Ataque de Denegación de Servicio Distribuido utilizando Contenedores

Distributed Denial of Service Attack using Containers

Jairo Quilumbaquin, Camila Rivera, Stalin Rivera, Dylan Tipán
Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE
{jsquilumbaquin, carivera14, sbrivera1, djtipan2}@espe.edu.ec

Resumen

Este estudio se enfoca en adquirir experiencia en programación de Dockers mediante la creación de un Botnet de Denegación de Servicios Distribuido (DDoS). La estructura implica la sincronización de Bots a través de un coordinador, que luego dirige un ataque al servidor objetivo en un momento específico. La motivación radica en comprender el funcionamiento de los ataques DDoS y la manipulación de redes distribuidas para fines maliciosos, así como en explorar las vulnerabilidades subyacentes en sistemas informáticos. La implementación práctica del Botnet proporciona una comprensión de los protocolos de red y la seguridad cibernética. Los resultados subrayaron la capacidad de crear y coordinar un ataque DDoS utilizando técnicas de programación de Dockers, lo que destaca la importancia de las medidas de seguridad para mitigar este tipo de amenazas. Además, el estudio resalta la necesidad de concienciación y acciones preventivas por parte de los administradores de sistemas y desarrolladores de software para proteger las infraestructuras digitales contra ataques cibernéticos.

Palabras clave: *Botnet, Denegación de Servicio, Docker, Python, Ciberseguridad.*

Abstract

This study focuses on gaining experience in Dockers programming by creating a Distributed Denial of Service (DDoS) Botnet. The structure involves synchronizing bots through a coordinator, which then directs an attack to the target server at a specific time. The motivation lies in understanding how DDoS attacks work and in manipulating distributed networks for malicious purposes, as well as in exploring the underlying vulnerabilities in computer systems. Practical implementation of the Botnet provides an understanding of network protocols and cybersecurity. The results underscored the ability to create and to coordinate a DDoS attack by using Dockers programming techniques, and by highlighting the importance of security measures to mitigate these types of threats. Furthermore, the study highlights the need for awareness and preventive actions by system administrators and software developers to protect digital infrastructures against cyberattacks.

Keywords: *Botnet, Service Deny, Docker, Python, Cybersecurity*

Fecha de Recepción: 19/03/2024 - Aceptado: 22/03/2024 - Publicado: 31/03/2024
ISSN: 2477-9253 – DOI: http://dx.doi.org/10.24133/RCSD.VOL09.N01.2024.05

I. Introducción

La apertura y escalabilidad de la tecnología de redes impulsa la creación de diversas aplicaciones en línea, conectando a las personas a escala mundial. Este auge de las aplicaciones informáticas aumenta el flujo de información vital a través de las redes públicas. Hoque, Bhattacharyya, y Kalita (2015) opinan que los sistemas informáticos basados en redes, vitales para el uso personal y profesional, están diseñados para optimizar la utilización de recursos compartidos. El Internet al ser una red de escala global que conecta múltiples dispositivos, también tiene una gran serie de vulnerabilidades que son potenciales objetivos de ataques informáticos.

Debido a sus vulnerabilidades inherentes, los sistemas en red son blanco frecuente de diversos ataques, cuyo objetivo es obtener información sensible o perjudicar a los competidores (Kumari & Jain, 2023). A pesar de los recientes avances en el diseño de cortafuegos y criptografía, persisten las limitaciones. Los dispositivos de defensa que detectan las brechas ofrecen otro método preventivo. Sin embargo, cada día surgen ataques nuevos y sofisticados, y los ataques de denegación de servicio siguen siendo habituales y destructivos.

La base de este estudio se fundamenta en el trabajo realizado por Hoque, Bhattacharyya y Kalita (2015), quienes identificaron las botnets como amenazas significativas para la seguridad de las redes. En su análisis, proporcionan una visión general de las arquitecturas de las redes de bots, resaltando una serie de problemas que plantean desafíos de investigación en el campo de la detección y defensa contra los ataques de denegación de servicio distribuido (DDoS). Este enfoque no solo reconoce la gravedad de estas amenazas, sino que también señala la necesidad de desarrollar estrategias efectivas para mitigar su impacto y proteger la integridad de las infraestructuras de red.

El objetivo de este estudio es implementar una red de Botnets que permitan la emulación de un ataque DDoS, hacia un servidor ficticio que sirve como objetivo de ataque, para generar experiencias y casos de prueba en un entorno controlado, que sirva como herramienta para probar distintos métodos de mitigación y detección de ataques. Además, se implementó la red haciendo uso de contenedores docker para estandarizar la implementación de la herramienta, simplificar su configuración y facilitar su despliegue. Los resultados muestran la funcionalidad de la implementación y la eficacia de la herramienta empleada para el ataque.

La principal contribución de este estudio fue la creación de una red de contenedores Docker que ejecutan un código en Python para la implementación de la arquitectura de botnet maestro-esclavo. La red de contenedores se usa para llevar a cabo un ataque DDoS. Se explora así un enfoque distinto al uso común de Docker y las distintas aplicaciones de las tecnologías de contenerización actuales.

El resto del artículo ha sido organizado como sigue: La sección 2 establece el marco teórico sobre el cual se levanta nuestro artículo. La sección 3 describe los materiales y métodos empleados para el desarrollo de la herramienta. La sección 4, explica la evaluación de resultados y discusión. Finalmente, se presentan las conclusiones y trabajo futuro en la sección 5.

II. Trabajos Relacionados

A continuación, se muestra una lista de trabajos relacionados con el tema desarrollado en el artículo, de forma que se tenga una visión más amplia de las redes de Botnets y su aplicación en los ataques DDoS:

Hoque et al. (2015) presentan una revisión exhaustiva de los ataques de denegación de servicio distribuido (DDoS), enfocándose en las amenazas que plantean las botnets. El estudio ofrece perspectivas sobre las causas y los tipos de ataques DDoS, así como una taxonomía para una mejor comprensión. Además, se abordan los aspectos técnicos de una variedad de herramientas de lanzamiento de ataques. Los autores examinan las arquitecturas de las botnets y cómo se han desarrollado y utilizado en el lanzamiento de ataques DDoS.

Vishwakarma y Jain (2019) proponen una técnica basada en honeypots y aprendizaje automático para detectar malware en dispositivos IoT y protegerse de ataques DDoS basados en botnets. Su objetivo es abordar las nuevas variantes de malware, como los ataques Zero-Day, a medida que aumentan estos ataques. Utilizan datos de honeypots para entrenar un modelo de aprendizaje automático dinámico, lo que demuestra un avance en la defensa contra los ataques DDoS Zero-Day en el Internet de las cosas.

Acosta-Tejada, Sanchez-Galan, y Torres-Batista (2023) abordan el desequilibrio de datos en la clasificación de ataques de denegación de servicio distribuido (DDoS) y sugieren una solución utilizando datos sintéticos. Para obtener datos adicionales, utilizan el conjunto de datos CICDDoS2019 y Redes Generativas Antagónicas (GAN). Demuestran que el uso de GANs mejora significativamente la precisión, con tasas del 98-99% utilizando GANs en comparación con métodos de clasificación convencionales.

Lizares Figueroa y López Benavides (2017) desarrollaron un módulo de seguridad que funciona con Apache y Linux para prevenir y detectar ataques de denegación de servicio distribuido (DDoS) en servidores web. Para realizar pruebas contra ataques DDoS, utilizaron máquinas virtuales con Kali Linux en varios equipos físicos. Utilizaron un diseño metodológico que incluyó un examen previo sin protección y un examen posterior con la implementación del módulo de seguridad.

Narváez, Romero, y Núñez (2010) presentaron un estudio sobre cómo evaluar los mecanismos de protección contra los ataques de denegación de servicio (DoS y DDoS). Proponen evaluar las técnicas de ejecución de ataques mediante herramientas y desarrollar una herramienta personalizada utilizando tecnología de múltiples hilos, aplicada en escenarios de pruebas reales. Implementaron mecanismos de monitoreo para detectar ataques DoS y DDoS examinando el tráfico de la red. Los resultados experimentales demuestran que estas protecciones son efectivas contra ciertos tipos de ataques. No obstante, reconocen que no son 100% efectivas y destacan la importancia de estrategias de seguridad adecuadas.

Molano Mendoza (2024) ofrecen una descripción detallada de los ataques distribuidos de denegación de servicios (DDoS), discutiendo las motivaciones de los atacantes, cómo afectan las industrias afectadas y las repercusiones. El documento ofrece pautas para la prevención y contención de incidentes de seguridad y compara una variedad de tipos de ataques DDoS. Se proporciona una lista completa de las técnicas, herramientas y estrategias utilizadas para llevar a cabo estos ataques.

Tamayo Portero (2023) ofrece un enfoque para detectar ataques de denegación de servicio (DDoS) que se activan a través de botnets en redes definidas por software (SDN). Esta investigación examina los problemas de seguridad que surgen en las redes SDN, que enfrentan amenazas similares a las redes convencionales. El framework utilizando herramientas de código abierto como Mininet y OpenDaylight, probado en una topología de red centralizada con BYOB y SNORT, es parte del estudio. Los hallazgos experimentales demuestran una detección efectiva de ataques DDoS en tiempo real, lo que permite una solución rápida y precisa para proteger las redes SDN de este tipo de amenazas.

Bárbaro y Durante (2022) proponen la creación de una plataforma de mitigación de ataques de denegación de servicio distribuidos (DDoS) en la nube que utilice software libre y estándares abiertos. Esta plataforma

incluye múltiples componentes, incluidos nodos de limpieza de tráfico distribuidos que se pueden administrar a través de una interfaz web única. La iniciativa tiene como objetivo ayudar a las pequeñas y medianas empresas a lidiar con los altos costos de los servicios de mitigación de DDoS.

Pacheco Manotas (2022) explora cómo funcionan los ataques de denegación de servicio distribuido (DDoS), cómo afectan y cómo evitarlos. Destaca la frecuencia de estos ataques en áreas como el sector financiero y propone medidas para reducirlos. Además, le brinda un entendimiento profundo de las técnicas de ataque y las estrategias de mitigación al compartir su experiencia como entidad afectada por estos ataques, tanto en su plataforma web como en sus servicios de telefonía.

III. Marco teórico

3.1. Botnet

Una botnet se refiere a una colección de ordenadores infectados con malware, comúnmente conocidos como zombis, controlados por una entidad maestro conocida como bootmaster (Tuan et al., 2020). Estos zombis operan bajo la dirección remota del bootmaster, que emite instrucciones para diversas acciones destructivas. La arquitectura de los mecanismos de mando y control de la botnet, que puede ser P2P, HTTP, DNS o basada en IRC, dictamina cómo se gestionan los bots. Los ciberdelincuentes aprovechan estas redes de bots para ejecutar actividades como el envío de mensajes de spam, la realización de ataques de DoS o el robo de información personal, como contraseñas de cuentas bancarias o direcciones de correo electrónico. En particular, el spam constituye más del 80% del tráfico de correo electrónico, y las redes de bots suelen utilizarse para la difusión de este tipo de comunicaciones.

3.2. Ataques DDos (Denegación de servicios distribuidos)

Un ataque de denegación de servicio se caracteriza por un esfuerzo deliberado para impedir que los usuarios autorizados accedan a un servicio. En un ataque de denegación de servicio distribuido, se emplean múltiples entidades atacantes. Este artículo aborda específicamente los ataques de denegación de servicio (DDoS) dentro del dominio informático, en los que se engaña a la víctima para que reciba tráfico malicioso, con el consiguiente daño potencial (Mirkovic & Reiher, 2004).

3.3. Python

Python es un lenguaje de programación de alto nivel creado por Guido van Rossum y lanzado por primera vez en 1991. Una de las características más destacadas de Python es su sintaxis simple y legible, lo que lo hace ideal para principiantes y profesionales por igual. Esto lo hace versátil y portátil, ya que los programas escritos en Python pueden ejecutarse en una variedad de plataformas sin modificaciones. Se utiliza en una variedad de campos, incluyendo desarrollo web, análisis de datos, inteligencia artificial, aprendizaje automático, automatización de tareas, scripting, desarrollo de juegos, y más.

3.4. Docker

De acuerdo con (Docker Inc, 2024) docker se presenta como una plataforma abierta destinada al desarrollo, distribución y ejecución de aplicaciones. Su función principal radica en la capacidad de separar las aplicaciones de la infraestructura, lo que facilita la entrega rápida de software. Al utilizar Docker, es posible administrar tanto las aplicaciones como la infraestructura de manera eficiente. Su principal aplicación reside en

simplificar el proceso de envío, prueba y despliegue de código, lo que conlleva a una reducción significativa del tiempo necesario desde la creación del código hasta su puesta en producción.

IV. Materiales y Métodos

4.1. Prognosis

Performance evaluation of Botnet DDoS attack detection using machine learning: Tuan et al. evaluaron la eficacia de métodos para detectar ataques DDoS de botnets mediante algoritmos de aprendizaje automático. Utilizaron SVM, ANN, NB, DT y USML, evaluando conjuntos de datos como UNBS-NB 15 y KDD99. Su enfoque se centra en la ciberseguridad y otros campos relacionados.

Evaluación de ataques de denegación de servicio distribuido (DDoS) y mecanismos de protección: Narváez et al. evaluaron mecanismos de protección contra ataques de DoS y DDoS, utilizando herramientas y desarrollando una propia con tecnología de múltiples hilos. Implementaron soluciones de monitoreo y seguridad basadas en software, como políticas de filtrado con IPTables. Reconocen la importancia de estrategias de seguridad adecuadas.

Detección de ataques de denegación de servicio activados mediante botnets en redes definidas por software: Tamayo et al. propusieron un enfoque para detectar ataques DDoS activados por botnets en redes SDN. Utilizaron herramientas de código abierto como Mininet y OpenDaylight, probando en una topología de red centralizada con BYOB y SNORT. Su investigación demostró una detección efectiva en tiempo real de ataques DDoS.

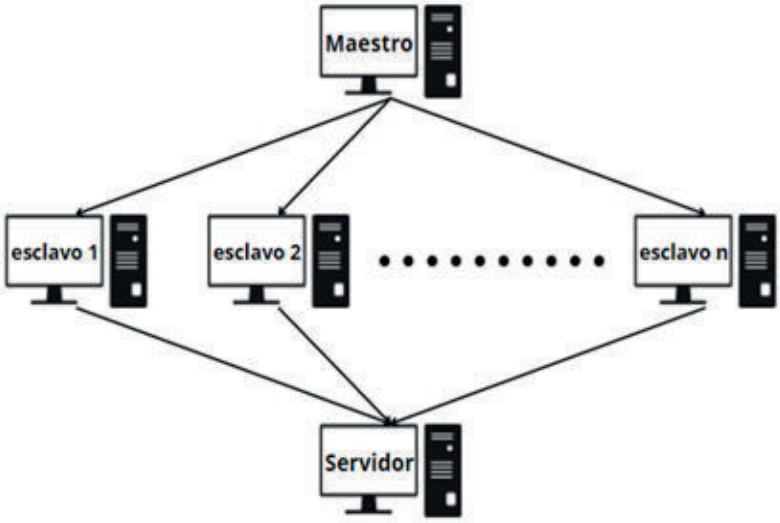
Mitigación de ataques de denegación de servicio distribuidos en la nube: Barbaro et al. propusieron la creación de una plataforma de mitigación de ataques DDoS en la nube utilizando software libre y estándares abiertos. Su enfoque incluyó nodos de limpieza de tráfico distribuidos administrables desde una única interfaz web. Su objetivo era ayudar a las pequeñas y medianas empresas a enfrentar los altos costos de los servicios de mitigación de DDoS.

Ataques de Denegación de Servicio Distribuido, ¿Cómo Evitarlos y Cómo Enfrentarlos? Pacheco et al. exploraron cómo funcionan los ataques DDoS, cómo afectan y cómo evitarlos. Destacaron la frecuencia de estos ataques en sectores como el financiero y propusieron medidas para reducirlos. Su enfoque se basó en compartir experiencias como entidad afectada por estos ataques, brindando una comprensión profunda de las técnicas de ataque y las estrategias de mitigación.

4.2. Topología de red

El diagrama presentado en la Figura 1 representa la red de bots diseñada específicamente para ejecutar ataques DDoS. Esta red implementa una topología seleccionada en base al estudio de Hoque, Bhattacharyya y Kalita (2015). La elección de esta topología se justifica por su capacidad para sincronizar el ataque entre los dispositivos de manera eficiente. En este modelo, el maestro emite la señal de confirmación del ataque, lo que desencadena automáticamente la actividad de los bots, dirigida hacia el objetivo previamente establecido.

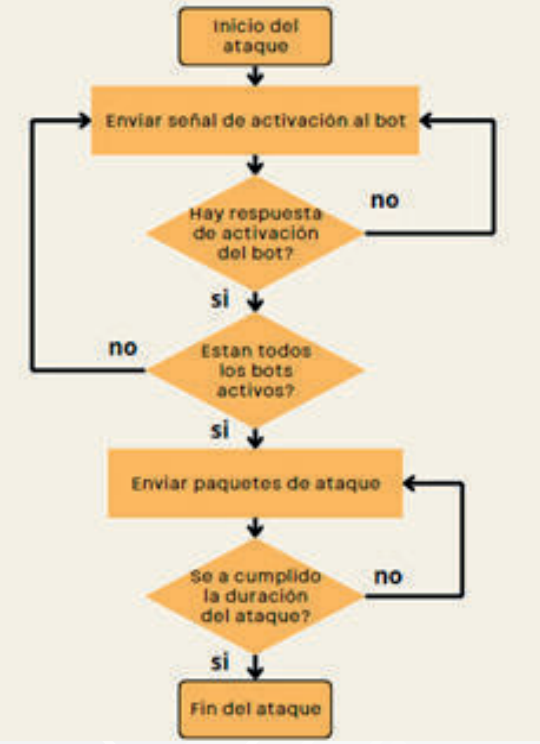
Figura 1: *Topología general de la red de bots*



4.3. Diagrama de flujo de la ejecución del ataque

En la figura 2 se aprecia un diagrama de flujo de la ejecución del ataque llevado a cabo haciendo uso de la botnet implementada con contenedores. En la misma se brinda una breve descripción de cómo funciona el programa y los pasos a seguir para la activación de los esclavos y el envío de los paquetes maliciosos hacia el servidor durante un periodo determinado de tiempo.

Figura 2: *Diagrama de flujo de la ejecución del ataque DDoS*



4.4. Configuración de la Botnet

A continuación, se detallan los scripts utilizados mediante botnet para configurar los elementos de nuestra red y asegurar el funcionamiento y la ejecución del DDoS:

a) Bot.py

El script Bot crea un socket en el puerto especificado y comienza a escuchar al Master. Una vez que el Máster establece la conexión, primero se autentica mediante un proceso de saludo. Si la autenticación es exitosa, el Bot envía al Máster su tiempo actual. Luego, el Master responde con un mensaje que contiene el nombre de host y el número de puerto del servidor objetivo, teniendo en cuenta cualquier diferencia de tiempo entre el Bot y el Master.

Luego, el Bot espera hasta el momento especificado para el ataque. Una vez que es hora de atacar, el Bot se conecta al Servidor Objetivo en el puerto especificado y comienza a enviar mensajes durante 30 segundos, es decir, simula un ataque de denegación de servicio (DoS). Una vez que se completan los 30 segundos y el ataque ha terminado, el Bot se desconecta del Servidor Objetivo y termina su ejecución.

b) Master.py

El script Master lee el archivo de texto que contiene la información del Bot y luego recorre cada entrada del Bot. Cada Bot, se conecta al puerto especificado y se autentica a través de un proceso de saludo. Una vez autenticado, el Master solicita al Bot su tiempo actual y determina cualquier diferencia de tiempo entre él y el Bot. Luego, envía al Bot el nombre de host del Servidor Objetivo y el puerto al que atacar, así como cuándo atacar.

Una vez que esta información se ha transmitido al Bot, continúa con el siguiente Bot hasta que todos los Bots son notificados sobre el ataque.

c) TargetServer.py

El script del Servidor Objetivo simplemente ejecuta un servidor ficticio en el puerto especificado. Escucha en el puerto para que los clientes se conecten, para cada cliente nuevo, crea un nuevo hilo para manejar al cliente. Dentro del manejador de clientes, simplemente continúa recibiendo datos del cliente hasta que el cliente termine la conexión. En este caso, escuchará a que los Bots se conecten, una vez conectados, se creará un hilo separado para cada uno. Dentro del hilo, seguirá recibiendo datos del Bot durante 30 segundos.

4.5. Configuración de los contenedores Docker

Dentro del proyecto Docker se utilizó para crear contenedores que simulan un entorno distribuido de bots y un servidor. Cada contenedor representa un componente del sistema (servidor, clientes/bots) y se ejecuta de manera aislada en su propio entorno. La forma en que se empleó docker dentro del proyecto se detalla a continuación:

a) Creación de Redes y Contenedores

Se crea una red llamada “web_server” utilizando el comando docker network create. Luego, se lanzan varios contenedores (server, client1, client2, clientM) en la misma red utilizando docker run. Estos contenedores se comunican entre sí a través de la red creada.

b) Configuración del Entorno

Dentro de cada contenedor, se instalan las dependencias necesarias como Python 2, nano, y otros paquetes específicos para el proyecto.

c) Configuración de contenedores

Se crea un directorio llamado "test" dentro de cada contenedor, y se editan archivos específicos del proyecto utilizando el editor de texto nano.

d) Instalación de bibliotecas adicionales

Para el contenedor del servidor, se instala la biblioteca timex mediante pip.

a) Ejecución del proyecto

Se ejecutan los scripts Python en cada contenedor para iniciar el servidor, bots y el maestro. Estos scripts son TargetServer.py, Bot.py, y Master.py.

b) Configuración de ataque DDos

Se lanza un ataque DDoS simulado ejecutando los scripts del servidor, bots y el maestro con configuraciones específicas, como puertos y tiempos de ataque.

c) Visualización de resultados

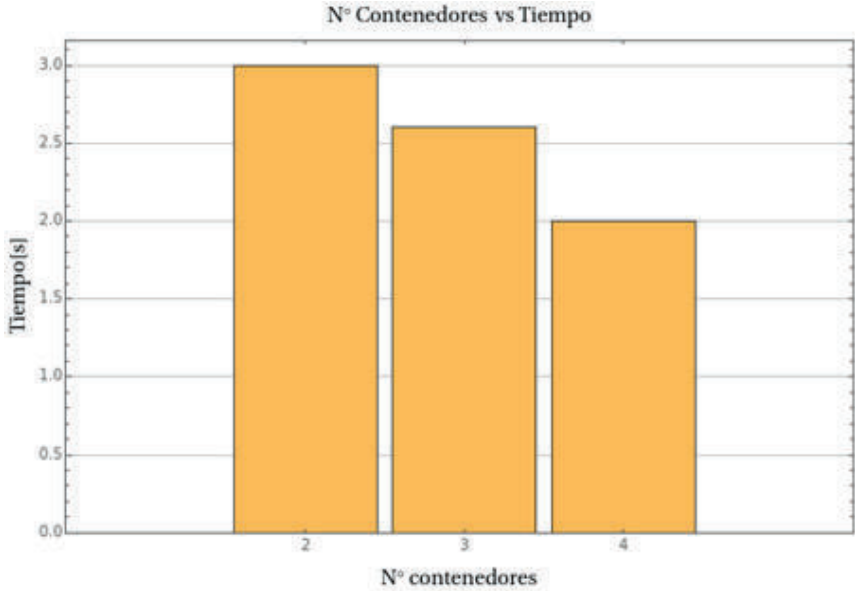
Se utiliza el comando tee para redirigir la salida del servidor a un archivo de registro llamado log.txt. Esto permite visualizar y analizar los resultados del ataque.

V. Evaluación de Resultados y Discusión

5.1. Resultados

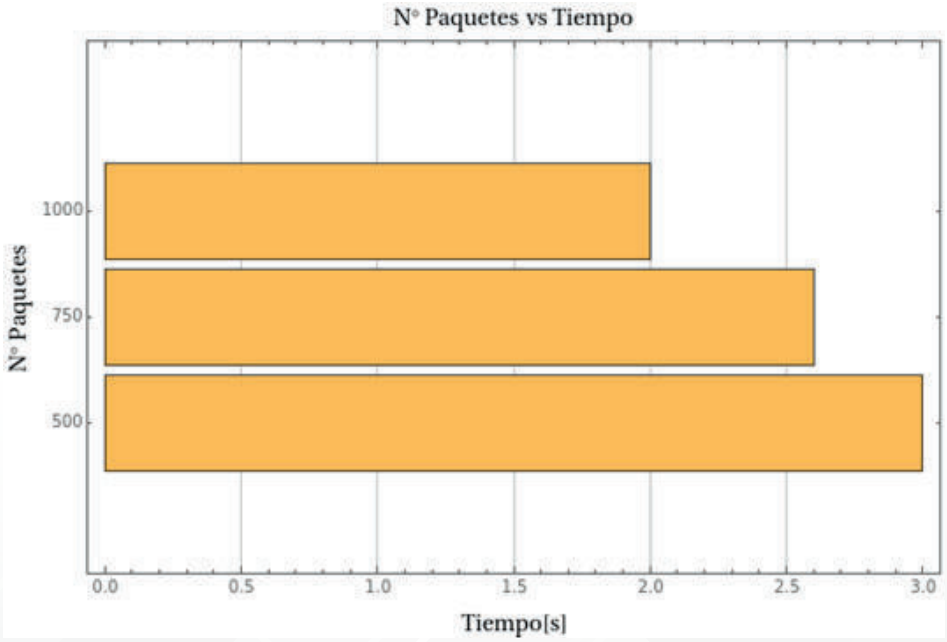
En la Figura 3 se observa el diagrama de barras del número de contenedores docker empleados para realizar el ataque, respecto al tiempo que tarda el servidor en quedar fuera de servicio. Ante el ataque sincronizado de la botnet, se puede apreciar que a medida que aumenta el número de contenedores que se encuentran atacando el servidor, también disminuye el tiempo de forma cuasi lineal. Se debe tomar en cuenta que el aumento mínimo de contenedores ya influye de alta manera al tiempo de ataque al servidor. Esto demuestra que al usar una alta cantidad de contenedores es posible realizar una caída de un servidor sin altas protecciones en tiempos ínfimos.

Figura 3: *Diagrama de barras Docker vs Tiempo*



En la Figura 4 se observa el diagrama de barras del número de paquetes totales enviados hacia el servidor hasta su caída. De la gráfica se puede interpretar que a medida que aumentan el número de paquetes enviados hacia el servidor por nuestra red, también disminuye la cantidad de tiempo necesaria para hacer colapsar al servidor. Es notable observar también como el aumento de contenedores reduce el número de paquetes a utilizar. Esto permite inferir que cuantos más contenedores se ocupen, menos recursos se necesitan enviar al servidor para conseguir su colapso. Del mismo modo, con el uso de menos contenedores, lo recomendable sería utilizar más envíos de paquetes para reducir esta brecha de tiempo.

Figura 4: *Diagrama de barras Docker vs Tiempo*



5.2. Discusión

Los resultados muestran que un mayor número de contenedores Docker utilizados en el ataque DDoS, se correlaciona con un tiempo más corto para que el servidor quede fuera de servicio. Esta relación lineal coincide con la literatura existente. Además, se observa que un aumento en el volumen de paquetes enviados hacia el servidor también reduce el tiempo necesario para que colapse. Estos hallazgos respaldan la comprensión de cómo al manipular la infraestructura de contenedores afecta la eficacia de los ataques DDoS. Sin embargo, el estudio tiene limitaciones y no considera todas las variables posibles.

El uso de contenedores para ataques DDoS presenta serios problemas de seguridad. La capacidad de escalabilidad y fácil distribución de los contenedores permite la rápida creación y despliegue de instancias, facilitando ataques a gran escala desde múltiples ubicaciones y dificultando su mitigación. Aunque los contenedores ofrecen aislamiento, este no es tan fuerte como el de las máquinas virtuales, aumentando el riesgo de escalada de privilegios y el potencial daño, ya que un atacante podría comprometer un contenedor y afectar otros contenedores o al sistema anfitrión.

La utilización de recursos compartidos en los sistemas de contenedores es otra consecuencia importante. En la mayoría de los casos, estos contextos comparten recursos de red, CPU y memoria. Estos recursos pueden ser consumidos de forma desproporcionada por un ataque DDoS realizado desde el interior de un contenedor, lo que podría perjudicar la disponibilidad de otros servicios que operan en el mismo entorno. Esta lucha por los recursos puede provocar un deterioro del rendimiento general del sistema y la interrupción de servicios vitales, intensificando los efectos del ataque y complicando la respuesta y la recuperación.

VI. Conclusiones y Trabajo Futuro

La implementación de una red de botnets utilizando contenedores Docker ha demostrado ser efectiva para emular y evaluar ataques DDoS en un entorno controlado. Esto proporciona una herramienta valiosa para probar métodos de detección y mitigación de ataques en un contexto realista, facilitando así el desarrollo de estrategias de defensa cibernética más sólidas. Asimismo, detectar y mitigar ataques DDoS en entornos de contenedores es desafiante, requiriendo técnicas especializadas y adaptables. La supervisión y detección de actividades anómalas se complican por la naturaleza dinámica de los contenedores, por lo que puede ser necesario desarrollar estrategias de monitoreo específicas. Métodos proactivos como la itinerancia de servidores, que cambian regularmente la ubicación de los servidores, pueden fortalecer la resistencia del sistema al dificultar ataques persistentes.

La correlación observada entre el aumento en el número de contenedores Docker y el volumen de paquetes enviados hacia el servidor con la reducción del tiempo necesario para su colapso, destaca la importancia de comprender y abordar las vulnerabilidades en la infraestructura de red. Esto subraya la necesidad de medidas preventivas y defensivas más robustas en la protección contra ataques DDoS, así como la continua investigación en el desarrollo de tecnologías y métodos de seguridad cibernética.

Finalmente, en sistemas de contenedores, el análisis de big data puede ser útil para detectar y detener ataques DDoS, ya que permite identificar patrones inusuales en el tráfico de red y responder rápidamente. Además, medidas de seguridad estrictas y una adecuada segmentación de la red pueden reducir el impacto de estos ataques. Políticas de control de acceso y aislamiento de contenedores disminuyen la superficie de ataque y evitan la propagación del ataque. Estas medidas, junto con una monitorización proactiva y sofisticada, son esenciales para proteger sistemas basados en contenedores.

Referencias

Acosta-Tejada, D., Sanchez-Galan, J., & Torres-Batista, N. (2023, September). Abordando el Desequilibrio de Datos en Clasificación de ataques de denegación de servicio distribuido (ddos). In Congreso Nacional de Ciencia y Tecnología–APANAC (pp. 117-126).

Barbaro, C. D., & Durante, M. (2022). Mitigación de ataques de denegación de servicio distribuidos en la nube (Doctoral dissertation, Universidad Nacional de La Plata).

Docker Inc. (9 de junio de 2024). Docker docs. Obtenido de Docker overview: https://docs.docker.com/get-started/overview/

Kumari, P., & Jain, A. K. (2023). A comprehensive study of DDoS attacks over IoT network and their countermeasures. *Computers & Security*, 127, 103096.

Lizares Figueroa, P. G., & López Benavides, M. A. (2017). Prevención y detección de ataques de denegación de servicio distribuido (DDOS) implementando el módulo QOS en el servidor web apache.

Macas, M., Wu, C., & Fuertes, W. (2022). A survey on deep learning for cybersecurity: Progress, challenges, and opportunities. *Computer Networks*, 212, 109032. https://doi.org/10.1016/j.comnet.2022.109032

Mirkovic, Jelena; Reiher, Peter (2004). A taxonomy of DDoS attack and DDoS defense mechanisms, 34(2), 39–0.doi:10.1145/997150.997156.

Molano Mendoza, F. E. (2024). Descripción de los ataques distribuidos de denegación de servicios DDoS.

N. Hoque, D. K. Bhattacharyya and J. K. Kalita, “Botnet in DDoS Attacks: Trends and Challenges,” in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2242-2270, Fourthquarter 2015, doi: 10.1109/COMST.2015.2457491.

Narváez, D., Romero, C., & Núñez, M. (2010). Evaluación de ataques de Denegación de servicio DoS y DDoS, y mecanismos de protección. *GEEKS DECC-REPORTS*, 2(1).

Pacheco Manotas, M. (2022). Ataques de denegación de servicio distribuido, Cómo evitarlos y cómo enfrentarlos.

Tamayo Portero, J. O. (2023). Detección de ataques de denegación de servicio activados mediante botnets en redes definidas por software (Master’s thesis, Quito: EPN, 2023.).

Vishwakarma, R., & Jain, A. K. (2019, April). A honeypot with machine learning based detection framework for defending IoT based botnet DDoS attacks. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 1019-1024). IEEE.