tráfico llegue. Esto significa que los elementos de red ya tienen una tabla de flujo completa antes de recibir los paquetes de red. Por lo que este enfoque, dependiendo de las capacidades de los elementos de red (principalmente de memoria), puede reducir la latencia y mejorar el rendimiento. Mientras que, en un modo reactivo, los flujos nuevos responden al tráfico mediante la consulta al controlador para crear una regla en la tabla de flujo basada en la instrucción. Es así que, los paquetes entran al conmutador, el agente OpenFlow en el conmutador hace una búsqueda en las tablas de flujo y si no se encuentra ninguna coincidencia para el flujo, el conmutador realiza una consulta al controlador esperando instrucciones (Open Networking Foundation, 2015a).

A pesar de que el modo proactivo parecería ser la respuesta para evitar que gran parte del tráfico llegue al controlador, existe una limitación importante basada en la aplicabilidad. El modo proactivo puede ser usado en entornos donde el tráfico es totalmente predecible, mientras que el modo reactivo puede responder a entornos de tráfico con características de gran variabilidad.

Tomando en cuenta que, como se explicó en el capítulo 1 de esta tesis doctoral, un gran porcentaje de implementaciones de SDN se realizan en entornos de tráfico variable por naturaleza (WAN, redes acceso), se puede asumir que el modo de gestión de las tablas de flujo es reactivo. Por lo que, al existir consultas al controlador con mayor continuidad, se incrementa la posibilidad de que el tráfico Openflow que llega desde los elementos de red, tenga fines maliciosos, que de no ser detectados pueden causar deterioro o daño total en la SDN. Esto desde luego puede ser aprovechado por atacantes para efectuar peticiones que inunden el canal Openflow y por consiguiente saturen y dejen indisponible al controlador.

Con el propósito de validar la teoría relacionada con los modos proactivo y reactivo, se ha optado por utilizar el controlador ONOS. En este contexto, se ha observado que, inicialmente, ONOS está configurado por defecto en el modo reactivo. Además, se ha notado que, de forma predeterminada, ONOS incluye una opción que establece un tiempo de expiración para las entradas de flujo, lo que implica que cada 10 segundos los conmutadores realizan consultas al controlador (ONOS, 2015).

En virtud de este comportamiento, se desarrolla un escenario de ataque de DDoS en el cual el atacante logra atravesar los conmutadores y alcanzar al controlador. Los ataques de DDoS tienen como objetivo saturar rápidamente los canales de comunicación, lo cual puede llegar al deterioro total o parcial de la red, evitando que los usuarios puedan acceder a los servicios ofrecidos. Generalmente, estos ataques son lanzados desde diferentes computadoras desde un centro gestionado por *botnets* o *host zombies* (Santos et al., 2020). Por consiguiente, el escenario cuenta con una *botnet* que se encuentra generando numerosas peticiones maliciosas cronometradamente, de manera cíclica y repetitiva entre cada uno de su *hosts*, mediante ICMP. En este sentido, dado que el conmutador ya no es capaz de responder a las peticiones debido al tiempo de expiración de las entradas de flujo comienza a consultar al controlador hasta que ocurre la denegación del servicio. En la Figura 5.1 se puede observar lo explicado en este apartado.

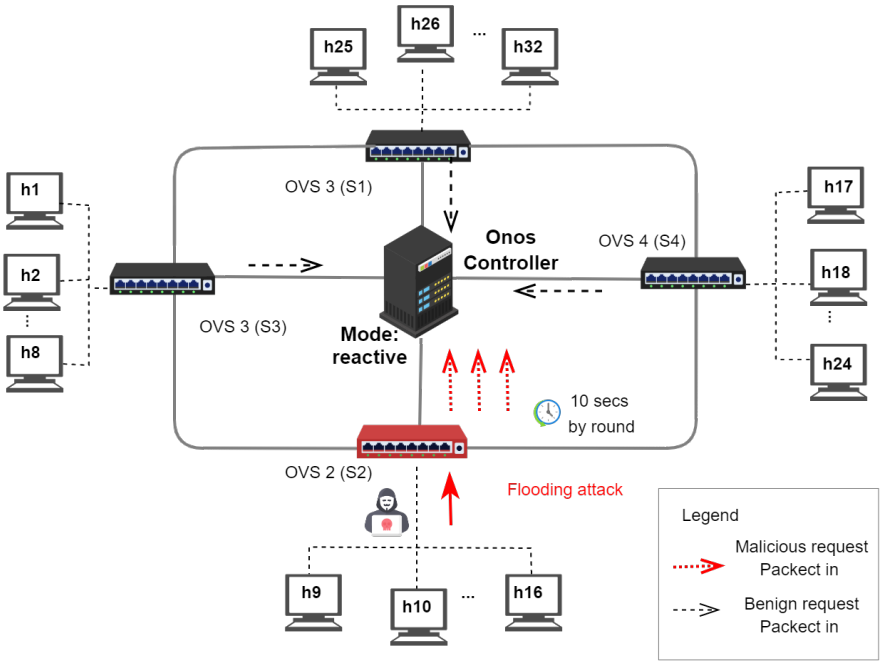

Figura 5.1: Escenario de ataque

Las premisas para este escenario se basan en la presunción de una intención maliciosa, es decir que las consultas al controlador tienen el propósito de interrumpir el servicio. Asimismo, se asume que el atacante ha logrado eludir las defensas perimetrales, como los contrafuegos, mediante el ocultamiento de su identidad y ubicación. Se asume además que el atacante conoce el funcionamiento del comportamiento de las tablas de flujo y está consciente de la alta probabilidad de que la configuración del controlador del lado la víctima se mantiene en modo reactivo. En este escenario, el atacante cuenta con recursos suficientes para generar múltiples peticiones simultáneas a través de una o varias *botnets* y por consiguiente es capaz de mantener la actividad maliciosa durante un lapso de tiempo suficiente para alcanzar el objetivo de la denegación del servicio. Por último, este escenario se fundamenta en la premisa de que el administrador de la infraestructura enfrenta limitaciones respecto a la supervisión del tráfico en la SBI, lo que dificulta la identificación de los eventos de ciberseguridad.

5.2.2 Escenario de cambios: Contexto, debilidades aprovechadas y acciones

En entornos SDN, caracterizados por su elevada dinámica, las acciones ejecutadas en uno o varios de sus componentes pueden inducir modificaciones en otros elementos. En consecuencia, resulta coherente que cada acción sobre un elemento SDN específico genere registros de actividad, dependiendo de la naturaleza de la operación. En este contexto, es esencial que los controladores registren los cambios cuando existen alteraciones a nivel de la topología de la red, actualizaciones de flujo o de los enlaces y también de acciones que afecten a las aplicaciones. Este último punto es de especial interés y constituye el enfoque central de este escenario, dada la actual preocupación sobre la ciberseguridad de las aplicaciones, como se destaca en el Capítulo 2 de esta tesis doctoral.

La relevancia de este hecho se manifiesta en que las aplicaciones pueden provenir de fabricantes de los controladores, desarrolladores de aplicaciones o terceros, lo que introduce la posibilidad de instalación de aplicaciones maliciosas. Estas aplicaciones podrían activar otras aplicaciones de manera constante, agotando recursos del controlador o incluso permitir el acceso no autorizado al control de la red. Visto esto, se vuelve crucial registrar de manera adecuada los cambios realizados en las aplicaciones.

En este sentido, en el entorno de ONOS se ha observado que la instalación o activación de una aplicación con relación de dependencia desencadena diversas reacciones en varias aplicaciones subyacentes. Esto se debe a que la eficiencia operativa de la aplicación principal depende de la activación de otras aplicaciones que son requisitos fundamentales. Desafortunadamente, en numerosas ocasiones, los cambios efectuados por las aplicaicones subyacentes no quedan debidamente registrados, lo cual representa un aspecto crítico en el ámbito de la ciberseguridad y también en el aspecto forense dado que esto dificultaría el avance de cualquier investigación.

En la Figura 5.2 se puede observar un ejemplo de las dependencias existentes entre una aplicación principal y las aplicaciones requeridas o subyacentes.

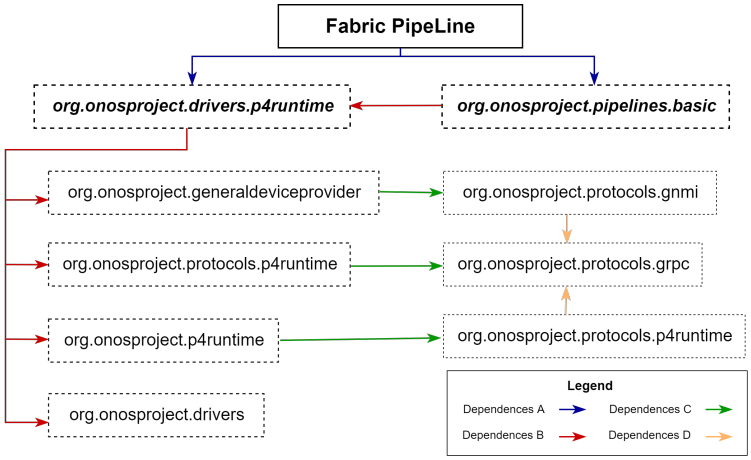

Figura 5.2: Relación de dependencia en aplicaciones SDN

Considerando este comportamiento, para este escenario se proponen acciones de cambio sobre las aplicaciones en relación de dependencia.

5.3 Ambiente de implementación

5.3.1 Especificaciones del entorno global de despliegue

Para llevar a cabo la implementación del modelo de inteligencia de filtrado, adquisición y tratamiento de datos, y de preservación de la evidencia se han utilizado entornos virtuales, los cuales están conformado por siete servidores con las características definidas en la Tabla 5.1. Cada una de estas máquinas virtuales conlleva una función relacionada ya sea con el despliegue de los componentes SDN o los escenarios, con la inteligencia de filtrado, adquisición y tratamiento de datos o con la preservación de la evidencia.

Tabla 5.1: Especificaciones técnicas de servidores para el ambiente de implementación

Componentes	**CPU**	**RAM**	**OS**
Máquina Plano de control	Intel(R) Xeon(R) E5-2650 v4 6 x 2.20GHz	12 GB	Ubuntu 20.04.5 LTS
Máquina Plano de datos	Intel(R) Xeon(R) E5-2650 v4 6 x 2.20GHz	12 GB	Ubuntu 20.04.5 LTS
Máquina CSIRT	Intel(R) Xeon(R) E5-2650 v4 6 x 2.20GHz	12 GB	Ubuntu 20.04.5 LTS
Máquina IS	Intel(R) Xeon(R) E5-2650 v4 2 x 2.20GHz	4 GB	Ubuntu 20.04.5 LTS
Máquina SS	Intel(R) Xeon(R) E5-2650 v4 2 x 2.20GHz	4 GB	Ubuntu 20.04.5 LTS
Máquina FS	Intel(R) Xeon(R) E5-2650 v4 2 x 2.20GHz	4 GB	Ubuntu 20.04.5 LTS
Máquina de Evaluación	Intel(R) Xeon(R) E5-2650 v4 4 x 2.20GHz	8 GB	Ubuntu 20.04.5 LTS

De estos siete, dos son utilizados para desplegar los componentes de la red SDN, diferenciando el plano de control y el plano de datos. Luego, cuatro representan a la infraestructura gestionada por cada uno de los interesados, siendo estos: IS, SS, FS y el CSIRT que es quien orquesta el proceso de preservación de la evidencia. Dentro del servidor gestionado por CSIRT también se despliega el modelo de inteligencia del filtrado. Finalmente, el último servidor fue utilizado para la evaluación de los modelos. La Figura 5.3 se representa gráficamente el ambiente de implementación.

5.3.2 Despliegue de escenarios: Ataque y de cambios

Despliegue del escenario de ataque

Como se explicó anteriormente para la implementación de los componentes SDN (plano de control y plano de datos) se desplegaron dos máquinas virtuales. En la primera máquina virtual se utilizó una imagen de ONOS en versión 2.8.0, instanciada sobre *Docker* (versión: 24.0.4, versión API : 1.43 y versión Go: go1.20.5). Sin embargo, dado que se necesitaba crear una nueva imagen personalizada que contenga *scripts* necesarios para la obtención de los registros del controlador, se definió un archivo *DockerFile* para generar dicha imagen.

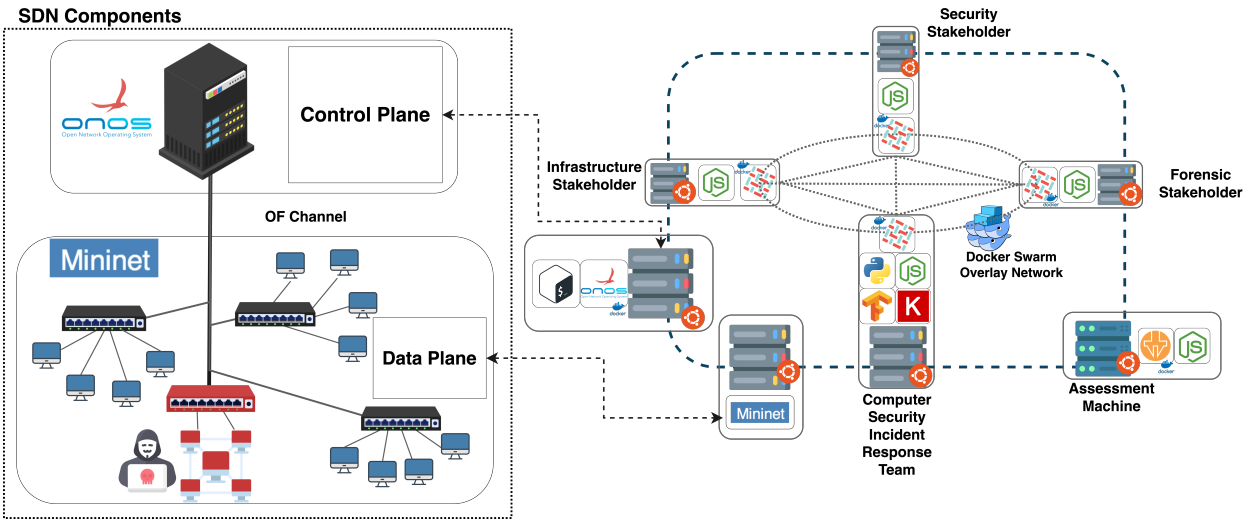

Figura 5.3: Ambiente de implementación global

De igual manera, se consideró la apertura de los siguientes puertos en *Docker*: 6633, 6653, 8181, 8101, 5005, 830. Su apertura obedece a que Openflow como tal realiza el despliegue en los puertos 6633 y opcionalmente en el puerto 6653, el puerto 8181 es donde se ejecuta la interfaz gráfica del usuario (GUI) del controlador ONOS, el 8101 sirve para usar la interfaz de línea de comandos (CLI) del controlador, el 5005 es un puerto de debbuging de ONOS, es decir por donde enviará los errores en caso de encontrarlos y por último el puerto 803 que es de verificación de *key fingerprint* oficial de *Docker*, es decir que valida que es un software auténtico.

Posteriormente, se automatizaron los servicios para iniciar o detener el controlador desarrollando un *script* en *bash* (services.sh). En este *script* también se considera la conservación del entorno de *Docker* en caso de la existencia de instancias huérfanas que impidan el acceso al controlador. Una vez que inició el controlador ONOS, se activaron dos aplicaciones importantes: *OpenFlow Provider Suite* (org.onosproject.openflow) y *Reactive Forwarding* (org.onosproject.fwd). Ambas habilitan la transmisión de paquetes usando el protocolo Openflow y de esta manera se responde a las peticiones de la red.

En la segunda máquina se instanció Mininet usando una topología personalizada. Para ello se desarrolló un *script* en *Python* (Topology.py) que despliega cuatro conmutadores OVS v1.3 en una distribución de malla, enlazados al controlador ONOS iniciado anteriormente e interconectando 32 *hosts*, como se muestra en la Figura 5.1.

Una vez desplegada la topología y con el controlador activo, se desarrollaron cuatro *scripts* en *Python*, para generar tráfico utilizando la herramienta hping3[1] . La generación de tráfico fue gestionada en cuatro etapas: *Initial flooding, fast traffic, faster traffic* y *flood attack*. La primera etapa, hace referencia al proceso normal de reconocimiento que realiza la SDN en el inicio de actividades. Es decir, es el tráfico que se genera cuando cada uno de los conmutadores envía y recibe información para reconocer el estado de la SDN. En las etapas subsiguientes

[1]https://www.kali.org/tools/hping3/

(*fast traffic, faster traffic y flood attack*), se delinearon límites distintivos entre cada categoría de tráfico considerando los valores estadísticos derivados de la observación del comportamiento de la red (Chae et al., 2019). En este caso, la determinación de tráfico y la gestión de los límites entre los diferentes rangos se basó en el análisis de las tasas de paquetes por segundo y las cargas del *payload*. En el contexto descrito, cada uno de los *scripts* incorpora temporizadores y *payloads* con cargas aleatorias dentro de los rangos predefinidos. La introducción de esta variabilidad en las cargas tiene como objetivo evitar linealidad sobre los datos.

Despliegue del escenario de cambios

Como ya se mencionó en el diseño del escenario de cambios, se ha propuesto la generación de acciones sobre aplicaciones con relación de dependencia. En este sentido se realizará la instalación y activación de las aplicaciones descritas en la Tabla 5.2, en el controlador ONOS.

Tabla 5.2: Aplicaciones SDN seleccionadas para escenario de cambios

Aplicación	ID	Categoría
Fabric PipeLine	org.onosproject.pipelines.fabric	Pipeline
BGP router	org.onosproject.bgprouter	Traffic Engineering
Artemins	org.onosproject.artemis	Monitoring
Ciena 5162 Drivers	org.onosproject.drivers.ciena.c5162	Drivers

Así también se han seguido dos enfoques para llevar a cabo las acciones de instalación y activación de las aplicaciones referidas en la Tabla 5.2. Por un lado, se llevaron a cabo estas acciones utilizando la interfaz gráfica de usuario de ONOS (ONOS GUI), y por otro lado, se empleó la interfaz de línea de comandos de ONOS (ONOS CLI). La elección de dos métodos distintos para realizar estas acciones se basa en la necesidad de generar un análisis exhaustivo del comportamiento de los registros cuando existen cambios. La interfaz gráfica de usuario (ONOS GUI) ofrece una representación visual intuitiva donde se exponen de la topología de la red y facilita la interacción con los recursos gráficos del controlador. En contraste, la interfaz de línea de comandos (ONOS CLI) proporciona una alternativa potente y eficiente, permitiendo la ejecución de tareas avanzadas y *scripts* automatizados.

5.3.3 Despliegue del modelo de inteligencia de filtrado, adquisición y tratamiento de datos

En el tercer servidor asignado para CSIRT se despliega el modelo de inteligencia de filtrado, adquisición y tratamiento de datos, Para su implementación se utilizó lenguajes de programación como *Python* y *JavaScript*, frameworks como *Keras* y *TensorFlow*, y herramientas como *Jupyter Notebook* y *Wireshark*, entre otros.

El modelo de inteligencia de filtrado, adquisición y tratamiento de datos está compuesto por el detector de tráfico inusual y el detector de comportamientos inusuales, los cuales son explicados en detalle a continuación.

Detector de tráfico inusual

La propuesta del detector de tráfico inusual, en esta implementación, utiliza modelos de inteligencia artificial. En este sentido, el primer paso fue la creación de un conjunto de datos propietario obtenido a partir de la escucha del canal Openflow en tiempo real. Se ha decidido trabajar un conjunto de datos propietario proveniente del tráfico Openflow, con el objetivo de analizar los metadatos inherentes al protocolo y aprovechar las características del tráfico, teniendo en cuenta el encapsulamiento presente. La generación de este conjunto de datos es fundamental, ya que se empleará para el entrenamiento, validación y prueba del modelo de inteligencia artificial destinado a filtrar el tráfico. En la sección subsiguiente, se proporciona información más detallada respecto a los conjuntos de datos.

Conjuntos de datos: Importancia y creación

Los conjuntos de datos son un compendio de la información que puede ser usada en el campo de la inteligencia artificial para identificar patrones, tendencias y relaciones entre las variables. Desempeñan un papel vital en el entrenamiento y evaluación de cualquier enfoque que use algoritmos de aprendizaje automático o profundo, pues el éxito o fracaso del despliegue de un modelo de inteligencia artificial se basa en la calidad de la información que le fue entregado. El tratamiento de los datos de entrada representa uno de los desafíos más significativos en la aplicación de algoritmos de inteligencia artificial, pues la presencia de datos de baja calidad o con ruido puede dar lugar a resultados imprecisos. En este sentido, se ha observado que casi el 90 % del tiempo dedicado a la implementación de un modelo de inteligencia artificial se destina a la comprensión, obtención y procesamiento del conjunto de datos.

Para este modelo se ha decidido utilizar un conjunto de datos propietario debido a que la mayoría de las soluciones existentes para realizar procesos de filtrado de tráfico se basan en conjuntos de datos provenientes de redes tradicionales, no específicas de SDN. La decisión tomada es crucial, ya que al proporcionar una solución de filtrado para las SDN, se debe considerar el proceso de encapsulamiento del protocolo, en este caso Openflow. Además, se presta atención a la comprensión detallada de la metadata necesaria para entrenar el modelo de inteligencia artificial, lo cual asegura una mayor relevancia y adaptabilidad de la solución a las características específicas de SDN.

Con el objetivo de enriquecer el entendimiento sobre el tema, se ofrece una breve explicación acerca de los conjuntos de datos disponibles y comúnmente empleados para el entrenamiento y evaluación de soluciones de detección de tráfico en redes. Es así que existen conjuntos de datos relacionados con la gestión del tráfico en redes, tanto privados como públicos, siendo estos últimos aquellos que se han puesto a disposición para servir como puntos de referencia. A continuación, se presentan las particularidades de conjuntos de datos que suelen ser utilizados para probar soluciones de seguridad para SDN:

- **KDD'99 y NSL-KDD,** son conjuntos que incluyen una amplia variedad de actividades de tráfico de redes tradicionales y por consiguiente de diversos datos, lo que los vuelve muy atractivos para realizar pruebas de detección de intrusos. El primero, se utilizó en una competencia internacional para detección de intrusiones en redes, llamada KDD Cup 1999. Este conjunto de datos contiene alrededor de cinco millones de registros de tráfico de red.

El segundo conjunto de datos representa una mejora significativa en cuanto a la calidad de los datos en comparación con el conjunto KDD'99. Con aproximadamente 130.000 registros, ofrece una representación mejorada de la información. Desafortunadamente, a pesar de la cantidad de información existente en ambos conjuntos de datos, se presentan inconvenientes significativos: la redundancia en los registros y su antigüedad. Estos conjuntos no han tenido actualizaciones sustanciales que incorporen nuevos vectores de ataque durante casi dos décadas, por lo que puede sugerir un entendimiento atemporal de las amenazas (Canadian Institute for Cybersecurity, 1999).

- **DARPA,** ha presentado conjuntos de datos tanto en 1998 como en 1999. Los conjuntos de datos incluyen tráfico normal y ataques simulados, en los cuales destacan DoS, R2L, U2R y *probing*, de redes tradicionales. En el año 2000, los datos fueron probados en escenarios denominados LLDOS 1.0, LLDOS 2.0.2 y ataques de Windows NT, en los cuales se asume que el adversario es novato y que los defensores actúan de manera ingenua (MIT, 1999). Dentro de las limitaciones de estos conjunto de datos se encuentran el ruido existente en los datos, la desactualización de ataques y se ha evidenciado el uso de una versión desactualizada de TCP.
- **ISCXIDS 2012,** fue creado por el Centro de investigación en comunicaciones, de la Universidad de New Brunswick. Los datos provienen de tráfico de red en un entorno de laboratorio simulado. Este conjunto de datos incluye actividad normal, denegación de servicio HTTP, DDoS utilizando una *botnet*, ataques de fuerza bruta SSH. Dado que fue propuesto hace más de una década no existen actualizaciones al momento (Shiravi et al., 2012).
- **CICIDS 2017,** es un conjunto de datos basado en ISCX 2012, pero contiene un mayor número de características y registros. Al ser extremadamente grande contiene redundancia y ruido en la información. En este sentido, puede haber desequilibrios en la categorización de ataques y afectar la precisión del modelo.
- **CSE-CIC-IDS2018,** tiene dos clases de perfiles *B-profiles*, relacionado con tráfico normal y *A-profiles* para escenarios de ataque. Al igual que en CICIDS 2017, existe un desbalanciamiento de las clases lo que puede limitar la capacidad de extraer características útiles para entrenar modelos precisos (Leevy et al., 2018).

Cabe indicar que los autores Elsayed et al., 2020 publicaron un conjunto de datos denominado *InSDN* tomado de tráfico Openflow. En la presentación del conjunto de datos los autores indican que han trabajado con 80 características con 56 categorías y que una de las principales limitaciones de este conjunto de datos es el desequilibrio entre las clases, lo cual puede provocar un alto nivel de falsos negativos y una baja precisión en la evaluación. Desafortunadamente, el conjunto de datos mencionado no está disponible de manera pública hasta la entrega de esta tesis, lo cual ha incentivado aún más la idea de la presentación de un conjunto de datos propietario.

Teniendo en cuenta estas observaciones y la importancia del uso de un conjunto de datos funcional para cumplir con los objetivos de esta tesis doctoral, en esta sección se detalla el proceso seguido para la creación de un conjunto de datos propio obtenido de tráfico Openflow. Para el efecto, la presente propuesta sigue una metodología de extracción, transformación

y carga denominada *ETL pipeline*, por sus siglas en inglés. Mediante el *ETL pipeline* se preparan los datos crudos del tráfico de la SBI para el análisis y posterior uso en el modelo de inteligencia artificial.

Bajo esta premisa, se desarrollaron varios *scripts* en *Python*. El primer *script* (Extractor.py) contempla la escucha activa del canal Openflow usando Pyshark, el cual es un *wrapper* de *Python* que trabaja con disectores de *Wireshark*. *PyShark* depende de la biblioteca *Pcap* que captura paquetes de red y se mantiene bajo el capó de *Tcpdump*. Durante el experimento de 4 semanas (del 25/05/2023 al 29/06/2023), se obtuvo una muestra de 522.886 instancias y mediante el primer *script* se extrajeron todas las características del paquete Openflow, haciendo uso de la metadata presente en cada uno de los paquetes. Para comprender la estructura del paquete Openflow y su desencapsulamiento se empleó *Wireshark*. El proceso de extracción de características comprende la estructura presentada en alto nivel en la Figura 5.4.

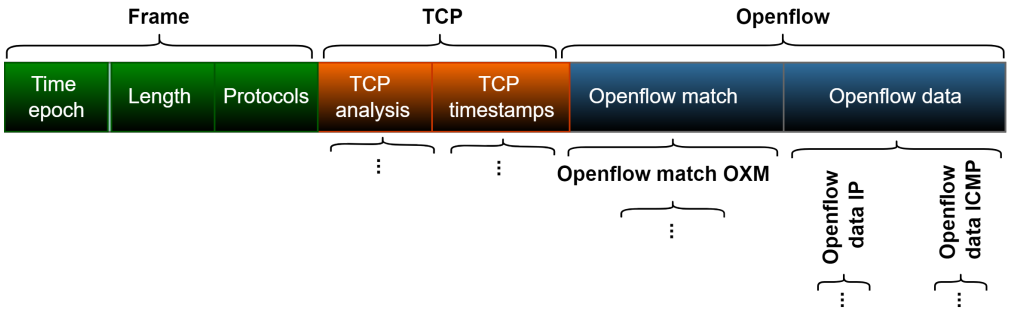

Figura 5.4: Estructura de alto nivel para extracción de metadatos en paquetes Openflow

En el proceso de extracción se obtuvieron un total de 52 características, las cuales pasaron al segundo *script* (featureextractor.py) para un proceso de reducción de la dimensionalidad. Esta reducción se la realizó con el apoyo tres técnicas basadas en el *valor F (ANOVA)*, el coeficiente de correlación de *Pearson* y el regresor RF, usando de la biblioteca *scikit-learn*[2] y con los algoritmos de aprendizaje automático de RF y LR. Como resultado de la reducción se obtuvieron las ocho características más significativas del tráfico Openflow. Las Figuras 5.5, 5.6 y 5.7 muestran los resultados de la reducción de dimensionalidad con cada uno de los algoritmos usados.

Así también en la Tabla 5.3 se detallan las características obtenidas de tráfico en el canal Openflow marcando las más representativas.

Una vez que se obtuvieron las características más significativas del tráfico OpenFlow, la información pasó por un proceso de limpieza de datos haciendo uso de otro *script* desarrollado en *Python* (Cleaner.py). Se procedió a eliminar los registros que contenían más del 75 % de datos nulos o NaN, siguiendo las mejores prácticas de los procesos *ETL*. Este umbral se seleccionó para garantizar la calidad de los datos, y en este proceso se descartó menos del 0.02 % de los paquetes.

[2]https://scikit-learn.org

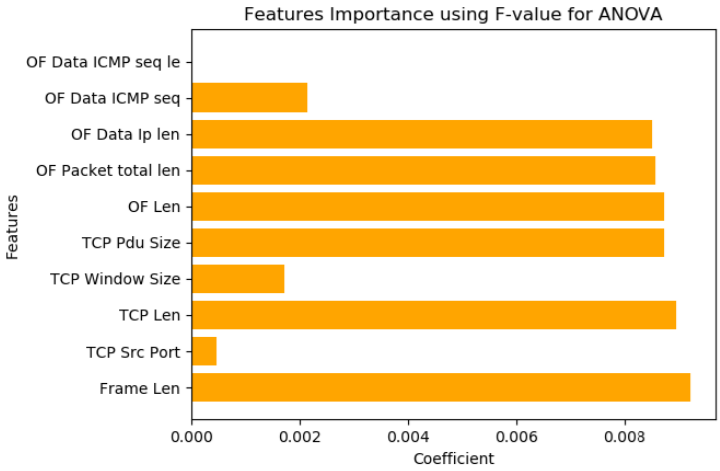

Figura 5.5: Importancia de las características usando *F-value (ANOVA)*

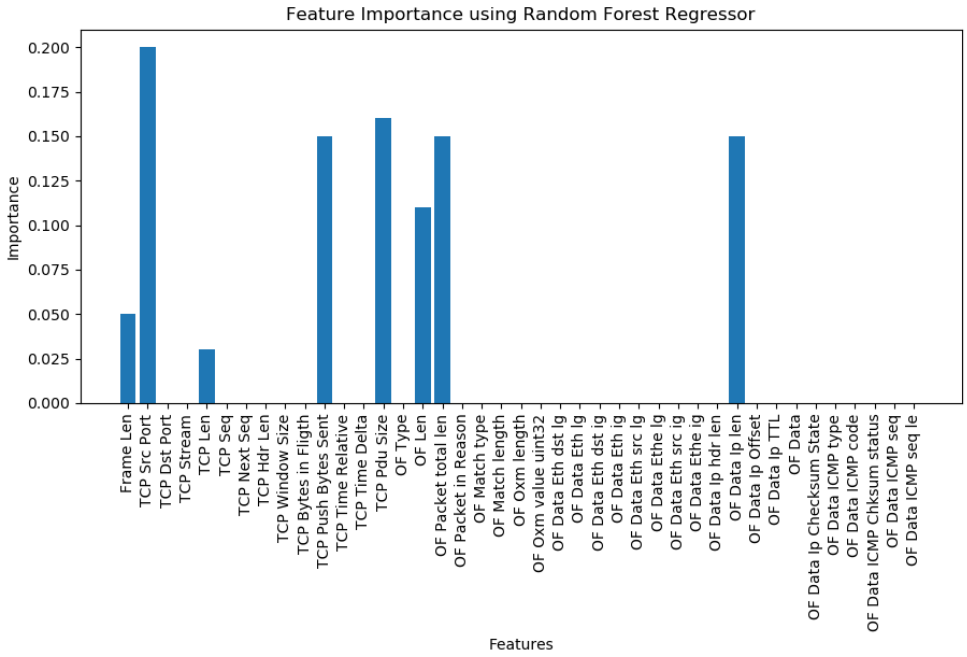

Figura 5.6: Importancia de las características usando *Random Forest Regressor*

Posteriormente, con los datos limpios, a través de un *script* (Setbuilderstad.py) se estructuraron tensores, que son grupos de paquetes Openflow estructurados como matrices, formados de 10x8, y por cada característica de cada tensor se aplicaron funciones estadísticas. Estas funciones estadísticas fueron consideradas observando el comportamiento de los datos de acuerdo con su representatividad.

Finalmente, se etiquetaron los tensores con 0 y 1. Donde 0 representa el tráfico usual y 1 el tráfico inusual. También se empleó *one-hot encoding* para el tratamiento de las variables categóricas y *standard scaler* para la normalización de los datos. En primer lugar, se optó por utilizar *one-hot encoding* como estrategia para la representación de variables categóricas.

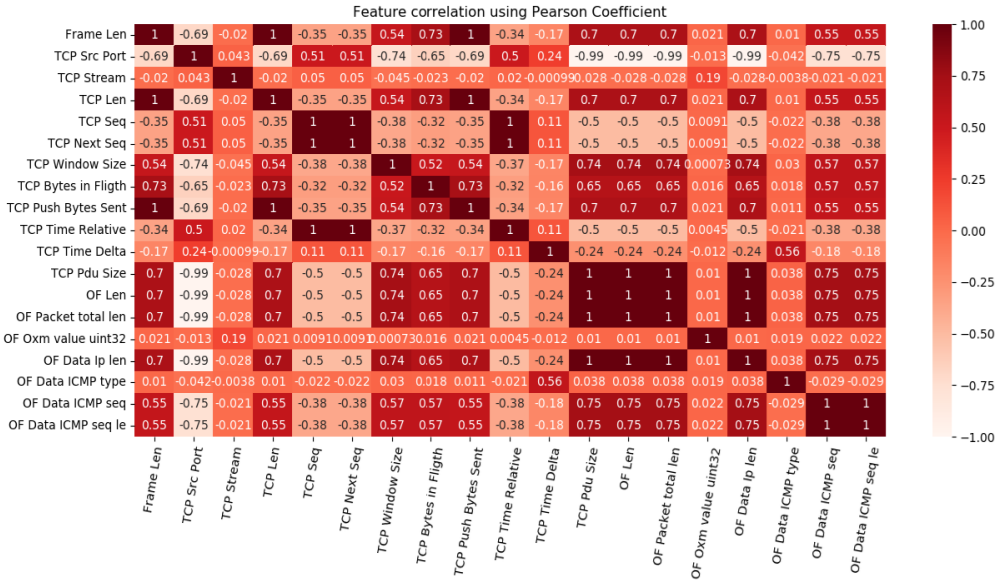

Figura 5.7: Mapa de calor de la correlación de características usando coeficiente de *Pearson*

Tabla 5.3: Características obtenidas de tráfico OpenFlow

Característica		**Característica**		**Característica**		**Característica**	
FrameTimeEpoch		TCP Bytes in Fligth	✓	OF Data Eth dst		OF Data Ip TTL	
Frame Len	✓	TCP Push Bytes Sent	✓	OF Data Eth dst lg		OF Data	
Frame Protocols		TCP Time Relative		OF Data Eth lg		OF Data Ip Checksum	
TCP Src Port		TCP Time Delta		OF Data Eth dst ig		OF Data Ip Checksum State	
TCP Dst Port		TCP Pdu Size		OF Data Eth ig		OF Data Ip src	
TCP Stream	✓	OF Type		OF Data Eth src		OF Data Ip dst	
TCP Len	✓	OF Len		OF Data Eth src lg		OF Data ICMP type	
TCP Seq		OF Packet total len	✓	OF Data Ethe lg		OF Data ICMP code	
TCP Next Seq		OF Packet in Reason		OF Data Eth src ig		OF Data ICMP Chksum status	
TCP Hdr Len		OF Match type		OF Data Ethe ig		OF Data ICMP seq	✓
TCP Window Size	✓	OF Match length		OF Data Ip hdr len		OF Data ICMP seq le	
TCP Checksum		OF Oxm length		OF Data Ip len		OF Data ICMP data time	
TCP ACK RRT		OF Oxm value uint32		OF Data Ip Offset		OF Data ICMP data len	

La elección de esta codificación se basa en que puede transformar las variables en vectores binarios, lo cual facilita su integración eficiente en modelos como el planteado en esta sección. Posteriormente, se ha optado por el uso de *standard scaler* para la estandarización de características numéricas. Esta elección radica en evitar disparidades significativas entre las características, transformándolas de manera que presenten una media de cero y una desviación estándar de uno. En este planteamiento, donde se consideran rangos para la generación de tráfico, la falta de escala podría inducir un sesgo hacia un modelo lineal, limitando su capacidad de aprendizaje a una única característica, como el tamaño del paquete. La estandarización, en este sentido, promueve un tratamiento equitativo de las variables, contribuyendo así a la robustez y generalización del modelo de AI ante diversos tipos de tráfico.

En este punto ya se ha creado un conjunto de datos propio a partir de los datos del canal Openflow. Consecuentemente, se crearon dos archivos, uno denominado *training dataset* y otro

nombrado *test dataset*. El *training dataset* trabaja con 70 % de los datos tratados y etiquetados, mientras que el *test dataset* representa el 30 % sobrante. Es importante recordar, que los datos de prueba no están etiquetados ni ordenados. Finalmente, como parte de las contribuciones de esta tesis doctoral se entrega el conjunto de datos creado, el cual se encuentra disponible en un repositorio de *Kaggle* [3].

Entrenamiento del modelo de inteligencia artificial

En el servidor CSIRT, haciendo uso de *Keras* se generó una red neuronal de 5 capas: una de entrada, una de salida y tres ocultas. La NN cuenta con 32 perceptrones en sus cuatro primeras capas y con uno en la última, lo que permiten identificar si el tráfico es *usual* o *inusual*. Para la creación del modelo se utilizaron las bibliotecas *scikit-learn, tensorflow, Pandas* y *Numpy*.

La NN trabaja con funciones de activación *Relu* y *Sigmoid*. Por una parte, *Relu* fue usada en las capas ocultas dado que es más fácil de converger en comparación con otras funciones de activación. A través de esta función de activación se introdujeron no linealidades en el modelo, con lo cual se activan las unidades con valores positivos y desactivan o apagan las unidades con valores negativos. Por otro lado, *Sigmoid*, conforme las mejores prácticas en entrenamiento de modelos, se la utiliza siempre en la capa de salida, cuando se tiene problemas de clasificación binaria, en este caso la clasificación del tráfico *usual* e *inusual* (0 y 1).

Así mismo, se activó la función *early stopping*. Esta función permite detener el proceso de entrenamiento del modelo a fin de evitar el sobreajuste de los datos, lo cual contribuye a que el modelo sea más generalizado y por ende sea capaz de predecir de mejor manera con nuevos datos. Una vez que se entrenó el modelo de AI, el siguiente paso fue su exportación e inclusión dentro de un *script* desarrollado en *Python*, lo cual es explicado más adelante.

En la Figura 5.8 se puede apreciar el comportamiento del entrenamiento del modelo en relación con la exactitud y a la pérdida.

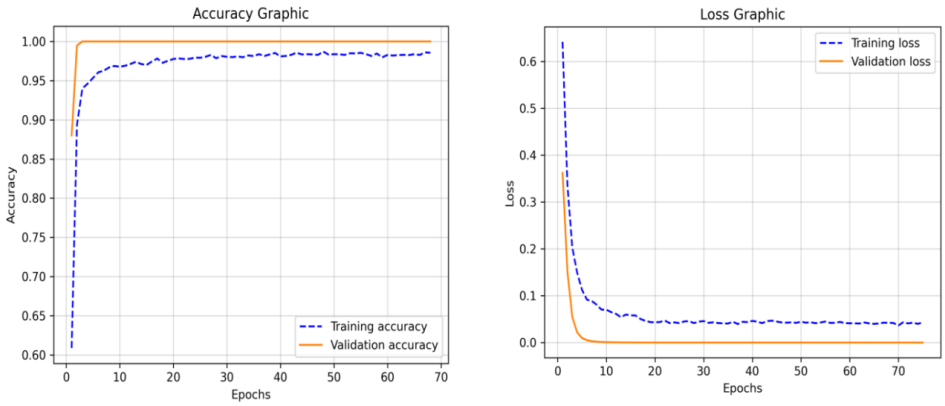

Figura 5.8: Comportamiento del entrenamiento en relación con la exactitud y la pérdida

[3]https://www.kaggle.com/datasets/datasetdfirupm/datasetdfirupm

Detector de comportamientos inesperados

El detector de comportamientos inesperados se apoya de información de aplicaciones, la topología, los dispositivos, los flujos y enlaces; y los accesos al controlador, para garantizar la completitud y coherencia de los datos. En este caso, para probar el modelo, se ha optado por la obtención de la información de dos fuentes: los registros de actividades y la información extraída mediante la API del controlador. Para la obtención de los registros de actividad se desarrolló un *script* escrito en *bash* (readLog.sh) que forma parte de la imagen personalizada del controlador y que fue definida previamente. Este *script* permite obtener los registros de actividades del controlador, reportando en tiempo real desde la instancia del controlador ONOS hasta el servidor de CSIRT *(Refiérase a Figura 5.3)*.

Simultáneamente, se implementó una aplicación basada en un microservicio dentro del servidor CSIRT desarrollado en *JavaScript*. Para esto se utilizó *Express.js* a fin de definir un servidor web mínimo y rutas, gestionar peticiones o respuestas e interactuar con una base de datos, lo cual facilitó la implementación de una API REST. La aplicación a más de exponer una interfaz para recibir los registros de actividad, realiza consultas HTTP de tipo *GET* cada segundo a la API del controlador, para obtener información de las aplicaciones, flujos, topología y hosts. Para intercambio de la información se configuró un esquema de autenticación básica HTTP.

Luego, tanto los registros de actividades como la información extraída de la API del controlador son enviados a una base de datos no relacional para su archivo efímero y consumo, en este caso se seleccionó *MongoDB 5.0.19*. En un paso siguiente, en la misma aplicaión se efectúan una serie validaciones de coherencia entre la información que se desprende de los registros en contraste con la información extraída de la API del controlador para la comprobación de cambios. En este sentido, al ir comprobando información de ambas fuentes esta aplicación va construyendo un nuevo registro que contenga información consolidada.

Constitución global del modelo de inteligencia de filtrado, adquisición y tratamiento de datos

El modelo de inteligencia de filtrado, adquisición y tratamiento de datos es un desarrollo en *Python* (Filter.py) que reune el modelo entrenado de AI para la detección de tráfico inusual y el detector de comportamientos inesperados. En este sentido, al activarse cualquiera de los dos detectores, que actúan como disparadores, este modelo comienza con la adquisición y tratamiento de datos para el proceso de preparación forense con la esquematización de la evidencia. Así también, en lo que respecta al proceso de respuesta de incidentes el *script* activa una *flow rule* para descartar las peticiones con patrones similares al modelo entrenado, mediante la API de ONOS.

5.3.4 Despliegue del modelo de preservación de evidencia

Para la implementación del modelo de preservación descrito en la sección 4.3 se ha recurrido al uso de *Hyperledger Fabric* (HLF), siendo un framework de referencia dentro de las redes *blockchain* privadas.

HLF es una parte integral del proyecto *Hyperledger* de la fundación Linux, el cual introduce

un modelo de *blockchain* privado de código abierto, brindando un entorno controlado y seguro para aplicaciones organizacionales. Se ha elegido a HLF debido a que mantiene una arquitectura modular, características de privacidad y énfasis en la gestión de identidades. La arquitectura de HLF está compuesta de un conjunto de elementos, entre ellos: nodos, autoridad de certificación, servicio de ordenamiento, canales y *chaincode*.

Los nodos son los participantes de la red, en este caso las infraestructuras de SS, IS, FS y CSIRT, siendo este último el orquestador del proceso de preservación en co-responsabilidad con los otros nodos. HLF consta de diferentes tipos de nodos:

- **Nodos pares o *peers*:** Son nodos de aprobación y compromiso desempeñan roles en la simulación, aprobación, validación y compromiso de transacciones.
- **Nodos de ordenación:** Mantienen el orden de las transacciones para garantizar la consistencia en toda la red.
- **Nodos de cliente:** Inician transacciones enviando propuestas a nodos de aprobación.

Por otra parte, la autoridad de certificación es la encargada de la emisión de certificados X.509, lo cual permite mantener comunicación segura mientras se autentican y autorizan transacciones en la red *blockchain*.

Mientras que el servicio de ordenamiento, que está constituido por los nodos de ordenación es responsable de la creación de bloques y el mantenimiento del orden de las transacciones en la red *blockchain*, garantizando que todas las partes de la red tengan el mismo historial de transacciones. Este servicio, vital para evitar inconsistencias en el libro contable (*ledger*, en inglés), utiliza algoritmos de consenso como *RAFT* o *Kafka*.

Respecto a los canales, son un mecanismo para permitir la comunicación y transacciones privadas entre los participantes en la red *blockchain*. En detalle, un canal es una cadena de bloques independiente al cual solo los participantes autorizados tienen acceso.

Finalmente en el núcleo de HLF se encuentra el concepto de contratos inteligentes, denominado *chaincode*, en el cual se definen las reglas para la ejecución de transacciones en la red. Los *chaincode* son programas modulares y extensibles que se codifican en lenguajes como *Go* o *JavaScript*, ejecutándose dentro de contenedores *Docker* seguros. Debido a esta flexibilidad, es posible desarrollar contratos inteligentes personalizados. El *chaincode* se instala e instancia en *peers* o nodos de aprobación, con capacidades de actualización sin problema, permitiendo adaptarse continuamente a la lógica del negocio sin interrumpir el servicio de transaccionalidad en la red *blockchain*.

En esta implementación se seleccionó a HLF versión 2.2. Como se puede observar en la Figura 5.3, los nodos de la red *blockchain* están implementados sobre los servidores que representan a los interesados: IS, SS, FS. CSIRT. Cada servidor ejecuta un nodo peer normal y un nodo de ordenamiento. En esta implementación, para el servicio de ordenamiento se utiliza *RAFT*, un protocolo tolerante a fallos para lograr el consenso entre los nodos ordenamiento con respecto al orden de las transacciones. Todos los *peers* forman un canal, por lo tanto, comparten un solo registro o *ledger*. El contrato inteligente (*chaincode*), se ha desarrollado en *Javascript* empleando la SDK provista por HLF. De igual manera, es preciso indicar que el material

Tabla 5.4: Registro de evidencia

Id	***Start***	***End***	***Event Type***	***Event Details***

62d2d15118dc9f8bef3397f0	1657983272310	1657983272641	Application Activation Event	{
				“eventDescription”:[“Removing features: onos-apps-bgprouter[2.8.0.SNAPSHOT,2.8.0.SNAPSHOT]”,
				“Changes to perform:”,
				“Region: root”,
				“Bundles to uninstall:”,
				“org.onosproject.onos-apps-bgprouter/2.8.0.SNAPSHOT”,
				“org.onosproject.onos-apps-routing-common/2.8.0.SNAPSHOT”,
				“Stopping bundles:”,
				“org.onosproject.onos-apps-routing-common/2.8.0.SNAPSHOT”,
				“Unregistering commands for bundle org.onosproject.onos-apps-routing-common/2.8.0.SNAPSHOT”,
				“BgpSessionManager stopped”,
				“org.onosproject.onos-apps-bgprouter/2.8.0.SNAPSHOT”,
				“BgpRouter stopped”,
				“Uninstalling bundles:”,
				“org.onosproject.onos-apps-bgprouter/2.8.0.SNAPSHOT”,
				“org.onosproject.onos-apps-routing-common/2.8.0.SNAPSHOT”,
				“Refreshing bundles:”,
				“org.onosproject.onos-apps-bgprouter/2.8.0.SNAPSHOT (Bundle will be uninstalled)”,
				“org.onosproject.onos-apps-routing-common/2.8.0.SNAPSHOT (Bundle will be uninstalled)”,
				“Done.”,
				“Application org.onosproject.bgprouter has been deactivated”],
				“requiredApps”:[“org.onosproject.fibinstaller”,
				“org.onosproject.route-service”]}”,
				“___v”:0
				}

criptográfico de los nodos fue generado previamente usando el binario *cryptogen* disponible en la herramienta.

En la Figura 5.3 se puede observar que todas las instancias de HLF están siendo ejecutadas sobre *Docker* y deben interconectadas, para lo cual se utiliza *Docker Swarm* para generar una red *overlay* y de esta manera poder orquestar adecuadamente las instancias.

También se debe indicar que, conforme a lo señalado en la sección 4.3, donde se describe el modelo de preservación de la evidencia, se considera la premisa de una evidencia ya consolidada. En la Figura 5.9 se presenta la estructura de la evidencia utilizada para este despliegue y en la Tabla 5.4 se presenta una muestra de un registro de evidencia.

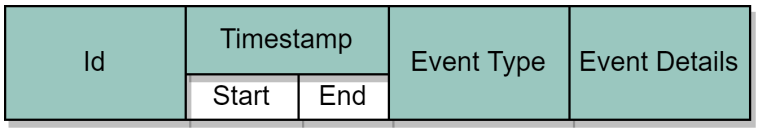

Figura 5.9: Estructura de evidencia

5.4 Evaluación de los modelos

5.4.1 Modelo de inteligencia de filtrado, adquisición y tratamiento de datos

Desempeño individual

En este caso es pertiente evaluar el desempeño del modelo de AI dentro del modelo de inteligencia de filtrado, adquisición y tratamiento de datos. Para ello, se han considerado *folds* (pliegues o particiones), los cuales se refieren a la división del conjunto de datos en subconjuntos de menor proporción para realizar una validación cruzada. En este caso, se ha considerado una validación cruzada con K=5, es decir *5-fold cross-validation*. De igual manera, se han seleccionado las siguientes métricas de rendimiento para evaluar el modelo de AI presentado:

- *Accuracy* (Exactitud)

  Esta métrica se calcula dividiendo la suma de las predicciones correctas (positivas y negativas) entre el número total de instancias.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{T}} \tag{5.1}$$

- *Precision* (Precisión)

  Precisión se centra en la calidad de las predicciones positivas. La precisión es la relación entre el número de verdaderos positivos y el número total de predicciones positivas realizadas (verdaderos positivos y falsos positivos). Esta métrica se calcula como:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5.2}$$

- *Recall* (Sensibilidad)

  Es la proporción de los verdaderos positivos, frente a la suma de verdaderos positivos y falsos negativos . *Recall* se calcula como:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5.3}$$

- *F1-score*

  Es una métrica que combina *Precision* y *Recall*, proporcionando un equilibrio entre la calidad de las predicciones positivas y la capacidad de identificar todas las instancias positivas. Es especialmente útil en situaciones donde hay un desequilibrio entre las clases. *F1-score* se calcula como la media armónica de *Precision* y *Recall*:

$$F1\text{-}score = \frac{2 \times (\mathrm{Precision} \times \mathrm{Recall})}{\mathrm{Precision} + \mathrm{Recall}} \tag{5.4}$$

En la Figura 5.10 se observa la evaluación realizada considerando los valores promedios de: *Accuracy, Precision, Recall y F1 Score* por cada *fold*. De igual manera, en la Tabla 5.5 se detallan cada uno de los valores obtenidos en cada *fold* junto con su valor promedio.

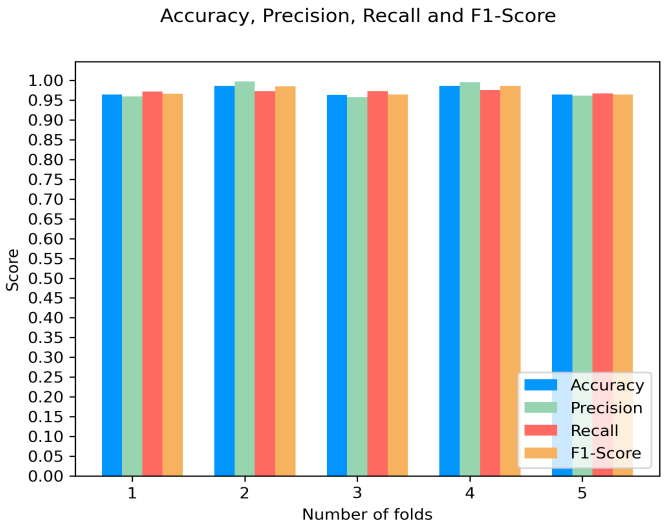

Figura 5.10: Evaluación global por cada *fold*

Tabla 5.5: Resultados de la evaluación global por *folds*

Folds	Accuracy	Precision	Recall	F1-Score
Fold 1	0.965	0.96	0.972	0.966
Fold 2	0.986	0.998	0.973	0.985
Fold 3	0.964	0.958	0.973	0.965
Fold 4	0.986	0.996	0.976	0.986
Fold 5	0.965	0.962	0.967	0.965
Promedio	0.9732	0.9748	0.9722	0.9734

En esta fase los resultados promedio reflejan que el modelo de AI tiene un rendimiento de alto nivel, indicando que el 97.32 % de todas las predicciones son correctas. Así también este modelo ha sido capaz de clasificar una tasa del 97.48 % de verdaderos positivos de las instancias que realmente son positivas (verdaderos y falsos), es decir tiene una capacidad sólida para evitar hacer predicciones positivas incorrectas. Esto va en concordancia con el valor de la captura del 97.22 % de todas las instancias positivas presentes en el conjunto de datos. De igual manera, como se explicó, a pesar que el valor de *F1-score* suele ser utilizado en situaciones donde hay un desequilibrio en la distribución de clases, se ha optado por analizarlo, obteniendo un 97.34 % lo cual indica que existe un excelente equilibrio entre la capacidad del modelo de AI para clasificar correctamente instancias positivas y su habilidad para evitar falsos positivos.

En resumen, los valores promedios de las métricas expuestos en la Tabla 5.5 sugieren que el modelo de manera global tiene un alto rendimiento en la clasificación.

Por otra parte, en la Figura 5.11 se observa la evaluación del modelo tomando en consideración las medidas de *Accuracy, Precision, Recall y F1-Score* de las clases o categorías. En este caso, tráfico *usual* e *inusual*. Así también en la Tabla 5.6, se presenta el detalle de los resultados.

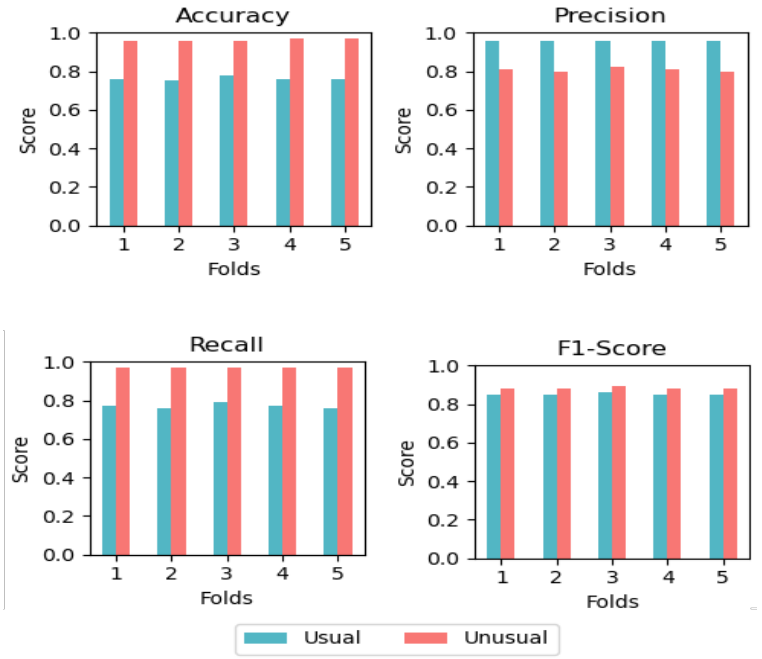

Figura 5.11: Evaluación de clases por *fold*

Tabla 5.6: Resultados de la evaluación por clases y por *fold*

Folds	Clase	Accuracy	Precision	Recall	F1-Score
Fold 1	***Usual***	0.76	0.96	0.77	0.85
Fold 2		0.75	0.96	0.76	0.85
Fold 3		0.78	0.96	0.79	0.86
Fold 4		0.76	0.96	0.77	0.85
Fold 5		0.76	0.96	0.76	0.85
Promedio		0.762	0.96	0.77	0.852
Fold 1	***Unusual***	0.96	0.81	0.97	0.88
Fold 2		0.96	0.8	0.97	0.88
Fold 3		0.96	0.82	0.97	0.89
Fold 4		0.97	0.81	0.97	0.88
Fold 5		0.97	0.8	0.97	0.88
Promedio		0.964	0.808	0.97	0.882

Conforme los resultados obtenidos se puede ver que el 76.2 % de todas las instancias (tanto verdaderas positivas como verdaderas negativas) fueron clasificadas correctamente para la clase denominada *usual*. Del total de instancias clasificadas como clase *usual*, el 96 % son verdaderamente instancias de esta clase. Esto sugiere que, cuando el modelo predice la clase *usual*, tiende a ser preciso, de esta manera ha capturado correctamente el 77 % de todas las instancias que son de la clase *usual*. También al obtener un *F1-score* de 85.2 % se observa buen equilibrio entre *precision* y *recall* para la esta clase.

En lo que respecta a la clase *inusual*, los resultados revelan que el 96.4 % de todas las instancias fueron clasificadas correctamente para la esta clase. Del total de instancias clasificadas como clase *inusual*, el 80.8 % son verdaderamente instancias que pertenecen a esta clase y de esta forma el modelo de AI ha logrado capturar correctamente el 97 % de casos reales de la clase *inusual*. Por último, el 88.2 % de la métrica *F1-score* sugiere un buen equilibrio entre *precision* y *recall* para la clase *inusual*, al igual que el de la clase *usual*.

Como otro método de evaluación se ha seleccionado el área bajo la curva ROC (*AUC-ROC*). Esta medida de evaluación permite revisar el rendimiento global del clasificador, es decir mide el rendimiento y la capacidad de discriminación de este modelo de clasificación binaria. *AUC-ROC* trabaja la relación entre la tasa de verdaderos positivos (sensibilidad o *Recall*) y la tasa de falsos positivos a medida que varía el umbral de decisión del clasificador. La Figura 5.12 muestra los resultados obtenidos en los cinco *folds*.

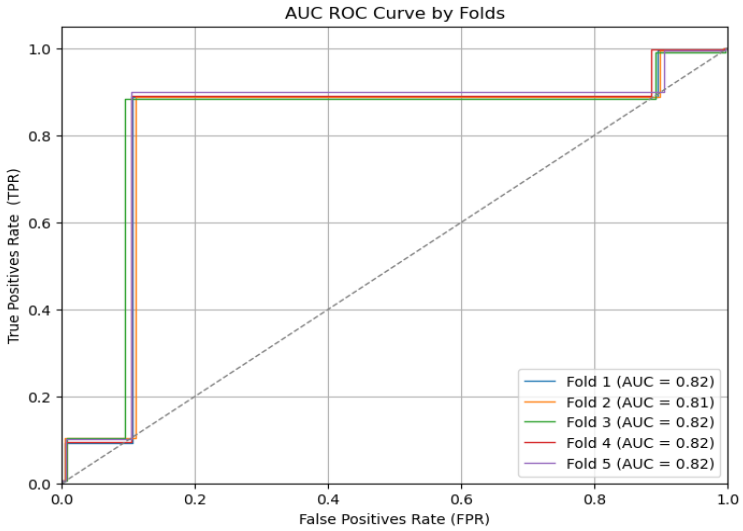

Figura 5.12: Evaluación *AUC-ROC*

Considerando que un valor de AUC cercano a 1.0 indica un buen rendimiento del modelo en términos de clasificación, mientras que un valor cercano a 0.5 sugiere un rendimiento similar al azar, los resultados revelan que los valores de AUC son relativamente consistentes en todos los *folds*, manteniendo un rango de 0.81 a 0.82, lo que indica que el modelo tiene una capacidad consistente para discriminar entre clases positivas y negativas en diferentes subdivisiones del conjunto de datos.

Desempeño a nivel comparativo

En la Tabla 5.7 se presentan las principales diferencias encontradas entre el modelo de AI propuesto en esta tesis doctoral en comparación con varios trabajos relacionados, que han sido ampliamente abordados en la sección 3.3.4. En la Tabla 5.7, se han considerado las investigaciones más recientes y representativas que exploran el uso de algoritmos de inteligencia artificial en el ámbito de la ciberseguridad en SDN, con un enfoque particular en los conjuntos de datos empleados, los mecanismos para reducción de dimensionalidad, el ataque relacionado,

los algoritmos de inteligencia artificial usados, el posicionamiento de la solución de detección y los resultados de exatitud del modelo.

Referencia	Tipo de tráfico	Conjunto de datos	Mecanismos de reducción de dimensionalidad	Tipo de ataque considerado	Modelo de AI usado	Posicionamiento de la solución de detección	Exactitud del modelo de AI
(Maeda et al., 2019)	Tradicional	CTU-13/ ISOT/ propietario	Sript propietario	Botnets	MLP	Plano de aplicación	99.20 %
(Song et al., 2017)	Tradicional	KDD'99	RF	U2R / R2L/DoS/ Probe	RF	Plano de aplicación	98-99 %
(Garg et al., 2019)	Tradicional	KDD'99/ CMU/ TIET	RBM	HTTP	SVM	Plano de aplicación	99.02 %
(Malik et al., 2020)	Tradicional	CICIDS 2017	CNN	Port scan/ cross site scripting/ botnet	LSTM	Plano de control	98.60 %
(Tang et al., 2019)	Tradicional	NSL-KDD/ CICIDS2017	Manual por referencias	DoS/ R2L/ U2R/Probe	GRU- RNN	Plano de control	89 %
(C. Li et al., 2018)	Tradicional	ISCX2012	No definido	DDoS	RNN	Plano de datos	99 %
(Sahoo; Tripathy et al., 2020)	Tradicional	NSL-KDD	KPCA	DDoS	SVM + GA	Plano de control	98 %
(Makuvaza et al., 2021)	Tradicional	CICIDS 2017	Manual/ Literature reference	DDoS	DNN	Plano de datos	97.59 %
(Dehkordi et al., 2021)	Tracicional	ISCX / CTU-13/ ISOT	Manual	DDoS	BayesNet, J48, RandomTree, LR REPTree	Plano de control	99.48 %
(Fouladi et al., 2022)	Tradicional	MAWI/ FIFA	DWT	DDoS	AE-NN	Plano de datos	No definido
(Singh et al., 2022)	Tradicional	KDD'99	IU-ROA	DDoS	CNN	Plano de control	90.06 %
(Gadallah et al., 2024)	Tradicional	NSL-KDD	Manual	DDoS	BGRU	Plano de control y datos	99.91 %
Esta tesis doctoral	**SDN**	**Propietario**	**RF+LR**	**DDoS**	**DNN**	**Plano de control y datos**	**97.2 %**

Tabla 5.7: Evaluación comparativa con otras propuestas

5.4.2 Modelo de preservación de evidencia

Para el modelo de preservación se ha considerado adecuado evaluar el comportamiento de la red *blockchain*, para lo cual se ha elegido analizar la tasa de rendimiento efectiva (*throughput*) y la latencia.

- Tasa de rendimiento efectiva

  Una vez instanciado el *chaincode* entre los nodos que conforman la red descentralizada, la primera métrica a evaluar es la tasa de rendimiento efectiva o *throughput*, la cual representa la cantidad de transacciones que la red puede procesar exitosamente en un periodo de tiempo. Para este experimento, la red ha sido sometida a un conjunto de ráfagas de transacciones (500 transacciones), en diferentes tasas, iniciando en 25 TPS y hasta las 500 TPS en intervalos de 25.

  A las transacciones de este conjunto las podemos clasificar en dos tipos: transacciones de lectura o consulta y transacciones de escritura. Para el primer tipo solo se realiza una operación de lectura al *ledger* mas no una modificación por lo que no se requiere un consenso. En el caso del segundo grupo, estas transacciones implican una operación que modifica el estado del *ledger* y por lo tanto requieren un consenso de los nodos de la red. Ambos tipos de transacciones invocan funciones definidas en el *chaincode* tanto para lectura como para escritura.

  Dentro de la evaluación cada ráfaga de transacciones tanto de lectura como de escritura fue ejecutada 30 veces, y basados en los resultados entregados por la herramienta de

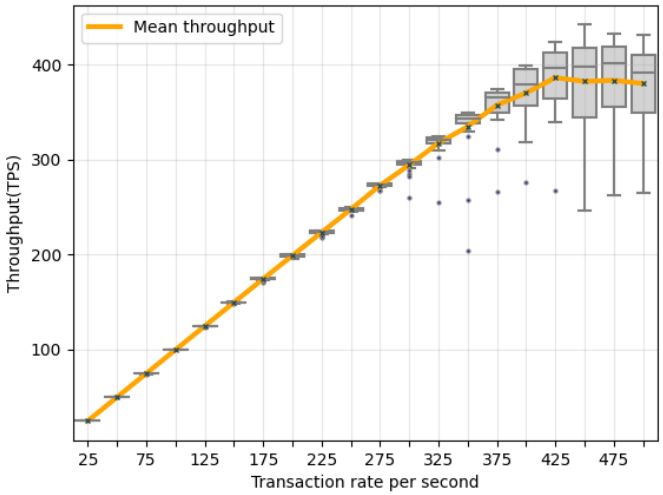

Figura 5.13: *Throughput* medio para el caso de transacciones de lectura sobre el *ledger*

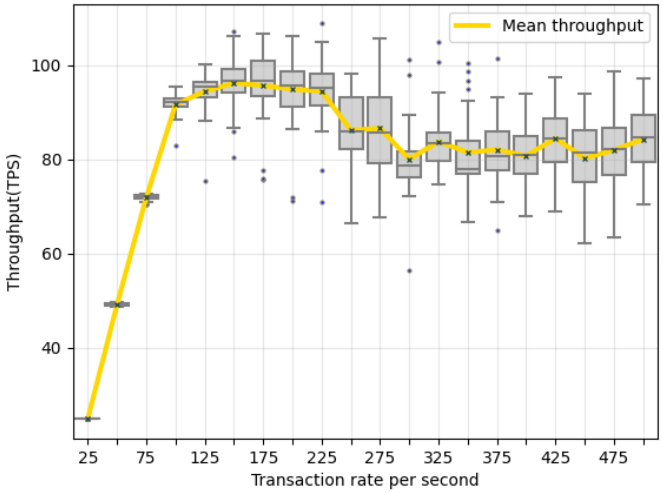

Figura 5.14: *Throughput* medio para el caso de transacciones de escritura sobre el *ledger*

medición se pudo determinar un valor de medio de *throughput* (Ψ) para cada caso. Como se puede apreciar en las Figura 5.13 y 5.14.

En el caso de transacciones que involucran operaciones de lectura (Figura 5.13), Ψ crece linealmente hasta la ráfaga cuya tasa de transacción por segundo es de 250. A partir de ese punto y hasta la décimo séptima ráfaga se observa que Ψ crece en menor proporción. Desde ese punto, se observa que el valor de Ψ oscila ligeramente entre los 385 TPS y 380 TPS, siendo el primero el valor máximo promedio de *throughput* para la operación de lectura. En el caso de operaciones de escritura sobre la red (Figura 5.14), Ψ crece linealmente únicamente hasta los 75 TPS, posterior a eso, su valor siempre es

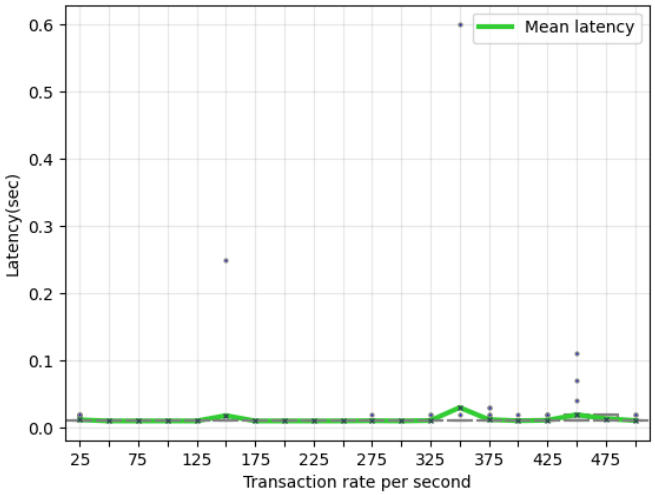

Figura 5.15: Latencia media para el caso de transacciones de lectura sobre el *ledger*

inferior al de la tasa transaccional, lo que significa que la red no es capaz de escribir el número total de transacciones por segundo sobre la ledger en cada ráfaga. Por ejemplo, para el caso de la quinta ráfaga, Ψ alcanza un valor de 93.7 TPS, y no es sino hasta el séptimo grupo de transacciones donde se alcanza el Ψ_{MAX} de 95.4 TPS. A partir de ese momento el valor el valor se mantiene ligeramente igual hasta la ráfaga cuya tasa de transaccionalidad es de 225 TPS. Posterior a ese punto, Ψ disminuye y sus valores fluctúan entre 86.50 TPS y 79.80 TPS.

- Latencia

  La segunda métrica evaluada es la latencia, la cual representa el intervalo de tiempo transcurrido entre la propuesta de transacción y su ejecución. Figuras 5.15 y 5.16 ilustran el valor medio de esta métrica (Γ) para las operaciones lectura y de escritura respectivamente.

  En el caso de transacciones de lectura, Γ se mantiene oscilando entre 0.010 seg y 0.013 seg a lo largo del experimento. Sin embargo, existen tres casos particulares en donde Γ supera dicho rango. Es así que en el sexto grupo de transacciones, Γ logra un valor medio de 0.018 segundos. El segundo caso ocurre en la ráfaga donde la tasa de transaccionalidad es de 350 TPS, en este punto Γ alcanza su valor máximo (Γ_{MAX}) de 0.030 segundos, más del doble del valor medio de todo el experimento. Finalmente, el ultimo valor inusual de Γ, se observa en el antepenúltimo conjunto de transacciones (450 TPS), con 0.018 segundos.

  En el caso de transacciones de escritura, Γ crece hasta los 3.63 segundos a medida de que la tasa de transacionalidad aumenta hasta las 300 TPS. A partir de ese punto el valor de Γ oscila entre un rango de 3.73 segundos y 3.30 segundos, siendo el primero el valor máximo promedio de latencia durante esta evaluación.

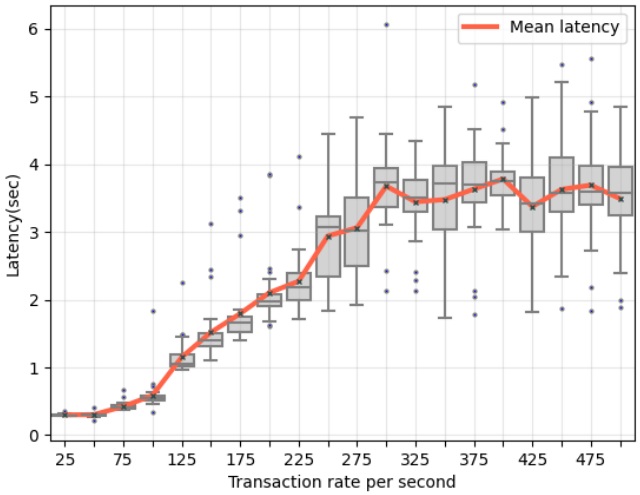

Figura 5.16: Latencia media para el caso de transacciones de escritura sobre el *ledger*

5.5 Resumen del capítulo

En este capítulo se presentó la implementación y validación de las contribuciones detalladas previamente en el capítulo 4. Se estableció un entorno de validación enfocado en la gestión de preparación forense y respuesta a incidentes de seguridad en SDN. Este entorno abarcó dos eventos principales: un ataque al controlador SDN y cambios en los elementos SDN. Para lo cual, se elaboró un escenario de ataque de tipo DDoS utilizando el controlador ONOS, aprovechando una debilidad encontrada en la gestión de las tablas de flujo en modo reactivo. En lo que refiere al escenario de cambios, se presentó el contexto, debilidades aprovechadas y acciones a efectuar sobre algunas aplicaciones SDN. De igual manera, se expuso en detalle el entorno de implementación de los modelos de inteligencia de filtrado, adquisición y tratamiento de datos, así como de preservación de evidencia, junto con los paquetes de software utilizados. Por último, se efectuaron evaluaciones sobre los modelos y se presentaron los resultados en función de sus métricas.

En consecuencia, a través de esta implementación y validación, se comprobó la viabilidad técnica de integrar procesos de respuesta a incidentes con procesos de preparación forense en SDN, destacando el uso de varias herramientas, lenguajes de programación, manejo de datos y la comprensión de protocolos. Por supuesto, esta validación representa un punto de partida fundamental para futuros avances en la implementación de DFIR en entornos SDN. Es importante destacar que esta investigación pretende extenderse a topologías más complejas y con diferentes escenarios de ataque y cambios. Asimismo, se vislumbra la oportunidad de ampliar la revisión de los nodos de ordenamiento en el modelo de preservación, así como de incorporar nuevas técnicas y herramientas que enriquezcan y fortalezcan aún más el despliegue de estos modelos, para aportar soluciones relevantes y eficaces en el ámbito de la ciberseguridad en SDN.

Capítulo 6

CONCLUSIONES, FUTURAS LÍNEAS DE INVESTIGACIÓN Y DIVULGACIÓN DE RESULTADOS

6.1 Conclusiones

En la presente tesis doctoral se ha presentado un conjunto de contribuciones orientadas a la gestión de incidentes de ciberseguridad y preparación forense en redes definidas por software, las cuales han sido reunidas en los capítulos 2, 3, 4 y 5 .

La primera parte de la investigación y primera contribución se centró en comprender en profundidad los inconvenientes de ciberseguridad presentes en las SDN, a fin de exponer ampliamente la problemática que debía ser abarcada. Para ello, mediante el estudio del estado del arte, se exploró la arquitectura SDN, identificando los problemas de ciberseguridad en cada uno de sus planos e interfaces, y se analizaron las soluciones más representativas con la metodología de modelado de amenazas STRIDE. Durante esta parte de la investigación, se concluyó que las propuestas referenciadas en el estado del arte abordan los problemas de ciberseguridad de forma fragmentada en lugar de integrada. Es decir, se proponen soluciones para el ámbito de ciberseguridad sin considerar otras preocupaciones como las de índole legal, pues un incidente puede llegar a tener repercusiones en campos jurídicos. En este sentido se observó una gran oportunidad de investigación en las áreas de respuesta a incidentes y de preparación forense.

Posteriormente, avanzando en la investigación, se identificaron limitaciones en los marcos de referencia presentados por la ISO y por el NIST en relación a la gestión de pruebas digitales y gestión de incidentes, respectivamente. Concluyendo que cada uno de estos marcos está destinado a ser usado desde un enfoque investigativo o desde un enfoque técnico, sin existir una sinergia entre ellos a pesar de su proximidad, por lo que cubren parcialmente las necesidades reales de las organizaciones. También se analizó la inclusión de las ciencias forenses y de respuesta a incidentes de ciberseguridad en SDN, encontrando una total fragmentación de las áreas y determinando que las SDN han tenido un retraso en la incorporación de estas

disciplinas debido a los retos conceptuales, pragmáticos o de entorno.

De igual manera, en la revisión de los actos jurídicos de contexto europeo relativos a la ciberseguridad, se llegó a concluir que hace falta mantener un elevado nivel de conocimientos técnicos para ayudar a los Estados miembros de la Unión Europea a mejorar la capacidad y preparación para abordar incidentes relacionados con la seguridad de redes e información, evitando la duplicación de tareas y promoviendo las sinergias regionales.

A fin de abordar estas carencias, el tercer capítulo de esta tesis doctoral introdujo un framework y una arquitectura para la preparación forense y respuesta a incidentes de ciberseguridad en SDN, con lo cual se logró satisfacer necesidades presentes en las áreas técnicas y legales ante la presencia de incidentes de ciberseguridad en las SDN. En la concepción de estas propuestas se logró derribar cuidadosamente algunos retos que impedían la unificación de estas dos disciplinas y también ampliar la visión de gestión de la ciberseguridad para promover su uso. Dentro de las propuestas de este capítulo, se hace evidente el aporte significativo respecto a la relación generada entre ambos conceptos, respetando adecuadamente cada uno de sus campos de acción para evitar conflictos entre ambas cuando se articula la información requerida para el proceso DFIR. Dentro de las contribuciones, también se prestó atención significativa a la participación de interesados, llevando a la propuesta a mantener el componente de gestión más allá de que se introduzca únicamente como una herramienta tecnológica.

Continuando con las contribuciones, en el cuarto capítulo se presentaron dos modelos prevalentes para usarlos en la arquitectura propuesta. Se eligió el desarrollo de dos modelos debido a la amplitud de la arquitectura, la prioridad de cada uno de ellos y en consecuencia su profundidad. Es así que, se desenvolvió el modelo de filtrado, adquisición y tratamiento de datos, y el modelo de preservación. Ambos fueron elegidos por considerarse puntos críticos del proceso DFIR. Por un lado, el primer modelo permitió usarse como un disparador de inicio de los procesos, lo cual básicamente se vuelve la columna vertebral de la propuesta, ya que con un correcto filtrado se logra obtener una selección adecuada de información, imprescindible para iniciar una investigación legal o para hallar la causa raíz de manera técnica. Mientras que, la preservación de la evidencia se torna una parte neurálgica, pues a través de presentación del modelo se pudo cumplir con los principios de admisibilidad de la evidencia, respetando la cadena de custodia en lo relativo a su almacenamiento. De esta manera, el modelo explicó claramente la compartición de la información a cada interesado en el proceso DFIR conforme los roles que le son asignados dentro de una red privada y descentralizada de nodos.

Para la conceptualización del primer modelo, se investigó el uso de inteligencia artificial junto con una variedad de técnicas y algoritmos, así como la aplicación de máquinas de estado finitas. Dada la importancia de esta propuesta, se optó por la obtención de un conjunto de datos propietario tomado de la interfaz sur. Se hace especial énfasis en este punto, dado que generalmente las propuestas que hacen procesos de filtrado usan conjuntos de datos de redes tradicionales, lo cual puede impactar en la resolución del problema y por consiguiente en su aplicabilidad. Para ello, también fue necesario profundizar en los procesos de extracción, transformación y carga de datos (ETL) del tráfico de red, considerando la estructura de encapsulamiento de los paquetes. En cuanto al segundo modelo, se eligió trabajar con tecnologías de ledger distribuidos privados o permisionados para mantener la evidencia en un entorno seguro y transparente con los interesados del proceso DFIR. Esta decisión permitió

registrar y auditar transacciones, asegurando la integridad de la información y permitiendo un acceso controlado y autorizado. La comprensión de esta tecnología fue esencial para poder ofrecer una solución robusta.

El quinto capítulo está relacionado con la validación de las propuestas referidas en el cuarto capítulo, lo cual permitió demostrar la viabilidad de las propuestas en un entorno virtualizado con herramientas de software, presentando los resultados conforme sus propias métricas de evaluación. Así para el modelo de inteligencia de filtrado, adquisición y tratamiento de datos se consideraron las métricas de *Accuracy, Precision, Recall, F1-score* y *AUC-ROC*, observando un rendimiento robusto en la clasificación de tráfico inusual. En este punto, aunque los valores de precisión para la clase usual son elevados, se identificó una oportunidad de mejora en el Recall. En la evaluación del modelo de preservación, se analizó el desempeño de la red *blockchain* en términos de su *throughput* y latencia. En general, se observó un rendimiento satisfactorio tanto para transacciones de lectura como de escritura, aunque se identificó un área de mejora en las transacciones de escritura. Esto sugiere la necesidad de analizar la cantidad de nodos de ordenamiento en relación con los recursos asignados para optimizar este aspecto.

En conclusión, a través de esta tesis doctoral, se ha demostrado la viabilidad de integrar los conceptos de preparación forense y de respuesta a incidentes de manera armoniosa, sin comprometer los procedimientos establecidos en cada una de las disciplinas. Confirmando que esta investigación es pionera en ofrecer una visión holística para responder a los desafíos de ciberseguridad que enfrentan los despliegues de SDN.

Las contribuciones presentadas representan un avance significativo en la gestión de la ciberseguridad en entornos SDN. Estas contribuciones, que incluyen un framework, una arquitectura y dos modelos prevalentes, fueron diseñadas para satisfacer tanto las necesidades técnicas como legales de las organizaciones, facilitando la recopilación de evidencia y la identificación de las causas subyacentes de los incidentes de ciberseguridad. Esto conlleva beneficios implícitos adicionales como la reducción de la duplicidad de esfuerzos, la optimización de los recursos, una gestión eficiente de la información y la reducción del tiempo necesario para obtener datos relevantes.

De igual manera, comprendiendo que esta tesis abarca dos grandes disciplinas desde un enfoque totalmente nuevo, resulta evidente que aún mucho trabajo por hacer a fin de contar con entornos SDN más robustos en términos de ciberseguridad. Por lo que, a continuación se entregan algunas líneas de investigación que pueden ser exploradas en el apasionante mundo de DFIR en SDN.

6.2 Futuras líneas de investigación

La presente tesis doctoral contribuye con bases técnicas y filosóficas hacia el desarrollo de las redes definidas por software, más allá de constituir aportaciones desde el ámbito de ciberseguridad, pues al incorporar medidas de seguridad sólidas en estas redes, se promueve su reconocimiento y confiabilidad. Esta tesis marca el inicio para un tipo de gestión de incidentes de ciberseguridad con visión organizacional, esperando que nuevas investigaciones puedan

alinearse a este enfoque.

Las nuevas líneas de investigación se pueden enmarcar en el desarrollo de modelos que acompañen la arquitectura propuesta, la utilización de diferentes técnicas para el manejo de datos y el almacenamiento en ambientes distribuidos.

Las contribuciones presentadas en esta tesis tienen un enfoque de propósito general, lo que implica que pueden ser aplicadas en diversos entornos y situaciones. No obstante, para validar su efectividad en un contexto más amplio, sería necesario llevar a cabo pruebas en entornos más complejos y que utilicen diferentes tipos de controladores SDN. Además, sería beneficioso someter la solución a una variedad más amplia de ataques y topologías para evaluar su robustez y adaptabilidad en situaciones cotidianas.

En lo que respecta al proceso de detección de cambios en los elementos SDN, resultaría interesante explorar el uso de otras técnicas de análisis de texto para mejorar la eficiencia cuando existen grandes volúmenes de datos, especialmente durante la comparación de cadenas para comprobar la coherencia de las fuentes y corroboración de información. Respecto al proceso filtrado y pensando en la respuesta a incidentes, se podría incorporar RNN para mejorar los procedimientos de detección temprana, de esta manera se podría trabajar con secuencias de datos, recordando información de estados anteriores y aplicarla al procesamiento de datos futuros.

Considerando los resultados actuales del modelo de inteligencia artificial utilizado para el filtrado, adquisición y tratamiento de datos, sería prudente aumentar la cantidad de datos de entrenamiento y extender el período de experimentación. Al incrementar el número de instancias utilizadas para entrenar el modelo y prolongar el tiempo de experimentación, se podría mejorar la precisión y eficacia en la detección, permitiendo una mejor adaptación a una variedad más amplia de escenarios y condiciones.

Dado que la preparación forense se alinea con la optimización de recursos para la obtención de evidencia, en líneas futuras de investigación se podrían mejorar las implementaciones respecto a las redes *blockchain* permisionadas, realizando un análisis minucioso entre recursos computacionales y cantidad de nodos de ordenamiento. Para ello, sería necesario identificar los requisitos de procesamiento de datos y las capacidades de cómputo de cada nodo de la red *blockchain*, junto con simulaciones de carga para medir el rendimiento de cada nodo durante el procesamiento de transacciones.

6.3 Divulgación de resultados

Las contribuciones de esta tesis doctoral han enriquecido significativamente el estado del arte en el ámbito de ciberseguridad, de la preparación forense y de la gestión de incidentes de ciberseguridad en las SDN, manifestándose a través de publicaciones en artículos de revistas indexadas en el Journal Citation Reports (JCR) y en Scimago Journal Rank (SJR), y en congresos internacionales. A continuación, se presenta un desglose de las publicaciones derivadas del desarrollo de esta investigación doctoral.

6.3.1 Publicaciones

Revista

- Jiménez, M. B., Fernández, D., Rivadeneira, J. E., Bellido, L., and Cárdenas, A., "A Survey of the Main Security Issues and Solutions for the SDN Architecture", in IEEE Access, vol. 9, pp. 122016-122038, 2021, doi: 10.1109/ACCESS.2021.3109564

  Número de citas: 64

  Cuartil SJR: Q1

  Cuartil JCR: Q2

  Factor de Impacto: 3.476

- Jiménez, M. B., Fernández, D., Rivadeneira, J. E., and Flores-Moyano, R., "A Filtering Model for Evidence Gathering in an SDN-oriented Digital Forensic and Incident Response Context", in IEEE Access, vol. 12, pp. 75792-75808, 2024, doi:10.1109/ACCESS.2024.3405588

  Cuartil SJR: Q1

  Cuartil JCR: Q2

  Factor de Impacto: 3.476

Conferencia

- Jiménez, M. B. and Fernández, D., "A Framework for SDN Forensic Readiness and Cybersecurity Incident Response", 2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2022, pages 112-116, doi: 10.1109/NFV-SDN56302.2022.9974648

  Lugar y fecha de conferencia: Chandler, Arizona, Estados Unidos de Norteamérica, 14-16 Noviembre 2022

  Número de citas: 7

6.3.2 Colaboraciones

Contexto externo

- Rivadeneira, J. E., Jiménez, M. B., Marculescu, R., Rodrigues, A., Boavida, F., and Sá Silva, J., "A blockchain-based privacy preserving model for consent and transparency in human-centered internet of things", ACM/IEEE Conference on Internet of Things Design and Implementation, 2023, pages 301–314, doi.org/10.1145/3576842.3582379

  Lugar y fecha de conferencia: San Antonio, Texas, Estados Unidos de Norteamérica, 9-12 Mayo 2023

  Número de citas: 6

Referencias

ABDULAZEEZ, Adnan Mohsin; ZEEBAREE, Diyar; ZEBARI, Dilovan; ZEBARI, Rizgar R; ZEEBAREE, Diyar Qader; ZEBARI, Dilovan Asaad y SAEED, Jwan Najeeb, 2020. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Article in Journal of Applied Science and Technology Trends*. Vol. 01, pp. 56-70. ISSN 2708-0757. Available from DOI: 10.38094/jastt1224.

AGBORUBERE, Belema y SANCHEZ-VELAZQUEZ, Erika, 2018. OpenFlow communications and TLS security in software-defined networks. In: *Proceedings - 2017 IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*. Institute of Electrical y Electronics Engineers Inc. Vol. 2018-Janua, pp. 560-566. ISBN 9781538630655. Available from DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.88.

AHMAD, Ijaz; NAMAL, Suneth; YLIANTTILA, Mika y GURTOV, Andrei, 2015. *IEEE Communications Surveys and Tutorials*. Vol. 17, Security in Software Defined Networks: A Survey. Institute of Electrical y Electronics Engineers Inc. N.º 4. ISSN 1553877X. Available from DOI: 10.1109/COMST.2015.2474118.

AHMAD, Parvez; JACOB, Sven y KHONDOKER, Rahamatullah, 2018. Security Analysis of SDN Applications for Big Data. In: *Lecture Notes in Networks and Systems*. Springer. Vol. 30, pp. 39-55. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_3.

AKYILDIZ, Ian F.; LEE, Ahyoung; WANG, Pu; LUO, Min y CHOU, Wu, 2014. A roadmap for traffic engineering in software defined networks. *Computer Networks*. Vol. 71, pp. 1-30. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2014.06.002.

ALLYBOKUS, Zaid; AVRACHENKOV, Konstantin; LEGUAY, Jeremie y MAGGI, Lorenzo, 2018. Multi-path alpha-fair resource allocation at scale in distributed software-defined networks. *IEEE Journal on Selected Areas in Communications*. Vol. 36, pp. 2655-2666. ISSN 15580008. Available from DOI: 10.1109/JSAC.2018.2871293.

ALSAEEDI, Mohammed; MOHAMAD, Mohd Murtadha y AL-ROUBAIEY, Anas A., 2019. Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey. *IEEE Access*. Vol. 7, pp. 107346-107379. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2019.2932422.

ALSHRA'A, Abdullah Soliman y SEITZ, Jochen, 2019. Using INSPECTOR Device to Stop Packet Injection Attack in SDN. *IEEE Communications Letters*. Vol. 23, n.º 7, pp. 1174-1177. ISSN 15582558. Available from DOI: 10.1109/LCOMM.2019.2896928.

ARASHLOO, Mina Tahmasbi; KORAL, Yaron; GREENBERG, Michael; REXFORD, Jennifer y WALKER, David, 2016. SNAP: Stateful network-wide abstractions for packet processing. In: *SIGCOMM 2016 - Proceedings of the 2016 ACM Conference on Special Interest Group*

on Data Communication. Association for Computing Machinery, Inc, pp. 29-43. ISBN 9781450341936. Available from DOI: 10.1145/2934872.2934892.

ARBETTU, Ramachandra Kamath; KHONDOKER, Rahamatullah; BAYAROU, Kpatcha y WEBER, Frank, 2016. Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers. *2016 17th International Telecommunications Network Strategy and Planning Symposium, Networks 2016 - Conference Proceedings*, pp. 37-44. ISBN 9781467389914. Available from DOI: 10.1109/NETWKS.2016.7751150.

ARTMANN, David y KHONDOKER, Rahamatullah, 2018. Security Analysis of SDN WiFi Applications. In: *Lecture Notes in Networks and Systems*. Springer. Vol. 30, pp. 57-71. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_4.

ASSEFA, Beakal Gizachew y ÖZKASAP, Öznur, 2019. A survey of energy efficiency in SDN: Software-based methods and optimization models. *Journal of Network and Computer Applications*. Vol. 137, pp. 127-143. ISSN 10958592. Available from DOI: 10.1016/j.jnca.2019.04.001.

AUJLA, Gagangeet Singh; SINGH, Maninderpal; BOSE, Arnab; KUMAR, Neeraj; HAN, Guangjie y BUYYA, Rajkumar, 2020. BlockSDN: Blockchain as a Service for Software Defined Networking in Smart City Applications. *IEEE Network*. Vol. 34, n.º 2, pp. 83-91. ISSN 1558156X. Available from DOI: 10.1109/MNET.001.1900151.

AZZOUNI, Abdelhadi; BOUTABA, Raouf; TRANG, Nguyen Thi Mai; PUJOLLE, Guy; DENG, Shuhua; GAO, Xing Xieping; LU, Zebin y GAO, Xing Xieping, 2017. sOFTDP: Secure and Efficient Topology Discovery Protocol for SDN. *IEEE Transactions on Information Forensics and Security*. Vol. 13, n.º 3, pp. 695-705. ISSN 15566013. Available from DOI: 10.1109/TIFS.2017.2765506.

BADOTRA, Sumit y PANDA, Surya Narayan, 2021. SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking. *Cluster Computing*. Vol. 24, n.º 1, pp. 501-513. ISBN 1058602003133. ISSN 15737543. Available from DOI: 10.1007/s10586-020-03133-y.

BANNOUR, Fetia; SOUIHI, Sami y MELLOUK, Abdelhamid, 2018. Distributed SDN Control: Survey, Taxonomy, and Challenges. *IEEE Communications Surveys and Tutorials*. Vol. 20, pp. 333-354. ISSN 1553877X. Available from DOI: 10.1109/COMST.2017.2782482.

BANSE, Christian y RANGARAJAN, Sathyanarayanan, 2015. A secure northbound interface for SDN applications. In: Institute of Electrical y Electronics Engineers Inc. Vol. 1, pp. 834-839. ISBN 9781467379519. Available from DOI: 10.1109/Trustcom.2015.454.

BARKI, Lohit; SHIDLING, Amrit; METI, Nisharani; NARAYAN, D. G. y MULLA, Mohammed Moin, 2016. Detection of distributed denial of service attacks in software defined networks. In: *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*. Institute of Electrical y Electronics Engineers Inc., pp. 2576-2581. ISBN 9781509020287. Available from DOI: 10.1109/ICACCI.2016.7732445.

BENAMRANE, Fouad; MAMOUN, Mouad Ben y BENAINI, Redouane, 2017. New method for controller-to-controller communication in distributed SDN architecture. *International Journal of Communication Networks and Distributed Systems*. Vol. 19, pp. 357-367. ISSN 17543924. Available from DOI: 10.1504/IJCNDS.2017.086493.

BENTON, Kevin; CAMP, L Jean y SMALL, Chris, 2013. OpenFlow Vulnerability Assessment. In: *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined*

networking - HotSDN '13. New York, New York, USA: ACM Press. ISBN 9781450321785. Available also from: http://www.wireshark.org/security/.

BENZEKKI, Kamal; FERGOUGUI, Abdeslam El y ALAOUI, Abdelbaki El Belrhiti El, 2016. Devolving IEEE 802.1X authentication capability to data plane in software-defined networking (SDN) architecture. *Security and Communication Networks*. Vol. 9, pp. 4369-4377. ISSN 19390114. Available from DOI: 10.1002/sec.1613.

BESHLEY, Mykola; KLYMASH, Mikhailo; SCHERM, Ilona; BESHLEY, Halyna y SHKOROPAD, Yuriy, 2022. Emerging network technologies for digital transformation: 5G/6G, IoT, SDN/IBN, cloud computing, and blockchain. In: Springer, pp. 1-20.

BHANDARI, Sandeepak y JUSAS, Vacius, 2020. An abstraction based approach for reconstruction of timeline in digital forensics. *Symmetry*. Vol. 12, p. 104. ISSN 2073-8994.

BIANCHI, Giuseppe; BONOLA, Marco; CAPONE, Antonio y CASCONE, Carmelo, 2014. *Computer Communication Review*. Vol. 44, Openstate: Programming platform-independent stateful openflow applications inside the switch. Association for Computing Machinery. N.º 2. ISSN 19435819. Available from DOI: 10.1145/2602204.2602211.

BIANCHI, Giuseppe; BONOLA, Marco; PONTARELLI, Salvatore; SANVITO, Davide; CAPONE, Antonio y CASCONE, Carmelo, 2016. Open Packet Processor: a programmable architecture for wire speed platform-independent stateful in-network processing. Available from arXiv: 1605.01977.

BIDAJ, Andi; AURA, Tuomas y RØSTAD, Lillian, 2016. Security Testing SDN Controllers Thesis supervisors.

BINNAR, Pranita; BHIRUD, Sunil y KAZI, Faruk, 2023. Security Analysis of Industrial Iot Using Digital Forensics Incident Response. *Available at SSRN 4586376*. Available from DOI: https://doi.org/10.1016/j.csa.2023.100034.

BJÖRK, Magnus, 2011. Successful SAT Encoding Techniques. *Journal on Satisfiability, Boolean Modeling and Computation*. Vol. 7, pp. 189-201. Available from DOI: 10.3233/SAT190085.

BONOLA, Marco; BIFULCO, Roberto; PETRUCCI, Luca; PONTARELLI, Salvatore; TULUMELLO, Angelo y BIANCHI, Giuseppe, 2017. Implementing advanced network functions for datacenters with stateful programmable data planes. In: *IEEE Workshop on Local and Metropolitan Area Networks*. IEEE Computer Society. Vol. 2017-June. ISBN 9781538607282. ISSN 19440375. Available from DOI: 10.1109/LANMAN.2017.7972130.

BOSSHART, Pat; DALY, Dan; GIBB, Glen; IZZARD, Martin; MCKEOWN, Nick; REXFORD, Jennifer; SCHLESINGER, Cole; TALAYCO, Dan; VAHDAT, Amin; VARGHESE, George y WALKER, David, 2014. P4: Programming protocol-independent packet processors. *Computer Communication Review*. Vol. 44, n.º 3, pp. 87-95. ISSN 19435819. Available from DOI: 10.1145/2656877.2656890.

BRÄUNING, Marco y KHONDOKER, Rahamatullah, 2018. Analysis of SDN Applications for Smart Grid Infrastructures. In: *Lecture Notes in Networks and Systems*. Springer. Vol. 30, pp. 99-110. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_7.

BUSINESSWIRE, 2020. *Global Software-Defined Networking Market (2020 to 2025) - Software-Defined Networking for 5G Presents Opportunities* [online]. [visited on 2023-03-26]. Available from: https://www.businesswire.com/news/home/20200817005303/en/Global-Software-Defined-Networking-Market-2020-to-2025---Software-Defined-Networking-for-5G-Presents-Opportunities.

CAMPESATO, Oswald, 2020. *Artificial intelligence, machine learning, and deep learning*. Mercury Learning e Information.

CANADIAN INSTITUTE FOR CYBERSECURITY, 1999. *NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB*. Available also from: https://www.unb.ca/cic/datasets/nsl.html.

CAPONE, Antonio; CASCONE, Carmelo; NGUYEN, Alessandro Q.T. y SANSÒ, Brunilde, 2015. Detour planning for fast and reliable failure recovery in SDN with OpenState. *2015 11th International Conference on the Design of Reliable Communication Networks, DRCN 2015*, pp. 25-32. ISBN 9781479977956. Available from DOI: 10.1109/DRCN.2015.7148981.

CAPROLU, Maurantonio; RAPONI, Simone y PIETRO, Roberto Di, 2019. FORTRESS: An Efficient and Distributed Firewall for Stateful Data Plane SDN. *Security and Communication Networks*. Vol. 2019, p. 6874592. ISSN 1939-0114. Available from DOI: 10.1155/2019/6874592.

CARRIER, Brian y SPAFFORD, Eugene H, 2003. Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence Fall*. Vol. 2. Available also from: www.ijde.org.

CASCONE, Carmelo; POLLINI, Luca; SANVITO, Davide y CAPONE, Antonio, 2015. Traffic Management Applications for Stateful SDN Data Plane. In: *Proceedings - European Workshop on Software Defined Networks, EWSDN*. Institute of Electrical y Electronics Engineers Inc., pp. 85-90. ISBN 9781509001804. Available from DOI: 10.1109/EWSDN.2015.66.

CASEY, Eoghan, 2002. Error, uncertainty, and loss in digital evidence. *International journal of digital evidence*. Vol. 1, pp. 1-45.

CHAE, Younghun; KATENKA, Natallia y DIPIPPO, Lisa, 2019. An Adaptive Threshold Method for Anomaly-based Intrusion Detection Systems. *2019 IEEE 18th International Symposium on Network Computing and Applications, NCA 2019*. ISBN 9781728125220. Available from DOI: 10.1109/NCA.2019.8935045.

CHANG, Rui; LIN, Zhaowen; SUN, Yi y XU, Jie, 2019. MD-UCON: A Multi-Domain Access Control Model for SDN Northbound Interfaces. In: *Journal of Physics: Conference Series*. IOP Publishing. Vol. 1187, p. 32091. N.º 3. ISSN 17426596. Available from DOI: 10.1088/1742-6596/1187/3/032091.

CHAVEZ, Adrian R.; STOUT, William M.S. y PEISERT, Sean, 2016. Techniques for the dynamic randomization of network attributes. In: Institute of Electrical y Electronics Engineers Inc. Vol. 2015-Janua. ISBN 9781479986910. ISSN 10716572. Available from DOI: 10.1109/CCST.2015.7389661.

CHECK POINT, 2022. *Check Point Software's Mid-Year Security Report*. Available also from: https://blog.checkpoint.com/2022/01/10/check-point-research-cyber-attacks-increased-50-year-over-year/.

CHIKHALE, Ankush y KHONDOKER, Rahamatullah, 2018. Security Analysis of SDN Cloud Applications. In: *Lecture Notes in Networks and Systems*. Springer. Vol. 30, pp. 19-38. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_2.

CISCO, 2019. *Software-Defined Networking (SDN) Definition - Cisco*. Available also from: https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html.

CISCO, 2014. OpFlex: An Open Source Approach, pp. 1-3. Available also from: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731304.html.

CISCO, 2020. Trends Report 2020 Global 2020 Global Networking Trends Report The evolving role of the IT network 7.

CONGDON, Paul; NETWORKING, Procurve; BLATHERWICK, Peter y NETWORKS, Mitel, 2004. 802.1AB Overview Link Layer Discovery Protocol, p. 17. Available also from: http://www.ieee802.org/3/frame_study/0409/blatherwick_1_0409.pdf.

CONRAD, Eric; MISENAR, Seth y FELDMAN, Joshua, 2012. Domain 9: Legal,Regulations, Investigations,and Compliance. *CISSP Study Guide*, pp. 389-427. Available from DOI: 10.1016/B978-1-59749-961-3.00010-8.

CUI, Hongyan; CHEN, Zunming; YU, Longfei; XIE, Kun y XIA, Zongguo, 2018. Authentication mechanism for network applications in SDN environments. In: IEEE Computer Society. Vol. 2017-Decem, pp. 1-5. ISBN 9781538627679. ISSN 13476890. Available from DOI: 10.1109/WPMC.2017.8301788.

DARGAHI, Tooska; CAPONI, Alberto; AMBROSIN, Moreno; BIANCHI, Giuseppe y CONTI, Mauro, 2017. A Survey on the Security of Stateful SDN Data Planes. *IEEE Communications Surveys and Tutorials*. Vol. 19, n.º 3, pp. 1701-1725. ISSN 1553877X. Available from DOI: 10.1109/COMST.2017.2689819.

DEHKORDI, Afsaneh Banitalebi; SOLTANAGHAEI, Mohammad Reza y BOROUJENI, Farsad Zamani, 2021. The DDoS attacks detection through machine learning and statistical methods in SDN. *Journal of Supercomputing*. Vol. 77, pp. 2383-2415. ISSN 15730484. Available from DOI: 10.1007/s11227-020-03323-w.

DENG, Shuhua; GAO, Xing; LU, Zebin y GAO, Xieping, 2018. Packet injection attack and its defense in software-defined networks. *IEEE Transactions on Information Forensics and Security*. Vol. 13, n.º 3, pp. 695-705. ISSN 15566013. Available from DOI: 10.1109/TIFS.2017.2765506.

DEWI, Christine y CHEN, Rung-Ching, 2019. Random forest and support vector machine on features selection for regression analysis. *Int. J. Innov. Comput. Inf. Control*. Vol. 15, pp. 2027-2037.

DHAWAN, Mohan; PODDAR, Rishabh; MAHAJAN, Kshiteej y MANN, Vijay, 2015. SPHINX: Detecting Security Attacks in Software-Defined Networks. In: vol. 15, pp. 8-11.

DIXIT, Vaibhav Hemant y DOUPÉ, Adam, 2018. AIM-SDN : Attacking Information Mismanagement in SDN-datastores. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security - CCS '18*, pp. 664-676. ISBN 9781450356930.

DU, Sang Gyun; LEE, Jong Won y KIM, Keecheon, 2018. Proposal of GRPC as a new northbound API for application layer communication efficiency in SDN. In: pp. 1-6.

DUNSIN, Dipo; GHANEM, Mohamed C; OUAZZANE, Karim y VASSILEV, Vassil, 2023. A Comprehensive Analysis of the Role of Artificial Intelligence and Machine Learning in Modern Digital Forensics and Incident Response. *arXiv preprint arXiv:2309.07064*.

DURAIRAJAN, Ramakrishnan; SOMMERS, Joel y BARFORD, Paul, 2014. Controller-Agnostic SDN debugging. In: ACM, pp. 227-233. ISBN 9781450332798. Available from DOI: 10.1145/2674005.2674993.

DUY, Phan The; HOANG, Hien Do; HIEN, Do Thi Thu; KHANH, Nguyen Ba y PHAM, Van Hau, 2019. SDNLog-Foren: Ensuring the integrity and tamper resistance of log files for SDN forensics using blockchain. In: Institute of Electrical y Electronics Engineers Inc., pp. 416-421. ISBN 9781728151632. Available from DOI: 10.1109/NICS48868.2019.9023852.

ELSAYED, Mahmoud Said; LE-KHAC, Nhien An y JURCUT, Anca D., 2020. InSDN: A novel SDN intrusion dataset. *IEEE Access*. Vol. 8, pp. 165263-165284. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2020.3022633.

ELYAS, Mohamed; AHMAD, Atif; MAYNARD, Sean B. y LONIE, Andrew, 2015. Digital forensic readiness: Expert perspectives on a theoretical framework. *Computers and Security*. Vol. 52, pp. 70-89. ISSN 01674048. Available from DOI: 10.1016/J.COSE.2015.04.003.

ENISA, 2017. Electronic evidence -a basic guide for First Responders. ISBN 978-92-9204-111-3. Available from DOI: 10.2824/068545.

EUROPEAN PARLIAMENT AND EUROPEAN UNION COUNCIL, 2018. *European Electronic Communications Code EUR-Lex - 32018L1972 - EN - EUR-Lex*. Available also from: https://eur-lex.europa.eu/eli/dir/2018/1972/oj/spa.

EUROPEAN PARLIAMENT AND EUROPEAN UNION COUNCIL, 2019. *EUR-Lex - 32019R0881 - EN - EUR-Lex*. Available also from: https://eur-lex.europa.eu/legal-content/ES/ALL/?uri=CELEX%3A32019R0881.

EUROPEAN PARLIAMENT AND EUROPEAN UNION COUNCIL, 2022. *EUR-Lex - 32022L2555 - EN - EUR-Lex*. Available also from: https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32022L2555.

FEAMSTER, Nick; REXFORD, Jennifer y ZEGURA, Ellen, 2014. The road to SDN. *ACM SIGCOMM Computer Communication Review*. Vol. 44, pp. 87-98. ISSN 0146-4833. Available from DOI: 10.1145/2602204.2602219.

FERGUSON, Andrew D.; GUHA, Arjun; LIANG, Chen; FONSECA, Rodrigo y KRISHNAMURTHI, Shriram, 2013. Participatory networking: An API for application control of SDNs. In: vol. 43, pp. 327-338. ISBN 9781450320566. ISSN 01464833. Available from DOI: 10.1145/2534169.2486003.

FORCES, 2014. *RFC 7391 - Forwarding and Control Element Separation (ForCES) Protocol Extensions*. Available also from: https://tools.ietf.org/html/rfc7391.

FOULADI, Ramin Fadaei; ERMIŞ, Orhan y ANARIM, Emin, 2022. A DDoS attack detection and countermeasure scheme based on DWT and auto-encoder neural network for SDN. *Computer Networks*. Vol. 214, p. 109140. ISSN 1389-1286. Available from DOI: 10.1016/J.COMNET.2022.109140.

GADALLAH, Waheed G.; IBRAHIM, Hosny M. y OMAR, Nagwa M., 2024. A deep learning technique to detect distributed denial of service attacks in software-defined networks. *Computers Security*. Vol. 137, p. 103588. ISSN 0167-4048. Available from DOI: 10.1016/J.COSE.2023.103588.

GARG, Sahil; KUMAR, Neeraj; RODRIGUES, Joel J.P.C. y RODRIGUES, Joel J.P.C., 2019. Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective. *IEEE Transactions on Multimedia*. Vol. 21, pp. 566-578. ISSN 15209210. Available from DOI: 10.1109/TMM.2019.2893549.

GITHUB, 2020. *GitHub - noxrepo/pox: The POX network software platform* [online]. [visited on 2021-02-09]. Available from: https://github.com/noxrepo/pox.

GRÖMPING, Ulrike, 2009. Variable importance assessment in regression: linear regression versus random forest. *The American Statistician*. Vol. 63, pp. 308-319. ISSN 0003-1305.

GUARINO, Alessandro, 2013. Digital Forensics as a Big Data Challenge. *ISSE 2013 Securing Electronic Business Processes*, pp. 197-203. Available from DOI: 10.1007/978-3-658-03371-2_17.

HANMER, Robert; JAGADEESAN, Lalita; MENDIRATTA, Veena y ZHANG, Heng, 2018. Friend or Foe: Strong Consistency vs. Overload in High-Availability Distributed Systems and SDN. In: Institute of Electrical y Electronics Engineers Inc., pp. 59-64. ISBN 9781538694435. Available from DOI: 10.1109/ISSREW.2018.00-30.

HAUSER, Frederik; SCHMIDT, Mark; HABERLE, Marco y MENTH, Michael, 2020. P4-MACsec: Dynamic Topology Monitoring and Data Layer Protection with MACsec in P4-Based SDN. *IEEE Access*. Vol. 8, pp. 58845-58858. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2020.2982859.

HOANG, Hien Do; DUY, Phan The y PHAM, Van Hau, 2019. A security-enhanced monitoring system for northbound interface in SDN using blockchain. In: Association for Computing Machinery, pp. 197-204. ISBN 9781450372459. Available from DOI: 10.1145/3368926.3369709.

HONG, Sungmin; XU, Lei; WANG, Haopei y GU, Guofei, 2015. Poisoning network visibility in software-defined networks: New attacks and countermeasures. In: *NDSS*. Vol. 15, pp. 8-11.

HU, Tao; YI, Peng; HU, Yuxiang; LAN, Julong; ZHANG, Zhen y LI, Ziyong, 2020. SAIDE: Efficient application interference detection and elimination in SDN. *Computer Networks*. Vol. 183, p. 107619. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2020.107619.

HU, Tao; ZHANG, Zhen; YI, Peng; LIANG, Dong; LI, Ziyong; REN, Quan; HU, Yuxiang y LAN, Julong, 2021. SEAPP: A secure application management framework based on REST API access control in SDN-enabled cloud environment. *Journal of Parallel and Distributed Computing*. Vol. 147, pp. 108-123. ISSN 07437315. Available from DOI: 10.1016/j.jpdc.2020.09.006.

HUANG, Lei; QIN, Jie; ZHOU, Yi; ZHU, Fan; LIU, Li y SHAO, Ling, 2023. Normalization Techniques in Training DNNs: Methodology, Analysis and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 45, pp. 10173-10196. ISSN 19393539. Available from DOI: 10.1109/TPAMI.2023.3250241.

HUANG, Xueli; DU, Xiaojiang y SONG, Bin, 2017. An effective DDoS defense scheme for SDN. In: *IEEE International Conference on Communications*. Institute of Electrical y Electronics Engineers Inc. ISBN 9781467389990. ISSN 15503607. Available from DOI: 10.1109/ICC.2017.7997187.

HUNT, Patrick; KONAR, Mahadev; GRID, Yahoo !; JUNQUEIRA, Flavio P; REED, Benjamin y RESEARCH, Yahoo !, 2010. *ZooKeeper: Wait-free coordination for Internet-scale systems*. Available also from: https://www.usenix.org/.

HWANG, Ren Hung; NGUYEN, Van Linh y LIN, Po Ching, 2019. StateFit: A security framework for SDN programmable data plane model. *Proceedings - 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks, I-SPAN 2018*. Vol. 1, pp. 168-173. ISBN 9781538685341. Available from DOI: 10.1109/I-SPAN.2018.00035.

IETF, 2011. *RFC 6241 - Network Configuration Protocol (NETCONF)*. Available also from: https://tools.ietf.org/html/rfc6241.

IETF, 2013. *RFC 7047-The Open vSwitch Database Management Protocol*. Available also from: https://tools.ietf.org/html/rfc7047.

ILHA, Alexandre da Silveira; LAPOLLI, Angelo Cardoso; MARQUES, Jonatas Adilson y GASPARY, Luciano Paschoal, 2020. Euclid: A Fully In-Network, P4-based Approach for Real-Time DDoS Attack Detection and Mitigation. *IEEE Transactions on Network and Service Management*. Vol. 4537, n.º c, pp. 1-19. ISSN 19324537. Available from DOI: 10.1109/TNSM.2020.3048265.

IMRAN, Muhammad; DURAD, Muhammad Hanif; KHAN, Farrukh Aslam y ABBAS, Haider, 2020. DAISY: A Detection and Mitigation System against Denial-of-Service Attacks in Software-Defined Networks. *IEEE Systems Journal*. Vol. 14, n.º 2, pp. 1933-1944. ISSN 19379234. Available from DOI: 10.1109/JSYST.2019.2927223.

INTERPOL, 2022. *Digital forensics*. Available also from: https://www.interpol.int/How-we-work/Innovation/Digital-forensics.

ISO, 2012. *ISO/IEC 27037:2012 - Information technology — Security techniques — Guidelines for identification, collection, acquisition and preservation of digital evidence*. Available also from: https://www.iso.org/standard/44381.html.

ISO, 2015a. *ISO - ISO/IEC 27042:2015 - Information technology — Security techniques — Guidelines for the analysis and interpretation of digital evidence*. Available also from: https://www.iso.org/standard/44406.html.

ISO, 2015b. *ISO/IEC 27041:2015 - Information technology — Security techniques — Guidance on assuring suitability and adequacy of incident investigative method*. Available also from: https://www.iso.org/standard/44405.html.

ISO, 2015c. *ISO/IEC 27043:2015 - Information technology — Security techniques — Incident investigation principles and processes*. Available also from: https://www.iso.org/standard/44407.html.

ITODO, Cornelius; VARLIOGLU, Said y ELSAYED, Nelly, 2021. Digital forensics and incident response (DFIR) challenges in IoT platforms. *Proceedings - 2021 4th International Conference on Information and Computer Technologies, ICICT 2021*, pp. 199-203. ISBN 9781665413992. Available from DOI: 10.1109/ICICT52872.2021.00040.

JAIN, Rajat y KHONDOKER, Rahamatullah, 2018. Security Analysis of SDN WAN Applications—B4 and IWAN. In: *Lecture Notes in Networks and Systems*. Springer. Vol. 30, pp. 111-127. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_8.

JAKHAR, Deepack y KAUR, Ishmeet, 2020. Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and experimental dermatology*. Vol. 45, pp. 131-132. ISSN 1365-2230.

JARRETT, Aaron y CHOO, Kim-Kwang Raymond, 2021. The impact of automation and artificial intelligence on digital forensics. *Wiley Interdisciplinary Reviews: Forensic Science*. Vol. 3, e1418. ISSN 2573-9468.

JAVED, Abdul Rehman; AHMED, Waqas; ALAZAB, Mamoun; JALIL, Zunera; KIFAYAT, Kashif y GADEKALLU, Thippa Reddy, 2022. A Comprehensive Survey on Computer Forensics: State-of-the-Art, Tools, Techniques, Challenges, and Future Directions. *IEEE Access*. Vol. 10, pp. 11065-11089. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2022.3142508.

JOHANSEN, Gerard, 2017. *Digital forensics and incident response*. Packt Publishing Ltd. ISBN 1787285391.

KARAKUS, Murat y DURRESI, Arjan, 2017a. *A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)*. Vol. 112. Elsevier B.V. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2016.11.017.

KARAKUS, Murat y DURRESI, Arjan, 2017b. *Journal of Network and Computer Applications*. Vol. 80, Quality of Service (QoS) in Software Defined Networking (SDN): A survey. Academic Press. ISSN 10958592. Available from DOI: 10.1016/j.jnca.2016.12.019.

KARIE, Nickson M. y VALLI, Craig, 2021. Digital Forensic Readiness Implementation in SDN: Issues and Challenges. Available also from: https://arxiv.org/abs/2107.13759v1.

KATT, Basel y PRASHER, Nishu, 2018. Quantitative security assurance metrics: REST API case studies. *Proceedings of the 12th European Conference on Software Architecture: Companion Proceedings*, pp. 1-7.

KATTA, Naga; HIRA, Mukesh; KIM, Changhoon; SIVARAMAN, Anirudh y REXFORD, Jennifer, 2016. HULA: Scalable load balancing using programmable data planes. In: *Symposium on Software Defined Networking (SDN) Research, SOSR 2016*. New York, New York, USA: Association for Computing Machinery, Inc, pp. 1-12. ISBN 9781450334518. Available from DOI: 10.1145/2890955.2890968.

KATTA, Naga; ZHANG, Haoyu; FREEDMAN, Michael y REXFORD, Jennifer, 2015. Ravana: Controller fault-tolerance in software-defined networking. In: Association for Computing Machinery, Inc, pp. 1-12. ISBN 9781450334518. Available from DOI: 10.1145/2774993.2774996.

KEELE, Staffs, 2007. *Guidelines for performing systematic literature reviews in software engineering*. Technical report, ver. 2.3 ebse technical report. ebse.

KENT, K; CHEVALIER, S; GRANCE, T y DANG, H, 2006. Guide to integrating forensic techniques into incident response. Available from DOI: 10.6028/NIST.SP.800-86.

KHAN, Suleman; GANI, Abdullah; ABDUL WAHAB, Ainuddin Wahid; GUIZANI, Mohsen y KHAN, Muhammad Khurram, 2017. Topology Discovery in Software Defined Networks: Threats, Taxonomy, and State-of-the-Art. *IEEE Communications Surveys and Tutorials*. Vol. 19, n.º 1, pp. 303-324. ISSN 1553877X. Available from DOI: 10.1109/COMST.2016.2597193.

KHAN, Suleman; GANI, Abdullah; WAHAB, Ainuddin Wahid Abdul; ABDELAZIZ, Ahmed; KO, Kwangman; KHAN, Muhammad Khurram y GUIZANI, Mohsen, 2016. Software-defined network forensics: Motivation, potential locations, requirements, and challenges. *IEEE Network*. Vol. 30, pp. 6-13. ISSN 08908044. Available from DOI: 10.1109/MNET.2016.1600051NM.

KHAN, Yunus y VERMA, Sunita, 2021. An Intelligent Blockchain and Software-Defined Networking-Based Evidence Collection Architecture for Cloud Environment. *Scientific Programming*. Vol. 2021. ISSN 10589244. Available from DOI: 10.1155/2021/7294206.

KHORSANDROO, Sajad y TOSUN, Ali Saman, 2018. Time Inference Attacks on Software Defined Networks: Challenges and Countermeasures. In: *IEEE International Conference on Cloud Computing, CLOUD*. IEEE Computer Society. Vol. 2018-July, pp. 342-349. ISBN 9781538672358. ISSN 21596190. Available from DOI: 10.1109/CLOUD.2018.00050.

KHRAISAT, Ansam; GONDAL, Iqbal; VAMPLEW, Peter y KAMRUZZAMAN, Joarder, 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. Vol. 2, n.º 1, pp. 1-22. ISSN 25233246. Available from DOI: 10.1186/s42400-019-0038-7.

KIM, Green; AN, Junghyun y KIM, Keecheon, 2017. A study on authentication mechanism in SEaaS for SDN. In: Association for Computing Machinery, Inc, pp. 1-6. ISBN 9781450348881. Available from DOI: 10.1145/3022227.3022277.

KOTSIANTIS, S B; KANELLOPOULOS, D y PINTELAS, P E, 2007. Data preprocessing for supervised leaning. World Academy of Science, Engineering and Technology. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*. Vol. 1.

KREUTZ, Diego; RAMOS, Fernando M.V.; VERISSIMO, Paulo Esteves; ROTHENBERG, Christian Esteve; AZODOLMOLKY, Siamak y UHLIG, Steve, 2015. Software-defined

networking: A comprehensive survey. *Proceedings of the IEEE*. Vol. 103, pp. 14-76. ISSN 15582256. Available from DOI: 10.1109/JPROC.2014.2371999.

KRONGBARAMEE, Pakapol y SOMCHIT, Yuthapong, 2018. Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch. In: Institute of Electrical y Electronics Engineers Inc. ISBN 9781538655382. Available from DOI: 10.1109/JCSSE.2018.8457354.

KUMAR, Prashant; TRIPATHI, Meenakshi; NEHRA, Ajay; CONTI, Mauro y LAL, Chhagan, 2018. SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN. *IEEE Transactions on Network and Service Management*. Vol. 15, n.º 4, pp. 1545-1559. ISSN 19324537. Available from DOI: 10.1109/TNSM.2018.2861741.

LAGRASSE, Maxime; SINGH, Avinash; MUNKHONDYA, Howard; IKUESAN, Adeyemi y VENTER, Hein, 2020. Digital forensic readiness framework for software-defined networks using a trigger-based collection mechanism. *Proceedings of the 15th International Conference on Cyber Warfare and Security, ICCWS 2020*, pp. 296-305. ISBN 9781912764525. Available from DOI: 10.34190/ICCWS.20.045.

LAM, Jun Huy; LEE, Sang Gon; LEE, Hoon Jae y OKTIAN, Yustus Eko, 2015. Securing distributed SDN with IBC. In: IEEE Computer Society. Vol. 2015-Augus, pp. 921-925. ISBN 9781479989935. ISSN 21658536. Available from DOI: 10.1109/ICUFN.2015.7182680.

LAM, Junhuy; LEE, Sang Gon; LEE, Hoon Jae y OKTIAN, Yustus Eko, 2016. Securing SDN Southbound and Data Plane Communication with IBC. *Mobile Information Systems*. Vol. 2016. ISSN 1875905X. Available from DOI: 10.1155/2016/1708970.

LAMPORT, Leslie, 1983. The weak Byzantine generals problem. *Journal of the ACM (JACM)*. Vol. 30, pp. 668-676. ISSN 0004-5411.

LATAH, Majd y TOKER, Levent, 2020. Load and stress testing for SDN's northbound API. *SN Applied Sciences*. Vol. 2, n.º 1Latah, M., Toker, L. (2020). Load and stress testing for SDN's northbound API. SN Applied Sciences, 2(1), 1–8. https://doi.org/10.1007/s42452-019-1917-y, pp. 1-8. ISBN 0123456789. ISSN 2523-3963. Available from DOI: 10.1007/s42452-019-1917-y.

LATIF, Zohaib; SHARIF, Kashif; LI, Fan; KARIM, Md Monjurul y WANG, Yu, 2019. A comprehensive survey of interface protocols for software defined networks. *arXiv*, pp. 1-30.

LEE, Chanhee y SHIN, Seungwon, 2016. SHIELD: An automated framework for static analysis of SDN applications. In: Association for Computing Machinery, Inc, pp. 29-34. ISBN 9781450340786. Available from DOI: 10.1145/2876019.2876026.

LEE, Chanhee; YOON, Changhoon; SHIN, Seungwon y CHA, Sang Kil, 2018. INDAGO: A New Framework For Detecting Malicious SDN Applications. In: IEEE Computer Society. Vol. 2018-Septe, pp. 220-230. ISBN 9781538660430. ISSN 10921648. Available from DOI: 10.1109/ICNP.2018.00031.

LEEVY, Joffrey L y KHOSHGOFTAAR, Taghi M, 2018. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data. Available from DOI: 10.1186/s40537-020-00382-x.

LEROUX, Olivier, 2004. Legal admissibility of electronic evidence1. *International Review of Law, Computers amp; Technology*. Vol. 18, pp. 193-220. ISSN 1360-0869. Available from DOI: 10.1080/1360086042000223508.

LEWIS, Benjamin; BROADBENT, Matthew y RACE, Nicholas, 2019. P4ID: P4 Enhanced Intrusion Detection. In: *IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2019 - Proceedings*. Institute of Electrical y Electronics

Engineers Inc. ISBN 9781728145457. Available from DOI: 10.1109/NFV-SDN47374.2019.9040044.

LI, Chuanhuang; WU, Yan; YUAN, Xiaoyong; SUN, Zhengjun; WANG, Weiming; LI, Xiaolin y GONG, Liang, 2018. Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *International Journal of Communication Systems*. Vol. 31, n.º 5. ISSN 10991131. Available from DOI: 10.1002/dac.3497.

LI, Yahui; WANG, Zhiliang; YAO, Jiangyuan; YIN, Xia; SHI, Xingang; WU, Jianping y ZHANG, Han, 2019. MSAID: Automated detection of interference in multiple SDN applications. *Computer Networks*. Vol. 153, pp. 49-62. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2019.01.042.

LIN, Pingping; BI, Jun; CHEN, Ze; WANG, Yangyang; HU, Hongyu y XU, Anmin, 2014. WE-bridge: West-east bridge for SDN inter-domain network peering. In: Institute of Electrical y Electronics Engineers Inc., pp. 111-112. ISBN 9781479930883. ISSN 0743166X. Available from DOI: 10.1109/INFCOMW.2014.6849180.

LIN, Ting Yu; WU, Jhen Ping; HUNG, Pei Hsuan; SHAO, Ching Hsuan; WANG, Yu Ting; CAI, Yun Zhan y TSAI, Meng Hsun, 2020. Mitigating SYN flooding attack and ARP spoofing in SDN data plane. In: *APNOMS 2020 - 2020 21st Asia-Pacific Network Operations and Management Symposium: Towards Service and Networking Intelligence for Humanity*. Institute of Electrical y Electronics Engineers Inc., pp. 114-119. ISBN 9788995004388. Available from DOI: 10.23919/APNOMS50412.2020.9236951.

LIN, Xiaodong, 2018. Timeline Analysis. In: ed. por LIN, Xiaodong. Springer International Publishing, pp. 257-269. ISBN 978-3-030-00581-8. Available from DOI: 10.1007/978-3-030-00581-8_12.

LUTTGENS, Jason T. y PEPE, Mathew., 2014. Incident response computer forensics. ISBN 0071798684.

MACEDO, Ricardo; CASTRO, Rafael De; SANTOS, Aldri; GHAMRI-DOUDANE, Yacine y NOGUEIRA, Michele, 2016. Self-organized SDN controller cluster conformations against DDoS attacks effects. In: Institute of Electrical y Electronics Engineers Inc. ISBN 9781509013289. Available from DOI: 10.1109/GLOCOM.2016.7842259.

MAEDA, Shogo; KANAI, Atsushi; TANIMOTO, Shigeaki; HATASHIMA, Takashi y OHKUBO, Kazuhiko, 2019. A Botnet Detection Method on SDN using Deep Learning. In: *2019 IEEE International Conference on Consumer Electronics, ICCE 2019*. Institute of Electrical y Electronics Engineers Inc. ISBN 9781538679104. Available from DOI: 10.1109/ICCE.2019.8662080.

MAKUVAZA, Auther; JAT, Dharm Singh y GAMUNDANI, Attlee M., 2021. Deep Neural Network (DNN) Solution for Real-time Detection of Distributed Denial of Service (DDoS) Attacks in Software Defined Networks (SDNs). *SN Computer Science*. Vol. 2, pp. 1-10. ISSN 26618907. Available from DOI: 10.1007/S42979-021-00467-1/FIGURES/7.

MALIK, Jahanzaib; AKHUNZADA, Adnan; BIBI, Iram; IMRAN, Muhammad; MUSADDIQ, Arslan y KIM, Sung Won, 2020. Hybrid Deep Learning: An Efficient Reconnaissance and Surveillance Detection Mechanism in SDN. *IEEE Access*. Vol. 8, pp. 134695-134706. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2020.3009849.

MARIN, Eduard; BUCCIOL, Nicola y CONTI, Mauro, 2019. An in-depth look into SDN topology discovery mechanisms: Novel attacks and practical countermeasures. In: *Proceedings of the ACM Conference on Computer and Communications Security*. New York, NY,

USA: ACM, pp. 1101-1114. ISBN 9781450367479. ISSN 15437221. Available from DOI: 10.1145/3319535.3354194.

MATIAS, Jon; GARAY, Jokin; MENDIOLA, Alaitz; TOLEDO, Nerea y JACOB, Eduardo, 2014. FlowNAC: Flow-based network access control. In: Institute of Electrical y Electronics Engineers Inc., pp. 79-84. ISBN 9781479969197. Available from DOI: 10.1109/EWSDN.2014.39.

MATSUMOTO, Stephanos; HITZ, Samuel; ZURICH, Eth y PERRIG, Adrian, 2014. Fleet: Defending SDNs from Malicious Administrators. ISBN 9781450329897. Available from DOI: 10.1145/2620728.2620750.

MCKEOWN, Nick; ANDERSON, Tom; BALAKRISHNAN, Hari; PARULKAR, Guru; PETERSON, Larry; REXFORD, Jennifer; SHENKER, Scott y TURNER, Jonathan, 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review*. Vol. 38, pp. 69-74. ISSN 0146-4833.

MICROSOFT, 2009. *The STRIDE Threat Model: Microsoft Docs*. Available also from: https://docs.microsoft.com.

MIT, 1999. *1999 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory*. Available also from: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

MOHAMMADI, Reza; JAVIDAN, Reza; KESHTGARY, Manijeh; CONTI, Mauro y LAL, Chhagan, 2017. Practical extensions to countermeasure DoS attacks in software defined networking. In: Institute of Electrical y Electronics Engineers Inc. Vol. 2017-Janua, pp. 1-6. ISBN 9781538632857. Available from DOI: 10.1109/NFV-SDN.2017.8169839.

MOHAMMMED, Sheena y SRIDEVI, R, 2020. A survey on digital forensics phases, tools and challenges. In: Springer, pp. 237-248.

MOSHREF, Masoud; BHARGAVA, Apoorv; GUPTA, Adhip; YU, Minlan y GOVINDAN, Ramesh, 2014. Flow-level State Transition as a New Switch Primitive for SDN. In: *Proceedings of the third workshop on Hot topics in software defined networking - HotSDN '14*. New York, New York, USA: ACM Press. ISBN 9781450329897. Available also from: http://dx.doi.org/10.1145/2620728.2620729.

MOUSAVI, Seyed Mohammad y ST-HILAIRE, Marc, 2015. Early detection of DDoS attacks against SDN controllers. In: *2015 International Conference on Computing, Networking and Communications, ICNC 2015*. Institute of Electrical y Electronics Engineers Inc., pp. 77-81. ISBN 9781479969593. Available from DOI: 10.1109/ICCNC.2015.7069319.

MUGITAMA, Satria Akbar; CAHYANI, Niken Dwi Wahyu y SUKAMO, Parman, 2020. An Evidence-Based Technical Process for OpenFlow-Based SDN Forensics. In: Institute of Electrical y Electronics Engineers Inc. ISBN 9781728161426. Available from DOI: 10.1109/ICoICT49345.2020.9166215.

MUNKHONDYA, Howard; IKUESAN, Adeyemi R. y VENTER, Hein S., 2020. A case for a dynamic approach to digital forensic readiness in an SDN platform. *Proceedings of the 15th International Conference on Cyber Warfare and Security, ICCWS 2020*, pp. 584-593. Available from DOI: 10.34190/ICCWS.20.049.

MUSUMECI, Francesco; FIDANCI, Ali Can; PAOLUCCI, Francesco; CUGINI, Filippo y TORNATORE, Massimo, 2022. Machine-learning-enabled ddos attacks detection in p4 programmable networks. *Journal of Network and Systems Management*. Vol. 30, pp. 1-27. ISSN 1064-7570.

MUSUMECI, Francesco; IONATA, Valentina; PAOLUCCI, Francesco; CUGINI, Filippo y TORNATORE, Massimo, 2020. Machine-learning-assisted DDoS attack detection with P4 language. In: *IEEE International Conference on Communications*. Institute of Electrical y Electronics Engineers Inc. Vol. 2020-June. ISBN 9781728150895. ISSN 15503607. Available from DOI: 10.1109/ICC40277.2020.9149043.

NATANZI, Seyed Bagher Hashemi y MAJMA, Mohammad Reza, 2017. Secure distributed controllers in SDN based on ECC public key infrastructure. *2017 International Conference on Electrical and Computing Technologies and Applications, ICECTA 2017*. Vol. 2018-Janua, pp. 1-5. ISBN 9781538608722. Available from DOI: 10.1109/ICECTA.2017.8252015.

NATANZI, Seyed Bagher Hashemi y MAJMA, Mohammad Reza, 2018. Secure northbound interface for SDN applications with NTRU public key infrastructure. In: *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation, KBEI 2017*. Institute of Electrical y Electronics Engineers Inc. Vol. 2018-January, pp. 0452-0458. ISBN 9781538626405. Available from DOI: 10.1109/KBEI.2017.8325020.

NGUYEN, Tri Hai y YOO, Myungsik, 2016. Attacks on host tracker in SDN controller: Investigation and prevention. In: *2016 International Conference on Information and Communication Technology Convergence, ICTC 2016*. Institute of Electrical y Electronics Engineers Inc., pp. 610-612. ISBN 9781509013258. Available from DOI: 10.1109/ICTC.2016.7763545.

NGUYEN, Tri Hai y YOO, Myungsik, 2017. Analysis of link discovery service attacks in SDN controller. In: *International Conference on Information Networking*. IEEE Computer Society, pp. 259-261. ISBN 9781509051243. ISSN 19767684. Available from DOI: 10.1109/ICOIN.2017.7899515.

NGUYEN, Xuan-Nam; SAUCEZ, Damien; BARAKAT, Chadi y TURLETTI, Thierry, 2015. Rules placement problem in OpenFlow networks: A survey. *IEEE Communications Surveys Tutorials*. Vol. 18, n.º 2, pp. 1273-1286. ISSN 1553-877X.

NIFE, Fahad y KOTULSKI, Zbigniew, 2018. New SDN-oriented authentication and access control mechanism. In: *Communications in Computer and Information Science*. Springer Verlag. Vol. 860, pp. 74-88. ISBN 9783319924588. ISSN 18650929. Available from DOI: 10.1007/978-3-319-92459-5_7.

NIST, 2012. Computer Security Incident Handling Guide Recommendations of the National Institute of Standards and Technology. Available from DOI: 10.6028/NIST.SP.800-61r2.

NIST, 2014. *NVD-CVE-2014-5035*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2014-5035.

NIST, 2017a. *CVE-CVE-2017-1000411*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2017-1000411.

NIST, 2017b. *NVD - CVE-2015-7516*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2015-7516.

NIST, 2017c. *NVD - CVE-2017-1000080*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2017-1000080.

NIST, 2017d. *NVD - CVE-2017-13763. [Online]. Available:https://nvd.nist.gov/vuln/detail/CVE-2017-13763*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2017-13763.

NIST, 2017e. *NVD-CVE-2014-8149*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2014-8149.

NIST, 2017f. *NVD-CVE-2015-1610*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2015-1610.

NIST, 2017g. *NVD-CVE-2015-1611*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2015-1611.
NIST, 2017h. *NVD-CVE-2017-1000081*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2017-1000081.
NIST, 2017i. *NVD-CVE-2017-1000406*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2017-1000406.
NIST, 2018a. *CVE-2018-1132*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2018-1132.
NIST, 2018b. *NVD - CVE-2018-1000614*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2018-1000614.
NIST, 2018c. *NVD - CVE-2018-1000615*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2018-1000615.
NIST, 2018d. *NVD - CVE-2018-1000616*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2018-1000616.
NIST, 2018e. *NVD-CVE-2018-1078*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2018-1078.
NIST, 2019. *NVD-CVE-2019-13624*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2019-13624.
NIST, 2020a. *CVE - CVE-2014-8730*. Available also from: https://nvd.nist.gov/vuln/detail/cve-2014-8730.
NIST, 2020b. *CVE - CVE-2020-1968*. Available also from: https://nvd.nist.gov/vuln/detail/CVE-2020-1968.
NIST, 2022. Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT). Available from DOI: 10.6028/NIST.IR.8428.
OCHOA ADAY, Leonardo; CERVELLÓ PASTOR, Cristina y FERNÁNDEZ FERNÁNDEZ, Adriana, 2015. Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks, pp. 1-6. Available also from: http://upcommons.upc.edu/handle/2117/77672.
ONGARO, Diego y OUSTERHOUT, John, 2014. In search of an understandable consensus algorithm. In: pp. 305-319. ISBN 1931971102.
ONOS, 2015. *Experimental Features - ONOS 1.1 - Wiki*. Available also from: https://wiki.onosproject.org/display/ONOS11/Experimental+Features#ExperimentalFeatures-TestingReactiveForwarding.
ONOS, 2020. Available also from: https://wiki.onosproject.org/display/ONOS/Security-Mode+ONOS.
OPEN NETWORK FOUNDATION, 2020. *Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions*. Available also from: https://opennetworking.org/onos.
OPEN NETWORKING FOUNDATION, 2014. *SDN architecture*. Available also from: www.opennetworking.org.
OPEN NETWORKING FOUNDATION, 2015a. OpenFlow Switch Specification Version 1.5.1. Vol. 1, p. 283. Available also from: http://www.opennetworking.org.
OPEN NETWORKING FOUNDATION, 2015b. *Principles and Practices for Securing Software-Defined Networks*. Tech. rep. Available also from: www.opennetworking.org.
OPEN NETWORKING FOUNDATION, 2016. *Open Networking Foundation and ON.Lab to Merge to Accelerate Adoption of SDN*. Tech. rep. Available also from: https://opennetworking.

org/news-and-events/press-releases/open-networking-foundation-and-on-lab-to-merge-to-accelerate-adoption-of-sdn.

OPEN NETWORKING FOUNDATION, 2018a. *OpenFlow® Conformance Certification*. Vol. 1. Available also from: https://www.opennetworking.org/product-certification/.

OPEN NETWORKING FOUNDATION, 2018b. *Open Networking Foundation*. Software-Defined Networking (SDN) Definition - Open Networking Foundation [online]. [visited on 2020-04-19]. Available from: https://www.opennetworking.org/sdn-definition/.

OPENDAYLIGHT, 2016. *Security Considerations — OpenDaylight Documentation Aluminum documentation* [online]. [visited on 2021-02-10]. Available from: https://docs.opendaylight.org/en/latest/getting-started-guide/security_considerations.html.

OWASP, 2020. *XML External Entity (XXE) Processing | OWASP*. Available also from: https://owasp.org/www-community/vulnerabilities/XML_External_Entity_(XXE)_Processing.

PADEKAR, Hitesh; PARK, Younghee; HU, Hongxin y CHANG, Sang Yoon, 2016. Enabling dynamic access control for controller applications in software-defined networks. In: Association for Computing Machinery. Vol. 06-08-June, pp. 51-61. ISBN 9781450338028. Available from DOI: 10.1145/2914642.2914647.

PAHLAJANI, Sunny; KSHIRSAGAR, Avinash y PACHGHARE, Vinod, 2019. Survey on private blockchain consensus algorithms. In: IEEE, pp. 1-6. ISBN 172811604X.

PANDYA, Mudit Kalpesh; HOMAYOUN, Sajad y DEHGHANTANHA, Ali, 2018. Forensics investigation of openflow-based SDN platforms. *Advances in Information Security*. Vol. 70, pp. 281-296. Available from DOI: 10.1007/978-3-319-73951-9_14.

PARK, Jaehong y SANDHU, Ravi, 2002. Towards usage control models: beyond traditional access control. In: Association for Computing Machinery (ACM), p. 57. Available from DOI: 10.1145/507711.507722.

PHEMIUS, Kévin; BOUET, Mathieu y LEGUAY, Jérémie, 2014. DISCO: Distributed multi-domain SDN controllers. In: IEEE Computer Society. ISBN 9781479909131. Available from DOI: 10.1109/NOMS.2014.6838330.

POLAT, Huseyin; POLAT, Onur y CETIN, Aydin, 2020. Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models. *Sustainability 2020, Vol. 12, Page 1035*. Vol. 12, p. 1035. ISSN 2071-1050. Available from DOI: 10.3390/SU12031035.

POLLITT, Mark M., 2007. An ad hoc review of digital forensic models. *Proceedings - SADFE 2007: Second International Workshop on Systematic Approaches to Digital Forensic Engineering*, pp. 43-52. ISBN 0769528082. Available from DOI: 10.1109/SADFE.2007.3.

POURVAHAB, Mehran y EKBATANIFARD, Gholamhossein, 2019. Digital Forensics Architecture for Evidence Collection and Provenance Preservation in IaaS Cloud Environment Using SDN and Blockchain Technology. *IEEE Access*. Vol. 7, pp. 153349-153364. Available from DOI: 10.1109/ACCESS.2019.2946978.

PRIETO, Javier Gomez; DROUGKAS(ENISA), Athanasios; GROENLAND, Jelger y (EXPERTS), Alessandro Lazari, 2023. *Demand Side of Cyber Insurance in the EU — ENISA* [online]. [visited on 2023-03-11]. Available from: https://www.enisa.europa.eu/publications/demand-side-of-cyber-insurance-in-the-eu.

PROJECT FLOODLIGHT, 2016. *Architecture - Floodlight Controller - Project Floodlight*. Available also from: https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller.

PROJECTS, LF, 2019. *Home - OpenDaylight*. Available also from: https://www.opendaylight.org/.

RAGHAVAN, Sriram, 2013. Digital forensic research: current state of the art. *Csi Transactions on ICT*. Vol. 1, pp. 91-114. ISSN 2277-9078.

RAZA, Sarwar y LENROW, David, 2013. Open Networking Foundation North Bound Interface Working Group (NBI-WG) Charter. *ONF*.

REITH, Mark; CARR, Clint y GUNSCH, Gregg, 2002. An examination of digital forensic models. *International Journal of digital evidence*. Vol. 1, pp. 1-12.

RIFAI, Myriana; HUIN, Nicolas; CAILLOUET, Christelle; GIROIRE, Frederic; MOULIERAC, Joanna; PACHECO, Dino Lopez y URVOY-KELLER, Guillaume, 2017. MINNIE: An SDN world with few compressed forwarding rules. *Computer Networks*. Vol. 121, pp. 185-207. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2017.04.026.

ROOHITAVAF, Mohammad; REN, Kun; AHN, Jung Sang; ZHANG, Gene; KULKARNI, Sandeep S.; KANG, Woon Hak y BEN-ROMDHANE, Sami, 2019. Session guarantees with raft and hybrid logical clocks. In: Association for Computing Machinery, pp. 100-109. ISBN 9781450360944. Available from DOI: 10.1145/3288599.3288619.

RYU SDN FRAMEWORK COMMUNITY, 2017. *Ryu SDN Framework*. Vol. 32. ISSN 00049417. Available also from: https://ryu-sdn.org/.

SAGARE, Anagha Anilkumar y KHONDOKER, Rahamatullah, 2018. Security Analysis of SDN Routing Applications. In: Springer, vol. 30, pp. 1-17. ISSN 23673389. Available from DOI: 10.1007/978-3-319-71761-6_1.

SAHOO, Kshira Sagar; PUTHAL, Deepak; TIWARY, Mayank; RODRIGUES, Joel J.P.C.; SAHOO, Bibhudatta y DASH, Ratnakar, 2018. An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. *Future Generation Computer Systems*. Vol. 89, pp. 685-697. ISSN 0167739X. Available from DOI: 10.1016/j.future.2018.07.017.

SAHOO, Kshira Sagar; TRIPATHY, Bata Krishna; NAIK, Kshirasagar; RAMASUBBAREDDY, Somula; BALUSAMY, Balamurugan; KHARI, Manju y BURGOS, Daniel, 2020. An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks. *IEEE Access*. Vol. 8, pp. 132502-132513. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2020.3009733.

SAKIC, Ermin y KELLERER, Wolfgang, 2019. Response Time and Availability Study of RAFT Consensus in Distributed SDN Control Plane. *IEEE Transactions on Network and Service Management*. Vol. 15, pp. 304-318. Available from DOI: 10.1109/TNSM.2017.2775061.

SALFATI, Eran y PEASE, Michael, 2022. Digital Forensics and Incident Response (DFIR) Framework for Operational Technology (OT). Available from DOI: 10.6028/NIST.IR.8428.

SANTOS, Reneilson; SOUZA, Danilo; SANTO, Walter; RIBEIRO, Admilson y MORENO, Edward, 2020. Machine learning algorithms to detect DDoS attacks in SDN. *Concurrency and Computation: Practice and Experience*. Vol. 32, e5402. ISSN 1532-0634. Available from DOI: 10.1002/CPE.5402.

SCHLETTE, Daniel; CASELLI, Marco y PERNUL, Günther, 2021. A comparative study on cyber threat intelligence: The security incident response perspective. *IEEE Communications Surveys Tutorials*. Vol. 23, pp. 2525-2556. ISSN 1553-877X.

SCOTT-HAYWARD, Sandra, 2017. Trailing the Snail: SDN Controller Security Evolution. Available also from: https://arxiv.org/abs/1711.08406v1.

SECCI, Stefano; DIAMANTI, Alessio; SANCHEZ, José Manuel Vilchez; BAH, Mamadou Tahirou; VIZARRETA, Petra; MACHUCA, Carmen Mas; SCOTT-HAYWARD, Sandra y SMITH, Dylan, 2019. Security and Performance Comparison of ONOS and ODL controllers.

SHIRAVI, Ali; SHIRAVI, Hadi; TAVALLAEE, Mahbod y GHORBANI, Ali A, 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *computers security*. Vol. 31, pp. 357-374. ISSN 0167-4048.

SHOHANI, Reza Bakhtiari y MOSTAFAVI, Seyed Akbar, 2020. Introducing a New Linear Regression Based Method for Early DDoS Attack Detection in SDN. In: *2020 6th International Conference on Web Research, ICWR 2020*. Institute of Electrical y Electronics Engineers Inc., pp. 126-132. ISBN 9781728110516. Available from DOI: 10.1109/ICWR49608.2020.9122310.

SHOU, Chaofan, 2021. Porkfuzz: Testing stateful software-defined network applications with property graphs. In: pp. 1660-1662.

SHRIVASTAVA, Pragati; AGARWAL, Annanay y KATAOKA, Kotaro, 2018. Poster: Detection of topology poisoning by silent relay attacker in SDN. In: *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*. New York, New York, USA: ACM Press, pp. 792-794. ISBN 9781450359030. Available from DOI: 10.1145/3241539.3267763.

SHU, Zhaogang; WAN, Jiafu; LIN, Jiaxiang; WANG, Shiyong; LI, Di; RHO, Seungmin y YANG, Changcai, 2016. Traffic Engineering in Software-Defined Networking: Measurement and Management. *IEEE Access*. Vol. 4, pp. 3246-3256. ISSN 21693536. Available from DOI: 10.1109/ACCESS.2016.2582748.

SINGH, Avinash; IKUESAN, Richard Adeyemi y VENTER, Hein, 2022. Secure Storage Model for Digital Forensic Readiness. *IEEE Access*. Vol. 10, pp. 19469-19480. ISSN 2169-3536. Available from DOI: 10.1109/ACCESS.2022.3151403.

SKOWYRA, Richard; XU, Lei; GU, Guofei; DEDHIA, Veer; HOBSON, Thomas; OKHRAVI, Hamed y LANDRY, James, 2018. Effective topology tampering attacks and defenses in Software-Defined networks. In: *Proceedings - 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2018*. Institute of Electrical y Electronics Engineers Inc., pp. 374-385. ISBN 9781538655955. Available from DOI: 10.1109/DSN.2018.00047.

SONG, Chungsik; PARK, Younghee; GOLANI, Keyur; KIM, Youngsoo; BHATT, Kalgi y GOSWAMI, Kunal, 2017. Machine-learning based threat-aware system in software defined networks. In: *2017 26th International Conference on Computer Communications and Networks, ICCCN 2017*. Institute of Electrical y Electronics Engineers Inc. ISBN 9781509029914. Available from DOI: 10.1109/ICCCN.2017.8038436.

SU, Shang Chiuan; CHEN, Yi Ren; TSAI, Shi Chun y LIN, Yi Bing, 2018. Detecting P2P Botnet in Software Defined Networks. *Security and Communication Networks*. Vol. 2018. ISSN 19390122. Available from DOI: 10.1155/2018/4723862.

SUN, Chen; BI, Jun; CHEN, Haoxian; HU, Hongxin; ZHENG, Zhilong; ZHU, Shuyong y WU, Chenghui, 2017. SDPA: Toward a Stateful Data Plane in Software-Defined Networking. *IEEE/ACM Transactions on Networking*. Vol. 25, pp. 3294-3308. ISSN 10636692. Available from DOI: 10.1109/TNET.2017.2726550.

TANG, Tuan Anh; MCLERNON, Des; MHAMDI, Lotfi; ZAIDI, Syed Ali Raza y GHOGHO, Mounir, 2019. Intrusion detection in sdn-based networks: Deep recurrent neural network

approach. In: Springer, pp. 175-195. ISSN 23639466. Available from DOI: 10.1007/978-3-030-13057-2_8.

TOOTOONCHIAN, Amin y GANJALI, Yashar, 2010. Hyperflow: A distributed control plane for openflow. In: vol. 3.

TOSHNIWAL, Bhavesh; JOSHI, Kalpana D.; SHRIVASTAVA, Pragati y KATAOKA, Kotaro, 2019. BEAM: BEhavior-based access control mechanism for SDN applications. In: Institute of Electrical y Electronics Engineers Inc. Vol. 2019-July. ISBN 9781728118567. ISSN 10952055. Available from DOI: 10.1109/ICCCN.2019.8846954.

TSENG, Yuchia; NAÏT-ABDESSELAM, Farid y KHOKHAR, Ashfaq, 2018. SENAD: Securing network application deployment in software defined networks. In: Institute of Electrical y Electronics Engineers Inc. Vol. 2018-May. ISBN 9781538631805. ISSN 15503607. Available from DOI: 10.1109/ICC.2018.8422405.

TSENG, Yuchia; PATTARANANTAKUL, Montida; HE, Ruan; ZHANG, Zonghua y NAIT-ABDESSELAM, Farid, 2017. Controller DAC: Securing SDN controller with dynamic access control. In: Institute of Electrical y Electronics Engineers Inc. ISBN 9781467389990. ISSN 15503607. Available from DOI: 10.1109/ICC.2017.7997249.

TSENG, Yuchia; ZHANG, Zonghua y NAIT-ABDESSELAM, Farid, 2016. ControllerSEPA: A security-enhancing SDN controller plug-in for OpenFlow applications. In: IEEE Computer Society. Vol. 0, pp. 268-273. ISBN 9781509050819. Available from DOI: 10.1109/PDCAT.2016.064.

UJCICH, Benjamin E.; JERO, Samuel; EDMUNDSON, Anne; WANG, Qi; SKOWYRA, Richard; LANDRY, James; BATES, Adam; SANDERS, William H.; NITA-ROTARU, Cristina y OKHRAVI, Hamed, 2018. Cross-app poisoning in software-defined networking. In: Association for Computing Machinery, pp. 648-663. ISBN 9781450356930. ISSN 15437221. Available from DOI: 10.1145/3243734.3243759.

VIET, An Nguyen; VAN, Luan Phung; MINH, Hoang Anh Nguyen; XUAN, Huy Duong; NGOC, Nam Pham y HUU, Thanh Nguyen, 2017. Mitigating HTTP GET flooding attacks in SDN using NetFPGA-based OpenFlow switch. In: *ECTI-CON 2017 - 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Institute of Electrical y Electronics Engineers Inc., pp. 660-663. ISBN 9781538604496. Available from DOI: 10.1109/ECTICon.2017.8096324.

VIJAY TIJARE, Poonam y VASUDEVAN, Deepika, 2016. The Northbound APIs of Software Defined Networks. © *International Journal of Engineering Sciences & Research Technology*. Vol. 501, n.º January 2019. ISSN 2277-9655. Available from DOI: 10.5281/zenodo.160891.

WALTON, Douglas y REED, Chris, 2008. Evaluating corroborative evidence. *Argumentation*. Vol. 22, pp. 531-553. ISSN 0920427X. Available from DOI: 10.1007/S10503-008-9104-0/FIGURES/13.

WANG, Haopei; XU, Lei y GU, Guofei, 2015. FloodGuard: A DoS Attack Prevention Extension in Software-Defined Networks. In: *Proceedings of the International Conference on Dependable Systems and Networks*. IEEE Computer Society. Vol. 2015-Septe, pp. 239-250. ISBN 9781479986293. Available from DOI: 10.1109/DSN.2015.27.

WANG, Haopei; YANG, Guangliang; CHINPRUTTHIWONG, Phakpoom; XU, Lei; ZHANG, Yangyong y GU, Guofei, 2018. Towards fine-grained network security forensics and diagnosis in the SDN era. In: Association for Computing Machinery. Vol. 14, pp. 3-16. ISBN 9781450356930. ISSN 15437221. Available from DOI: 10.1145/3243734.3243749.

WANG, Tao y CHEN, Hongchang, 2017. SGuard: A lightweight SDN safe-guard architecture for DoS attacks. *China Communications*. Vol. 14, n.º 6, pp. 113-125. ISSN 16735447. Available from DOI: 10.1109/CC.2017.7961368.

WANG, Tao; CHEN, Hongchang; CHENG, Guozhen y LU, Yulin, 2018. SDNManager: A safeguard architecture for SDN DoS attacks based on bandwidth prediction. *Security and Communication Networks*. Vol. 2018. ISSN 19390122. Available from DOI: 10.1155/2018/7545079.

WU, Di; LI, Jie; DAS, Sajal K.; WU, Jinsong; JI, Yusheng y LI, Zhetao, 2018. A novel distributed denial-of-service attack detection scheme for software defined networking environments. In: *IEEE International Conference on Communications*. Institute of Electrical y Electronics Engineers Inc. Vol. 2018-May. ISBN 9781538631805. ISSN 15503607. Available from DOI: 10.1109/ICC.2018.8422448.

XIANG, Shuangqing; ZHU, Huibiao; XIAO, Lili y XIE, Wanling, 2018. Modeling and verifying topoguard in openflow-based software defined networks. In: *Proceedings - 2018 12th International Symposium on Theoretical Aspects of Software Engineering, TASE 2018*. Institute of Electrical y Electronics Engineers Inc. Vol. 2018-Janua, pp. 84-91. ISBN 9781538673058. Available from DOI: 10.1109/TASE.2018.00019.

XIE, Shengxu; XING, Changyou; ZHANG, Guomin y ZHAO, Jinlong, 2021. A Table Overflow LDoS Attack Defending Mechanism in Software-Defined Networks. *Security and Communication Networks*. Vol. 2021. ISSN 19390122. Available from DOI: 10.1155/2021/6667922.

XING, Jiarong; CHEN, Ang y EUGENE NG, T. S., 2020. Secure State Migration in the Data Plane. In: *Proceedings of the 2020 ACM SIGCOMM Workshop on Secure Programmable Network Infrastructure, SPIN 2020*. Association for Computing Machinery, pp. 28-34. ISBN 9781450380416. Available from DOI: 10.1145/3405669.3405822.

XING, Jiarong; WU, Wenqing y CHEN, Ang, 2019. Architecting Programmable Data Plane Defenses into the Network with FastFlex. In: *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*. New York, NY, USA: ACM. ISBN 9781450370202. Available also from: https://doi.org/10.1145/3365609.3365860.

XU, Jianfeng; WANG, Liming y XU, Zhen, 2020. An enhanced saturation attack and its mitigation mechanism in software-defined networking. *Computer Networks*. Vol. 169, p. 107092. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2019.107092.

YAZDINEJAD, Abbas; PARIZI, Reza M.; DEHGHANTANHA, Ali y CHOO, Kim Kwang Raymond, 2020. P4-to-blockchain: A secure blockchain-enabled packet parser for software defined networking. *Computers and Security*. Vol. 88, p. 101629. ISSN 01674048. Available from DOI: 10.1016/j.cose.2019.101629.

YIN, H; XIE, H; TSOU, T; LOPEZ, D; ARANDA, P y SIDI, R, 2012. Sdni: A message exchange protocol for software defined networks (sdns) across multiple domains. *IETF draft, work in progress*.

YU, Haisheng; QI, Heng y LI, Keqiu, 2020. WECAN: an Efficient West-East Control Associated Network for Large-Scale SDN Systems. *Mobile Networks and Applications*. Vol. 25, pp. 114-124. ISSN 15728153. Available from DOI: 10.1007/s11036-018-1194-9.

ZAIDI, Zainab; FRIDERIKOS, Vasilis; YOUSAF, Zarrar; FLETCHER, Simon; DOHLER, Mischa y AGHVAMI, Hamid, 2018. Will SDN be part of 5G? *IEEE Communications Surveys and Tutorials*. Vol. 20, n.º 4, pp. 3220-3258. ISSN 1553877X. Available from DOI: 10.1109/COMST.2018.2836315.

ZHANG, Tianzhu; BIANCO, Andrea; DOMENICO, Samuele De y GIACCONE, Paolo, 2016. The Role of Inter-Controller Traffic for Placement of Distributed SDN Controllers. *Computer Communications*. Vol. 113, pp. 1-13. Available from DOI: 10.1016/j.comcom.2017.09.007.

ZHANG, Xiaoquan; CUI, Lin; WEI, Kaimin; TSO, Fung Po; JI, Yangyang y JIA, Weijia, 2021. *A survey on stateful data plane in software defined networks*. Vol. 184. Elsevier B.V. ISSN 13891286. Available from DOI: 10.1016/j.comnet.2020.107597. no tiene como parte primaria la seguridad.

ZHANG, Yuan; CUI, Lin; WANG, Wei y ZHANG, Yuxiang, 2018. *A survey on software defined networking with multiple controllers*. Vol. 103. Academic Press. ISSN 10958592. Available from DOI: 10.1016/j.jnca.2017.11.015.

ZHOU, Yadong; CHEN, Kaiyue; ZHANG, Junjie; LENG, Junyuan y TANG, Yazhe, 2018. Exploiting the Vulnerability of Flow Table Overflow in Software-Defined Network: Attack Model, Evaluation, and Defense. *Security and Communication Networks*. Vol. 2018. ISSN 19390122. Available from DOI: 10.1155/2018/4760632.

ZHU, Liehuang; KARIM, Md Monjurul; SHARIF, Kashif; LI, Fan; DU, Xiaojiang y GUIZANI, Mohsen, 2019. *SDN controllers: Benchmarking performance evaluation*. Available also from: http://arxiv.org/abs/1902.04491.

ZHU, Shao Ying; SCOTT-HAYWARD, Sandra; JACQUIN, Ludovic e HILL, Richard, 2017. *Guide to Security in SDN and NFV: Challenges, Opportunities, and Applications*. Springer. ISBN 3319646532.

ZINGE, Prachi y CHATTERJEE, Madhumita, 2018. Comprehensive study of digital forensics branches and tools. *The International Journal of Forensic Computer Science*. Vol. 13, pp. 22-28. ISSN 18099807. Available from DOI: 10.5769/J201801002.