Sistema de Detección de Ataques EDoS en Entornos Cloud

TRABAJO DE FIN DE GRADO

Julio Javier López Giménez

José Ángel Madrona Martini

Lorenzo Susarte Trujillano

Directores:

Luis Javier García Villalba

Ana Lucila Sandoval Orozco

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Madrid, Junio de 2015

Agradecimientos

Quisiéramos agradecer a Luis Javier García Villalba y a Ana Lucila Sandoval Orozco, los Directores de este Trabajo, el apoyo brindado.

Asimismo, quisiéramos agradecer la dedicación de Jorge Maestre Vidal. Sin su inestimable ayuda, este Trabajo no habría sido posible.

Finalmente, nuestro más sincero agradecimiento al resto de miembros del Grupo GASS (Grupo de Análisis, Seguridad y Sistemas, http://gass.ucm.es), Grupo de Investigación del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid, por las facilidades ofrecidas.

Resumen

La computación en la nube constituye el modelo de provisión de recursos de cualquier plataforma o aplicación web como servicio, bajo demanda y de forma dinámica (elasticidad) a través de internet. Se concibe como un nuevo paradigma informático que permite ofrecer los distintos servicios a través de la red. Esto aporta un importante conjunto de ventajas sobre los servidores físicos convencionales, pero también implica nuevos desafíos. De entre ellos cabe destacar el rápido crecimiento de las brechas de seguridad, que viene propiciado por la adaptación de diferentes tipos de amenazas convencionales, siendo los ataques de Denegación de Servicio Distribuido (DDoS) una de las más frecuentes. A los ataques de denegación de servicio que buscan agotar la sostenibilidad económica de los servicios en la nube se les denomina ataques EDoS (*Economic Denial of Sustainibility*). Se trata de una amenaza muy reciente, que ha puesto en alerta a la comunidad investigadora y a las diferentes organizaciones para la ciberdefensa. Sin embargo, a pesar del peligro que conllevan, existen muy pocas propuestas para detenerlos. Con el fin de contribuir a su mitigación, en este trabajo se propone una estrategia de detección de intrusiones especializada en el reconocimiento de estos tipos de ataques. La aproximación realizada combina la construcción de métricas a partir del análisis de la entropía del tráfico monitorizado, con la construcción de series temporales y modelos predictivos capaces de predecir el estado de la red. Esto permite identificar comportamientos impredecibles, y por lo tanto, anómalos, que revelen la presencia de amenazas EDoS. La propuesta ha sido evaluada con la colección de muestras de dominio público CAIDA'07, y bajo su despliegue en un caso de uso real. En ambos casos se ha comportado de manera eficaz y precisa, demostrando su capacidad de afrontar los objetivos propuestos.

Palabras clave

Cloud, Denegación de Servicio, Detección de Intrusiones, Entropía, Holt-Winters, Nube, Redes, Seguridad de la Información.

Abstract

The cloud computing model is the provision of resources for any platform or web application as a service, on demand and dynamically (elasticity) through internet. It is conceived as a new computing paradigm that can offer different services through the network. This provides an important set of advantages over conventional physical servers, but it also involves new challenges. Among them include the rapid growth of security breaches, which is facilitated by the adaptation of different types of conventional threats, with the Distributed Denial of Service (DDoS) attacks one of the most frequent. For the denial of service attacks that seek exhaust the economic sustainability of cloud services are called EDoS attacks (Denial of Economic Sustainibility). This is a very recent threat, which has alerted the research community and various organizations for cyber defense. However, despite the danger they entail, there are very few proposals to stop them. In order to contribute to mitigation, in this paper a strategy intrusion detection specialist in the recognition of these types of attacks is proposed. The approach made, combining the construction of metrics based on the analysis of entropy of the monitored traffic, with the construction of time series and predictive models capable of predicting the state of the network. This allows to identify unpredictable behavior, and therefore anomalous, indicating the presence of EDoS threats. The proposal has been evaluated with the collection of samples CAIDA'07 public domain and under their deployment in a case of actual use. In both cases it has acted efficiently and accurately, demonstrating its ability to meet the objectives.

Keywords

Cloud, Denial of Service, Entropy, Holt-Winters, Information Security, Intrusion Detection System, Network.

Los abajo firmantes autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores el presente Trabajo Fin de Grado: "Sistema de Detección de Ataques EDoS en Entornos Cloud", realizado durante el curso académico 2014-2015 bajo la dirección de Luis Javier García Villalba y Ana Lucila Sandoval Orozco en el Departamento de Ingeniería del Software e Inteligencia Artificial, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Julio Javier López Giménez

José Ángel Madrona Martini

Lorenzo Susarte Trujillano

Índice General

Índice de Figuras

Lista de Acrónimos

ACK	*Acknowledgement*
ARIMA	*Autoregressive Integrated Moving Average*
CAIDA	*Center for Applied Internet Data Analysis*
DARPA	*Defense Advanced Research Projects Agency*
DDoS	*Distributed Denial of Service*
DGA	*Domain Generation Algorithm*
DNS	*Domain Name System*
DoS	*Denial of Service*
EDoS	*Economic Denial of Sustainability*
ENISA	*European Union Agency for Network and Information Security*
HTTP	*Hypertext Transfer Protocol*
IaaS	*Infrastructure as a Service*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
PaaS	*Platform as a Service*
ROC	*Receiver Operating Characteristic*
SaaS	*Software as a Service*
SIP	*Intrusion Detection Working Group*
TCP	*Transmission Control Protocol*
TOR	*The Onion Router*
UDP	*User Datagram Protocol*
VF	*Virtual Firewall*
VoIP	*Voice over Internet Protocol*
XML	*eXtensible Markup Language*

Capítulo 1

Introducción

En la actualidad, la computación en la nube o *cloud computing* y los servidores virtuales han ganando mucho terreno a los servidores físicos convencionales. Esto es debido la gran cantidad de ventajas que ofrecen, entre las que destacan su bajo coste, y una mayor agilidad, flexibilidad y escalabilidad. El primero se debe principalmente al ahorro en gastos de mantenimiento, licencias, y a la adopción del paradigma de pago por uso. Los últimos, a la facilidad de suplir nuevos recursos de cómputo en los casos en que sea requerido. Sin embargo, su despliegue también conlleva una serie de inconvenientes, focalizados principalmente en problemas de disponibilidad, conectividad y seguridad.

En este trabajo se aborda el problema de la seguridad. Su elección ha sido motivada por los avisos emitidos por las diferentes organizaciones relacionadas con la ciberdefensa, quienes han detectado un rápido crecimiento de este tipo de amenazas. Los ataques contra la computación en la nube han evolucionado de las estrategias clásicas de intrusión, a la explotación de vulnerabilidades específicas de su entorno, haciendo que los esquemas de defensa convencionales resulten menos efectivos.

De entre estas nuevas amenazas, cabe destacar el aumento de los ataques dirigidos contra la sostenibilidad de los servicios ofertados. Se trata de un tipo de intrusión que busca comprometer la economía de la víctima, afectando directamente a su modelo de pago por uso. La manera más habitual de alcanzar este objetivo consiste en forzar al servicio desplegado sobre la nube a realizar una gran cantidad de trabajo adicional. Esto conlleva la contratación de recursos adicionales, encareciendo la factura de la víctima, y haciendo que su uso sea inviable.

Por lo general, las amenazas contra la sostenibilidad alcanzan el éxito cuando van de la mano de herramientas utilizadas en otros contextos para la ejecución de ataques de denegación de servicio. A esta combinación se la denomina ataques de denegación de la sostenibilidad de un servicio en la nube, también conocidos

como EDoS. El trabajo realizado se centra en su estudio, y en la propuesta de un esquema defensivo para su identificación. Para su mayor comprensión, a continuación se describe una serie de conceptos previos, los objetivos fijados, y la estructura del resto del documento.

1.1. Conceptos Previos

1.1.1. Servidores Físicos y Virtuales

Un servidor físico es definido como la utilización o configuración de recursos hardware y software, ubicados en un determinado lugar. Por su parte, los servidores virtuales son instalaciones de software realizadas sobre servidores físicos, con capacidad de alojamiento de diferentes máquinas virtuales. Estas comparten entre sí recursos comunes, comportándose de manera complementaria a pesar de abarcar diferentes tareas.

En la actualidad, está cambiando la mentalidad a la hora de alojar las aplicaciones y servicios web en la red. En sus orígenes predominaba la utilización servidores físicos. Esto ofrecía características fijas y poco flexibles, las cuales eran difíciles de modificar por los clientes. Para hacer frente a este problema, la parte contratante cada vez recurre con más frecuencia a las tecnologías relacionadas con la computación en la nube, desencadenando un crecimiento de la demanda de servidores virtuales.

De entre las mejoras que esto supone, cabe destacar su facilidad de arranque, escalabilidad, abaratamiento de costes de mantenimiento y licencias o su sencillez de administración. Sin embargo deben afrontarse problemas relacionados con la contratación de terceras partes, conectividad, disponibilidad, privacidad y seguridad.

1.1.2. Modelos de Provisión

Los modelos de provisión son conjunto de servicios computacionales que permiten gestionar los recursos albergados en internet. Habitualmente aplican alguno de los siguientes esquemas: *mainframe, cliente/servidor o cloud*. A continuación se describe cada uno de ellos:

- **Mainframe**: El aprovisionamiento *mainframe* es de características centralizadas, lo que facilita un alto grado de computación y espacio de almacenamiento. A pesar de su eficiencia, presenta elevados costes de gestión y mantenimiento.
- **Cliente/Servidor**: El aprovisionamiento Cliente/Servidor consiste en un conjunto de máquinas y servidores destinados a computación y almacenamiento

distribuido, promoviendo la agilidad de los servicios y ofreciendo costes bajos de gestión. Su mayor defecto son las licencias obligatorias del software, las cuales acarrean un importante coste económico.

- **Cloud**: El aprovisionamiento *cloud* se realiza desde grandes superficies, con centros de datos repletos de máquinas y servidores de alta escalabilidad. Esto permite la optimización de la eficiencia y la agilidad de los servicios ofrecidos, aplicando el modelo económico de pago por uso. A diferencia del modelo cliente/servidor, ahorra los costes de licencias.

1.1.3. Computación en la Nube

La computación en la nube constituye el modelo de provisión de recursos de cualquier plataforma o aplicación web como servicio, bajo demanda y de forma dinámica (elasticidad) a través de internet. Se concibe como un nuevo paradigma informático que permite ofrecer los distintos servicios a través de la red. Se caracteriza por la rápida movilización de recursos, la cual permite una rápida adaptación a la variabilidad de la demanda. Sus principales características, parecidas a las descritas en servidores virtuales, son: costes variables, escalabilidad, elasticidad, trabajo colaborativo, ahorro energético, coherencia y ubicuidad. Dentro del *cloud*, y según el tipo de servicio prestado, pueden distinguirse los siguientes tipos de capas Fig. 1.1:

- **SaaS (Software como Servicio)**: El *SaaS* ofrece una aplicación completa como servicio bajo demanda y perite el uso colaborativo por parte de varios usuarios. El uso de las aplicaciones se limita al acceso a través de un navegador web, sin necesidad de instalaciones.

- **PaaS (Plataforma como servicio)**: El *PaaS* es la capa intermedia de la computación en la nube. Permite la abstracción completa de un entorno de desarrollo y ofrece un conjunto de complementos y paquetes alternos, con los que poder completar la aplicación implementada sobre el mismo. Entre los complementos se pueden encontrar distintas librerías que proporcionen funcionalidades específicas, también conocidas como *APIs*.

- **IaaS (Infraestructura como servicio)**: El *IaaS* es la capa más baja de la computación en la nube. Define la infraestructura que sostiene las capas superiores. Entre los distintos componentes que se pueden gestionar bajo ella, se encuentran los propios servidores, almacenamiento, capacidades de cómputo que sirvan de sustento para las capas superiores, y las aplicaciones alojadas en

él. En esta capa se proporcionan las capacidades particulares del *cloud* como la escalabilidad y la distribución de carga de trabajo entre servidores

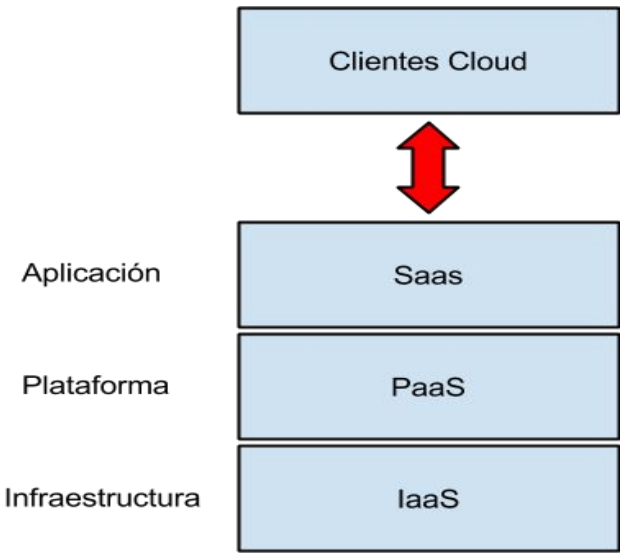

Figura 1.1: Capas del *Cloud*

Pueden distinguirse tres tipos de nubes dentro del *cloud*, clasificadas en función de su administrador. A continuación se describen brevemente:

- **Nube pública**: El mantenimiento de la nube pública es desempeñado por terceras personas, en lugar de por la propia organización contratante. Dentro de este conjunto de sistemas de almacenamiento y servidores circularan los datos de varias organizaciones, de los cuales el usuario final solo vera los suyos, y podrá realizar distintas configuraciones de los mismos.

- **Nube privada:** El mantenimiento de la nube privada es desempeñado por la propia organización contratante. Se trata de una opción adaptada a las compañías que necesiten un alto nivel de protección de datos.

- **Nube híbrida**: La nube híbrida combina los dos tipos anteriores. En este caso, una parte de la información es gestionada por la organización contratante, y el resto es de dominio público.

- **Nube comunitaria**: La nube comunitaria está destinada a ser utilizada con una finalidad común entre colectivos de participantes. Puede ser mantenida por el proveedor, o por organizaciones contratantes.

1.1.4. Virtualización

La virtualización es la tecnología que permite la existencia de la computación en la nube, ya que proporciona la flexibilidad y escalabilidad necesarias para satisfacer los demandas de los usuarios. Habitualmente es definida como la capacidad de arrancar o ejecutar máquinas virtuales en un hipervisor *cloud*. En este proceso, cada máquina virtual simula el funcionamiento completo de una maquina física, conllevando toda su implementación, lo que incluye aspectos tan relevantes como su propio *kernel*, sistema operativo o aplicaciones.

1.2. Objetivos del Trabajo

El principal objetivo de este trabajo es la realización de un sistema de detección de intrusos o IDS, para la identificación de ataques de inundación EDoS dirigidos contra infraestructura *cloud*. Esto implica satisfacer los siguientes objetivos secundarios:

- La investigación de las técnicas de defensa y ataque más relevantes en este tipo de arquitecturas, haciendo especial hincapié en aquellas que se relacionan con los ataques EDoS.
- El desarrollo de estrategias para la extracción, y la interpretación de las características del tráfico que se dirige hacia ellas.
- La elaboración de métricas que permitan el modelado de dicha información.
- La construcción de modelos predictivos que permitan la identificación de anomalías estadísticas, capaces de desenmascarar intentos de intrusión.
- La elaboración de una metodología de evaluación que suple en la medida de lo posible, las carencias de los esquemas convencionales.
- La verificación de la eficiencia de la propuesta.

1.3. Estructura del Documento

En este capítulo, y a modo de introducción, han sido descritos los conceptos previos y el objetivo del trabajo realizado.

En el capítulo 2 se indican las principales características de los ataques de denegación de servicio.

En el capítulo 3 se analizan las técnicas defensivas frente a dicha amenaza, y se repasan los trabajos previos de mayor relevancia.

En el capítulo 4 se describen las diferentes propiedades de la entropía y los modelos de predicción basados en series temporales, que han sido tenidos en cuenta en la elaboración de esta propuesta.

En el capítulo 5 se introduce el sistema de detección de ataques EDoS desarrollado.

En el capítulo 6 explica la experimentación realizada, y la metodología de evaluación adoptada.

En el capítulo 7 se discuten los resultados obtenidos. Finalmente, en el capítulo 8 se presentan las conclusiones y propuestas de trabajos futuros.

Capítulo 2

Ataques de Denegación de Servicio

En la actualidad existen diferentes amenazas capaces de atentar contra la integridad, confidencialidad, autenticidad y la disponibilidad de la información, siendo la denegación de servicio una de las más relevantes. En este capítulo se distinguen sus principales características y métodos de actuaci'on, profundizando en aquellos dirigidos específicamente contra entornos de computacón en la nube.

2.1. Descripción y Características Generales

La denegación de servicio, también conocidos como ataques DoS, tienen por objetivo el que un servicio sea inaccesible a los usuarios legítimos. Aunque existen muchas variedades, sus métodos de intrusión más comunes consisten en el agotamiento de los recursos de cómputo o el ancho de banda de las víctimas. Generalmente son ejecutadas desde sencillas plataformas cuyo uso no requiere conocimientos avanzados. De este modo se ha disparado su popularidad, convirtiendo su mitigación en uno de los principales desafíos de las organizaciones para la ciberdefensa. Este es el caso de la Agencia Europea de Seguridad de las Redes y de la Información (ENISA), quienes observaron un incremento de más de un 70 % en el periodo de tiempo delimitado entre los años 2013 y 2014. Además, otras organizaciones han avisado de la tendencia a su aplicación como medio para alcanzar objetivos más complejos. Por ejemplo, en el informe de anual de amenazas publicado por la empresa Symantec en 2015, se describen métodos de uso más controvertidos, como el enmascaramiento de las acciones realizadas por *software* malicioso o el encubrimiento de operaciones de blanqueo de capitales [2].

Con el avance de las medidas defensivas, han aparecido variantes mucho más complejas, que son capaces de maximizar el daño causado y dificultar las tareas de prevención. Este es el caso de los ataques de denegación de servicio distribuidos o

DDoS. Estos ataques parten de diferentes puntos de origen. Por lo general, los nuevos focos son un gran número de máquinas infectadas, las cuales son controlados por una máquina central o administradora, a los que se conoce como redes de ordenadores zombis o *botnets*. Tecnológicamente representan uno de los ataques más sencillos de ejecutar, pero también más dañinos. Además, la tecnología que opera detrás del mantenimiento de las botnets cada vez es más sofisticada, dificultando todaía más su detección. En Fig.2.1 se muestra un sencillo ejemplo de intrusión mediante DDoS.

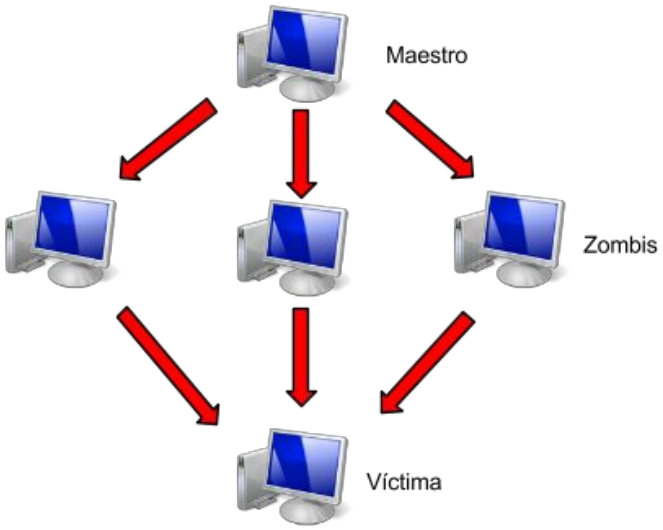

Figura 2.1: Ejemplo de ataque DDoS

En ella el ordenador que actúa como controlador (denominado *master*) ha infectado 3 ordenadores, a los cuales encargara el ataque en conjunto contra la maquina víctima, tratando de denegar su servicio. Nótese que en la realidad, la ejecución de esta amenaza conlleva la participación de una cantidad mucho mayor de sistemas comprometidos.

2.2. Motivación

La ejecución de ataques de denegación de servicio parte de diferentes motivaciones. En [3] se recopilan las más frecuentes, siendo enumeradas a continuación:

- **Ganancia financiera o económica**: Los ataques impulsados por motivos económicos o financieros son una de las principales preocupaciones de las organizaciones, ya que atentan directamente contra su economía. El perfil del

atacante que normalmente los desempeña habitualmente presenta conocimientos tecnológicos y financieros avanzados, y suele guardar relación con la víctima.

- **Venganza**: Los ataques motivados por venganza suelen tener su origen en individuos u organizaciones frustrados con otra organización o gobierno. En su perfil no se suelen incluir conocimientos tecnológicos avanzados.
- **Creencias ideológicas**: Los atacantes impulsados por creencias ideológicas, habitualmente comprenden motivos sociales, políticos o religiosos. Detrás de estos móviles se encuentran usuarios con conocimientos técnicos avanzados, y son la causa de importantes famosas acciones de sabotaje, como las que afectaron a Estonia (2007), Irán (2009) o WikiLeaks (2010) [4].
- **Desafío intelectual**: Algunos atacantes encuentran en la ejecución de intrusiones con éxito un desafío intelectual. Bajo estos perfiles normalmente se encuentran jóvenes entusiastas de la seguridad de la información, con ganas de demostrar sus capacidades.
- **La guerra cibernética:** Los atacantes que originan estas acciones habitualmente pertenecer a fuerzas armadas de estados u organizaciones terroristas, que guiadas por motivaciones pol'iticas, persiguen comprometer una amplia gama de secciones críticas de sus adversarios. Los objetivos potenciales de estos ataques incluyen, pero no se limitan a, los departamentos ejecutivos y agencias civiles, organizaciones financieras privadas o públicas, y las infraestructuras de energía, agua o telecomunicaciones.

2.3. Ataques DoS en la Nube

Dado que este trabajo se centra en un tipo de amenaza propia de los entornos de computación en la nube, es conveniente describir brevemente los diferentes métodos para el agotamiento de recursos dirigidos contra dichas plataformas. La detección de DoS en la nube habitualmente se lleva a cabo a nivel de red y de aplicación, siendo su capa de actuación, el principal eje de la clasificación. En Fig. 2.2 se muestra la organización de esta taxonomía.

2.3.1. Ataques Dirigidos contra la Capa de Red

Los ataques dirigidos contra la capa de red se basan en la explotación de errores y vulnerabilidades de los protocolos de comunicación más utilizados, destacando entre

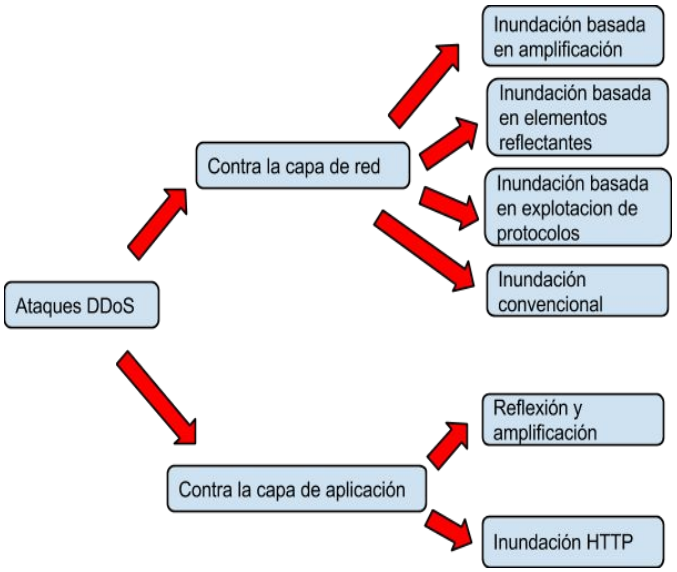

Figura 2.2: Clasificación de ataques DoS contra la infraestructura *Cloud*

ellos HTTP, TCP, UDP, ICMP y DNS. Todos ellos consisten en el envío masivo de información con el fin de desbordar la capacidad de cómputo de los servidores.

Tal y como se determina en [5], existen dos tipos de inyecciones de tráfico capaces de comprometer un servicio por medio de inundación. En primer lugar, es posible la generación constante y continuada de tráfico malicioso, lo que se conoce como inundación de tasa alta. Por otro lado, y con el fin de evadir las técnicas de detección, es posible la inyección de patrones menos ruidosos, y en diferentes frecuencias. A esto se lo denomina inundación de tasa baja. Un ejemplo típico de ellos se trata en [6], donde los ataques buscan explotar vulnerabilidades del protocolo TCP mediante ataques con ráfagas caracterizadas por patrones ON/OFF.

La inundación contra servidores *Cloud* se divide en cuatro categorías:

- **Inundación convencional**: Los atacantes de inundación convencionales consisten en interrumpir la conectividad del usuario legítimo al agotar el ancho de banda de la víctima. Existen diferentes formas de llevarse a cabo, destacando entre ellas las inundaciones mediante peticiones UDP, ICMP, DNS y VoIP [7].

- **Inundación basada en la explotación de protocolos**: Los ataques de inundación basados en la explotación de protocolos, aprovechan características específicas o errores de implementación de algunos de los protocolos de la víctima con el fin de consumir cantidades excesivas de recursos, y en muchos casos, agotar las colas que permiten el inicio de sesión de nuevos usuarios.

Algunos ejemplos de ellos son la inundación de peticiones TCP SYN, TCP SYN-ACK, ACK con PUSH ACK o RST/FIN.

- **Inundación basada en elementos reflectores**: Los ataques de inundación basados en reflexión, suelen enviar solicitudes falsas (por ejemplo, solicitud de *ecco request* en ICMP) en lugar de las solicitudes directas, a los reflectores. Su modo de actuación habitualmente se divide en tres etapas: En la primera fase, se infectan los sistemas. En la segunda fase, una vez infectados los sistemas (zombies), se les ordena la inyección de las solicitudes a terceras partes (reflectores), con la dirección IP de origen de la víctima. En la última fase, los reflectores envían la respuesta a la víctima, constituyendo un ataque de DDoS. En [8] se tratan estas amenazas en profundidad.
- **Inundación basada en amplificación**: Para amplificar el daño de un ataque, los atacantes explotan vulnerabilidades en servicios, lo que les permite generar grandes mensajes, o mensajes múltiples, a partir de sencillos datagramas. Las técnicas de reflexión y de amplificación habitualmente trabajan de manera cooperativa. Uno ejemplo de ellos se observa en los ataques conocidos como *smurf*, donde los atacantes envían solicitudes con direcciones IP de origen falsificadas a un gran número de reflectores, y explotan características de difusión IP de los paquetes para amplificar su impacto [3].

2.3.2. Ataques Dirigidos contra la Capa de Aplicación

Los ataques DDoS en la capa de aplicación se centran en interrumpir los servicios del usuario legítimo por el agotamiento de los recursos del servidor (por ejemplo, *sockets*, CPU, memoria, ancho de banda o base de datos y el ancho de banda de E/S). Generalmente consumen menos ancho de banda y son más sigilosos que los ataques volumétricos contra la capa de red. Su gran similitud con el tráfico legítimo dificulta considerablemente las tareas de detección. Adicionalmente, los ataques de inundación a nivel de aplicación suelen representar el mismo impacto que los que se dirigen contra servicios de red, ya que también explotan características específicas de protocolos, tales como HTTP, DNS, o SIP.

En la infraestructura *Cloud* se distinguen dos grandes amenazas a nivel de aplicación: los ataques de reflexión y amplificación, y la inundación mediante solicitudes HTML. A continuación se describe cada una de ellas:

- **Reflexión y amplificación**: Los ataques reflectantes y de amplificación utilizan las mismas técnicas de los ataques en la capa de red. En Fig 2.3 se muestra un ejemplo típico de amplificación DNS.

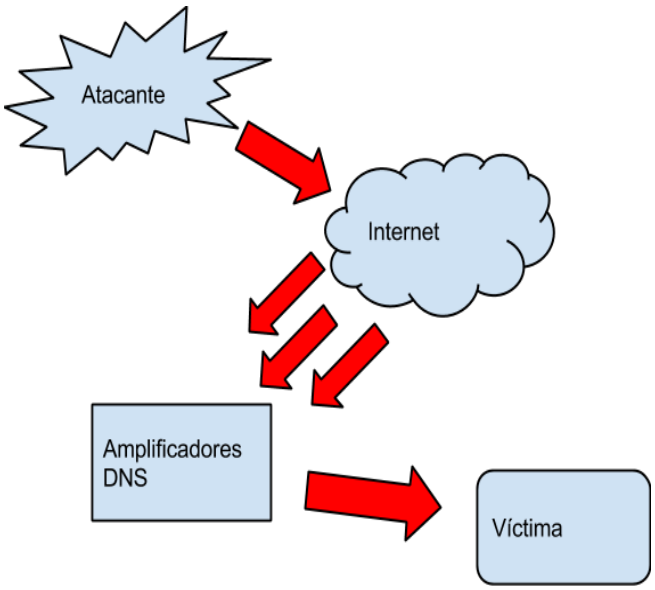

Figura 2.3: Ejemplo de amplificación DNS

En ella se combinan mecanismos de reflexión y amplificación: Los atacantes zombies generan pequeñas consultas DNS con las direcciones IP de origen falsas. Dado que a menudo las respuestas a peticiones DNS presentan un mayor tamaño que las solicitudes, el ataque alcanzará a la víctima habiendo incrementado su impacto [8].

Otro ejemplo de ataque en la capa de aplicación que emplea la técnica de reflexión, es la inundación VoIP [3]. En Fig. 2.4 puede observarse cómo los atacantes envían grandes cantidades de paquetes VoIP falsificados a través del *proxy* SIP, con un rango muy grande de direcciones IP de origen. En consecuencia, el servidor VoIP víctima tiene que distinguir las conexiones VoIP legítimas, de las conexiones maliciosas. Esto provoca un gran consumo de recursos, alcanzándose con frecuencia la disminución de la calidad de su servicio.

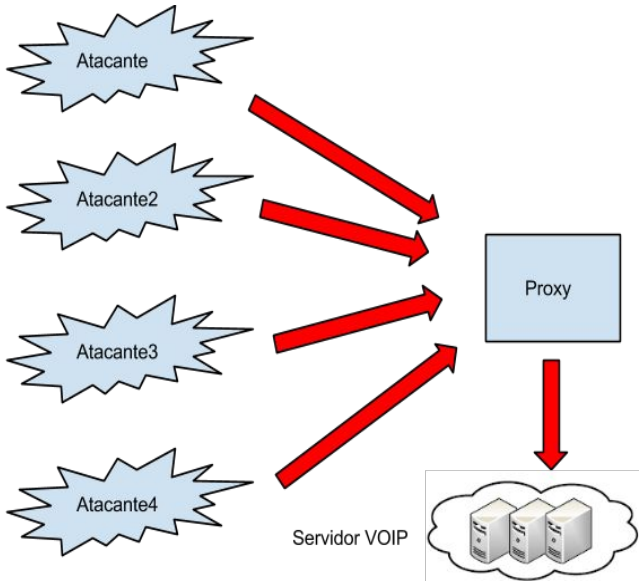

Figura 2.4: Ejemplo de reflexión SIP

- **Inundación HTTP**: Existen cuatro principales tipos de ataques dirigidos contra los servicios HTTP ofrecidos por la infraestructura *Cloud*: ataques de inundación mediante inicios de sesión, ataques de inundación mediante peticiones, ataques asimétricos y ataques de respuesta lenta. Los primeros tratan de colapsar las colas de peticiones de inicio de sesión con solicitudes que nunca podrán ser respondidas. Esto dificulta el procesamiento de peticiones originadas por usuarios legítimos. Por otro lado, los ataques en la inundación de peticiones (habitualmente Get/Post) tratan de desbordar al servidor mediante el procesamiento de solicitudes de usuarios que previamente han iniciado sesión. Su éxito suele depender de la envergadura de la red de sistemas zombies desde la que se inicie la intrusión. Los ataques asimétricos son una variante de la inundación de peticiones, que tiene como característica la explotación de vulnerabilidades en protocolos que permiten realizar diferentes solicitudes desde un único datagrama. Finalmente, los ataques de respuesta lenta se basan en mantener las conexiones HTTP establecidas el mayor tiempo posible. Esto puede conseguirse de diferentes manera, como por ejemplo, mediante el envío de peticiones parciales con un único encabezado, o sin salto de línea, de manera continuada y muy lentamente. En [9] cada una de estas amenazas es revisadas en mayor profundidad.

2.4. Ataques EDoS

La computación en la nube permite contratar servidores para poder atender a un gran número de solicitudes de un servicio. La introducción de plataformas *Cloud* ricas en recursos, donde los usuarios pagan basándose en su uso, ha transformado la denegación de servicio distribuida en un problema muy importante para la economía de las organizaciones dependientes de estas tecnologías.

Este nuevo tipo de amenazas son denominadas ataques de denegación de la sostenibilidad económica de los servidores o EDoS, y buscan que los servicios víctima se vuelven insostenibles ante la imposibilidad de facturar los servicios contratados. En Fig. 2.5 se muestra un ejemplo de ataque EDoS.

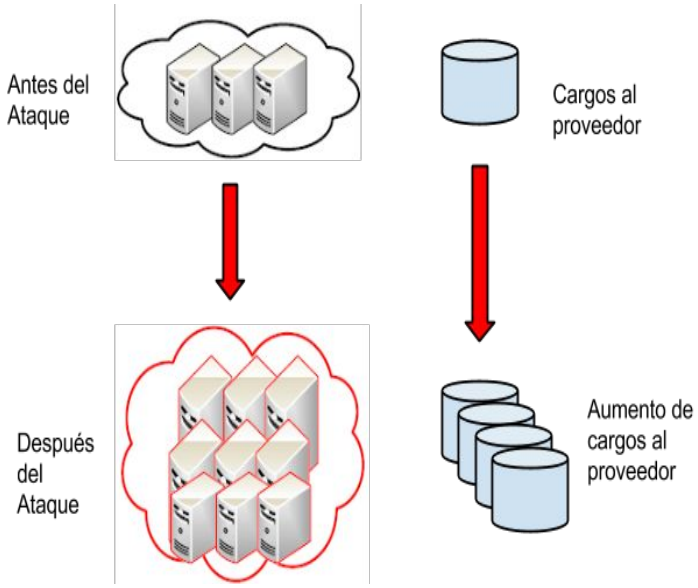

Figura 2.5: Ejemplo de ataque EDoS

En ella puede observarse como en la situación legítima, la víctima tiene contratado unos determinados servicios. Estos tienen la capacidad de auto-escalado, con el fin de satisfacer sus necesidades de cómputo. Al producirse el ataque, la carga de trabajo del sistema crece considerablemente. Esto conlleva un escalado de recursos mucho mayor, lo que aumenta la cantidad de servicios contratados por la víctima, y por lo tanto su facturación.

Capítulo 3

Técnicas de Defensa

A continuación son descritas las principales técnicas de defensa frente a ataques de denegación de servicio. El capítulo está dividido en tres secciones: en la primera, se explican las diferentes acciones habitualmente consideradas para reducir el problema de la denegación de servicio; en la segunda, se describen las principales publicaciones orientadas a la defensa convencional frente a este tipo de amenazas; finalmente, se profundiza en aquellas que han tenido como objeto de análisis, los ataques basados en el agotamiento de la sostenibilidad de los servicios en la nube.

3.1. Acciones Defensivas

Las acciones defensivas frente a la denegación de servicio pueden efectuarse sobre diferentes contextos y escenarios. Sin embargo, tal y como se indica en [1], tienen en común los objetivos parciales que tratan de alcanzar. En base a esto, generalmente son agrupadas en cuatro categorías: prevención, detección, identificación del origen y mitigación. Nótese que al considerar como eje de clasificación su momento de actuación, se presenta una clasificación idéntica. Esto puede percibirse con mayor claridad en Fig. 3.1.

En ella se observa que para que un ataque alcance el éxito, en primer lugar deberá de traspasar las medidas de prevención. A continuación tendrá que evadir los sistemas de detección. En caso de no lograrlo, el defensor tratará de localizar su origen, y de aplicar medidas de mitigación.

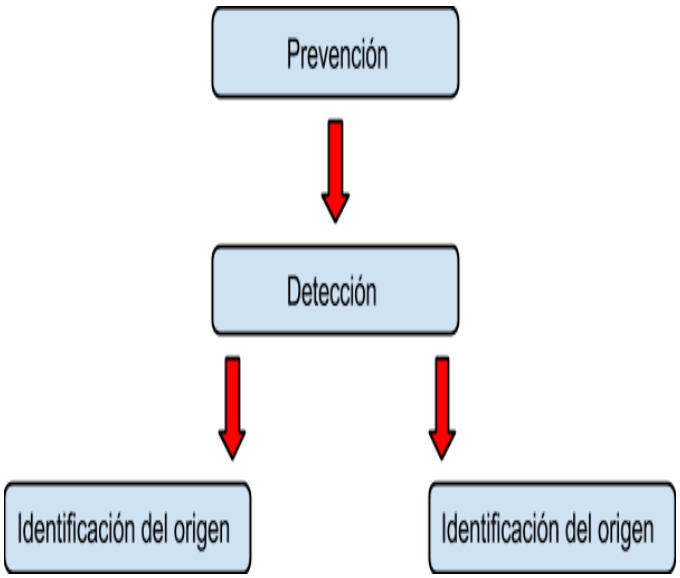

Figura 3.1: Distribución de acciones defensivas

A continuación se explica brevemente cada uno de estos grupos de acciones:

- **Prevención**: Las acciones de prevención de ataques tienen como objetivo minimizar el daño que producen dichas intrusiones. Se trata de un conjunto de medidas similares a las aplicadas para preservar la calidad de servicio del entorno protegido. En la actualidad han demostrado ser especialmente eficaces frente a amenazas sencillas. Sin embargo, a menudo deben afrontar nuevos desafíos, tales como la suplantación de direcciones IP (*IP Spoofing*), la dificultad del modelado del tráfico en entornos excesivamente heterogéneos, o la explotación de la organización de internet de manera descentralizada. Esto hace que sean poco efectivas en muchos casos de uso reales.

- **Detección**: Las acciones de detección tienen como objetivo distinguir las actividades maliciosas de las legítimas. Como en la prevención, existen varios hitos que en la actualidad siguen representando un auténtico desafío para la comunidad investigadora. Entre ellos destaca el problema de las altas tasas de falsos positivos o la distinción de eventos legítimos de características especialmente similares a los ataques, tales como los *Flash Crowds*. En [10] y [11] se explica en detalle este tipo de eventos y su impacto en las labores defensivas.

- **Mitigación**: Las acciones de mitigación comprenden las contramedidas a aplicar tras la detección de una intrusión: la gestión de cuellos de botella, reacciones intermedias en redes y las reacciones en el extremo final del ataque. La

gestión de cuellos de botella involucra la limitación de los recursos a fin de paliar los efectos de los ataques. En las reacciones intermedias tienen lugar los métodos convencionales de actuación, tales como el filtrado de tráfico o la construcción de listas de acceso. Finalmente, las reacciones en el extremo final del ataque involucran el seguimiento constante del tráfico bidireccional que fluye entre la red origen y el resto de internet. Por medio de la comparación periódica de las estadísticas medidas, con las de los modelos de flujos de tráfico normales, es posible determinar si el extremo está ejecutando un ataque, y en tal caso, evitar su propagación. En [7] , estas estrategias son discutidas en mayor detalle.

Para su complementación, algunos autores son partidarios de despliegues de seguridad conjuntos e integrados. Esto permite la cooperación de diversas acciones dirigidas contra amenazas de denegación de servicio. Una de las tendencias más frecuentes es la combinación de métodos de estrangulamiento del ancho de banda, con filtrado y marcado de paquetes. Estos esquemas se caracterizan por detectar el ataque cerca de las víctimas, y por la ejecución del filtrado de paquetes cerca del atacante. Según las características de la red, el nodo afectado podría avisar a sus nodos vecinos acerca de las intrusiones identificadas, permitiéndoles reaccionar con rapidez frente a posibles réplicas. Su finalidad demuestra sus carencias, ya que sólo filtran o limitan las tasas de paquetes maliciosos.

En [16] se propuso uno de los trabajos pioneros en éste campo, basado en la construcción de paquetes con características predefinidas que permiten evaluar su legitimidad. Pero a pesar de su popularidad, resultó ser poco efectivo frente a amenazas de gran magnitud. En [9] se trató este problema mediante la distribución de la carga de trabajo entre distintos elementos de red. Para ello se aplica la idea del *pushback* y el marcado de paquetes.

Otro importante aporte en este campo es [18], en el que se propone un marco para el intercambio de información y servicios entre los diferentes nodos de la red. Su finalidad es convertir arquitecturas de seguridad aisladas, en esquemas defensivos distribuidos. Dentro de ese marco, los nodos colaboran y cooperan para conseguir una defensa efectiva frente a ataques de gran magnitud, tratándose el mismo problema de [16] y [9].

En términos generales, los trabajos con tendencia a la hibridación presentan la ventaja de requerir bajos costes computacionales y de comunicación. Además presentan capacidad de filtrado de paquetes. Sin embargo son fácilmente evadibles, y raramente consiguen identificar la dirección del atacante original.

3.2. Defensa Frente Ataques DDoS

En la actualidad, la investigación en el área de la defensa frente a ataques de denegación de servicio se centra en las acciones de detección, identificación del origen y mitigación. Esto es debido a la habitual falta de efectividad de las acciones de prevención, y su escasa complejidad. Las propuestas que asumen como objetivo su detección, se basan principalmente en el análisis de tráfico que circula a través del entorno protegido, en busca de patrones de ataques conocidos o comportamientos anómalos. Para ello se han aplicado diferentes técnicas, tales como el análisis estadístico CUSUM con transformadas de ondícula (*wavelets*) [20], modelos probabilistas basados en Markov [21], teoría del caos [22], algoritmos genéticos [23], análisis forense mediante visualización [24], lógica difusa [25] o el estudio de variaciones en la entropía [26][27]. Estos últimos son de especial relevancia a la hora de comprender el trabajo realizado, ya que durante su transcurso se ha adoptado una métrica muy similar.

En [26] se demuestra que el análisis de las variaciones en la entropía del volumen de tráfico monitorizado se comporta de manera mucho más precisa que las otras estrategias, en la audición de redes que presentan características heterogéneas, tal y como sucede en la mayor parte de las redes actuales. Además se evalúa el uso de diferentes tipos de entropía. El estudio concluye con que la entropía de la información (o entropía de Shannon) es la que se comporta de manera menos restrictiva. Sin embargo, la entropía de Rènyi con índices de ajuste elevados (factor *alpha*) lo hace de manera más restrictiva, siendo propensa a la emisión de una mayor cantidad de falsos positivos. Por otro lado, en [27] se recopila una gran cantidad de propuestas relacionadas con la aplicación de la entropía en la denegación de servicio. Además se demuestra su vulnerabilidad frente a determinados ataques de evasión, basados en la adaptación de la inyección de tráfico malicioso a los patrones que presenta el modo de uso habitual de la red.

La mitigación de los ataques de denegación de servicio reúne una serie de aproximaciones adaptadas a la reducción total o parcial del daño causado por la intrusión. De entre los temas de mayor candencia destacan el uso de puzles para el reconocimiento de usuarios no humanos [28], trampas y señuelos [29], ampliación del ancho de banda, filtrado y protocolos de seguridad [30]. Los tres últimos también pueden ser desplegados como herramientas de acción preventiva. Para la identificación del origen de los ataques son frecuentes las propuestas orientadas al seguimiento de la trayectoria de los paquetes malintencionados. En [31] se recopila una gran cantidad de aproximaciones actuales, y se introduce un nuevo esquema de seguimiento uniforme. En [32] se estudia la influencia de las características de la topología de la red

en la eficacia de las estrategias de marcados de paquetes.

3.3. Defensa frente Ataques EDoS

Aunque algunas propuestas han tratado de acercar las métodos defensivos convencionales al entorno de computación en la nube, muy pocas de ellas han conseguido una diferenciación clara respecto de sus predecesoras. Un ejemplo claro de ello se encuentra en [33], donde se trata un tipo de ataque que combina peticiones HTTP con XML (HX-DoS). En esta aproximación son consideradas dos propuestas anteriores: la primera de ellas se encarga de valorar el riesgo imbuido en los mensajes recibidos [34], y en la segunda se decide su grado de legitimidad [35]. Otro ejemplo es [36], donde se propone la división del espacio de la nube, en diferentes zonas de riesgo. Cada una de ellas requiere de diferentes privilegios, de manera que las más restringidas solo sean disponibles para los accesos de mayor confianza. Asimismo, diversos autores han optado por evaluar sensores convencionales en el marco de sistemas *Cloud*. Este es el caso de aproximaciones como [37] y [38].

El principal punto de inflexión de esta tendencia tuvo lugar con la definición de un tipo de ataque de denegación de servicio complejo, y dirigido exclusivamente contra dicho entorno de computación: los ataques de denegación de sostenibilidad (EDoS). Su investigación comenzó en el año 2008 [1], aunque la mayor parte de las contramedidas propuestas se basaron en métodos convencionales.

En [39] fueron descritos como amenazas que tienen como objetivo denegar un servicio en la nube por medio de causar su insostenibilidad económica. De este modo explotan una de las mayores características de la computación *Cloud*: el pago por uso. En este trabajo además se discute en detalle su similitud con los ataques de denegación de servicio convencionales, y la efectividad de las diferentes estrategias defensivas.

En [31] su impacto es representado mediante modelos analíticos. Asimismo, en [32] se aplican experimentos sobre casos de uso reales capaces de demostrar el importante riesgo que conllevan. En su desarrollo fueron consideradas las métricas aplicadas en el equilibrado del proveedor Amazon EC2, y una serie de estadísticas relacionadas con su modo de empleo.

En [42] se propone la arquitectura EDoS-Shield basada en la combinación de cortafuegos virtuales (VF) con nodos de verificación (V-nodos). Los VF despliegan listas de acceso de diferentes características, mientras que los V-nodos se encargan de su mantenimiento. En la legitimación de direcciones participan pruebas basadas en puzles.

Otra propuesta basada en la resolución de rompecabezas es [35]. En ella los sistemas *Cloud* operan en dos modos, delimitados por el nivel de congestión de la red: normal y sospechoso. En el primero las peticiones son solicitadas con normalidad. Pero en el segundo los extremos deben resolver sencillos esquemas de cifrado mediante ataques por fuerza bruta, a fin de permitir su comunicación.

En [36] la defensa frente a EDoS se basa en la limitación del uso de recursos de cómputo. Esto implica la implementación de un importante entramado de elementos intermedios, tales como cortafuegos o sensores virtuales, capaces de hacer un seguimiento individualizado del modo de uso de cada máquina virtual.

La propuesta EDoS Armor [36] propone una solución doble a este problema: la admisión y el control de la congestión. Lo primero se consigue mediante la limitación del número de clientes que pueden enviar solicitudes al mismo tiempo. El control de congestión modifica las prioridades de accesos a determinados recursos en función de su naturaleza, contenido, y perfil de usuario.

Finalmente, en [38] se trata un tipo concreto de ataque EDoS, dirigido específicamente contra las primeras páginas (o índices) de los servicios *web*. De entre sus aportaciones destaca la demostración de que los ataques EDoS basados en peticiones HTTP-Get son los más efectivos, y la introducción de un marco de detección basado en el modelado del comportamiento legítimo de los usuarios en la colección de muestras publicada por la agencia norteamericana DARPA en el año 1999.

3.4. Metodologías de Evaluación

A lo largo de los años se ha publicado una gran cantidad de colecciones de muestras y metodologías para la evaluación de herramientas orientadas a la defensa frente a la denegación de servicio. Sin embargo, dada la dificultad de recolección y la actualidad de los ataques EDoS, ninguna de ellas se ha especializado en este tipo de amenaza. En consecuencia no existen criterios aprobados unánimemente por la comunidad investigadora para su evaluación.

Para medir la efectividad de las nuevas propuestas, la mayor parte de los autores ha optado por el uso de colecciones de capturas de ataques de dominio público (tales como KDD'99, DARPA'99, FIFA World Cup'98, etc.) o por aplicar herramientas de generación de tráfico capaces de imitan el comportamiento de los ataques (entre las que destacan D-ITG, Harpoon, *Curl-loader* y DDOSIM). En [49] se resume cada una de ellas, y es resaltada la poca calidad de los métodos de verificación que vienen usándose en la actualidad.

Esto es propiciado por el hecho de que que las capturas de ataques utilizadas

tradicionalmente son obsoletas, y muy dispares de los modelos de tráfico actuales. De entre ellas únicamente recomiendan el uso de CAIDA'07 [40]. Además, el uso de herramientas de simulación implica una importante pérdida de realismo, y complica la comparación de los resultados con los de sistemas defensivos similares [41]. Esto se complica todavía más con el hecho de que cada publicación tiende a aplicar sus propios criterios de evaluación.

Capítulo 4

Entropía y Series Temporales

En este capítulo se describen las dos técnicas conjuntas de detección que se han utilizado en el desarrollo e implementación de la propuesta: entropía, y análisis predictivo de series temporales. La entropía aplicada a la teoría de la información, mide su grado de incertidumbre. En esta propuesta forma parte de la métrica que determina la incertidumbre de los paquetes que llegan al detector. Por otro lado, las series temporales son definidas como secuencias de datos, observaciones o valores medidos en determinados momentos, ordenados de forma cronológica. Los datos se pueden estudiar en un grupo de intervalos iguales o desiguales en el tiempo. En esta propuesta permiten caracterizar la evolución de la entropía del tráfico monitorizado. A partir estas observaciones es posible la construcción de modelos predictivos capaces de reconocer situaciones anómalas.

4.1. Entropía

El concepto de entropía ha sido adaptado a distintos campos de investigación, como la termodinámica, mecánica estadística o la teoría de la información. Habitualmente se define como una medida del desorden o una medida de la incertidumbre de una determinada fuente de informacipn o datos. Además, en cualquier proceso permite acotar, reducir o eliminar la incertidumbre. En este caso ha sido tratada desde el punto de vista de la teoría de la información. La entropía asociada a la teoría de la información, más conocida como la entropá de Shannon, fue planteada por el matemático estadounidense Claude E. Shannon en el año 1948 [61] . En su desarrollo asumió las siguientes premisas:

- La medida de la información o los datos entrantes debe ser proporcional o lineal, es decir, un cambio pequeño de las probabilidades de aparición no debe modificar significativamente el resultado final de la entropía.

- Si todos los datos y variables de la información a tratar tienen la misma probabilidad de aparición, el valor de la entropía final será máxima.

Finalmente fue definida a partir de la siguiente expresión:

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$

donde H es el valor de la entropía, n el número de observaciones realizadas, y p_i la probabilidad de aparición de la observación i en el conjunto total de muestras. La definición calcula el logaritmo en base 2, asumiendo que la información a tratar es representada mediante codificación binaria. A continuación se describen algunas de sus propiedades más importantes:

- La entropía no puede ser negativa.
- Las probabilidades calculadas deben estar entre 0 y 1. Siendo estas las probabilidades de un tipo de dato entre el total de los datos.
- La entropía se encuentra acotada superiormente, evitando pérdidas.
- La entropía será máxima en el caso de que todos los valores tengan la misma probabilidad.
- Si una de las probabilidades que entran dentro del cálculo de la entropía es 0, esta será nula.
- El valor de la entropía es más grande cuando el paquete de datos X presenta una distribución uniforme, y es 0 si un dato tiene probabilidad 1. Para todas las otras distribuciones de X, la entropía vará entre 0 y el valor máximo de entropía.
- El máximo valor alcanzable es $\log n$ (siendo n cada número de términos diferentes), y el logaritmo con la misma base que la codificación en que se representa la información.
- El valor de la entropía es adimensional, es decir, carece de unidad.

En determinadas circunstancias, es recomendable el cálculo de la entropía normalizada. Esta es definida en la siguiente expresión:

$$H_N(X) = \frac{H(X)}{\log n}$$

Existen otros tipos de modelos (entropía) enmarcados en el área de la teoría de la información. A continuación se describen tres de los más frecuentes. Concretamente, la entropía de Rènyi, entropía de Hartley y entropía conservadora.

- **Entropía de Rènyi**: La entropía de Rènyi es una generalización que engloba varios tipos de entropía Shannon, entropía de colisión, Hartley y conservadora) [62]. Es definida mediante la siguiente expresión:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log(\sum_{i=1}^{n} p_i^\alpha)$$

  donde X es la variable aleatoria de la distribución de probabilidades de las observaciones. Por el mismo motivo que la entropía de Shannon, por defecto considera el logaritmo de base 2. La característica más importante de la entropía de Rènyi es su orden, parametrizado mediante el factor α, tal que $0 \leq \alpha$ y $\alpha = 1$. Cada posible valor de lleva a una particularización, destacando entre ella:

  - En el caso de $\alpha = 0$ corresponde con la entropía de Hartley.
  - En el caso de $\alpha = 1$ corresponde con la entropía de Shannon.
  - En el caso de $\alpha = 2$ corresponde con la entropía de colisión.
  - En el caso de $\alpha = \infty$ corresponde con la entropía conservadora.

- **Entropía de Hartley**: La entropía de Hartley fue propuesta por Ralph Hartley en el año 1928 [63]. Su cálculo depende directamente del número observaciones a partir de las que se construye la variable X. Para ello es considerada una muestra denominada A , que es la que se introduce en la siguiente formula, dando como resultado la entropía final:

$$H_0(A) = \log_b |A|$$

  A diferencia de la entropía de Shannon, Hartley aplica la base del logaritmo 10. Entre sus características destaca la relación entre la incertidumbre con el conjunto muestral: a mayor incertidumbre, mayor conjunto de muestras.

- **Entropía conservadora**: La entropía conservadora (Min-Entropy) Es la particularización de la entropía de Rènyi con valor más bajo [64]. Al igual que

sus predecesoras mide la imprevisibilidad de una distribución no uniforme. Es definida de la siguiente manera:

$$H_{min}(A|B)_\rho = -inf_{\sigma B} D_{max}(\rho_{AB}||I_A \otimes \sigma_B)$$

De entre sus propiedades destaca que nunca supera a la entropía de Shannon ni a la entropía máxima de Hartley. De ahí el nombre de entropía conservadora.

4.2. Series Temporales

Las series temporales son definidas como colecciones de observaciones de una variable recogidas secuencialmente en el tiempo. A menudo se usan para estudiar la relación causal entre diversas variables que cambian con el tiempo y se influyen entre sí. Cuando la esperanza matemática de dichas variables aleatorias es constante o varía de manera cíclica, se dice que la serie es estacionaria y no tiene tendencia secular. Su análisis clásico se basa en la suposición de que los valores que toma la variable de observación es la consecuencia de cuatro componentes, cuya actuación conjunta da como resultado los valores medidos. Los componentes son:

- **Tendencia secular**: Indica la marcha general y persistente del fenómeno observado. Es un componente de la serie que refleja la evolución a largo plazo.
- **Variación estacional:** Componente causal debido a la influencia de ciertos fenómenos que se repiten de manera periódica. Recoge las oscilaciones que se producen en esos períodos de repetición.
- **Variacion ciclica**: Componente que recoge las oscilaciones periódicas de amplitud superiores. Se caracteriza por movimientos normalmente irregulares alrededor de la tendencia, en los que a diferencia de las variaciones estacionales, existen período y amplitud variables, pudiendo clasificarse como cíclicos, cuasicíclicos o recurrentes.
- **Variacion aleatoria**: También denominada residuo, no muestran ninguna regularidad (salvo las regularidades estadísticas), debidos a fenómenos de carácter ocasional.
- **Variacion trasciente**: Componente de carácter accidental y errático debido a fenómenos aislados que son capaces de modificar el comportamiento de la serie.

La manera habitual de aprovechar la información que ofrecen las series temporales, es la construcción de modelos capaces de pronosticar su evolución en futuras observaciones. A continuación se describen dos de las estrategias de predicción más importantes: los modelos autorregresivos integrados de media móvil (ARIMA) y el triple alisamiento exponencial (Holt-Winters).

4.2.1. Modelos ARIMA

La metodología de los modelos ARIMA fue formalizada por Box y Jenkins en 1976, por lo que usualmente se les denomina modelos Box-Jenkins [56]. Este modelo viene determinado por el hecho de que una serie temporal que se trata de predecir es generada por un proceso estocástico (termino estadístico que define el concepto matemático para caracterizar una sucesión de variables aleatorias que varían en función del tiempo habitualmente) generalizado en un modelo. También asume que la serie es univariante, lo que conlleva las siguientes consecuencias.

- Utilizando este método, se da la ventaja de no necesitar distintas series de datos, es decir usar distintas variables referidas a un mismo periodo de tiempo.
- Al no incluir un conjunto más amplio de variables explicativas, no se atiende a las relaciones que existen entre toda las variables que se hayan utilizado en el diseño del modelo, perdiendo la capacidad de efectuar un estudio más completo y detallado. Esto deja de lado parte del estudio previo realizado y su uso posterior, por lo que es aquí donde erradica su principal desventaja.

El nombre genérico de ARIMA viene determinado por A (autoregresivo), I (Integrado) y MA (medias Móviles), y a menudo es denotado mediante $ARIMA(p, d, q)$, donde p es el número de parámetros autoregresivos, d es el número de diferenciaciones para que la serie sea estacionaria y q es el número de parámetros de medias móviles.

Para realizar la estimación mediante un modelo ARIMA se requiere de una serie temporal que cuente con un elevado número de observaciones. Dado que estos modelos se construyen en las bases de ARMA, es importante conocer su composición.

El modelo $ARMA(p, q)$ viene representada por la siguiente ecuación:

$$Y_t = \phi_0 + \phi_1 y_{t=1} + L + \phi_p y_{t=p} + a_t - \theta_1 a_{t=1} - L - \theta_q a_{i=q}$$

donde la parte autorregresiva (AR) del modelo es $\phi_1 y_{t=1} + L + \phi_p y_{t=p}$ y la parte de medias móviles (MA) es $-\theta_1 a_{t=1} - L - \theta_q a_{i=q}$. Los coeficientes de los parámetros ϕ_0, ϕ_1, L, ϕ_p, θ_1, θ_p son calculados mediantes métodos estadísticos.

La metodología básica para la construcción del modelo predictivo es ilustrada en Fig. 4.1, y sigue los siguientes pasos:

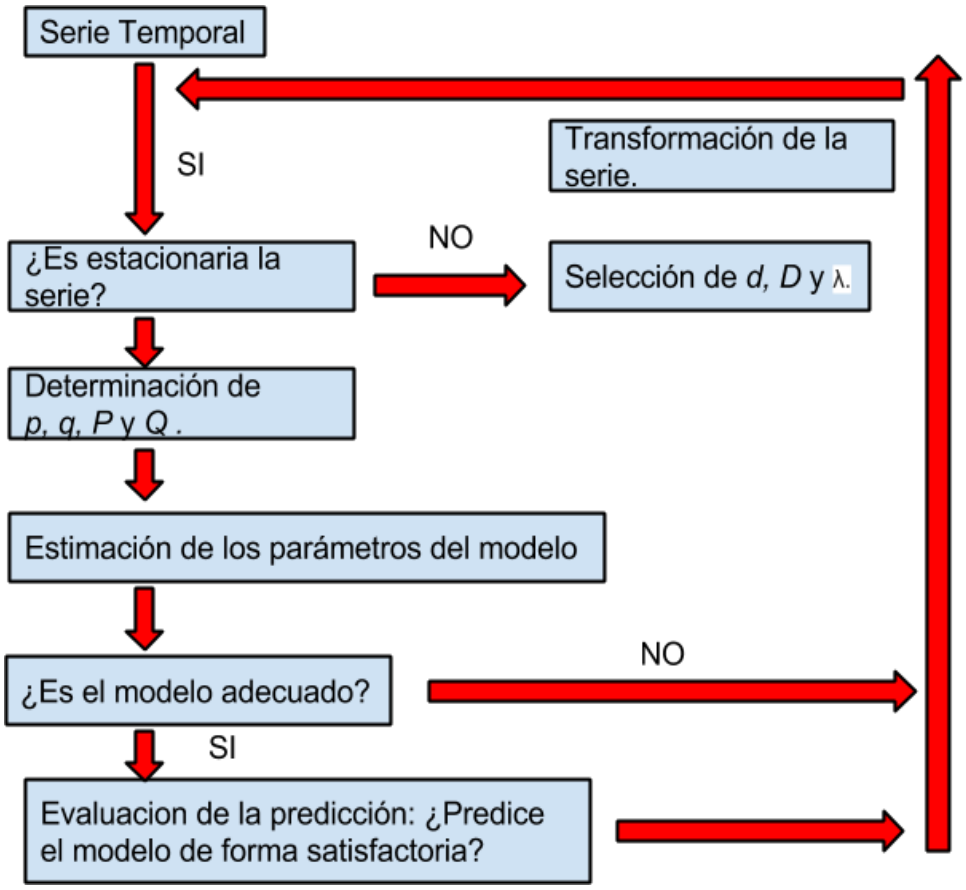

Figura 4.1: Descripción gráfica de la metodología en 4 fases de *Box-Jenkins*

- Identificación del posible modelo ARIMA asociado a la serie temporal. Para ello:
  - La serie observada debe convertirse en una serie estacionaria, tras la ejecución de diferentes transformaciones.
  - A partir de los parámetros p y q se construye un modelo ARMA asociado a la serie estacionaria.
- Estimación de los componentes AR y MA por el método de máxima verosimilitud. En este paso se conocen los errores y residuos del propio modelo.
- Análisis de la parte residual con el fin de detectar dependencias entre ellos o que siguen un proceso de ruido blanco. Esta última situación se caracteriza con que

sus valores de señal en dos tiempos distintos no guardan ninguna correlación estadística. En el caso de que los residuos muestren un cierta estructura, se repiten los pasos anteriores.

- Predicción a partir del modelo construido y evaluación. Si el error de predicción es elevado, se repite el proceso.

4.2.2. Triple Alisamiento Exponencial

Las series temporales también pueden ser analizadas a partir de métodos de descomposición, donde se destaca el triple alisamiento exponencial, conocido como método de Holt-Winters [65]. En términos generales, los algoritmos de descomposición separan uno o varios componentes de la serie tratada, siendo su tendencia, factor cíclico, estacionalidad y componente irregular, los más utilizados. Según la descomposición que se realice, el resultado final del análisis dependerá de la integración de los distintos componentes en distintos modos. Los modos más usados son el aditivo y el multiplicativo. El primero se lleva a cabo mediante la suma de los componentes extraídos. Las fluctuaciones que se producen en la serie no se ven afectadas por la tendencia. Por otro lado, el modo multiplicativo calcula el producto de los elementos separados. En este caso, las fluctuaciones de la serie varían con la tendencia. El método Holt-Winters arroja resultados más precisos cuando la serie temporal tratada presenta una tendencia más o menos lineal, y contiene factor estacional. Su versión aditiva es representada mediante el siguiente sistema:

$$a_t = \alpha(Y_t - S_{t-p}) + (1-\alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1-\beta)b_{t-1}$$

$$S_t = \gamma(Y_t - a_t) + (1-\gamma)S_{t-p}$$

donde a_t representa el valor de alisado en la parte constante de t, b_t representa el alisado en la tendencia y S_t el alisado en la parte estacional. Asimismo, Y_t es la observación en el periodo de tiempo t y p el número de estaciones por año. Los factores de ajuste son α, β y γ , tales que $0 < \alpha$, β, $\gamma < 1$, donde α lo es para la constante, β lo es para la tendencia y γ lo es para el componente estacional. La predicción se calcula mediante la expresión:

$$\hat{Y} = a_\tau + \tau b_\tau + S_\tau$$

Donde x el número de observaciones que serán pronosticadas, τ es el número de observaciones realizadas, a_τ es el valor del alisado de la constante en τ, b_τ el valor

del alisado de la tendencia en τ y S_τ el valor del alisado de la parte estacional en τ. Uno de los principales problemas de Holt-Winters es la necesidad de inicializar los valores a_t, b_t y S_t. La manera más aceptada de hacerlo en el modelo adictivo sigue las siguientes ecuaciones:

$$a_p = \frac{1}{p}(Y_1 + Y_2 + K + Y_p)$$

$$b_p = \frac{1}{p}[\frac{Y_{p+1} - Y_1}{p} + \frac{Y_{p+2} - Y_2}{p} + K + \frac{Y_{p+p} - Y_p}{p}]$$

$$S_1 = Y_1 - a_p;\ S_2 = Y_2 - a_p;\ K;\ S_p = Y_p - a_p$$

El método aditivo es especialmente eficaz cuando la serie presenta tendencia y su componente estacional es aditiva. Sin embargo, en el caso de que esta última sea multiplicativa es recomendable el uso del método multiplicativo. Este es definido mediante el siguiente sistema:

$$a_t = \alpha(\frac{Y_t}{S_{t-p}}) + (1 - \alpha)(a_{t-1} + b_{t-1})$$

$$b_t = \beta(a_t - a_{t-1}) + (1 - \beta)b_{t-1}$$

$$S_t = \gamma(\frac{Y_t}{a_t}) + (1 - \gamma)S_{t-p}$$

Su predicción es calculada mediante:

$$\hat{Y} = (a_\tau + \tau b_\tau)S_\tau$$

Y la iniciación es prácticamente igual a la del método adictivo, a excepción de los siguientes cambios:

$$S_1 = \frac{Y_1}{a_p};\ S_2 = \frac{Y_2}{a_p};\ K;\ S_p = \frac{Y_p}{a_p}$$

Holt-Winters en muchos casos ha demostrado ser superior a ARIMA en rendimiento, e incluso en la precisión de sus predicciones a corto plazo. Además necesita una menor cantidad de observaciones para ser capaz de realizar predicciones con éxito. Sin embargo, en el resto de casos ARIMA tiende a comportarse con mayor precisión en cualquier otro tipo de pronósticos. Nótese que de manera excepcional, estas situaciones pueden variar en función del contexto y la naturaleza de la serie temporal.

Capítulo 5

Detección de Ataques EDoS sobre Entornos Cloud

En este capítulo se explica en detalle el sistema propuesto. Para ello se hace especial hincapié en su desarrollo e implementación, así como en todos aquellos factores y etapas que han llevado a su finalización y buen funcionamiento. A continuación se describe su arquitectura, y cada uno de sus componentes. Estos engloban desde la captura del tráfico de la red protegida, hasta la emisión de las alertas finales.

5.1. Arquitectura

La arquitectura de la propuesta consta de seis componentes, organizados tal y como se muestra en Fig.5.1. El primero de ellos se encarga de la extracción de las características principales del tráfico monitorizado por medio de un *parser*. A continuación, otro componente se encarga del cálculo de la métrica principal del analizador: la entropía del volumen de tráfico. Esta es tratada en una siguiente etapa, en forma de series temporales, lo que permite la aplicación del modelo predictivo Holt-Winters para el pronóstico de futuras observaciones. A continuación, la calidad de las predicciones es determinada por su proximidad a intervalos de predicción. La identificación de errores de predicción revela la existencia de anomalías, lo que permite que el componente encargado de la toma de decisiones etiquete las observaciones como legítimas o maliciosas. Finalmente, una última etapa se encarga de dar formato a las alertas, antes de ser emitidas al operador.

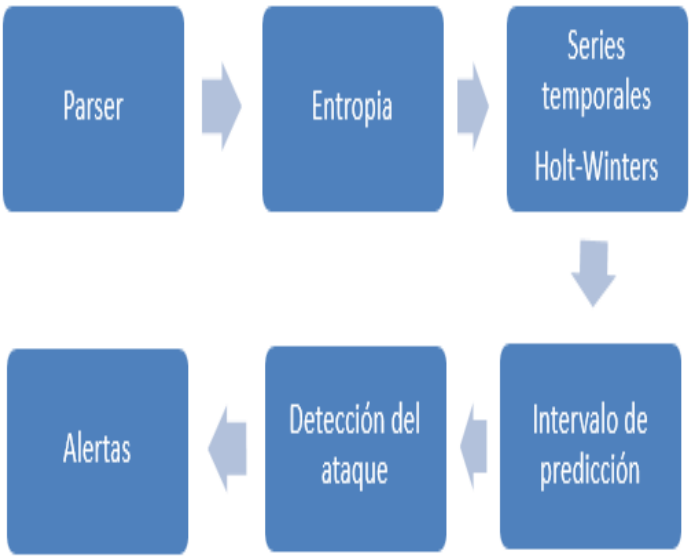

Figura 5.1: Arquitectura del sistema de detección

5.2. Extracción de la Información

La etapa de extracción de la información, realiza la primera toma de contacto con la información a analizar. Durante su transcurso, el tráfico real que fluye hacia el entorno protegido es capturado. A continuación se realiza su *parseo*, es decir, la traducción del tráfico entrante al lenguaje y estructuras de datos que utiliza el detector. Esta tarea se lleva a cabo considerando únicamente la estructura de los paquetes TCP donde se encuentran alojadas las tramas HTTP. La elección de este tipo de tráfico se justifica mediante la consulta a los distintos trabajos de la bibliografía especializados en ataque **EDoS**, tales como [48] [49]. En ellos se señala a las peticiones HTTP, como principal foco de este tipo de amenazas.

El proceso de *parseo* accede a cada una de las partes del datagrama donde se encuentra la información necesaria para su estudio. Dentro del paquete se encuentra el encapsulamiento de encabezados Ethernet, IP y TCP, tal y como se muestra en Fig.5.2. De su contenido IP se obtienen las direcciones IP origen y destino, y el puerto del destinatario (en tráfico HTTP, este último es 80). De la cabecera Ethernet se extraen datos como su tamaño o las direcciones MAC origen y destino. Además, cada encabezado permite la extracción de metadatos, que ayudan a la validación de la información extraída. Algunos de ellos son:

- Los tipos de protocolos de encabezados encapsulados, los cuales permiten comprobar si el datagrama presenta algún contenido HTTP.

- El tamaño de cada encabezado para realizar comprobaciones de errores.
- El tamaño total del paquete.

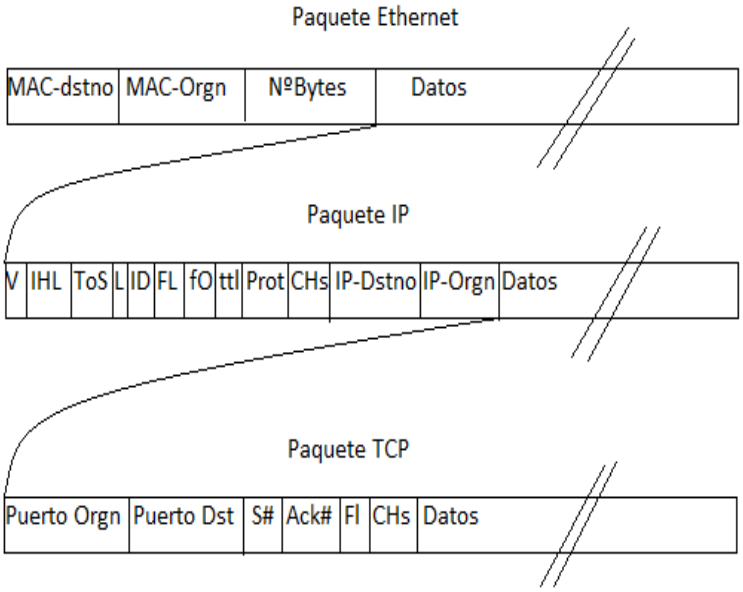

Figura 5.2: Encapsulado de encabezados TCP/HTTP.

El sistema tiene la capacidad de extraer información, tanto de capturas de tráfico reales en tiempo real, como de colecciones de trazas previamente capturadas. Esto último se consigue mediante su compatibilidad con el formato .pcap, el cual constituye una de las representaciones más frecuentes en los conjuntos de muestras de dominio público.

El *parser* ha sido implementado utilizando la librería JNetPcap de JAVA y con Libpcap para C, utilizando finalmente la implementación en JAVA, donde:

- **LibPcap** [48]: LibPcap es una librería de código abierto escrita en C que ofrece al usuario un conjunto de métodos y funcionalidades con las que poder capturar paquetes de la capa de red, así como la manipulación de ficheros .pcap. Es portable a un gran número de sistemas operativos. En Fig.5.3 se ejemplifica las fases que sigue para la extracción de información: inicialización de las estructuras de datos y métodos para desarrollar el programa, indicación del filtrado de los paquetes que son recogidos, captura y extracción de datos.

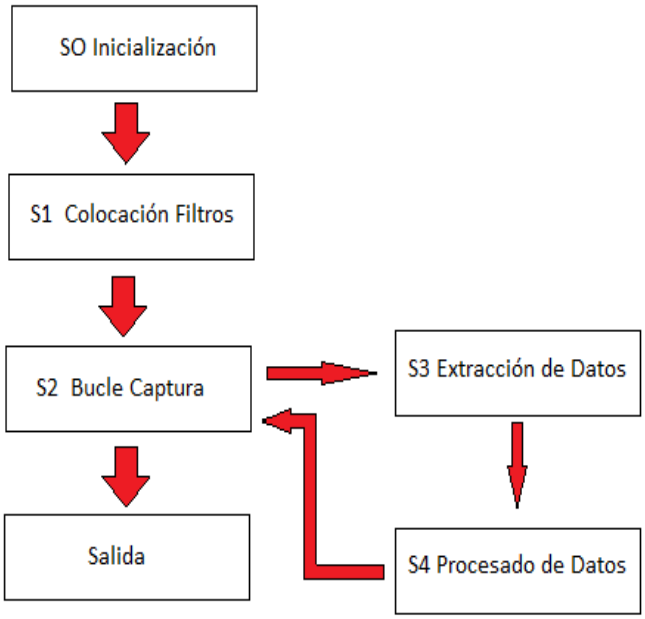

Figura 5.3: Esquema uso general Libpcap.

- **JnetPcap** [48]: JnetPcap es una librería de código abierto escrita en Java. Permite descodificar un conjunto de paquetes de distintos protocolos capturados en tiempo real. Esta librería utiliza una mezcla nativa y Java para un rendimiento óptimo apoyándose en WinPcap, Fig.5.4. Contiene licencia LGPL.

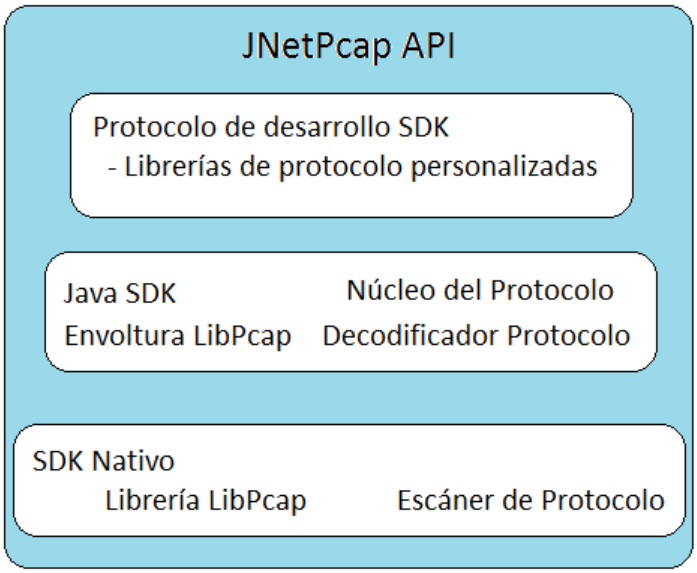

Figura 5.4: Arquitectura principal de jNetPcap.

5.3. Métrica: Entropía de Shannon

En esta segunda fase se calcula el grado de incertidumbre de la información observada. Para ello se aplica la entropía de la información (o entropía de Shannon). De entre todas las entropías estudiadas se ha elegido la de Shannon por su menor tendencia a la emisión de falsos positivos [48]. Se asume que este es uno de los mayores problemas en la detección basada en anomalías, dado que las amenazas DDoS son altamente visibles. De este modo, y dada su menor grado de restricción, se espera compensar el error en la identificación de ataques verdaderos con aquel que se obtiene al procesar tráfico legítimo.

En este caso la implementación del algoritmo se ha realizado tanto en Java como en C. En la experimentación se ha utilizado la de java por su sincronización con el *parser*, realizado en el mismo lenguaje. Los datos entrantes a la entropía, y a partir de los cuales se calcularán las distintas probabilidades para su desempeño, consistirán en tuplas: IP origen, IP destino y puerto. De este modo, cada tupla representa un individuo de la población observada. Cada vez que una tupla aparece en una observación, se acumula su frecuencia de aparición. A partir de ella es posible calcular la probabilidad que ocupa en la distribución de la muestra. Cada tipo posible de tupla es una posible instancia de la variable aleatoria X en la entropía de Shannon.

Es importante destacar que el número de paquetes a partir de los cuales es calculada la entropía de cada observación, puede determinar la precisión del sistema. En entornos de monitorización muy homogéneos a menudo es suficiente con una cantidad pequeña (entre 50-200). Sin embargo, es un contexto heterogéneo se requieren muchos más (normalmente más de 1000). Este segundo caso afecta a la velocidad de reacción del sistema, ya que es necesario esperar a la llegada de más información, para tomar decisiones. Esto lleva a la conclusión de que el sistema se comporta con mayor eficiencia en entornos de audición más homogéneos.

5.4. Predicción mediante Holt-Winters

Mediante la aplicación del algoritmo predictivo Holt-Winters es posible pronosticar la entropía de futuras conexiones en un cierto intervalo de tiempo. Esto es debido a que conocer la tendencia que va a adquirir el desorden de tráfico, permite reconocer comportamientos inesperados en la red, los cuales a menudo son indicios de actividades maliciosas. Los parámetros de ajuste de Holt-Winters en el sistema, son los siguientes:

- Y_t: Observación en instante de tiempo t.
- α: Coeficiente de suavizado a nivel
- β: Coeficiente se suavizado de tendencia.
- γ: Coeficiente de suavizado de sesión.
- **Periodo**: Calentamiento de la serie temporal. Los datos de una serie completa vienen determinados por periodos, y se debe estimar el factor de tendencia de un periodo a otro.

La versión de Holt-Winters implementada coincide con el modelo aditivo descrito en el capítulo anterior. En ella se requiere de un periodo de calibrado (delimitado por la variable periodo), para la inicialización del sis-tema de ecuaciones. El sistema es configurado con los parámetros α, β y γ, cuyo valor depende del caso de uso. El pronóstico realizado solo tiene en cuenta la siguiente observación, es decir, considerando que el instante actual es t, se estima la entropía en Y_{t+1}.

5.5. Intervalos de Predicción

Los intervalos de predicción actúan como dos umbrales (superior e inferior), cuyo rebasamiento revela la presencia de comportamientos inesperados, y por lo tanto, anómalos. Son construidos a partir de las siguientes expresiones:

$$USup(t) = p_0 + \sqrt{var(En_t)}$$
$$Uinf(t) = p_0 - \sqrt{var(En_t)}$$

donde p_0 es la predicción en el la última observación realizada, y En_t el error de predicción en t. Este último viene dado por la diferencia entre la predicción y la observación en t. Nótese que la varianza es calculada a partir del error de predicción en las últimas t observaciones. Asimismo, la expresión incluye un parámetro de ajuste K, el cual permite regular el nivel de restricción del sistema.

5.6. Toma de Decisiones: Detección del Ataque

La decisión de si una observación en t corresponde con actividades legítimas o maliciosas tiene en cuenta los umbrales de predicción de ese mismo instante de tiempo. Cuando la diferencia que existe entre su entropía y el valor en el umbral

superior es mayor que 0, se considera un comportamiento anómalo; en consecuencia se emite una alerta. Sucede lo mismo en el caso en que la observación presenta un valor menor que el del umbral inferior en t. Sin embargo, si la observación se sitúa entre ambos umbrales, representa un comportamiento esperado. En este caso, se interpreta que corresponde con el modo de uso habitual, y por lo tanto legítimo de la red, no siendo indicador de riesgo.

5.7. Emisión de Alertas

Una vez detectado un ataque, es notificado por el sistema vía consola de comandos. Al analizar un *dataset*, la herramienta hace especial hincapié en dos valores: tasa de acierto y tasa de falsos positivos. La primera indica la cantidad de ataques verdaderos reconocidos con éxito, mientras que la segunda, la cantidad de trazas legítimas etiquetadas erróneamente como amenazas. Por lo tanto, un sistema de detección ideal alcanza una tasa de acierto del 100 % y una tasa de falsos positivos del 0 %.

El sistema también genera logs de alertas, cuyo formato es mostrado en Fig.5.5. En ella se muestra la precisión del sistema y las trazas que han desencadenado alertas, en un experimento con trazas legítimas de la colección CAIDA'07.

```
porcentaje falsos positivos: 23.444977
porcentaje aciertos:76.55502
Ataques:
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte621;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte624;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte625;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte628;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte662;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte665;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte667;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte668;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte670;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte673;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte676;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte678;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte680;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte681;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte693;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte695;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte696;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte697;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte701;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte705;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte707;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte711;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte712;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte720;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte727;
Datasets CAIDA 2007/Legitimo/equinix-chicago.dirA.20130529-125710.UTC.anon_parte728;
```

Figura 5.5: Muestra de los resultados de un caso de prueba CAIDA 2007.

Capítulo 6

Experimentación

En este capítulo se explican los aspectos más relevantes de la experimentación realizada, destacando su implementación, los conjuntos de muestras a analizar, y la metodología de evaluación aplicada.

6.1. Implementación y Escenarios de Pruebas

El proceso de implementación del sistema de detección propuesto ha sido el siguiente: el primer lugar se desarrolló un prototipo inicial en C. Este capturaba tráfico mediante un *parser* basado en la popular librería libpcap [66]. Era capaz de calcular la entropía de las observaciones realizadas, entendiendo por observaciones, secuencias de paquetes monitorizados en un cierto intervalo de tiempo. También permitía la creación de pronósticos mediante la implementación del método Holt-Winters.

Sin embargo, debido a problemas de portabilidad, la herramienta fue migrada a Java, lo que conllevó su reconstrucción bajo el entorno de trabajo *Netbeans* [74]. En esta ocasión, la librería que permitió la captura de tráfico fue JnetPcap [67]. Al prototipo inicial se le añadieron mejoras relacionadas con la inicialización de las funciones de Holt-Winters, y con la elaboración de intervalos de predicción. Dado que esta ha sido la versión con la que se ha experimentado, también ha sido dotada de un módulo encargado de sintetizar y formatear las alertas emitidas, así como de la representación visual de las series temporales por medio de la librería JFreeChart [69]. El despliegue de la propuesta se ha llevado a cabo en los siguientes escenarios de pruebas:

- **Entorno Local**: La experimentación en entorno local ha considerado capturas de tráfico de dominio público. Por lo tanto, la información procesada son archivos que resumen los datos necesarios de cada paquete. Este entorno consta de un solo equipo con el sistema de detección instalado, el cual accede uno a uno a dichos ficheros, y emite su etiquetado.

- **Entorno Cloud**: Para evaluar la eficacia de la propuesta en la nube, ha sido instalada sobre un *sandbox* de la plataforma OpenNebula [70]. Esta permite la gestión de un centro de procesamiento de datos virtual, de características públicas, privadas o híbridas, que ofrece infraestructura como servicio (SaaS). Es un *software* libre y de código abierto sujeto a la licencia Apache 2. Se ha aplicado VirtualBox como hypervisor donde insertar la máquina virtual de OpenNebula, la cual presenta sistema operativo CentOS 6. Para la gestión de la máquina virtual se utiliza la interfaz host que ofrece OpenNebula, llamada Sunstone. Esta permite observar su configuración, y lanzar varias máquinas virtuales desde dentro del *Cloud*. Finalmente, se configuró la máquina virtual para asignarle una IP automática mediante el protocolo DHCP. Esto se llevó a cabo cambiando el adaptador de red a: adaptador solo – anfitrión, quedando así a plena disposición para el resto de la experimentación.

6.2. Colección de Muestras

Para la verificación de los resultados obtenidos por la herramienta implementada, se han utilizado tres colecciones de muestras. Las dos primeras son de dominio público (CAIDA'07 y Kdd-Cup'99), mientras que a última (ON-EDoS) reúne tráfico real capturado en el escenario de pruebas sobre entorno *Cloud*. A continuación se describe cada una de ellas:

- **CAIDA'07** [49]: Este conjunto de datos ha sido respaldado en numerosas ocasiones a lo largo de la bibliografía. Se trata del estándar funcional de validación de propuestas para la defensa frente a ataques de denegación de servicio más actualizado. Contiene aproximadamente una hora de trazas de tráfico anónimo de un ataque DDoS, detectado el 4 de agosto de 2007 en el centro de supercomputación Equinix de San Diego (Estados Unidos). La traza de una hora se divide en archivos .pcap de 5 minutos. El tamaño total del conjunto es de 5,3GB (sin comprimir: 21 GB). El flujo de ataque contra la víctima y la respuesta de la víctima se encuentra incluida en las trazas del ataque. La carga útil ha sido eliminada de todos los paquetes.

- **kdd-Cup'99** [71]: La colección KDD-Cup'9 fue publicada en el marco del evento *"Third International Knowledge Discovery and Data Mining Tools Competition"* del año 1999, en el que diversos participantes compitieron por desarrollar el sistema de detección de intrusiones más preciso. Sus muestras son una nueva versión de la colección DARPA'98, capturadas por la Agencia de Proyectos de Investigación Avanzados de Defensa estadounidense (DARPA) en colaboración con los MIT Lincoln Labs, en el año 1998. Contiene diferentes tipos de amenazas, entre las que se incluyen cinco ejemplos de ataques de denegación de servicio. De ellos han sido elegidos dos para esta experimentación: *smurf* y *neptune*. Esto es debido a que el resto tiene escasa representación. En la actualidad, Kdd-Cup'99 es considerada una colección "comprometida", dadas ciertas irregularidades en los procesos de captura. Pero a pesar de ello, sigue siendo un referente, necesario para comparar las nuevas propuestas con las antiguas.

- **ON-EDoS**: La colección ON-EDoS ha sido creada exclusivamente para la evaluación de esta propuesta. Contiene muestras de trazas de tráfico legítimo y malicioso capturadas en el escenario de pruebas basado en OpenNebula. El tráfico atacante se realizó a partir de una combinación de la herramienta de inyección de solicitudes LOIC (*Low Orbit Ion Cannon*) [72], con varios scripts en JAVA que implementan las características propias de los ataques EDoS. De este modo se consigue una gran cantidad de usuarios con sesiones legítimas inicializadas en el servidor en la nube, capaces de desbordar su capacidad de cómputo a partir de la inyección de solicitudes HTTP (Get/Post), y de este modo, forzar la adquisición de nuevos recursos. Para generar tráfico legítimo se utilizaron varios scripts de comportamiento similar al de los populares *Web-crawlers*, pero que emiten peticiones GET/POSTPOST, alternando los dos métodos, y colocando varias hebras o paradas entre ellos, simulando comportamientos diferentes.

6.3. Metodología de Evaluación

La metodología de evaluación aplicada implica la combinación de pruebas con colecciones de muestras públicas aprobadas por la comunidad investigadora, con un caso de uso real, en el que efectivamente se trata el problema de los ataques EDoS. De este modo, el primero permite validar la precisión de la propuesta a partir de su comparación con trabajos previos. Por otro lado, es posible demostrar que el sistema ha sido capaz de reconocer los ataques EDoS a los que se ha enfrentado.

El principal criterio de evaluación aplicado es su precisión, tanto en la identificación de ataques verdaderos (tasa de acierto) como en el rechazo de tráfico legítimo (tasa de falsos positivos). Por lo tanto es deseable alcanzar una tasa de acierto elevada, y una tasa de falsos positivos baja. Dado que el sistema presenta diferentes parámetros de ajustes, es importante valorar en qué manera la variación del nivel de restricción de la propuesta ha afectado a dichas tasas. Esto ha sido realizado por medio de su representación en el espacio ROC (*Receiver Operating Characteristic*), donde el eje X indica la evolución de las tasas de falsos positivos, y el eje Y las de acierto. Esto permite su valoración general en base al estudio del área bajo la curva ROC (AUC).

Capítulo 7

Resultados de la Experimentación

En este capítulo se describen los resultados de la experimentación realizada. Para su mayor comprensión, han sido divididos en dos bloques: resultados al analizar colecciones de muestras públicas, y resultados al analizar ataques EDoS. A continuación se describe en detalle cada uno de ellos.

7.1. Muestras Públicas

Tal y como se indicó en la metodología de evaluación, se ha experimentado con un *datasets* de dominio público: CAIDA'07. Los resultados son descritos a continuación.

7.1.1. Resultados con CAIDA'07

En la experimentación con CAIDA'07, el nivel de restricción del sistema ha sido calibrado mediante la variación del factor de ajuste K de los intervalos de predicción. Asimismo se realizaron variaciones en los valores principales de la predicción de Holt-Winters (*Alpha*, *Beta* y *Gamma*), es decir, el conjunto de variables entrantes al modelo. También se ha modificado el número de paquetes que engloba cada observación, aunque en la mayor parte del proceso, este estuvo fijado en 50 paquetes/observación. El periodo de calentamiento inicial de las ecuaciones de Holt-Winters se mantuvo constante durante toda la fase de pruebas.

En general, al monitorizar ataques la herramienta se comportó tal y como se observa en Fig. 7.1. En ella se muestra un ejemplo de su reacción ante uno de los ataques, donde el eje X indica el instante de tiempo en que se han producido observaciones, y el eje Y el valor de la entropía de los paquetes que reúnen. La evolución de la entropía (en rojo) se muestra poco estable, con importantes variaciones en la incertidumbre del tráfico. A lo largo del ataque son superados los umbrales supe-

riores (en verde) e inferiores (azul), emitiéndose diversas alertas. Se han probado diferentes configuraciones de parámetros. En Fig. 7.3 se muestra una de las hojas de *Excel* con que fueron puestos en común, tomando como criterio la variación de K.

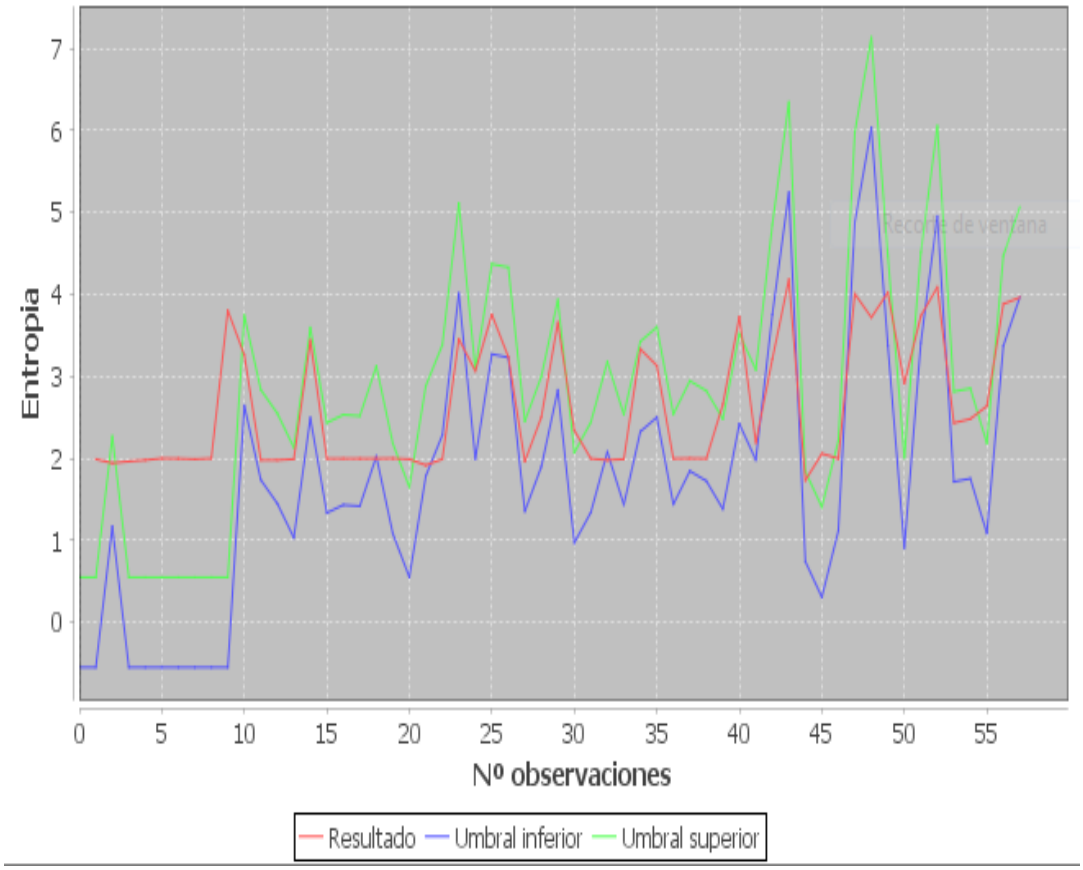

Figura 7.1: Ejemplo de análisis de tráfico malicioso en CAIDA'07.

La comparación de los resultados obtenidos ha permitido la representación en el espacio ROC de la tendencia observada. Esta se muestra en Fig. 7.3, donde el eje X indica la tasa de falsos positivos y el eje Y la de acierto. La situación más deseable se alcanza cuando la curva se sitúa con mayor proximidad a la posición (1,0). Este es el caso en que todos los ataques son reconocidos con éxito, pero ninguna traza legítima ha sido etiquetada erróneamente. Se han obtenido valores muy próximos a este punto.

A partir de la curva ROC es posible concluir que la herramienta es poco sensible a los parámetros de ajuste. Como se observa en su parte inicial, se produce una mejora muy rápida en la detección de ataques verdaderos al aumentar la restricción, sin apenas afectar a la tasa de falsos positivos. Este valor se estabiliza cerca del 98 % y no varía, a pesar de seguir aumentando la restricción.

					Observaciones		LEGITIMO	ATAQUE
Aplha	Beta	Gamma	Valor Umbra	Calentamiento			% Falsos Pos	ACIERTOS
0,7	0,7	0,7	0,23	10	50		49	100
0,7	0,7	0,7	0,24	10	50		44	100
0,7	0,7	0,7	0,26	10	50		30,62	98,11
0,7	0,7	0,7	0,28	10	50		22	98,11
0,7	0,7	0,7	0,3	10	50		17	98,11
0,7	0,7	0,7	0,33	10	50		9,5	98,11
0,7	0,7	0,7	0,35	10	50		7,6	98,11
0,7	0,7	0,7	0,36	10	50		7,1	98,11
0,7	0,7	0,7	0,37	10	50		6,6	98,11
0,7	0,7	0,7	0,38	10	50		5,7	98,11
0,7	0,7	0,7	0,39	10	50		4,7	98,11
0,7	0,7	0,7	0,41	10	50		3,34	98,11
0,7	0,7	0,7	0,45	10	50		2,87	98,11
0,7	0,7	0,7	0,5	10	50		1,4	98,11
0,7	0,7	0,7	0,54	10	50		0,95	98,11
0,7	0,7	0,7	0,55	10	50		0,47	94,3
0,7	0,7	0,7	0,59	10	50		0	94,3
0,7	0,7	0,7	0,6	10	50		0	94,3
							eje x	eje Y

Figura 7.2: Resultados de diferentes configuraciones en CAIDA'07.

La configuración óptima del detector se alcanzó con los siguientes valores:

- $\alpha = 0{,}7$.
- $\beta = 0{,}7$.
- $\Gamma = 0{,}7$
- $K = 0{,}54$.
- periodo $= 10$.
- Número paquetes/observación $= 50$.

Con esta se obtuvo una tasa de acierto próxima al 98 %, y un error de detección del 0,95 % al procesar muestras de tráfico legítimo, lo que prácticamente iguala a las mejores propuestas de la bibliografía.

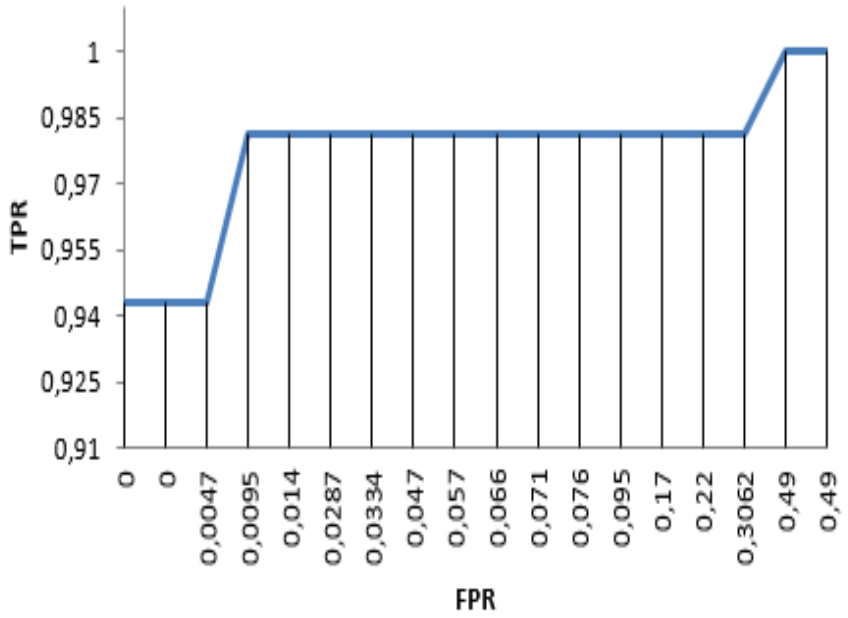

Figura 7.3: Curva ROC de la precisión al analizar las muestras CAIDA'07.

7.2. Ataques EDoS

El calibrado de la herramienta desplegada sobre entorno *Cloud* ha sido similar al de los experimentos anteriores, solamente se modificado el umbral de detección, Fig. 7.4.

Aplha	Beta	Gamma	Valor Umbra	Calentamiento			% Falsos Pos	ACIERTOS
0,7	0,7	0,7	2,1	10	50		0	0,8
0,7	0,7	0,7	2	10	50		0	0,84
0,7	0,7	0,7	1,9	10	50		0,04	0,92
0,7	0,7	0,7	1,8	10	50		0,12	0,92
0,7	0,7	0,7	1,7	10	50		0,2	0,92
0,7	0,7	0,7	1,6	10	50		0,28	0,96
0,7	0,7	0,7	1,5	10	50		0,32	0,96
0,7	0,7	0,7	1,4	10	50		0,36	1
0,7	0,7	0,7	1,3	10	50		0,4	1
0,7	0,7	0,7	1,2	10	50		0,4	1
0,7	0,7	0,7	1,1	10	50		0,48	1
0,7	0,7	0,7	1	10	50		0,52	1

Figura 7.4: Resultados de diferentes configuraciones en EDOS.

La mejor configuración probada ha arrojado una tasa de acierto del 92 % tras el análisis de 25 intentos de intrusión distintos. La tasa de falsos positivos es del 4 %. Fig. 7.5

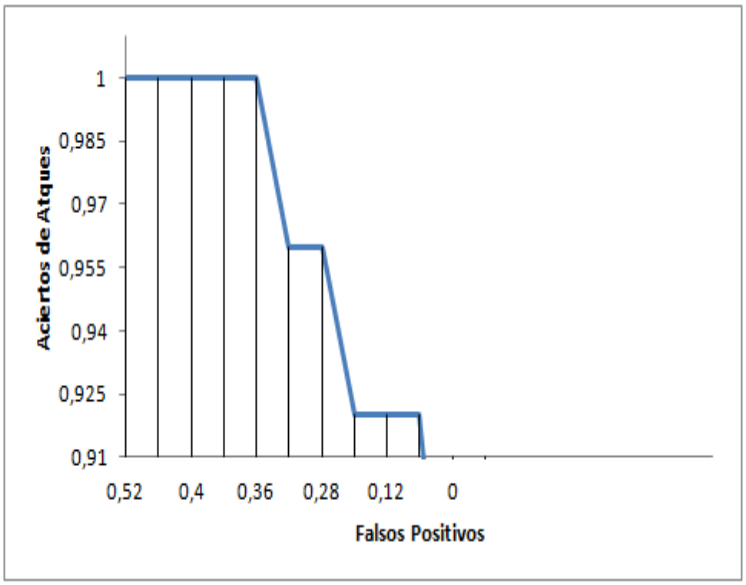

Figura 7.5: Curva ROC de la precisión al analizar las muestras EDOS.

Estos resultados son peores que los obtenidos al analizar las colecciones públicas. El principal motivo es que este modelo de tráfico es actual, y por lo tanto, presenta una mayor heterogeneidad; hoy en día existe una mayor cantidad de servicios disponibles que hace 7 años. A pesar de ello, es una excelente opción para defender una infraestructura *Cloud* de amenazas EDoS, e incluso otros tipos de denegación de servicio basados en inundación HTTP.

Capítulo 8

Conclusiones y Trabajo Futuro

8.1. Conclusiones

El objetivo de nuestra investigación ha sido el desarrollo de un sistema de detección de intrusiones para la detección de los ataques de denegación de servicio distribuidos que buscan agotar la sostenibilidad de los servicios ofrecidos en la nube (EDoS). Para ello, en primer lugar se ha estudiado e investigado en profundidad acerca de los diferentes tipos de técnicas de ataque DDoS, y sus contramedidas. Esto ha llevado a la elaboración de un completo estado del arte. De entre las propuestas estudiadas, se ha elegido el uso de la entropía y series temporales como métricas para la herramienta de detección propuesta. Estas son analizadas mediante el método predictivo Holt-Winters, el cual permite pronosticar el grado de incertidumbre de la red protegida en futuras observaciones. Se ha determinado que cuando algón valor observado, supera ciertos umbrales de predicción, se está produciendo un comportamiento anómalo. Esto quiere decir que el tráfico ha dejado de comportarse de manera predecible. Las anomalías son indicadores de amenazas basadas en denegación de servicio, y por lo tanto, conllevan la emisión de alertas. El sistema propuesto ha sido evaluado con diferentes colecciones de ataques DDoS de dominio póblico aprobados en múltiples ocasiones por la comunidad investigadora. La alta precisión alcanzada tras su análisis demuestra que se trata de una estrategia competitiva, y eficaz contra ataques convencionales. Sin embargo esto no termina de demostrar su eficacia al identificar ataques EDoS. Para ello se ha construido un complejo escenario de pruebas, en el que la herramienta es desplegada en las propias máquinas virtuales que pretenden ser defendidas. Al igual que en el caso anterior, la precisión ha sido muy alta, validando la eficacia de la propuesta en su caso de uso específico. A la vista del esfuerzo realizado y de los resultados obtenidos, es posible concluir en que los objetivos fijados se han cumplido satisfactoriamente. Además se han abierto diferentes

líneas de investigación alternativas, y se ha recopilado suficiente información como para incentivar el emprendimiento de futuras propuestas similares.

8.2. Trabajo Futuro

De las diferentes futuras líneas de investigación que han quedado abiertas, cabe destacar aquellas que basan en la mejora del propio sistema de detección, y las que tratan de extender la experimentación realizada a fin de conocer otras competencias.

Las primeras involucran la implementación de otro tipo de métricas, como pueden ser diferentes tipos de entropía, u otro tipo de parámetros extraídos del tráfico analizado. También es de especial interés valorar las posibles mejoras ofrecidas por otro tipo de algoritmos predictivos, como pueden ser los modelos de autorregresión.

En cuanto a la experimentación, sería conveniente evaluar la propuesta con otro tipo de ataques DDoS, como los ataques a largo plazo o *slow rate*. También sería importante tener en cuenta otro tipo de ataques EDoS como por ejemplo, los de inundación XML, y sus métodos de evasión.

Capítulo 9

Contribuciones

Contribución de Julio Javier López Giménez

- **Investigación**

  1. Realicé una investigación conjunta sobre los ataques de tipo de denegación de servicio, leyéndome artículos de diferentes ámbitos sobre ataque, predicción y defensa realizados en diferentes épocas, haciendo hincapié en las técnicas utilizadas en la actualidad.
  2. En este aspecto, me focalicé en el estado del arte de técnicas de ataque que puede haber en DDoS. Su posible motivación para realizar estos ataques, los tipos que hay, capa de red o capa de aplicación y los ataques específicos que existen en los servicios *cloud*.
  3. El siguiente paso que realizamos, fue la implementación del sistema de detección, donde investigué junto a mi compañero Lorenzo las diferentes entropías que se han utilizado para realizar una medida del desorden en el tráfico, además de la investigación de series temporales e intervalos de predicción en diferentes artículos y tesis.
  4. Además, de la investigación que realicé sobre la infraestructura y capas del *cloud*, para la futura implementación de nuestra herramienta.

- **Desarrollo**

  1. Parser: Investigué junto con mis compañeros las diferentes librerías que podíamos utilizar para parsear el tráfico del cloud, realizando de manera conjunta el método parser con la librería LibPcap de Java.
  2. Entropía: Analicé las diferentes entropías que investigamos mi compañero Lorenzo y yo, eligiendo la entropía de *Shannon* como la más adecuada.

Implementando un método del cálculo de la entropía cada X observaciones en el tráfico.

3. Holt-Winters: Busqué información sobre los tipos de predicción más adecuada para nuestra herramienta, depurando el código realizado por mis compañeros Lorenzo y Jose.
4. Intervalos de Predicción: investigué información de los intervalos de predicción realizados en diferentes proyectos y tesis, realicé junto con mi compañero Jose la implementación del cálculo de los intervalos de predicción.
5. Experimentación: Realicé junto con mis compañeros de la investigación de estándares utilizados en la comunidad científica, además de la captura de tráfico legítimo y de ataque en el *cloud*.
6. Ajuste: Realicé junto con mis compañeros la fase de ajuste de las experimentaciones recogidas en la captura del tráfico del *cloud* y del estándar CAIDA.
7. Análisis de resultados: Realizado de manera conjunta con mis compañeros el análisis de los resultados anteriores, con el fin de saber si tiene un rendimiento adecuado nuestro detector implementado.

- **Memoria**

1. Capítulo 1: Realicé el estudio de la infraestructura *cloud*, ayudando en la escritura de algunos apartados a mi compañero Lorenzo.
2. Capítulo 2: Este capítulo fue escrito por mí y mis compañeros de manera conjunto, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
3. Capítulo 3: Este capítulo fue escrito por mí y mis compañeros de manera conjunto, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
4. Capítulo 4: realice la investigación de las diferentes entropías, escribiendo en este capítulo todo lo que concierne a la investigación.
5. Capítulo 5: Este capítulo fue escrito por mí y mis compañeros de manera conjunta, escribiendo las partes que implementé de la herramienta y que ayudé en el desarrollo de la herramienta.
6. Capítulo 6, 7 y 8: La escritura de estos capítulos lo realice de manera conjunta con mis compañeros, ya que las fases de pruebas y análisis de

resultados lo realizamos de manera conjunta, además de las ideas de los futuros trabajos a realizar.

7. Finalmente, realicé el paso de la memoria de Word a Látex.

Contribución de José Ángel Madrona Martini

- **Investigación**

  1. Al principio del proyecto realicé una investigación sobre el conjunto de técnicas ya existentes tanto de ataque y defensa, relacionado con la Denegación de Servicio. Para ello leí un conjunto de papers analizándolos para sacar ciertas conclusiones con las que orientar nuestra investigación.
  2. Al orientarse nuestro proyecto en la defensa de estos ataques, leí un conjunto de papers sobre técnicas de defensa concretas. Estudiando así distintas variantes ya implementadas para nuestro proyecto.
  3. Estuve informándome sobre los ataques EDoS ya que en la investigación se centró en los entornos *cloud*.
  4. Para la implementación de la herramienta investigué sobre las técnicas combinadas de detección de la entropía con modelos de predicción. Y así poder avanzar en la herramienta correctamente.

- **Desarrollo**

  1. Parser: investigué junto con mi compañero Lorenzo y Julio las diferentes librerías que podíamos utilizar para parsear el tráfico en la herramienta, realizando de manera conjunta la parte del parseo del tráfico.
  2. Entropía: ayude en la investigación de la entropía.
  3. Holt Winters:investigué junto con mi compañero Lorenzo y Julio las diferentes series temporales, implementando el método del cálculo de la predicción *Holt-Winters*.
  4. Intervalos de Predicción: investigué información sobre la implementación de los intervalos de predicción realizados en diferentes *papers* y tesis, realizando junto con mi compañero Julio la implementación del cálculo de los intervalos de predicción.
  5. Experimentación: realicé junto con mis compañeros de la investigación de estándares utilizados en la comunidad científica, además de la captura de tráfico legítimo y de ataque en el *cloud*.

6. Ajuste: realicé conjuntamente con mis compañeros el ajuste y calibración de la herramienta. Utilizando pruebas de ataque legítimo y atacante sacado del dataset CAIDA '07.
7. Análisis de resultados: junto con mis compañeros realicé el estudio final de todos los datos extraídos de las pruebas anteriores, tanto para el entorno físico como el *cloud*.

- **Memoria**

1. Capítulo 1: Realicé un estudio del entorno e infraestructura *cloud*, para ayudar a la escritura del mismo.
2. Capítulo 2: Este capítulo fue escrito por mi y mis compañeros de manera conjunto, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
3. Capítulo 3: Este capítulo fue escrito por mi y mis compañeros de manera conjunto, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
4. Capítulo 4: Realicé como ya he dicho anteriormente un estudio de entropía y series temporales. Para escribir junto a mi compañero Julio Javier de este capítulo.
5. Capítulo 5: Este capítulo fue escrito por mi y mis compañeros de manera conjunta, escribiendo las partes que implementé y que ayudé en el desarrollo de la herramienta.
6. Capítulo 6, 7 y 8: La escritura de estos capítulos lo realice de manera conjunta con mis compañeros, ya que las fases de pruebas y análisis de resultados lo realizamos de manera conjunta, además de las ideas de los futuros trabajos a realizar.
7. Finalmente, realicé el paso de la memoria de Word a Latex. Utilizando un editor de Latex.

Contribución de Lorenzo Susarte Trujillano

- **Investigación**

1. Realicé una investigación conjunta con mis compañeros sobre los tipos de ataque y defensa que tienen que ver con los ataques de denegación de servicio (DoS). Leyendo un gran número de artículos y alguna tesis con la que afianzar conceptos y así poder determinar el camino de la investigación.

2. Posteriormente me centré en la lectura de técnicas de defensa que ya existen para la detección de este tipo de ataques.
3. Al tomar la investigación el rumbo hacia el entorno *cloud*, realice como en la fase anterior un estudio sobre las técnicas de ataques EDos, sus consecuencias y su detección.
4. Durante la realización del proyecto constantemente estuve leyendo artículos con los que sacar información sobre las técnicas que aplicábamos a nuestro sistema de detección.
5. Estudiando tanto tipos de entropía, series temporales, intervalos y modelos de predicción.

- **Desarrollo**

1. Parser: Junto con mis compañeros estuvimos informándonos sobre las diferentes librerías para la programación e implementación del parser. La implementación tanto en C como en Java la realizamos de manera conjunto, ya que era una de las partes vitales del proyecto, el parseo del tráfico.
2. Entropía: En este caso leí un conjunto de *papers* para informarme sobre los distintos tipos de entropía, y así abordar la mejor para nuestra investigación. La implementación del mismo lo realicé conjunto a mi compañero Julio Javier, realizándola en Java y C, utilizando la entropía de Shannon finalmente.
3. Holt Winters: Realicé una investigación sobre el conjunto de series temporales y modelos de predicción mediante la lectura de distintos papers sobre ellos. Para posteriormente realizar la implementación en código java del mismo con mi compañero José Ángel. Utilizando el modelo de predicción *Holt-Winters*.
4. Intervalos de Predicción: En este apartado investigué sobre los intervalos de predicción para su uso en la herramienta.
5. Experimentación: Realicé junto a mis compañeros una fase de experimentación en la que poder depurar la herramienta y mejorarla. Utilizando el conjunto de pruebas CAIDA '07. Realizando pruebas tanto de tráfico atacante como tráfico legítimo.
6. Ajuste: Llevé a cabo junto con mis compañeros la fase de ajuste y anotación de las distintas pruebas con un rigor estadístico para la memoria.

Utilizando Excel para la anotación de los mismos. Y así poder llevar a cabo la siguiente fase de análisis.

7. Análisis de resultados: Analicé junto con mis compañeros los datos extraídos de las fases anteriores con el fin de conocer los resultados positivos de nuestra herramienta.

- **Memoria**

1. Capítulo 1: Realicé el estudio y escritura de la introducción.
2. Capítulo 2: Este capítulo fue escrito por mi y mis compañeros de manera conjunta, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
3. Capítulo 3: Este capítulo fue escrito por mi y mis compañeros de manera conjunto, ya que estuvimos realizando durante varios meses iniciales un estudio del estado del arte descrito en el mismo.
4. Capítulo 4: Ayudé a mis compañeros en la investigación de la misma, con la lectura de varios artículos.
5. Capítulo 5: Realicé en conjunto con mis compañeras la escritura del capítulo completo, especializándome en las partes que yo implementé de la herramienta.
6. Capítulo 6, 7 y 8: La escritura de estos capítulos los realicé de manera conjunta con mis compañeros. Ya que las ideas para la conclusión y futuros trabajos fueron expuestas por los tres. Y la fase de pruebas la realizamos también de manera conjunta.
7. Finalmente, realicé el paso de la memoria de formato Word a Latex. Dividiéndome con mis compañeros los capítulos junto a la inserción en los mismos de imágenes y fórmulas.

Bibliografía

[1] L. Marinos, ENISA (2015), Threat Landscape 2014. Available: https://www.enisa.europa.eu/.

[2] Symantec (2015), Internet Security Threat Report 2014, Vol. 19. Available at: https://www.symantec.com.

[3] T. Peng, C. Leckie, K. Ramamohanarao. "Survey of network-based defense mechanisms countering the DoS and DDoS problems", ACM Computing Surveys, Vol. 39 (1), No. 3, pp. 1-42, 2007.

[4] R. Sandoval-Almazan, J. R. Gil-Garcia. "Towards cyberactivism 2.0? Understanding the use of social media and other information technologies for political activism and social movements", Government Information Quarterly, Vol. 31 (3), pp. 365-378, July 2014.

[5] W. Wei, F. Chen, Y. Xia, G. Jin. "A rank correlation based detection against distributed reflection DoS attacks", IEEE Communications Letters, Vol. 17 (1), pp. 173-175, January 2013.

[6] Y. Tang, X. Luo, Q. Hui, R.K.C. Chang. "Modeling the Vulnerability of Feedback-Control Based Internet Services to Low-Rate DoS Attacks", IEEE Transactions on Information Forensics and Security, Vol. 9 (3), pp. 339-353, 2014.

[7] C. Douligeris, A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art", Computer Networks, Vol. 44 (5), pp. 643-666, April 2004.

[8] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, S. Gritzalis. "DNS amplification attack revisited", Computers & Security, Vol. 39, part B, pp. 475-485, November 2013.

[9] S. T. Zargar, J. Joshi, D. Tipper. "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks", IEEE Communications Surveys & Tutorials, Vol. 15 (4), pp. 2046-2069, March 2013.

[10] J. Pan,H. Hu, Y. Liu. "Human behavior during Flash Crowd in web surfing", Physica A: Statistical Mechanics and its Applications, Vol. 413 (1), pp. 212-219, November 2014.

[11] W. Zhou, W. Jia, S. Wen, Y. Xiang, W. Zhou. "Detection and defense of application-layer DDoS attacks in backbone web traffic", Future Generation Computer Systems, Vol. 38, pp. 36-46, September 2014.

[12] D. Seo, H. Lee, A. Perrig, "APFS: Adaptive Probabilistic Filter Scheduling against distributed denial-of-service attacks", Computers & Security, Vol. 39 (B), pp. 366-385, November 2013.

[13] X. Wang, M. Yang, J. Luo. "A novel sequential watermark detection model for efficient traceback of secret network attack flows", Journal of Network and Computer Applications, Vol. 36 (6), pp. 1660-1670, November 2013.

[14] L. Ablon, M. C. Libicki, A. A. Golay (RAND Corporation) (2014), Markets for Cybercrime Tools and Stolen Data: Hackers' Bazaar. Resource available at http://www.rand.org.

[15] A. Karami, M. Guerrero-Zapata. "A hybrid multiobjective RBF-PSO method for mitigating DoS attacks in Named Data Networking", Neurocomputing, Vol. 151 (3), pp. 1262-1282, March 2015.

[16] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, S. Shenker. "Controlling high bandwidth aggregates in the network", ACM SIGCOMM Computer Communication Review, Vol. 30 (3), pp.62-73, July 2002.

[17] R. Chen, J.M. Park, "Attack diagnosis: throttling distributed denial-of-service attacks close to the attack sources", in Proceedings of the 14th IEEE International Conference on Computer Communications and Networks (ICCN), San Diego, CA, USA, pp. 275-280, October 2005.

[18] J. Mirkovic, M. Robinson, P. Reiher, "Alliance formation for DDoS defense", in Proceedings of the 2003 Workshop on New Security Paradigms (NSPW), The Banff Centre, Canada, pp. 11-18, September 2013.

[19] K. Argyraki, and D. R. Cheriton, Scalable network-layer defense against internet bandwidth-flooding attacks, IEEE/ACM Transactions on Networks, Vol. 17(4), pp. 1284-1297, August 2009.

[20] C. Callegari, S. Giordano, M. Pagano, T. Pepe. "Wave-CUSUM: improving CUSUM performance in net-work anomaly detection by means of wavelet analysis", Computers & Security, Vol. 31 (5), pp. 727-735, July 2012.

[21] S. Shin, S. Lee, H. Kim, S. Kim. "Advanced probabilistic approach for network intrusion forecasting and detection", Expert Systems with Applications, Vol. 40 (1), pp. 315-322, January 2013.

[22] Y. Chen, X. Ma, X. Wu. "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory", IEEE Communications Letters, Vol. 17 (5), pp. 1052-1054, May 2013.

[23] S.M. Lee, D.S. Kim, J.H. Lee, J.S. Park. "Detection of DDoS attacks using optimized traffic matrix", Computers & Mathematics with Applications, Vol. 63 (2), pp. 501-510, September 2012.

[24] Y. Cai, R.M. Franco, M. García-Herranz. "Visual latency-based interactive visualization for digital forensics", Journal of Computational Science, Vol. 1 (2), pp. 115-120, June 2010.

[25] P.A.R. Kumar, S. Selvakumar. "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems", Computer Communications, Vol. 36 (3), pp. 303-319, February 2013.

[26] M.H. Bhuyan, D. K. Bhattacharyya, J.K. Kalita. "An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection", Pattern Recognition Letters, Vol. 51 (1), pp. 1-7, August 2014.

[27] I. Ozcelik, R.R. Brooks. "Deceiving entropy based DoS detection", Computers & Security, Vol. 48 (1), pp. 234-245, February 2015.

[28] B.B. Zhu, J. Yan, G. Bao, M. Yang, N. Xu. "Captcha as Graphical Passwords: A New Security Primitive Based on Hard AI Problems", IEEE Transactions on Information Forensics and Security, Vol. 9 (6), pp. 891-904, April 2014.

[29] K. E. Heckman, M. J. Walsh, F. J. Stech, T. A. O'Boyle, S. R. DiCato, A. F. Herber. "Active cyber defense with denial and deception: A cyber-wargame experiment", Computers & Security, Vol. 37, pp. 72-77, September 2013.

[30] S. Khanna, S.S. Venkatesh, O. Fatemieh, F. Khan, C.A. Gunter. "Adaptive Selective Verification: An Efficient Adaptive Countermeasure to Thwart DoS Attacks", IEEE/ACM Transactions on Netwowking, Vol. 20 (3), pp. 715-728, June 2012.

[31] N.M. Alenezi, M.J. Reed. "Uniform DoS Traceback", Computers & Security, Vol. 45 (1), pp. 17-26, September 2014.

[32] A.R. Kiremire, M.R. Brust, V.V. Phoha. "Using network motifs to investigate the influence of network topology on PPM-based IP traceback schemes", Computer Networks, Vol. 72 (1), pp. 14-32, October 2014.

[33] A. Chonka, J. Abawajy. "Detecting and mitigating HX-DoS attacks against Cloud web services", In Proceedings of the 15th IEEE International Conference on Network-Based Information Systems (NBIS'12), Melbourne, VIC, Australia, pp. 429-434, September 2012.

[34] Y. Xiang, A. Chonka, W. Zhou, J. Singh. "Detecting and tracing DDoS attacks by intelligent decision prototype (IDP)", In Proceedings of the 6th Annual IEEE International Conference on Web and Pervasive Security (PerCom'08), Hong Kong, Chine, pp. 578-583, March 2008.

[35] C. Yu, H. Kai. "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis", Journal of Parallel and Distributed Systems, Vol. 66 (9), pp. 1137-1151, September 2006.

[36] K. Beaty, A. Kundu, V. Naik, A. Acharya. "Network-level access control management for the Cloud", In Proceedings of the IEEE International Conference on Cloud Engineering (IC2DE), Redwood City, CA, US, pp. 98-107, March 2013.

[37] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. Ping Liu. "A system for denial-of-service attack detection based on multivariate correlation analysis", IEEE Transactions on Parallel and Distributed Systems, Vol 25 (2), pp. 447-456, December 2013.

[38] M. Ficco, M. Rak. "Stealthy Denial of Service Strategy in Cloud Computing", IEEE Transactions on Cloud Computing, Vol. 3 (1), pp. 80-94, March 2015.

[39] P. Sujatha M. Kumar, R. Korra. "Mitigation of economic distributed denial of sustainability (EDDoS) in cloud computing". In Proceedings of the International Conference on Advances in Engineering and Technology, (ICAET-2011), Nagapattinam, India, May 2011.

[40] F. Al-Haidari, M. Sqalli, K. Salah. "Evaluation of the Impact of EDoS Attacks Against Cloud Computing Services", Arabian Journal for Science and Engineering, Vol. 40 (3), pp. 773-785, March 2015.

[41] J. Idziorek, M. Tannian. "Exploiting Cloud Utility Models for Profit and Ruin", in Proceedings of the IEEE International Conference on Cloud Computing (CLOUD'11), Washington, DC, US, pp. 33-40, July 2011.

[42] M. H. Sqalli, F. Al-Haidari, K. Salah. "EdoS Shield - A Two-Steps Mitigation Technique against EDoS At-tacks in Cloud Computing". In Proc of the 4th IEEE International Conference on Utility and Cloud Computing (UCC'11), Melbourne, Australia, December 2011.

[43] F. Al-Haidari, M.H. Sqalli, K. Salah. "Enhanced EDoS-Shield for Mitigating EDoS Attacks Originating from Spoofed IP Addresses", in Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Livepool, UK, pp. 1167-1174, June 2012.

[44] M.N. Kumar, P. Sujatha, V. Kalva, R. Nagori. "Mitigating Economic Denial of Sustainability (EDoS) in Cloud Computing Using In-cloud Scrubber Service", in Proceedings of the IEEE Fourth International Conference on Computational Intelligence and Communication Networks (CICN), Mathura, India, pp. 535-539, November 2012.

[45] Z.A. Baig, F. Binbeshr, "Controlled Virtual Resource Access to Mitigate Economic Denial of Sustainability (EDoS) Attacks against Cloud Infrastructures", in Proceedings of the IEEE International Conference on Cloud Computing and Big Data (CloudCom-Asia'13), Fuzhou, Chine, pp. 346-353, December 2013.

[46] M. Masood, Z. Anwar, S.A. Raza, M.A. Hur. "EDoS Armor: A cost effective economic denial of sustainability attack mitigation framework for e-commerce applications in cloud environments", In Proceedings of the 16th International Multi Topic Conference (INMIC'13), Lahore, Pakistan, pp. 37-42, December 2013.

[47] B. Saini, G. Somani. "Index Page Based EDoS Attacks in Infrastructure Cloud", in Proceedings of Second International Conference on Recent Trends in Computer Networks and Distributed Systems Security (SNDS), Trivandrum, India. Communications in Computer and Information Science Vol. 420, pp. 382-395, March 2014.

[48] S. Bhatia, D. Schmidt, G. Mohay, A. Tickle. "A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events", Computers & Security, Vol. 40 (1), pp. 95-107, February 2014.

[49] The CAIDA UCSD (2015), "DDoS Attack 2007 Dataset". Available: http://www.caida.org.

[50] Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, 1963.

[51] K. Zyczkowski "Renyi extrapolation of Shannon entropy", Open Syst. Inf. Dynamics, Vol. 10 (3), pp.297-310, 2003.

[52] Liying Li, Jianying Zhou, Ning Xiao, "DDoS Attack Detection Algorithms Based on Entropy Computing", Proceedings of the 9th International Conference ICICS 2007, Zhengzhou, China December 12-15, 2007.

[53] B. Song, J Heo, and C. S. Hong, "Collaborative Defense Mechanism Using Statistical Detection Method Against DDoS Attacks", IEICE Transactions on Communications, E90-B, 2007, pp. 2655-2644.

[54] Giseop No, Ilkyeun Ra, "Communications and Information Technology", Proceedings of the 9th International Symposium on ISCIT 2009. 28-30 Sept 2009.

[55] A. Chonka, "Cloud Security Defence to Protect Cloud Computing Against HTTP-DoS and XML-DoS Attacks", Journal of Network and Computer Applications, 2010.

[56] G. Box, G. Jenkins, and G. Reinsel, "Time Series Analysis", Holden-day San Francisco, 1970.

[57] P. Brockwell and R. Davis, "Introduction to time series and forecasting". Springer Verlag, 2002.

[58] A. Yaacob, I. Tan, S. Chien, and H. Tan, "Arima based network anomaly detection, " in Proceedings of the Second International Conference on Communication Software and Networks. IEEE, 2010, pp. 205-209.

[59] Sang and S. Li, "A predictability analysis of network traffic", Computer Networks, 2012.

[60] Maciej Szmit and Anna Szmit, "Use of Holt-Winters Method in the Analysis of Network Traffic: Case Study", Communications in Computer and Information Science, p. 224. 2011, Poland.

[61] C.E. Shannon. “A Mathematical Theory of Communication”, The Bell System Technical Journal, Vol. 27 (3), pp. 397-423, July 1948.

[62] A. Renyi.“On Measures of Entropy and Information”, in Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, US, Vol. 1, pp. 547-561, June 1961.

[63] R.V.L. Hartley. “Transmission of information”, The Bell System Technical Journal, Vol. 7 (3), pp. 535-563, July 1928.

[64] R. Konig, R. Renner, C. Schaffner. “The operational meaning of min-and max-entropy”, IEEE Transactions on Information Theory, Vol. 55 (9), 4337-4347, August 2009.

[65] P.R. Winters. “Forecasting Sales by Exponentially Weighted Moving Averages”, Management Science, Vol. 6 (3), pp. 324-342, April 1960.

[66] Libpcap (2015). Available: http://sourceforge.net/projects/libpcap/.

[67] JnetPcap (2015). Available: http://jnetpcap.com/.

[68] NetBeans (2015). Available: https://netbeans.org/.

[69] JFreeChart (2015). Available: http://www.jfree.org/jfreechart/.

[70] OpenNebula (2015). Available: http://opennebula.org/.

[71] KDD Cup 1999 (2015). Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[72] Low Orbit Ion Cannon (LOIC) (2015). Available: http://sourceforge.net/projects/loic/.

Parte I

Anexo

Apéndice A

Introduction

At present, cloud computing and virtual servers have gained considerable ground to conventional physical servers. This is due to the large number of advantages offered, among which are its low cost, and greater agility, flexibility and scalability. The first is mainly due to savings in maintenance costs, licenses, and the adoption of pay-per-use paradigm. The following, for ease of supplying new computing resources where required. However, their deployment also has a number of drawbacks, mainly focused on problems of availability, connectivity and security.

In this paper the security problem is approached. Their choice was motivated by the notices issued by the different organizations involved in cyber defense, who have detected a rapid growth of such threats. Attacks on cloud computing has evolved from the classic strategies of intrusion, to exploitation of specific vulnerabilities of your environment, making schemes be less effective than conventional defense.

Of these new threats stands out because increased attacks on the sustainability of the services offered. It is a type of intrusion that seeks to engage the victim's economy, directly affecting their pay per use model. The most common way to achieve this is to force the service deployed on the cloud to perform a lot of extra work. This involves hiring additional resources, more expensive bill of the victim, and making their use unfeasible.

Generally, threats to the sustainability achieve success when they go hand in hand with tools used in other contexts to execute denial of service attacks. This combination is called a denial of the sustainability of a cloud service, also known as EDoS. The work focuses on the study, and the proposal of a defensive scheme for identification. For better understanding, are described below a series of previous concepts, the objectives and the structure of the rest of the document.

A.1. Previous Concepts

A.1.1. Physical and virtual servers

A physical server is defined as the use or configuration of hardware and software resources, located in a particular place. For its part, the virtual servers are software installations performed on physical servers, with hosting capacity of different virtual machines. They share common resources among themselves, behaving in a complementary way despite covering different tasks.

At present the mentality is changing at the time of host applications and web services on the network. Originally predominated the physical servers use. This offered fixed and inflexible characteristics, which were difficult to modify by customers. To deal with this problem, the contracting part increasingly more frequent recourse to technologies related to cloud computing, triggering an increase in demand for virtual servers.

Among the improvements that this implies, include its ease of starting, scalability, lower costs of maintenance and licenses its ease of administration. However problems need to be addressed by hiring third parties, connectivity, availability, privacy and security.

A.1.2. Provision models

Delivery models are set computing services that manage the resources hosted on the Internet. Usually applied any of the following schemes: mainframe, client / server or cloud.

Below each of them is described:

- **Mainframe**: The supply is centralized characteristics, which facilitates a high degree of computing and storage space. Despite its efficiency, it has high management and maintenance costs.
- **Cliente/Servidor**: Provisioning Client / Server consists of a set of machines and servers for distributed computing and storage to promoting agility of services and offering low costs. Its biggest flaw is compulsory licensing of software, which carry a significant economic cost.
- **Cloud:** Cloud provisioning is from big surfaces, with centers full of machines and highly scalable data servers. This allows optimizing the efficiency and agility of services offered, using the economic model of pay per use. Unlike the client / server model, saves licensing costs.

A.1.3. Cloud computing

The cloud computing model is the provision of resources for any platform or web application as a service, on demand and dynamically (elasticity) through internet. It is conceived as a new computing paradigm that can offer different services through the network. It is characterized by rapid mobilization of resources, which enables quick adaptation to demand variability. Its main features, similar to those described in virtual servers, are variable costs, scalability, elasticity, collaborative work, energy savings, consistency and ubiquity. Within the cloud, and the type of service provided, can distinguish the following types of layers Fig. 1.1:

- **SaaS (Software as a service)**: SaaS provides a complete application as a service on demand and evaluate the collaborative use by several users. The application usage limited to access through a web browser, without installations.

- **PaaS (Plattform as a service)**: PaaS is the middle layer of cloud computing. It allows complete abstraction of a development environment and provides a set of complements and alternative packages with which to complete the application deployed on it. Among the accessories you can be found different libraries that provide specific functionality, also known as *APIs*.

- **IaaS (Infraestructure as a service)**: The IaaS is the lowest layer of cloud computing. Defines the infrastructure that supports the higher layers. Among the various components that can be managed under it, are own servers, storage, computing capabilities that serve as support for the upper layers, and applications hosted on it. In this layer the particular capabilities of the cloud as scalability and distribution of workload among servers are provided.

Can be distinguished three types of clouds in the cloud, classified according to their manager. The following describes briefly:

- **Public Cloud**: The maintenance of public cloud is performed by third persons, rather than by the contracting organization. Within this set of server and storage systems circulate data from several organizations, of which the end user will only see his own, and can perform various configurations thereof.

- **Private Cloud**: Maintenance of private cloud is performed by the contracting organization. This is an option suited to companies that need a high level of data protection.

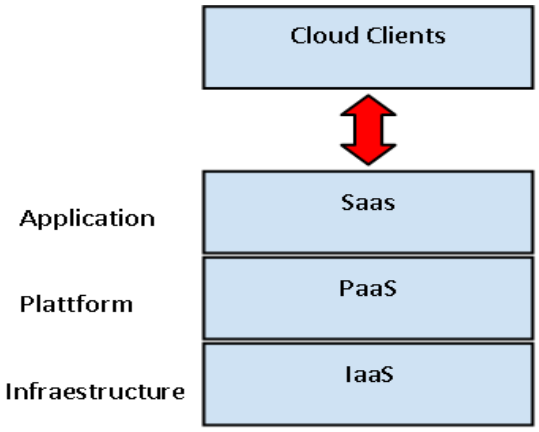

Figura A.1: Cloud Layers

- **Hybrid Cloud**: The hybrid cloud combines the two previous types. In this case, part of the information is managed by the contracting organization, and the rest is public domain.
- **Community Cloud**: The community cloud is designed to be used with a common purpose among groups of participants. It can be maintained by the provider, or by contracting organizations.

A.1.4. Virtualization

Virtualization is the technology that allows the existence of cloud computing, providing flexibility and scalability to satisfy the demands of users. It is usually is defined as the ability to boot and run virtual machines in a cloud hypervisor. In this process, each virtual machine simulates the complete operation of a physical machine, leading the entire implementation, including aspects such as its kernel, operatic system or applications.

A.2. Objectives work

The main objective of this work is the realization of an intrusion detection system or IDS, for the identification of EDoS flood attacks directed against cloud infrastructure. This means satisfying the following secondary objectives:

- The investigation of the most important techniques of defense and attack in such architectures, with particular emphasis on those that relate to the EDoS attacks.

- Developing strategies for extraction, and the interpretation of the characteristics of the traffic that is directed towards them.
- The development of metrics that allow modeling of such information.
- Building predictive models to identify statistical anomalies, able to unmask intrusion attempts.
- The development of an evaluation methodology that supplies to the extent possible, the shortcomings of conventional schemes.
- • Verification of the efficiency of the proposed.

Apéndice B

Conclusions and Future Work

B.1. Conclusions

The aim of our research has been the development of an intrusion detection system for detecting distributed denial of service seeking to exhaust the sustainability of the services offered in the cloud (EDoS). To do this, first we have studied and investigated in depth about the different types of techniques DDoS attack and countermeasures. This has led to the elaboration of a complete state of the art.

Among the proposals studied, we have chosen the use of entropy and temporal series as metrics for the proposed detection tool. These are analyzed using predictive Holt-Winters, which allows us to predict the uncertainty of future observations in the protected network. We have determined that when an observed value exceeds certain thresholds prediction is occurring anomalous behavior. This means that traffic has ceased to behave predictably. The anomalies are indicators based threats denial of service, and therefore involve issuing alerts.

The proposed system has been evaluated with different collections of public domain DDoS attacks on multiple occasions approved by the research community. The high achieved accuracy after analysis shows that it is competitive, and effective strategy against conventional attacks. However it does not stop to demonstrate their effectiveness in identifying EDoS attacks. For this we have built a complex test scenario in which the tool is deployed in its own virtual machines pretending to be defended. As in the previous case, the accuracy was very high, validating the effectiveness of the proposal as a specific use case.

In view of the efforts made and the results obtained, we may conclude that the objectives set have been satisfactorily fulfilled. We have also opened different lines of research alternatives and has gathered sufficient information to encourage entrepreneurship of future similar proposals.

B.2. Future Work

Of the different future research that have been open, include those based on improving the detection system itself, and trying to extend the experimentation to learn other skills.

The first involves the implementation of other metrics, such as different types of entropy, or other parameters extracted from the traffic analyzed. Also of special interest to evaluate possible improvements offered by other predictive algorithms, such as autoregression models.

In terms of experimentation, it would be evaluating the proposal with other types of DDoS attacks, such as long-term attacks or slow rate. It would also be important to consider other attacks EDoS such as the XML flood, and their methods of evasion.