Simulación de ataques redes IP en un entorno virtual controlado utilizando Hping3, Scapy y OpenSSL

Simulation of IP Network Attacks in a Controlled Virtual Environment Using hping3, Scapy, and OpenSSL

Marley Morales, Steven Pozo, Erick Ramírez
Departamento de Ciencias de la Computación, Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador.
{jmmorales11, sjpozo1, epramirez2}@espe.edu.ec

Resumen

Los ataques de redes IP son muy comunes dentro de sistemas informáticos y tienen como fin extraer información confidencial y sensible del usuario por parte de actores malintencionados. Este estudio aborda tres tipos de ataques de redes IP: el ataque de denegación de servicios (DoS); la suplantación de identidad, conocida como Spoofing y por último, el Sniffing que se basa en interceptar y escuchar el tráfico de red sin consentimiento del usuario legítimo. Para llevar a cabo la simulación, se emplearon herramientas como Scapy y Hping3, las cuales permiten la inyección y manipulación avanzada de paquetes ICMP y además el replicar escenarios de forma controlada y segura. Como parte de la solución, se desarrollaron métodos de mitigación, con Python para bloquear IPs maliciosas, un programa en C de encriptación de datos para contrarrestar ataques Sniffing. Finalmente, se propuso la configuración y ajustes de red para evitar suplantación de identidad. Como resultado de este estudio, se encontraron estrategias efectivas que aumentan la ciberseguridad ante estos tres tipos de ataques IPs.

Palabras Claves: *DoS, Spoofing, Sniffing, Hping3, Scapy, Ataques a Redes IP.*

Abstract

IP network attacks are very common within computer systems, and they are intended to extract confidential and sensitive user information by malicious actors. This study addresses three types of IP network attacks: denial of service (DoS) attacks, identity theft, known as Spoofing, and finally, Sniffing, which is based on intercepting and listening to network traffic without the consent of the legitimate user. To carry out the simulation, tools such as Scapy and Hping3 were used, which allow the injection and advanced manipulation of ICMP packets and also the replication of scenarios in a controlled and secure manner. As part of the solution, mitigation methods were developed with Python to block malicious IPs, and a C program for data encryption to counter Sniffing attacks. Finally, network configuration and settings were proposed to prevent identity theft. As a result of this study, effective strategies were found to increase cybersecurity against these three types of IP attacks.

Keywords: *DoS, Spoofing, Sniffing, Hping3, Scapy, IP Network Attacks.*

Fecha de Recepción: 18/05/2024 - Aceptado: 21/06/2024 – Publicado: 28/06/2024
ISSN: 2477-9253 – DOI: http://dx.doi.org/10.24133/RCSD.VOL09.N02.2024.03

I. Introducción

Las redes IP son una infraestructura fundamental de comunicación en Internet y redes privadas. El protocolo IP es el que define como se dirigen y enrutan los paquetes para llegar a su destino a través de redes conectadas. Al aprovechar la conectividad a las redes, un dispositivo puede sufrir de un ciberataque si no se aplica una capa de seguridad. Los ciberataques son intentos maliciosos para acceder a sistemas informáticos, redes o datos sin autorización. Estos ataques son realizados con el objetivo de obtener información, interrumpir servicios, o destruir datos (Fernandes, Ciardhuáin, & Antunes, 2024).

Existen diferentes ataques que aprovechan la conectividad y el enrutamiento que proporciona el protocolo IP, tales como, DoS, spoofing y sniffing (Morales , 2016). Estos ataques explotan las vulnerabilidades en redes y sistemas informáticos, cada uno con un objetivo diferente. El DoS o ataque de denegación de servicio es una forma de que el sistema o red se vuelva inaccesible para sus usuarios legítimos. El spoofing es un ataque que se encarga en falsificar las identidades para hacerse pasar por un usuario legítimo. Otro ataque es el Sniffing el cual permite interceptar y analizar el tráfico de la red y capturar información.

Ante este escenario, y con el fin de evitar ser víctima de uno de estos ataques se proponen alternativas de bloqueos IP en tiempo real o por intento de entrada. La alternativa propuesta, se enfoca en mitigar ataques DoS o Fuerza Bruta, como lo menciona Fuertes (2023), estas estrategias pueden combinarse con algún firewall, filtrado de contenido, VPN que ayuden a detectar y bloquear este tipo de ataques antes de que se inunde de paquetes a los servidores o a la red. Los dos ataques se encargan de sobrecargar de solicitudes con masivos envíos de paquetes. En el caso de ataques Sniffing se propone la encriptación de datos con OpenSS, para un envío seguro de paquetes por la red IP.

El resto del articulo ha sido organizado como sigue: En el capítulo 2 se presentan los trabajos relacionados, cuyos autores, han propuesto y generado pruebas similares con base en los ataques de redes IP. En el capítulo 3 se describen los materiales usados en estas simulaciones, así como los métodos implementados en cada tipo de ataque. En el capítulo 4 se muestra la forma que se mitigó cada ataque junto con los resultados obtenidos mediante datos estadísticos. En el capítulo 5, se realiza la discusión y conclusiones en base a las simulaciones de ataques. Finalmente, se presentan los trabajos futuros.

II. Trabajos relacionados

Binbusayyis (2024) propone un modelo que utiliza la combinación aleatoria Ramdom Grove Blend en capas de Multi Layer Perceptron ya que se enfoca en la necesidad de mejorar la detección de ataques en redes. Se utiliza un conjunto de datos UNSW-NB15 y Scapy tool para generar datos en tiempo real. Este sistema alcanza una precisión del 98% para detectar ataques y así se refuerza la seguridad en redes.

Chávez (2011) implementó un escenario simple con servicios de voz, datos y video para realizar ataques DDoS, incluyendo inundaciones TCP SYN y ataques UDP. Utilizó hping por su simplicidad, aunque no logró colapsar toda la red, afectando solo a servicios específicos.

Gonzales et al (2016) proponen evaluar controles en la red para detectar y mitigar ataques Man-in-the-Middle (MITM) mediante el protocolo ARP. Utilizando herramientas de código abierto como Ettercap, los autores demostraron que los atacantes podían capturar información sensible, comprometiendo la confidencialidad de la red. Como resultado, se sugirió implementar mecanismos de detección, como Snort, y configurar características antispoofing en switches y routers para mejorar la seguridad de la red corporativa.

Gorgone Carvajal (2023) desarrolló un laboratorio virtual que simula una red industrial, utilizando GNS3 y herramientas como Scapy y hping3 para realizar simulaciones de ciberataques, aunque enfrentó dificultades con protocolos industriales.

Javanmardi et al. (2024) proponen M-RL el cual es un sistema de detección de intrusiones ligero, y consciente de la movilidad para contrarrestar ataques de inundaciones DDoS UDP en redes loT-Fog. Este sistema no solo detecta ataques DDoS, también aborda los dispositivos loT. M-RL logra una precisión superior al 99%, incluso en la movilidad de los nodos. Este enfoque resulta alta resistencia contra la falsificación de direcciones de origen basadas en software.

Rivera et al. (2020) evaluaron ataques DDoS y de Fuerza Bruta en un entorno virtualizado con Kali Linux, desarrollando un mecanismo de detección y mitigación a nivel de iptables y un Web Application Firewall (WAF), validando la efectividad de sus medidas.

Zapata (2012) propuso evaluar y mitigar ataques a redes IP, enfocándose en el spoofing mediante simulaciones en plataformas como VMware y VirtualBox, logrando bloquear intentos de conexión no autorizados con mecanismos de seguridad efectivos.

Estos trabajos han contextualizado la presente investigación, muestran el estado del arte, han identificado avances previos, y permiten destacar las

III. Materiales y Métodos

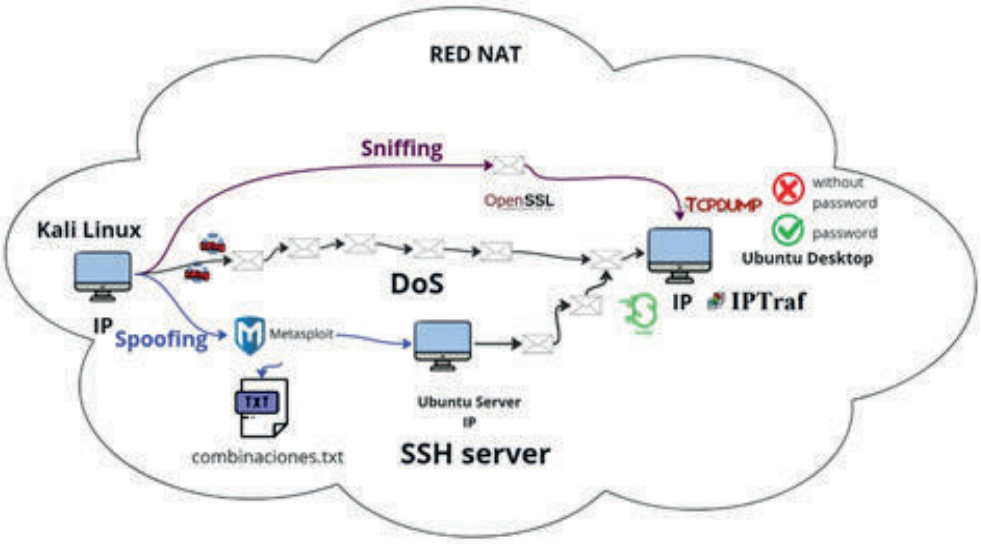
Para explicar esta sección, es conveniente plantear las preguntas de investigación, ya que son necesarias para decidir que herramientas utilizar para mitigar los ataques de DoS, Spoofing y Sniffing:

- RQ1: ¿Cómo proteger los datos que se envían por redes IP para evitar ataques Sniffing?
- RQ2: ¿Cómo bloquear el tráfico que entra a una red IP para evitar un ataque DoS?
- RQ3: ¿Cómo identificar direcciones IP y bloquearlas al intentar acceder a un servicio de SSH mediante ataques de fuerza bruta?

Una vez que se realizó el estado del arte y se identificaron los métodos, técnicas y herramientas para la experimentación, procedemos a diseñar la topología y a configurar los servicios y herramientas necesarias para las pruebas. En la Figura 1, se presenta la topología usada para la simulación de ataques dentro una red NAT con computadores virtualizados, lo cual representa el entorno de red virtualizado, controlado y confiable para llevar a cabo los ataques a redes IP de manera segura.

El entorno virtual consta de máquinas virtuales con Ubuntu Desktop, Ubuntu Server y Kali Linux. Los equipos están conectados mediante una red NAT, misma que permite la conexión entre los nodos y el servicio de acceso a Internet que le provee el host anfitrión. A continuación se explican los diferentes ataques perpetrados a redes IP, propuestos en esta investigación:

Figura 1: *Topología experimentación de ataques y herramientas utilizadas*

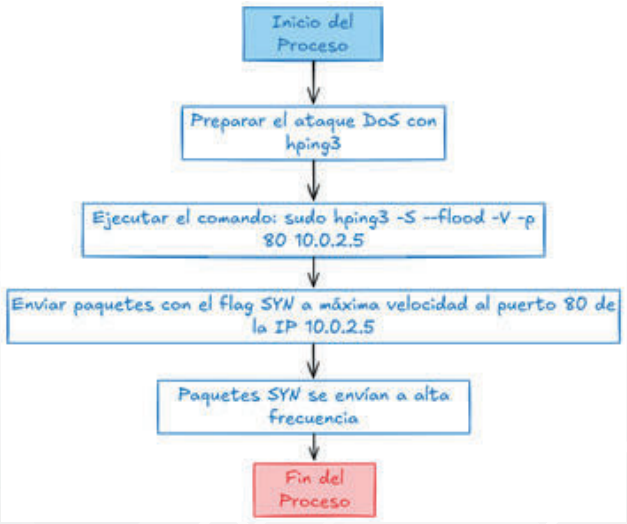


3.1. Ataques DoS/DDoS usando Hping3

Los ataques DoS (Denegación de Servicios) buscan desestabilizar un servicio o recurso de red, interrumpiendo su funcionamiento para los usuarios legítimos. Estos ataques inundarán un servidor con múltiples solicitudes, sobrecargando su capacidad y explotando vulnerabilidades en protocolos como TCP, UDP o ICMP. Para llevar a cabo estos ataques, se utiliza la herramienta Hping3, que permite personalizar el envío de paquetes. En un entorno controlado de máquinas virtuales, se empleó el comando hping3-S–flood-V-p 80 10.0.2.5 para enviar paquetes con el flag SYN a alta velocidad.

El flujo del ataque DoS con hping3 implica la creación y envío masivo de paquetes para saturar el servicio objetivo. El proceso comienza con la preparación del ataque, seguido del lanzamiento donde se configuran los parámetros y el puerto de destino. Este enfoque permite simular el ataque de manera segura y efectiva, como se ilustra en el flujograma del ataque DoS presentado en la Figura 2.

Figura 2: *Ataque DoS a Ubuntu Desktop*

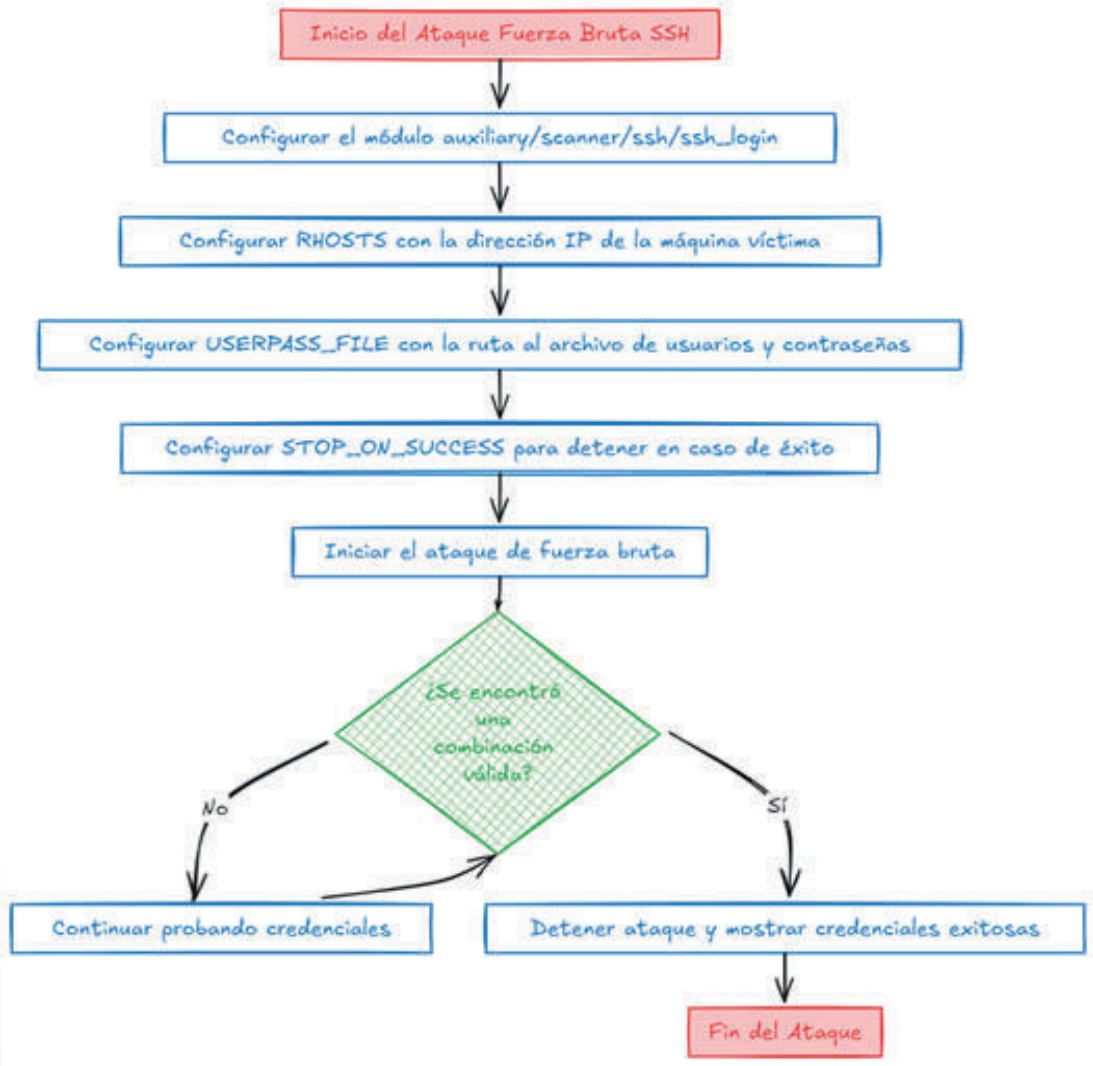


Cabe señalar que un ataque DoS/DDoS se puede implementar a través de Metasploit framework, que es una suite de herramientas enfocadas en la prueba de vulnerabilidades de cualquier sistema. Dentro de Kali Linux se tiene ya instalado esta herramienta. Para esto se emplean módulos relacionados con DoS Apache en Metasploit – framework. El total de módulos disponibles encontrados fueron de 216. Sin embargo, utilizaremos "dos/http/apache_range_dos".

3.2. Ataques de fuerza bruta a un servidor de SSH

Los ataques de Fuerza Bruta son aquellos que intentan ingresar a un servicio ya sea HTTP, SHH, Mail con usuarios y contraseñas que se cree que son las credenciales válidas. Para realizar estos ataques utilizaremos los módulos de Metasploit Framework, específicamente, los módulos para los servicios de SSH (Secure Shell) junto con un archivo .txt donde se alojará todos los usuarios y contraseñas de prueba para ingresar al servicio de SSH de la máquina virtual de Ubuntu Server, como se visualizó en la Figura 1. La Figura 3 muestra el flujograma para configurar y realizar el ataque de fuerza bruta al servidor de SSH en la máquina de Ubuntu Server, así mismo la configuración del módulo de ssh_login en Metasploit:

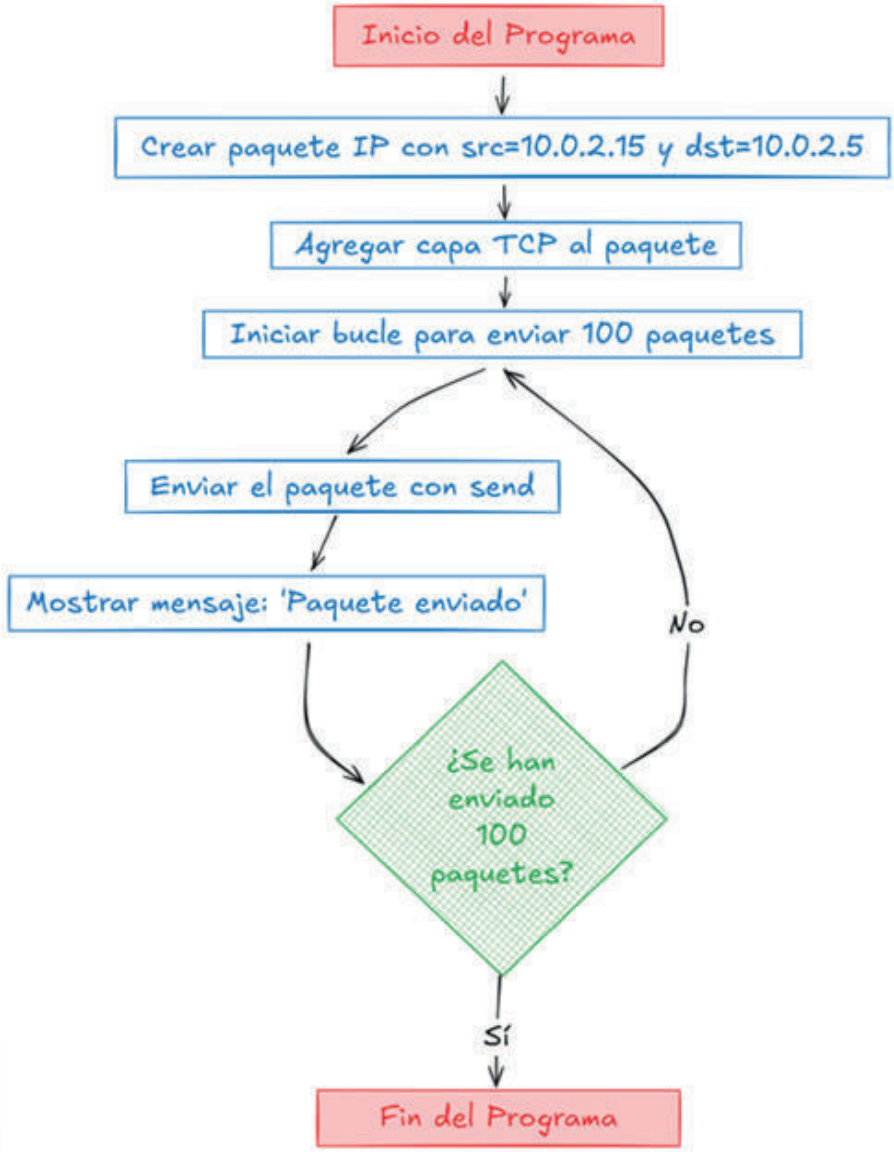
Figura 3: *Flujograma del proceso de ataque de fuerza bruta SSH con Metasploit*



3.3. Ataques Spoofing usando Scapy

Los ataques Spoofing son una técnica en la cual se suplanta la identidad de una red IP para atacar a otra máquina haciendo parecer que provienen de otra fuente. Para cumplir este objetivo se debe conocer la dirección IP de la máquina objetivo. En la Figura 4, se muestra el flujograma de cómo se implementó el algoritmo para realizar el ataque de suplantación de IP en el cual se envían 100 paquetes a la dirección IP objetivo, que en este caso es la máquina virtual de Ubuntu Server.

Figura 4: *Flujograma del programa realizado en Python que permite ejecutar un ataque Spoofing*



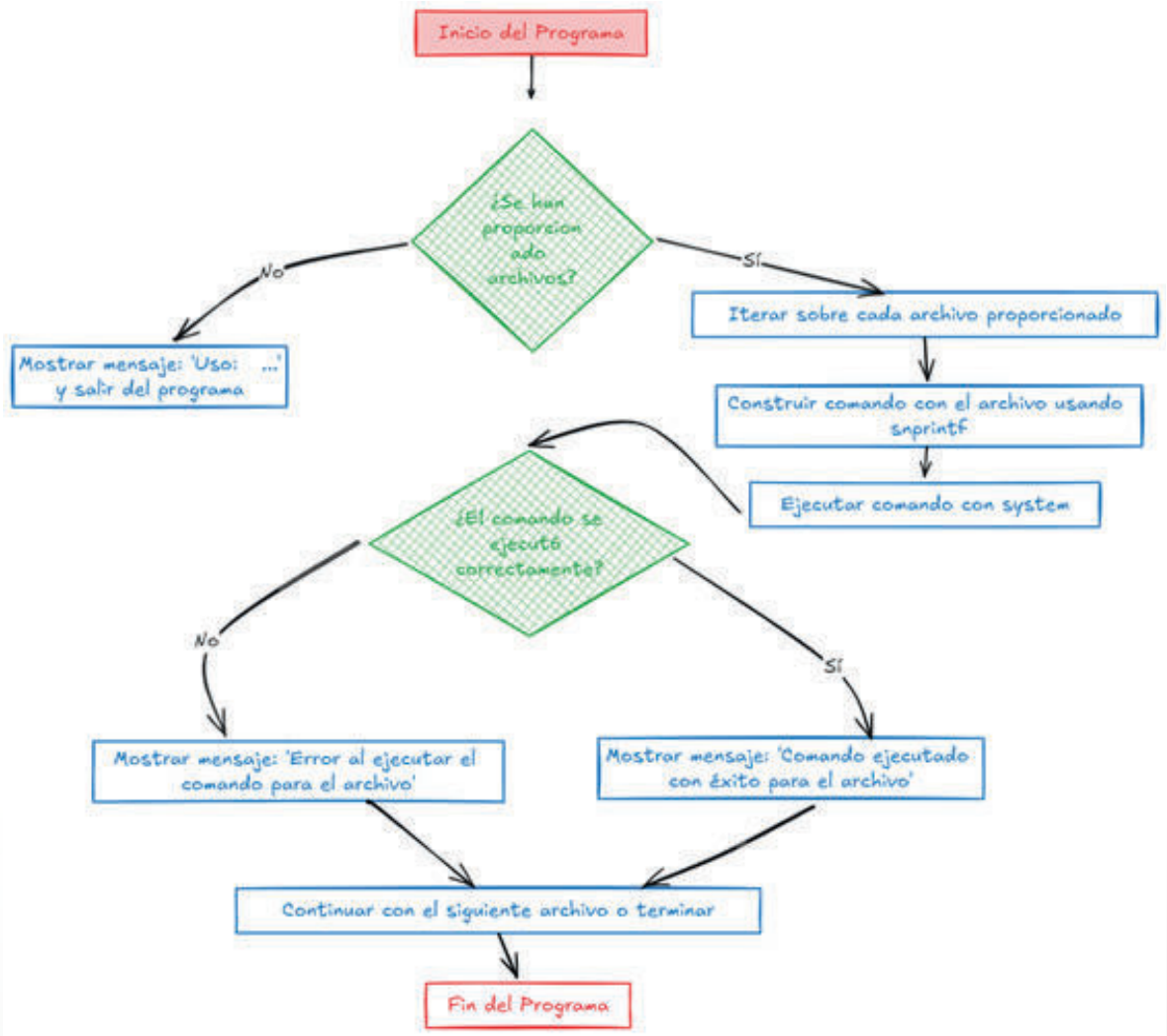
3.4. Ataques Sniffing usando Hping3

Los ataques de Sniffing consisten en la intercepción de paquetes de forma ilegal, con el fin de capturar esos datos para después suplantarlos o robarlos. Tal como se menciona Anú et al (2018), este tipo de ataques se lo realiza principalmente en la capa de enlace de datos del modelo OSI, siendo este el punto clave para capturar

paquetes. En este caso para realizar el ataque de Sniffing simulado dentro de la red NAT creada, se usará herramientas necesarias para el envió, captura y análisis de datos.

La herramienta que se usó para realizar el ataque fue Hping3, el cual permite manipular paquetes TCP/IP. En la simulación se hizo uso de paquetes ICMP personalizados, como es el caso de un archivo de texto con contenido desencriptado. Por otro lado, para capturar los paquetes en la red IP, se usó la tcpdump, el cual permite capturar paquetes de red en tiempo real, siendo el punto clave para capturar el archivo de texto con la información desencriptada que se envió por la red IP. Por último, se analizó el paquete capturado usando Wireshark, permitiendo observar la data que se ha enviado desde Kali Linux hacia Ubuntu desktop. Para efectuar el ataque de Hping3, se creó un programa en C, el cual contiene la estructura lógica de como recibe los archivos planos para ser enviados a la ip 10.0.2.5 correspondiente a la máquina atacada de Ubuntu desktop. En la Figura 5. Se muestra el flujograma que permite realizar el enviar el paquete.

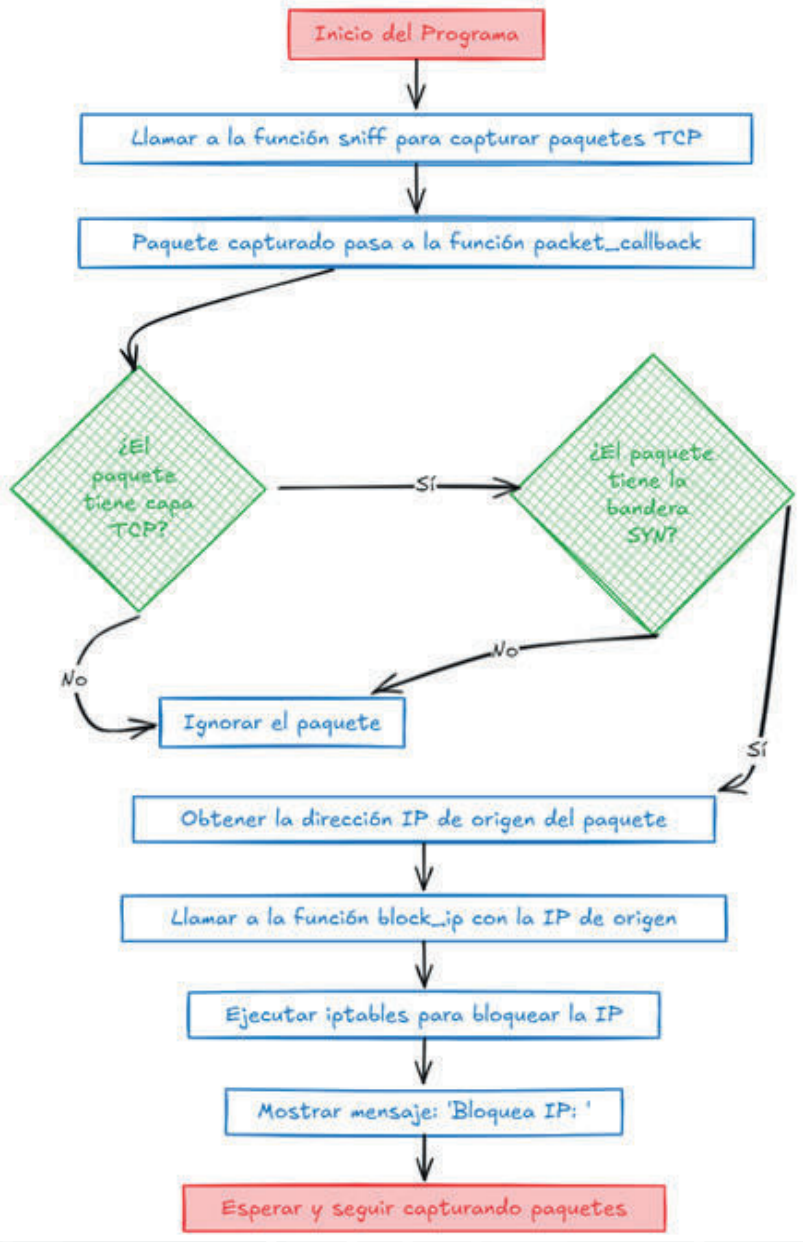
Figura 5: *Flujograma del programa en C para enviar paquetes con Hping3 desde Kali Linux hacia Ubuntu Desktop*



3.5. Implementación para la prevención de DoS mediante Python

Luego de la simulación en el entorno controlado, para prevenir este ataque, se desarrolló un programa en Python haciendo uso de la librería Scapy. El objetivo fue capturar paquetes TCP de la red y bloquear las direcciones IP que envían solicitudes con el flag SYN. La librería Scapy esta creada para interactuar directamente con los paquetes de la red, y con IPtable se bloquea los paquetes. En la Figura 6 se muestra el flujograma del programa en Python que se ejecuta en Ubuntu Desktop como una capa para bloquear las solicitudes flag SYN. La finalidad del programa es que en tiempo real capture los paquetes entrantes, y de esta manera los bloquea, para que el servidor local de la máquina que está siendo atacada no colapse.

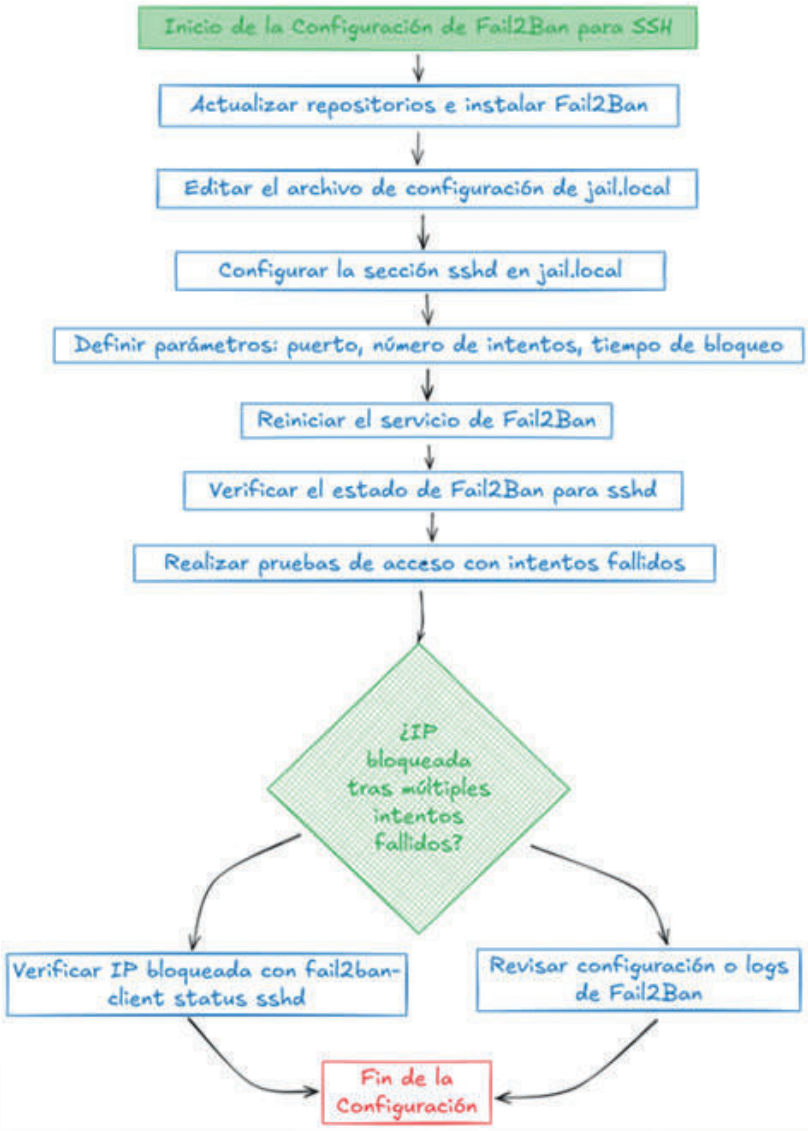
Figura 6: *Flujograma de bloqueo de ataques DoS utilizando Scapy para captura de paquetes malignos y bloqueo de IPs*



3.6. Prevención de ataques de Fuerza Bruta a un servidor de SSH

Previamente en la sección 3.2 se explicó cómo realizar un ataque de fuerza bruta usando el módulo "ssh_login" de Metasploit. Ahora se explicará cómo prevenir este tipo de ataques en el servidor mediante la herramienta fail2ban. Fail2ban monitorea archivos de registro como /var/log/auth.log y bloquea las direcciones IP que han realizado demasiados intentos fallidos de inicio de sesión en SSH. En la Figura 7 se muestra el proceso para comprender de mejor forma como configurar e implementar un baneo o bloqueo de las redes IP que intentan ingresar al sistema Ubuntu Server.

Figura 7: *Flujograma que muestra la instalación e implementación de Fail2ban para proteger SSH en una máquina virtual.*

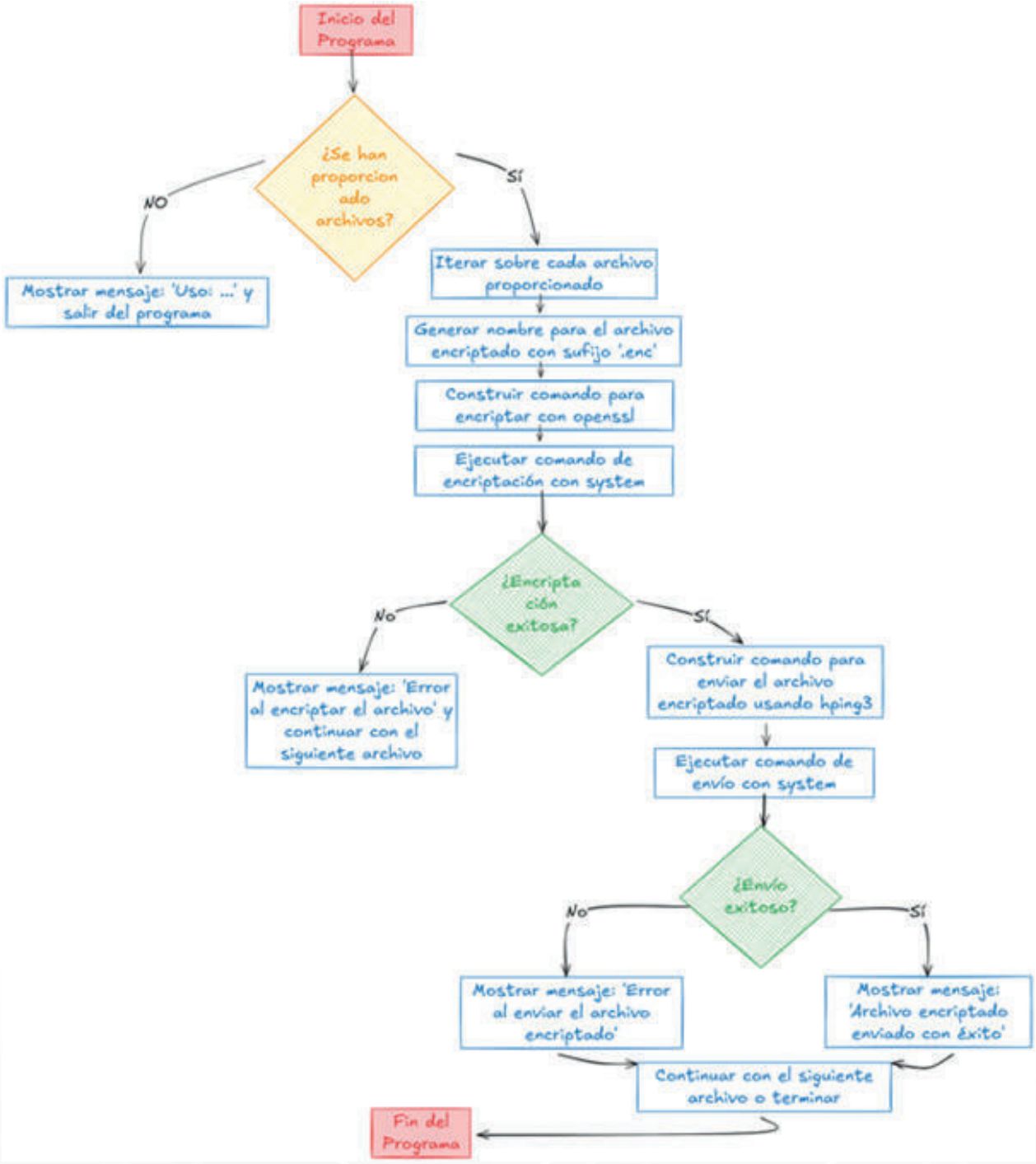


3.7. Implementación para la prevención de ataques Sniffing con C

Para prevenir el Sniffing se utilizó OpenSSL que permitió encriptar los archivos de texto, los cuales fueron enviados mediante la red. Este proceso de ataque realizado con hping3, es el mismo mostrado en la sección III Materiales y Métodos, con la diferencia que ahora los datos enviados son seguros. Para lograr esto se creó

un programa que, en C, que permite llevar a cabo todo el proceso de ataque, envío y encriptación de datos. A continuación, en la Figura 8 se muestra el flujograma del algoritmo implementado.

Figura 8: *Flujograma de agregación y envío de archivos encriptados con OpenSSL en simulación de ataque con Hping3*



IV. Evaluación de Resultados

4.1. Resultados

En este apartado se presentan los resultados obtenidos tras la ejecución de ataques de Fuerza Bruta, Sniffing y DoS, cada uno de ellos en un ambiente controlado. Las estrategias que se emplearon para mitigar sus efectos han sido evaluadas para conocer la efectividad de las medidas implementadas. Los resultados se presentan en función de las preguntas de investigación planteadas en este proyecto:

RQ1: ¿Cómo proteger los datos que se envían por redes IP para evitar ataques Sniffing?

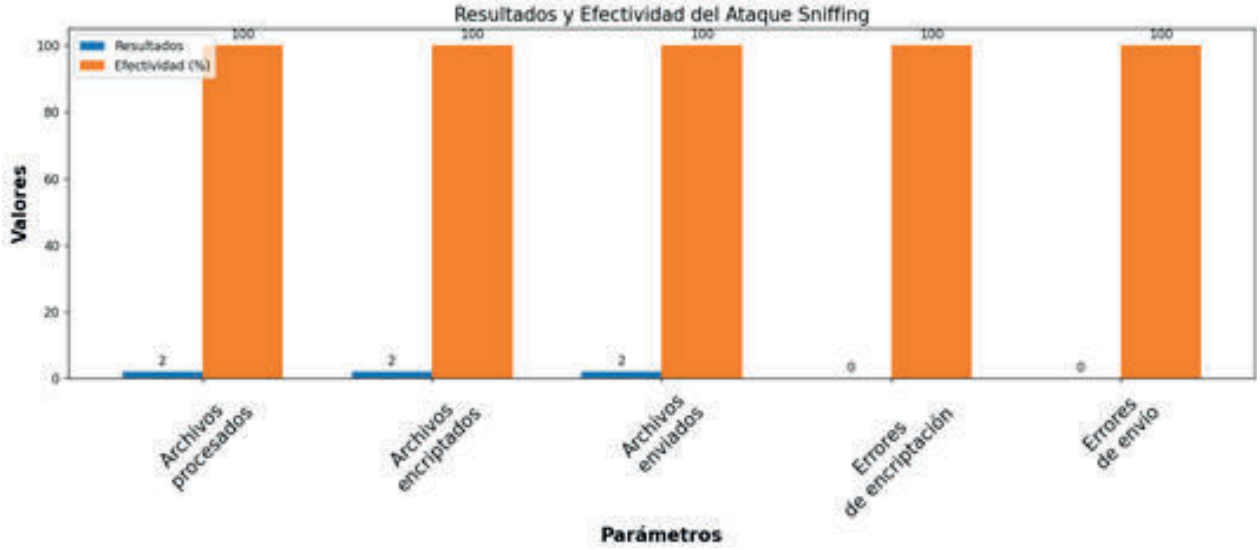
Para proteger los datos que se envían por una red IP, es importante encriptarlas para asegurar su información. Al usar OpenSSL, estos archivos adquieren seguridad al enviarse y al interceptarse. Por ello, la efectividad mostrada en el ataque como en la transmisión es de un 100% en cada parámetro, ya que, al encriptar los datos se ha demostrado que la seguridad de los paquetes enviados es relativamente alta. En la Tabla 1, se muestra los resultados del ataque y las estadísticas de transmisión.

Tabla 1: *Parámetros y resultados de efectividad en el ataque con Hping3 y encriptación de archivos con OpenSSL*

Resultado del ataque Sniffing en el proceso de envío.	**Parámetros**	**Resultados**	**Efectividad**
	Archivos procesados	2	100%
	Archivos encriptados	2	100%
	Archivos enviados	2	100%
	Errores de encriptación	0	100%
	Errores de envío	0	100%
Estadística de transmisión	**Parámetros**	**Resultados**	**Efectividad**
	Paquetes enviados	2	100%
	Paquetes exitosos	2	100%
	Tiempo de transmisión (segundos)	0	
	Datos enviados (bytes)	360	100%
	Tasa de transmisión de datos (bytes/segundo)	1135646.69	100%

Como se observa en la Tabla 1, se trabajó con dos archivos de texto para realizar el ataque, cada uno con información única. El primer archivo contiene datos sensibles. Dichos datos corresponden a un usuario, contraseña y número de tarjeta de crédito, y el segundo es un archivo con un mensaje común y corriente, los cuales no contaban con seguridad. En la figura 9, se observa que la efectividad en los 5 parámetros descritos en el proceso del ataque fue exitosa en su totalidad.

Figura 9: *Gráfica estadística que muestra la efectividad en relación con los archivos manipulados durante el proceso*



RQ2: ¿Cómo bloquear el tráfico que entra a una red IP para evitar un ataque DoS?

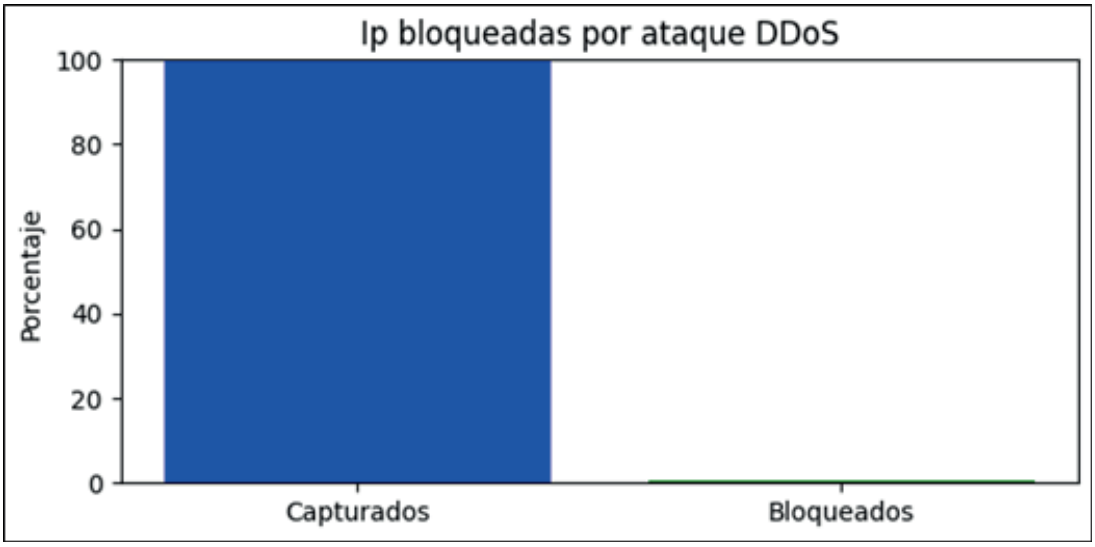
Como una solución para los ataques DoS se desarrolló un programa en Python para bloquear las IP que están causando el ataque. Al evaluar si es factible esta solución se encontró la diferencia de más del 99% causando que no sea efectivo. En la Tabla 2 se detallan los parámetros de evaluación en base a dos resultados con dos diferentes rangos de tiempo. En el resultado 1 se muestra una diferencia del 99.77% y en el resultado 2 una diferencia del 99.56% entre paquetes capturados y bloqueados.

Tabla 2: *Parámetros de evaluación para validar los paquetes capturados bloqueados.*

Parámetros	Resultados 1	Resultados 2
Tiempo	30822594.64	24759000
Cantidad de paquetes capturados	272690	82040
Cantidad de paquetes capturados y bloqueados	640	361

La solución que se planteó arrojó resultados no adecuados, debido a la estructura que se maneja en el programa en Python, pues el programa captura un paquete y bloquea ese paquete y mientras realiza ese proceso el resto de los paquetes siguen ingresando y no se están bloqueando. En la Figura 10 se observa la diferencia de la cantidad de paquetes capturados, pero no bloqueados con la cantidad de paquetes que si fueron bloqueados.

Figura 10: *Comparativa de la cantidad de paquetes capturados y bloqueados*



RQ3: ¿Cómo identificar direcciones IP y bloquearlas al intentar acceder a un servicio de SSH mediante ataques de fuerza bruta?

Para evitar los ataques de fuerza se investigó fail2ban que proporciona jaulas que protegen los servicios dentro del servidor. La efectividad de esta herramienta es del 100% a los parámetros que se ha establecido en cuanto de tiempo de bloqueo de la IP atacante, el periodo de tiempo para observar los eventos que ocurren y el número máximo de intentos fallidos.

Al realizar los ataques de fuerza bruta al servidor de SSH en la máquina de Ubuntu Server, se empleó el parámetro VERBOSE dentro del módulo ssh_login de Metasploit framework. Esto ayudó a obtener qué credenciales fueron exitosas y cuales no, así como también ver el estado de la conexión al servicio de SSH como se muestra en la Tabla 3.

Tabla 3: *Resultados del ataque de fuerza bruta en términos de éxito, errores y advertencias en el primer intento.*

Estado o evento	**Resultado o mensaje**
Éxito	Se ha escaneado 1 host (máquina Ubuntu Server) al 100%
Error	Fallaron 2 credenciales: - admin:admin - admin:1234
	No se ha establecido la conexión al servidor 3 veces
Advertencias	Sin base de datos activa para guardar las credenciales

Luego del primer intento, la jaula establecida en fail2ban bloquea la IP porque la máquina atacante ya ha sobrepasado el número límite de intentos de acceso al servidor SSH configurado en cinco (5), cuyos resultados se muestran en la Tabla 4.

Tabla 3: *Resultados del ataque en el segundo intento*

Estado o evento	Resultado o mensaje
Éxito	Se ha escaneado 1 host (máquina Ubuntu Server) al 100%
Error	Falló la conexión al servidor por 3 veces que se intentó.
Advertencias	Sin base de datos activa para guardar las credenciales

En la Figura 11 se ilustra la efectividad que tiene fail2ban tanto en el primer intento como en el segundo intento al realizar el ataque por fuerza Bruta que se mostró anteriormente.

Figura 11: *Efectividad de la herramienta fail2ban en los ataques de fuerza bruta*

4.2. Discusión

En relación a los ataques de Fuerza Bruta, realizados con Metasploit framework fue exitoso. Esto debido a que los parámetros se pueden configurar dentro del módulo como el host, el archivo para usuarios y contraseñas, el tiempo que tomará el script por cada intento, etc. Esto ayuda a que el ataque sea mucho más personalizado y adaptado a las características que tiene el servidor víctima. Si bien fue exitoso el ataque, implementar mecanismos de seguridad como fail2ban en el servidor, mitigó por completo el riesgo porque se establecieron parámetros como el tiempo de bloqueo y el número de intentos de 5.

En lo que concierne a los Ataques de DoS, se utilizó un programa en Python, desarrollado con la librería Scapy, para capturar paquetes TCP y bloquear direcciones IP que enviaban solicitudes con el flag SYN. Este enfoque fue exitoso, ya que el programa logró identificar y bloquear en tiempo real las solicitudes maliciosas, evitando que el servidor local de la máquina atacada colapsara. Además, la integración con IPtable permitió gestionar eficazmente el tráfico no deseado, lo que fortaleció la seguridad del sistema.

En relación a los ataques Sniffing los resultados fueron satisfactorios, ya que el uso de hping3 como herramienta para enviar los paquetes ICMP permitió realizar el ataque de manera exitosa. Así también, tcpdump

permitió capturar correctamente los paquetes provenientes de la máquina atacante, al igual que Wireshark para analizar el contenido de cada paquete. Al implementar OpenSSL en la encriptación de datos, permite generar una clave, el cual, en este caso fue “1234”. Con esta clave se puede desencriptar, pero, solo con el usuario de Ubuntu Desktop. Esto se logra gracias a la transformación de los hexadecimales a números binarios y usando OpenSSL y para desencriptar dichos datos, se usa OpenSSL para desencriptación, obtenido el contenido original.

V. Conclusiones y Trabajo Futuro

La investigación demuestra que la simulación de ataques a redes IP en un entorno virtual controlado es eficaz para evaluar su seguridad, y que las medidas de mitigación implementadas, como el uso de OpenSSL para encriptar los datos y evitar ataques de Sniffing, resultan confiables. Se utilizó C como lenguaje de programación debido a su versatilidad y robustez. Además, la simulación de ataques DoS permitió identificar y bloquear solicitudes maliciosas, protegiendo la disponibilidad de los servicios sin afectar su rendimiento, mientras que la implementación de fail2ban en Ubuntu Server bloqueó automáticamente las IP con intentos fallidos de inicio de sesión, reforzando la protección contra ataques de fuerza bruta en SSH.

Como trabajo futuro se plantea la integración de inteligencia artificial en la detección de intrusiones y la automatización de respuestas ante ataques, así como la evaluación de nuevas tecnologías emergentes en la ciberseguridad, como la seguridad en entornos de Internet de las cosas (IoT).

Referencias

Anu, P., & Vimala, S. (2018). A survey on sniffing attacks on computer networks. Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017, 2018-January, 1–5. https://doi.org/10.1109/I2C2.2017.8321914

Augusto, C., & Agudelo, R. Arquitectura para automatizar respuesta a incidentes de seguridad de la información relacionados con ataques internos mediante la ejecución de técnicas spoofing.

Binbusayyis, A. (2024). Reinforcing network security: Network attack detection using random grove blend in weighted MLP layers. *Mathematics*, 12(11), 1720. https://doi.org/10.3390/MATH12111720

Chávez, J. (2011, October). Simulación y análisis de mecanismos de defensa ante los ataques de denegación de servicios (DoS) en redes de área local convergentes. *Retrieved September* 3, 2024, http://bibdigital.epn.edu.ec/handle/15000/4282

Fernandes, P., Ciardhuáin, S., & Antunes, M. (2024). Unveiling malicious network flows using Benford’s law. *Mathematics,* 12(15), 2299. https://doi.org/10.3390/MATH12152299

Fuertes Díaz, W. M., & Macas Carrasco, M. A. (2023). Ciberseguridad [Tesis]. Universidad de las Fuerzas Armadas-ESPE. http://repositorio.espe.edu.ec/handle/21000/36481

Gorgone Carvajal, A. (2023). Simulación de ciber-ataques con GNS3 para validar la robustez de protocolos industriales. *Retrieved September* 3, 2024, https://hdl.handle.net/10630/27491

González, D. A., Pérez, M. E. G., & Bernal, L. P. (2016). Detección y mitigación de ataques ARP en la red corporativa de la división territorial holguín, ETECSA. *Telemática,* 15(1), 62–68. Retrieved September 3, 2024, https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/216

Javanmardi, S., Ghahramani, M., Shojafar, M., Alazab, M., & Caruso, A. M. (2024). M-RL: A mobility and impersonation-aware IDS for DDoS UDP flooding attacks in IoT-Fog networks. *Computers & Security*, 140, 103778. https://doi.org/10.1016/j.cose.2024.103778

Morales Jeria, M. A. (2016). Plataforma experimental de ciberseguridad sobre infraestructura virtualizada para mitigar los ataques de denegación de servicio. Retrieved September 3, 2024, http://repositorio.espe.edu.ec/jspui/handle/21000/11609

Rivera, E., Cárdenas, M., & Chiriboga, W. (2020, April). Evaluación de ataques DDoS y fuerza bruta utilizando entorno virtual Kali Linux como plataforma experimental. *Dilemas contemporáneos: Educación, Política y Valores*. https://doi.org/10.46377/DILEMAS.V35I1.2248

Zapata, L. (2012, December). Evaluación y mitigación de ataques reales a redes IP utilizando tecnologías de virtualización de libre distribución. *Ingenius,* (8), 11–19. https://doi.org/10.17163/INGS.N8.2012.02