CAMPUS
DE EXCELENCIA
INTERNACIONAL

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros Informáticos

TRABAJO FIN DE GRADO

DESPLIEGUE DE SISTEMA MULTI-AGENTE PARA LA DETECCIÓN Y MITIGACIÓN DE ATAQUES DE DENEGACIÓN DE SERVICIO.

Autor: Juan Cano de Benito

Director: D. Javier Bajo Pérez

MADRID, JULIO 2018

RESUMEN

Los ataques de tipo denegación de servicio (DDOS) es un tipo de ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible para los usuarios. Supone el tercer tipo de ataque más frecuente en el mundo; y afecta, principalmente, a empresas de Estados Unidos y Europa, provocando pérdidas millonarias. De hecho, países como EEUU dedican gran parte de su inversión en prevenir, mitigar y gestionar estos tipos de ataques.

Para afrontar este problema, los sistemas inteligentes son considerados como los mejores sistemas en relación coste/beneficio. Dentro de estos sistemas, surge el concepto de agente, el cual se define como una entidad inteligente, adaptable, autónoma y proactiva capaz de funcionar en ecosistemas heterogéneos compuestos por diferentes entornos, dispositivos y plataformas.

Este trabajo plantea una solución basada en un Sistema Multi-Agente (MAS) capaz de detectar y mitigar ataques de tipo DDOS en redes de telecomunicaciones. Más concretamente, aquellos con mayor frecuencia, como pueden ser los basados sobre los protocolos ICMP, UDP y TCP. Este sistema desplegado se comunica con una red neuronal diseñada explícitamente para predecir el protocolo utilizado en el ataque. Además, el Sistema Multi-Agente se diseña en base a la metodología "*Plan, Do, Check, & Atc*", propia de la norma ISO 27001.

Este sistema ha sido simulado en una red virtualizada. En dicha simulación, varios sistemas atacan periódicamente a una red defensora, mitigando el ataque y garantizando a su vez el acceso a los usuarios. Los resultados han sido significativos. Los agentes obtuvieron éxito mitigando varios tipos de ataque, no siendo así en uno de los casos en el cual la red no era capaz de frenar el ataque.

Con todo esto, se han conseguido la mayoría de los objetivos. Sin embargo, es necesario estudiar más profundamente estos tipos de ataque dada su variedad, magnitud e impacto.

ABSTRACT

Denial-of-Service (DDOS) is a network attack on a computer system or a network itself that disables the access to a certain service or resource to users. It is the third most frequent type of network attack in the world; and it affects, mainly, companies from United States and Europe, causing millions in losses. Moreover, countries such as US allocates a large amount of their investment to prevent, mitigate and manage these attacks.

Intelligent systems are considered the best cost/benefit system to deal with those threats. Within these systems, the concept of Agent emerges, which is defined as an intelligent, adaptable, autonomous and proactive entity, capable of functioning in heterogeneous ecosystems comprise of different environments, devices and platforms.

The present work proposes a Multi-Agent System (MAS) solution, capable of detecting and mitigating DDOS in networks. Concretely, DDOS based on ICMP, UDP and TCP protocols, considered the most frequent. The MAS communicates with a neural network designed explicitly to predict the protocol used during the attack. Furthermore, the MAS has been designed based on the "*Plan, Do, Check, & Atc*" methodology, main concept of the ISO 27001 standard.

This system has been simulated in a virtualized network. During simulation, several systems have been periodically attacking a network. The defender network must mitigate the attack and guaranteeing access to users at the same time. The results have been significant. The agents succeed mitigating several attacks. Although in one case, the network was not able to recover.

Having said that, most of the objectives have been achieved. Nevertheless, it is necessary to study these threats in a deepener way, considering their variety, magnitude, and impact.

"Ningún ordenador ha sido jamás diseñado para ser consciente de lo que está haciendo; pero la mayor parte del tiempo, nosotros tampoco lo somos".

- Marvin Minsky (1927 – 2016) -

ÍNDICE

ÍNDICE DE FIGURAS

1. DESCRIPCIÓN DEL PROBLEMA

1.1 Introducción

En un estudio del 2017 realizado por Accenture (1), es posible comprobar que el mercado objetivo de los ciber ataques, y al que más perjudica, es el sector financiero, estimando hasta 18 millones de dólares de pérdidas en el sector por cada 254 compañías.

Adicionalmente, según el mismo estudio, el ataque de tipo denegación de servicio (DDOS), el cual es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos y normalmente provoca la pérdida de la conectividad de la red por el consumo de banda ancha de la red de la víctima o sobrecarga de los recursos computacionales del sistema atacado, es el tercer tipo de ataque más frecuentemente utilizado, afectando principalmente a empresas de Estados Unidos y Europa, siendo el tercero que más invierten en resolver en Estados Unidos y provocando pérdidas de un millón y medio de dólares por cada 254 empresas. También es posible comprobar en el mismo estudio que el mejor sistema en relación costes/beneficios para prevenir los ciber ataques son los sistemas inteligentes, con un ahorro de costos valorado en hasta 3 millones de dólares. Según CISCO (2), esta amenaza está creciendo, es más efectiva y con nuevas técnicas de ataque. En 2017, dos de cada cinco empresas experimentaron una ampliación de ataques y eficacia (2).

Dado que las soluciones más apropiadas para afrontar estos tipos de ataques son los sistemas inteligentes (1), parece adecuado analizar este tipo de sistema, que es un programa de computación que reúne características y comportamientos asimilables al de la inteligencia humana o animal. En este proyecto se utilizan los sistemas multi-agente o inteligencia artificial distribuida, dadas las ventajas que aportan a la hora de abordar el problema de forma distribuida.

No existe una definición formal y precisa de lo que es un agente, pero el término de agente viene del latín “agere”, que significa hacer. Agente deriva del participio “agens”, que expresa la capacidad de acción o actuación de una entidad. El concepto de agente, en informática, es generalmente visto como entidad inteligente, que caracteriza a una entidad software adaptable que puede funcionar en distintos entornos o plataformas y es capaz de realizar de forma inteligente y autónoma distintos objetivos. Sus características más destacables son: funcionamiento continuo y autónomo, comunicación con el entorno y con otros agentes (humanos o no), robustez, adaptabilidad (realizar objetivos y tareas en distintos dominios), razonamiento y aprendizaje (ambos necesarios para que se comporte inteligentemente) y movilidad (capaz de desplazarse entre los nodos de una red). (3)

Los agentes en un sistema multi-agente suele referirse por lo general a entidades, pero, sin embargo, podrían ser un robot o un ser humano (3).

En el mundo real, los sistemas multi-agente suelen aplicarse a juegos de ordenador, películas, sistemas de defensa, transporte, logística, sistemas de información geográfica, diagnóstico de enfermedades y, en lo que en este proyecto respecta, tecnología de red. (4)

Los sistemas multi-agente resultan especialmente adecuados para abortar los ataques de denegación de servicio, ya que es posible diseñar agentes autónomos especializados en tareas tales como la monitorización, la clasificación o la visualización entre otras. La

monitorización de la red consiste en la supervisión permanente del estado de la red para evitar las infracciones de la política de seguridad. El objetivo de la seguridad de red es proporcionar la estabilidad en los sistemas monitoreados. Sin lugar a duda, los sistemas de seguridad de la red deben lidiar con multitud de problemas. El mayor problema para los sistemas de seguridad de red es encontrar una diferencia entre el comportamiento de un usuario normal y un potencial atacante (5). Para afrontar la complejidad del problema, se deben aplicar mecanismos que permitan afrontar no solo el conocimiento de patrones de ataques previamente conocidos, si no la falta de coherencia y la inconsistencia.

Finalmente, debido a todo lo anteriormente expuesto, este trabajo es conveniente, los ciber ataques son un problema actual que cada año cobran más relevancia y hace uso de una tecnología que está cobrando un gran auge de implementación en empresas, al ser un sistema más rentable que cualquier otro método, por lo que se propone la creación de un sistema multi agente para la detección de ataques DDOS y clasificar estos ataques en una red neuronal, para generalizar comportamientos a partir de la información suministrada por los agentes.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un sistema multi-agente para la detección de ataques de tipo DDOS en redes de telecomunicaciones. Los sistemas multi-agente son mecanismos de inteligencia artificial distribuida que resultan especialmente adecuados para resolver problemas de ataques de denegación de servicio, dadas sus capacidades, tales como autonomía, adaptación o proactividad.

Un ejemplo básico de ataque DDOS sería, por ejemplo, una “Inundación UDP”, en el que el atacante supera los puertos aleatorios en el host destino con paquetes IP que contienen datagramas UDP. El host receptor verifica aplicaciones asociadas con este datagrama y responderá al host cliente, a medida que se reciben y responden más paquetes UDP, el sistema quedará colapsado y no responderá a otros clientes.

1.2.2 Objetivos específicos

- Determinar y analizar la problemática de los ataques de denegación de servicios que se puedan producir en la red.

- Revisar el estado del arte de la técnica, que será todas aquellas herramientas tecnológicas que se han utilizado en este trabajo.

- Diseñar e implementar un sistema multi-agente para diagnosticar los ataques en las redes de telecomunicaciones en distintos escenarios, ya sean viniendo de un único objetivo o de varios objetivos.

- Creación una red neuronal que ayude a clasificar ataques de red. Esta red neuronal será llamada por uno de los agentes para predecir porque protocolo se produce el ataque.

- Por último, la evaluación de los resultados obtenidos en la fase de diseño e implementación y sugerir posibles mejoras.

1.3 Planificación

La planificación que se ha tomado es la siguiente:

- **Descripción del problema.** Corresponde con la introducción y motivación del problema que se desarrollará.

- **Gestión de riesgos.** Se describen y detallan tipos de ataques DDOS, indicando algoritmos para su detección y prevención, que se usará en este trabajo.

- **Estado de la técnica.** Se detallará el planteamiento del problema, contextualizando el marco teórico y las herramientas utilizadas para resolverlo. Incluyendo la fase de procesamiento de datos seguidos del posterior análisis por el sistema agente.

- **Diseño.** Se diseñarán los distintos escenarios, planteamientos y actos que deba realizar los agentes en cada uno.

- **Implementación.** Se implementará el sistema agente en cada uno de los escenarios definidos, se evaluará su comportamiento y su consiguiente validación.

- **Evaluación y conclusiones.** Tras las simulaciones, se evaluará si los resultados son los esperados y la conclusión de cada prueba, si son satisfactorias o se podrían mejorar en un futuro, y sus líneas futuras.

2. ESTADO DEL ARTE DE LA TÉCNICA

2.1 Gestión de Seguridad de la Información

El Sistema de Gestión de Seguridad de la Información (SGSI) es el principal concepto sobre el que se conforma la norma ISO 27001 (6). Esta gestión se deberá realizar mediante un proceso sistémico, documentado y conocido por toda la empresa.

El SGSI consiste en preservar la confidencialidad, integridad y disponibilidad, adaptándose a los cambios internos y externos de la organización (7), estos consisten en:

- Confidencialidad: la información es privada y no se debe exponer a individuos o entidades no autorizadas.

- Integridad: Mantener de forma compleja y exacta la información y métodos del proceso.

- Disponibilidad: Acceder y utilizar la información y los sistemas de tratamiento cuando las entidades, procesos o individuos lo requieran.

En un ataque informático, cualquiera de estos puntos podría quedar expuesto. La ISO 27001, por lo tanto, incorpora la metodología PDCA (Plan Do Check Act), siendo este un círculo de Deming que proporciona una mejora continua:

- Plan (Planificar): Es una fase de diseño del SGSI, realizando una evaluación de los riesgos de la información y la selección de los controles adecuados.

- Do (Hacer): La fase en la que se desarrolla la implantación y operación de los controles de gestión de riesgos.

- Check (Controlar): Esta fase tiene como objetivo revisar y evaluar el desempeño del SGSI.

- Act (Actuar): En esta última fase se realizan los cambios pertinentes cuando sean necesarios para corregir fallos y mejorar el rendimiento.

Se hará uso de la normativa ISO 27001 para tener un plan de actuación dependiendo del escenario, incorporando esta técnica tanto en escenarios favorables, como para corregir y mejorar técnicas en escenarios desfavorables.

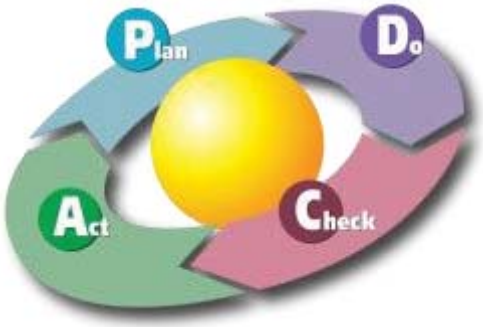


Ilustración 1. Círculo de Deming con la metodología PDCA.

2.2 Gestión de incidentes

El principal objetivo de la gestión de incidentes es restaurar el funcionamiento correcto del servicio tan pronto como sea posible y minimizar el efecto adverso sobre las operaciones del negocio. (8)

La gestión de incidentes se puede dividir en (9):

- Incidente: Interrupción no planificada o reducción de calidad de un servicio.

- Problema: Es una condición causada por una serie de incidentes o que tienen problemas comunes. Es más grave que un incidente y necesita un estudio posterior para evitar futuros incidentes.

- Workaround: Solución temporal para resolver problemas.

- Error conocido: Problema del cual se conoce su causa y solución temporal (workaround).

Para la gestión de incidentes, tendremos un proceso con los siguientes pasos (8):

- Registro: El incidente debe ser registrado.

- Clasificación: El incidente se caracteriza en términos de tipo, impacto y urgencia, lo que conlleva a un tipo de prioridad.

- Coincidencia: Una solución (workaround) puede existir ya si el incidente coincide con un problema conocido o condición de error.

- Resolución: Aplicación de la solución para restablecer el servicio normal.

- Cierre: El incidente se cierra una vez el servicio ha sido restaurado.

2.3 Agentes

Como ya se mencionó en el capítulo introductorio de este trabajo fin de grado, un agente es una entidad que es capaz de percibir su entorno, procesar tales percepciones y responder o actuar autónomamente en un ambiente de manera correcta para alcanzar su objetivo, aprendiendo o utilizando su conocimiento para lograrlo.

Este ambiente puede ser físico (por ejemplo, un termostato o una cámara réflex) o computacional (fuentes de datos, computo u otros agentes). Se pueden distinguir dos tipos de agentes: agentes inteligentes autónomos y agentes inteligentes abstractos, respectivamente, pero al ser un comportamiento dirigido a objetivos que hacen uso de la inteligencia, se puede resumir los tipos en uno solo: agentes racionales. (10)

Según su grado de inteligencia y capacidades percibidas, se distinguen 5 tipos de agentes:

- Agentes con reflejos simples: Actúan solo sobre la base de percepción actual. Se basan en la regla de condición-acción.

- Agentes basados en modelos: Pueden manejarse en entornos parcialmente observables. Su estado se almacena en el interior del agente manteniendo una estructura que describe la parte del mundo que no se puede ver.

- Agentes basados en objetivos: Amplían las capacidades de los agentes basados en modelos, mediante el uso de los objetivos. Un objetivo es una situación deseable, eso permite al agente elegir entre multitud de posibilidades, seleccionando la más adecuada para alcanzar el objetivo.

- Agentes basados en utilidad: A diferencia del anterior tipo de agente, que solo puede detectar un objetivo realizado o no realizado, este agente mide cuan de deseable es un objetivo en particular, este tipo de agente mide los objetivos y elige el que mejor se adecue a la mejor opción.

- Agentes con aprendizaje: El agente puede operar en un entorno desconocido y volverse más competente de lo que su conocimiento inicial podría permitir. Es capaz de realizar mejoras, toma objetivos y decide la acción.

Un agente, según Wooldridge (11), posee las características de reactividad, proactividad y habilidad social:

- Reactividad: Los agentes reaccionan a los cambios en su ambiente según lo consideren oportuno.

- Proactividad: Los agentes tienen un comportamiento orientado a metas, por lo que cambiará su comportamiento para alcanzar una meta.

- Habilidad social: Los agentes son capaces de interactuar con otros agentes. Pueden negociar e interactuar de una forma cooperativa, por medio de un lenguaje entre agentes, el cual les permite comunicarse en lugar de simplemente intercambiar datos. Un agente puede comunicarse con otros agentes informáticos, o con un ser humano (también considerado agente, como dijimos anteriormente) ya sea para transmitirle un resultado o para informarle de un problema que no encuentre solución.

2.4 Multi-Agentes

Un sistema multi-agente es aquel sistema que comprende dos o más agentes. A pesar de que pueda existir una meta global en el sistema, cada agente podrá tener su propia meta. Estos agentes deberían tener habilidad social y ser capaces de comunicarse entre sí. (11)

En un sistema multi-agente, los agentes se pueden dividir de más simples a más complejos, y pueden no ser necesariamente inteligentes. Diferenciamos dos tipos de sistemas agentes dentro de un sistema multi-agente: (12)

- Agentes pasivos: Agentes sin objetivos.

- Agentes activos: Agentes con objetivos.

Para explicar estos dos tipos de agentes, podría asemejarse al acertijo del lobo, la cabra y la col. Un agente pasivo sería una col, los agentes activos serían el lobo y la cabra, el lobo tiene como objetivo comerse a la cabra, la cabra tiene como objetivo comerse la col y la meta global es llegar al otro lado del río.

Además, los agentes en un sistema multi-agente, tienen varias características importantes: (13)

- Autonomía: Agentes parcialmente independientes, autoconscientes y autónomos.

- Vista local: Ningún agente tiene una vista global del sistema.

- Descentralización: Ningún agente controla a otro.

Que el agente sea autónomo significa que es su decisión atender o no la petición que le hayan hecho y con qué prioridad. Esto es muy útil cuando un agente recibe multitud de peticiones y no puede responder a todas en un tiempo razonable. Al no estar unidos directamente, se podrá sacar un agente del sistema mientras los demás siguen en ejecución, esto permite la flexibilidad y extensibilidad.

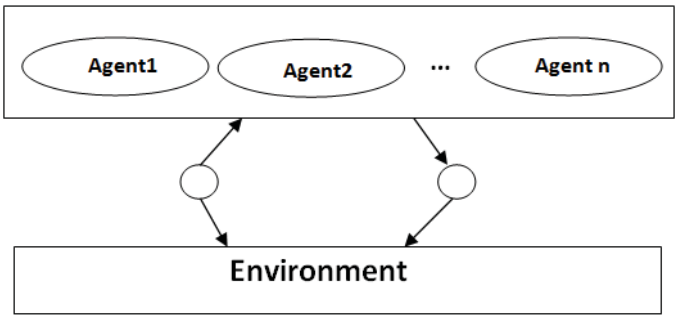


Ilustración 2. Ejemplo de sistema multi-agente.

2.5 Sistemas Multi-Agentes en sistemas de seguridad

Los sistemas de telecomunicaciones son un campo donde los sistemas multi-agentes han sido usados con éxito. Los sistemas de telecomunicaciones son grandes redes distribuidas con componentes interconectados que necesitan ser monitoreados y administrados en tiempo real, y forman un mercado competitivo donde las compañías de telecomunicaciones y los proveedores de servicio buscan distinguirse del resto de competidores proporcionando servicios mucho mejores, más rápidos y confiables.

2.6 Ataque de Denegación de Servicios

Un ataque distribuido de denegación de servicio (DDOS) es un intento malicioso de interrumpir el tráfico normal de un servidor, servicio o red, abrumando el objetivo o la infraestructura circundante. Cuantos más sistemas informáticos se utilicen para este ataque contra un único objetivo, más eficiente será el ataque.

Estos sistemas informáticos comúnmente serán una red de máquinas, infectadas con malware. A esta red de máquinas se le conoce como botnet. Una vez se ha establecido una botnet, el atacante atacará a la dirección IP objetivo, abrumando a solicitudes al objetivo. Debido a que cada bot puede ser un dispositivo legítimo de Internet, separar el tráfico de ataque del tráfico normal puede ser difícil. (14)

Los ataques DDOS se dirigen a diversos componentes de una conexión. Para entender los diferentes tipos de ataques, primero es necesario mencionar las capas de conectividad del modelo de interconexión de sistemas abiertos (OSI), ISO/IEC 7498-1.

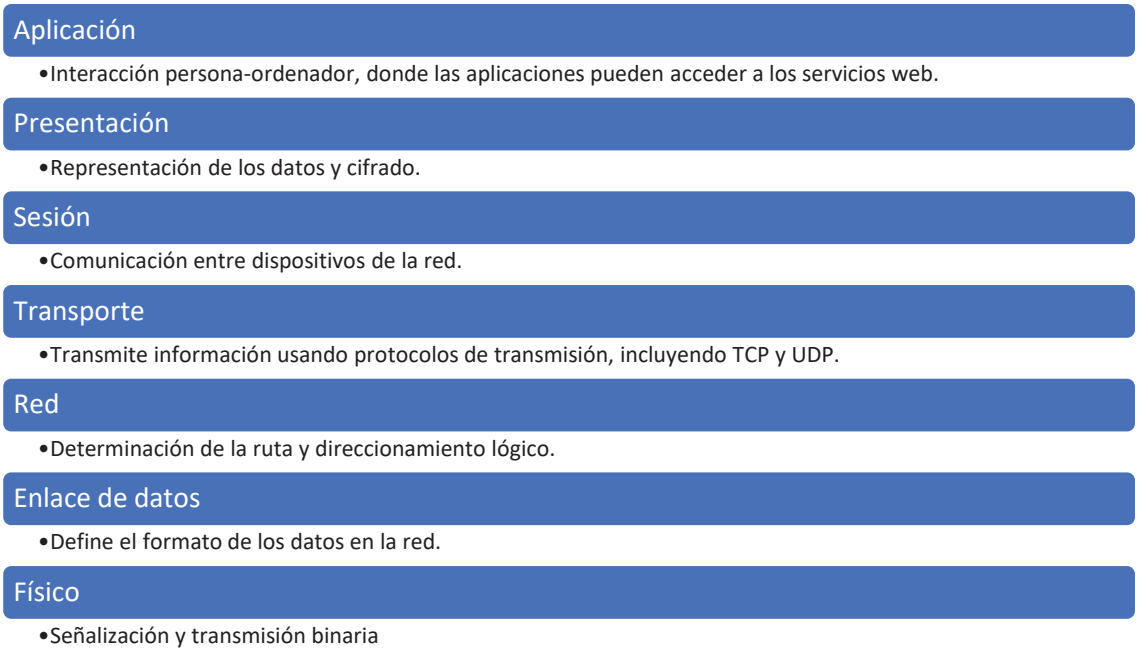


Ilustración 3. Pila de capas o niveles del modelo OSI

Dentro de los ataques de denegación de servicios se pueden distinguir tres categorías: ataques en la capa de aplicación, ataque de agotamiento de estado y ataques volumétricos. Los ataques por analizar en este trabajo serán del 2.6.1 al 2.6.4, pero en la revisión del estado del arte se describirán la gran mayoría de ataques de denegación de servicios existentes.

2.6.1 Ping flood

Una inundación ping es un ataque DDOS de capa 3 (Red) en el cual se intenta abrumar al objetivo mediante la solicitud de paquetes ICMP, haciendo inaccesible el tráfico normal.

La solicitud Ping es una herramienta de diagnóstico de red. Los mensajes ICMP se utilizan con el fin de diagnosticar el estado y conectividad del dispositivo, así como la conexión entre remitente y dispositivo. La solicitud requiere tanto ancho de banda en la respuesta entrante como en la saliente, por lo que un ataque de este tipo tendrá como objetivo abrumar a la víctima con solicitudes y respuestas y sobrecargar la red con tráfico falso.

Se puede evitar deshabilitando o limitando la funcionalidad ICMP del objetivo.

2.6.2 HTTP flood

Un ataque de inundación HTTP (15) está diseñado para colapsar el objetivo con solicitudes HTTP. Es un ataque de capa 7 (Aplicación).

La mitigación de estos ataques es altamente compleja, ya que el tráfico malicioso es prácticamente indistinguible del tráfico normal. Existen dos tipos de inundación HTTP:

- Ataque HTTP GET: en este tipo se coordinan varios dispositivos para enviar múltiples solicitudes de imágenes, archivos u otro recurso. Cuando el objetivo se inunde con solicitudes / respuestas entrantes, se producirá la denegación de servicio.
- Ataque HTTP POST: Se enviarán solicitudes de tipo POST, por ejemplo, enviar un formulario en un sitio web. El procesamiento de manejar los datos del formulario y ejecutar comandos para introducirlos en una base de datos es más intenso que ancho de banda y procesamiento para enviar la solicitud POST. Por lo que este ataque coordinará la inundación de la red con el gasto de recursos interno del dispositivo.

Como ya hemos dicho anteriormente, este tipo de ataques son complejos. Una de las formas de evitarlo sería denegando el servicio una vez se sospeche del inicio del ataque a todas las siguientes conexiones, tanto atacantes como legítimas. Una solución en el ataque de tipo HTTP POST sería implementando un captcha para probar si se trata de un bot o no.

2.6.3 SYN flood

Una inundación SYN se produce al enviar repetidamente paquetes de solicitud de conexión SYN, el atacante puede llegar a desbordar todos los puertos de la máquina del servidor destino, haciendo que responda de manera lenta o deje de responder. Este tipo de ataque se utiliza debilidades de la capa 3 y 4 (16).

Estos ataques funcionan explotando el proceso handshake de una conexión TCP. Para establecer una conexión TCP, primero el cliente envía un paquete SYN para iniciar la conexión; segundo, el servidor responde a ese paquete con un paquete SYN / ACK para reconocer la información y finalmente, el cliente devuelve un paquete ACK para confirmar la recepción del paquete del servidor. Una vez completos estos 3 pasos, la conexión TCP está abierta y se pueden enviar y recibir datos.

Para producir una denegación de servicios, se explotará este funcionamiento, primero, enviará un gran volumen de paquetes SYN al servidor destino, a menudo con IP falsas; segundo, el servidor responderá a cada una de las solicitudes y dejará un puerto abierto listo para recibir la respuesta y por último, mientras el servidor espera el paquete ACK final, que nunca llega, el atacante continúa enviando más paquetes SYN. Cada paquete hace que el servidor mantenga temporalmente un puerto abierto y una vez se han utilizado todos los puertos disponibles, el servidor no podrá funcionar correctamente. Esto provoca que la conexión esté semiabierta.

En este tipo de ataques, el servidor objetivo deja continuamente conexiones abiertas y espera a que cada conexión expire antes que los puertos vuelvan a estar disponibles.
Una inundación SYN puede ocurrir de tres maneras diferentes:

- Ataque directo: Una inundación SYN donde la IP no se falsifica, se considera ataque directo. El atacante solo usa un dispositivo con una dirección IP real para crear el ataque, por lo que el atacante es muy vulnerable al descubrimiento y es fácil de mitigar. Para crear el estado de conexión semiabierta, el atacante evita que

su dispositivo responda a los paquetes SYN-ACK del servidor. Este se puede lograr mediante reglas de firewall. Este método se puede combinar con una botnet, ya que en este caso al atacante no le importará camuflar la IP del dispositivo.

- Ataque falso: El atacante falsifica su dirección IP en cada paquete SYN que envía para inhibir los esfuerzos de mitigación y hacer que su identidad sea más difícil de descubrir. A pesar de ello, mitigarlo es difícil pero no imposible, puesto que los paquetes pueden rastrearse hasta su origen.
- Ataque distribuido: Si se crea un ataque con botnet, la probabilidad de rastrear el ataque hasta su origen es baja. Para un nivel de ofuscación adicional, el atacante puede falsificar la IP de los dispositivos de la botnet.

El ataque de tipo SYN flood se puede mitigar mediante:

- Aumento de la cantidad máxima de conexiones medio abiertas que permitirá el sistema operativo y reservar recursos de memoria adicionales para hacer frente a las nuevas solicitudes. Si el sistema no tiene suficiente memoria para poder manejar el tamaño de la cola de espera, el sistema se ralentizará, pero puede ser mejor que evitar el colapso del sistema.
- Sobreescritura de la conexión semiabierta más antigua. Esto requiere que las conexiones legítimas puedan establecerse completamente en menos tiempo que un atacante llene el sistema con paquetes SYN maliciosos. Esta defensa es efectiva ante ataques de poco volumen.
- Mediante la creación de cookies por parte del servidor. El servidor responde a cada solicitud con un paquete SYN-ACK pero luego descarta la solicitud SYN de la acumulación, eliminando la solicitud de memoria y dejando el puerto abierto y listo para una nueva conexión. Este sistema pierde cierta información sobre la conexión TCP, pero mantiene el sistema sin dejar de dar servicio a los usuarios legítimos.

2.6.4 UDP flood

Una inundación UDP es un tipo de ataque en el que se envía una gran cantidad de paquetes UDP con el objetivo de bloquear la capacidad del defensor para procesar y responder. El cortafuegos que protege el servidor de destino también puede abrumarse como resultado de la inundación de paquetes UDP (16).

Una inundación UDP funciona principalmente explotando los pasos que un servidor toma cuando responde a un paquete UDP enviado a uno de sus puertos. Cuando un servidor recibe un paquete UDP se siguen dos pasos de respuesta, primero el servidor verifica si se están ejecutando programas que estén escuchando solicitudes en el puerto especificado y, si no hay programas en ese puerto, el servidor envía un paquete ICPM para informar que el destino no está disponible.

A medida que se recibe un paquete UDP, sigue los pasos para procesar la solicitud y utiliza los recursos del servidor en el proceso. Cuando se transmiten paquetes UDP, cada paquete incluirá la dirección IP del dispositivo fuente. Durante este tipo de ataques, un atacante no utilizará su IP real, si no una IP falsificada de origen, impidiendo la ubicación del atacante y saturando con mensajes de respuesta el servidor objetivo.

Debido a que los recursos del servidor pueden agotarse rápidamente cuando se recibe una gran cantidad de paquetes UDP, dará como resultado la denegación de servicio al tráfico normal.

Para mitigar este tipo de ataques, la mayoría de los sistemas operativos limitan la tasa de respuesta de paquetes ICMP, en parte también para interrumpir ataques de inundación ICMP. La desventaja de este tipo de mitigación es que, durante un ataque, pueden filtrarse también paquetes legítimos. Si la inundación UDP tiene un volumen lo suficientemente alto como para saturar la tabla de estado del servidor, cualquier mitigación será insuficiente ya que se producirán cuellos de botella.

2.6.5 Otros ataques DDOS

Otros ataques de tipo DDOS son:

- Ataque Memcached: El atacante suplanta las solicitudes a un servidor UDP memcached (sistema caché de base de datos para acelerar la red y sitios web) vulnerable, que luego inunda a una víctima específica con tráfico de internet, lo que puede saturar los recursos de la víctima (17).
- Ataque de amplificación NTP. Ataque basado en la disparidad en el costo del ancho de banda entre un atacante y el recurso web del objetivo. Cuando la disparidad en el costo se magnifica en muchas solicitudes, el tráfico resultante puede interrumpir la infraestructura de la red (18).
- Ataque de amplificación DNS. Ataque basado en la disparidad en el costo del ancho de banda entre un atacante y el recurso web del objetivo. Cuando la disparidad en el costo se magnifica en muchas solicitudes, el tráfico resultante puede interrumpir la infraestructura de la red. A diferencia de la amplificación NTP, los ataques de amplificación de DNS reflejan y amplifican el tráfico de servidores DNS no protegidos a fin de ocultar el origen del ataque y aumentar su efectividad (19).
- Ataque SSDP. Ataque que explota los protocolos de red de Universal Plug and Play (UPnP) para enviar una cantidad amplificada de tráfico a una víctima específica, abrumando la infraestructura del objetivo y desconectando su recurso web (20).
- Inundación DNS. El atacante inunda los servidores DNS de un dominio en un intento para interrumpir la resolución DNS para ese dominio. Al interrumpir la resolución DNS, se comprometerá la capacidad de un sitio web, API o aplicación web para responder al tráfico legítimo (21).
- Ataque low and slow. A diferencia del resto de ataques que se basan más en la fuerza bruta, estos requieren muy poco ancho de banda. Debido a que no requieren una gran cantidad de recursos para llevar a cabo, los ataques pueden iniciarse desde un único dispositivo (22).
- Ping de la muerte. Es uno de los ataques de red más antiguos. Los ordenadores más modernos ya no son vulnerables a este tipo de ataque. Consiste simplemente en crear un datagrama cuyo tamaño total supere el máximo autorizado (65536 bytes). Cuando un paquete se envía a un sistema que contiene esta vulnerabilidad, se produce la caída del sistema (23).

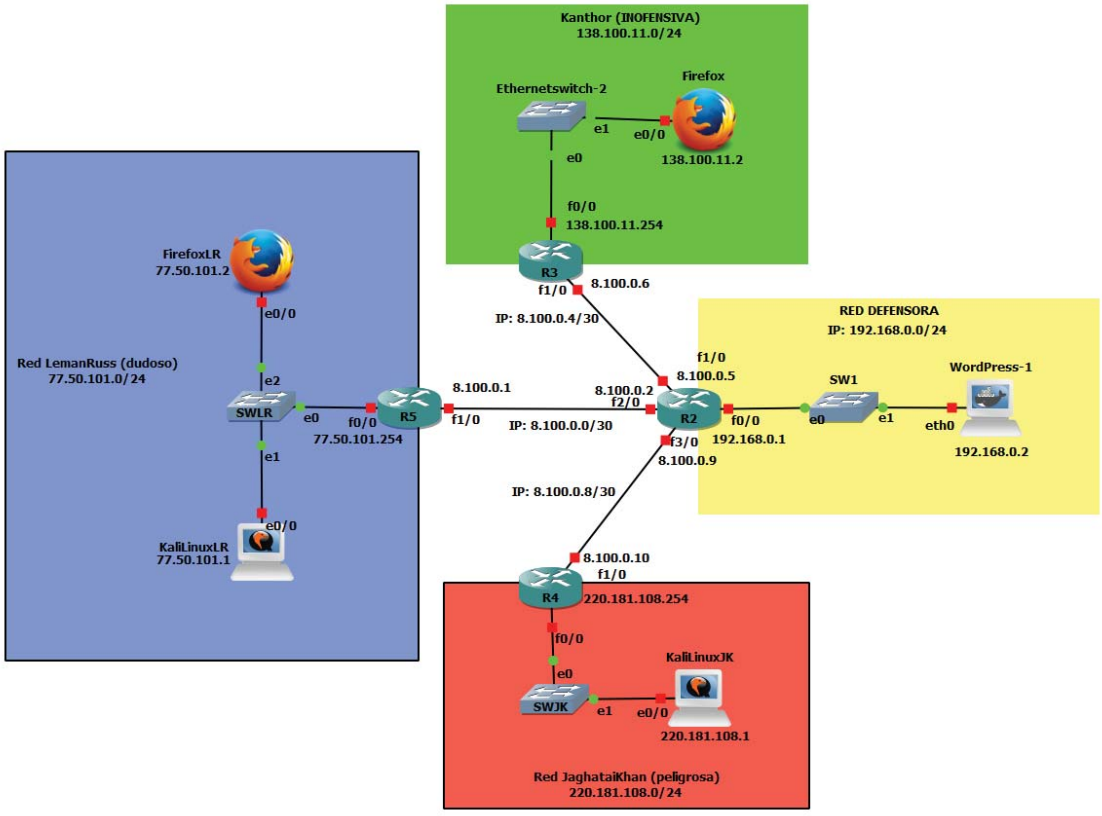
- Ataque pitufo. Se envían grandes cantidades de tráfico ICMP a la dirección de broadcast, todos ellos teniendo la dirección de origen de la víctima. Este ataque es fácil de mitigar, por ejemplo, en enrutadores CISCO se puede evitar este ataque con el comando "*no ip directed-broadcast*" (24).

Tras revisar el estado del arte de la técnica es posible apreciar que los ataques DDOS son un problema que requieren de soluciones innovadoras. Los sistemas multi-agente permiten establecer un enfoque distribuido y con capacidades avanzadas para la predicción. En el siguiente apartado se presenta el diseño del entorno propuesto para la definición del sistema multi-agente propuesto en este trabajo.

3. DISEÑO

3.1 Diseño de la red

Pensando en diferentes redes origen y su potencial peligro dependiendo de ello, se simularán 4 redes: una red basada en una IP de una universidad europea que llamaremos Kanthor (inofensivo), otra red basada en una IP de una zona que llamaremos LemanRuss, la cual pertenece a un país euroasiático (dudosa) y otra red, basada en una zona que llamaremos JaghataiKhan, la cual pertenece a un país asiático (peligrosa).



3.2 Metodología y herramientas usadas

Para el funcionamiento de la red, se han simulado en GNS3:

- Enroutador CISCO 7200. Enrutadores dirigidos a empresas, que serían las principales afectadas en un ataque DDOS.
- Firefox. Aplicación para tener un navegador Firefox en GNS3, desde estos dispositivos se harán las peticiones legítimas al servidor.
- Kali Linux. Sistema operativo basado en Debian GNU/Linux para la auditoría y seguridad informática en general. Éste nos servirá para simular ataques SYN Flood, HTTP Flood, ICMP Flood y UDP Flood.

En cuanto a la metodología para el desarrollo de sistemas multi-agente, la metodología propuesta por Nikraz (15) presenta fases de planificación, análisis, diseño, implementación y pruebas, limitadas por la funcionalidad de JADE. El diagrama de flujo de la metodología para desarrollo de sistemas multi-agente se presenta en la siguiente ilustración.

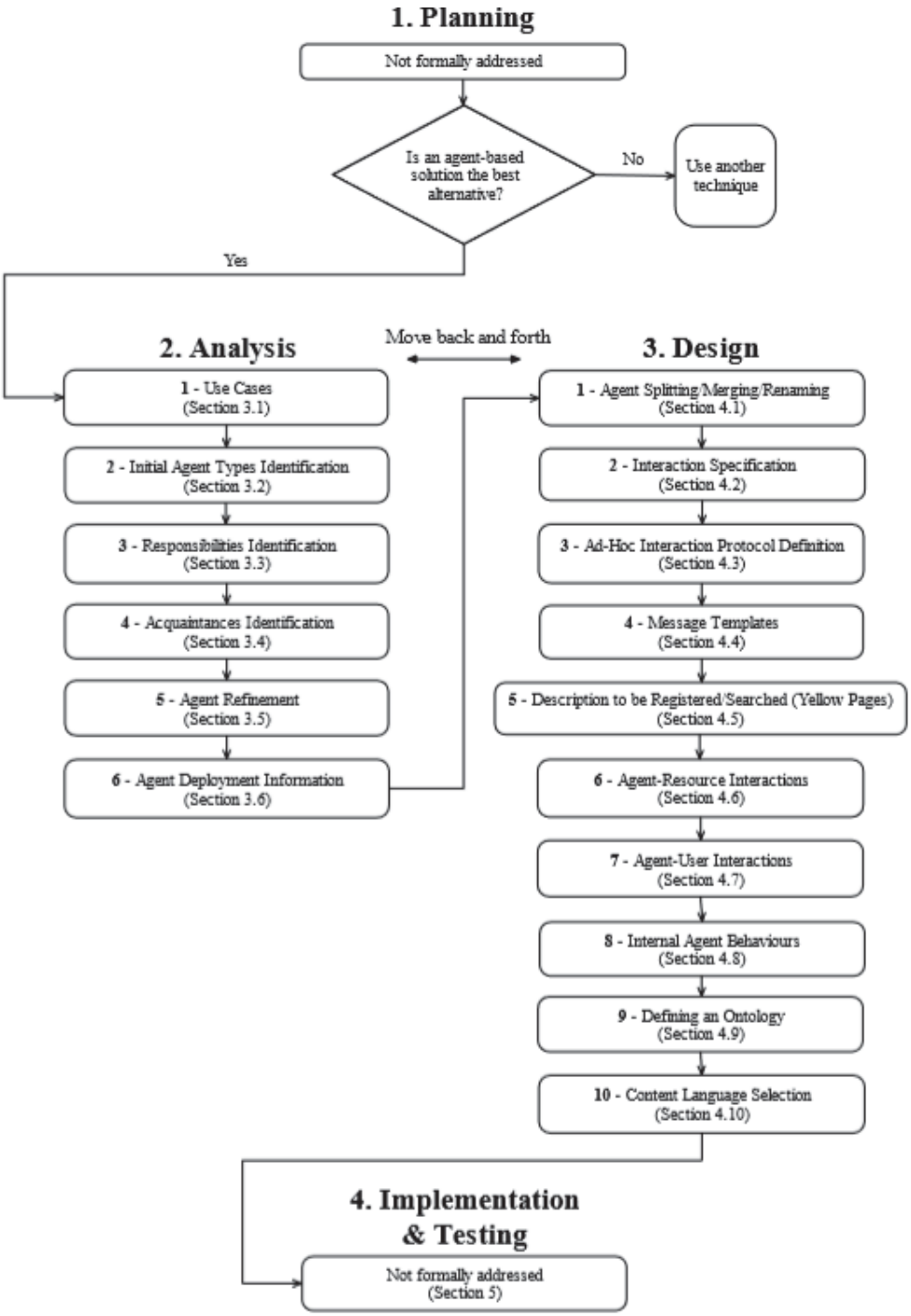


Ilustración 4. Metodología de análisis y diseño de Sistemas Multi-Agente. Nikraz, M (15)

En lo que respecta a los agentes, se ha usado la herramienta JADE, un framework de JAVA y para el aprendizaje automático, se hará uso de la aplicación WEKA.

3.3 Planificación

La fase de planificación, planificación y pruebas no son tratadas en la metodología de Nikraz, sin embargo, se incluyen en una decisión en el diagrama de flujo de la metodología, el cual indica si el desarrollador decide hacer uso de una solución basada en

agentes o no. Si no se hace uso de un sistema agente, se debe buscar una solución alternativa, en caso contrario, se seguirá a la fase de análisis.

En este trabajo, la solución propuesta contribuye a la prestación de un servicio de prevención DDOS a un caso particular y, como vimos en el punto 1, el sistema multi-agente en estos casos serían la mejor opción.

3.4 Análisis

En la etapa de análisis, se pretende construir una solución basada en agentes. Debe incluir casos de uso, de agentes y sus responsabilidades, interacciones y despliegue.

3.4.1 Casos de uso

En este punto se definen los casos de uso, que permiten determinar los requerimientos funcionales del sistema.

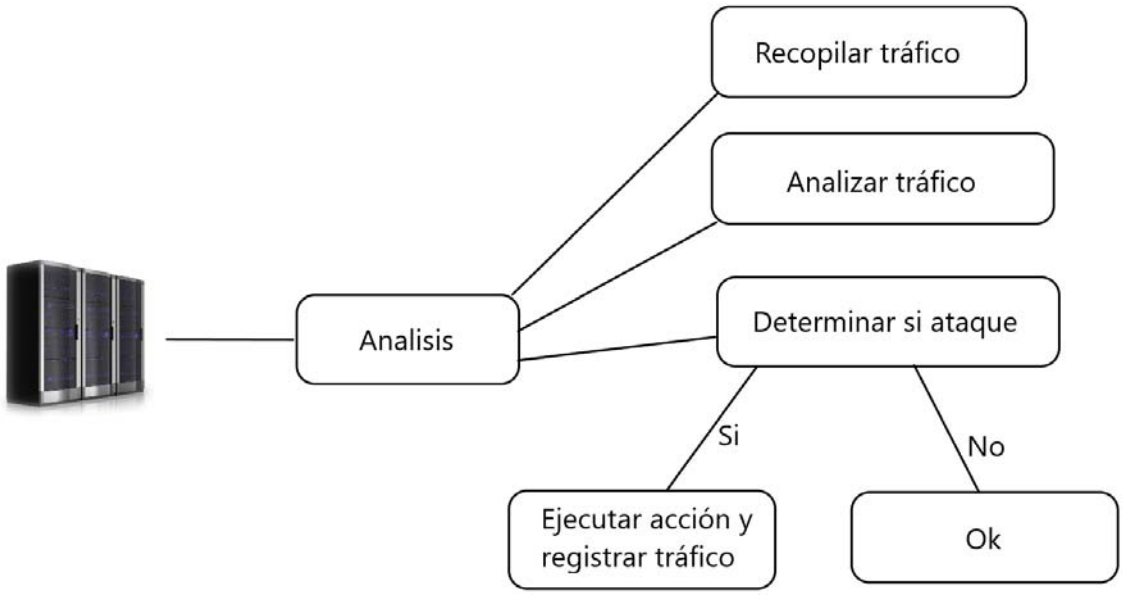


Ilustración 5. Casos de uso del sistema.

Para saber si ha habido ataque o no, se recopila el tráfico, se analiza el tráfico y el experto en seguridad determinará si ha habido ataque o no. En el caso que no haya habido ataque, se descarta la información y se continúa el tráfico normalmente. Si hay un ataque, se ejecuta una acción determinada y se registrará el tráfico web que provocó el ataque.

3.4.2 Identificación inicial de los tipos de agentes

En este segundo paso, se identifican los agentes asociados a usuarios, así como agentes por cada uno de los recursos utilizados. Los usuarios podrían ser solamente uno (el ingeniero de seguridad), pero se ha decidido crear un agente por cada evento que tiene el caso de uso. En la siguiente ilustración se expondrá la identificación de agentes.

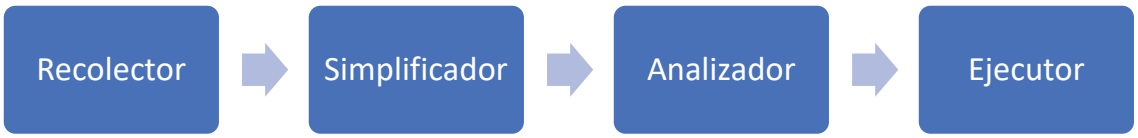


Ilustración 6. Identificación de agentes.

3.4.3 Identificación de responsabilidades

En este tercer paso, para cada agente identificado se hace una lista de sus principales responsabilidades. Deben derivar de los casos de usos definidos.

Agente	Responsabilidades
Recolector	Recolectan todo el tráfico de la web y la dividen individualmente, quedándose con los resultados adecuados.
Simplificador	Este agente es un agente doble, primero coge cada paquete individual y lo despedaza, guardando en una base de datos la información oportuna. Si el simplificador detecta algún movimiento anómalo, se lo comunicará al agente analizador. Si en un periodo de tiempo no ha habido ninguna alarma, descartará los elementos guardados en la base de datos
Analizador	Analiza los elementos de la base de datos y determina la posibilidad de que esté ocurriendo un ataque.
Ejecutor	Si hay un ataque, ejecutará las acciones pertinentes para mitigar el ataque de denegación de servicios, generando un log del ataque y las acciones ejecutadas.

Ilustración 7.Responsabilidades de los agentes.

3.4.4 Identificación de las interacciones

En el cuarto paso, se deben encontrar las interacciones que se presentan entre los agentes.

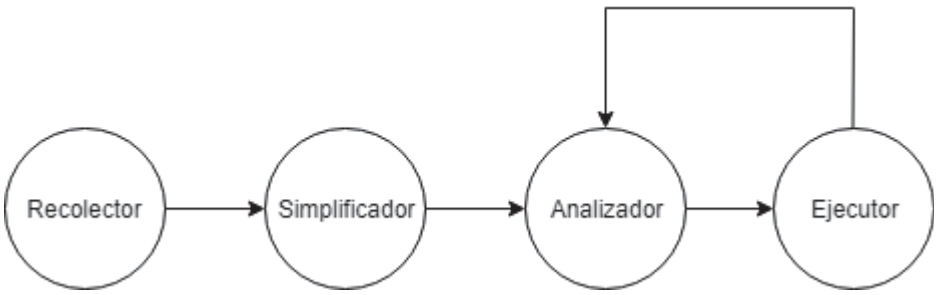


Ilustración 8. Diagrama de agentes con sus interacciones.

El recolector mandará los paquetes al simplificador, el simplificador, si detecta un movimiento anormal de paquetes, se lo mandará al analizador y el analizador determinará si hay ataque o no. Si hay ataque, se lo mandará al ejecutor, este actuará en consecuencia y mandará al analizador el resultado.

4. ESCENARIO

4.1 Análisis del escenario

Para este escenario, se ha propuesto una empresa especializada en maquinaria. Esta empresa, al ser especializada, recibirá poco tráfico comparada con grandes empresas. Al no tener encontrar ningún estudio sobre visitas web en empresas pequeñas, se usarán los datos de una empresa real que ha facilitado los datos de visitas. Al ser estos datos sensibles, la empresa permanecerá anónima.

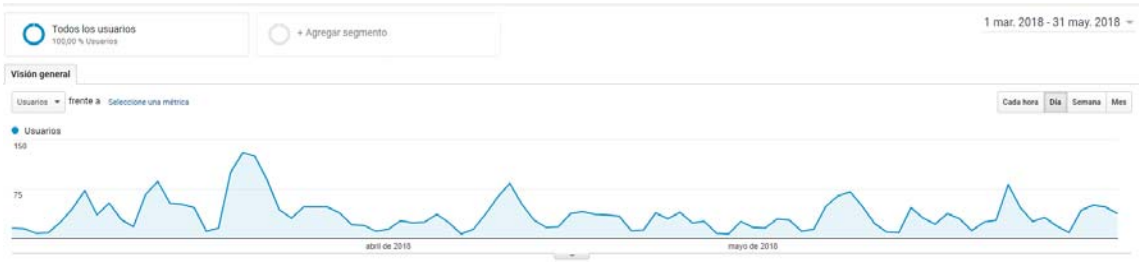


Ilustración 9. Google Analytics global de la empresa.

Esta empresa, en 3 meses, ha recibido una media de 35 usuarios diarios. El día que dicha empresa ha recibido más visitas son 131 usuarios.

Comparando horas en estos 3 meses, el pico de visitas han sido 20. Al ser una empresa internacional, que vende en países de Europa, Asia y Latinoamérica, recibe visitas diariamente a cualquier hora del día, como se observa ver a continuación en la siguiente ilustración.

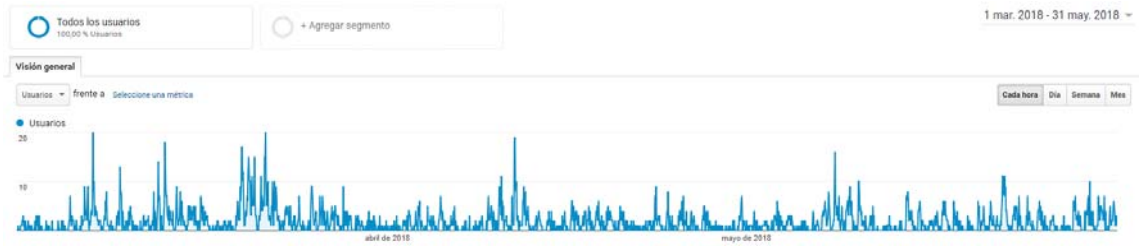


Ilustración 10. Google Analytics filtrando por horas.

Viendo estos datos, se establecerá un límite de 131 usuarios simultáneamente, sabiendo que la empresa, puntualmente, recibió dichas visitas en un día y siendo optimistas, podría ocurrir en un determinado momento.

En cuanto a la web simulada, nuestra web tiene una media de 42 paquetes TCP por conexión, hay que establecer límites para suponer un ataque DDOS (16):

- En el caso de los paquetes TCP, por dichos datos, se establecerá un límite de 5502 paquetes para generar una alarma al agente analizador.
- En el caso de los paquetes UDP, Cisco establece un límite de 1000 límite en algunos enrutadores por defecto (17), por lo que estableceremos un límite de 1000 paquetes UDP.
- En el caso de los paquetes ICMP, se elegirá el 5% del total de banda ancha disponible (18), cada paquete tendrá un peso de (19):

Longitud ICPMP = Encabezado Ethernet (14 bytes) + Encabezado IP (20 bytes) + Encabezado ICMP (8 bytes) + Tamaño de carga ICMP (32 bytes) = 74 bytes

Por lo que se establecerá un límite de 1562 paquetes ICMP, como veremos más adelante.

4.2 Ataques

4.2.1 Ataque ICMP

En un escenario normal, desde un solo terminal, se envían alrededor de 2-3 paquetes por segundo:

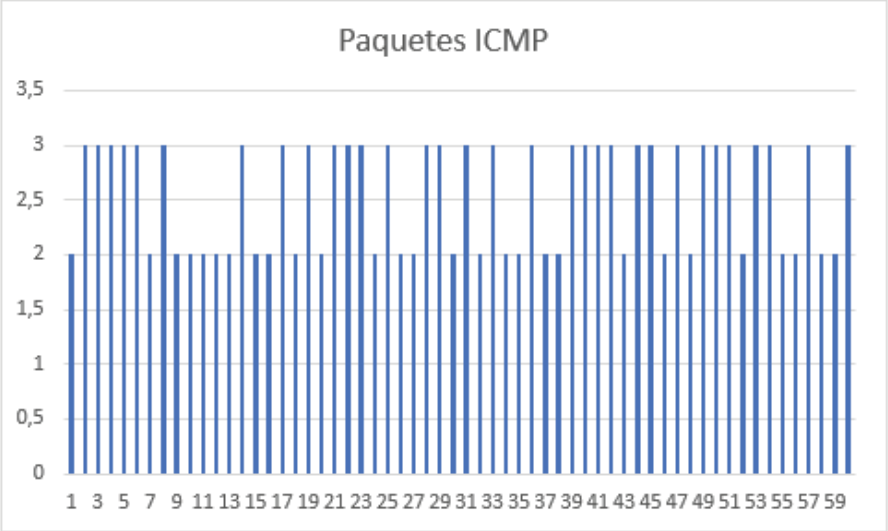


Ilustración 11. Tráfico ICMP en un escenario normal desde un único computador.

Sin embargo, en un escenario de ataque ICMP, ejecutado por el comando "hping3 --flood --rand-source --icmp -p %PUERTO% %IP%", el número de paquetes asciende a 800 paquetes los primeros instantes, a entre 1050 – 1200 paquetes en los siguientes instantes.

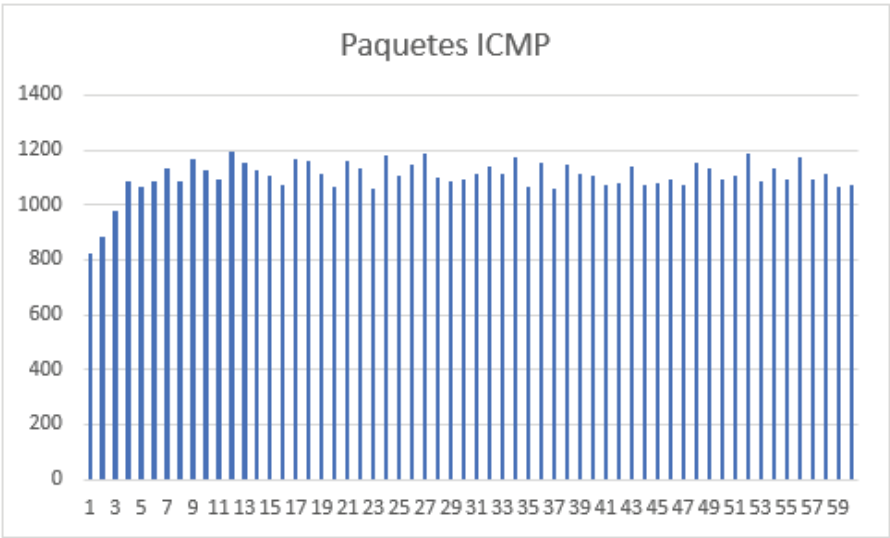


Ilustración 12.Tráfico ICMP en un escenario de ataque desde un único atacante.

Ante este ataque, se propondrá limitar el número de paquetes ICMP, se propone limitar los paquetes ICMP que el enrutador puede tratar tal y como proponen J. Udhayan y R. Anitha (18), al tratarse de un enrutador CISCO, se podrá hacer con el comando "ip icmp rate-limit", cuyo uso se describe a continuación.

ip icmp rate-limit unreachable [df] [ms] [**log** [packets] [interval-ms]]

Example:

```
Switch(config)# ip icmp
rate-limit unreachable df log
1100 12000
```

Specifies the rate limitation of ICMP unreachable destination messages and the error message log threshold for generating a message. The default is no unreachable messages are sent more often than once every half second.

The arguments and keywords are as follows:

- **df**—(Optional) When Don't Fragment (DF) bit is set in the ICMP header, a datagram cannot be fragmented. If the **df** keyword is not specified, all other types of destination unreachable messages are sent.
- **ms**—(Optional) Interval at which unreachable messages are generated. The valid range is from 1 ms to 4294967295 ms.
- **log**—(Optional) List of error messages. The arguments are as follows:
  - *packets*—(Optional) Number of packets that determine a threshold for generating a log. The default is 1000 packets.
  - *interval-ms*—(Optional) Time limit for an interval for which a logging message is triggered. The default is 60000 ms, which is 1 minute.

Note Counting begins as soon as this command is configured.

Ilustración 13. Instrucción para limitar el número de paquetes ICMP que se reciben.

Según Udhayan y Anitha (18), el número de paquetes ICMP permitidos para una red debe ser entre el 5% y el 1% de la banda ancha.

Capacity of LAN/ WAN/ MAN/ Internet	ICMP bandwidth for (5%) rate limit	Allowed ICMP Approx. bits per sec	Approx. Number of ICMP packets allowed
2Mbps	0.1Mbps	1,00,000	1562
4Mbps	0.2 Mbps	2,00,000	3125
10Mbps	0.5Mbps	5,00,000	7812
100Mbps	5Mbps	50,00,000	78125
1Gbps	50Mbps	500,00,000	781250
2Gbps	100Mbps	10,00,00,000	1562500
10Gbps	0.5Gbps	50,00,00,000	7812500
100Gbps	5Gbps	5,00,00,00,000	78125000

Ilustración 14. Trafico ICMP con el 5% de banda ancha.

Al tener un entorno simulado, se ha decidido establecer el límite en el mínimo de paquetes, 1562 para no sobrecargar el sistema.

La solución que se ha propuesto limitando el número de paquetes ICMP, por lo que no afectará en gran parte a usuarios legítimos esta limitación, y se genera un reporte ICMP emitido por el agente Ejecutor llamado "ICMPReport_%FECHA%.txt", que se verá más extensamente en el punto de "Resultados":

```
Reporte ICMP con fecha: %FECHA%
**************************************************
Acción a ejecutar:
ip icmp rate-limit unreachable df log 1100 12000

Listado de IPs detectadas:
%LISTAIP%

Nº total de paquetes:
%PaquetesTotales%

1º IP más frecuente: %IP1% con %NPaquetesIP1% paquetes y un (valueIP1*100)/PaquetesTotales %
2º IP más frecuente: %IP2% con %NPaquetesIP2% paquetes y un (valueIP1*100)/PaquetesTotales %
3º IP más frecuente: %IP2% con %NPaquetesIP3% paquetes y un (valueIP1*100)/PaquetesTotales %
```

Ilustración 15. Reporte preliminar generado por el agente Ejecutor en el caso de ataque ICMP.

4.2.2 Ataques UDP

En un escenario normal, desde un solo terminal, se envían, por diversos métodos, como por ejemplo “echo 'test' > /dev/udp/%IP%/%PUERTO%”, alrededor de 5-9 paquetes por segundo:

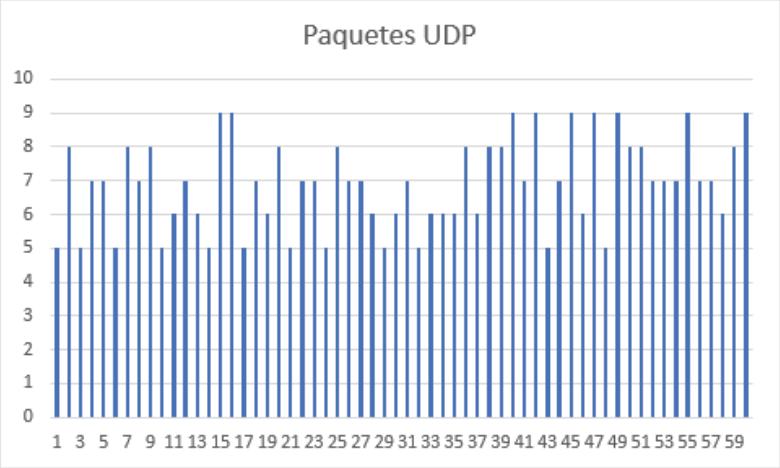


Ilustración 16. Paquetes UDP enviados en un escenario normal.

Sin embargo, en un escenario de ataque UDP, el número de paquetes asciende a entre 1050 – 2700 paquetes en los siguientes segundos, como se puede ver en la ilustración siguiente, ejecutado por el comando “hping3 --flood --rand-source --udp -p %PUERTO% %IP%”.

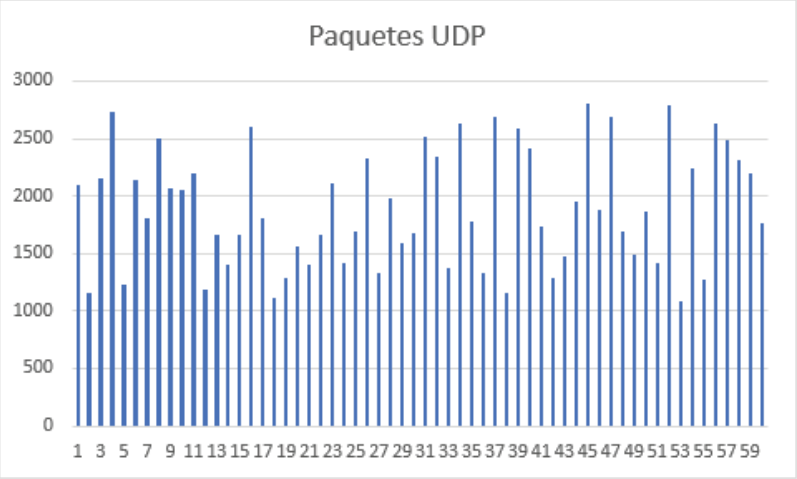


Ilustración 17. Paquetes UDP durante un ataque desde un solo computador.

Según Aarti Singh y Dimple Juneja (20), la mejor opción es bloquear o limitar las comunicaciones, en este caso, al no hacer uso de UDP en el escenario, optaremos por bloquear las comunicaciones UDP en el caso que se produzca un ataque. Se generará un reporte UDP emitido por el agente ejecutor llamado "UDPReport_%FECHA%.txt", que se verá más extensamente en el punto de "Resultados":

```
Reporte UDP con fecha: %FECHA%
******************************************************
Acción a ejecutar:
mls qos
access-list 150 permit udp any any
class-map UDP_Traffic
  match access 150
policy-map Control
 class UDP_Traffic
  police 10000000 60000 exceed-action drop
int fa1/0
 service-policy input Control
int fa2/0
 service-policy input Control
int fa3/0
 service-policy input Control

Listado de IPs detectadas:
%LISTAIP%

N° total de paquetes:
%PaquetesTotales%

1° IP más frecuente: %IP1% con %NPaquetesIP1% paquetes y un (valueIP1*100)/PaquetesTotales %
2° IP más frecuente: %IP2% con %NPaquetesIP2% paquetes y un (valueIP1*100)/PaquetesTotales %
3° IP más frecuente: %IP2% con %NPaquetesIP3% paquetes y un (valueIP1*100)/PaquetesTotales %
```

Ilustración 18. Reporte preliminar generado por el agente Ejecutor en el caso de ataque UDP.

4.2.3 Ataques TCP

En este caso, se podrán distinguir varios tipos de ataques TCP, TCP SYN flood y HTTP flood.

En primer lugar, en un escenario normal, desde un solo cliente, se envían alrededor de 50-70 paquetes TCP por segundo:

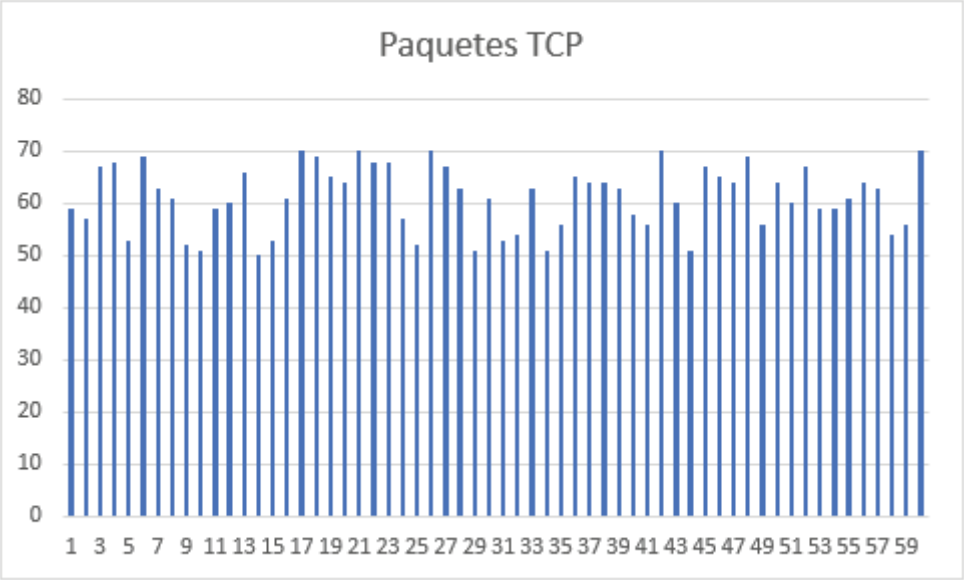


Ilustración 19. Paquetes TCP enviados en un escenario normal.

A continuación, se verá los paquetes TCP que se generarán durante diversos métodos de ataque TCP.

4.2.3.1 TCP SYN Flood

Durante un ataque TCP SYN Flood, el número de paquetes TCP asciende a 1400 – 2000 por segundo, ejecutado por el comando "hping3 -S --flood -V %IP%".

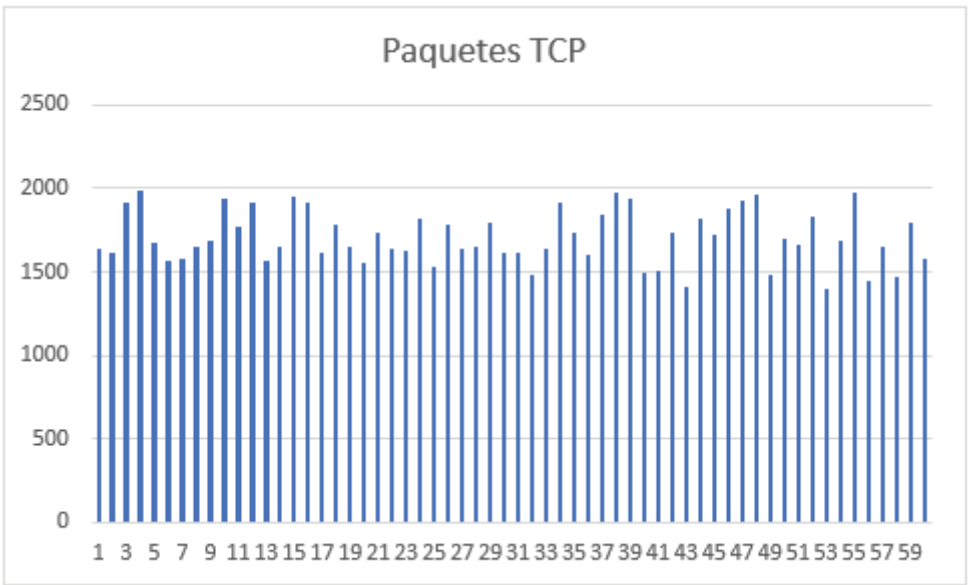


Ilustración 20. Paquetes TCP enviados en un escenario de ataque TCP SYN Flood.

4.2.3.2 TCP flood

Durante un ataque TCP Flood, el número de paquetes TCP asciende a 2400 – 4300 por segundo, ejecutado por el comando “nping --tcp-connect -rate=90000 -c 900000 -q %IP%”.

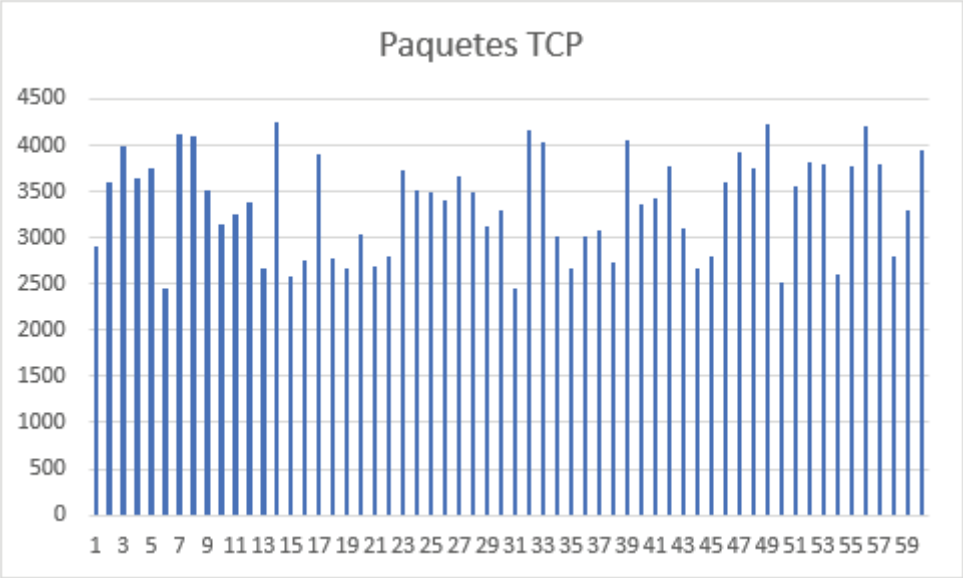


Ilustración 21. Paquetes TCP enviados en un escenario de ataque TCP.

4.2.3.3 Prevención

Según Wesley M. Eddy (21), para prevenir los ataques TCP SYN flood, la caché SYN parece ser el mejor mecanismo de defensa disponible, puesto que el resto son ineficaces. Al usar cookies TCP, un host concatena algunos bits secretos locales, una estructura de datos que contiene la direcciones IP y los puertos TCP, el número de secuencia SYN inicial y algunos datos que identifican los bits secretos. Se calcula un MD5 en todos estos bytes y algunos bits se truncan desde el valor hash en el que se colocará la secuencia de números SYN-ACK. En general se utilizan, al menos, 3 bytes de los bits hash, lo que significa que todavía hay cerca de 2^24 bytes de esfuerzo requerido para adivinar una cookie válida sin conocer los bits secretos locales.

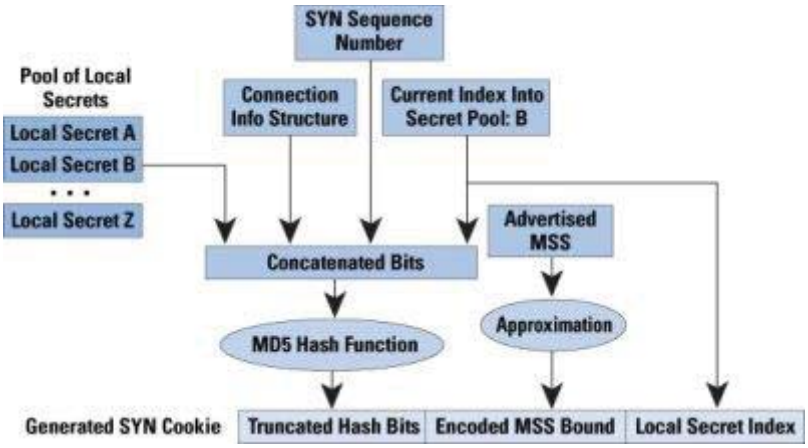


Ilustración 22. Proceso de generación y validación de cookies TCP-SYN.

En el resto de los casos, es difícil de solucionar, la mejor de las opciones es el bloqueo de comunicaciones entre el posible atacante y el defensor. Si un servidor recibe miles de conexiones de la misma IP solicitando el mismo recurso dentro de un periodo corto de tiempo, se insinúa que no es algo legítimo.

Al usar enrutadores CISCO, la solución propuesta ha sido, en cuanto a los ataques TCP SYN, usar la guía "Referencia de comandos de seguridad de CISCO IOS", más específicamente, el capítulo "comandos de intercepción de TCP" (22). De este modo, las solicitudes de conexión pueden pasar por el enrutador hasta el servidor, pero se vigilan hasta que se establezcan. Si no logran establecerse en 30 segundos, se envía un restablecimiento al servidor para aclarar su estado.

En cuanto al resto de ataques TCP, se cortarán las comunicaciones del host atacante, a no ser que el atacante genere IP aleatorias, en este caso se deberá tratar de otra forma.

Al igual que con ICMP y UDP, el agente generará un log con las acciones tomadas y la información necesaria.

```
Reporte TCP con fecha: %FECHA%

Tipo: SYN Flood

Acción a ejecutar:

ip tcp intercept mode intercept
ip tcp intercept watch-timeout 15

Listado de IPs detectadas:
%LISTAIP%

N° total de paquetes:
%PaquetesTotales%

1° IP más frecuente: %IP1 con %NPaquetesIP1% paquetes y un (valueIP1*100)/PaquetesTotales %
2° IP más frecuente: %IP1 con %NPaquetesIP2% paquetes y un (valueIP2*100)/PaquetesTotales %
3° IP más frecuente: %IP1 con %NPaquetesIP3% paquetes y un (valueIP3*100)/PaquetesTotales %
```

Ilustración 23. Reporte preliminar generado por un ataque SYN Flood.

```
Reporte TCP con fecha: %FECHA%

Tipo: HTTP Flood

Acción a ejecutar:

access-list 101 deny ip any host %IP%
access-list 101 permit ip any any
interface fa1-3/0
ip access-group 101 in

Listado de IPs detectadas:
%LISTAIP%

N° total de paquetes:
%PaquetesTotales%

1° IP más frecuente: %IP1 con %NPaquetesIP1% paquetes y un (valueIP1*100)/PaquetesTotales %
2° IP más frecuente: %IP1 con %NPaquetesIP2% paquetes y un (valueIP2*100)/PaquetesTotales %
3° IP más frecuente: %IP1 con %NPaquetesIP3% paquetes y un (valueIP3*100)/PaquetesTotales %
```

Ilustración 24. Reporte preliminar generado por un ataque TCP Flood.

4.3 Agentes

Un total de 4 agentes han sido desplegados, tal y como se ha indicado anteriormente en la ilustración 8.

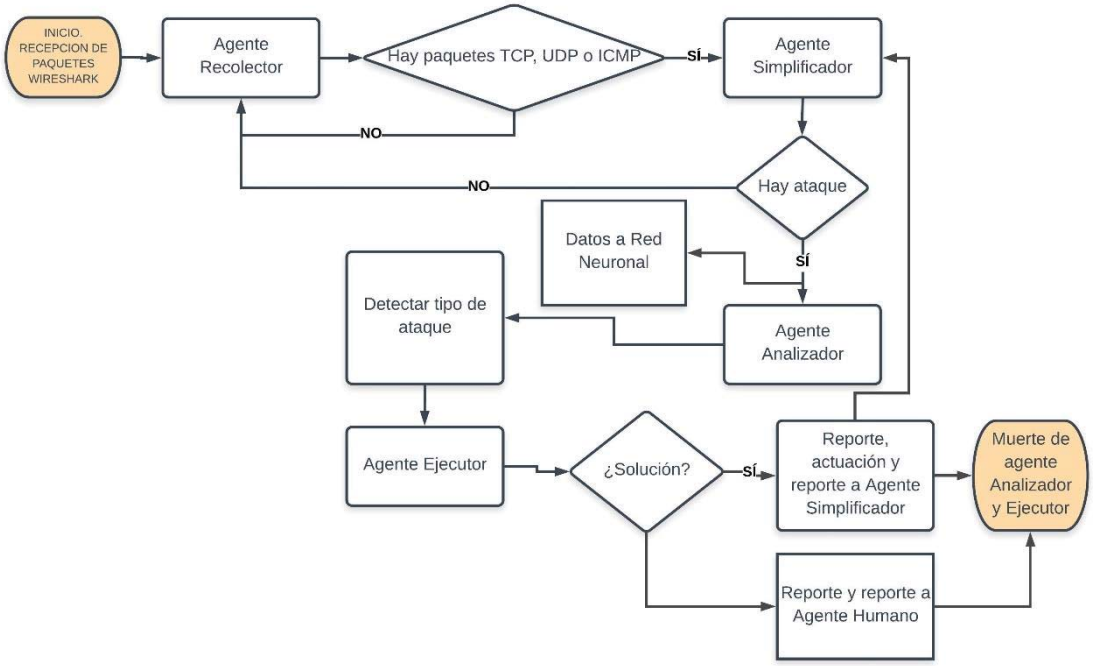


Ilustración 25. Diagrama de flujo sobre las interacciones de los agentes entre ellos.

4.3.1 Recolector

Este agente recogerá los ficheros más antiguos Wireshark y pasará cada paquete de interés a JSON gracias a la librería GSON, para su fácil manipulación posterior. El agente será de tipo TickerBehaviour, ya que permite definir un comportamiento cíclico que se ejecutará periódicamente con una tarea (en nuestro caso, cada segundo). Si no hay archivos Wireshark, dormirá durante 2 segundos antes de volver a ponerse en ejecución.

Este recolector tendrá un filtro para recoger los ficheros ICMP, TCP y UDP y IP destino las 3 interfaces de la red defensora.

```
String PCAP_FILE_KEY = RecolectorBehaviour.class.getName() + ".pcapFile";
String PCAP_FILE = System.getProperty(PCAP_FILE_KEY, ficheroAntiguo.getPath());
Gson gson = new GsonBuilder().setPrettyPrinting().create();

try {
    handle = Pcaps.openOffline(PCAP_FILE, TimestampPrecision.NANO);
    handle.setFilter("(icmp or tcp or udp) && (dst host 8.100.0.2 || dst host 8.100.0.9 || dst host 8.100.0.5)", BpfProgram.BpfCompileMode.OPTIMIZE);
} catch (PcapNativeException e) {
    handle = Pcaps.openOffline(PCAP_FILE);
    handle.setFilter("(icmp or tcp or udp) && (dst host 8.100.0.2 || dst host 8.100.0.9 || dst host 8.100.0.5)", BpfProgram.BpfCompileMode.OPTIMIZE);
}

for(int j=0; j<=5502; j++) {
    try {
        Packet packet = handle.getNextPacketEx();
        Writer writer = new FileWriter(JSON_DIR + dateFormat.format(date) +".json");
        handle.getTimestampPrecision();
        System.out.println(packet);
        gson.toJson(packet, writer);
        writer.close();
    } catch (TimeoutException e) {
        System.out.println("Recolector termina");
    } catch (EOFException e) {
        System.out.println("Recolector termina con archivo Wireshark");
        break;
    }
}
handle.close();
ficheroAntiguo.delete();
return true;
```

Ilustración 26. Código de recolección de ficheros Wireshark y su paso a JSON.

4.3.2 Simplificador

Agente de tipo CyclicBehaviour, que representa un comportamiento que debe ejecutarse una serie de veces, en este caso, será de infinitas veces.

Este agente recogerá los archivos JSON creados por el agente Recolector y los guardará en una base de datos. Así mismo, el agente posee unos paquetes "límites", establecidos en el escenario indicado. Si el número límite se excede, se creará un nuevo agente, llamado Analizador. Si en 1 minuto no se ha excedido el límite, se borrarán los datos guardados en la base de datos puesto que no habrá indicaciones de peligro.

```
if(paquetesICMP >= 1562) {
    avisoICMP = true;
    String [] args = new String[1];
    args[0] = "ICMP";
    AgentContainer c = getContainerController();
    try {
        AgentController a = c.createNewAgent(AnalizadorAgent.NICKNAME+Math.random()*100, AnalizadorAgent.class.getName(), args);
        a.start();
    }catch (Exception e){}
    System.out.println("AVISO ICMP");
}else if (paquetesTCP >= 5502) {
    avisoTCP = true;
    String [] args = new String[1];
    args[0] = "TCP";
    AgentContainer c = getContainerController();
    try {
        AgentController a = c.createNewAgent(AnalizadorAgent.NICKNAME+Math.random()*100, AnalizadorAgent.class.getName(), args);
        a.start();
    }catch (Exception e){}
    System.out.println("AVISO TCP");
}else if (paquetesUDP >= 1000) {
    avisoUDP = true;
    String [] args = new String[1];
    args[0] = "UDP";
    AgentContainer c = getContainerController();
    try {
        AgentController a = c.createNewAgent(AnalizadorAgent.NICKNAME+Math.random()*100, AnalizadorAgent.class.getName(), args);
        a.start();
    }catch (Exception e){}
    System.out.println("AVISO UDP");
```

Ilustración 27. Establecimiento de límites y creación dinámica de agentes analizadores.

4.3.3 Analizador

Este agente se creará de manera dinámica, será de tipo OneShotBehaviour. Este tipo de comportamiento siempre devuelve true, se ejecutará solo una vez y de forma ininterrumpida. Una vez haya cumplido su función, el agente "morirá".

El analizador creará un nuevo agente Ejecutor de manera dinámica de la misma forma que el agente Simplificador, pasando como argumento el tipo de ataque.

Este agente se encargará básicamente de crear agentes de tipo Ejecutor y de analizar los paquetes TCP recibidos, para saber a qué tipo de ataque se enfrenta el entorno y reaccionar.

4.3.4 Ejecutor

Al igual que el agente analizador, será un agente dinámico de tipo OneShotBehaviour, de forma que solo se ejecutará una vez y finalizará su ejecución.

Este agente ejecutor actuará de la manera más oportuna para parar un ataque. Finalmente, generará un reporte del ataque y avisará al agente Simplificador para que siga capturando paquetes de un tipo determinado.

4.3.5 Agent Base

Método auxiliar que ayuda a registrar y eliminar agentes, indicando al crearse, los parámetros nuevos, el nombre del agente.

5. PRUEBAS Y RESULTADOS

En este apartado se presentan las simulaciones realizadas para validar el sistema propuesto, así como los resultados obtenidos. Se realizan tres tipos de simulaciones: Ataques ICMP, ataques UDP y ataques TCP.

5.1 Simulación de ataque ICMP

En esta primera simulación, se ejecutarán 2 pruebas, una sin falsificar la IP y otra, con la IP falsificada.

Primero, se ha aumentado el número de paquetes ICMP por defecto que utiliza CISCO (1 paquete por cada 500ms)

En el primer caso, se han seguido recibiendo la cantidad de paquetes, pero las respuestas se limitaron a 1 por cada segundo. El agente tarda de media 85 segundos en detectar el ataque DDOS y en actuar

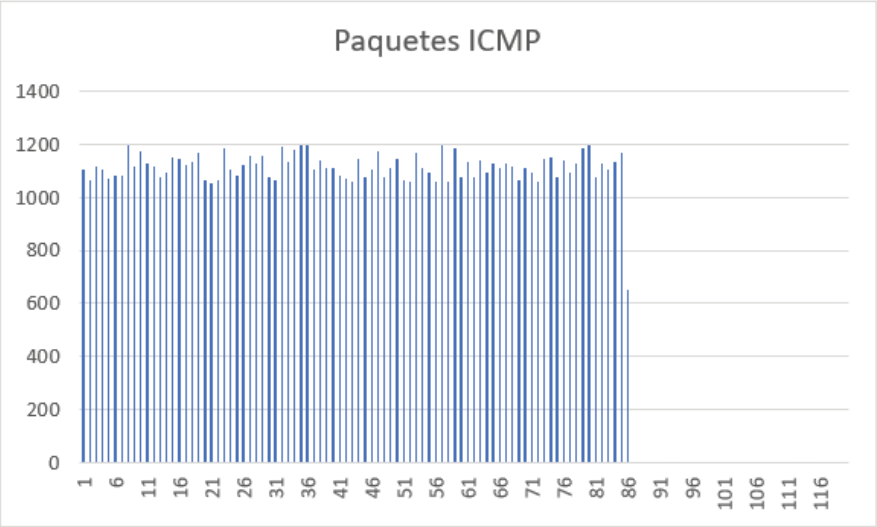


Ilustración 28. Ataque ICMP y mitigación (primer caso).

Tal y como aconseja CISCO (23), se ha indicado al agente el uso del comando “no ip unreachables” para evitar la generación de mensajes que podrían tener un efecto de incremento de CPU del enrutador. El log, después del ataque, es el siguiente:

```
Reporte ICMP con fecha: 2018-07-01_04-23-18

*********************************************************

Acción a ejecutar:
no ip unreachables
ip icmp rate-limit unreachable 1000

Listado de IPs detectadas:
220.181.108.1

Número total de paquetes:
1562

1º IP más frecuente: 220.181.108.1 con 1000 paquetes y un 100%
```

Ilustración 29. Log del ataque ICMP (primer caso).

En el segundo caso, ha sido prácticamente igual que el primer caso:

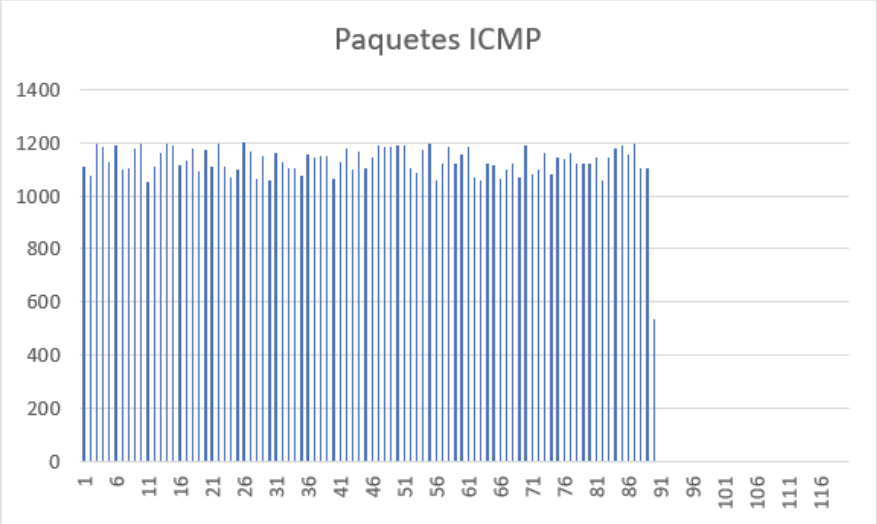


Ilustración 30. Ataques ICMP y mitigación (segundo caso).

En este caso, el agente recolector tiene un límite de recoger 5502 paquetes por segundo (el número máximo de paquetes que se establece en el entorno, en este caso, por el límite TCP) y se generaron 5502 IP aleatorias, al tener una cuota establecida en 131 usuarios simultáneos, se ha obviado este dato y, por lo tanto, el log generará, en este caso, el siguiente reporte:

```
Reporte ICMP con fecha: 2018-07-01_18-08-25

*********************************************************

Acción a ejecutar:
no ip unreachables
ip icmp rate-limit unreachable 1000

Número de IPs detectadas:
1562

Número total de paquetes:
1562
```

Ilustración 31. Log del ataque ICMP (segundo caso).

5.2 Simulación de ataque UDP

En esta segunda simulación, el agente ha tardado de media unos 70 segundos en reaccionar y ejecutar las acciones necesarias. Se han usado políticas de supervisión de tráfico de CISCO (24) para controlar los paquetes UDP que, desde el momento detectado, se desechan. En este caso, al no necesitarlo, se desecharán todos los paquetes.

Al igual que con el anterior caso, se generarán 2 logs diferentes, en el caso que el atacante haya suplantado la IP y otro en el caso que mantenga la misma IP.

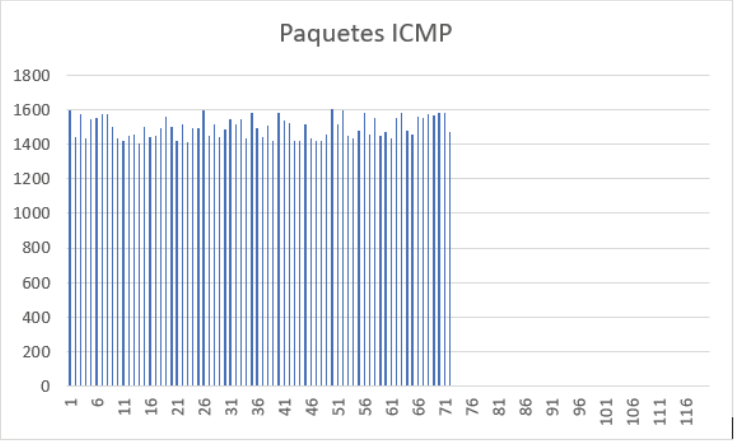


Ilustración 32. Paquetes UDP generados durante el ataque.

```
Reporte UDP con fecha: 2018-07-01_19-27-13

**********************************************************

Acción a ejecutar:
access-list 150 permit udp any any
class-map UDP_Traffic
match access 150
policy-map Control
class UDP_Traffic
police 10000 1500 conform-action drop exceed-action drop
int fa1/0
service-policy input Control
int fa2/0
service-policy input Control
int fa3/0
service-policy input Control

Listado de IPs detectadas:
220.181.108.1

Número total de paquetes:
1000

1º IP más frecuente: 220.181.108.1 con 1000 paquetes y un 100%
```

Ilustración 33. Reporte del agente Ejecutor al detectar un ataque UDP.

5.3 Simulación de ataque TCP

En esta última simulación, los agentes deberán ser más cuidadosos, pues se deberá detectar que tipo de ataque se está produciendo.

5.3.1 SYN Flood

En este caso, según Mehdi Ebady y Angela Amphawan (25), hay varios métodos de detectar un ataque SYN-Flood. Para este trabajo, se ha elegido una técnica basada en la tasa de llegada de paquetes TCP con el bit SYN activo.

Este método tiene ciertas desventajas, puesto que no hay un valor umbral explícito para determinar el tráfico normal y es difícil determinar este valor en una tasa baja debido a falsos positivos y falsos negativos. Además, calcular una estimación es difícil ya que el tráfico de red tiene características lineales y de ráfaga.

Por otro lado, el método posee ciertas ventajas, ya que en una simulación se puede controlar la medición estadística en el valor umbral, refleja el comportamiento del flujo del paquete y como se pueden medir los valores falsos positivos y falsos negativos y, por último, el valor umbral está determinado.

Es por ello por lo que este método que vamos a implementar tendrá una alta tasa de falsos positivos y negativos, consumo de memoria y tiempo de CPU, pero también tendrá una alta tasa de detección de ataques de SYN de tráfico alto.

En primer lugar, el número de paquetes que se reciben no disminuyen, sin embargo, se cortará antes la comunicación de establecimiento de conexión, para que el enrutador no se quede sin puertos.

Para establecer el límite, lo primero será observar de nuevo nuestros datos de Google Analytics.

Sesiones ? ↓	Duración media de la sesión ?
46 (1,19 %)	00:00:44
42 (1,09 %)	00:00:48
40 (1,03 %)	00:05:13
38 (0,98 %)	00:02:50
38 (0,98 %)	00:00:28
28 (0,72 %)	00:00:20
23 (0,59 %)	00:02:12
22 (0,57 %)	00:01:21
20 (0,52 %)	00:01:09
20 (0,52 %)	00:01:46

Ilustración 34. Duración media de cada sesión en el escenario real.

Calculando el tiempo medio de cada usuario este último mes, se tendrá una duración media de 1 minuto y 40 segundos por sesión. Si tenemos un límite de 131 usuarios, en 2 minutos tendremos aproximadamente 200 peticiones SYN por cada 2 minutos.

Para asegurar fallos de establecimientos de conexión y tener un margen, el límite será de 1000 paquetes con SYN ACK activo para determinar que se trata de un ataque SYN ACK, por lo que se detectará un ataque DDOS si se excede el 18% del límite con paquetes con el bit activo.

```
Reporte TCP con fecha: 2018-07-01_22-23-13

**********************************************************

Acción a ejecutar:
access-list 171 permit ip any any
ip tcp intercept watch-timeout 15

Listado de IPs detectadas:
77.50.101.0
138.100.11.2
220.181.108.0

Número total de paquetes:
5502

1º IP más frecuente: 220.181.108.0 con 3442 paquetes y un 63 %
2º IP más frecuente: 77.50.101.0 con 2013 paquetes y un 37 %
3º IP más frecuente: 138.100.11.2 con 47 paquetes y un 0 %
```

Ilustración 35. Reporte generado por el agente Ejecutor ante un ataque SYN Flood.

5.3.2 TCP Flood

En cuanto a otro ataque TCP, se ha optado finalmente por el uso de rate-limit (26), el agente generará un reporte con la siguiente información:

```
Reporte TCP con fecha: 2018-07-01_22-59-50

**********************************************************

Acción a ejecutar:
access-list 171 permit ip any any
interface fa1/0
rate-limit input access-group rate-limit 200 8000 8000 8000 conform-action set-mpls-exp-imposition-transmit 4 exceed-action set-mpls-exp-imposition-transmit 0
interface fa2/0
rate-limit input access-group rate-limit 200 8000 8000 8000 conform-action set-mpls-exp-imposition-transmit 4 exceed-action set-mpls-exp-imposition-transmit 0
interface fa3/0
rate-limit input access-group rate-limit 200 8000 8000 8000 conform-action set-mpls-exp-imposition-transmit 4 exceed-action set-mpls-exp-imposition-transmit 0

Número de IPs detectadas:
5502

Número total de paquetes:
5502
```

Ilustración 36. Reporte generado por el agente Ejecutor ante un ataque TCP Flood.

Estas instrucciones (27) podrían crear un tráfico más lento en clientes legítimos, pero sería posible la mitigación del ataque DDOS, aunque no es eficaz totalmente.

6. CLASIFICACIÓN

En cuanto a la clasificación, aprendizaje y visualización de datos, la herramienta Weka proporciona estos mecanismos de machine learning. El algoritmo clasificador usado el algoritmo clasificador bayesiano ingenuo al ser el mejor algoritmo de detección DDOS (28). Al estar los agentes encargados de la detección de los ataques de denegación de servicio, la red neuronal se encargará de predecir por qué protocolo e IP se producirá el ataque. En el caso de un ataque, el agente simplificador llamará a un método auxiliar que analizará todo el paquete Wireshark y clasificará por que protocolo se está produciendo el ataque.

En el siguiente caso, se muestra un ataque TCP por 3 IP diferente.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      178925               99.9743 %
Incorrectly Classified Instances        46                0.0257 %
Kappa statistic                          0.3783
Mean absolute error                      0.0001
Root mean squared error                  0.009
Relative absolute error                128.9147 %
Root relative squared error            161.1684 %
Total Number of Instances           178971

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               ?         0,000     ?           ?       ?          ?        ?         ?         LOOP
               ?         0,000     ?           ?       ?          ?        ?         ?         CDP
               ?         0,000     ?           ?       ?          ?        ?         ?         0x6002
               1,000     0,000     1,000       1,000   1,000      0,483    1,000     1,000     TCP
               1,000     0,000     0,233       1,000   0,378      0,483    1,000     0,394     HTTP
Weighted Avg.  1,000     0,000     1,000       1,000   1,000      0,483    1,000     1,000

=== Confusion Matrix ===

     a      b      c      d      e   <-- classified as
     0      0      0      0      0 |      a = LOOP
     0      0      0      0      0 |      b = CDP
     0      0      0      0      0 |      c = 0x6002
     0      0      0 178911     46 |      d = TCP
     0      0      0      0     14 |      e = HTTP
```

Ilustración 37. Clasificación y predicción del protocolo ante un ataque TCP.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       97288               65.3264 %
Incorrectly Classified Instances     51638               34.6736 %
Kappa statistic                          0.3136
Mean absolute error                      0.0766
Root mean squared error                  0.195
Relative absolute error                 81.3125 %
Root relative squared error             89.8575 %
Total Number of Instances           148926

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0,304     0,000     1,000       0,304   0,467      0,552    0,927     0,340     138.100.11.2
               0,607     0,271     0,786       0,607   0,685      0,326    0,700     0,847     220.181.108.1
               0,729     0,393     0,531       0,729   0,614      0,327    0,700     0,468     77.50.101.1
Weighted Avg.  0,653     0,317     0,690       0,653   0,658      0,327    0,700     0,704
```

Ilustración 38. Clasificación y predicción del origen del ataque.

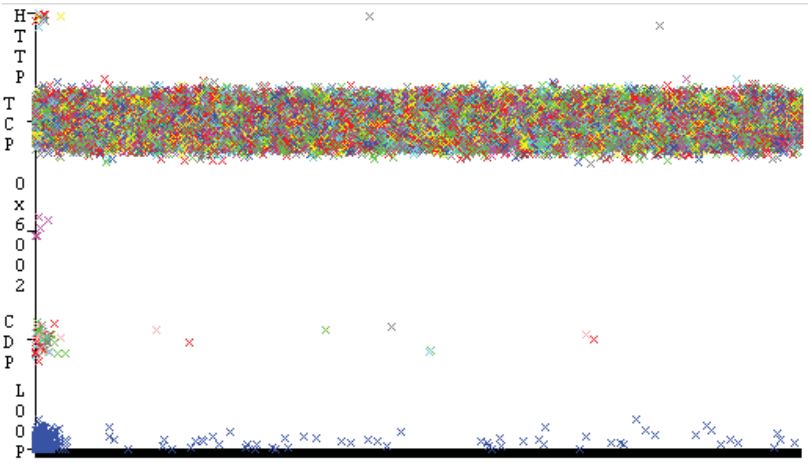


Ilustración 39. Tráfico producido ante un ataque TCP.

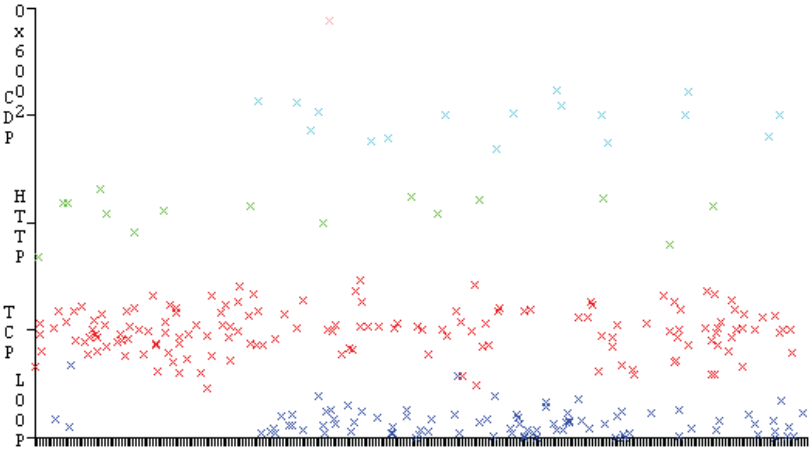


Ilustración 40. Tráfico producido en un periodo normal.

En otro ejemplo, se simula el ataque TCP e ICMP simultáneamente, para comprobar cómo se comporta el agente a pesar de que el Agente Simplificador solo le avisa de un ataque TCP, los resultados de la red neuronal serán:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      132931               95.369  %
Incorrectly Classified Instances      6455                4.631  %
Kappa statistic                          0.9043
Mean absolute error                      0.0263
Root mean squared error                  0.1289
Relative absolute error                 11.0986 %
Root relative squared error             37.4929 %
Total Number of Instances           139386

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     LOOP
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     CDP
               0,925    0,000    1,000      0,925    0,961      0,908    1,000     1,000     ICMP
               1,000    0,075    0,892      1,000    0,943      0,908    1,000     1,000     TCP
Weighted Avg.  0,954    0,029    0,959      0,954    0,954      0,908    1,000     1,000

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
   136     0     0     0 |     a = LOOP
     0    25     0     0 |     b = CDP
     0     0 79659  6455 |     c = ICMP
     0     0     0 53111 |     d = TCP
```

Ilustración 41. Clasificación y predicción del protocolo ante un ataque.

Sin embargo, ante un ataque en el cual el atacante genera IP aleatorias, es incapaz de clasificar y aprender sobre el origen del ataque.

7. CONCLUSIONES

En este TFG se ha presentado el diseño e implementación de un sistema multi agente con el fin de mitigar ataques de denegación de servicio.

La aplicación cumple con su trabajo de mitigación y generación de logs al usuario. Se ha mostrado el procedimiento realizado para cumplir los objetivos iniciales y los resultados obtenidos ante los ataques.

Sin embargo, estas mitigaciones no son totalmente efectivas tienen diferentes ventajas y debilidades. Por ejemplo, un ataque de tipo ICMP será más fácil de mitigar que un ataque TCP, puesto que, pese a la limitación, el servidor seguirá contestando peticiones con fin de prestar servicio a accesos legítimos al servidor, corriendo el riesgo de quedar inoperativo finalmente si el ataque se produce en grandes proporciones.

En el caso de IP aleatorias, la seguridad disminuye, puesto que será muy difícil distinguir quien es un cliente legítimo de quien es un atacante.

8. LÍNEAS FUTURAS

Este trabajo deja abiertas posibles líneas de continuación, tales como:

- Detectar nuevos tipos de ataques DDOS y aplicar detección de inyección de código en paquetes de tipo HTTP.
- En el caso de machine learning, aplicar técnicas de clasificación más complejas.
- Mejorar la eficiencia de los agentes.

- Uso de enrutadores de otras marcas para comprobar su comportamiento ante estos ataques y comparar los resultados obtenidos.

Referencias

1. **Accenture.** Accenture - Cost of cyber crime study. [Online] 2017. https://www.accenture.com/t20170926T072837Z__w__/us-en/_acnmedia/PDF-61/Accenture-2017-CostCyberCrimeStudy.pdf.

2. **CISCO.** *Annual CyberSecurity Report - The attack landscape.* 2018.

3. **Mas, Ana.** *Agentes software y sistemas multiagente: conceptos, arquitecturas y aplicaciones.* s.l. : Prentice Hall, 2004.

4. **Xie, Jing and Liu, Chen-Ching.** Multi-agent systems and their applications. [Online] 14 Jul 2017. https://www.tandfonline.com/doi/full/10.1080/22348972.2017.1348890.

5. **Kolaczek, G., Pieczynska, A., Juszczyszyn, K., Grzech, A., Katarzyniak, R., Nguyen.** A mobile agent approach to intrusion detection in network systems. *Knowledge-Based Intelligent Information and Engineering Systems.* 2005.

6. **International Organization for Standarization.** *Information technology - Security techniques - Information security management system - Requirements (ISO/IEC 27001:2005).* 2005.

7. **Pallas, Gustavo and Corti, María Eurgenia.** *Metodología de Implantación de un SGSI en.* Uruguay : Facultad de Ingeniería, Universidad de la República.

8. **Ferreira, Diogo and Mira da Silva, Miguel.** *Using process mining for ITIL assessment: a case study with incident management.* Lisboa, Portugal : s.n., 2008.

9. **Brenner, M.** *Classifying ITIL Processes; A Taxonomy under Tool Support Aspects.* University of Munich : Munich Network Management Team, 2008.

10. **Norvig, Peter and Russell, Stuart J.** *Artificial Intelligence: A Modern Approach (2en ed.).* New Jersey : Prentice Hall. ISBN 0-13-790395-2, chpt. 2.

11. **Wooldridge, M. and Jennings, N. R.** *Application of intelligent agents.* Ney York, USA : Springer, 1998.

12. **Kubera, YUoann, Mathieu, Philippe and Picault, Sébastien.** *Everything can be Agent!* Toronto, Canada : s.n., 2010.

13. **Wooldridge, Michael.** *An Introduction to MultiAgent Systems.* 2002.

14. **Cloudflare.** [Online] https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/.

15. **Cid, Daniel.** sucuri. [Online] 6 febrero 2014. https://blog.sucuri.net/2014/02/layer-7-ddos-blocking-http-flood-attacks.html.

16. **Software Engineering Institute, Carnegie-Mellon University.** *TCP SYN Flooding and IP Spoofing Attacks.* 1996.

17. **Bai, Kai.** *Analysis and Prevention of Memcache UDP Reflection Amplification.* Hubei, China : International Journal of Science Vol.5 No.3 , 2018. ISSN: 1813-4890.

18. *The Rise and Decline of NTP DDoS Attacks.* **Jakub Czyz, Michael Kallitsis, Michael Kallitsis, Christos Papadopoulos, Michael Bailey.** Ann Arbor, MI, USA : Proceedings of the 2014 Conference on Internet Measurement Conference, 2014. ISBN: 978-1-4503-3213-2.

19. *DNS amplification attack revisited.* **Marios Anagnostopoulos, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, Stefanos Gritzalis.** Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece : Computers & Security Volume 39, Part B, November 2013, Pages 475-485, 2013.

20. *Amplification Hell: Revisiting Network Protocols for DDoS Abuse.* **Rossow, Christian.** Horst Gortz Institute for IT-Security, Ruhr University Bochum, Germany : NDSS, 2014.

21. **Peter M. Thornewell, Lisa M. Golden,.** *DNS flood protection platform for a network. US8261351B1* F5 Networks, Inc., Seattle, WA (US) , 04 09 2012.

22. *Towards Mitigation of Low and Slow Application DDoS Attacks.* **Mark Shtern, Roni Sandel, Marin Litoiu.** Boston, MA, USA : IEEE, 2014. 978-1-4799-3766-0.

23. *Ping of death.* **Kenney, M.** 1996, Vols. Insecure. org, vol. 2.

24. *Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet.* **Kumar, Sanjeev.** San Jose, CA, USA : IEEE, 2007. 0-7695-2911-9.

25. *A Methodology for the Analysis and Design of Multi-Agent Systems using JADE.* **M. Nikraz, G. Caire y P. Bahri.** Rockingham : s.n., 2006, Vol. International Journal of Computer Systems Science & Engineering.

26. *Using Jpcap API to Monitor, Analyse and Report Network Traffic for DDoS Attacks.* **G, Dileep Kumar.** s.l. : Conference: 2014 14th International Conference on Computational Science and Its Applications (ICCSA), 2014. 10.1109/ICCSA.2014.18 .

27. **CISCO.** https://www.cisco.com/c/en/us/td/docs/routers/xr12000/software/xr12k_r4-2/system_monitoring/configuration/guide/b_sysmon_cg42xr12k/b_sysmon_cg42xr12k_chapter_011.html. [Online]

28. *Demystifying and Rate Limiting ICMP hosted DoS/DDoS Flooding Attacks with Attack Productivity Analysis.* **Udhayan, J. and Anitha, R.** Patiala, India : IEEE, 2009. 10556652.

29. **Chandel, Raj.** http://www.hackingarticles.in/understanding-guide-icmp-protocol-wireshark/. [Online] 7 Octubre 2017.

30. *Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks.* **AARTI SINGH, DIMPLE JUNEJA.** Institute of Computer Technology & Business Management : International Journal of Engineering Science and Technology , 2010. Vol. 2(8), 2010, 3405-3411.

31. **Eddy, Wesley M.** Defenses Against TCP SYN Flooding Attacks - The Internet Protocol Journal. *CISCO.* [Online] 2006.

https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-34/syn-flooding-attacks.html. Volume 9, Number 4.

32. **CISCO.** Cisco IOS Security Configuration Guide, Release 12.2. *Chapter: Configuring TCP Intercept (Preventing Denial-of-Service Attacks).* [Online] 12 2 2014. https://www.cisco.com/c/en/us/td/docs/ios/12_2/security/configuration/guide/fsecur_c/scfdenl.html.

33. —. Identifying and Mitigating the Distributed Denial of Service Attacks Targeting Financial Institutions. [Online] 1 Diciembre 2015. https://tools.cisco.com/security/center/viewAMBAlert.x?alertId=27115. 27115.

34. —. CISCO. [Online] 25 Agosto 2003. https://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/fspolic.html#wp1042034.

35. *Review of syn-flooding attack detection mechanism.* **Mehdi Ebady Manna, Angela Amphawan.** 1, School of computing, University Utara Malaysia, Kedah, Malaysia : International Journal of Distributed and Parallel Systems, 2012, Vol. 3. 10.5121/ijdps.2012.3108.

36. **Deal, Richard.** *Cisco Router Firewall Security.* 2005. ISBN-10: 1-58705-175-3.

37. **CISCO.** Cisco IOS Quality of Service Solutions Command Reference, Release 12.2. [Online] https://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/command/reference/fqos_r/qrfcmd8.html.

38. *An Approach of DDOS Attack DetectionUsing Classifiers.* **Johnson KH, Tanmay De.** 2015, Vols. Trust Based Node Scheduling Protocol for Target Coverage in Wireless Sensor Networks (pp.429-437). DOI 10.1007/978-81-322-2550-8_41.

Este documento esta firmado por

Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Fri Jul 06 20:42:25 CEST 2018
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)