Universidad de Valladolid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Trabajo de Fin de Grado

Grado en Ingeniería de Tecnologías Específicas de Telecomunicación

Detección de ataques distribuidos de baja intensidad mediante análisis estadístico del tráfico entrante

Autor:
Iván Herrero Alonso

Tutores:
D. Federico Jesús Simmross Wattenberg
D. Antonio Tristán Vega

Valladolid, septiembre 2024

I have nothing to offer but
blood, toil, tears and sweat.

Winston S. Churchill

Resumen

En las últimas tres décadas, los ataques de denegación de servicio distribuido (DDoS) han sido una de las mayores amenazas para la infraestructura de Internet, provocando grandes cantidades de euros en daños y afectando a la reputación de numerosas organizaciones. La evolución de la computación en la nube ha dado lugar a nuevos tipos de ataques, como los de denegación de servicio de baja tasa (LDoS) y distribuidos de baja tasa (LDDoS). Estos ataques emplean ráfagas de tráfico de baja intensidad, permitiéndoles camuflarse entre el tráfico normal y eludir a los sistemas de detección. Este trabajo explora técnicas para detectar ataques LDDoS mediante el análisis estadístico del tráfico de red entrante. El estudio se centra en identificar y analizar las características de los patrones de tráfico LDDoS, empleando para ello métodos en los dominios del tiempo y la frecuencia, y evaluando el rendimiento que presentan distintos algoritmos. Los resultados de las simulaciones muestran tanto la eficacia como las deficiencias de los métodos de detección basados en la densidad espectral de potencia, ofreciendo nuevas perspectivas para mejorar la seguridad de las redes frente a estas amenazas sigilosas.

Keywords— LDoS, LDDoS, Entropía, Densidad Espectral de Potencia (PSD), Distribución de Poisson

Abstract

Over the past three decades, distributed denial of service (DDoS) attacks have been one of the biggest threats to Internet infrastructure, causing large amounts of euros in damage and affecting the reputation of many organisations. The evolution of cloud computing has given rise to new types of attacks, such as low-rate denial of service (LDoS) and low-rate distributed denial of service (LDDoS) attacks. These attacks employ bursts of low-intensity traffic, allowing them to camouflage among normal traffic and evade detection systems. This essay explores techniques to detect LDDoS attacks through statistical analysis of incoming network traffic. The study focuses on identifying and analysing the characteristics of LDDoS traffic patterns using time and frequency domain methods, and evaluating the performance of different algorithms. The results of the simulations show both the effectiveness and shortcomings of detection methods based on power spectral density, offering new perspectives for improving network security against these stealthy threats.

Keywords— LDoS, LDDoS, Entropy, Power Spectral Density (PSD), Poisson Distribution

Agradecimientos

Este trabajo es el punto final a cuatro años de duro trabajo, en todo este tiempo he conocido a algunas personas, tanto compañeros de clase como profesores, las cuales han dejado profundas huellas en mí y de las que siempre llevaré buenos recuerdos.

En primer lugar me gustaría agradecerles a mis tutores, Fede y Antonio, la oportunidad de realizar este trabajo con el que tanto he aprendido y por estar ahí para guiarme o corregirme en los momentos en los que me encontraba perdido sin saber que debía hacer a continuación

También a mis compañeros del Radioclub (David, Juantxi, Marco, Ana y tantos otros) los cuales siempre han estado dispuestos a animarme cuando me encontraba frustrado, a prestarme una oreja cuando comenzaba a divagar sobre diferentes posibilidades de estudio o a expresar sus propios puntos de vista sobre el tema, los cuales en más de una ocasión me han llevado a reflexionar más a fondo sobre mis propias ideas.

En esta sección debo hacer una mención a quien ha sido mi compañero de batallas durante todo este recorrido, Eduardo, con quien desde un primer momento formé equipo y sin el que jamás habría podido llegar tan lejos.

Finalmente, me gustaría agradecer especialmente a mi familia por todo lo que han hecho por mí a lo largo de toda mi vida, siempre apoyándome en mis decisiones (por más descabelladas que fueran) y siempre mostrándome su cariño en todo momento, a pesar de que en algunos momentos mi carácter me impida ver que tienen mi bienestar en mente. Por todo ello solo puedo decir gracias por todo y os quiero.

Índice

Índice de Figuras

Índice de Tablas

Índice de Códigos

1 Introducción

El concepto de ciberseguridad hace referencia a la habilidad de proteger equipos, redes y programas frente a ataques que tienen como objetivo acceder, cambiar o destruir datos sensibles, extorsionar a usuarios o interrumpir los servicios normales.

La implementación de medidas efectivas en el ámbito de la ciberseguridad es una tarea que presenta un gran desafío en la actualidad, debido a la cantidad de dispositivos que existen y a la creciente capacidad de innovación de los atacantes, los cuales se están volviendo cada vez más profesionales, llegando en algunos casos a actuar como entidades empresariales, con una jerarquía y objetivos claramente definidos y aceptando inversiones a cambio de una parte de las ganancias de su siguiente ataque.

La Agencia de la Unión Europea para la Ciberseguridad (ENISA) es un organismo de inteligencia que se encarga de garantizar la ciberseguridad de las comunicaciones y servicios de la administración comunitaria de la Unión Europea, de los distintos países que la forman y de sus ciudadanos. Todos los años esta entidad publica un informe en el que recoge el panorama de amenazas de todo el año.

En su edición del 2023 [Cyb23] enumeran ocho grupos principales de amenazas que destacan por su prominencia a lo largo de los años y por su aparición e impacto significativos. Uno de estos grupos de amenazas es el de los ataques de denegación de servicio distribuido, DDoS. En este análisis se reveló que fueron la segunda categoría de ataques más usados, quedando situados únicamente por detrás de los ataques de *ramsomware*, tal y como se puede ver en la figura 1.

Bajo el nombre de DDoS se incluyen una gran variedad de técnicas de ataque que comparten un mismo objetivo principal: disminuir la disponibilidad de los sistemas y los datos o disminuir la calidad de servicio de estos, y comparten algunas características pese a tener distintas implementaciones. Uno de los ataques que se encuentran en esta categoría son los ataques en los que nos vamos a centrar en este trabajo, los ataques DDoS de baja intensidad, también llamados *Low-Rate Distributed Denial of Service* (LDDoS), los cuales se van a analizar más en detalle en el capitulo 2.3.

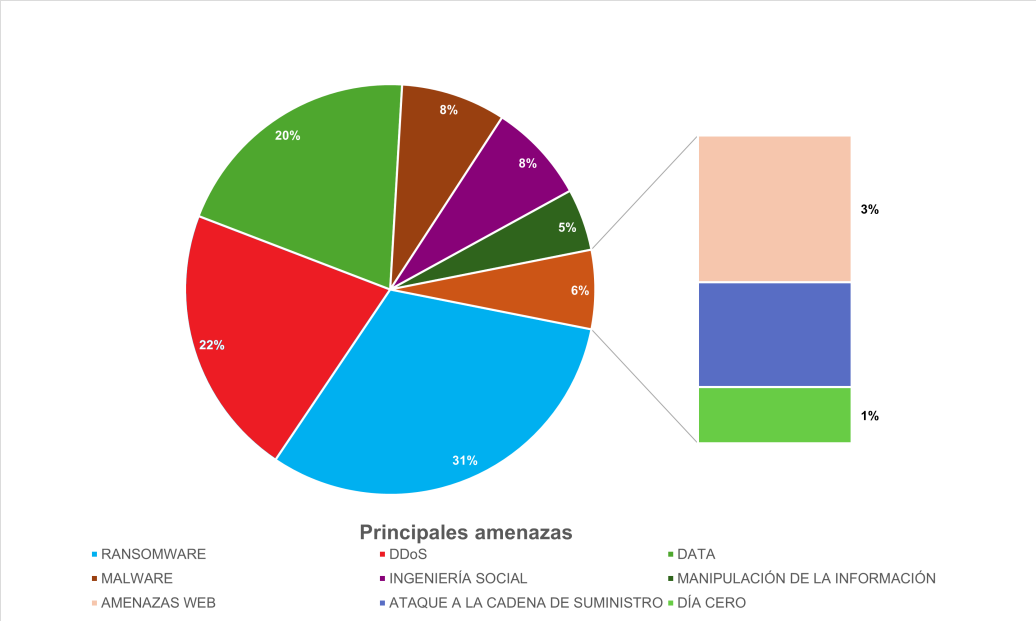

Figura 1: Tipos de amenaza principales en 2023 [Cyb23].

1.1 Motivación

En la actualidad la ciberseguridad se ha convertido en una prioridad para las organizaciones y gobiernos de todo el mundo debido al constante crecimiento de las amenazas cibernéticas. Los ataques contra la infraestructura de red no solo afectan a la disponibilidad de los servicios, sino que también ponen en riesgo la privacidad, la integridad de los datos y la estabilidad económica de países y empresas. Dentro de este contexto, los ataques LDDoS representan un desafío particularmente difícil de mitigar. El principal problema a la hora de tratar con estos ataques es su capacidad para camuflarse como tráfico legítimo, lo que hace que logren evadir a muchos de los mecanismos de seguridad tradicionales.

La aparición y evolución de estos nuevos ataques ha llevado al desarrollo de un gran número de herramientas de detección que implementan técnicas muy diversas. Pero pese a la existencia de estos nuevos sistemas, aún no estamos completamente seguros. Esto se debe a que los diferentes mecanismos presentan deficiencias que pueden ser explotadas para lograr llevar a cabo los ataques. Además, el hecho de que muchos de los datos que se emplean para el desarrollo de sistemas de detección proceden de simulaciones y de entornos que han sido configurados con unas condiciones precisas en

mente, hacen que no se tenga en cuenta la gran variedad de situaciones que se pueden dar en las redes reales, haciendo que en última instancia algunas de estas herramientas sean ineficaces.

1.2 Objetivos

En este trabajo se pretende estudiar algunas de las herramientas que son de utilidad para aquellos que se dedican al ámbito de la ciberseguridad al permitir detectar de forma eficaz los ataques LDDoS, los cuales presentan un gran desafío para muchos de los sistemas que se usan en la actualidad debido a la complejidad que les dota el tener la apariencia de tráfico legitimo.

En los siguientes puntos se va a analizar diversos sistemas que se han desarrollado en los últimos años y qué se basan en distintas metodologías de análisis para lograr comprender que fallos pueden tener algunas de las últimas propuestas. Esto servirá para sentar las bases sobre las que se podría construir un nuevo sistema.

Pese a que DDoS celebró en 2021 su vigésimo quinto aniversario sigue siendo una de las mayores amenazas para el buen funcionamiento de las redes, y por ello en este trabajo se pretende ayudar al desarrollo de una herramienta que permita a los equipos de defensa prepararse contra una de sus variedades, mejorando de esta forma las técnicas defensivas.

El objetivo de este trabajo es estudiar los diversos métodos existentes en la literatura que permiten detectar la presencia de ataques LDDoS en el tráfico que entra a una red o en el tráfico que llega a un servidor, contrastando su eficacia frente a distintos patrones de tráfico malicioso y proponer posibles formas de mejorar dichos métodos de detección. Para lograr el objetivo final de este trabajo se va a necesitar desarrollar las siguientes habilidades:

- Aprender a realizar labores de forense de red y a realizar búsquedas de documentación científica de calidad y confiable en el campo de la ciberseguridad.
- Aprender sobre los distintos métodos de análisis y los beneficios que presentan en diferentes situaciones.
- Ampliar los conocimientos y habilidades en el tratamiento de capturas de tráfico usando Wireshark.

- Aumentar los conocimientos en el área de la estadística, aplicando lo aprendido a la elaboración del código de Matlab.
- Aprender más sobre el proceso de ingeniería y el resto de habilidades adquiridas durante la realización del grado en Ingeniería de Telecomunicaciones, específicamente en la rama de Telemática.

1.3 Herramientas y Métodos

En este punto se va a detallar las diversas herramientas que se han usado para las distintas tareas y las opciones que se descartaron por no cumplir con alguno de los criterios buscados. Además de esto, también se recogen las características de los diferentes equipos usados en el desarrollo del trabajo.

1.3.1 *Software*

Dado que el sistema de detección que se pretende desarrollar requiere realizar un gran número de operaciones matemáticas, se han contemplado varias opciones que permiten la realización de cálculos con facilidad y permiten la obtención de gráficos a partir de los datos que se están empleando.

Entre las alternativas estudiadas para la implementación se encuentran Matlab, Octave y Python, puesto que estos lenguajes son ampliamente usados tanto en la industria como en el ámbito académico. A continuación se va a presentar cada uno de ellos y posteriormente se realizará un análisis comparativo por medio de la tabla 1.

Matlab (Matrix Laboratory) es un entorno de programación y un lenguaje de alto nivel escrito en C, C++ y Java, diseñado especialmente para el cálculo numérico, el análisis de datos, el desarrollo de algoritmos y la visualización de datos. Además, debido a su amplio uso en entornos industriales existen una gran gama de *software* y herramientas con las que puede integrarse de forma fluida.

Un ejemplo destacado de estas herramientas con las que puede integrarse, es Simulink. Un entorno de programación visual basado en diagramas de bloques que sirve para diseñar, simular y desplegar código generado a

partir de los modelos sin necesidad de que el usuario escriba el código manualmente.

GNU Octave es un lenguaje de programación escrito en C, C++ y Fortran, principalmente diseñado para llevar a cabo cálculos numéricos. Tiene un alto grado de compatibilidad con Matlab y a menudo se usa como una alternativa de código abierto, aunque presenta algunas diferencias en la forma de nombrar funciones y en otros aspectos menores. Un problema que se puede enfrentar al usar este lenguaje es que, pese a tener una comunidad bastante activa, el soporte y la documentación disponibles no son tan extensos ni profesionales como los de Matlab.

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general que es conocido por su simplicidad. En los últimos años ha aumentado mucho su popularidad, siendo una de las principales herramientas usadas en la industria en un gran número de áreas, incluyendo el desarrollo web, el análisis de datos, la inteligencia artificial, el desarrollo de aplicaciones y más.

Su uso tan diverso hace que existan una gran cantidad de bibliotecas y *frameworks*, entre los que cabe destacar NumPy para la realización de cálculos numéricos y Matplotlib para la visualización de datos. Además la comunidad es una de las más grandes y activas, lo que proporciona abundantes recursos de aprendizaje y soporte.

Finalmente se decidió usar el lenguaje de programación Matlab para el desarrollo del sistema de detección, debido a que al investigar se determinó que era el lenguaje más optimizado a la hora de hacer cálculo numérico, especialmente cuando se debe operar con matrices; además ofrece una gran facilidad a la hora de generar gráficas a partir de una amplia variedad de funciones, lo que permite representar los datos con los que se está operando de manera efectiva sin necesidad de tener que instalar librerías adicionales para la visualización. El mayor inconveniente a la hora de usar Matlab es que este tiene una licencia bastante cara, llegando en algunos casos a valer cerca de mil euros, pero en este caso la Universidad de Valladolid proporciona una licencia gratuita a sus estudiantes.

La existencia de un entorno de desarrollo integrado (IDE) oficial, el cual recibe el mismo nombre que el lenguaje y se muestra en la figura 2, proporciona acceso a recursos educativos, documentación oficial, foros de la comunidad y librerías que han sido desarrolladas tanto por la comunidad como por los desarrolladores de Matlab, las cuales se pueden instalar de

Característica	**Matlab**	**Octave**	**Python**
Tipo de Software	Propietario	Abierto	Abierto
Licencia	Entre 69€ y 3650€ según versión	-	-
Finalidad	Cálculos matemáticos	Cálculos matemáticos	General
Uso Principal	Industria y academia	Academia	Amplio espectro
Bibliotecas y Herramientas	Amplia variedad y especializadas	Menos extensas que las de Matlab	Gran cantidad de bibliotecas cientificas y de datos
Gráficos y visualización	Gran variedad de gráficas y potentes herramientas	Menos opciones que Matlab	Matplot, Seaborn y Plotly para visualización
Comunidad y Soporte	Amplia comunidad y soporte profesional	Comunidad activa pero menor a la de Matlab	Comunidad muy amplia con un extenso soporte
Desempeño	Optimizado para realizar cálculos	Puede ser más lento que Matlab	Eficiente, pero puede ser más lento al realizar cálculos intensivos
Integración	Con herramientas industriales	Poca integración con otras herramientas	Integracion con otros lenguajes y tecnologías

Tabla 1: Comparativa de los posibles lenguajes sobre los que desarrollar la herramienta de detección.

forma sencilla. También cuenta con las opciones de soporte y de ayuda por parte de miembros del equipo técnico de MathWorks que ayudan mucho a todos aquellos que estén iniciándose en este lenguaje o a aquellos que ya sean veteranos.

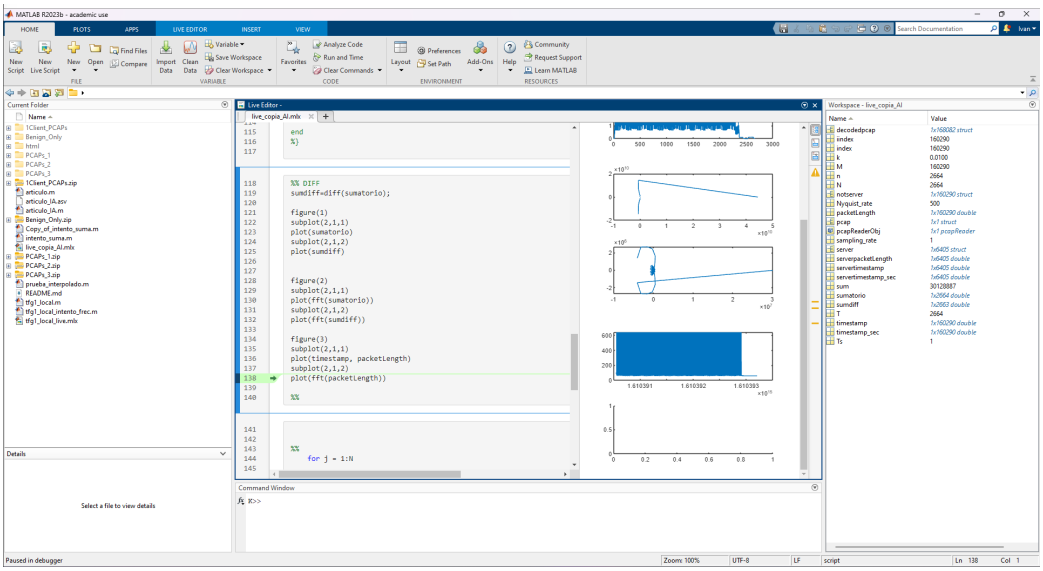

Figura 2: Interfaz de usuario de Matlab. Se han eliminado las líneas que revelan la ubicación de los ficheros en el ordenador.

Una herramienta adicional que se ha usado como apoyo es Wireshark. Este software es un analizador de paquetes de red de código abierto que permite examinar los detalles del tráfico en varios niveles, para realizar análisis, solucionar problemas en redes de comunicación y efectuar auditorías de seguridad gracias a que captura todo tipo de información que pasa por una conexión.

Además de que permite ver el tráfico en un momento dado, también permite examinar los datos contenidos en ficheros de capturas previamente guardadas. Dentro de la interfaz de usuario, que se muestra en la figura 3, se puede ver que al seleccionar un paquete se devuelve un análisis profundo de cada uno de los protocolos de red. Como en algunos casos, al estar en modo promiscuo, el ordenador puede llegar a mostrar en pantalla un número de paquetes difícil de manejar, esta herramienta posee la opción de aplicar filtros de captura o de visualizado en función de las necesidades de los usuarios.

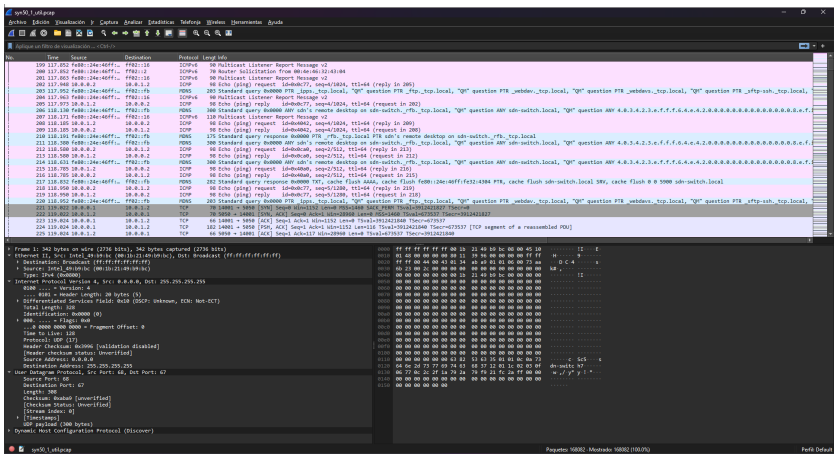

Figura 3: Interfaz de usuario de Wireshark.

1.3.2 *Hardware*

En el ámbito del desarrollo y la ejecución de *software* las características *hardware* juegan un papel fundamental en el rendimiento y la eficiencia de los procesos. Este punto se va a centrar en llevar a cabo un análisis de dos ordenadores utilizados en las fases de desarrollo y ejecución del código de detección. Mediante una comparación de sus componentes se pretende destacar la relación entre las especificaciones técnicas de los dispositivos y el comportamiento y la eficiencia de las tareas computacionales.

Se examinarán aspectos como el procesador, la memoria RAM, el almacenamiento, la tarjeta gráfica y otro componentes relevantes que impactan en la capacidad de los sistemas. Para ello se va a proceder a la elaboración de una tabla que recoja todas estas características de los equipos que se han empleado: estos serían un ordenador de sobremesa y un portátil .

Dispositivo	**CPU**	**Velocidad Base**	**Núcleos**	**Memoria RAM**	**Almacenamiento HDD**	**Almacenamiento SSD**
Sobremesa	AMD Ryzen 9 5900X	3.7 GHz	12	32 GB	4 TB	4.5 TB
Portátil	Intel i7-4510U	3.1 GHz	2	8 GB	-	1 TB

Tabla 2: Especificaciones de los ordenadores empleados.

1.4 Estructura de la Memoria

La estructura de este trabajo consta de **8** capítulos principales, empezando por el actual, en el que se introduce la temática y se describen pautas y objetivos que forman el estudio, además de presentar cuál es la razón tras la elección de este tema y explicar a qué se ha debido la elección de las herramientas *software* que se han empleado para la realización del proyecto y los dispositivos a los que se ha tenido acceso para ello.

En segundo lugar, el capítulo 2 habla de la historia y realiza un análisis de las características de los ataques DoS, DDoS y LDDoS. Además se hablará de algunos ataques famosos y de las repercusiones que tuvieron para dar a entender la magnitud que pueden tener los ataques de denegación de servicio. En el caso de los ataques LDDoS se describen algunas de las estrategias de ataque más famosas y la evolución que estas han sufrido en algunos casos.

El capitulo 3 se utilizará para ver algunas de las soluciones existentes que se han desarrollado en los últimos años y que se basan en distintas formas de analizar los datos para determinar si existe la presencia de algún tipo de anomalía en estos. Esto nos permitirá sacar conclusiones de cuáles son algunas de las mejores prácticas y enfoques a la hora de analizar tráficos de ataque en redes, y nos permitirá comparar distintos métodos de análisis y de inferencia para determinar en qué casos es más recomendable aplicar cada uno.

Posteriormente, en el capítulo 4, se presentarán posibles problemas que pueden estar presentes al seleccionar algunos métodos de detección y que podrían comprometer la seguridad de las redes en las que se vean implantados, al no ser capaces de llevar a cabo una detección eficaz de los ataques.

En el capítulo 5 se analizarán los métodos empleados en la creación de una herramienta de detección con la que se pretende comprobar algunas de las ideas presentadas en la sección anterior y se presentan las bases de datos que se han utilizado para hacer las pruebas. En esta sección también se mostrarán los resultados que se han obtenido al emplear el sistema de detección sobre estos datos simulados.

Este Trabajo de Fin de Grado pone el punto final en el capitulo 6 con una conclusión global y un desarrollo en el que se recogen posibles vías de investigación para mejorar la herramienta o ampliarla de cara a futuras líneas de desarrollo.

Se completa este trabajo con dos apéndices: el Apéndice A, en el cual se recoge el código que se ha desarrollado en Matlab, con comentarios explicativos de cada una de las secciones de las herramientas desarrolladas; y el Apéndice B, donde se recogen los desarrollos teóricos usados para respaldar la teoría principal que se estudia en este trabajo. Para finalizar, se presentan las referencias que se han usado para apoyar las ideas que se presentan a lo largo del trabajo.

2 Ataques de Denegación de Servicio Distribuidos de Baja Tasa (LDDoS)

En la bibliografía actual son pocos los autores que definan los términos LDoS o LDDoS sin antes aportar una pequeña explicación acerca de qué son los ataques de denegación de servicio y cuáles son sus principales características. Por ello este capítulo seguirá los distintos pasos que han seguido los ataques de denegación de servicio a lo largo de su evolución.

En la primera sección se analizará el concepto de los ataques DoS y los momentos más relevantes de su historia, como su aparición y algunos ejemplos de ataques que dejaron grandes impactos. Seguidamente, se analizarán las características de los ataques DDoS y los eventos relevantes que han ocasionado, y por último, pasaremos a centrarnos en los más recientes y los que son de mayor interés para nuestro estudio, los ataques LDDoS. De estos analizaremos tecnologías y métodos de estudio relevantes a su caracterización antes de pasar a introducir el modelo general de estos ataques y las estrategias más relevantes.

2.1 Denegación de servicio (DoS)

Tradicionalmente, cuando se habla de ciberseguridad se suele hacer referencia a las siglas CIA, que representan los conceptos de Confidencialidad, Integridad y Disponibilidad (*Confidentiality, Integrity and Availability* en inglés). Los ataques de denegación de servicio atentan contra la disponibilidad de los sistemas.

Cuando una entidad o algún usuario de la red quiere dar un servicio por medio de Internet, este debe emplear una serie de recursos, como pueden ser los ciclos de CPU, la memoria, el ancho de banda de los enlaces o los búferes de conexión. Estos recursos son fundamentales para garantizar el funcionamiento de los servicios y su accesibilidad por parte de los usuarios.

Durante un ataque DoS un atacante, por medio de un único host o de un grupo reducido que se encuentra en una misma localización, trata de agotar estos recursos, haciendo que el sistema deje de ser accesible, o de reducirlos, afectando así a la calidad del servicio (QoS) ofrecida a los usuarios. Al leer [MF09] y [Yu13] vemos que ambos autores están de acuerdo en que existen un gran número de estrategias mediante las cuales se pueden lanzar

estos ataques, pero que generalmente se pueden diferenciar en 2 categorías:

- *Ataques a vulnerabilidades:* Estos ataques usan mensajes elaborados basándose en los puntos débiles de la víctima y logran que esta quede bloqueada o caiga. La solución a este tipo de ataques es la más sencilla, ya que únicamente requiere que se aplique un parche, solventando así las vulnerabilidades del sistema.

- *Ataques de inundación:* En esta categoría el atacante manda una gran cantidad de tráfico inútil a la víctima, a fin de reducir el ancho de banda o agotar los recursos que un host emplearía para dar servicio al tráfico legítimo. Esto lo hace fácil de detectar. Al ser un ataque que envía un volumen de tráfico muy superior al normal durante un largo periodo de tiempo, la detección puede ser tan simple como fijar un umbral que se sitúe por encima de la tasa de tráfico que suele tener el servidor bajo condiciones normales.

2.1.1 Historia de los ataques DoS

La red tal y como la conocemos tiene su origen en el año 1969. Fue en este año cuando se estableció el primer enlace de ARPANET. Esta conexión iba desde la Universidad de California, Los Ángeles (UCLA) hasta el Instituto de Investigación de Standford (SRI) y fue por medio de este enlace por el cual Leonard Kleinrock y Douglas Engelbart mandaron el que sería el primer mensaje entre 2 ordenadores [Lei09]

Tan solo 5 años después David Dennis llevaría a cabo el primer ataque DoS deliberado. D. Dennis en aquel momento tan solo tenía 13 años y era un estudiante de University High School. Tras haber escuchado de la existencia de un comando de los terminales PLATO (Programmed Logic Automated Teaching Operations), el primer sistema de enseñanza asistida por ordenador, que permitía interactuar con los periféricos conectados a este, y que tenía una vulnerabilidad que causaba que el terminal que no tuviera periféricos quedara bloqueado hasta que se reencendiera, surgió su curiosidad.

Queriendo averiguar qué pasaría si todos los ordenadores de una sala quedaran en ese estado de bloqueo, procedió a escribir un código que enviaría ese comando a un gran numero de equipos al mismo tiempo. De esta forma 31 ordenadores quedaron bloqueados y la aceptación de la instrucción *ext* remota paso a desactivarse, solucionando el problema [Dea10]. Como se puede ver en este ejemplo este ataque explotaba una vulnerabilidad en los

terminales.

Durante la década de 1990, comenzaron a popularizarse los *Internet Relay Chat* (IRC). Con la aparición de canales de chat no registrados comenzaron también guerras por lograr obtener los poderes de administrador, haciendo que el administrador fuera desconectado. De esta forma usuarios peleaban por el control lanzándose altas cantidades de tráfico con las que agotaban los recursos de sus rivales, forzando la desconexión de los canales por medio de ataques de inundación [Hir19].

Este año 2024 los ataques DoS cumplirán 50 años. Pese a esto la amenaza que constituyen para las infraestructuras de red sigue siendo un riesgo importante, que debe ser tenido en cuenta por los administradores de sistemas y por el personal de ciberseguridad de entidades, ya sean empresariales o gubernamentales. La tendencia actual a la hora de realizar los ataques es recurrir al uso de grandes cantidades de ordenadores en lugar de uno solo, es decir, que cada vez más se están abandonando los ataques DoS para lanzar DDoS.

2.2 Denegación de Servicio Distribuida (DDoS)

Este ataque es de los más comunes en la red actual tal y como se muestra en la figura 1, habiéndose convertido sus siglas en el icono representativo de todos los ataques de denegación de servicio independientemente de que estos sean DoS, DDoS o LDDoS. Pese a esto, en este apartado se va a prestar especial atención a las características que lo distinguen del ataque presentado anteriormente y se van a ver algunas de las variedades que pueden tener además de algunos ejemplos reales.

Los ataques de denegación de servicio distribuidos o DDoS se caracterizan por usar una gran cantidad de hosts como atacantes, que pueden encontrarse en torno a los cientos o los miles, logrando generar de esta forma cantidades de trafico con las que llevan a cabo una inundación masiva con la que se logra abrumar a la víctima [Kas12]. Un ejemplo con el que se puede comprender este problema es si se piensa en una tienda y lo que pasaría si todos los habitantes de una ciudad decidieran ir al mismo tiempo a comprar a esta, al haber tantos clientes no se podría atender a todos y estos se verían forzados a hacer largas colas.

Para lograr tener acceso a tal magnitud de ordenadores los atacantes

primero deben haber construido una red de ordenadores sobre los que han tomado el control. Esta red recibe el nombre de "botnet". Estas redes tienen una estructura jerárquica que se muestra en la figura 4, compuesta por 3 tipos de equipos [Wu11]:

- *Atacante:* Es el controlador detrás del ataque, ordena a todos los maestros iniciar el ataque.
- *Maestros:* Estos equipos también pueden recibir el nombre de servidores *command and control* (C&C) y son equipos comprometidos que son capaces de controlar a varios agentes o bots.
- *Agentes:* Son hosts comprometidos popularmente llamados zombies o bots. En ellos se está ejecutando un programa por medio del cuál reciben las ordenes de los C&C y generan los flujos de tráfico que forman el ataque.

Una ventaja de usar un gran número de equipos en comparación a uno solo es que a la hora de tomar medidas tras la detección del ataque es mucho más difícil de bloquear la ubicación de la que proviene el tráfico de ataque y también dificulta la localización del atacante principal. Existen muchas técnicas por medio de las cuales los atacantes pueden hacerse con el control de equipos para agregarlos a la botnet. Las más comunes recurren a la ingeniería social para hacer que un usuario descargue un malware a su ordenador.

Estos hosts infectados, bots, pueden operar de formas diferentes que los separa en amplificadores y reflectores. Los amplificadores son capaces de aumentar el volumen del ataque usando direcciones de difusión (broadcast) como destino de retorno de los paquetes. Los ataques que usan estos mecanismos se conocen como ataques de amplificación [Far05]. Los reflectores devuelven uno o más paquetes por cada paquete recibido, los servidores Web y DNS se pueden incluir en esta categoría, ya que envían paquetes de respuesta a todas las peticiones que reciben. Este tipo de ataque es llamado Denegación de Servicio Distribuido Reflejado (DRDoS, *Distributed Reflection/Reflective Denial of Service*) [INC21].

2.2.1 Historia de los ataques DDoS

Los ataques de Denegación de Servicio siguieron evolucionando: en 1996 uno de los primeros proveedores de servicios de Internet (*Internet Ser-*

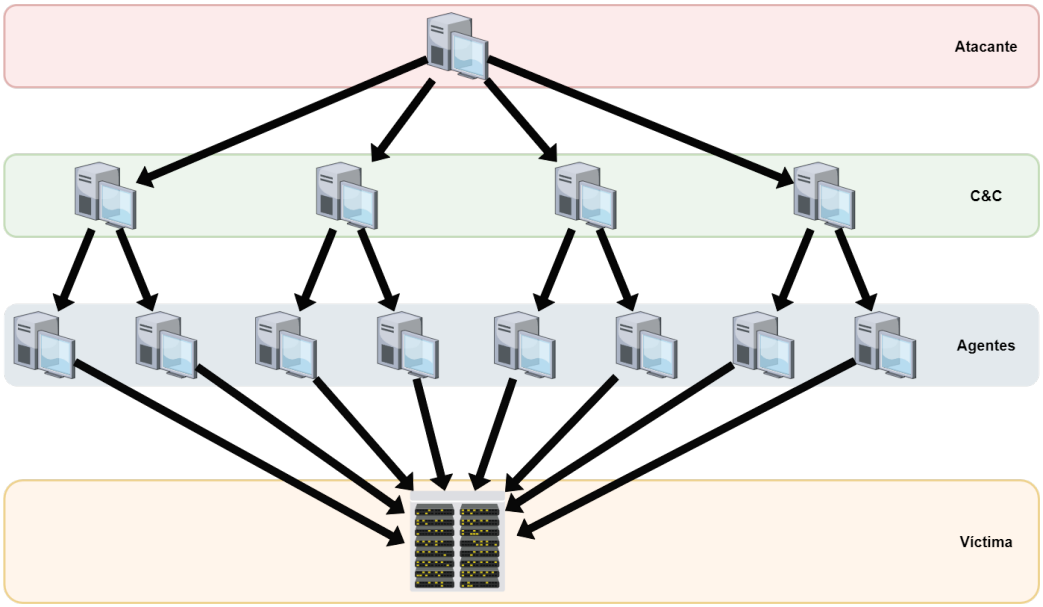

Figura 4: Estructura de una red botnet.

vice Provider, ISP) fue afectado por un ataque distribuido bajo el nombre de SYN flood o inundación SYN del que tardó 36 horas en recuperarse, convirtiendo a este caso en el primer ataque DDoS conocido [Cal96].

El ataque SYN flood consiste en enviar un gran número de paquetes de solicitud de conexión (SYN) a la víctima usando una dirección IP falsificada (IP spoofing). De esta forma el atacante logra sobrecargar las máquinas que actúan como servidor haciendo que no pueda garantizar el servicio a los clientes legítimos.

Años después se produciría el primer ataque DDoS a gran escala, que usaba una herramienta especializada llamada Trinoo. Por medio de esta herramienta un hacker logro abrumar las redes internas de la Universidad de Minnesota durante más de dos días por medio de una inundación UDP lanzada desde 114 equipos infectados con Trinoo [Ost19].

El año 2000 fue muy importante en todo el ámbito de la informática debido al temor por el llamado efecto 2000 o Y2K, pero en el ámbito de los ataques de Denegación de Servicio también es un año con una gran relevancia. Esto se debe a Michael Calce, también conocido como “MafiaBoy”.

A inicios de febrero de ese año, Michael Calce lanzó su primer ataque DDoS contra las infraestructuras de Yahoo logrando desconectar su página

web durante una hora. Esto constituyó el primer paso en una serie de ataques bajo el nombre de Proyecto Rivolta. Como respuesta a este acto, otro hacker lanzó un ataque contra Buy.com, cosa que "MafiaBoy" tomó como un desafío para probar sus capacidades como hacker, lo que dió lugar a una serie de ataques contra los servidores de FIFA, eBay, CNN y Amazon, entre otras. Todas estas entidades cayeron frente al ataque.

La respuesta por parte de los gobiernos canadiense y estadounidense fue lanzar una investigación dirigida por el FBI y la Real Policía Montada de Canadá, los cuales lograron dar con Michael Calce gracias a agentes infiltrados en uno de los grupos de hackers a los que este pertenecía. Durante el juicio se le acusó de haber generado pérdidas por valor de 7.5 millones de dólares y varias fuentes apuntan a que estos ataques hicieron que se endurecieran las leyes relativas a delitos informáticos en EE.UU. [Maf22].

En los últimos años, con el desarrollo de Internet de las Cosas o *Internet of Things* (IoT), el número de dispositivos que están conectados a la red ha aumentado, y con ello el número de hosts que pueden ser infectados. Esto ha hecho posible que los ataques que se realizan tengan cada vez picos de tráfico mayores. Esta evolución es fácilmente visible en la figura 5.

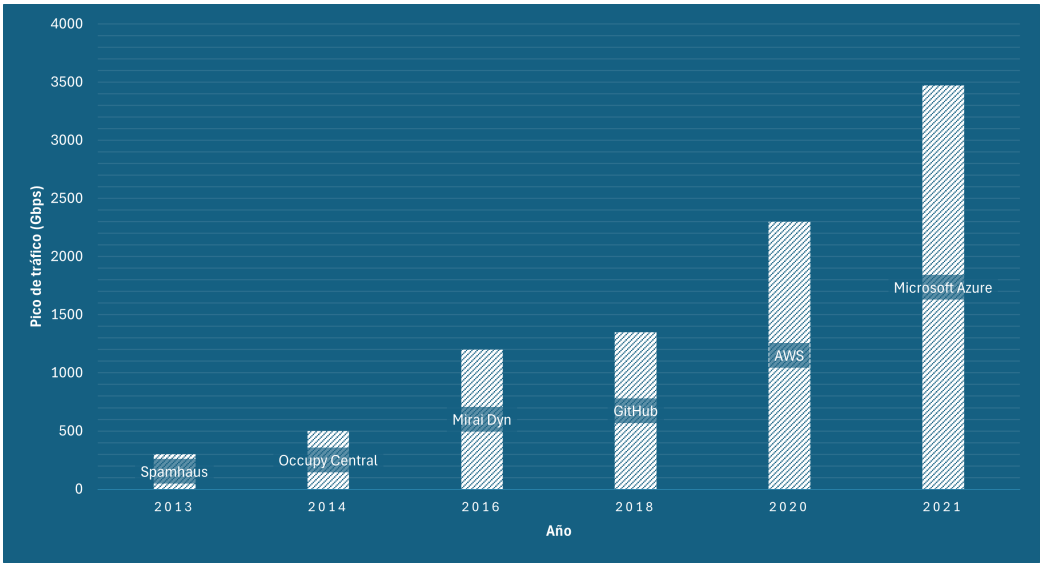

Figura 5: Comparativa de los mayores picos de tráfico de ataques DDoS.

En la figura se mencionan algunos de los ataques más relevantes de la última década. A continuación se va a explicar brevemente en qué consistieron y cuáles fueron los motivos tras estos.

El ataque a Spamhaus fue un ataque que ocurrió en 2013. El servicio anti-spam fue golpeado con un pico de 300 Gigabits por segundo (Gbps), lo que lo convirtió en el mayor ataque registrado hasta ese momento, el cuál fue lanzado por Cyerbunker, una compañía de hosting que proporciona sus servicios a casi cualquier web entre las que se encuentran algunos de los mayores spammers [Vij13].

Occupy Central era una campaña de desobediencia civil que surgió en Hong Kong en el año 2014 a favor del sufragio universal. Durante esta campaña fueron numerosos los ataques DDoS que se realizaron contra medios independientes de Hong Kong. Entre todos estos el más destacado fue el ataque a PopVote que tuvo un pico máximo de 500 Gbps debido a que se emplearon cinco botnets diferentes. PopVote era una página donde se realizaban votaciones simuladas [Ols14].

Durante 2016 la botnet Mirai lanzó una serie de ataques que superaron ampliamente todos los récords anteriores. Mirai es un malware de estilo gusano que infecta a dispositivos IoT forzándolos a formar parte de una red maliciosa [Ant22]. Entre estos ataques, el que sufrió el proveedor de servicios de nombres de dominio (DNS) Dyn fue el mayor. Como consecuencia de este ataque muchos servicios como Netflix, GitHub, Reddit o Twitter quedaron inaccesibles [VN16]. El motivo tras este ataque es desconocido a día de hoy, pero no faltan las fuentes que sugieren que pudo deberse a un truco publicitario o que pudo ser una protesta contra la persecución a Julian Assange.

Pero incluso frente a amenazas de estas magnitudes existen quienes logran"sobrevivir". En 2018 GitHub sufrió un ataque que tenía un factor de amplificación de alrededor de 50000, lo que permitió superar la magnitud del ataque que sufrió Dyn. Los responsables tras este ataque se cree que fueron hackers respaldados por el gobierno chino [New18].

Durante los últimos años las tecnologías que proporcionan servicios en la nube han cobrado mucha importancia y una gran cantidad de negocios han realizado una migración a estas, lo que ha creado un nuevo objetivo para todos los atacantes al poder inutilizar una gran cantidad de servicios atacando solo a un objetivo. Los últimos 2 ataques de la figura 5 fueron contra dos de las plataformas más relevantes de computación en la nube: Amazon Web Services (AWS) y Microsoft Azure. Los ataques tuvieron unos tráficos que sobrepasaban los 2 Tbps.

En el caso de AWS los recursos de la empresa no fueron capaces de resistir el ataque y el servicio se vio interrumpido, aunque los efectos no fueron

tan severos como se esperaría [Pin20]. Por el otro lado, el ataque DDoS más grande registrado hasta la fecha fue ejecutado contra un cliente de Azure. Gracias a los sistemas de protección de la plataforma de Microsoft se pudo mitigar el incidente al escalar los recursos para absorber el tráfico [Azu22].

Como se puede ver, muchos de estos ataques tenían la ideología como motivación. Un ejemplo muy claro de esto puede ser observado en los informes de amenazas que empresas como Cloudflare han publicado en los últimos dos años. En su informe del cuarto cuatrimestre de 2023 dedicaban una buena parte a los ataques DDoS y su relación con la operación *Iron Swords*, nombre bajo el que se conoce la ofensiva militar lanzada por Israel contra Hamás [Yoa24]. Algo similar se vio a lo largo de los informes publicados en el año 2022, donde dedicaban una sección a la guerra ruso-ucraniana.

Figura 6: Ataques DDoS contra sitios web israelíes y palestinos, clasificado por sector [Yoa24].

2.3 Ataques de Denegación de Servicio de Baja Tasa (LDoS y LDDoS)

2.3.1 Introducción

A lo largo de los años el desarrollo de redes de gran escala a hecho que surjan nuevas estrategias de ataque cada vez más complejas. Esto ha ocasionado la aparición de una nueva categoría de ataques de denegación de servicio: los ataques DoS de baja tasa (LDoS) y Low-Rate DDoS (LDDoS).

Los ataques LDDoS presentan un gran cambio frente a los ataques de inundación, también conocidos como Flood DDoS, ya que en lugar de emplear una gran cantidad de recursos se centran en causar el mayor daño posible mientras usan una baja tasa de tráfico, es decir, optimizan el uso de los recursos del atacante para agotar parcial o completamente los recursos de la víctima.

Esta optimización les permite pasar desapercibidos en el tráfico de las grandes redes mientras afectan a las conexiones legítimas y por ello la principal víctima de estos ataques son las plataformas de computación en la nube y los centros de datos, que acostumbran a dar un servicio bajo demanda. Por ello los ataques LDDoS pueden ocultarse de los sistemas de detección mientras roban flujos que estarían destinados a clientes.

Otra ventaja que presentan para los atacantes estos tipos de ataque es que requieren de un menor número de zombies comparado con los ataques de fuerza bruta más tradicionales. Por ello representan un nuevo desafío para las herramientas de detección que se encuentran actualmente en el mercado.

En la actualidad existen varias estrategias de denegación de servicio que usan esta aproximación, como se puede ver en el punto 2.3.5, pero todas ellas tienen una estructura similar, la cuál se va a presentar en esta sección.

Antes de comenzar a explicar los principios en los que se basan estos ataques es importante comprender que muchos de ellos tienen como base algunas de las características de *Transmission Control Protocol*. Por ello es importante comprender en qué se basa este protocolo y cuáles son sus propiedades.

La complejidad que estos ataques presentan a la hora de modelarse hace que el desarrollo de un modelo analítico para evaluar su rendimiento

sea de gran dificultad. Por ello en esta sección se detallan las características comunes que pueden apreciarse al estudiarse diferentes estrategias de ataque entre las que se encuentran el modelo desarrollado por G. Maciá-Fernandez et al. en [MF08b] o el modelo de A. Kuzmanovic y E. W. Knightly en [Kuz03].

A continuación, en las siguientes subsecciones se va a explicar: el protocolo Transmission Control Protocol (TCP), que es el mecanismo por sobre el que se llevan a cabo la mayoría de las conexiones de Internet; la teoría de colas, que es un mecanismo que permite elaborar modelos matemáticos con los que estudiar el comportamiento de los sistemas telemáticos; y para finalizar, se explicará el modelo general de los ataques LDDoS y varias estrategias por medio de las que puede llevarse a cabo. Todo lo anteriormente mencionado es importante para la elaboración de este TFG, debido a que el protocolo TCP presenta una serie de mecanismos habitualmente explotados por los atacantes durante la configuración de los parámetros de los ataques LDDoS. Además, la teoría de colas permite que se lleven a cabo estudios acerca de cómo se verán afectados los sistemas víctimas de los ataques sin tener que elaborar complejas simulaciones.

2.3.2 Transmission Control Protocol (TCP)

En [Bla09] se define *Transmission Control Protocol (TCP)* y sus características. TCP es el protocolo de capa de transporte sobre el que se realizan la mayoría de las conexiones de Internet. Permite la comunicación entre dos hosts que han negociado una serie de parámetros relativos a las características de la conexión antes de iniciar la transmisión de los datos (orientado a conexión), y garantiza la entrega de paquetes sin errores (fiable) y en el mismo orden en el que se enviaron.

Para lograr estas tres características posee una serie de mecanismos mediante los cuales es capaz de determinar la tasa de envío adecuada de la conexión y la recepción correcta o fallida de los mensajes enviados. Además, por medio de estos métodos, los extremos finales de las conexiones son capaces de coordinarse para lograr usar de forma eficiente sus recursos.

En aquellos casos en los que estos mecanismos se ven sobrepasados los sistemas terminan congestionados, lo que lleva al descarte de paquetes que deberán ser reenviados. Esto es lo que ocurre en los ataques de denegación de servicio. A continuación, se describen algunos de los términos relacionados con las conexiones TCP y que guardan mucha relación con los ataques

LDDoS:

- *Tiempo de Ida y Vuelta o Round-Trip Time (RTT):* hace referencia al tiempo que pasa entre la emisión de un segmento TCP y la recepción del asentimiento correspondiente.

- *Asentimiento o Acknowledgement (ACK):* es un mensaje que confirma al emisor de un segmento TCP que este ha llegado al otro extremo correctamente. Su contrapartida es el NACK (*Negative ACK*), el cual informa que ha ocurrido un error en la recepción.

- *Ventana de Recepción o Reception Window (rwnd):* es una variable que anuncia la cantidad de bytes que el destino de la comunicación puede recibir en un solo segmento. Un segmento es el paquete de bytes que constituyen la unidad de datos del protocolo TCP.

- *Ventana de Congestión o Congestion Window (cwnd):* es una variable que limita la cantidad de datos que se pueden enviar. En todo momento se debe cumplir que el último número de secuencia no debe ser mayor a la suma entre el último número de secuencia asentido y el mínimo entre cwnd y rwnd.

- *Slow Start:* forma parte de un mecanismo de control de congestión que controla la tasa a la que se envían los paquetes. Durante esta fase el número de paquetes enviados aumenta exponencialmente hasta que la cwnd llega al ssthresh (*slow start threshold*, véase más abajo).

- *Evitación de Congestión o Congestion Avoidance:* este algoritmo comienza a emplearse cuando cwnd esta por encima de ssthresh. Durante esta fase el crecimiento de la cwnd es mucho menor, siendo al ritmo de 1 segmento por cada *RTT* hasta que ocurre una pérdida.

- *Slow Start Threshold (ssthresh):* es un límite variable qué determina en que momento se debe usar el algoritmo de *slow start* o el de *congestion avoidance* para controlar la transmisión de datos.

TCP posee varias maneras por las que es capaz de recuperarse de las pérdidas de segmentos, pero en este caso la que más nos interesa es la recuperación por medio de temporizador. En aquellos casos en los que el emisor de un segmento no obtenga respuesta de parte del receptor tras el envío de datos TCP o reciba menos de 3 ACKs duplicados empleará un temporizador de retransmisión que recibe el nombre de *retransmission timeout* (*RTO*) para asegurar que los datos son recibidos correctamente.

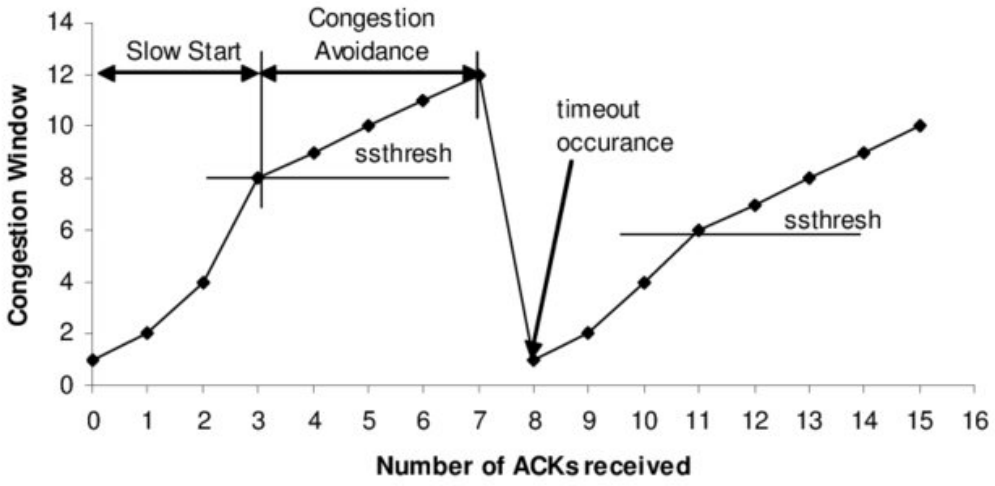

Figura 7: Evolución de la ventana de congestión de TCP [Kli07]

El valor de este temporizador inicialmente está fijado a 1 segundo ($minRTO$) de acuerdo a lo establecido en el RFC 6298, [Sar11], en el cual también se detalla que el RTO depende de todos los valores anteriores del RTT y de la desviación que este presente, además de la granularidad del reloj usado para los cálculos.

Sabiendo esto podemos ver cuáles son los pasos que tienen lugar en una conexión donde este temporizador tenga efecto. El extremo emisor A enviará datos al extremo receptor B y pondrá en marcha el RTO después de enviar el último segmento. Tras recibir estos B enviará un asentimiento correspondiente a cada paquete, lo que reinicia el temporizador. Pero en el caso de que el último de los paquetes de A se perdiera, el temporizador llegaría a cero y forzaría el reenvío de dicho paquete, recuperándose de la pérdida mientras el valor de RTO se multiplica por 2.

2.3.3 Teoría de Colas

La teoría de colas es una herramienta matemática que permite estudiar el comportamiento de sistemas que tienen un conjunto de recursos limitados para atender una serie de peticiones emitidas por los clientes, para de esta forma poder escalar adecuadamente los recursos y gestionar los sistemas de esperas.

Esta teoría estudia sistemas en los que existe una competición por los recursos, por lo que en algunos momentos es posible que existan más peticiones de las que es posible atender, teniendo que esperar para ser atendidas o descartadas en aquellos casos en los que el sistema esté congestionado, es decir, en aquellos casos en los que las colas de acceso al servicio estén llenas.

En el libro escrito por P. Arias et al. [PA03], se explican en detalle todas las características fundamentales de la teoría de colas. A partir de esto podemos establecer que para poder modelar un sistema de colas se debe prestar especial atención a los siguientes puntos:

1. *Patrón de llegadas:* las peticiones que el sistema recibe llegan en momentos que no pueden predecirse y pueden distinguirse en llegadas independientes o dependientes, en función de si el número de clientes que hayan enviado peticiones al servidor afecta a los tiempos de llegadas.

2. *Mecanismo de servicio:* cada petición que llega al sistema tarda un tiempo en recibir el servicio. Esto se denomina tiempo de servicio, y puede depender de la configuración del sistema.

3. *Disciplina de la cola:* establece la forma en que las distintas peticiones que llegan a la cola ganan el acceso al servicio. La más común es la disciplina llamada *First Come First Served* (FCFS). Esta forma de dar servicio garantiza que los usuarios son atendidos en el mismo orden en el que llegan a la cola. En algunos casos puede no ser la más apropiada ya que la duración de algunas tareas podría dejar el sistema bloqueado durante largos plazos.

2.3.4 Modelo general de un ataque LDDoS

Antes de dar un modelo general para los ataques se deben establecer las suposiciones bajo las cuales se estará trabajando. Si bien distintas estrategias pueden tener diferentes parámetros, todas tienen en común las mismas consideraciones. El patrón de llegada de las peticiones por parte de los clientes legítimos depende de un proceso de Poisson con una tasa λ_c, y la capacidad de las colas de servicio se denota como C.

Además, muchos ataques asumen que el servidor víctima es un servidor iterativo, es decir, que solo atiende una petición en cada instante, y que las colas que posee almacenarán las peticiones recibidas en el mismo orden en el que llegaron hasta que se termine de dar servicio a la petición que se estaba

procesando, momento en el que la petición más antigua pasará al servidor y una posición de la cola se liberará. Gracias a la simplificación introducida por la teoría de colas, los sistemas modelados sobre los cuales se efectúan los ataques son fácilmente escalables.

Una vez que se han repasado los conceptos anteriores podemos pasar a definir las características de los ataques LDDoS, que son la agregación de múltiples procesos LDoS. Los ataques LDoS consisten en el envío malicioso de paquetes de ataque a una víctima, de tal forma que se logre llevar a cabo un ataque DoS a una baja tasa de tráfico.

En la literatura, estos ataques son en esencia ráfagas que explotan la homogeneidad de mecanismos como el temporizador de retransmisión (RTO) de los flujos TCP, [Kuz03], o los tiempos de servicio (Ts) para peticiones del mismo tamaño, [MF08b]. Por lo general estas amenazas se pueden expresar en función de 3 parámetros clave que representaremos como R, T_n y d, como se puede ver en la figura 8 y se describen a continuación.

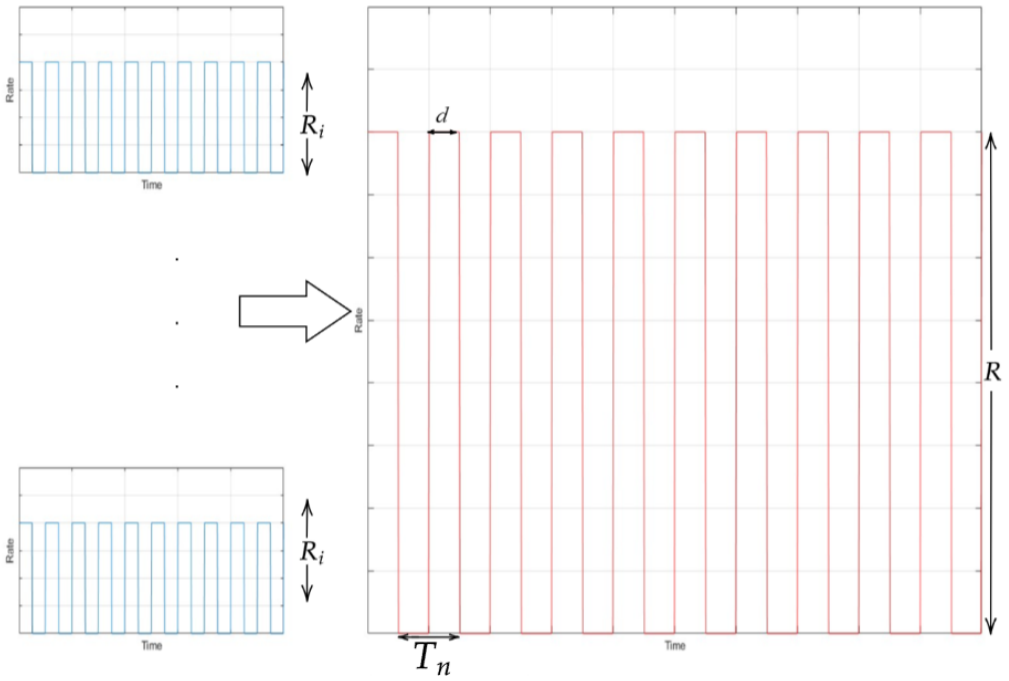

Figura 8: Formación de un ataque LDDoS a partir de varios flujos LDoS.

El periodo en el que se repiten los pulsos de ataque será representado como T_n, el valor que tendrá el periodo será de una escala mayor a la duración de la ráfaga y dependerá de la estrategia que se use en el ataque, es decir, depende de la vulnerabilidad a explotar en el sistema de la víctima.

d es la forma con la que se expresa la duración de las ráfagas de ataque, es decir, los momentos en los que el atacante se encuentra activamente transmitiendo mensajes de ataque. El tiempo que dura cada ráfaga será por lo general de alguna magnitud menor a la de T_n y es inversamente proporcional a la tasa R, ya que cuanto menor sea el d deseado, mayor deberá ser R para garantizar el envío de los paquetes de ataque de forma que la ecuación resultante es:

$$d = \frac{1}{R} \tag{1}$$

El valor de la duración deberá ser lo suficientemente largo como para inducir la denegación de servicio pero no lo suficientemente largo como para poder ser detectado, tal y como ocurre con las denegaciones de servicio por inundación, en las que podríamos considerar que el valor de d es el mismo que el de T_n.

La tasa de cada una de las ráfagas del ataque se denota como R, que en el caso de ser un ataque LDDoS será equivalente al sumatorio de los volúmenes de todas las ráfagas emitidas por los miembros de la botnet que efectúa el ataque. De esta forma la ecuación resultante sería la siguiente:

$$R = \sum_{i=1}^{N} R_i \tag{2}$$

donde N es el número de zombies que forman parte de la botnet desde la que se efectúa el ataque.

2.3.5 Estrategias de ataque mediante LDDoS

Existen una gran cantidad de técnicas por medio de las cuales se pueden lanzar ataques. Cada una de estas formas tiene sus propias características que dependen de la cantidad de daño que se pretende infligir en la víctima y de las vulnerabilidades que se pretende explotar. Por ello esta sección se va a destinar a analizar algunos de los mecanismos más conocidos por medio de los que se pueden lanzar ataques LDDoS. Para ello se va a prestar especial atención a las particularidades de los sistemas a los que afectan y cómo estas determinan los parámetros del ataque.

2.3.5.1 *Shrew*

Uno de los primeros ataques de baja tasa de los que se tiene constancia es el llamado *Shrew Attack*, que se traduciría como ataque de la musaraña. Estos pequeños mamíferos son uno de los cazadores más efectivos del reino animal gracias a su habilidad para matar presas más grandes que ellos usando un veneno que deja a sus víctimas en un estado comatoso. Este ataque es descrito por primera en el año 2003 en [Kuz03]. Este artículo explica que el ataque se basa en la explotación de la homogeneidad que presenta el RTO en las conexiones TCP por medio del envío de pulsos cuadrados que forman ráfagas maliciosas.

El ataque descrito por A. Kuzmanovic y E. W. Knightly trabaja con dos escalas temporales, la menor de ellas será la de los tiempos de ida y vuelta, los cuales oscilan entre las decenas y centenas de milisegundos, y la mayor corresponderá a la escala del temporizador de retransmisión, que es de segundos. El primer objetivo del ataque *Shrew* será que al agregar el tráfico normal recibido por el servidor con el volumen R del ataque se deberá exceder la capacidad del enlace que forma el cuello de botella de la comunicación, C, durante un tiempo d, que pertenece a la escala temporal de RTT, se producirán pérdidas y los flujos TCP entrarán en *timeout* e intentarán enviar un nuevo paquete RTO segundos después.

Si el periodo T_n se aproxima al valor del RTO del flujo TCP este quedará atrapado en un bucle en el que continuará teniendo pérdidas mientras trata de salir del estado de *timeout*, por lo que el rendimiento de la conexión será casi nulo. En aquellos casos en los que el valor de T_n sea cercano al valor del temporizador pero esté fuera del rango, ocurrirá una degradación parcial del rendimiento. Por ello es crítico que el periodo sea lo más cercano posible al valor de temporización de forma que el pulso activo esté dentro del rango.

Como se ha visto en la sección 2.3.2 el valor del RTO se duplica mientras la ventana de congestión reduce su valor a 1; si el paquete sigue siendo perdido se seguirá duplicando al finalizar cada nuevo temporizador, pero si por el contrario el paquete logra ser transmitido correctamente al finalizar el temporizador ($t = 1$) su respuesta llegará en el instante $t = 1 + RTT$, el flujo entrará en el estado *slow start*, lo que permite el envío de 2 nuevos paquetes y al recibir el asentimiento a ambos el valor de los componentes que intervienen en la formula del temporizador serán recalculados de tal forma que en la mayoría de casos el valor de este retornará a 1 segundo o no se desviará en más de 1 RTT, tal y como se demuestra en [Kuz03].

Adicionalmente, el artículo analiza el efecto del ataque bajo un diverso número de condiciones, a partir de las cuales llega a las siguientes conclusiones:

- Los ataques LDoS resultan efectivos contra flujos TCP de corta y larga duración, por ello constituyen un peligro para Internet.
- En un entorno con un valor de RTT heterogéneo la efectividad del ataque depende de los flujos con menores tiempos de ida y vuelta.
- Todo flujo periódico de baja tasa puede ser dañino para las conexiones TCP con un bajo RTT si el periodo coincide con una de las frecuencias nulas de TCP. Estas frecuencias corresponden a valores del periodo en los que el rendimiento de TCP es de 0. Esto ocurre cuando el valor de T_n es igual a $T_n = minRTO$ o $T_n = \frac{minRTO}{2}$.
- Tanto los *routers* de la red como los mecanismos en los extremos de la red solo son capaces de mitigar pero no de eliminar la efectividad del ataque. Este punto no puede asegurarse a día de hoy con los mecanismos de escalado de recursos que introducen los sistemas de computación en la nube.

Una década después de la presentación de este tipo de ataque surgieron dos nuevas versiones que buscaban mejorar el rendimiento del ataque mientras disminuían aun más la carga en los equipos del atacante, estas versiones son el ataque *NewShrew*, [Luo14], y el ataque *Full Buffer Shrew*, [Gui06, Yue16, Yue21].

En el caso de *NewShrew* se busca explotar la fase de *slow start* además de la homogeneidad del temporizador de retransmisión, y por ello se permite que los flujos legítimos entren en esa fase después de cada pulso de ataque. Mientras los flujos ven su tasa de envío incrementar exponencialmente, el atacante estará esperando a que esta llegue a un determinado valor para lanzar el siguiente pulso, de forma que el volumen de tráfico de ataque no necesitará ser tan elevado para volver a producir el vencimiento de los temporizadores. De esta forma el atacante logra una mejor efectividad en sus ataques y disminuye la tasa de sus ataques al mismo tiempo que consigue evadir las contramedidas diseñadas para *Shrew*.

Full Buffer Shrew, *FB-Shrew*, toma su periodo de los instantes de tiempo en los que el búfer de acceso al sistema se encuentra lleno y usa una tasa mínima para causar congestión a nivel de la capa de transporte. La duración del pulso, d, coincide con el tiempo que el búfer está completamente

lleno y se drenará en el momento en que el atacante deje de transmitir. Desde ese momento se mantendrá vacío hasta que la tasa de tráfico de las conexiones legítimas sea igual a la del enlace de menor capacidad. Si bien diversos autores señalan errores en la investigación de M. Guirguis, A. Bestavros y I. Matta, [Gui06], por no considerar la competencia entre los paquetes TCP y los paquetes de ataque, al asumir que todos los paquetes legítimos serán descartados, muchos están de acuerdo en que los parámetros del ataque deberán ser los siguientes, $d = RTT$ o $d = 2RTT$ y $R = C$, a fin de lograr aumentar el periodo y aprovechar los paquetes TCP para que sean estos los que ocupen las colas.

2.3.5.2 Low-Rate DoS Against Application Servers (LoRDAS)

El ataque LoRDAS (*Low-Rate DoS Against Application Servers*), en español llamado ataque DoS de baja tasa contra servidores de aplicaciones, fue presentado por G. Maciá-Fernández, J. E. Díaz-Verdejo y P. García-Teodoro por primera vez en [MF06] y desde entonces han dedicado numerosos artículos a estudiar este tipo de ataques. Al igual que *Shrew* explota una vulnerabilidad, la cual en vez de localizarse en el protocolo TCP se encuentra a nivel de aplicación.

El objetivo del ataque LoRDAS es impedir que las peticiones de los usuarios legítimos reciban servicio. Para lograr esto el atacante ocupa completamente las colas de acceso al servicio y vigila los momentos en los que se termina de dar servicio a una solicitud con el fin de recuperar esa posición lo más rápidamente posible.

Las primeras versiones de este ataque tenían como objetivo servidores iterativos, pero las últimas versiones han sido modificadas para también ser capaces de afectar a servidores concurrentes, [MF08a], los cuales pueden llevar a cabo el procesamiento en paralelo de peticiones. Dado que la mayoría de los servidores que se encuentran en Internet son servidores concurrentes, la existencia de un ataque confeccionado para adaptarse a ellos supone un gran riesgo.

Los servidores concurrentes son capaces de procesar varias peticiones en paralelo gracias al uso de varias máquinas o CPUs (concurrencia real) o con una sola CPU (concurrencia virtual). En el caso de la concurrencia virtual se puede o bien generar un proceso padre que produzca varios procesos hijos o se puede tener un solo proceso con varios hilos. En un instante especifico

solo una petición está siendo procesada, ya que cada hilo o proceso tiene unos tiempos asignados en los que puede hacer uso de los recursos para tratar sus respectivas peticiones. Esto también nos permite contemplar el caso de los servidores iterativos como un caso de concurrencia en el que solo existe un elemento de procesado.

La estrategia que emplea LoRDAS para causar la indisponibilidad de la víctima es llevar a esta a un estado en el que todas las colas del sistema se encuentren llenas, es decir, que este se encuentre saturado. Por ello uno de los factores más importantes a tener en cuenta es el tiempo que se tarda en dar servicio a cada petición.

El tiempo de servicio, T_s, se puede considerar una variable aleatoria dependiente de diversos parámetros, entre los cuales el tamaño de las peticiones es la que mayor variación puede provocar. Por ello se supondrá que el tamaño de las peticiones será constante, de forma que el tiempo de servicio sea determinista y fijo.

Dado que el resto de variables también pueden provocar ligeras desviaciones en el tiempo de servicio, usando el teorema central del límite podemos aproximar la distribución de este tiempo a una variable normal de fórmula:

$$f(T_s) = N(\bar{T_s}, var(T_s)) \tag{3}$$

Para que el ataque tenga la mayor efectividad posible, el atacante tratará de hacerse con todas las posiciones de las colas para llevar al servidor al estado de saturación, al mismo tiempo que compite por ser el primero en ocupar las posiciones que se irán abriendo cuando se emita una respuesta a las peticiones de la cola. Por ello el periodo de ataque oscilará alrededor del tiempo de servicio.

El ataque de G. Maciá-Fernández et al. se define como un ataque periódico que alterna entre una fase inactiva y una fase activa, en la que envía las peticiones y que se define por medio de los siguientes parámetros:

- *Intervalo (Δ):* este será el tiempo entre la emisión de dos peticiones en el intervalo activo.
- *Fase activa (t_{ontime}):* es el momento en el que se trata de capturar las posiciones liberadas en las colas de servicio mediante la transmisión

de paquetes a una tasa $1/\Delta$, por ello la duración será de $t_{ontime} = \Delta(n_r - 1)$, donde n_r es el número de peticiones enviadas.

- *Fase inactiva ($t_{offtime}$):* es el intervalo entre fases activas. En este tiempo no se emiten mensajes de ataque.

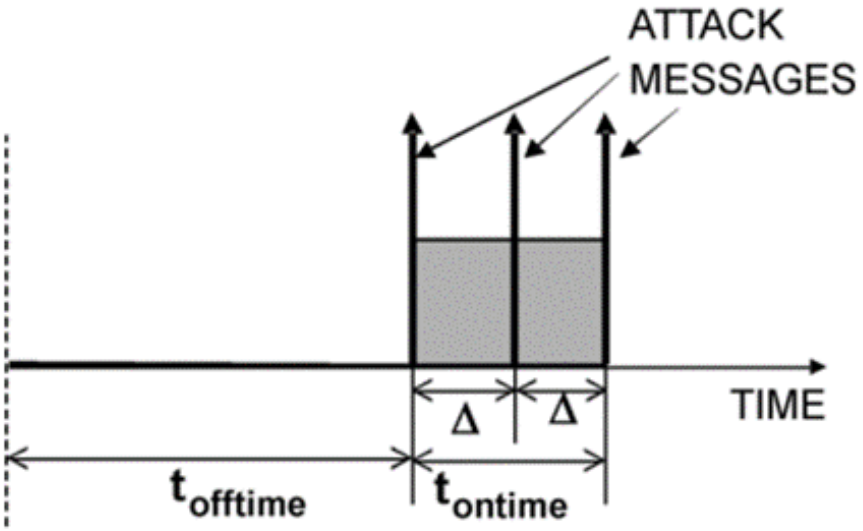

Figura 9: Diagrama del ataque LoRDAS, forma y parámetros [MF09].

El contenido de los mensajes de ataque no necesita ser elaborado a medida, como en otros ataques a vulnerabilidades, y por ello estos pueden ser idénticos a los mensajes legítimos. Este hecho hace que el ataque pueda ser considerado una denegación de servicio "parcial", ya que el sistema víctima está activo procesando peticiones pero, al mismo tiempo, los usuarios no pueden obtener respuesta.

[Xu10] propone una nueva versión de este ataque en la que introduce un comportamiento cambiante a lo largo del ataque, el cual divide en 2 etapas distintas, arrebatamiento y acaparamiento, y dos estados diferentes, captura y vigilancia, para cada uno de los hilos que actúan en esta nueva estrategia. Las etapas son las distintas fases del ataque.

Los estados podrían ser vistos como la forma en que los distintos hilos que participan en el ataque se comportan. El primer estado es el estado de captura, en el que los hilos del ataque buscan conseguir una posición de la cola de la víctima; el segundo es el estado de vigilancia, en el cual los hilos que ya hayan capturado una posición emiten una serie de peticiones de acuerdo con el periodo del ataque, a fin de mantener dicha posición.

Durante la etapa de arrebatamiento todos los hilos se encuentran en el estado de captura hasta que las colas se llenan, es decir, hasta el momento en que se comienzan a rechazar peticiones. En ese instante se pasa a la etapa de acaparamiento, en la que existen tanto hilos que están buscando capturar posibles posiciones que estén bajo control de usuarios legítimos (estado de captura) como hilos que se dedican a mantener ocupadas las posiciones que se encuentran en manos del atacante (estado de vigilancia).

2.3.5.3 Reducción de Calidad

La Calidad de Servicio (*Quality of Service*, QoS) es una medida del rendimiento percibido por los usuarios, es decir, entre los extremos de las conexiones que se realizan en Internet o en las redes telefónicas. En el caso de Internet esta medida tiene en cuenta parámetros de las redes como son la tasa de error, el ancho de banda o el rendimiento o los retardos entre muchos otros.

Los ataques de Reducción de Calidad (RoQ) pueden considerarse ataques de denegación de servicio "parciales" ya que no causan una inaccesibilidad a los servicios ofrecidos por los servidores objetivo, sino que buscan sobrecargar los *routers* de la frontera de tal forma que las peticiones legítimas se vean descartadas, cosa que daña la Calidad de Servicio de las conexiones. Los ataques *Shrew* podrían considerarse un tipo de ataque RoQ puesto que dejan pasar algunos mensajes mientras buscan dañar el rendimiento de las conexiones.

Los ataques de reducción usan pulsos periódicos de corta duración pero de gran tasa binaria, a fin de llevar a los sistemas a estados en los que se ven saturados, lo que hace que los mecanismos de control de congestión se vean desactivados, operando de forma ineficiente hasta que logre recuperarse, momento en el que volverá a ser golpeado por el ataque.

En el caso del protocolo TCP el sistema que controla la congestión es AIMD (*Additive Increase/Multiplicative Decrease*). Este mecanismo tiene el poder de adaptar la ventana de congestión para sacar el máximo partido a la capacidad útil de los enlaces hasta que ocurre una pérdida.

Este tipo de ataques pueden lanzarse explotando vulnerabilidades de diferentes elementos de las redes, como balanceadores de cargas [Gui07] o sistemas finales [Gui05, Sri11], para lograr grandes daños.

Entre los trabajos de autores que dedican sus esfuerzos a esta área de investigación es común encontrar que se presta especial atención a la potencia del ataque, π, la cual se obtiene por medio de la siguiente formula

$$\pi = \frac{D}{C^{\frac{1}{\Omega}}} \tag{4}$$

donde se puede apreciar la relación con el daño causado, D, y el coste, C, de lanzar el ataque.

La tercera variable de la que depende la potencia sería la agresividad del atacante, Ω, que indica a cuál de los otros dos factores se le da más relevancia, puesto que un Ω elevado será signo de un deseo de causar el mayor daño posible a cualquier coste.

De acuerdo a esto último, los ataques DoS pueden considerarse ataques que reducen la calidad de las comunicaciones hasta el mínimo al impedir todo el acceso, es decir, que infligen el máximo daño posible, lo que hace que tengan un coste superior al que presentan los ataque RoQ, cosa que se demuestra en [Gui04].

3 Métodos de Análisis Existentes

En la actualidad la ciberdelincuencia y la ciberseguridad son dos grandes fuentes de ingresos, tal y como se menciona en [Mor23], por ello constantemente están apareciendo nuevas amenazas que llevan a grandes inversiones, con el fin de producir nuevos sistemas de detección y respuesta frente a estas. Dado que cada tipo de ataque tiene unas características propias, los mecanismos de detección suelen usarlas para identificar la amenaza y posteriormente tomar las medidas que se consideren más apropiadas en cada caso.

Por eso cuando se está estudiando un tipo de ataque se debe prestar especial atención a las contramedidas que se están desarrollando frente a este, por eso en este capítulo se va a proceder a hacer un análisis de parte de la literatura que se ha elaborado alrededor de los sistemas de detección de ataques de denegación de baja tasa.

Como F. Simmross Wattenberg explica en el capítulo 2 de su tesis doctoral, [SW09], los sistemas de detección están formados de tres fases además de la etapa de validación. Para cada una de esas fases existe una gran variedad de técnicas diferentes que se pueden emplear.

En el caso de la primera etapa, en la que se capturan los datos, se puede elegir si se quiere redirigir todo el tráfico o si se prefiere solo recibir una serie de indicadores del tráfico. A partir de esto en la siguiente fase se extraen aquellas características que sean representativas del tráfico por medio de la teoría de la información, la teoría de la señal o la inteligencia artificial entre otros métodos, y en último lugar, estos datos se usan para determinar si el patrón observado es normal o anómalo pudiendo usar diversas técnicas.

En esta sección vamos a separar los sistemas de detección de forma más sencilla: no se va a atender a las tecnologías o metodologías empleadas, sino que se va a separar en aquellos que usan principalmente el dominio temporal, aquellos que usan el dominio frecuencial y aquellos que se basan en características a la hora de extraer parámetros con los cuales se pretende determinar si se está sufriendo un ataque de denegación de baja intensidad.

3.1 Dominio Frecuencial

El estudio por medio del dominio frecuencial hace uso de técnicas de procesado de señal a fin de encontrar los ataques LDDoS que se ocultan en el tráfico recibido por la víctima. Para ello los datos deben ser capturados en el dominio temporal y luego representados en función de la frecuencia por medio del uso de la transformada discreta de Fourier (DFT) [Bla06].

La señal transformada se distribuye en las distintas bandas de frecuencia en aquellos casos en los que el sistema no se encuentra bajo ataque, mientras que se concentra en bandas de menor frecuencia durante los ataques debido a la naturaleza periódica de estos.

He et al. [He09] propone un mecanismo de detección que hace uso del análisis wavelet (DSBWA). Este sistema, desarrollado a partir de las características periódicas y de pulsos cortos de LDoS, demuestra por medio de una serie de simulaciones en NS-2 [NS-] tener unas altas tasas de detección al mismo tiempo que ejerce un consumo bajo de los recursos computacionales.

El sistema parte de la suposición de que el proceso de llegada de los paquetes al *router* frontera puede definirse como un proceso estocástico. Apoyándose en otros estudios se establece que el tráfico normal es autosimilar y mantiene una correlación a largo plazo en una determinada escala de tiempo y que su proceso de llegadas es un proceso estacionario generalizado [IMV64], mientras los flujos de ataque LDoS son periódicos y afectan al rendimiento de la red, haciendo que su superposición con el tráfico legítimo no pueda mantenerse estable debido a que la mayoría de mensajes legítimos serán descartados.

Por medio de la observación del espectro representado en la figura 10, se puede apreciar que existe una gran diferenciación entre el flujo TCP y el ataque. Esto se debe a que las conexiones legítimas están regidas por los mecanismos de control de TCP, y esto hace que su periodicidad dependa del RTT, que cambia dependiendo de la carga de la red y provoca que los picos se encuentren dispersos entre distintas bandas de frecuencia. Al mismo tiempo el periodo del orden de segundos que presenta LDoS hace que su espectro se concentre en las bandas más bajas. Las conclusiones que se pueden extraer es que la distribución espectral del proceso en condiciones normales será dispersa y equilibrada al combinar todo el tráfico UDP y TCP, mientras que en el caso de un ataque el espectro se concentra en la banda de menor frecuencia al descartar la mayor parte de flujos legítimos.

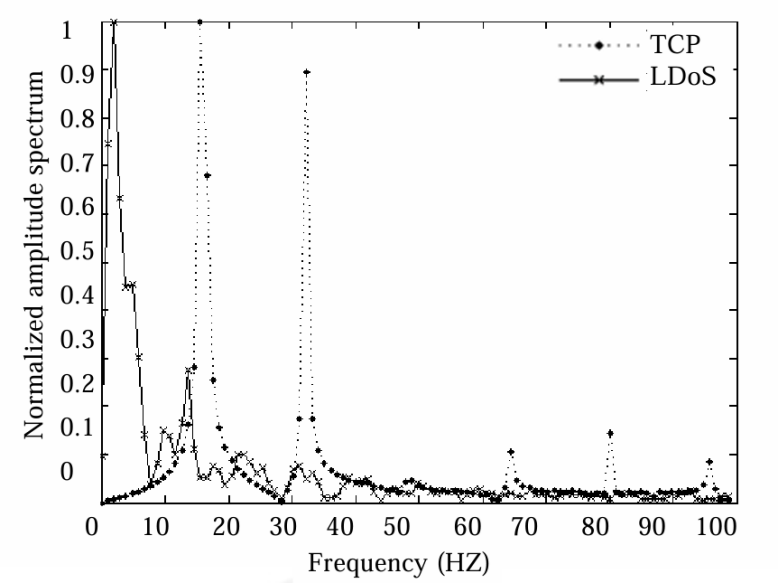

Figura 10: Diferencias entre el espectro de un flujo TCP y un flujo LDoS [He09].

Partiendo de esto los autores definen el proceso de llegada de tráfico como $X(n)$ y sus bandas $X_a(n)$ y $X_b(n)$ a partir de los cuales definen las características a extraer y que luego serán procesadas por una red neuronal entrenada por *backpropagation*:

1. El valor promedio de $|X_a(n)|$ en la banda del flujo de ataque.

2. La desviación estándar de $|X_a(n)|$ en esa misma banda.

3. La energía en la banda de ataque, la cual es la suma de la energía de cada una de las escalas cuya frecuencia central cae en la banda y las cuales se han representado empleando coeficientes wavelet.

4. El valor medio de la subbanda donde están los flujos de fondo, $X_b(n)$.

5. El factor de impulso, I, del tráfico de red $X(n)$ se emplea para medir la intensidad de las ráfagas en la red. En condiciones normales el tráfico es fluido y por ello I no es muy elevado. Dado a que en las redes pueden ocurrir ráfagas de alta intensidad de forma natural, la fórmula debe ser capaz de distinguir estos casos:

$$I = \frac{\text{máx}(X(n)}{\frac{1}{N}\sum_{n=1}^{N} X(n)} \tag{5}$$

Esto se puede lograr adaptando la fórmula 5 a $I = \frac{1/P \sum_{j=1}^{P} X^j(n)}{1/N \sum_{n=1}^{N} X(n)}$ donde X^j representa las P mediciones más grandes, en los casos donde P sea similar al número de impulsos LDoS en una ventana de observación, el impacto de las ráfagas no maliciosas disminuye, mientras que el de los ataques se mantiene inalterado.

Introduciendo estos 5 indicadores en la red neuronal se pretende detectar el LDoS y localizar el momento donde llega el pulso para rastrear la fuente. En aquellos casos en los que los atacantes trataran de emplear métodos por los que los flujos maliciosos provienen de diferentes orígenes, al converger en la víctima la implementación del sistema en un *router* fronterizo, seguirá dando buenos resultados.

Además, el algoritmo de *backpropagation* que se utiliza en el entrenamiento de redes neuronales basadas en el aprendizaje supervisado [Pro22], permite emplear la tasa de error de la propagación hacia delante de la red para alimentarla en el sentido contrario, a fin de ajustar los pesos dados a cada una de las conexiones, lo que permite aproximar lo máximo posible la salida real a la salida deseada.

Otro forma de detectar ataques sería la que Agrawal y Tapaswi [Agr18] proponen para entornos de computación en la nube. Esta propuesta hace uso de la densidad espectral de potencia para detectar y mitigar ataques LDDoS a partir de la monitorización de datos agregados de tráfico en tiempo real.

Los datos capturados se dividen en segmentos para analizarlos con mayor precisión. La longitud de estos debe ser lo suficientemente larga como para poder ver el patrón del ataque y lo suficientemente corta como para no distorsionar el patrón al fijar un periodo de captura demasiado largo. Después, se toman muestras equiespaciadas de cada uno de los segmentos usando una frecuencia calculada a partir del teorema de Nyquist [Sha49]:

$$f_{muestreo} \geq 2 f_{Nyquist} \tag{6}$$

A partir de estas muestras el sistema calcula la auto-correlación, que mide el parecido que existe en un proceso en 2 instantes temporales, y tras hacer esto se pasará al dominio frecuencial por medio del uso de la transformada discreta de Fourier, que es el equivalente de la transformada continua

para señales en las que no se tienen nada más que un conjunto de valores.

Una de las consecuencias de usar la DFT sobre la auto-correlación será que el resultado de esta operación dará la densidad espectral de potencia (PSD) de la señal. A partir de las variaciones de la potencia con respecto a la frecuencia se podrá determinar en qué casos ocurre un ataque. Al igual que establecía [He09], en aquellos casos en los que la distribución sea uniforme se asume que el tráfico es normal, mientras que si la energía se concentra en las bandas de baja frecuencia se puede indicar la existencia de un ataque LDDoS.

La eficiencia del trabajo de Agrawal et al. fue probada en un entorno real en el que el atacante lanzaba el ataque por medio de un servidor *C&C* que tenía control sobre 100 nodos comprometidos, tanto internos como externos, con los que llevaba a cabo diversas estrategias de ataques de baja intensidad como Slowloris, Slowpost, etc. Todas estas herramientas tienen en común que su objetivo es el protocolo Hypertext Transfer Protocol (HTTP) de la capa de aplicación del servidor.

N. Agrawal y S. Tapaswi no son los únicos autores que proponen el uso de la PSD para la detección de ataques. De hecho este es uno de los métodos más ampliamente estudiados e incluso existen algunos artículos que se centran en la adaptación de este método contra determinadas estrategias de ataque. Un ejemplo sería el artículo [Bry15], que se centra en la detección de los ataques LoRDAS contra la versión 2.2 del Servidor de HTTP Apache.

En [Che06], el mecanismo presentado para la detección de *Shrew* emplea la densidad de potencia como parte de un sistema colaborativo con capacidades de filtrado. Por medio de la comparación de la PSD con plantillas de características de ataque, cada uno de los nodos de la red es capaz de detectar los intentos de ataque. Cada nodo actúa como una raíz para un "árbol de detección" que contiene a todos los *routers* en una distancia de dos saltos. En estos árboles se propagarán los avisos de ataque para que solo los nodos afectados desplieguen las contramedidas con las que cortan las conexiones maliciosas.

Los métodos que emplean el dominio de la frecuencia parecen ser muy efectivos a la hora de identificar los ataques de baja tasa, esto se debe a que las condiciones de estudio bajo las que se realizan los experimentos de los artículos anteriores son muy limitadas y no incluyen grandes desviaciones con respecto a los modelos teóricos de ataque. Además, estos mecanismos de detección basados en frecuencia también introducen una mayor carga compu-

tacional si los comparamos con los métodos que trabajan en el dominio del tiempo, esto se debe a tener que convertir la señal temporal a frecuencial, lo que en algunos casos puede resultar en una alta tasa de falsos positivos.

3.2 Dominio Temporal

El análisis de ataques LDDoS en el dominio temporal elimina gran parte de la complejidad que presentan las aproximaciones desarrolladas sobre el análisis frecuencial, lo que resulta en la reducción de los tiempos de detección.

Entre los métodos que se pueden usar para llevar a cabo estudios en el dominio temporal uno de los más frecuentes es emplear la entropía de la información, también conocida como entropía de Shannon [Sha49]. Esta medida es una forma de cuantificar la incertidumbre asociada a la ocurrencia de un evento. Otra forma de definir este cuantificador estadístico es la usada en [TG22], donde se establece que la entropía de Shannon es la tasa media a la que se genera información en una fuente de datos estocástica.

La entropía de una señal $X(t)$ puede calcularse por medio de la siguiente fórmula:

$$S(X) = -\sum_{i=1}^{N} p(x_i) \log_2(p(x_i)) \tag{7}$$

donde $p(x_i)$ es la probabilidad de obtener el valor X_i.

H. Kumawat y G. Meena [Kum14] proponen un mecanismo con el que, comparando el valor de la entropía espectral con el valor umbral de detección, logran diferenciar los flujos maliciosos de los legítimos. Esta comparativa también servirá para diferenciar ataques LDoS de ataques DoS de inundación, a los que los autores se refieren como High-rate DoS (HDoS), puesto que la entropía de estos siempre será mayor a la de un flujo normal, que resulta ser mayor que la de un flujo LDoS.

Zhang et al. [Zha10] desarrolla un esquema de detección basado en entropía avanzada (AEB). Este sistema, al igual que el anterior, es capaz de distinguir los dos tipos de ataques DoS, pero además incluye unas adaptaciones que le permiten distinguir las oleadas masivas de peticiones legítimas que pueden ocurrir debido a la publicación de noticias u ofertas y también el tráfico normal en aquellos casos de LDoS que imitan la forma de estos.

Para lograr esto el sistema divide los valores de la entropía en varias categorías de gravedad, cada una de las cuales tiene unos estudios adicionales para determinar de forma precisa si está sucediendo un ataque antes de emitir una alarma.

Otra forma de emplear la entropía en este tipo de detección es la estudiada por Xiang et al. [Xia11]. En este artículo se proponen 2 métodos diferentes: el primero de estos emplea la entropía generalizada de orden α [Dup13], que corresponde a la fórmula:

$$H(X) = \frac{1}{1-\alpha} \log_2(\sum_{i=1}^{N} p(x_i)^\alpha) \tag{8}$$

Cuando se comparan los resultados de esta función con los obtenidos por la entropía de Shannon, fórmula 7, podemos apreciar que aumenta la diferencia entre las distribuciones de probabilidad [Bar92], y por ello podemos obtener mejores resultados ajustando el orden.

El segundo método presentado en este artículo es el que emplea distancias de información, es una medida de la diferencia existente entre 2 densidades de probabilidad. Numerosos autores usan la divergencia de Kullback–Leibler [Joy11] para la detección de ataques pero, Xiang et al. proponen un método alternativo usando una divergencia generalizada basada en un parámetro α.

Un estudio similar al de Xiang et al. [Xia11] fue realizado por Sahoo et al. [Sah18], en el marco de un centro de datos construido por medio de redes definidas por software (*Software Defined Network*, SDN).

Wu et al. [Wu18] describe una forma de detectar ataques por medio del uso de la transformada de Hilbert-Huang y la correlación de Pearson. La transformada de Hilbert-Huang [Hua98] es un método de análisis tiempo-frecuencia para señales no estacionarias y no lineales con el que se pueden ver las características de oscilación, mientras la correlación de Pearson [Kir08] es una expresión de la correlación lineal entre dos conjuntos de datos. La combinación de estas técnicas permite el análisis de ventanas de una longitud de un par de decenas de segundos.

La última metodología de análisis que se va a describir en esta sección es la presentada en [Wu19], que emplea la alineación de secuencia para identificar el patrón de ataque. Existen 2 formas en las que se pueden hacer estas alineaciones: la primera de ellas se denomina el alineamiento global. Este tipo

deberá ser usado en aquellos casos en que se sospeche que las dos secuencias a comparar tienen una longitud similar y un alto grado de parecido. La segunda forma, la local, busca patrones similares dentro de un fragmento de la secuencia. Esta será la mas apropiada cuando queremos analizar el trafico recibido, puesto que los retardos de las redes de comunicación y el envío de mensajes fuera de la ráfagas de ataque que podrían ser usados para camuflar el flujo maligno hacen que no podamos suponer la similitud de los datos a analizar.

3.3 Características de tráfico

Esta categoría de métodos de análisis pone su atención en la información contenida en los distintos campos que forman los paquetes recibidos por la víctima o en las métricas asociadas al funcionamiento y la evolución de las redes a lo largo de las conexiones. Prestando atención a estos parámetros se pretende encontrar pruebas que ayuden a determinar si los equipos se encuentran bajo ataque.

En los últimos años diversos autores han llegado a la conclusión de que el uso de parámetros individuales es un método de detección ineficiente, ya que el uso de una sola característica puede resultar en la emisión de un gran número de falsas alarmas. Por ello diversos autores extraen varios parámetros, los cuales pueden combinar en una nueva métrica o tratar de forma individual para luego agregar la salida de todos ellos para determinar si está ocurriendo un ataque.

Liu et al. [Liu20] extrae múltiples características, las cuales usa como entradas independientes para un clasificador que emplea el método de *K-Nearest Neighbor* (KNN). Una vez obtenidas las salidas del clasificador, estas se hacen pasar por una matriz que las fusiona en un único coeficiente, el cual será comparado con el umbral.

Las características empleadas en este artículo son la longitud de la cola, el numero de bytes confirmados por cada ACK y el tamaño de los paquetes. Estos parámetros reflejan los cambios que los ataques inducen sobre la víctima. En el caso de la cola, se puede observar que el ataque produce variaciones que son difíciles de apreciar en la media y la medida instantánea. Pero por medio del cálculo de la distancia Euclídea, es fácil observar que cuando se está sufriendo un ataque, el valor de la distancia aumenta.

Atendiendo al tamaño de los paquetes, por medio de la toma de muestras a lo largo del periodo de estudio, se puede comprobar que al sufrir un ataque la longitud de los paquetes se reducirá al perderse la eficiencia en el enlace hasta llegar a 0. Como consecuencia de esto el número de bytes confirmados por los ACKs también pasará a reducirse.

Por otro lado [Zha12] crea una nueva variable, la tasa de participación en la congestión o *congestion participation rate* (CPR). Para obtener esta métrica se toman medidas del número de paquetes de cada flujo que llegan por el enlace del *router* y cuando un paquete es desechado por la cola se asume que se está en un estado de congestión y se mide el número de paquetes del flujo que llegan durante ese estado. Debido a que los flujos TCP legítimos tratan de evitar las congestiones, la tasa de un flujo de ataque será mayor a la de los flujos de los clientes.

Wang et al. [Wan12], al igual que Zhang et al., crean nuevas métricas con las que llevar a cabo la detección. Estas serán *Network Traffic State* (NTS) y *Joint Deviation Rate* (JDR). NTS refleja el estado de la red en el punto de monitorización y es desarrollada a partir de la combinación de diferentes rasgos de la red. Por otro lado JDR describe la tasa de variación de NTS debido a que es la suma de las desviaciones de todos los rasgos que se combinan en la métrica NTS. La combinación de estas variables da lugar a un sistema con una alta tasa de aciertos y que requiere de unos registros de corta duración.

Para la generación del parámetro NTS se deberán agregar las siguientes características del tráfico:

- *Volumen de tráfico:* Midiendo el volumen de llegada a la víctima del ataque se pueden detectar fácilmente los ataques DDoS de inundación además de las ráfagas que pueden estar asociadas a los ataques LDDoS.

- *IP de origen:* En las grandes botnets se puede apreciar que existe una gran variedad de direcciones. La aparición de un gran numero de direcciones nuevas y la distribución de estas puede dar una pista de que el sistema se encuentra bajo amenaza.

- *IP de destino:* La distribución de las direcciones IP en la red de la víctima pasará a concentrarse alrededor de la dirección de una máquina determinada.

- *Puertos de origen y destino:* Algunas estrategias se centran alrededor de protocolos que tienen un puerto preestablecido, como HTTP, localizado

en el puerto 80. En condiciones de trafico normal la distribución de las conexiones a puertos tiende a no tener un foco, pero bajo ataque la mayoría del tráfico puede tener un mismo destino u origen.

- *Protocolo y longitud del paquete:* dependiendo del tipo de estrategia la longitud del paquete y el protocolo que se ha usado pueden variar en gran medida.

[Wu16] basa su análisis en el estudio [Liu04], que demuestra que a corto plazo el tráfico de red presenta características multifractales cuando se está operando en escalas temporales pequeñas. Estas características son observables en la mayoría de los servicios que se ofrecen por Internet, pero en el caso de los ataques LDoS el hecho de que algunas estrategias empleen UDP, que no tiene características fractales, hace que no pueda camuflarse frente a este tipo de estudio.

Por medio del análisis usando wavelets se calcula el índice de Hölder, el cual representa el peso de las ráfagas en el tráfico de la red en un instante determinado. Durante un ataque este índice se reduce en gran medida, debido a los cambios que se producen en las características multifractales, y por medio de la comparación con un umbral fijado a partir del análisis estadístico de la red se podrá determinar la existencia de un ataque.

Chistokhodova y Sidorov [Chi18] proponen un método de detección para ataques LDDoS contra HTTP. Para ello prestan atención a los datos transmitidos a nivel de aplicación y la secuencia de los paquetes entrantes. Estos datos le son alimentados como un vector de características a un mapa autoorganizado o red de Kohonen. Este método de estudio se basa en el aprendizaje no supervisado, por medio del que se reduce el tamaño del vector y esto se envía a un clasificador junto con los estados ocultos y las marcas temporales que se han calculado para cada uno de los clústers formados en la red de Kohonen.

Finalmente, Chen et al. [Che12], por medio del uso del teorema central del límite, aproxima los diferentes estados en los que se puede encontrar el tráfico a distribuciones normales con distinta media y desviación. Para el estado en el que la red está sufriendo un ataque LDoS usará la distribución $P \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$, donde $\hat{\mu}$ denota el valor medio y $\hat{\sigma}^2$ denota la varianza, ya que el trafico TCP experimentará una mayor fluctuación y por ello, para una media con el mismo valor entre los casos en los que la red no está sufriendo un LDoS y el caso en el que sí, podemos afirmar que la desviación típica será mayor en el caso del ataque.

Usando esta desviación se puede plantear un intervalo de confianza que nos permita concluir si la red se encuentra bajo ataque, ya que en los casos donde la red no está bajo la amenaza de un ataque de baja tasa existirá una mayor probabilidad de que el tráfico se localice en el intervalo. Debido a que el sistema presentaría una alta tasa de falsos positivos usando solamente el intervalo, también se debe cumplir que una cierta proporción de las medias obtenidas en un intervalo supere un umbral. Las medias serán calculadas asignando un peso mayor a las últimas observaciones, para que la reducción del tráfico causada por el ataque LDoS sea más significativa con respecto a otros ataques, al superar un valor específico.

4 Evaluación de Métodos de Detección de Ataques

Como se ha visto en la sección anterior, existe una gran variedad de técnicas por las cuales se pueden detectar los ataques de baja tasa, ya sean distribuidos o no. Pero como pretendemos demostrar en este trabajo, cada mecanismo tiene sus propias desventajas. Estas pueden ser explotadas por los atacantes de forma activa, mediante la creación de ataques con parámetros adaptados a las vulnerabilidades de los sistemas de detección, o de forma pasiva, en aquellos casos que pese a no adaptar el ataque al sistema de detección, el atacante logra su objetivo.

A continuación, se va a llevar a cabo la evaluación de tres mecanismos de detección diferentes, prestando especial atención a las suposiciones que estos hacen acerca de los entornos en los que se lleva a cabo su desarrollo, a fin de encontrar problemas que puedan tener en condiciones reales y que puedan ocasionar un *bypass* de las medidas de detección. El objetivo principal de este capítulo es comprobar si los sistemas de detección están preparados para hacer frente a las posibles variaciones del periodo del ataque.

4.1 Métodos Basados en Densidad Espectral de Potencia

Como ya se ha explicado con anterioridad, los ataques LDoS y LDDoS pueden ser expresados en función de 3 parámetros que dictaminan gran parte de las características del ataque. El periodo, T_n, es el parámetro de mayor relevancia para los métodos que emplean la densidad de potencia espectral, pues la expresión de este altera la forma en la que se distribuye la energía a lo largo del espectro.

Los métodos de detección basados en la densidad espectral de potencia (PSD), como los propuestos en [Agr18], [Bry15] y [Che06], centran su atención en la forma en que la energía de una señal se distribuye en función de la frecuencia. En los entornos de redes telemáticas esta energía depende de la forma en que se distribuyen los intervalos entre la llegada de paquetes a los distintos equipos y de los momentos en los que se concentra la llegada de tráfico.

La PSD tiene diversas aplicaciones en las redes telemáticas al permitir

estudiar el rendimiento de la redes, pudiendo así optimizarlas para aprovechar de mejor manera los recursos y mejorar la Calidad de Servicio (QoS). Pero además, también es una herramienta útil para el diagnóstico de problemas, al permitir ver los patrones de tráfico entre los distintos nodos, y la detección de anomalías, función en la que nos vamos a centrar.

Una característica de la PSD es que por medio de su uso se pueden apreciar los patrones periódicos e inusuales que presentan los ataques LDoS y LDDoS. Esta cualidad se manifestará como una serie de picos localizados a frecuencias específicas, tal y como se puede ver en la figura 10. Para ello será necesario emplear una resolución temporal y frecuencial que se adapte a la dinámica de la red y del ataque.

De acuerdo a los cálculos teóricos, los cuales se recogen en el Apéndice B, la efectividad de la detección basada en la PSD está muy relacionada con la frecuencia regular a la que se suceden los pulsos de los ataques.

Si nos fijamos en la fórmula de la densidad espectral de potencia de la señal de ataque, $S(\omega)$, es fácil averiguar a que se debe la efectividad de este método frente a la regularidad del periodo. En la fórmula que se muestra a continuación, podemos ver la estrecha relación con la transformada de Fourier de la señal de ataque, $X_N(\omega)$,

$$S_X(\omega) = \lim_{\tau_n \to \infty} E\left\{ \frac{1}{2\,\tau_N} \left| X_N(\omega) \right|^2 \right\} \tag{9}$$

en donde τ_N es el intervalo de tiempo considerado. En esta formula $|X_N(\omega)|^2$ es el módulo cuadrático de la transformada de Fourier, donde los sumatorios reflejan la interacción entre todos los pares de pulsos que forman el tráfico, la transformada de cada uno de los pulsos del ataque se representa como $P(\omega)$, siendo T_n y T_m los instantes de llegada de los pulsos.

$$|X_N(\omega)|^2 = |P(\omega)|^2 \sum_{n=1}^{N} \sum_{m=1}^{N} e^{j\omega(T_m - T_n)} \tag{10}$$

Lo que nos lleva a asegurar que para muchas de las herramientas de detección que emplean esta técnica, será necesario que el ataque tenga un periodo casi invariable, ya que en aquellos casos en los que el periodo de ataque se mantenga constante se podrá visualizar con certeza que gran parte de la energía se concentra en las bandas más bajas.

Por el contrario, si se emplea algún tipo de distribución, como la exponencial, que aleatoriza el tiempo entre ataques, la concentración dismi-

nuirá de forma que parecerá que la energía se mantiene dispersa, al igual que ocurre cuando se trata con tráfico normal, pasando así a ser indetectable.

Además de los problemas debidos a la aleatoriedad del periodo, también existe el problema de que al trabajar en el espectro frecuencial se debe llevar a cabo la transformada de Fourier, la cual consume una gran cantidad de recursos y tiempo, especialmente si se trabaja con ventanas de muy poca longitud. Esto se debe a que la complejidad de esta operación es $O(N \log N)$, donde N es el número de muestras que se consideran.

En resumen, los métodos de detección basados en la densidad espectral de potencia son herramientas potentes para el tratamiento de patrones regulares en el tráfico de red. Sin embargo, su efectividad se ve comprometida frente a ataques que utilizan intervalos de tiempos aleatorios, es decir, que simulan ser tráfico legítimo. Además, el coste de la transformada de Fourier limita su aplicación en tiempo real, especialmente en redes que experimenten altas cantidades de tráfico, ya que la detección del ataque se retrasaría, limitando de esta forma la capacidad de respuesta del sistema.

4.2 Métodos Basados en el Cálculo de la Entropía

La entropía es una forma de medir la incertidumbre o aleatoriedad de un conjunto de datos. En el contexto de la detección de ataques puede resultar una herramienta eficaz a la hora de detectar anomalías en el tráfico, al evaluar el grado de dispersión o la previsibilidad del tráfico. Los métodos derivados del cálculo de la entropía, como los que se han analizado en [TG22], [Kum14], [Zha10], [Xia11] y [Sah18], se basan en la idea de que el tráfico de red bajo ataque tiene características diferentes a las del tráfico normal.

La entropía de Shanon, cuya formula se muestra en la ecuación 7, es la más empleada en este tipo de estudios. Esto se debe a que x_i puede representar diferentes características del tráfico, como el número de paquetes recibido por unidad de tráfico o la distribución de las direcciones IP desde las que se envían los mensajes que entran a la red. Un cambio en estos patrones podría deberse a la presencia de un ataque, permitiendo detectar la anomalía.

Si se modela el tráfico que llega al punto de monitorización como el resultado de una superposición de procesos de Poisson, el valor de la entropía será elevado, debido a que las llegadas serán impredecibles y dispersas, y por ello dependiendo de la regularidad del ataque este podría pasar desapercibido.

Para los ataques LDoS y LDDoS con tiempos entre ataques fijos, el tráfico se volverá más predecible, ya que los paquetes que llegan al sistema lo hacen con una mayor regularidad. Esto provoca que la entropía disminuya con respecto al tráfico normal, que suele ser más aleatorio. Por lo tanto, este métodos será eficaz identificando patrones con un alto grado de regularidad.

Sin embargo, los ataques con un periodo aleatorio, en este caso modelado como una distribución exponencial, se comportan de forma más impredecible. Esto se debe a que la entropía podría ser muy similar a la obtenida de la red normal, ya que ambos presentan una distribución aleatoria para los tiempos de llegada. Esto hace que este tipo de ataques presenten una mayor dificultad a la hora de ser detectados, pudiendo pasar desapercibidos entre el tráfico legítimo.

Otro aspecto a tener en cuenta, es que si la duración del ataque es tan corta como para que no se produzca una variación lo suficientemente grande de la entropía, la sensibilidad de esta podría no ser suficiente para generar una alarma. Finalmente, el valor al que se fije el umbral utilizado para detectar anomalías tiene un gran impacto. Esto se debe a que si el valor es demasiado bajo se darán falsos positivos que pueden llevar a la generación innecesaria de alarmas.

Aunque los métodos basados en el cálculo de la entropía pueden resultar de gran utilidad para detectar cambios en las redes, su eficacia disminuirá en gran medida ante ataques cuyo modelo se aproxime al del tráfico legítimo. Para mejorar sus capacidades, es recomendable emplearlos junto a otra técnica de detección y ajustar los umbrales de alerta en función de las condiciones cambiantes de la red.

4.3 Métodos Basados en Alineación de Secuencias

El análisis de los distintos flujos de tráfico llevado a cabo en [Wu19] se realiza por medio de una técnica conocida como alineación de secuencia. Esta técnica, importada desde el campo de la bioinformática, busca encontrar patrones maliciosos repetitivos dentro de segmentos de las secuencias de tráfico.

El uso de segmentos en lugar de secuencias completas está muy relacionado con el hecho de que este método de detección está ideado para su empleo contra ataques síncronos, los cuales incrementan el periodo de ata-

que de forma exponencial a ritmo de $T_n = 2^n RTO$, donde n es el número del pulso. Además, a fin de reducir la carga computacional, únicamente se separa para su estudio el tráfico UDP, ya que es uno de los principales protocolos empleados por los atacantes que en lugar de explotar parámetros de las conexiones tratan de afectar a los enlaces, mediante el desbordamiento de estos.

Además de la sobrecarga computacional introducida al separar los mensajes según el protocolo y de identificar los diferentes flujos para llevar a cabo la comparación, también se puede encontrar el problema de que el atacante emplee diferentes longitudes de pulso o distintos patrones en cada uno de los pulsos emitidos por cada máquina, como las presentadas en la figura 11. De esta forma la capacidad de encontrar similitudes entre los distintos flujos se reducirá.

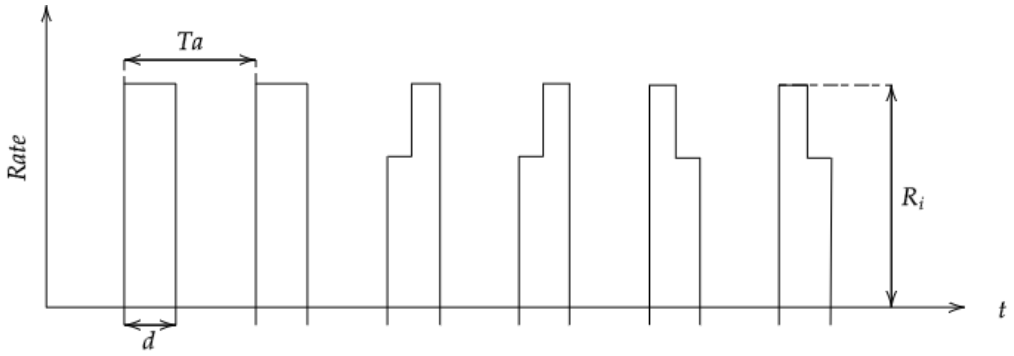

Figura 11: Diferentes patrones para ráfagas de LDoS.

Otras medidas que se pueden emplear para tratar de evitar los sistemas de detección es la introducción de ráfagas aleatorias de corta duración en los tiempos de inactividad. De esta manera también se reducen los parecidos entre los distintos flujos, o se puede tratar de forzar al mecanismo de detección a emitir alertas falsas que disminuirían la confianza de los equipos de seguridad en la herramienta.

A la hora de implementarse este tipo de sistemas debe tenerse en cuenta el desafío de la escalabilidad, puesto que a medida que la cantidad de dispositivos y el volumen de tráfico en la red aumentan, el proceso de alineación se vuelve más complejo. Lo que puede resultar en una disminución de la precisión y la velocidad de detección.

El análisis mediante la técnica de alineación de secuencias ofrece una herramienta con una gran capacidad a la hora de identificar patrones ma-

liciosos en ataques síncronos. Sin embargo, este enfoque conlleva muchas dificultades debido a la sobrecarga computacional y a la capacidad reducida para detectar variaciones en los patrones de ataque. Además, el empleo de ráfagas aleatorias y las variaciones en la forma de los pulsos hacen que los atacantes sean capaces de evadir la seguridad de estos sistemas.

5 Resultados

Este capítulo recoge los métodos que se han empleado a la hora de comprobar la verosimilitud de las hipótesis, las bases de datos que se han usado para ello, y los resultados obtenidos en las distintas pruebas. Estas incluyen tanto los de validación, con las que se pretende demostrar que el mecanismo desarrollado para la detección de ataques es funcional, como las que pretenden demostrar que un ataque con periodo dependiente de una distribución exponencial es imposible de detectar a diferencia de un ataque con periodo fijo.

5.1 Metodología

Como ya se ha visto en el capítulo 3, existen numerosas herramientas de detección, las cuales en algunos casos parecen presentar vulnerabilidades que disminuyen su eficacia o que las convierten en mecanismos completamente ineficaces a la hora de encontrar la presencia de atacantes en las redes.

En este caso, de entre las ideas que se presentan en la sección anterior nos vamos a centrar en la posibilidad de que los métodos basados en el cálculo de la densidad de potencia espectral son incapaces de detectar aquellos ataques LDoS y LDDoS que emplean periodos cuyo valor se modula como un proceso poissoniano.

Si bien los cálculos teóricos, los cuales se reflejan en el apéndice B, demuestran que esta idea es cierta, en el caso real podría no ser cierto, por ello se ha implementado un algoritmo de detección en Matlab con el que estudiar esta posibilidad.

En primera instancia se trató de estudiar la viabilidad del método a partir de la evolución en el número de bytes recibidos por un *router* a lo largo del tiempo. Para ello se empleó la base de datos que se explica en 5.2.1, a la cual se le sumaba un número de bytes generados por medio del código mostrado en el apéndice A.1.

A fin de unificar los datos de ataque generados con los datos de la tesis [SW09], el método realizaba una suma acumulativa del número de bytes para obtener unas muestras equiespaciadas con las que después se calculaba la diferencia de datos entre las distintas muestras. En el paso siguiente se generaban diferentes ventanas en las que se calculaba el cuadrado de la trans-

formada rápida de Fourier, que es una de las formas de obtener la densidad de potencia espectral, tal y como se muestra en la siguiente fórmula,

$$S_X(f) = |X(f)|^2 , \tag{11}$$

donde $X(f)$ es la transformada de Fourier de una secuencia $x(t)$.

Este algoritmo demostró no ser útil debido a que el muestreo de los datos en un intervalo de 5 segundos no aportaba una resolución adecuada. Además el método tenía una gran sobrecarga computacional, lo que hacía que tuviera un rendimiento demasiado bajo como para aplicarlo en una red actual.

El siguiente método que se desarrolló buscaba reducir los tiempos de cálculo del sistema. Por ello se buscó una base de datos con una mayor precisión temporal para evitar tener que llevar a cabo los pasos de suma, diferenciación y seccionado en ventanas, ya que eran los que agotaban un mayor número de recursos. La base de datos descrita en 5.2.2 cumplía estas condiciones.

La segunda técnica desarrollada, en vez de centrar su enfoque en la cantidad de bytes que llegan al router, se fija en los instantes en los que llegan los paquetes, y a partir de las marcas temporales genera una señal que refleja el número de mensajes en un instante determinado, tal y como se puede ver en la figura 12. La escala que se ha utilizado durante esta conversión es de una magnitud de milisegundos.

Esta señal, que expresa la concentración de mensajes, se emplea para calcular la densidad de potencia espectral por medio del método de Welch [Wel67]. Este método devuelve una estimación de la PSD al dividir los datos en segmentos superpuestos con los que computa un periodograma promediado. Esta técnica tiene la ventaja de que al calcular la media entre los distintos periodogramas se logra disminuir la varianza y las fugas espectrales. Las fugas espectrales pueden resultar en casos en los que se generan falsos positivos o negativos, debido a que es un fenómeno por el cual la energía de una frecuencia se desplaza a frecuencias cercanas, ya que se asume que la señal es periódica.

El código de esta herramienta se recoge y explica en detalle en el apéndice A.2. Este código grafica la salida del método de Welch expresando la PSD de forma lineal, es decir, que se expresará en Vatios(W)/Hercio(Hz), frente a la frecuencia que se expresa en Hz. Esta herramienta de detección se usa en la sección 5.3 para evaluar los métodos descritos en la literatura.

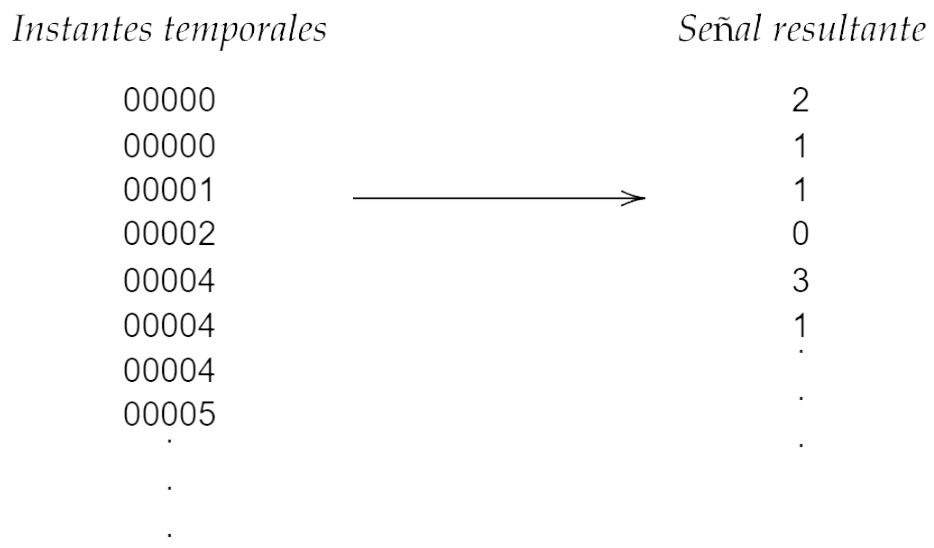

Figura 12: Conteo del número de mensajes recibidos en cada instante

5.2 Bases de Datos Empleadas

En este apartado se van a explicar los datos que forman cada una de las bases utilizadas a lo largo de este proyecto, y cómo se han manipulado para adecuarlos a su uso.

5.2.1 Datos Universidad de Valladolid

Esta colección de datos, que fue empleada en la realización de la tesis [SW09], está formada por dos bases de datos de tráfico real que se remontan a los años 2007 y 2008, y que fueron obtenidas a partir de enrutadores que formaban parte de la infraestructura de la Universidad de Valladolid. El primero de estos *routers* es un Cisco Catalyst 6509, que direccionaba todo el tráfico de las distintas sedes de la Universidad entre sí y con el resto de la red Iris.

Este *router* encamina un tráfico de entre 40 y 70 Mbps durante las horas laborales y se encuentra muy agregado debido a la gran cantidad de máquinas que se encuentran conectadas a las diferentes redes de la UVA. Por

otro lado, el segundo encaminador es el que se encuentra en la facultad de la Escuela Técnica Superior de Ingenieros de Telecomunicación (ETSIT). El modelo de este es un Cisco Catalyst 3550, ya que la tasa de tráfico que circula por él es mucho menor, 10 Mbps, y se encuentra menos agregado, aunque en algunos momentos experimenta picos de tráfico superiores.

Los datos se recolectaron por medio de contadores SNMP que devuelven la suma del número de bytes que han pasado por cada uno de los puertos, en el caso del *router* de la ETSIT, y de uno solo de los puertos del encaminador de la UVA en un intervalo de 5 segundos. Debido a que estos *routers* no podían sobrecargase, a fin de dar un servicio eficaz a la universidad, esta frecuencia de muestreo no se podía reducir.

Debido a que estos datos no presentaban ataques de baja intensidad, para este TFG se generaron por medio de Matlab unos datos simulados que siguen un patrón de pulsos intermitentes con un número de bytes asociados a cada pulso. Se produjo tanto un ataque con un periodo fijo como uno con un periodo dependiente de una distribución exponencial. Estos datos después se agregarán por medio de una función que realizaba la suma de todos los bytes en un determinado intervalo, en este caso de 5 segundos, a fin de mantener la base temporal inalterada.

5.2.2 Datos UTSA-2021-Low-rate-DoS-Attack

[Sys21] es una base de datos abierta que fue publicada junto al artículo [Ved21] por la Universidad de Texas en San Antonio. Esta universidad dedica gran parte de sus recursos a llevar a cabo diferentes investigaciones, entre las que se encuentran la búsqueda de innovaciones relativas al rendimiento, la seguridad y el desarrollo de redes y sistemas de gran velocidad.

La base de datos creada para probar los sistemas de detección basados en mecanismos de inteligencia artificial y presentados en el artículo anterior contiene distintas estrategias de ataque de baja intensidad junto con datos de tráfico legítimo generados en una red definida por software (SDN).

La SDN empleada en la generación de las capturas de tráfico consta de 4 máquinas Linux, de las cuales una actúa como un conmutador con un enlace de 1 Gbps; otra realiza las funciones del servidor web que será víctima de los ataques; las restantes estarán ejecutando procesos independientes que pueden interpretarse como los clientes legítimos y los atacantes. Las especificaciones técnicas de estas máquinas se detallan en el apartado 4.1 de [Ved21]

Máquina	**Sistema Operativo**	**Procesador**	**Memoria RAM**	**Tarjeta de Red**
Controlador SDN y Switch	Ubuntu 16.04 LTS	Intel i7-7700 3.69 GHz	32 GB	Ethernet Gigabit de 4 puertos
Servidor Web	Ubuntu 16.04 LTS	Intel Core2 Quad Q9550 2.83 GHz	8 GB síncronos	Ethernet Gigabit
Usuarios Legítimos	Ubuntu 16.04 LTS	Intel Core2 Quad Q9550 2.83 GHz	8 GB síncronos	Ethernet Gigabit
Atacantes	Ubuntu 16.04 LTS	Intel i7-2600 3.4 GHz	16 GB	Ethernet Gigabit

Tabla 3: Especificaciones de los ordenadores de la SDN.

o pueden ser comprobadas en la tabla 3.

El tráfico se recoge en las interfaces del switch que conectan el servidor con los clientes y los atacantes, pero en el análisis de los datos únicamente resultarán de importancia las tramas que llegan al servidor, es decir, el tráfico entrante, el cual contiene ataques del tipo SYN o del tipo *slow-read*. Esta primera estrategia sigue un patrón de pulsos con un periodo de 1 segundo y una duración de ráfaga de 0,1 segundos. Además la estrategia presenta tramos de 100 s donde se realizan ataques seguido de segmentos de la misma longitud donde solo existe tráfico de los clientes. Por otro lado la estrategia *slow-read* no sigue el patrón de pulsos.

Estos datos se usarán para validar el funcionamiento de la herramienta de detección, ya que se sabe que contienen un 30 % de datos de ataque, con estrategias que además son eficaces a la hora de llevar a cabo ataques contra infraestructuras reales.

5.2.3 Datos sintéticos

Estos datos los he generado por medio de un código de Matlab y constan de datos considerados legítimos y datos de ataque. La finalidad de esta base de datos es poner a prueba la teoría de que los ataques que no tienen un periodo fijo pueden ser pasados por alto al usar técnicas que emplean la densidad espectral de potencia. Para ello se han generado dos “capturas de

tráfico": una primera que contiene ataques con un periodo fijo y una que contiene un ataques modelados como un proceso de Poisson.

Antes de pasar a describir los datos de ataque se van a explicar los datos legítimos generados. Numerosos artículos han comprobado empíricamente que los procesos estocásticos, como es el casos de los procesos de Poisson, son útiles para modelar las llegadas a las colas, ya que desde el punto de vista de las redes los mensajes son generados de forma arbitraria e impredecible. Por ello los instantes de llegada de los paquetes se pueden considerar un proceso aleatorio y las llegadas de las diferentes sesiones son independientes entre si.

Para la creación de los datos, se han establecido una serie de parámetros que dictan la forma en que actúa la red. Estos son: la duración total que tendrá la simulación, el número de procesos clientes, que se correspondería con el numero de usuarios que acceden a los servicios, y la tasa a la que se recibirán los paquetes legítimos por segundo, este parámetro simula el tráfico de la red normal. La duración de la simulación es de gran importancia, puesto que de ser muy corto no se podrían visualizar correctamente los ataques. A partir de estos datos se puede simular el comportamiento aleatorio de una red real en la que existen varios clientes.

Por otra parte, para los ataques, se ha establecido un número de bots que participan en ellos, el número de ataques totales que se llevarán a cabo en la ventana de simulación y el periodo medio de los ataques, que en el caso del proceso poissoniano será la tasa media de generación de eventos y es el inverso de la media. Con estas variables se pueden producir los dos tipos de ataques que se quería estudiar:

1. *Ataques con Periodo Fijo:* el intervalo entre los pulsos de ataque en este caso es invariable, lo que produce ráfagas regulares en el tráfico. Esta característica se refleja en el espectro como una concentración en la energía alrededor de una frecuencia, lo que produce un pico de elevación.

2. *Ataques Modelados como una Distribución de Poisson:* para estos ataques se han generado tiempos de llegadas por medio de una distribución exponencial. Esto hace que no exista una concentración alrededor de ninguna frecuencia concreta, es decir, que no existe ningún pico que sobresalga entre los demás, lo que hace que no se puedan encontrar ataques.

La combinación de los datos legítimos con cada uno de los ataques

da lugar a dos colecciones de datos distintos, con los que podremos llevar a cabo las pruebas con las que se quiere demostrar que los ataques aleatorios no pueden ser predichos. Los valores de los parámetros que se han empleado a la hora de llevar a cabo los experimentos se recogen a continuación, mientras que el código de generación de los datos y la herramienta de detección descrita en la sección 5.1 se localizan juntos en el apartado A.2.

Parámetros del tráfico legítimo:

- Tasa de llegadas legítimas: $lambda\_legitimos = 5 paquetes/segundo$.
- Tiempo total de simulación: $T\_total = 100 segundos$.
- Número de clientes legítimos: $num\_clientes\_legitimos = 100$.

Parámetros del ataque LDDoS con periodo fijo:

- Número de máquinas que participan en el ataque: $num\_maquinas\_lddos1 = 500$.
- Número total de ataques durante la simulación: $num\_ataques\_lddos1 = 10000$.
- Periodo del ataque: $T\_ataque\_lddos1 = 1 segundo$.

Parámetros del ataque LDDoS con periodo fijo:

- Número de máquinas que participan en el ataque: $num\_maquinas\_lddos2 = 500$.
- Número total de ataques durante la simulación: $num\_ataques\_lddos2 = 10000$.
- Periodo del ataque: Distribución exponencial con una media de un segundo.

5.3 Resultados

En este apartado se presentan y analizan los resultados obtenidos de las simulaciones realizadas y de la validación del método de detección de ataques, la cual se ha llevado a cabo usando la base de datos presentada en 5.2.2. Por ello esta sección se va a dividir en dos partes principales: la primera se enfoca en poner a prueba el sistema de detección que se ha desarrollado,

y la segunda estudia los resultados de la aplicación de la herramienta sobre los datos simulados.

5.3.1 Validación del Método de Detección

Para evaluar la efectividad de la herramienta desarrollada se empleó una base de datos que contiene varios escenarios de ataque a distintas tasas y con el uso de diversas estrategias LDDoS. Esta validación es crucial a la hora de posteriormente llevar a cabo experimentos con los que se pretenda demostrar teorías, puesto que en caso de no cumplir el requisito de ser funcional los datos que se obtengan podrían no tener ningún tipo de validez.

Para llevar a cabo este proceso se han escogido 3 capturas de tráfico, las cuales se detallan a continuación:

- *client_normal_2_210111:* esta captura contiene tráfico benigno, el cual llega desde 8 máquinas diferentes .
- *Syn50:* esta captura de tráfico corresponde a un escenario donde existen tanto flujos benignos como malignos. En concreto existen 8 clientes diferentes y la estrategia de ataque que se emplea a un ritmo de 50 ataques por segundo es la inundación SYN.
- *slowread_1:* estos datos albergan el tráfico legitimo y de ataque que se genera de un escenario en el que están presentes 8 usuarios y una serie de ataques *Slow Read*.

En primer lugar, se va a analizar la captura que contiene únicamente datos benignos. A partir de todo lo que se ha estudiado llegamos a la conclusión de que el espectro de esta captura debe presentar una energía distribuida sin la existencia de un pico que sobresalga especialmente.

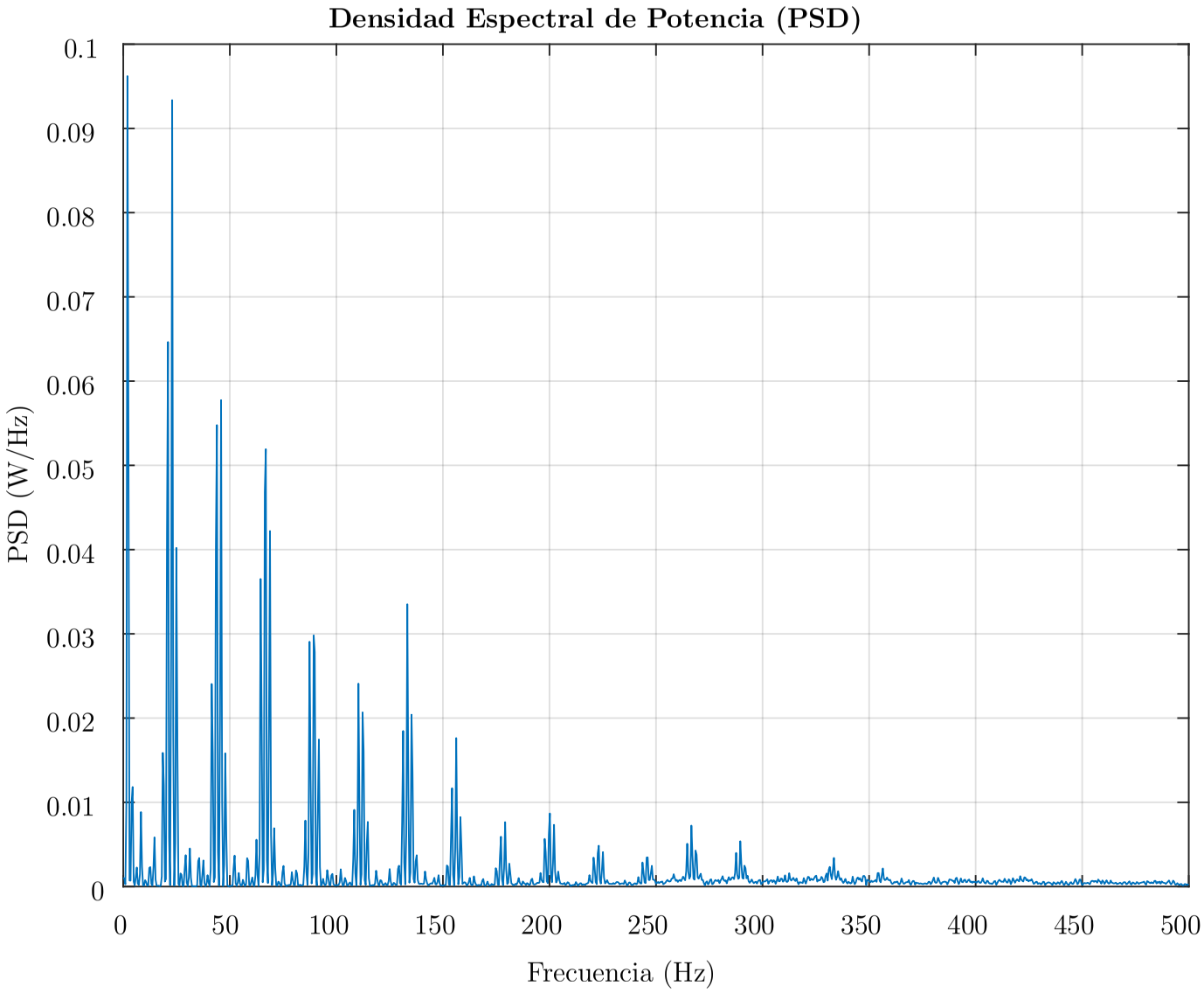

Figura 13: Densidad espectral de potencia del tráfico benigno.

Tal y como se puede ver en la figura 13, existen diversos picos de gran tamaño pero ninguno que supere con mucha diferencia a los demás, el único que podría cumplir esto es el que se sitúa en 0 Hz. La presencia de este pico puede deberse a diversos factores, uno de los más importantes es que la duración de los pulsos de ataque no permita asimilarlos como deltas de Dirac, tal y como se establece en el apéndice B, en ese caso la elevación que aparece en 0 Hz podría considerarse el resultado de la transformada de Fourier de los pulsos rectangulares, la cual da lugar a una sinc.

En este caso se relaciona con el hecho de que al existir una gran cantidad de paquetes, hay un tráfico que puede considerarse sostenido, es decir, que no hay momentos de pausa. Esto produce una señal cuya media es distinta de cero, lo que da lugar a una componente frecuencial de gran tamaño a esa frecuencia.

Por otro lado, al mirar los dos casos que presentan flujos dañinos (figuras 14 y 15) y que se están evaluando, se puede apreciar que existe, además del pico en 0 Hz, un pico de mayor magnitud que indica la presencia de un ataque LDDoS. El motivo de este depende de factores diferentes para cada una de las estrategias: en el caso de la inundación de mensajes TCP SYN el patrón de emisión de las ráfagas es el responsable de esa elevación en la PSD, mientras que en el caso del ataque *Slow Read* la causa de la elevación se debe a que se están emitiendo solicitudes HTTP lentas, lo que provoca pausas regulares en el flujo del tráfico, que es lo que se puede visualizar en el espectro.

En ambos casos también existen una serie de picos menores, los cuales pueden deberse a diferentes motivos, entre los cuales los más probables son que el switch tenga un mismo tiempo de servicio para las peticiones benignas de algunos clientes por lo que el tiempo entre una petición y la siguiente de la cola será idéntico. Otra posibilidad es que los mecanismos de control de flujo y control de congestión de TCP produzcan tiempos de pausa iguales al ajustar la ventana de recepción o al tener que retransmitir las peticiones que se hayan perdido por culpa de la congestión de la red.

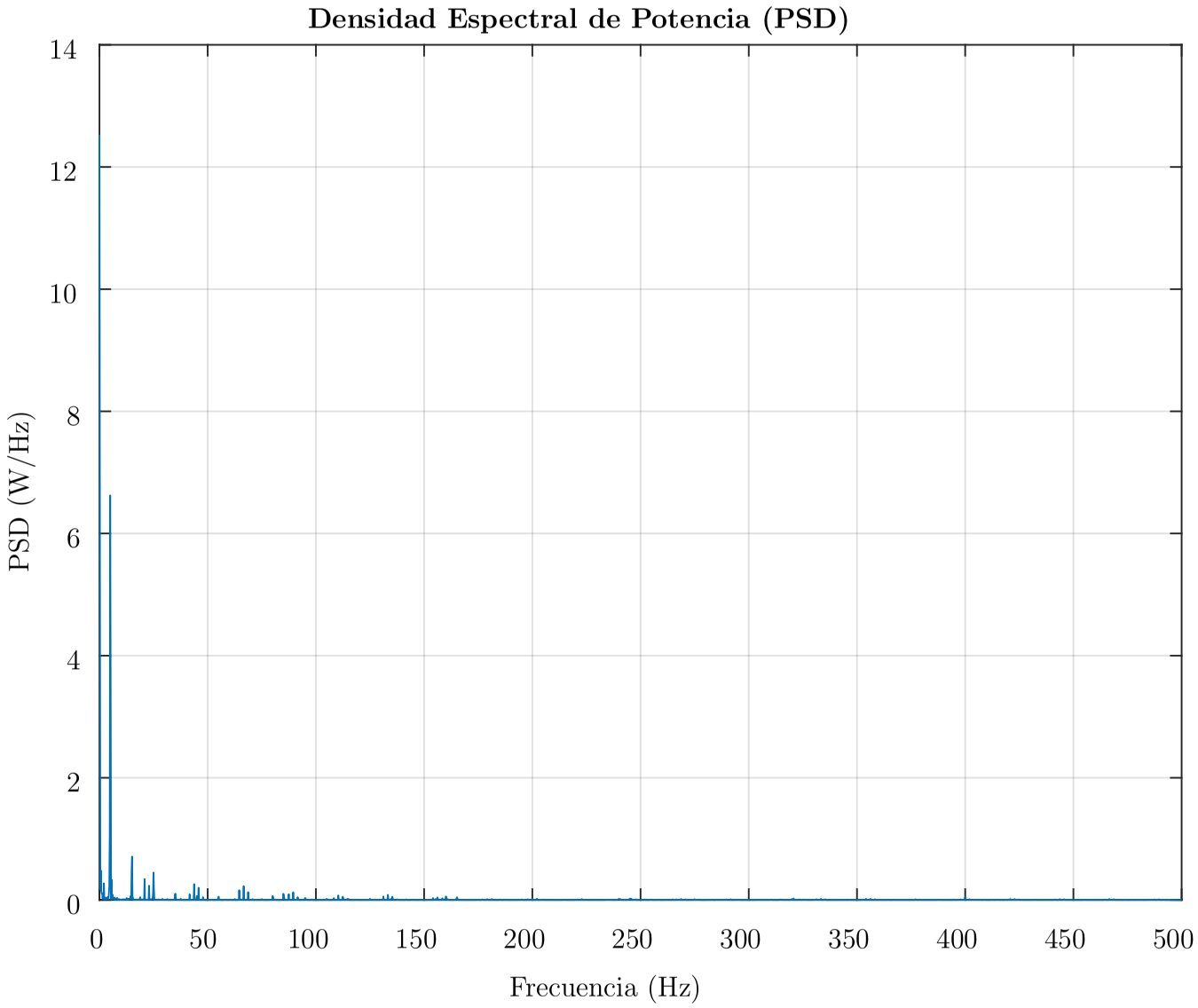

Figura 14: Densidad espectral de potencia del ataque *TCP SYN Flood*.

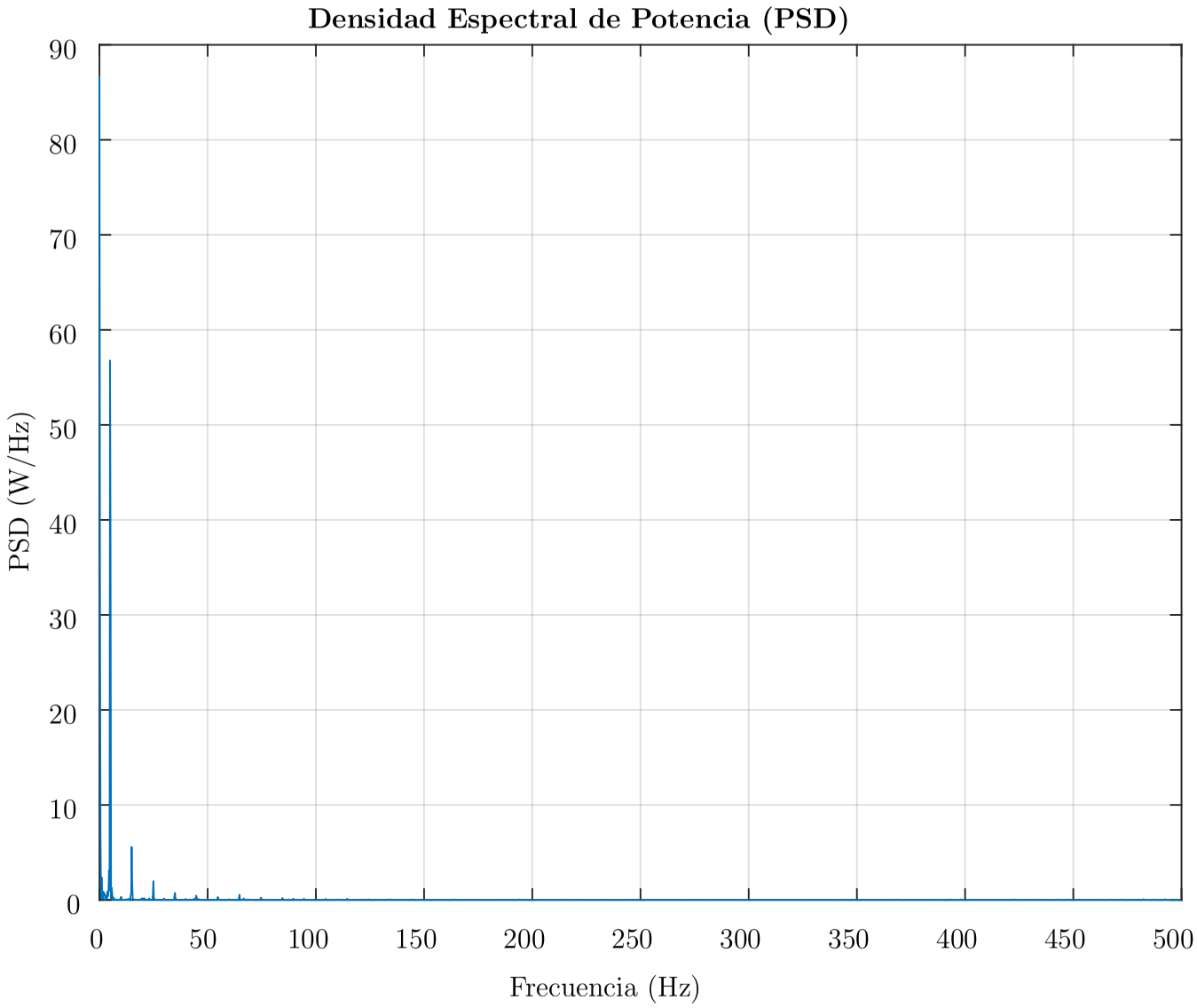

Figura 15: Densidad espectral de potencia del ataque *Slow Read*.

Tras ver que las gráficas de los ataques muestran que la detección se esta realizando de forma satisfactoria, se puede pasar a emplear este método para demostrar la teoría formulada con antelación.

5.3.2 Resultados de las Simulaciones de Tráfico

Con la intención de demostrar que los ataques con un periodo modelado según una distribución exponencial no se pueden detectar por medio del uso de técnicas que se fijan en la distribución de la energía, se han empleado los datos presentados en la sección 5.2.3.

Lo primero que se debe comprobar es si los periodos de los dos ataques que se están generando son correctos. Esto lo podemos hacer por medio de las figuras 16 y 17: al observar estas gráficas se puede ver que en la primera de ellas existen unas columnas equiespaciadas que tienen aproximadamente la misma altura, por lo que podemos deducir que se esta generando apropiadamente. En el caso de la segunda, las columnas presentan la misma altura pero las distancias entre ellas son diferentes, por lo que podemos apreciar que la distribución exponencial está en uso.

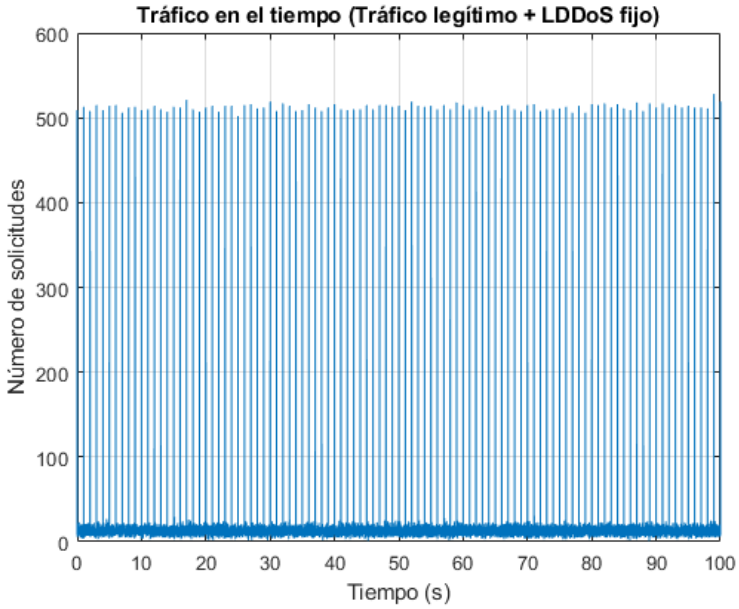

Figura 16: Número de solicitudes en el tiempo (ataque con periodo fijo).

El tráfico legítimo se puede visualizar como una línea de mayor densidad a lo largo del eje temporal de la gráfica: esto se debe a que el número

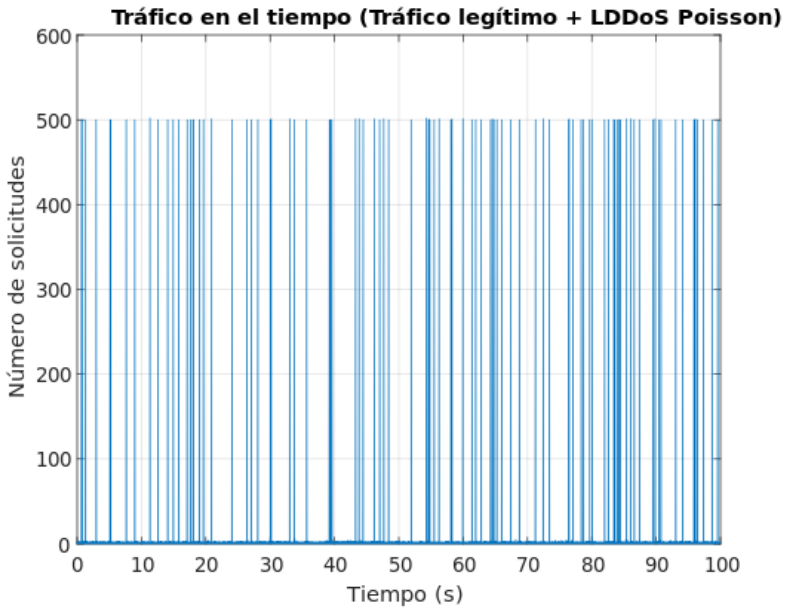

Figura 17: Número de solicitudes en el tiempo (ataque con periodo poissoniano).

de clientes legítimos es menor al de atacantes. Esto se debe a que al crear un modelo matemático extremadamente simplificado, en el cual no se reflejan los efectos de los mecanismos de control de congestión o de flujo de TCP, ni el bloqueo de las colas, se tuvo que dar un mayor peso a los efectos de los atacantes a fin de poder visualizar los efectos en el espectro sin tener que ampliar sobre la banda de menor frecuencia, como ocurría la figura 18, la cual se generó usando el mismo número de atacantes y de clientes.

Una forma alternativa de solventar este problema sería restar la media de la señal antes de calcular la PSD, ya que de esta forma se elimina la contribución de la media, tal como ocurre en la figura 19. Este método no se aplicó ya que en ciertos casos eliminaba algunos de los picos de energía.

Si se presta atención a las representaciones de los espectros, se puede ver que aunque en la figura 21 existen 400 atacantes más que usuarios, la energía se encuentra distribuida de forma uniforme sin existir ninguna concentración que destaque, es decir, que aunque las red desde la que se lanza el ataque cuenta con un número muy elevado de bots, que supera incluso al de los usuarios, si la distribución de este ataque es aleatoria no se podrá identificar y por ello no se podrá tomar ningún tipo de medida.

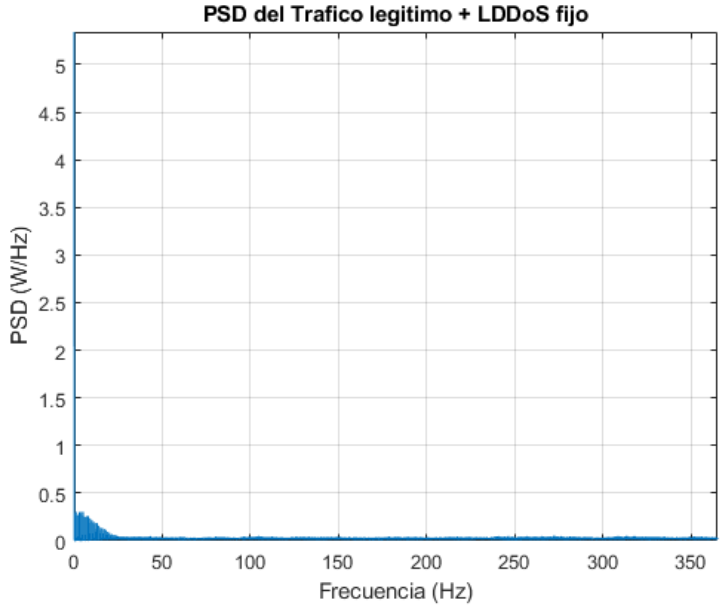

Figura 18: Densidad espectral de potencia del ataque de periodo fijo con el mismo número de atacantes y clientes.

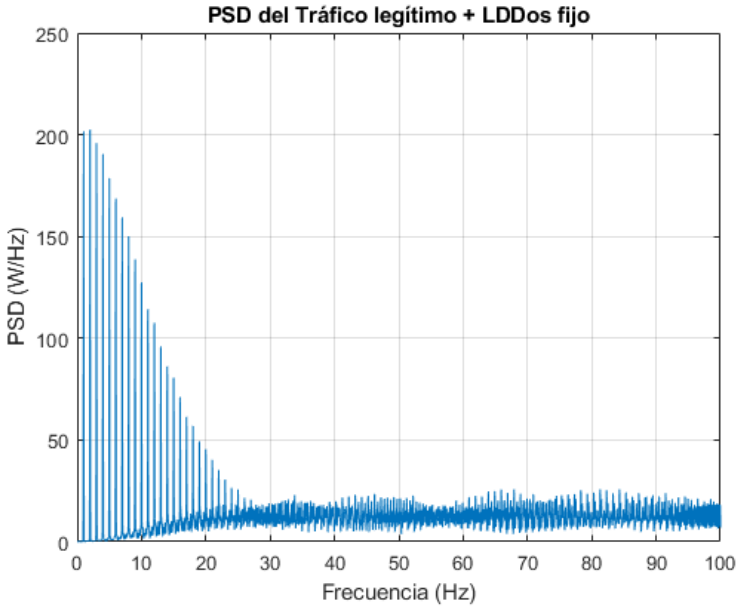

Figura 19: Densidad espectral de potencia del ataque de periodo fijo con el mismo número de atacantes y clientes sin las contribuciones de la media.

Por otro lado, la gráfica 20 presenta un lóbulo de gran tamaño en frecuencias bajas, al igual que pasaba en las figuras 18 y 19. Todas estas figuras son de casos en donde el periodo no varía, por lo que se puede afirmar que estos casos son detectables en situaciones donde existen un número de atacantes igual o superior al de clientes. En estas gráficas también se pueden ver otros picos que encajan con periodos de llegadas de datos legítimos que se repiten más.

Para finalizar, se ha querido comprobar hasta qué punto llegaba la capacidad de detección del sistema en los casos en que el periodo era fijo, y por medio de la repetición de la simulación se ha observado que para un número de 10 atacantes frente a 100 clientes es cuando comienza a costar diferenciar los picos debidos a los ataques. Por ello podemos afirmar que pese a que los mecanismos que usan la PSD son incapaces de detectar los ataques aleatorizados, son una herramienta de mucha utilidad frente a ataques de periodo invarible.

En resumen, los resultados obtenidos en este capítulo demuestran las limitaciones frente a ciertos tipos de ataques LDDoS con patrones aleatorios. Se ha logrado verificar que la PSD es una herramienta cuya eficacia disminuye al enfrentarse a ataques no periódicos, los cuales pueden camuflarse con mayor facilidad entre el tráfico legítimo de la red. Aunque también se evidencia la efectividad del método de detección basado en el cálculo de la densidad espectral de potencia para aquellos casos en los que los ataques presenten patrones altamente periódicos.

Estas pruebas resaltan la necesidad de desarrollar mecanismos de detección más robustos y diversos, que en algunos casos podrían tener que depender de la implementación de mecanismos híbridos que combinen varias técnicas de detección para que sean capaces de identificar los comportamientos maliciosos en situaciones de mayor complejidad.

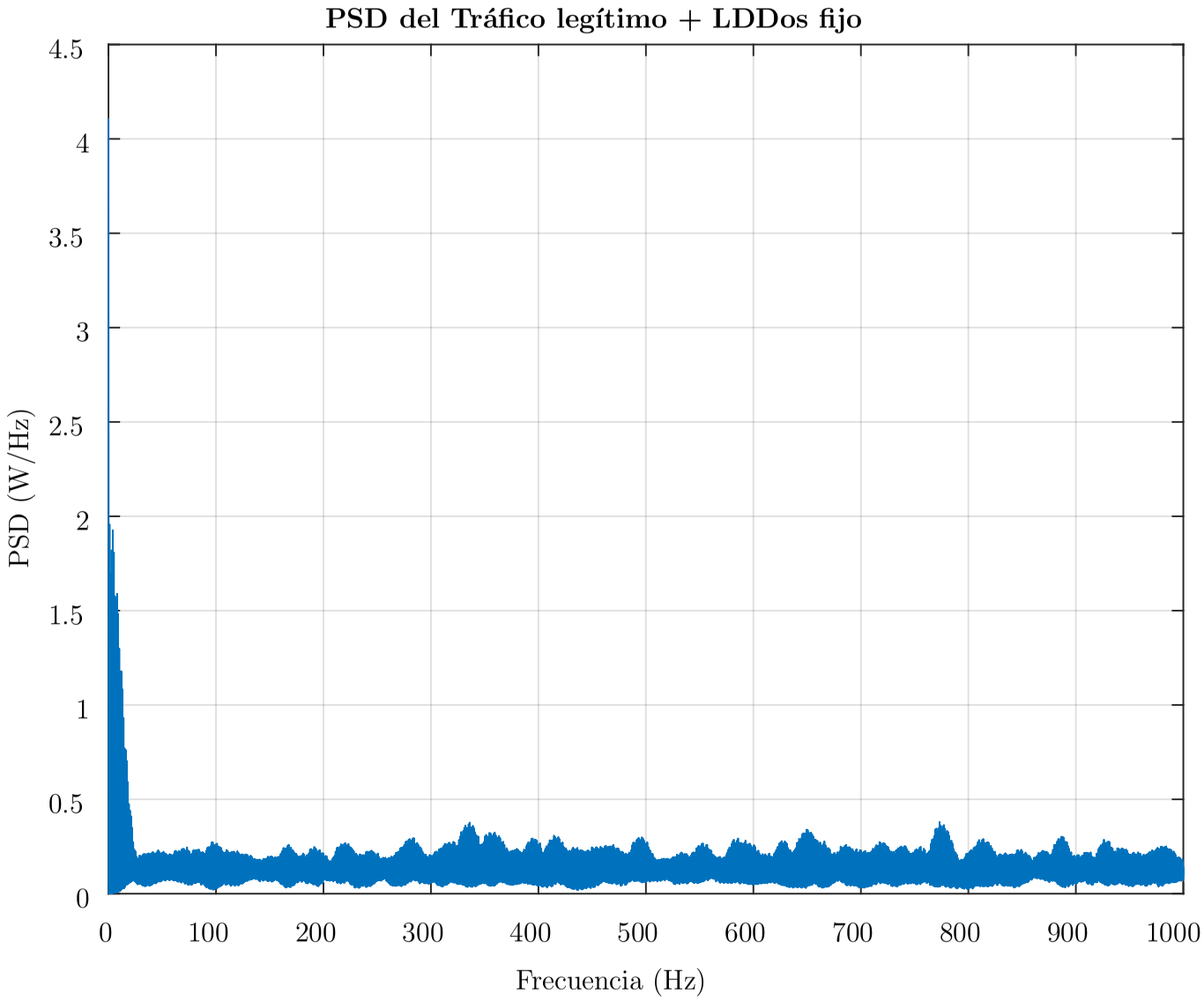

Figura 20: Densidad espectral de potencia del ataque de periodo fijo.

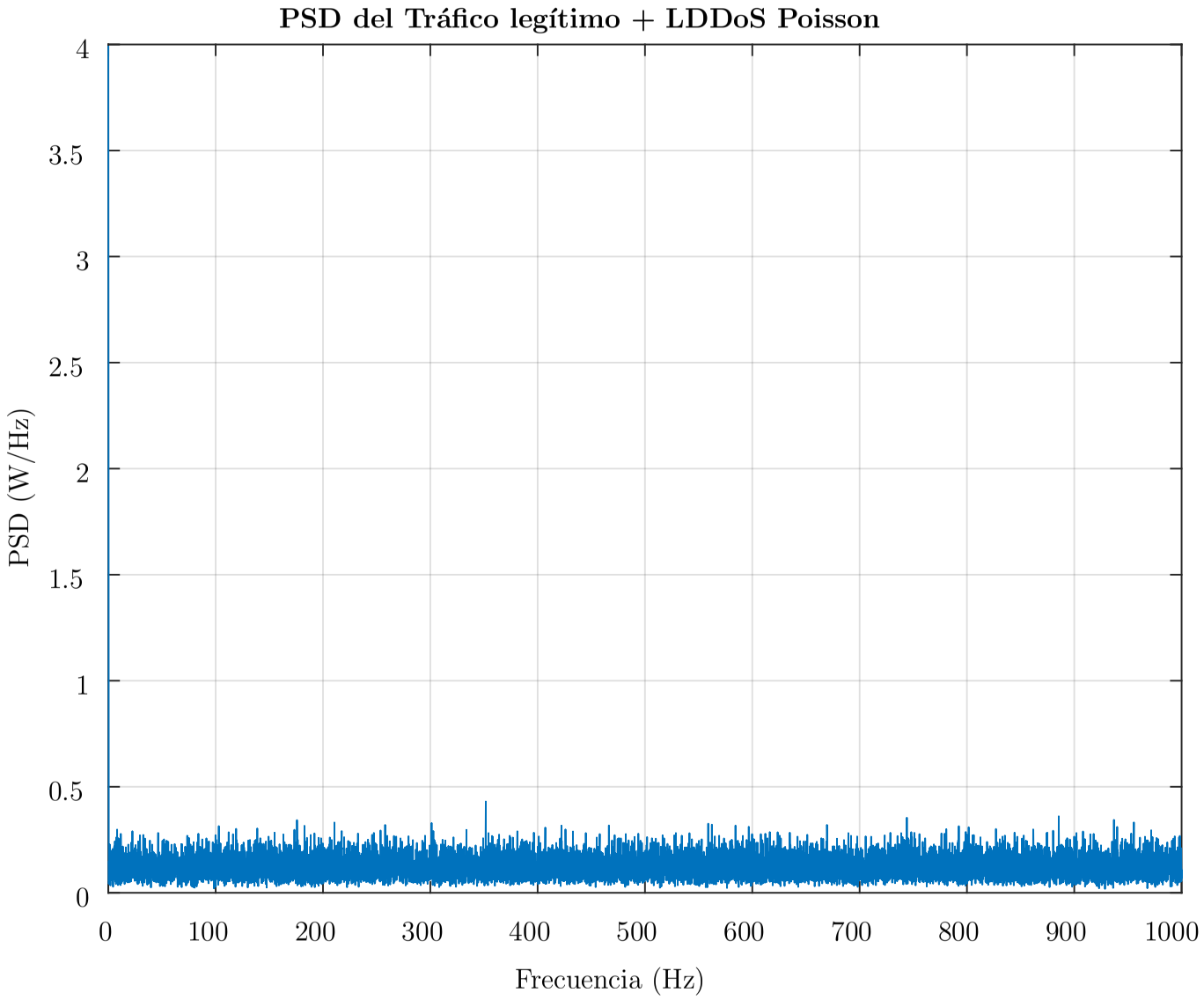

Figura 21: Densidad espectral de potencia del ataque de periodo poissoniano.

6 Conclusión

Los resultados de este trabajo de fin de grado pretenden ser de utilidad para todas aquellas personas que estén tratando de documentarse acerca de los ataques de baja intensidad, en cualquiera de sus dos versiones, ya tengan o no experiencia previa en este ámbito de la ciberseguridad. Esto se debe a la creciente amenaza que este tipo de ataques de denegación de servicio suponen para los nuevos entornos de computación en la nube o para las redes formadas por los dispositivos IoT. La amplia adopción de estas tecnologías en diferentes sectores económicos como los industriales o médicos, hacen que sea de gran importancia que las personas que vayan a dedicarse a la seguridad en estos campos estén familiarizadas con los términos relativos a LDoS y LDDoS.

Además, también se pretende que por medio de las ideas planteadas a lo largo de las dos secciones anteriores los lectores puedan llegar a identificar posibles errores presentes en los mecanismos de detección propuestos hasta el momento, y cuáles son las causas de estas vulnerabilidades. Para ello este documento presenta pruebas tanto teóricas como experimentales a partir de las cuales se quiere reflejar la ineficiencia de algunos métodos.

En concreto, el análisis de los métodos basados únicamente en el cálculo de la densidad espectral de potencia intenta dirigir la atención al gran problema que presentan estas técnicas, el cual es que los ataques que no tengan periodos muy repetitivos pueden acabar camuflándose entre el tráfico inocuo de la red, lo que hace que toda la inversión realizada para implementar estos mecanismos de detección sea insuficiente, ya que se podrá seguir sufriendo ataques que afectarán a la infraestructura de la empresa o entidad generando pérdidas.

Se puede afirmar que la idea principal, que sostiene que muchos de los sistemas de detección de ataques basados en el cálculo de la PSD y desarrollados hasta la fecha no cuentan con las herramientas necesarias para detectar los ataques con periodo aleatorio, ha sido demostrada en gran medida. Pero quedan aún muchas pruebas por efectuar, ya que todos los datos que se han empleado provienen de simulaciones en lugar de capturas efectuadas sobre redes reales. Además, debido al gran número de herramientas que existen en la actualidad, no se puede afirmar que todas padezcan de este mal, pues algunos autores ya remarcaban en su trabajo el problema presentado por las distribuciones aleatorias.

Por otro lado, muchas otras teorías que se planteaban en la sección 4 quedan aún por ser demostradas de forma práctica, ya sea por su complicación a la hora de implementarlas o debido a la falta de tiempo para llevar a cabo todos los experimentos necesarios para ello. Adicionalmente, el código que se han desarrollado a lo largo de este trabajo podrían no estar completamente optimizados.

6.1 Puntos de Mejora

Entre las posibles líneas de investigación a futuro que se relacionan con este trabajo se podría crear entornos de pruebas más cercanos a la realidad donde efectos como los retardos de propagación, procesado, etc. se tengan en cuenta y se vea el efecto que estos tienen, tanto sobre el ataque que llega al servidor como sobre la efectividad de los mecanismos de detección.

Todavía quedan un gran número de artículos que podrían presentar algunos de los inconvenientes presentados en este trabajo, pero hasta que se efectúen las pruebas, tanto teóricas como prácticas, sobre las técnicas que estos presentan no se puede afirmar con total seguridad. Entre las posibles herramientas que se han estudiado, aquellas que implementan inteligencia artificial en alguna de sus muchas formas parecen mostrarse como mecanismos más robustos y con menores margenes de error. Por ello, sería de gran interés poder estudiar si estos métodos también están sujetos a las limitaciones de los métodos convencionales o si por el contrario todos los agujeros presentes son parcheados, convirtiéndolos así en métodos casi infalibles.

A Código fuente

Esta sección recoge las diferentes funciones que se han elaborado durante la realización de este trabajo, acompañadas en cada caso de una breve explicación en la que se explican los parámetros que acepta como entradas y los que se podrán observar como salidas del sistema, además de las operaciones que se realizan.

A.1 LDDoS-Generator

Esta función tiene como objetivo generar la suma instantánea de todas las ráfagas que forman los ataques LDDoS para luego agregarlos a los datos de la Universidad de Valladolid, los cuales se explican en la sección 5.2.1. Este código está preparado para devolver, a partir de unos datos, dos ataques diferentes, uno que presenta un periodo fijo y uno que por otro lado presenta un periodo dependiente de una distribución exponencial.

Para evitar gastar muchos recursos computacionales, en lugar de generar las ráfagas de cada máquina de forma independiente y luego efectuar una suma, se multiplicará el valor de las ráfagas de un único atacante por el número total de bots que participan en el ataque.

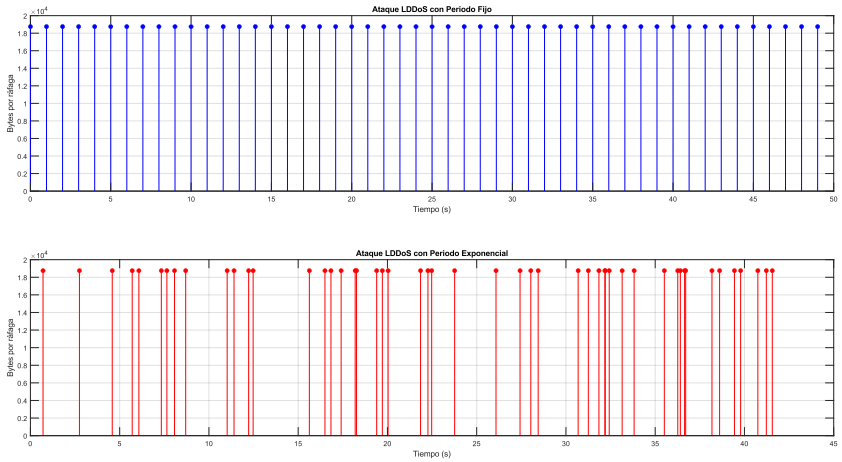

Figura 22: Salida comparada de los dos tipos de ataques generados.

```
% Periodo medio
T = 1;
% Duracion de la rafaga
d = 0.1;
% Tasa de transmision en bytes/segundo
R = 187500;
% Numero de rafagas
Number = 50;
% Numero de bots
n_bots=100;

[timestamp, burst] = LDDoS_Generator(T, d, R, Number,
    n_bots);

function [timestamp, burst] = LDDoS_Generator(T, d, R,
    Number, n_bots)
  % Periodo fijo para el primer tipo de ataque
  fixed_period = T;
  timestamp_fixed = (0:Number-1) * fixed_period;

  % Periodo exponencial para el segundo tipo de ataque
  Periodo = exprnd(T, 1, Number);
  timestamp_exponential = cumsum(Periodo);

  % Burst representa la cantidad total de bytes
      enviados en cada rafaga
  burst = d * R * ones(1, Number) * n_bots;

  % Representar los resultados y salida de datos
  figure;
  subplot(2,1,1);
  stem(timestamp_fixed, burst, 'b', 'filled');
  title('Ataque LDDoS con Periodo Fijo');
  xlabel('Tiempo (s)');
  ylabel('Bytes por rafaga');
  grid on;

  subplot(2,1,2);
  stem(timestamp_exponential, burst, 'r', 'filled');
  title('Ataque LDDoS con Periodo Exponencial');
  xlabel('Tiempo (s)');
```

```
39    ylabel('Bytes por rafaga');
40    grid on;
41
42    timestamp = {timestamp_fixed, timestamp_exponential};
43  end
```

Código 1: Código que genera dos tipos de ataques LDDoS.

A.2 Test de evaluación de métodos de ataque

El código que se presenta a continuación recoge las funciones y variables necesarias para generar una simulación de una serie de flujos, tanto legítimos como dañinos, con los que se pueden poner a prueba las capacidades de un sistema de detección basado en el análisis por medio de la densidad espectral de potencia (PSD).

A partir de la línea 52 del código, se pueden encontrar los mecanismos por los cuales se transforman los instantes de llegada de los paquetes de ambas colecciones a la señal que expresa la concentración de mensajes para cada instante temporal. También se puede encontrar la función pwelch, que lleva a cabo los cálculos de la PSD, y las líneas correspondientes a la generación de las gráficas.

Es importante notar que en el código se pueden encontrar casos donde se emplea la construcción 'nn' como reemplazo de la letra 'ñ', esta sustitución, al igual que la eliminación de las tildes, ha sido realizada para evitar errores en la interpretación del código.

```
%% Parametros para los paquetes legitimos
lambda_legitimos = 5;
T_total = 100;
num_clientes_legitimos = 100;

% Inicializacion de las variables de almacenamiento
t = zeros(num_clientes_legitimos, 1);
instantes_llegada_legitimos = [];

% Generar instantes de llegada de paquetes legitimos
   para cada cliente
for cliente = 1:num_clientes_legitimos
  while t(cliente) < T_total
    tiempo_entre_llegadas = exprnd(1/lambda_legitimos);
    t(cliente) = t(cliente) + tiempo_entre_llegadas;
    if t(cliente) < T_total
      instantes_llegada_legitimos = [
         instantes_llegada_legitimos; t(cliente)];
    end
  end
end

```

```
21 %% Datos de ataque LDDoS con periodo fijo
22 num_maquinas_lddos1 = 500;
23 num_ataques_lddos1 = 10000;
24 T_ataque_lddos1 = 1;
25
26 % Generar los tiempos de llegada de los ataques con
      ruido pequenno
27 tiempos_ataques_lddos1 = (0:num_ataques_lddos1-1) *
      T_ataque_lddos1 + randn(1, num_ataques_lddos1) *
      0.01;
28
29 % Crear un vector que represente la llegada de los
      paquetes en cada ataque
30 instantes_llegada_ataques1 = [];
31 for i = 1:num_ataques_lddos1
32   instantes_llegada_ataques1 = [
       instantes_llegada_ataques1; repmat(
       tiempos_ataques_lddos1(i), num_maquinas_lddos1, 1)
       ];
33 end
34
35 %% Datos de ataque LDDoS modelado como proceso de
      Poisson
36 num_maquinas_lddos2 = 500;
37 num_ataques_lddos2 = 10000;
38
39 % Generar los tiempos de llegada de los ataques con
      periodo exponencial
40 periodos_ataques_lddos2 = exprnd(1, [1,
      num_ataques_lddos2]);
41 tiempos_ataques_lddos2 = cumsum(periodos_ataques_lddos2
      ) + randn(1, num_ataques_lddos2) * 0.01;
42 tiempos_ataques_lddos2 = tiempos_ataques_lddos2(
      tiempos_ataques_lddos2 <= T_total);
43 instantes_llegada_ataques2 = [];
44 for i = 1:length(tiempos_ataques_lddos2)
45   instantes_llegada_ataques2 = [
       instantes_llegada_ataques2; repmat(
       tiempos_ataques_lddos2(i), num_maquinas_lddos2, 1)
       ];
46 end
```

```
47
48 %% Combinar y ordenar los tiempos de llegada (legitimos
       + ataques)
49 data1 = sort([instantes_llegada_legitimos;
       instantes_llegada_ataques1]);
50 data2 = sort([instantes_llegada_legitimos;
       instantes_llegada_ataques2]);
51
52 % 2. Crear la sennal de trafico  para cada coleccion (
       solicitudes por milisegundo)
53 fs = 1000;
54 time = 0:1/fs:(T_total-1/fs);
55 % Sennal para la Coleccion 1
56 signal1 = histcounts(data1, [time, time(end) + 1/fs]);
57 % Sennal para la Coleccion 2
58 signal2 = histcounts(data2, [time, time(end) + 1/fs]);
59
60 % 3. Generar las graficas para las sennales en el
       dominio del tiempo
61
62 % Grafica para la Coleccion 1
63 figure;
64 plot(time, signal1);
65 title('Trafico en el tiempo (Trafico legitimo + LDDoS
       fijo)');
66 xlabel('Tiempo (s)');
67 ylabel('Numero de solicitudes');
68 grid on;
69
70 % Grafica para la Coleccion 2
71 figure;
72 plot(time, signal2);
73 title('Trafico HTTP en el tiempo (Trafico legitimo +
       LDDoS Poisson)');
74 xlabel('Tiempo (s)');
75 ylabel('Numero de solicitudes');
76 grid on;
77
78 % 4. Calcular y representar la Densidad Espectral de
       Potencia (PSD) usando pwelch para ambas colecciones
79
```

```
% PSD para la Coleccion 1
figure;
[pxx1, f1] = pwelch(signal1, [], [], [], fs);
plot(f1, pxx1);
title('PSD del Trafico legitimo + LDDoS fijo');
xlabel('Frecuencia (Hz)');
ylabel('PSD (W/Hz)');
grid on;

% PSD para la Coleccion 2
figure;
[pxx2, f2] = pwelch(signal2, [], [], [], fs);
plot(f2, pxx2);
title('PSD Trafico legitimo + LDDoS Poisson');
xlabel('Frecuencia (Hz)');
ylabel('PSD (W/Hz)');
grid on;
```

Código 2: Código que genera la PSD de los dos tipos de ataques.

B Ataques generados a partir de una distribución de Poisson

En este apéndice trataremos de razonar por qué un tren de pulsos a intervalos aleatorios con distribución exponencial se puede modelar, aproximadamente, como un proceso de ruido blanco. Para ello vamos a calcular la densidad espectral de potencia del proceso de manera explícita, utilizando su definición, y comprobaremos que ésta es constante (salvo por el factor de forma que impone el propio pulso).

B.1 Planteamiento del problema

Un ataque LDDOS puede modelarse como se muestra en la figura 8. Por simplicidad supondremos que se envían pulsos idénticos $p(t)$ espaciados a intervalos temporales $T_1, T_2, \ldots, T_n, \ldots$, que se generan de manera aleatoria según una determinada ley de probabilidad. Asumiremos que es una ley exponencial de parámetro $\lambda > 0$, que es el inverso del tiempo medio entre pulsos:

$$T_n \sim \text{Exp}(\lambda) \;\Leftrightarrow\; f_{T_n}(t) = \lambda \exp(-\lambda\, t) u(t),\; \forall n \in \mathbb{Z}^+ \backslash \{0\}. \tag{12}$$

Esto significa que tenemos un proceso de Poisson, puesto que el número de pulsos que llegan en un intervalo temporal determinado T es una variable aleatoria de este tipo. Entonces, la señal $x(t)$ que representa el ataque LDDOS es un tren de pulsos:

$$x_N(t) = \sum_{n=1}^{N} p(t - T_n), \tag{13}$$

siendo N el número de pulsos enviados durante un intervalo τ_N de interés (la ventana de tiempo en la que estamos llevando a cabo el análisis). La transformada de Fourier del tren de pulsos puede calcularse de manera trivial utilizando las propiedades de linealidad y de desplazamiento temporal:

$$X_N(\omega) = \sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right), \tag{14}$$

siendo $P(\omega)$ la transformada de Fourier del pulso $p(t)$. La densidad espectral de potencia del proceso estocástico se define como:

$$S_X(\omega) = \lim_{\tau_n \to \infty} E\left\{ \frac{1}{2\,\tau_N} \left| X_N(\omega) \right|^2 \right\} = \lim_{\tau_N \to \infty} \frac{1}{2\,\tau_N} E\left\{ \left| X_N(\omega) \right|^2 \right\}, \tag{15}$$

que es la magnitud que calcularemos en las siguientes secciones. Por simplicidad, consideraremos que N (y por tanto τ_N) es suficientemente grande en la ventana temporal considerada como para que el análisis mediante DFT en esta ventana aproxime el valor teórico de la ecuación (15).

B.2 Cálculo de la densidad espectral de potencia $S_X(\omega)$

B.2.1 Módulo cuadrático de la transformada de Fourier

Calculamos primero el término $|X_N(\omega)|^2$ de la ecuación (15) a partir del valor de $X_T(\omega)$ en la expresión (14):

$$\begin{aligned}
|X_N(\omega)|^2 &= \left| \sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right) \right|^2 \\
&= \left(\sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right) \right) \left(\sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right) \right)^* \\
&= \left(\sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right) \right) \left(\sum_{m=1}^{N} P(\omega) \exp\left(-j\omega T_m\right) \right)^* \\
&= \left(\sum_{n=1}^{N} P(\omega) \exp\left(-j\omega T_n\right) \right) \left(\sum_{m=1}^{N} P(\omega)^* \exp\left(j\omega T_m\right) \right) \\
&= P(\omega)P(\omega)^* \left(\sum_{n=1}^{N} \exp\left(-j\omega T_n\right) \right) \left(\sum_{m=1}^{N} \exp\left(j\omega T_m\right) \right) \\
&= |P(\omega)|^2 \sum_{n=1}^{N} \sum_{m=1}^{N} \exp\left(-j\omega T_n\right) \exp\left(j\omega T_m\right) \\
&= |P(\omega)|^2 \sum_{n=1}^{N} \sum_{m=1}^{N} \exp\left(j\omega (T_m - T_n)\right), \qquad (16)
\end{aligned}$$

donde el asterisco simboliza el complejo conjugado. Aunque la expresión anterior involucra un doble sumatorio de números complejos, el resultado ha de ser un número real y positivo porque representa el módulo al cuadrado de un número complejo. Esto no es una contradicción: basta darse cuenta de que las exponenciales van a aparecer siempre en parejas de complejos conjugados; por ejemplo, si aparece el sumando para $\{n = 1, m = 3\}$, aparecerá también el sumando para $\{n = 3, m = 1\}$, y juntos sumarán

$\exp(j\omega(T_3-T_1))+\exp(j\omega(T_1-T_3)) = 2\cos(\omega(T_3-T_1))$. Los únicos sumandos que no aparecerán por parejas serán aquéllos que corresponden al caso $m = n$, pero en este caso $\exp(-j\omega(T_m - T_n)) = \exp(-j\omega(T_n - T_n)) = \exp(0) = 1$. Por tanto, podemos reescribir la ecuación (16) como:

$$\begin{aligned}
|X_N(\omega)|^2 &= |P(\omega)|^2 \left(\underbrace{\sum_{n=1}^{N} 1}_{m=n} + \underbrace{\sum_{n=1}^{N-1} \sum_{m=n+1}^{N} 2\cos\left(\omega(T_m - T_n)\right)}_{m\neq n,\ \text{únicos}} \right) \\
&= |P(\omega)|^2 \left(N + \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} 2\,\Re\left\{\exp\left(j\omega(T_m - T_n)\right)\right\} \right) \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} \exp\left(j\omega(T_m - T_n)\right) \right\} \right). \quad (17)
\end{aligned}$$

B.2.2 Esperanza del módulo cuadrático

A partir del resultado de la ecuación (17), calcularemos el término $E\left\{|X_N(\omega)|^2\right\}$ de la ecuación (15):

$$\begin{aligned}
E\left\{|X_N(\omega)|^2\right\} &= E\left\{ |P(\omega)|^2 \left(N + 2\,\Re\left\{ \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} \exp\left(j\omega(T_m - T_n)\right) \right\} \right) \right\} \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ \sum_{n=1}^{N-1} \sum_{m=n+1}^{N} E\left\{\exp\left(j\omega(T_m - T_n)\right)\right\} \right\} \right), \quad (18)
\end{aligned}$$

donde se ha tenido en cuenta la linealidad del operador esperanza y el hecho de que todas las magnitudes son deterministas, salvo las variables aleatorias T_m y T_n. Aunque el cálculo de $E\left\{\exp\left(j\omega(T_m - T_n)\right)\right\}$ pueda parecer complicado, en realidad no lo es: basta examinar el sumatorio de la ecuación (17) para comprobar que $m > n$ y por lo tanto $T_m - T_n$ es la suma de $m - n$ variables aleatorias exponenciales independientes e idénticamente distribuidas (los $m - n$ tiempos de llegada entre los pulsos n-ésimo y m-ésimo). Ésta es una variable de tipo Erlang con factor de forma $m - n > 0$ y tasa $\lambda > 0$ (el parámetro original de las variables exponenciales T_n), que está perfectamente caracterizada en la literatura.

En particular, $E\left\{\exp\left(j\omega(T_m - T_n)\right)\right\}$ define precisamente la función característica de esta variable $U_{m,n} = T_m - T_n$, cuyo valor es bien conocido[1]:

$$\Phi_{U_{m,n}}(\omega) \stackrel{\Delta}{=} E\{\exp(j\,\omega\,U_{m,n})\} = \left(\frac{\lambda}{\lambda - j\omega}\right)^{m-n} = \Phi_T(\omega)^{m-n}, \qquad (19)$$

donde $\Phi_T(\omega) = \frac{\lambda}{\lambda - j\omega}$ es precisamente la función característica de cada una de las variables exponenciales T_n.

B.2.3 Cálculo explícito de los sumatorios

Insertando el valor de la ecuación (19) en la ecuación (18):

$$\begin{aligned} E\left\{|X_N(\omega)|^2\right\} &= |P(\omega)|^2 \left(N + 2\,\Re\left\{\sum_{n=1}^{N-1}\sum_{m=n+1}^{N} \Phi_T(\omega)^{m-n}\right\}\right) \\ &= |P(\omega)|^2 \left(N + 2\,\Re\left\{\sum_{n=1}^{N-1} \Phi_T(\omega)^{-n} \sum_{m=n+1}^{N} \Phi_T(\omega)^{m}\right\}\right) \qquad (20) \end{aligned}$$

es inmediato identificar que el sumatorio interno en el subíndice m corresponde a la suma de una progresión geométrica de razón $\Phi_T(\omega)$ para cada valor de ω. Entonces:

$$\begin{aligned} &E\left\{|X_N(\omega)|^2\right\} \\ &= |P(\omega)|^2 \left(N + 2\,\Re\left\{\sum_{n=1}^{N-1} \Phi_T(\omega)^{-n} \frac{\Phi_T(\omega)^{N+1} - \Phi_T(\omega)^{n+1}}{\Phi_T(\omega) - 1}\right\}\right) \\ &= |P(\omega)|^2 \left(N + 2\,\Re\left\{\frac{\Phi_T(\omega)^{N+1}}{\Phi_T(\omega) - 1} \sum_{n=1}^{N-1} \Phi_T(\omega)^{-n} - \frac{1}{\Phi_T(\omega) - 1} \sum_{n=1}^{N-1} \Phi_T(\omega)\right\}\right) \\ &= |P(\omega)|^2 \left(N + 2\,\Re\left\{\frac{\Phi_T(\omega)^{N+1}}{\Phi_T(\omega) - 1} \sum_{n=1}^{N-1} \Phi_T(\omega)^{-n} - (N-1)\,\frac{\Phi_T(\omega)}{\Phi_T(\omega) - 1}\right\}\right). \end{aligned} \qquad (21)$$

[1]Demostrar este resultado es sencillo: puesto que la variable Erlang es la suma de $m-n$ variables exponenciales independientes, su densidad de probabilidad será la convolución de $m - n$ densidades de probabilidad exponenciales; ahora bien, la función característica de una variable aleatoria no es más que la transformada de Fourier de su densidad de probabilidad, por ello podemos usar la propiedad de convolución para concluir que la función característica de la variable Erlang es el producto de las $m - n$ funciones características individuales idénticas de cada una de las exponenciales. El cálculo de estas funciones individuales es trivial sin más que aplicar la definición.

El sumatorio en el subíndice n corresponde de nuevo a la suma de una progresión geométrica para cada ω (esta vez con razón $\Phi_T^{-1}(\omega)$) que podemos calcular explícitamente:

$$
\begin{aligned}
&E\left\{|X_N(\omega)|^2\right\} \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ \frac{\Phi_T(\omega)^{N+1}}{\Phi_T(\omega)-1} \, \frac{\Phi_T(\omega)^{-N} - \Phi_T(\omega)^{-1}}{\Phi_T(\omega)^{-1}-1} - (N-1)\,\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right) \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ \frac{1}{\Phi_T(\omega)-1} \, \frac{\Phi_T(\omega) - \Phi_T(\omega)^{N}}{\Phi_T(\omega)^{-1}-1} - (N-1)\,\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right) \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ \frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \, \frac{\Phi_T(\omega) - \Phi_T(\omega)^{N}}{1-\Phi_T(\omega)} - (N-1)\,\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right) \\
&= |P(\omega)|^2 \left(N + 2\,\Re\left\{ -\frac{\Phi_T(\omega)\left(\Phi_T(\omega) - \Phi_T(\omega)^{N}\right)}{\left(1-\Phi_T(\omega)\right)^2} - (N-1)\,\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right) \\
&= N\ |P(\omega)|^2 \left(1 - 2\,\frac{N-1}{N} \Re\left\{ \frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right. \\
&\left. - \frac{2}{N}\,\Re\left\{ \frac{\Phi_T^2(\omega)\left(1-\Phi_T(\omega)^{N-1}\right)}{\left(1-\Phi_T(\omega)\right)^2} \right\} \right).
\end{aligned}
\tag{22}
$$

B.2.4 Cálculo del límite para ventanas infinitamente grandes

Insertando el resultado de la ecuación (22) en la definición (15) obtenemos:

$$
\begin{aligned}
S_X(\omega) \;=\; & \lim_{\tau_N \to \infty} \frac{N}{2\,\tau_N}\ |P(\omega)|^2 \left(1 - 2\,\frac{N-1}{N} \Re\left\{ \frac{\Phi_T(\omega)}{\Phi_T(\omega)-1} \right\} \right. \\
& \left. - \frac{2}{N}\,\Re\left\{ \frac{\Phi_T^2(\omega)\left(1-\Phi_T(\omega)^{N-1}\right)}{\left(1-\Phi_T(\omega)\right)^2} \right\} \right).
\end{aligned}
\tag{23}
$$

Recordemos que τ_N es el tiempo que tardan en llegar N pulsos, es decir, $\tau_N = \sum_{n=1}^{N} T_n$ es la suma de un número elevado de variables aleatorias independientes e idénticamente distribuidas. Cuando N tiende a infinito, el teorema del límite central nos permite argumentar que τ_n tiende a una variable aleatoria gaussiana cuya media es $\frac{N}{\lambda}$ (siendo $\frac{1}{\lambda}$ la media de cada una de las exponenciales T_n) y cuya varianza es $\frac{N}{\lambda^2}$ (siendo $\frac{1}{\lambda^2}$ la varianza de cada una de las exponenciales T_n). Es decir, $\frac{\tau_N}{N}$ tenderá a una gaussiana de media $\frac{1}{\lambda}$ cuya varianza $\frac{1}{N\lambda^2}$ tiende a 0. En el límite, por tanto, $\frac{\tau_N}{N}$ será

simplemente la constante $\frac{1}{\lambda}$. Con este resultado podemos evaluar el límite en la ecuación (23) y sustituir la función característica $\Phi_T(\omega)$ por su valor:

$$
\begin{aligned}
& S_X(\omega) \\
&= \lim_{\tau_N\to\infty} \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - 2\,\frac{N-1}{N}\Re\left\{\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1}\right\} - \frac{2}{N}\,\Re\left\{\frac{\Phi_T^2(\omega)\left(1-\Phi_T(\omega)^{N-1}\right)}{\left(1-\Phi_T(\omega)\right)^2}\right\}\right) \\
&= \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - \lim_{\tau_N\to\infty} 2\,\frac{N-1}{N}\Re\left\{\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1}\right\} - \lim_{\tau_N\to\infty}\frac{2}{N}\,\Re\left\{\frac{\Phi_T^2(\omega)\left(1-\Phi_T(\omega)^{N-1}\right)}{\left(1-\Phi_T(\omega)\right)^2}\right\}\right) \\
&\overset{(3)}{=} \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - \lim_{N\to\infty} 2\,\frac{N-1}{N}\Re\left\{\frac{\Phi_T(\omega)}{\Phi_T(\omega)-1}\right\} - \lim_{N\to\infty}\frac{2}{N}\,\Re\left\{\frac{\Phi_T^2(\omega)\left(1-\Phi_T(\omega)^{N-1}\right)}{\left(1-\Phi_T(\omega)\right)^2}\right\}\right) \\
&\overset{(4)}{=} \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - 2\Re\left\{\frac{\frac{\lambda}{\lambda-j\omega}}{\frac{\lambda}{\lambda-j\omega}-1}\right\} - \lim_{N\to\infty}\frac{2}{N}\,\Re\left\{\frac{\left(\frac{\lambda}{\lambda-j\omega}\right)^2\left(1-\left(\frac{\lambda}{\lambda-j\omega}\right)^{N-1}\right)}{\left(1-\frac{\lambda}{\lambda-j\omega}\right)^2}\right\}\right) \\
&\overset{(5)}{=} \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - 2\Re\left\{\frac{\frac{\lambda}{\lambda-j\omega}}{\frac{j\omega}{\lambda-j\omega}}\right\} - \lim_{N\to\infty}\frac{2}{N}\,\Re\left\{\frac{\left(\frac{\lambda}{\lambda-j\omega}\right)^2\left(1-\cancel{\left(\frac{\lambda}{\lambda-j\omega}\right)^{N-1}}^{\;0}\right)}{\left(\frac{-j\omega}{\lambda-j\omega}\right)^2}\right\}\right) \\
&= \frac{\lambda}{2}\,|P(\omega)|^2 \left(1 - 2\Re\left\{\cancel{\frac{\lambda}{j\omega}}\right\} - \lim_{N\to\infty}\frac{2}{N}\,\Re\left\{\frac{\lambda^2}{-\omega^2}\right\}\right) = \frac{\lambda}{2}\,|P(\omega)|^2, \text{ si } \omega \neq 0. \qquad (24)
\end{aligned}
$$

En el paso (3) hemos utilizado el razonamiento previo de que cuando el número de pulsos tiende a infinito, el tiempo τ_N tiende también a infinito como N/λ; en el paso (4) se ha sustituido la función característica de la variable exponencial por su valor concreto; en el paso (5) se ha tenido en cuenta que el módulo de $\lambda/(\lambda - j\omega)$ será menor que 1 para cualquier $\omega \neq 0$ (el caso $\omega = 0$ debería estudiarse por separado: al tener el tren de pulsos una componente continua, necesariamente aparecerá una singularidad en el origen).

Como puede comprobarse, la densidad espectral de potencia del tren de pulsos LDDOS depende únicamente de la forma del espectro del pulso; si éste es suficientemente breve ($\delta \ll 1/\lambda$), podremos asimilarlo a una delta de Dirac, y por tanto $|P(\omega)|$ será muy aproximadamente plano, es decir, el tren de pulsos será muy aproximadamente un ruido blanco.

Referencias

[Agr18] N. Agrawal y S. Tapaswi. Low rate cloud DDoS attack defense method based on power spectral density analysis. *Information Processing Letters*, volumen 138, págs. 44–50, 2018.

[Ant22] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, J. Halderman, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, Y. Zhou, M. Bernhard, Z. Durumeric, D. Kumar, C. Lever, M. Kallitsis y L. Invernizzi. Understanding the Mirai Botnet Understanding the Mirai Botnet. *26th USENIX Security Symposium*, septiembre de 2022. `https://www.researchgate.net/publication/363475096_Understanding_the_Mirai_Botnet_Understanding_the_Mirai_Botnet`, Visitado el 23/07/2024.

[Azu22] Azure DDoS Protection—2021 Q3 and Q4 DDoS attack trends, enero de 2022. `https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/`, Visitadoel 23/07/2024.

[Bar92] A. Barron, L. Gyorfi y E. van der Meulen. Distribution estimation consistent in total variation and in two types of information divergence. *IEEE Transactions on Information Theory*, volumen 38, nº 5, págs. 1437–1454, 1992.

[Bla06] J. M. Blackledget. Chapter 3 - The Fourier Series. En J. M. Blackledget (editor), *Digital Signal Processing (Second Edition)*, Woodhead Publishing Series in Electronic and Optical Materials, págs. 57–74. Woodhead Publishing, second edition edición, 2006.

[Bla09] E. Blanton, D. V. Paxson y M. Allman. TCP Congestion Control. RFC 5681, septiembre de 2009.

[Bry15] J. Brynielsson y R. Sharma. Detectability of Low-Rate HTTP Server DoS Attacks using Spectral Analysis. En *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pág. 954–961. Association for Computing Machinery, New York, NY, USA, 2015.

[Cal96] R. E. Calem. New York's Panix Service Is Crippled by Hacker Attack. *The New York Times*, septiembre de 1996. `https://archive.nytimes.com/www.nytimes.com/library/cyber/week/0914panix.html` Visitado el 22/07/2024.

[Che06] Y. Chen y K. Hwang. Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *Journal of Parallel and Distributed Computing*, volumen 66, nº 9, págs. 1137–1151, 2006. Special Issue: Security in grid and distributed systems.

[Che12] K. Chen, H. Liu y X. Chen. EBDT: A method for detecting LDoS attack. En *2012 IEEE International Conference on Information and Automation*, págs. 911–916. 2012.

[Chi18] A. A. Chistokhodova y I. D. Sidorov. Novel Method For Low-Rate Ddos Attack Detection. *Journal of Physics: Conference Series*, volumen 1015, nº 3, pág. 032.024, mayo de 2018.

[Cyb23] E. U. A. for Cybersecurity, I. Lella, C. Ciobanu, E. Tsekmezoglou, M. Theocharidou, E. Magonara, A. Malatras y R. Svetozarov Naydenov. *ENISA threat landscape 2023 – July 2022 to June 2023*. ENISA, 2023.

[Dea10] B. Dear. Perhaps the First Denial-of-Service Attack? *PLATO History*, febrero de 2010. `http://www.platohistory.org/blog/2010/02/perhaps-the-first-denial-of-service-attack.html`, Visitado el 08/07/2024.

[Dup13] F. Dupuis, L. Krämer, P. Faist, J. Renes y R. Renner. *Generalized entropies*, págs. 134–153. octubre de 2013.

[Far05] S. Farraposo, L. Gallon y P. Owezarski. Network Security and DoS Attacks. 2005. `https://homepages.laas.fr/owe/METROSEC/Security_and_DoS.pdf`, Visitado el 8/07/2024.

[Gui04] M. Guirguis, A. Bestavros y I. Matta. Exploiting the Transients of Adaptation for RoQ Attacks on Internet Resources. julio de 2004.

[Gui05] M. Guirguis, A. Bestavros, I. Matta y Y. Zhang. Reduction of Quality (RoQ) attacks on Internet end-systems. volumen 2, págs. 1362 – 1372 vol. 2. 2005.

[Gui06] M. Guirguis, A. Bestavros y I. Matta. On the Impact of Low-Rate Attacks. En *2006 IEEE International Conference on Communications*, volumen 5, págs. 2316–2321. 2006.

[Gui07] M. Guirguis, A. Bestavros, I. Matta y Y. Zhang. Reduction of Quality (RoQ) Attacks on Dynamic Load Balancers: Vulnerability Assessment and Design Tradeoffs. págs. 857 – 865. junio de 2007.

[He09] Y.-X. He, Q. Cao, T. Liu, Y. Han y Q. Xiong. Low-rate DoS detection method based on feature extraction using wavelet transform. volumen 20, págs. 930–941, abril de 2009.

[Hir19] M. Hiralal y S. Padhiar. Basic Overview of DDOS Attack. pág. 1, abril de 2019.

[Hua98] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N.-C. Yen, C.-C. Tung y H. Liu. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, volumen 454, págs. 903–995, marzo de 1998.

[IMV64] N. Y. G. I. M.; Vilenkin. *Generalized Functions, Vol. 4: Applications of Harmonic Analysis*. Academic Press, 1964.

[INC21] INCIBE. DrDoS: características y funcionamiento. *INCIBE-CERT Blog*, abril de 2021. `https://www.incibe.es/incibe-cert/blog/drdos-caracteristicas-y-funcionamiento`, Visitado el 08/07/2024.

[Joy11] J. M. Joyce. *Kullback-Leibler Divergence*, págs. 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[Kas12] B. Kashyap y S. K. Jena. DDoS Attack Detection and Attacker Identification . *International Journal of Computer Applications*, volumen 42, nº 1, págs. 27–33, marzo de 2012.

[Kir08] W. Kirch (editor). *Pearson's Correlation Coefficient*, págs. 1090–1091. Springer Netherlands, Dordrecht, 2008.

[Kli07] D. Kliazovich, F. Granelli y M. Gerla. Performance improvement in wireless networks using cross-layer ARQ. *Computer Networks*, volumen 51, nº 15, págs. 4396–4411, 2007.

[Kum14] H. Kumawat y G. Meena. Characterization, Detection and Mitigation of Low-Rate DoS attack. En *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*, ICTCS '14. Association for Computing Machinery, New York, NY, USA, 2014.

[Kuz03] A. Kuzmanovic y E. W. Knightly. Low-rate TCP-targeted denial of service attacks: the shrew vs. the mice and elephants. En *Proceedings*

of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03, pág. 75–86. Association for Computing Machinery, New York, NY, USA, 2003.

[Lei09] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts y S. Wolff. A brief history of the internet. *SIGCOMM Comput. Commun. Rev.*, volumen 39, nº 5, pág. 22–31, octubre de 2009.

[Liu04] D. Liu y D. Shuai. Multifractal characteristic quantities of network traffic models. En *Grid and Cooperative Computing*, págs. 413–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[Liu20] L. Liu, H. Wang, Z. Wu y M. Yue. The detection method of low-rate DoS attack based on multi-feature fusion. *Digital Communications and Networks*, volumen 6, nº 4, págs. 504–513, 2020.

[Luo14] J. Luo y X. Yang. The NewShrew attack: A new type of low-rate TCP-Targeted DoS attack. En *2014 IEEE International Conference on Communications (ICC)*, págs. 713–718. 2014.

[Maf22] MafiaBoy, the hacker who took down the Internet, febrero de 2022. `https://www.blackhatethicalhacking.com/articles/hacking-stories/mafiaboy-the-hacker-who-took-down-the-internet/`, Visitado el 22/07/2024.

[MF06] G. Maciá-Fernández, J. E. Díaz-Verdejo y P. García-Teodoro. Assessment of a vulnerability in iterative servers enabling low-rate dos attacks. En *Proceedings of the 11th European Conference on Research in Computer Security*, ESORICS'06, pág. 512–526. Springer-Verlag, Berlin, Heidelberg, 2006.

[MF08a] G. Maciá-Fernández, J. E. Díaz-Verdejo, P. García-Teodoro y F. de Toro-Negro. LoRDAS: A Low-Rate DoS Attack against Application Servers. En *Critical Information Infrastructures Security*, págs. 197–209. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[MF08b] G. Maciá-Fernández, J. E. Díaz-Verdejo y P. García-Teodoro. Evaluation of a low-rate DoS attack against application servers. *Computers & Security*, volumen 27, págs. 335–354, diciembre de 2008.

[MF09] G. Maciá-Fernández, J. E. Díaz-Verdejo y P. García-Teodoro. Mathematical Model for Low-Rate DoS Attacks Against Application Servers. *IEEE Transactions on Information Forensics and Security*, volumen 4, nº 3, págs. 519–529, 2009.

[Mor23] S. Morgan. Cybercrime To Cost The World $9.5 trillion USD annually in 2024, octubre de 2023. `https://cybersecurityventures.com/cybercrime-to-cost-the-world-9-trillion-annually-in-2024`, Visitado el 18/09/2024.

[New18] L. H. Newman. GitHub Survived the Biggest DDoS Attack Ever Recorded, marzo de 2018. `https://www.wired.com/story/github-ddos-memcached/`, Visitado el 23/07/2024.

[NS-] The Network Simulator NS-2. Disponible en `https://www.isi.edu/websites/nsnam/`.

[Ols14] P. Olson. The Largest Cyber Attack In History Has Been Hitting Hong Kong Sites. *Forbes*, noviembre de 2014. `https://www.forbes.com/sites/parmyolson/2014/11/20/the-largest-cyber-attack-in-history-has-been-hitting-hong-kong-sites/?sh=3c8865e438f6`, Visitado el 23/07/2024.

[Ost19] E. Osterweil, A. Stavrou y L. Zhang. 20 Years of DDoS: a Call to Action. *arXiv: Networking and Internet Architecture*, volumen abs/1904.02739, abril de 2019. `https://arxiv.org/abs/1904.02739`, Visitado el 22/07/2024.

[PA03] J. J. Pazos Arias, A. Suárez González y R. I. Díaz Redondo. *Teoría de Colas y Simulación de Eventos Discretos*. Pearson-Prentice Hall, Madrid, 2003.

[Pin20] M. Pinho. AWS Shield Threat Landscape report is now available, mayo de 2020. `https://aws.amazon.com/es/blogs/security/aws-shield-threat-landscape-report-now-available/`, Visitado el 23/07/2024.

[Pro22] J. Prosise. *Applied Machine Learning and AI for Engineers: Solve Business Problems That Can't Be Solved Algorithmically*. O'Reilly, 2022.

[Sah18] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. Rodrigues, B. Sahoo y R. Dash. An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. *Future Generation Computer Systems*, volumen 89, págs. 685–697, 2018.

[Sar11] M. Sargent, J. Chu, D. V. Paxson y M. Allman. Computing TCP's Retransmission Timer. RFC 6298, junio de 2011.

[Sha49] C. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, volumen 37, nº 1, págs. 10–21, enero de 1949.

[Sri11] V. Srinivasan y N. Gopalan. A Flow Monitoring based Distributed Defense Technique for Reduction of Quality Attacks in MANET. *International Journal of Computer Applications*, volumen 21, Mayo de 2011.

[SW09] F. Simmross-Wattenberg. *Detección de anomalías en el tráfico agregado de redes IP basada en inferencia estadística sobre un modelo α-estable de primer orden.* Tesis Doctoral, Universidad de Valladolid, 2009.

[Sys21] Systems and Networks Lab. UTSA2021 Low rate DoS Attack. Informe técnico, Department of Computer Science, The University of Texas at San Antonio, 2021. `https://github.com/utsanetsys/UTSA-2021-Low-rate-DoS-Attack/tree/main`, Accesible el 01/09/2024.

[TG22] A. A. Torres-García, O. Mendoza-Montoya, M. Molinas, J. M. Antelis, L. A. Moctezuma y T. Hernández-Del-Toro. Chapter 4 - Preprocessing and feature extraction. En *Biosignal Processing and Classification Using Computational Learning and Intelligence*, págs. 59–91. Academic Press, 2022.

[Ved21] V. Vedula, P. Lama, R. V. Boppana y L. A. Trejo. On the detection of low-rate denial of service attacks at transport and application layers. *Electronics*, volumen 10, nº 17, 2021.

[Vij13] J. Vijayan. Update: Spamhaus hit by biggest-ever DDoS attacks, marzo de 2013. `https://www.computerworld.com/article/1530959/update-spamhaus-hit-by-biggest-ever-ddos-attacks.html`, Visitado el 23/07/2024.

[VN16] S. Vaughan-Nichols. The Dyn report: What we know so far about the world's biggest DDoS attack, octubre de 2016. `https://www.zdnet.com/home-and-office/networking/the-dyn-report-what-we-know-so-far-about-the-worlds-biggest-ddos-attack/`, Visitado el 23/07/2024.

[Wan12] F. Wang, H. Wang, X. Wang y J. Su. A new multistage approach to detect subtle DDoS attacks. *Mathematical and Computer Modelling*, volumen 55, nº 1, págs. 198–213, 2012. Advanced Theory and Practice for Cryptography and Future Security.

[Wel67] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, volumen 15, nº 2, págs. 70–73, junio de 1967.

[Wu11] Z. Wu, C. Wang y H. Zeng. Research on the comparison of Flood DDoS and Low-rate DDoS. En *2011 International Conference on Multimedia Technology*, págs. 5503–5506. julio de 2011.

[Wu16] Z. Wu, L. Zhang y M. Yue. Low-Rate DoS Attacks Detection Based on Network Multifractal. *IEEE Transactions on Dependable and Secure Computing*, volumen 13, nº 5, págs. 559–567, 2016.

[Wu18] X. Wu, D. Tang, L. Tang, J. Man, S. Zhan y Q. Liu. A Low-Rate DoS Attack Detection Method Based on Hilbert Spectrum and Correlation. En *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, págs. 1358–1363. 2018.

[Wu19] Z. Wu, Q. Pan, M. Yue y L. Liu. Sequence alignment detection of TCP-targeted synchronous low-rate DoS attacks. *Computer Networks*, volumen 152, págs. 64–77, 2019.

[Xia11] Y. Xiang, K. Li y W. Zhou. Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics. *IEEE Transactions on Information Forensics and Security*, volumen 6, nº 2, págs. 426–437, 2011.

[Xu10] X. Xu, X. Guo y S. Zhu. A queuing analysis for low-rate DoS attacks against application servers. En *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, págs. 500–504. 2010.

[Yoa24] O. Yoachimik y J. Pacheco. Informe sobre las amenazas DDoS en el 4º trimestre de 2023. *Cloudflare*, septiembre de 2024. `https://blog.cloudflare.com/ddos-threat-report-2023-q4-es-es`, Visitado el 23/07/2024.

[Yu13] S. Yu. *Distributed Denial of Service Attack and Defense*. Springer Publishing Company, Incorporated, 2013.

[Yue16] M. Yue, Z. Wu y M. Wang. A New Exploration of FB-Shrew Attack. *IEEE Communications Letters*, volumen 20, nº 10, págs. 1987–1990, 2016.

[Yue21] M. Yue, M. Wang y Z. Wu. Low-High Burst: A Double Potency Varying-RTT Based Full-Buffer Shrew Attack Model. *IEEE Transactions on Dependable and Secure Computing*, volumen 18, nº 5, págs. 2285–2300, 2021.

[Zha10] J. Zhang, Z. Qin, L. Ou, P. Jiang, J. Liu y A. X. Liu. An advanced entropy-based DDOS detection scheme. En *2010 International Conference on Information, Networking and Automation (ICINA)*, volumen 2, págs. V2–67–V2–71. 2010.

[Zha12] C. Zhang, Z. Cai, W. Chen, X. Luo y J. Yin. Flow level detection and filtering of low-rate DDoS. *Computer Networks*, volumen 56, nº 15, págs. 3417–3431, 2012.