PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

Trabajo de Titulación

# ANÁLISIS DE TRÁFICO DE RED PARA LA DETECCIÓN Y DEFENSA DE ATAQUES DE FUERZA BRUTA CON PYTHON.

AUTOR:

GABRIEL IVAN CÓRDOVA SOSA

DOCENTE DIRECTOR:

HENRY ROA

QUITO DM, JUNIO DE 2024

**INDICE**

Índice de Figuras ..... 5

Índice de Tablas ..... 6

CAPITULO I: INTRODUCCIÓN ..... 7

1.1 TEMA ..... 7

1.2. JUSTIFICACIÓN ..... 7

1.3. PLANTEAMIENTO DEL PROBLEMA ..... 7

1.4. OBJETIVOS ..... 8

1.4.1 GENERAL ..... 8

1.4.2 ESPÉCIFICOS ..... 8

1.5. ALCANCE ..... 9

CAPÍTULO II: MARCO TEÓRICO ..... 10

2.1. Revisión de la literatura ..... 10

2.2. Seguridad en sistemas y redes informáticas ..... 11

2.3. Ataques de fuerza bruta ..... 15

2.4. Herramientas y técnicas de análisis de tráfico de red ..... 18

2.5. IPTables ..... 27

2.6. Metodologías de Ataque de Fuerza bruta ..... 27

CAPITULO III METODOLOGÍA DE ATAQUE “PLAN DE ATAQUE” ..... 36

3.1. Relevancia y prevalencia del ataque: ..... 36

3.2. Disponibilidad de herramientas y material de referencia: ..... 36

3.3. Complejidad y eficacia del ataque: ..... 36

3.4. Relevancia para el contexto de la investigación: ..... 36

3.5. Entorno de prueba ..... 37

3.5.1. Instalación Virtual Box ..... 37
3.5.2. Instalación Kali Linux ..... 37
3.5.3. Implementación de Máquina Virtual de defensa nube Google ..... 38
CAPITULO IV PROPUESTA DE ATAQUE Y DEFENSA APLICADA AL PROTOCOLO FTP ..... 41
4.1 Ejecución del ataque de fuerza bruta ..... 41
4.1.1 Análisis de Puertos ..... 41
4.1.2. Escaneo inicial de puertos ..... 42
4.1.3. Escaneo avanzado de puertos ..... 44
4.1.3.3. Comprensión de la arquitectura del sistema: ..... 48
4.1.4. Ataque de ftp ..... 48
4.2. Metodología de Defensa “Plan de defensa de Ataques” ..... 51
4.2.1. Detección de Ataque de Fuerza Bruta en FTP ..... 52
4.2.2. Mitigación y defensa ..... 57
4.2.3. Desbloqueo de Baneo de Ip ..... 59
4.2.4. Validacion de la Herramientas Utilizadas ..... 61
4.2.5. Beneficios: ..... 66
4.2.6. Limitaciones: ..... 66
4.3. Resultados de la implementación del plan de ataque y defensa ..... 68
4.3.1. Ataque realizado ..... 68
4.3.2. Acción implementada ..... 68
4.3.3. Resultado de la acción implementada ..... 68
4.3.4. Implementación de medidas de defensa ..... 68
4.3.5. Resultado de la acción implementada ..... 69

4.3.6. Protección de servicios críticos: ..... 69
4.3.7. Análisis forense: ..... 69
Conclusiones y Recomendaciones ..... 71
Conclusiones ..... 71
Recomendaciones ..... 72
Bibliografía ..... 74

## Índice de Figuras

Figura 1. Riesgo ..... 13

Figura 2. Monitorización Activa ..... 25

Figura 3 Monitorización Pasiva ..... 26

Figura 4. Herramienta Hydra ..... 29

Figura 5. Herramienta Hydra ..... 30

Figura 6. Herramienta Medusa ..... 31

Figura 7. Ncrack ..... 32

Figura 14. Odoo Comunitario ..... 35

Figura 8. Virtual Box ..... 37

Figura 9. Kali Linux ..... 38

Figura 10. Virtualización de Pruebas Google Cloud ..... 39

Figura 11. Creación de instancia de Pruebas Google Cloud ..... 39

Figura 12. Instancia de Pruebas Google Cloud ..... 40

Figura 13. crear instancia de Pruebas Google Cloud ..... 40

Figura 15. escaneo nmap ..... 42

Figura 16 Escaneo Avanzado de puertos ..... 44

Figura 17 Escáner de Puertos ..... 46

Figura 18 ubicación de diccionarios de ataque Kali ..... 48

Figura 19 Proceso de Ataque FTP ..... 49

Figura 20 Script Python de Detección de Intrusos ..... 52

Figura 21. Análisis de Red ..... 57

Figura 22 ..... 57

Figura 23 Desbloquear Baneo ..... 59

## Índice de Tablas

Tabla 1. Métodos de ataques de Fuerza bruta ..... 28

Tabla 2. Herramientas de Fuerza Bruta ..... 33

Tabla 3 ..... 61

Tabla 4. Eficiencia ..... 62

Tabla 5. Integracion con otras soluciones ..... 63

Tabla 6. Recursos consumidos ..... 65

# CAPITULO I: INTRODUCCIÓN

## 1.1 TEMA

Análisis de tráfico de red para la detección y defensa de ataques de fuerza bruta con Python.

## 1.2. JUSTIFICACIÓN

Dada la creciente necesidad de proteger los sistemas y redes informáticas de ataques cibernéticos, los ataques de fuerza bruta se han convertido en una amenaza cada vez más preocupante. Estos ataques, que consisten en intentar adivinar contraseñas mediante la prueba de múltiples combinaciones, pueden causar daños significativos a los sistemas y la información almacenada. La vulnerabilidad de los sistemas a estos ataques destaca la importancia de desarrollar soluciones eficaces para su detección y mitigación.

El objetivo de este trabajo de titulación es identificar, analizar y defender contra los ataques de fuerza bruta utilizando el lenguaje de programación Python y técnicas de análisis de tráfico de red. La herramienta desarrollada permitirá identificar, analizarlos y crear instrucciones por medio de firewall para detener la amenaza, lo que mejorará significativamente la seguridad de los sistemas y redes.

Además, este trabajo de titulación busca contribuir al campo de la seguridad informática al proponer un enfoque innovador para la detección y defensa de ataques de fuerza bruta. La utilización de técnicas de análisis de tráfico de red y el lenguaje de programación Python permitirán desarrollar una solución más eficiente y efectiva en la prevención de estos ataques, lo que puede tener un impacto positivo en la protección de sistemas y datos críticos.

## 1.3. PLANTEAMIENTO DEL PROBLEMA

A nivel mundial las organizaciones y sistemas informáticos enfrentan constantes amenazas de seguridad, siendo los ataques de fuerza bruta uno de los más preocupantes. Estos ataques, que consisten en el intento de adivinar contraseñas de manera sistemática, pueden causar graves daños al comprometer la integridad de los sistemas y la confidencialidad de la información.

Si bien existen soluciones de seguridad para prevenir este tipo de ataques, como firewalls, sistemas de detección de intrusos y controles de acceso, estas suelen ser ineficaces o insuficientes ante la creciente sofisticación de las técnicas utilizadas por los ciber delincuentes.

El problema central radica en la necesidad de desarrollar una herramienta efectiva y eficiente que permita detectar, prevenir y mitigar los ataques de fuerza bruta en tiempo real. Dicha herramienta debe ser capaz de analizar el tráfico de red, identificar patrones de actividad sospechosa y aplicar contramedidas oportunas para proteger los sistemas.

Por lo tanto, el problema a resolver en este trabajo de titulación es:

Una solución de seguridad basada en análisis de tráfico de red para detectar, prevenir y mitigar ataques de fuerza bruta contra sistemas y redes informáticas podría diseñarse e implementarse mediante la monitorización del tráfico de red para identificar patrones anómalos como intentos de inicio de sesión fallidos a gran escala o aumentos repentinos del volumen de tráfico, implementando mecanismos de prevención y mitigación como sistemas de detección y prevención de intrusos, autenticación de dos factores y políticas de contraseñas robustas, realizando análisis forenses de los incidentes detectados para mejorar continuamente los algoritmos de detección, y automatizando los procesos de detección, alerta y respuesta ante ataques para una mayor eficiencia y rapidez de reacción.

## 1.4.OBJETIVOS

### 1.4.1 GENERAL

Analizar el tráfico de red mediante Wireshark para la detección y defensa de ataques de fuerza bruta utilizando Python.

### 1.4.2 ESPÉCIFICOS

- Analizar las técnicas de ataque de fuerza bruta y sus implicaciones en la seguridad de los sistemas informáticos, así como investigar y evaluar las herramientas de análisis de tráfico de red y sus capacidades para la detección de este tipo de ataques.
- Diseñar e implementar un prototipo de solución utilizando Wireshark y Python para la detección y mitigación de ataques de fuerza bruta, desarrollando algoritmos y técnicas de análisis de tráfico de red que permitan identificar patrones de comportamiento malicioso.
- Evaluar la eficacia y eficiencia de la solución propuesta mediante pruebas y validación en un entorno controlado, analizando los beneficios y limitaciones de la herramienta implementada.

## 1.5. ALCANCE

Este trabajo de investigación se enfocará en el desarrollo de una herramienta de detección y mitigación de ataques de fuerza bruta contra sistemas y redes informáticas, utilizando técnicas de análisis de tráfico de red y el lenguaje de programación Python.

Este se limitará al análisis de las técnicas de ataque de fuerza bruta y sus implicaciones en la seguridad informática, así como la evaluación de herramientas de análisis de tráfico de red para la detección de este tipo de ataques.

Adicional a esto se desarrollará el diseño e implementación de un prototipo de solución utilizando Wireshark y Python, que permita la identificación de patrones de tráfico de red asociados a ataques de fuerza bruta.

También se generará una evaluación de la eficacia y eficiencia de la herramienta desarrollada mediante pruebas en un entorno controlado, midiendo indicadores como la tasa de detección, el tiempo de respuesta y la precisión de las alertas.

Es importante destacar que esta investigación no abarcará la implementación de mecanismos de prevención o mitigación automatizada de los ataques de fuerza bruta, sino que se enfocará en la detección y análisis de este tipo de amenazas. Tampoco se incluirá el despliegue de la herramienta en entornos productivos fuera del entorno de pruebas controlado.

# CAPÍTULO II: MARCO TEÓRICO

## 2.1.Revisión de la literatura

Este trabajo de investigación tiene como objetivo abordar los desafíos de ciberseguridad a nivel mundial. Los autores han realizado una revisión exhaustiva del estado actual de este campo y, en base a las necesidades identificadas, proponen el diseño de una solución de seguridad informática que combina dos tecnologías: los sistemas multiagentes reactivos y BDI (creencias, deseos, intenciones) junto con las redes neuronales de aprendizaje profundo (Santiago & Sánchez Allende, 2016).

El propósito de esta solución es la detección y contención tanto de ataques informáticos tradicionales como de ataques más avanzados. Mediante la integración de estos enfoques tecnológicos, los investigadores buscan desarrollar una herramienta eficaz para hacer frente a los desafíos de seguridad digital a gran escala.

La investigación previa aborda la ciberseguridad de manera integral, combinando diferentes tecnologías como los sistemas multiagentes y las redes neuronales. Esto podría inspirarte a adoptar un enfoque similar en la investigación, integrando diferentes técnicas para lograr una solución más robusta. Adicional a esto la investigación menciona la capacidad de la solución propuesta para detectar y contener ataques informáticos tradicionales y avanzados. Esto podría ser relevante para la investigación, ya que los ataques de fuerza bruta pueden evolucionar y volverse más sofisticados en el futuro. Otro aporte se podría decir que es la integración de redes neuronales de aprendizaje profundo en la solución de la investigación previa puede aportar ideas y metodologías que podrías aplicar en tu proyecto para mejorar la detección de ataques de fuerza bruta. Los sistemas multiagentes y la arquitectura BDI (Beliefs, Desires, Intentions) mencionados en la investigación previa podrían inspirarte a incorporar elementos de automatización y adaptabilidad en tu solución, lo cual podría ser beneficioso para enfrentar ataques de fuerza bruta.

El análisis del tráfico de datos y los protocolos de la capa de enlace en redes LAN, específicamente en tecnologías Ethernet y Wifi 802.11. El objetivo es detectar posibles ataques a la red. El análisis se realiza mediante el monitoreo y captura de paquetes utilizando herramientas de software libre como Wireshark, y la visualización de conexiones y estadísticas de IP con herramientas como CommView. Se abordan diversos tipos de ataques en redes LAN, como spoofing, DNS spoofing, ARP spoofing y sniffing. Además, se examinan los diferentes estándares

y protocolos Wifi 802.11, sus vulnerabilidades y los mecanismos de protección y detección correspondientes. La red inalámbrica analizada es la red Wifi del laboratorio del Complejo Universitario "Computación y Redes" de la Universidad Estatal del Sur de Manabí (ÁLAVA CRUZATTY & ARCIA PLUA, 2021).

La investigación previa aborda el análisis del tráfico de datos y la detección de posibles ataques en redes LAN, lo cual puede ser relevante para la investigación, que se centra en el análisis de tráfico de red para la detección y defensa de ataques de fuerza bruta. El uso de herramientas de monitoreo y captura de paquetes, como Wireshark y Commview, empleadas en la investigación previa, puede proporcionar ideas y técnicas aplicables al análisis de tráfico de red para la detección de ataques de fuerza bruta. Aunque la investigación previa se enfoca en ataques como Spoof, DNS Spoofing, ARP spoofing y sniffing, los conceptos y técnicas de ataque pueden tener similitudes o relación con los ataques de fuerza bruta que pretendes abordar en la investigación (ÁLAVA CRUZATTY & ARCIA PLUA, 2021).

El análisis de las vulnerabilidades y mecanismos de protección en redes Wifi 802.11 realizado en la investigación previa puede aportar conocimientos relevantes para comprender los desafíos de seguridad en las redes y cómo enfrentar los ataques de fuerza bruta. La experiencia y los hallazgos de la investigación previa en el análisis de tráfico de red pueden servir como un punto de partida o referencia para el desarrollo de la propia metodología de análisis de la investigación y detección de ataques de fuerza bruta con Python (ÁLAVA CRUZATTY & ARCIA PLUA, 2021).

### 2.2.Seguridad en sistemas y redes informáticas

El objetivo principal de la seguridad en sistemas y redes es garantizar la total confiabilidad de los documentos, registros y archivos informáticos de una organización. El concepto de confiabilidad puede variar según los diferentes autores, contextos documentales y el tipo de organización a la que esté asociada la información. Sin embargo, en un contexto archivístico, donde se busca integrar un enfoque de seguridad informática con uno de preservación digital, la confiabilidad se puede definir en términos de seis características esenciales (Quiroz Zambrano & Macías Valencia, 2017).

En este contexto, la confiabilidad de la información implica asegurar que los documentos, registros y archivos informáticos mantengan de forma continua estas seis características clave, con

el fin de garantizar su integridad y autenticidad a lo largo del tiempo. Estas seis características esenciales de la confiabilidad de los documentos, registros y archivos informáticos son el punto focal del interés secundario de la seguridad informática desde la perspectiva de la preservación documental. El desafío radica en garantizar que estos seis elementos se mantengan en todo momento, a fin de asegurar la integridad y confianza en la información digital de la organización (Quiroz Zambrano & Macías Valencia, 2017).

El monitoreo y análisis del tráfico de red es crucial para el proyecto de investigación, ya que permite identificar patrones, anomalías y comportamientos sospechosos que puedan indicar la presencia de ataques de fuerza bruta. Esto es clave para la detección temprana de este tipo de amenazas. Además, la investigación se enfoca en desarrollar métodos eficaces para detectar e impedir los ataques de fuerza bruta, que son una forma de intrusión en la que los atacantes intentan adivinar contraseñas o credenciales de acceso de manera sistemática (Quiroz Zambrano & Macías Valencia, 2017).

Para abordar esta problemática, el trabajo también debe abordar los mecanismos de defensa y respuesta adecuados. Esto puede incluir el desarrollo de soluciones de software, como firewalls, sistemas de detección y prevención de intrusos, y herramientas de respuesta a incidentes.

### *1. 2.2.1. Amenazas y ataques a la seguridad informática*

Una amenaza informática es cualquier ocurrencia, evento o persona que tenga el potencial de causar daño a un sistema informático. Algunas de las principales tipificaciones de amenazas informáticas incluyen:

- Destrucción de datos
- Divulgación de información confidencial
- Robo de datos
- Modificación no autorizada de información
- Indisponibilidad del servicio o sistema

Estas amenazas se manifiestan en un lugar y tiempo específicos, con una determinada duración e intensidad. Esto conlleva a la materialización del riesgo en que se encuentra el sistema. Cabe destacar que una amenaza informática solo puede existir si hay una vulnerabilidad que pueda

ser aprovechada, independientemente de que dicha vulnerabilidad comprometa o no la seguridad del sistema de información. Es decir, la existencia de una amenaza está intrínsecamente ligada a la presencia de una vulnerabilidad que pueda ser explotada. Las amenazas informáticas representan todo aquel evento, persona u ocurrencia que tenga el potencial de causar daños diversos a los sistemas informáticos, y su materialización está condicionada a la presencia de vulnerabilidades que puedan ser aprovechadas (López, 2017, pág. 5).

Figura 1. Riesgo


Nota: Representación de las características de Riesgo (López, 2017).

Las amenazas y ataques a la seguridad informática representan diversos peligros que pueden comprometer la integridad, confidencialidad y disponibilidad de la información en sistemas y redes. En el contexto de su investigación sobre el "Análisis de tráfico de red para la detección y defensa de ataques de fuerza bruta con Python", estos tipos de amenazas y ataques son especialmente relevantes. Los ataques de fuerza bruta son una de las principales amenazas que el proyecto busca abordar. Estos ataques se caracterizan por intentos sistemáticos de adivinar contraseñas o credenciales de acceso, probando una gran cantidad de combinaciones posibles. El objetivo de estos ataques es obtener acceso no autorizado a sistemas, aplicaciones o recursos protegidos, lo que puede permitir a los atacantes realizar actividades maliciosas, robar información confidencial o incluso tomar el control de los sistemas (López, 2017).

Además de los ataques de fuerza bruta, los sistemas y aplicaciones también pueden presentar vulnerabilidades que pueden ser explotadas por los atacantes para ganar acceso no autorizado. Estos puntos débiles en los mecanismos de autenticación y control de acceso pueden ser identificados y aprovechados por los atacantes (López, 2017).

Otro tipo de amenaza relevante son los ataques de denegación de servicio (DoS), donde los atacantes buscan agotar los recursos del sistema, provocando que los usuarios legítimos no puedan acceder a los recursos y servicios (López, 2017).

Asimismo, los atacantes pueden utilizar técnicas de reconocimiento, como el escaneo de puertos o el análisis del tráfico de red, para recopilar información valiosa sobre los sistemas y redes. Esta información puede ser posteriormente utilizada para planificar y ejecutar ataques más sofisticados. Al comprender en profundidad estas amenazas y ataques a la seguridad informática, la investigación podrá desarrollar soluciones y estrategias de defensa más efectivas para proteger los sistemas y redes contra estos peligros (López, 2017).

### *2. 2.2.2. Importancia de la seguridad en organizaciones y sistemas computacionales*

Se puede describir que la importancia de reconocer lo vulnerables de la red más grande a nivel mundial tiende a crear ambiente de inquietud, ya que la globalización y la evolución tecnológica han llevado a que tanto los negocios personales como las industrias y grandes organizaciones deban innovar cada vez más. Sin embargo, la falta de seguridad adecuada puede dar lugar a pérdidas de información fundamental e incluso de efectivo, lo cual podría llegar a provocar el cierre definitivo de las actividades laborales (Arévalo Cordovilla, 2020).

Ante esta situación, el texto plantea la necesidad de crear un plan de contingencia e implementar medidas de seguridad efectivas. Esto con el objetivo de hacer frente a las posibles amenazas informáticas y evitar el robo masivo de información valiosa para los directivos (Arévalo Cordovilla, 2020).

Entender la importancia de anticipar y mitigar estos riesgos puede contribuir a la investigación, al brindar un marco conceptual para diseñar estrategias de seguridad efectivas que permitan salvaguardar la información crítica de las organizaciones ante amenazas emergentes, como los ataques de ingeniería social a través de canales de comunicación digitales (Arévalo Cordovilla, 2020).

## 2.3.Ataques de fuerza bruta

Los ataques de fuerza bruta representan una amenaza significativa para la seguridad de los sistemas informáticos, es por ello por lo que definimos a continuación algunos aspectos fundamentales

### *3. 2.3.1. Definición y características de los ataques de fuerza bruta*

El ataque de fuerza bruta es una técnica de intrusión que busca obtener acceso no autorizado a un servicio de red mediante la realización sistemática y repetitiva de intentos de inicio de sesión. Las credenciales utilizadas en estos ataques generalmente provienen de conjuntos predeterminados, contraseñas comúnmente usadas o credenciales previamente comprometidas a través de otros métodos (Gastón T. , 2015).

Este tipo de ataque es comúnmente utilizado en auditorías de seguridad para detectar debilidades en los sistemas de autenticación y autorización. Al probar exhaustivamente diferentes combinaciones de credenciales, se pueden identificar aquellos casos en los que las contraseñas utilizadas para acceder a un sistema son vulnerables, lo que podría permitir a un atacante obtener acceso no autorizado (Gastón T. , 2015).

Los ataques de fuerza bruta son una técnica utilizada por los atacantes para descifrar contraseñas o acceder a sistemas informáticos de forma no autorizada. Este método consiste en probar repetidamente múltiples combinaciones de credenciales hasta encontrar la correcta (kaspersky, 2024).

Algunas de las características clave de este tipo de ataques son:

- Enfoque sistemático y exhaustivo: Se prueban todas las posibles combinaciones de caracteres, números y símbolos hasta dar con la contraseña adecuada (Password Depot, 2024).
- Requiere mucho tiempo y recursos: Dependiendo de la complejidad de la contraseña, este proceso puede llevar desde segundos hasta años.
- Puede ser automatizado: Existen herramientas y scripts que facilitan y aceleran la ejecución de estos ataques de forma automatizada.
- Puede ser detectado: Los ataques de fuerza bruta generan patrones de tráfico y actividad sospechosos que pueden ser identificados por los sistemas de seguridad.

- Efectivo contra contraseñas débiles: Cuanto más sencilla y corta sea la contraseña, más probabilidades tiene el atacante de descubrirla.

### *4. 2.3.2. Técnicas y metodologías utilizadas en los ataques de fuerza bruta*

Los ataques de fuerza bruta se basan en la realización sistemática y exhaustiva de intentos de inicio de sesión a un sistema o servicio utilizando múltiples combinaciones de credenciales. Estas técnicas incluyen:

#### *2.3.2.1. Uso de diccionarios*

Los ataques de fuerza bruta tradicional, el atacante selecciona un objetivo y prueba diferentes contraseñas posibles contra ese nombre de usuario. Este enfoque se conoce como un ataque de diccionario. Los ataques de diccionario son la herramienta más básica dentro de los ataques de fuerza bruta. Si bien no son considerados ataques de fuerza bruta en sí mismos, a menudo se utilizan como un componente importante para descifrar contraseñas. Algunos atacantes emplean diccionarios exhaustivos e incorporan variaciones con caracteres especiales y números a las palabras del diccionario. También pueden utilizar diccionarios especializados de palabras. Sin embargo, este tipo de ataque secuencial y sistemático tiende a ser bastante complicado (kaspersky, 2024).

En general, los ataques de diccionario se utilizan como parte integral de los ataques de fuerza bruta, ya que permiten probar un amplio conjunto de contraseñas comunes y potencialmente válidas contra los objetivos seleccionados.

#### *2.3.2.2. Ataques de fuerza bruta híbridos*

También se define el tipo de ataque hibrido los ataques de diccionario, los atacantes también recurren a una combinación de métodos externos y razonamientos lógicos para intentar obtener acceso. Este enfoque se conoce como un ataque híbrido, que suele mezclar técnicas de ataques de diccionario y de fuerza bruta. En los ataques híbridos, los atacantes intentan adivinar combinaciones de contraseñas que mezclan palabras comunes con caracteres aleatorios. Por ejemplo, un ataque de fuerza bruta de este tipo podría probar contraseñas como "NuevaYork1993" o "Spike1234", que combinan palabras familiares con números y otros caracteres. Estos ataques híbridos aprovechan tanto el uso de diccionarios de contraseñas comunes como la generación

sistemática de variaciones aleatorias, con el objetivo de encontrar credenciales válidas que permitan acceder a los sistemas objetivo (kaspersky, 2024).

#### *2.3.2.3. Ataques de fuerza bruta inversa*

Otra de las técnicas descritas a diferencia de los ataques de diccionario y fuerza bruta convencionales, es este tipo de ataques de fuerza bruta invierten la estrategia. En lugar de probar múltiples contraseñas contra un nombre de usuario específico, el atacante comienza con una contraseña conocida y luego intenta encontrar millones de nombres de usuario que puedan coincidir con esa contraseña. Muchos de estos atacantes empiezan la búsqueda utilizando contraseñas que han sido filtradas y se encuentran disponibles en línea. Estas contraseñas suelen provenir de filtraciones de datos existentes, donde se han comprometido credenciales de usuarios. Al invertir el enfoque tradicional, este tipo de ataques de fuerza bruta busca aprovechar las contraseñas débiles o comprometidas que puedan estar siendo reutilizadas en múltiples cuentas. El objetivo es encontrar pares de nombre de usuario y contraseña válidos que permitan acceder a los sistemas objetivo (kaspersky, 2024).

#### *2.3.2.4. "Stuffing" Reutilización de credenciales comprometidas*

Se utilizan conjuntos de credenciales obtenidos a través de filtraciones de datos, ataques anteriores o el uso de credenciales predeterminadas en dispositivos o aplicaciones (Akamai, 2024).

#### *2.3.2.5. Paralización y distribución*

Se implementan técnicas de ejecución paralela y distribución de los intentos de inicio de sesión entre múltiples sistemas para aumentar la velocidad y eficiencia del ataque.

### *5. 2.3.3. Impactos y consecuencias de los ataques de fuerza bruta*

Los ataques de fuerza bruta pueden tener serias repercusiones tanto para los sistemas y servicios objetivo como para los usuarios afectados. Algunas de las principales consecuencias incluyen:

#### *2.3.3.1. Acceso no autorizado*

Los ataques de fuerza bruta se basan en intentos sistemáticos y repetitivos de inicio de sesión a un servicio de red, con el objetivo de obtener acceso no autorizado. Para llevar a cabo estos ataques, los perpetradores utilizan credenciales que potencialmente podrían ser válidas. Estas

credenciales empleadas suelen provenir de diferentes fuentes, como conjuntos de credenciales predeterminadas para ciertos dispositivos o aplicaciones, contraseñas comúnmente utilizadas por los usuarios, o credenciales que fueron comprometidas previamente a través de otros tipos de ataques (Gastón, Molinari, Venosa, Macia, & Lanfranco, 2015).

Las auditorías de seguridad, los ataques de fuerza bruta permiten identificar la mayoría de los casos en los que las credenciales utilizadas para acceder a un servicio presentan debilidades. Estas vulnerabilidades podrían permitir a un atacante obtener acceso no autorizado al sistema de información que se está analizando para el ataque o robo de información. Una vez que se obtienen las credenciales, los atacantes pueden acceder a datos sensibles, registros, correos electrónicos y otra información valiosa almacenada en los sistemas comprometidos.

#### *2.3.3.3. Interrupción de servicios*

Los intentos repetitivos de inicio de sesión pueden saturar y sobrecargar los sistemas, provocando la denegación de servicio y la indisponibilidad de aplicaciones y recursos en línea (kaspersky, 2024).

#### *2.3.3.4. Daños financieros*

El acceso no autorizado a cuentas bancarias, tarjetas de crédito u otros activos financieros puede resultar en pérdidas económicas significativas para las víctimas (interpol, 2024).

#### *2.3.3.5. Daños a la reputación*

Los incidentes de seguridad resultantes de estos ataques pueden dañar la reputación y la confianza de los usuarios en las organizaciones afectadas (Faster capital, 2024).

#### *2.3.3.6. Impacto legal y normativo*

Dependiendo de la jurisdicción, los ataques de fuerza bruta pueden constituir infracciones legales y acarrear sanciones y responsabilidades para los perpetradores (Hack Metrix, 2024).

### 2.4.Herramientas y técnicas de análisis de tráfico de red

El monitoreo y análisis del tráfico de red es fundamental para comprender y gestionar el flujo de información en una red. Existen diversas herramientas y métodos que permiten llevar a cabo estas tareas ( Aguilar, Kristell , Marxjhony , & Carlos , 2017).

### 6. *2.4.1. Wireshark: funcionamiento, capacidades y aplicaciones en seguridad informática*

Antes conocido como Ethereal, es uno de los analizadores de red más utilizados en la actualidad. Esta herramienta permite a los usuarios capturar y examinar el tráfico de red en un momento específico. Wireshark se ha convertido en la referencia principal para el análisis de tráfico de red gracias a sus numerosos módulos de decodificación, filtrado y visualización. Estas características hacen que el análisis de los paquetes capturados resulte una tarea sencilla y accesible para los usuarios. La amplia gama de funcionalidades que ofrece Wireshark, como la decodificación de más de 480 protocolos de red, el filtrado avanzado y la generación de estadísticas y gráficos, lo convierten en una herramienta fundamental para la comprensión en profundidad del tráfico de red. Esto la convierte en una herramienta indispensable para tareas como la detección de amenazas, la resolución de problemas de conectividad y el análisis forense de incidentes de seguridad (Ballestero Muñoz, 2021).

Wireshark es una de las aplicaciones líderes en el campo del análisis de tráfico de red, ofreciendo a los usuarios una interfaz intuitiva y un conjunto de funcionalidades avanzadas que facilitan el estudio y la comprensión del flujo de datos en las redes (Ballestero Muñoz, 2021).

Wireshark es una herramienta de análisis de protocolos de red ampliamente utilizada en el campo de la seguridad informática. Veamos con más detalle su funcionamiento, capacidades y aplicaciones:

#### *2.4.1.1. Funcionamiento de Wireshark*

Wireshark es un analizador de tráfico de red de código abierto que permite capturar, analizar y examinar en detalle el flujo de datos que circula a través de una red. Funciona interceptando y decodificando los paquetes de red, mostrando la información de cada capa del modelo OSI. Esto permite a los usuarios comprender en profundidad el tráfico de la red y detectar posibles problemas o actividades sospechosas (Ballestero Muñoz, 2021, pág. 45).

#### *2.4.1.2. Capacidades de Wireshark*

- Captura de tráfico en tiempo real a través de diversas interfaces de red.
- Permite el análisis de más de 480 Protocolos

- Decodificación de más de 1.000 protocolos de red, desde los estándares más comunes hasta los más especializados.
- Filtrado y búsqueda avanzada del tráfico capturado para identificar patrones y anomalías.
- Generación de estadísticas y gráficos sobre el uso de ancho de banda, protocolos, direcciones IP, puertos, etc.
- Exportación de datos capturados en múltiples formatos para su posterior análisis.
- Integración con otras herramientas de seguridad y monitorización.
- Aplicaciones de Wireshark en seguridad informática.
- Detección de actividades maliciosas: Wireshark permite analizar el tráfico de red en busca de patrones sospechosos, como escaneos de puertos, ataques de denegación de servicio (DoS) o intentos de intrusión.
- Análisis forense: La capacidad de Wireshark para capturar y reconstruir el tráfico de red lo convierte en una herramienta valiosa para investigar incidentes de seguridad y recopilar evidencia digital.
- Resolución de problemas de red: Wireshark ayuda a los administradores de red a identificar y solucionar problemas de conectividad, rendimiento y configuración de los dispositivos.
- Pruebas de penetración: Los profesionales de seguridad utilizan Wireshark para analizar y comprender el tráfico de red durante las pruebas de penetración, lo que les permite identificar y explotar vulnerabilidades.
- Educación y capacitación: Wireshark es ampliamente utilizado en la formación de profesionales de TI y seguridad, ya que permite comprender en profundidad el funcionamiento de los protocolos de red.
- Manejo de estadísticas de Trafico de red (Ballestero Muñoz, 2021, pág. 43).

### *7. 2.4.2. Técnicas de análisis de tráfico de red para la detección de ataques*

Las organizaciones se enfrentan al desafío de monitorizar y gestionar el cada vez más complejo tráfico de red en sus infraestructuras. Los tradicionales sistemas de detección de intrusiones basados en firmas, que buscan patrones conocidos de ataques, han demostrado ser insuficientes para hacer frente a las nuevas amenazas, como los ataques de día cero o los que utilizan técnicas de evasión. Por este motivo, la detección de anomalías en el tráfico de red se ha convertido en una técnica fundamental para la protección de las redes. Mediante métodos

estadísticos y de aprendizaje automático, estas soluciones pueden aprender el comportamiento normal de la red y detectar actividades inusuales que puedan indicar la presencia de ataques (Manageengine, 2024).

Describe que, a diferencia de los enfoques basados en firmas, la detección de anomalías no depende del conocimiento previo de patrones de ataque. En su lugar, analiza el tráfico en busca de desviaciones respecto a los patrones de comportamiento establecidos como normales. Esto permite identificar tanto ataques conocidos como desconocidos, brindando una protección más efectiva ante las amenazas emergentes. Las técnicas de detección de anomalías analizan diversos atributos del tráfico, como direcciones IP, puertos, protocolos y volúmenes, para crear un perfil de referencia del comportamiento de la red. Cualquier actividad que se desvíe significativamente de este perfil será marcada como sospechosa y generará una alerta para que los administradores puedan investigar y tomar las medidas necesarias. Esta visión holística del tráfico de red, combinada con la capacidad de adaptación del aprendizaje automático, convierte a la detección de anomalías en una herramienta efectiva y flexible para la protección de las infraestructuras de red ante una amplia gama de ataques, incluyendo aquellos que evaden los sistemas de detección tradicionales (Manageengine, 2024).

### 8. *2.4.3. Algoritmos y métodos de análisis de tráfico de red*

El análisis del tráfico de red involucra el uso de diversos algoritmos y técnicas para monitorear, inspeccionar y procesar los datos que circulan a través de las redes de comunicación. Algunos de los principales métodos y enfoques incluyen:

#### *2.4.3.1. Patrones de tráfico*

La detección de anomalías en redes de datos se basa en la identificación de patrones que se desvían del comportamiento normal del tráfico. Por lo tanto, está estrechamente relacionada con la modelización del tráfico de red. Para poder detectar comportamientos anormales, es necesario contar con modelos de tráfico precisos y estables que describan adecuadamente lo que constituye un tráfico libre de anomalías. Este paso de crear un modelo de tráfico correcto es fundamental, ya que un modelo incorrecto o inestable puede perjudicar seriamente la capacidad de detectar de forma precisa las anomalías, lo que provocaría un alto número de falsas alarmas. La precisión y estabilidad del modelo de tráfico utilizado es, por lo tanto, un elemento clave para lograr una detección de anomalías eficaz y confiable. Esto permite evitar generar demasiadas alertas falsas que puedan dificultar la tarea de los administradores de red. ***(Barrionuevo, Lopresti, Miranda, & Píccoli, 2016)***.

Se recopilan y analizan métricas como volumen de datos, tasas de transferencia, protocolos utilizados, orígenes y destinos, con el fin de identificar patrones y anomalías en el comportamiento de la red.

#### *2.4.3.2. Inspección de paquetes "Detección mediante inspección profunda de paquetes"*

Este segundo proceso de detección se lleva a cabo después de que un primer análisis haya encontrado indicios de la presencia de una posible red botnet en la red o de que se esté sufriendo un ataque DoS originado por una botnet. Esto puede ocurrir ya sea porque no se pudo detectar anteriormente o porque el ataque proviene de bots externos. En cualquiera de estos casos, es necesario realizar una comprobación minuciosa para verificar que dicha amenaza ha ocurrido realmente. Para ello, el trabajo de investigación de proponer ejecutar un nuevo proceso de detección a un nivel más bajo mediante una inspección profunda de paquetes (DPI). Esta técnica permite analizar en detalle el tráfico de red para confirmar la presencia de la amenaza botnet detectada anteriormente (Perez & Mart´ınez Perez, 2016).

Para realizar esta nueva fase de detección más profunda, se ha utilizado la solución obtenida en el proyecto Reclamo. En particular, para la detección de redes botnet o ataques DDoS, se ha adaptado la solución de Reclamo con el objetivo de llevar a cabo una inspección profunda de paquetes. El propósito de esta inspección en profundidad del tráfico de red es confirmar si la amenaza botnet o el ataque DDoS detectados previamente han ocurrido realmente en el entorno

analizado. Mediante este análisis detallado del tráfico, se busca validar los indicios iniciales y determinar con mayor precisión si la supuesta amenaza es real o no. (Perez & Mart´ınez Perez, 2016).

Para implementar esta segunda fase de detección, se colocan estratégicamente varios sistemas de detección de intrusiones (IDS) en la red de monitoreo. Estos IDS se configuran para analizar únicamente los paquetes de red de los posibles bots identificados durante la detección de alto nivel de la sección anterior. Esta configuración dependerá del tipo específico de botnet que se haya detectado previamente. Por ejemplo, una posible regla en Snort podría ser para detectar solicitudes HTTP de comandos que el bot tendría que ejecutar, dirigidas al servidor de comando y control. (Perez & Mart´ınez Perez, 2016).

#### *2.4.3.3. Técnicas de muestreo*

Para procesar grandes volúmenes de tráfico, se utilizan métodos de muestreo que permiten analizar una fracción representativa de los datos, reduciendo la carga computacional. El muestreo es beneficioso para algunas tareas de planificación de la capacidad de una red, ya que no es necesario analizar todos los flujos para entender el comportamiento de la red. Sin embargo, la calidad de la aproximación obtenida depende tanto de la frecuencia de muestreo elegida, como de la forma en que se seleccionan los paquetes a muestrear (Bastías López , 2017).

#### *2.4.3.4. Aprendizaje automático y detección de anomalías*

Se aplican algoritmos de aprendizaje de máquina para modelar y aprender el comportamiento "normal" de la red, permitiendo la identificación de actividades atípicas o potencialmente maliciosas (Bella Santos, 2019).

#### *2.4.3.5. Correlación de eventos y logs*

El análisis conjunto de registros de eventos, logs de aplicaciones y otros datos complementarios permite obtener una visión más completa del tráfico y detectar posibles actividades sospechosas (Camacho Echavarría & Moreno López, , 2018).

#### *2.4.3.6. Visualización y tableros de control*

La representación gráfica y la creación de tableros de control facilitan la interpretación y la comprensión del estado y las tendencias del tráfico de red por parte de los analistas. Estos algoritmos y metodologías de análisis de tráfico de red desempeñan un papel fundamental en la

detección temprana de amenazas, la optimización del rendimiento de la red y la toma de decisiones informadas sobre la seguridad y el funcionamiento de los entornos de comunicación (Ibáñez Moruno, Lévano Lévano, & Nieto Maldonado, 2016).

### *9. Monitoreo de seguridad de red*

El monitoreo de la seguridad es una parte crucial de la defensa de la red. Las organizaciones deben aceptar que sus redes eventualmente se verán comprometidas y poner más atención y recursos en los mecanismos de detección y respuesta. En las empresas e instituciones, el monitoreo de la seguridad de la red se define como un proceso cíclico que consta de tres fases: recopilación, detección y análisis. La fase de recopilación consiste en definir amenazas (Junco Romero & Rabelo Padua, 2018).

Hay que evaluar los riesgos y seleccionar los datos apropiados que se recopilarán. No se recomienda recopilar todas las fuentes de datos disponibles, sino más bien ser selectivo teniendo en cuenta amenazas específicas. Hay varios tipos diferentes de datos que se pueden recopilar: captura de paquetes completos, datos de flujo y datos de registro. Los datos de captura de paquetes completos, conocidos principalmente en formato PCAP, proporcionan una transcripción completa de la comunicación entre dos dispositivos de red. Su alto grado de detalle lo hace valioso para el análisis, pero por lo demás, resulta poco práctico debido a su enorme tamaño. Los datos de flujo representan un resumen de la comunicación entre dos puntos finales. Por lo general, incluye información sobre los hosts que se comunican, sus direcciones IP y puertos, el protocolo utilizado, marcas de tiempo y la cantidad de datos transferidos (Junco Romero & Rabelo Padua, 2018) .

La detección es el proceso mediante el cual se examinan los datos recopilados y la forma en que se generan alertas en función de los eventos observados y un conjunto de reglas de firma. Las alertas generadas luego se presentan a los analistas de seguridad para su inspección manual o se utilizan para un procesamiento automático posterior. En este trabajo, me centraré en los sistemas

#### *Monitoreo Activo*

El monitoreo activo se realiza introduciendo paquetes de prueba en la red o enviando paquetes a aplicaciones específicas, midiendo sus tiempos de respuesta. Este enfoque agrega tráfico a la red y se utiliza para medir su rendimiento (Junco Romero & Rabelo Padua, 2018).

Figura 2. Monitorización Activa


Nota: Representación de la Monitorización activa (A3sec, 2024)

***Técnicas de Monitoreo Activo:***

- Basadas en ICMP (Internet Control Message Protocol): diagnóstico de problemas de red, detección de retardo y pérdida de paquetes, cálculo del RTT (Round-Trip Delay Time), y disponibilidad de hosts y redes (Junco Romero & Rabelo Padua, 2018).

- Basadas en TCP (Transmission Control Protocol): medición de tasas de transferencia y diagnóstico de problemas a nivel de aplicación.

- Basadas en UDP (User Datagram Protocol): detección de pérdida de paquetes unidireccional y medición del RTT (traceroute).

***Monitoreo Pasivo***

Este enfoque se basa en la recolección y análisis del tráfico que circula por la red, sin agregar tráfico adicional. Se utilizan dispositivos como sondas, enrutadores, computadoras con software de análisis de tráfico, y dispositivos con soporte para protocolos como SNMP, RMON y herramientas como Netflow. Esto permite caracterizar y contabilizar el uso del tráfico de red (Junco Romero & Rabelo Padua, 2018).

Figura 3 Monitorización Pasiva


Nota: Representación de la Monitorización Pasiva (A3sec, 2024)

***Técnicas de Monitoreo Pasivo***

Obtención de estadísticas de utilización de ancho de banda y notificación de eventos inusuales (traps). Scripts que obtengan información importante de los dispositivos, captura de tráfico mediante puertos espejo o dispositivos intermedios que generen copias del tráfico, Análisis de tráfico o caracterización del tráfico por aplicación, direcciones IP, puertos, etc. utilizando sondas RMON o dispositivos intermedios también la identificación del tipo de tráfico por dirección, puertos TCP y tipo de aplicación, obtenidos de enrutadores o dispositivos de captura (Junco Romero & Rabelo Padua, 2018).

***Estrategias de Monitoreo***

Antes de implementar un sistema de monitoreo, es importante definir los elementos a monitorear, como utilización de ancho de banda, consumo de CPU y memoria, estado físico de conexiones, tipo de tráfico, alarmas y servicios. También se deben considerar los recursos humanos, económicos y de infraestructura disponibles (Junco Romero & Rabelo Padua, 2018).

Herramientas de Monitoreo:

- Cacti: solución completa de monitoreo de redes, con almacenamiento en RRDTools, múltiples métodos de recolección de datos, manejo de plantillas y alarmas.

- Net-SNMP: conjunto de aplicaciones para obtener información vía SNMP, incluyendo soporte para SNMP v3 y manejo de traps.

La topología típica incluye un servidor que realiza solicitudes SNMP a los dispositivos de red, los cuales envían la información solicitada a través de un agente SNMP. También es posible

que los dispositivos envíen mensajes trap al servidor SNMP cuando ocurren eventos inusuales. de detección de intrusiones en la red (NIDS) que procesan datos de flujo para identificar amenazas en el tráfico monitoreado (Junco Romero & Rabelo Padua, 2018).

### 2.5.IPTables

Iptables es una herramienta esencial de Linux diseñada para gestionar y filtrar el tráfico de red mediante reglas definidas. Su nombre, que deriva de "IP" y "tables", refleja su función principal de trabajar con tablas que contienen reglas específicas para determinar el destino de los paquetes de datos. Este sistema de filtrado de paquetes ofrece diversas funcionalidades, desde las más básicas hasta las avanzadas de administración de red. Con iptables, se puede bloquear o permitir el tráfico según criterios como direcciones IP, puertos, protocolos y estados de conexión (Zone, 2024).

Una de las características más destacadas de iptables es su capacidad para realizar la traducción de direcciones de red, lo que permite compartir una única dirección IP pública entre varios dispositivos de una red interna. También puede realizar el mapeo de puertos, redirigiendo el tráfico de un puerto específico a otro, facilitando la exposición de servicios internos a través de una sola dirección IP pública. Iptables realiza un seguimiento del estado de las conexiones, recordando si una conexión está establecida, en curso o es una respuesta a una conexión interna. Esto proporciona una capacidad avanzada para establecer reglas de estado y gestionar el tráfico de manera eficiente (Zone, 2024).

En términos de configuración, iptables utiliza reglas organizadas en diferentes tablas, como la tabla "filter" para el filtrado básico, la tabla "nat" para la traducción de direcciones, y la tabla "mangle" para la manipulación de paquetes.

### 2.6.Metodologías de Ataque de Fuerza bruta

Es importante comprender los diferentes métodos de ataque de fuerza bruta que pueden ser empleados, sus características clave. A continuación, se presenta una tabla que resume algunos de los principales métodos de ataque de fuerza bruta, su descripción.

Tabla 1.
Métodos de ataques de Fuerza bruta

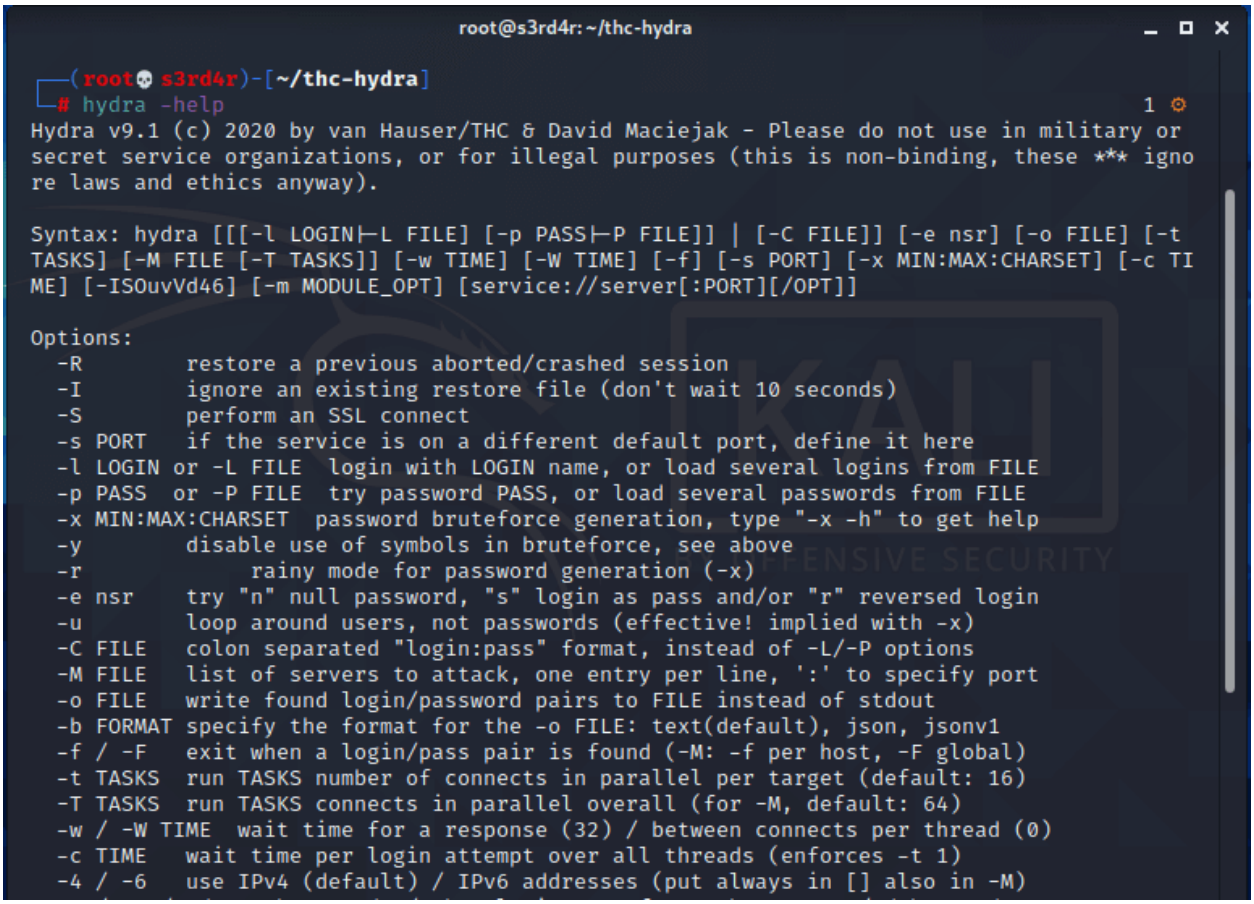
| Método de ataque de fuerza bruta | Descripción | Herramientas |
|---|---|---|
| Ataque de Fuerza Bruta de Contraseñas | Intento sistemático de todas las posibles combinaciones de caracteres para descubrir la contraseña de un sistema o cuenta. | Hidra, Medusa, Juan el Destripador, Hashcat |
| Ataque de Fuerza Bruta de Credenciales | Intento exhaustivo de diferentes combinaciones de nombres de usuarios y password | Hidra, Metasploit, Ncrack |
| Ataque de Fuerza Bruta de Cifrado | Intentar descifrar datos cifrados probando todas las posibles claves de cifrado. | Hashcat, Juan el Destripador, oclHashcat |
| Ataque de Fuerza Bruta de Hashes | Intente revertir un hash (resumen criptográfico) para obtener el valor original, como una contraseña. | Hashcat, Juan el Destripador, oclHashcat |
| Ataque de Fuerza Bruta de Redes Inalámbricas | Intenta descubrir la clave de acceso a una red Wi-Fi probando todas las combinaciones posibles. | Aircrack-ng, Wifite, Reaver |
| Ataque de Fuerza Bruta de Puertos | Intentar acceder a diferentes puertos de red de un sistema para identificar servicios vulnerables. | Nmap, Unicornscan, Angry IP Scanner |
| Ataque de F | Intenta adivinar nombres de directorios y archivos en un servidor web para descubrir información confidencial. | DirBuster, GoBuster, Wfuzz |
| Ataque de Fuerza Bruta de APIs | Intentar descubrir endpoints y métodos válidos de una API a través de pruebas exhaustivas. | Cartero, Suite Burp, OWASP ZAP |
| Ataque de Fuerza Bruta de OTP (Contraseñas de un solo uso) | Intente descubrir códigos de autenticación de un solo uso a través de múltiples intentos. | Hidra, Medusa, Hashcat |
| Ataque de Fuerza Bruta de Captchas | Intenta resolver de forma automatizada los desafíos de Captcha para evadir mecanismos de seguridad. | antiCaptcha, 2Captcha, MuertePorCaptcha |

Nota: Recopilacion de métodos de ataques(Checkpoint, 2024)

### *2.6.1. Hidra*

Según, Hydra es una herramienta de fuerza bruta en Kali Linux que automatiza ataques para adivinar contraseñas. Permite personalizar los ataques, trabajar en varios protocolos, y ofrece diferentes modos de operación. Utiliza multi-threading para optimizar el rendimiento y proporciona informes detallados sobre los resultados de los ataques (Medium, 2024).

Figura 4. Herramienta Hydra




Nota: detalles de las distintas opciones de ataque para la herramienta Hydra (Medium, 2024)

Para utilizar Hydra en un ataque a RDP en Kali Linux, simplemente sigue estos pasos: a. Abre una terminal en Kali Linux. b. Utiliza el comando siguiente para instalar Hydra si aún no lo has hecho (Ortega Candel, 2018).

*sudo apt-get update*

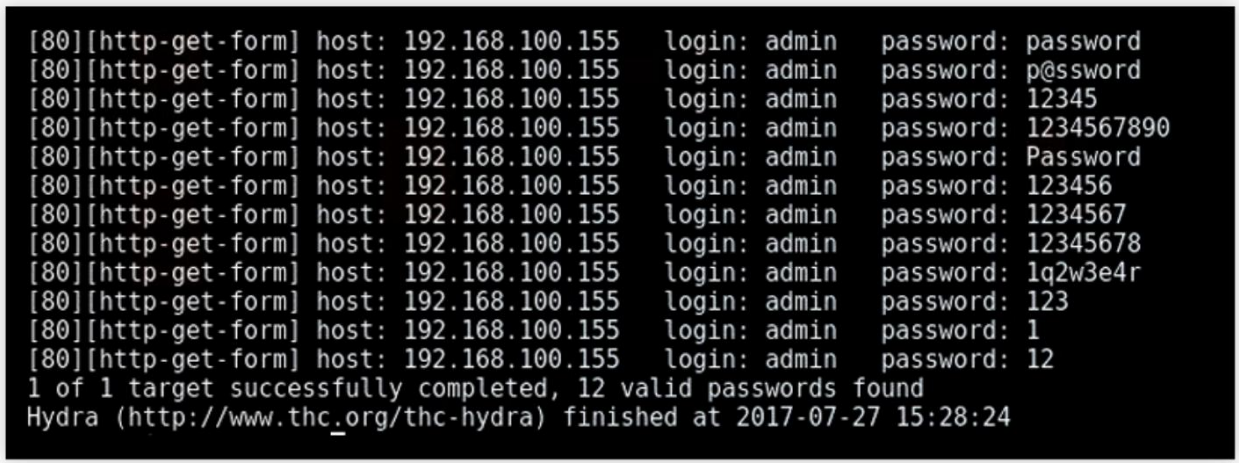
*sudo apt-get install hydra*

Ya instalado Hidra en Kali, procede a ejecutar el siguiente comando:

hydra -l usuario -P contraseñas.txt rdp://dirección_ip

Donde usuario es el nombre de usuario a probar, el archivo txt a pasar como diccionario de posibles contraseñas y por último la dirección ip de la victima de prueba (Ortega Candel, 2018)..

Figura 5. Herramienta Hydra




Nota: Representación del ataque utilizando el diccionario para usuario y contraseña (Imperva, 2024)

***Medusa***

Es una herramienta de fuerza bruta multi-threading que permite realizar ataques de adivinanza de credenciales en múltiples protocolos y servicios, como FTP, SSH, HTTP, SMB, entre otros. Esta herramienta es altamente personalizable y ofrece diversas opciones para configurar y ejecutar los ataques de fuerza bruta de acuerdo con las necesidades y especificaciones del usuario. Medusa es una opción popular para auditores de seguridad y profesionales de la ciberseguridad que buscan evaluar la fortaleza de contraseñas y credenciales en sistemas y redes (Medium, 2024).

Para instalar debe generar los siguientes comandos en la consola de kali linux:

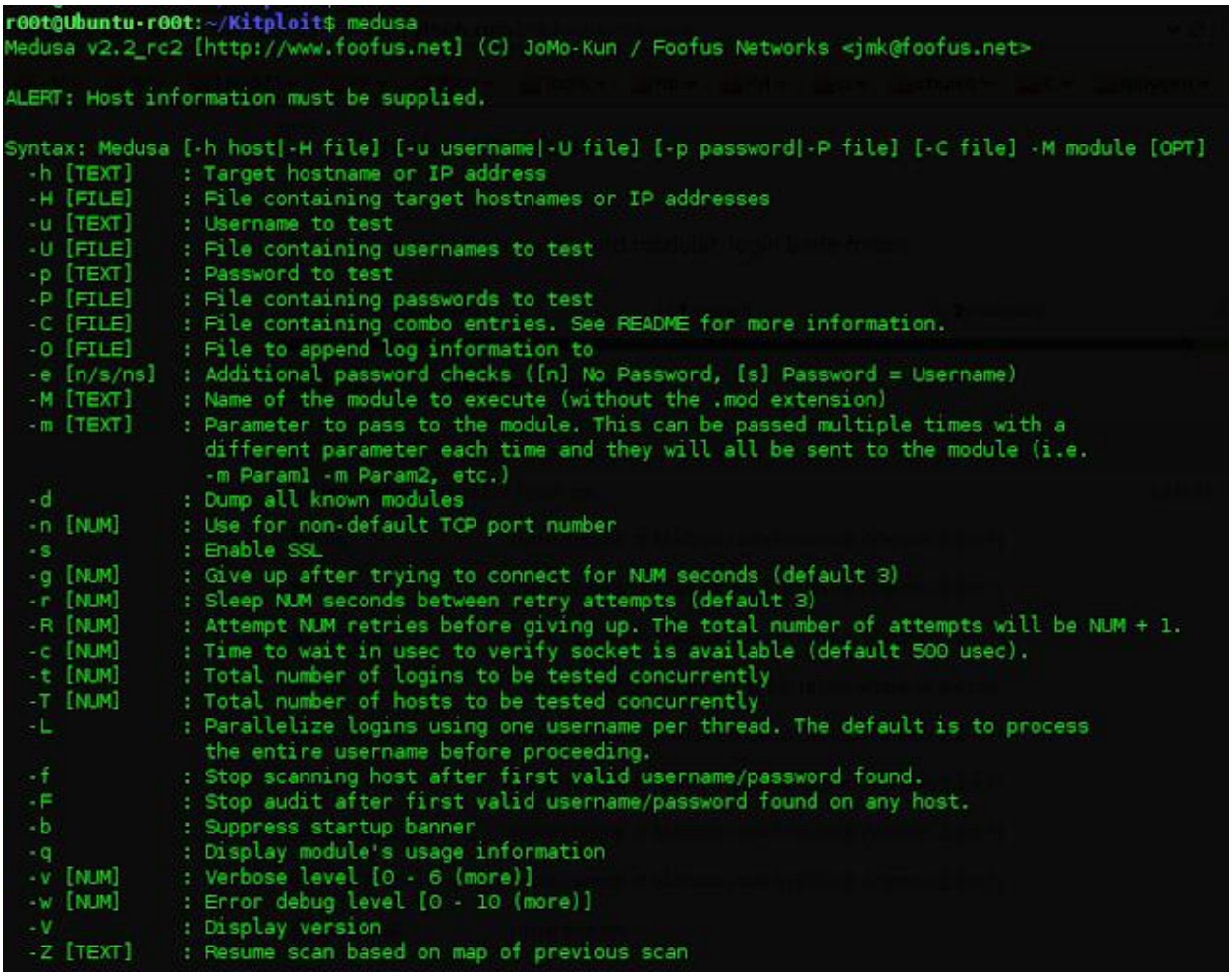
*sudo apt-get update*

*sudo apt-get install medusa*

ya cuando se instale se podrá realizar el ataque con el siguiente comando:

*medusa -h dirección_ip -u usuario -P contraseñas.txt -M rdp*

Figura 6. Herramienta Medusa




Nota: Diferentes opciones de ataque con la herramienta Medusa (Kitploit, 2024)

### *Ncrack*

Es una herramienta de auditoría de seguridad especializada en el escaneo de puertos, detección de vulnerabilidades y realización de ataques de fuerza bruta en diversos servicios y protocolos de red. Esta utilidad permite a los usuarios identificar posibles vulnerabilidades en sistemas y redes empresariales a través de pruebas de penetración automatizadas y personalizables. Ncrack es ampliamente utilizado por profesionales de la ciberseguridad para evaluar y mejorar la seguridad de las infraestructuras informáticas (Medium, 2024).

Para instalarle solo se necesita ingresar por consola el siguiente comando:

*sudo apt-get update*

*sudo apt-get install ncrack*

ya instalado solo inicia el ataque con el siguiente comando:

*ncrack -U usuario -P contraseñas.txt rdp://dirección_ip*

Figura 7. Ncrack


Nota: Ataque perpetrado con Ncrack (Hackin garticles, 2024)

### *10. 3.3. Herramientas de Kali Linux para el ataque de fuerza bruta*

- Descripción de las principales herramientas de Kali Linux que se utilizarán, como Hydra, Medusa, Ncrack, entre otras.

Tabla 2.
Herramientas de Fuerza Bruta

| Herramientas | Funcionabilidad | Protocoles/Áreas de enfoque | Ejemplos de uso |
|---|---|---|---|
| Hydra | Realiza ataques de fuerza bruta para adivinar contraseñas de forma automatizada | Contraseñas | Realizar pruebas de penetración en sistemas web |
| Medusa | Realiza ataques de fuerza bruta probando múltiples protocolos y credenciales | Múltiples protocolos | Auditar la seguridad de servidores FTP |
| Ncrack | Utilidad de auditoría de seguridad enfocada en el escaneo de puertos y detección de vulnerabilidades | Escaneo de puertos, detección de vulnerabilidades | Detectar vulnerabilidades en una red empresarial |

Nota: Recopilación de las principales herramientas de aplicación de fuerza bruta (Medium, 2024)

### *Implementación de Odoo versión Comunitaria*

La edición comunitaria de Odoo es una versión de código abierto de la plataforma que ofrece diversas aplicaciones empresariales como CRM, ventas, inventario, contabilidad y recursos humanos. Esta edición es gratuita y se caracteriza por ser altamente personalizable y adaptable a las necesidades particulares de cada empresa. A través de la colaboración con la comunidad de desarrolladores y usuarios, es posible acceder a módulos adicionales, soporte técnico y recursos de personalización que permiten ampliar las funcionalidades de Odoo de acuerdo con los requerimientos específicos de cada negocio. Esta desarrollado el Python y su base de datos es en PostgreSQL, su sistema de validación es un usuario y una contraseña lo que permite ser una fuente de pruebas de ataques de fuerza bruta, adicional su implementación en la nube de Google cloud se realizará a través de SSH lo que permitirá realizar ataques adicionales de prueba (Dizzet Utria, 2017).

***Pasos de Instalación por Consola:***

Los pasos para instalar Odoo Comunitario en Linux v17 a través de la consola son los siguientes:

***Actualizar los repositorios del sistema:***

*sudo apt-get update*

***Instalar las dependencias necesarias:***

*Sudo apt-get install python3-dev python3-pip python3-setuptools libxml2-dev libxslt1-dev libpango1.0-0 npm*

***Instalar PostgreSQL (base de datos utilizado por Odoo):***

*sudo apt-get install postgresql*

***Crear un nuevo usuario de base de datos en PostgreSQL:***

*sudo su - postgres -c "createuser -s odoo"*

***Descargar Odoo v17 desde el repositorio oficial:***

*sudo git clone https://www.github.com/odoo/odoo --depth 1 --branch 17.0 /opt/odoo*

***Instalar las dependencias de Python de Odoo:***

*sudo pip3 install -r /opt/odoo/requirements.txt*

***Configurar Odoo:***

*cd /opt/odoo*

*sudo cp /opt/odoo/debian/odoo.conf /etc/odoo.conf*

*sudo nano /etc/odoo.conf*

(Editar el archivo de configuración con los datos de conexión a la base de datos)

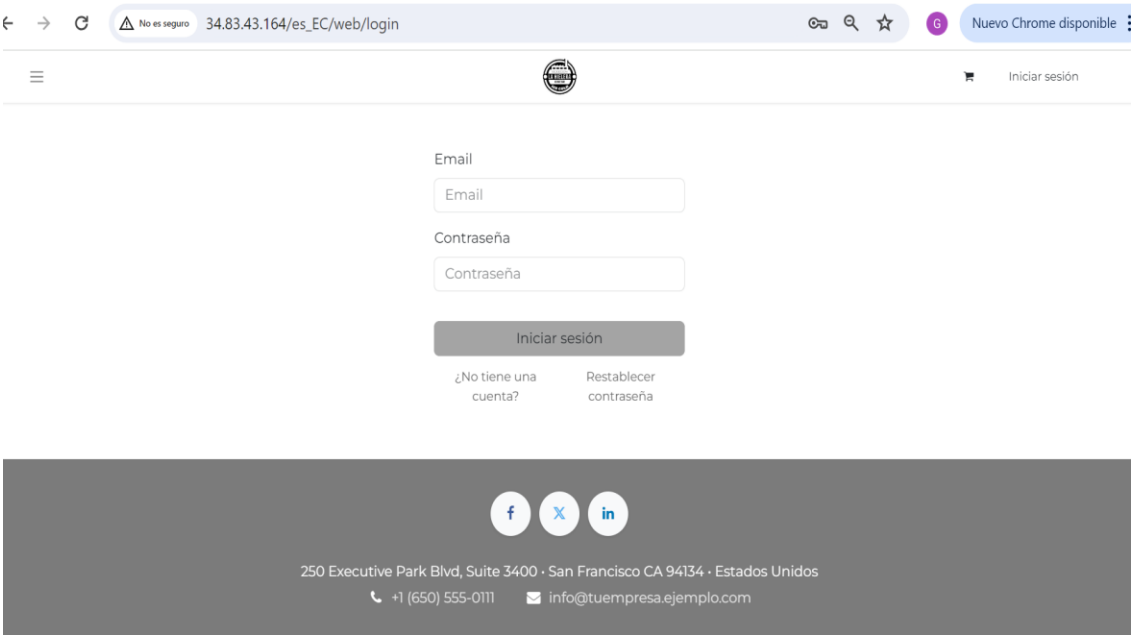
**Iniciar el servidor de Odoo:**

*/opt/odoo/odoo-bin -c /etc/odoo.conf*

Acceder a Odoo desde un navegador web:

*http://localhost en sus efectos por la Ip publica asignada por Google cloud*

Figura 8. Odoo Comunitario




Nota: Pagina de acceso a la plataforma de prueba de Odoo.

Estos son los pasos básicos para instalar Odoo Comunitario en Linux v17 a través de la consola. Es importante recordar que pueden variar dependiendo de la distribución de Linux que estés utilizando y de las configuraciones específicas de tu sistema.

# CAPITULO III METODOLOGÍA DE ATAQUE “PLAN DE ATAQUE”

Tras evaluar los diferentes métodos de ataque de fuerza bruta presentados en la tabla anterior, se ha decidido enfocar el trabajo de investigación en el Ataque de Fuerza Bruta de credenciales. Esta selección se basa en las siguientes consideraciones:

## 3.1.Relevancia y prevalencia del ataque:

Los ataques de fuerza bruta de credenciales son una de las técnicas de intrusión más comunes y ampliamente utilizadas por atacantes maliciosos. Debido a la proliferación de sistemas y aplicaciones que requieren autenticación, este tipo de ataque sigue siendo una amenaza importante y recurrente.

## 3.2.Disponibilidad de herramientas y material de referencia:

Existen numerosas herramientas de código abierto y comerciales que permiten implementar ataques de fuerza bruta de credenciales, como Hydra, Medusa, Metasploit, entre otras. Hay una amplia variedad de recursos, tutoriales y estudios de caso disponibles sobre la ejecución de este tipo de ataques, lo que facilitará el desarrollo de las simulaciones y el análisis de resultados.

## 3.3.Complejidad y eficacia del ataque:

Los ataques de fuerza bruta de credenciales combinan dos enfoques de fuerza bruta (usuario y contraseña), lo que los hace más efectivos que los ataques enfocados únicamente en las contraseñas. Aunque pueden ser más lentos que los ataques específicos de contraseñas pueden tener un mayor éxito en escenarios donde los usuarios utilizan combinaciones de credenciales comunes o predecibles.

## 3.4.Relevancia para el contexto de la investigación:

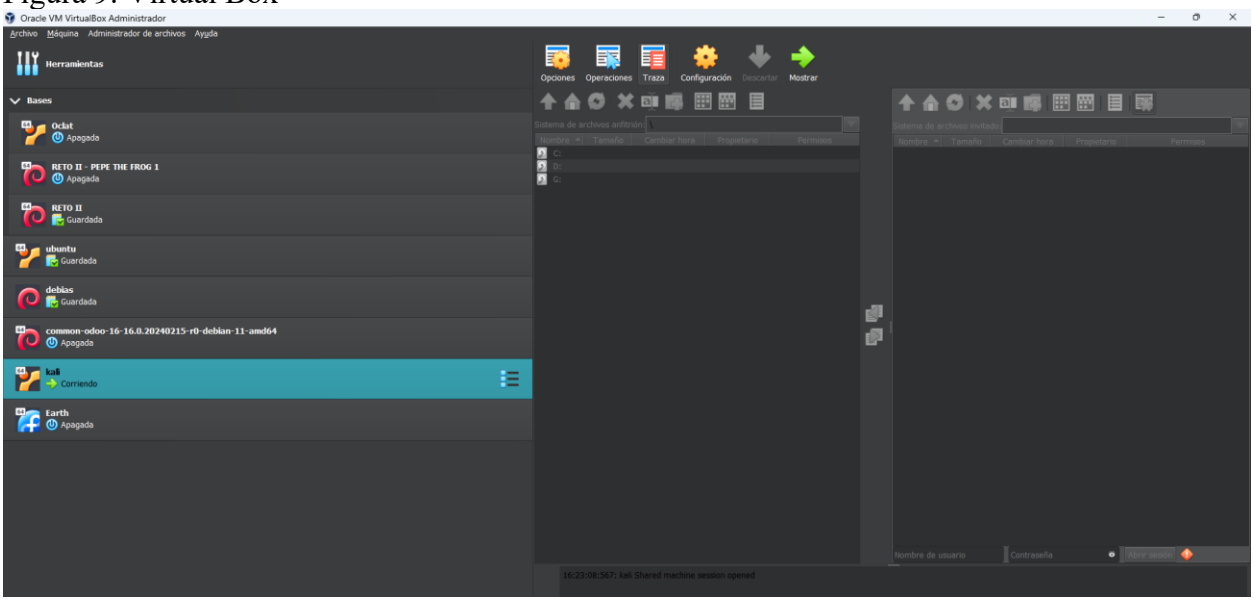
Dado que el objetivo del trabajo de investigación es el análisis de tráfico de red para la detección y defensa de ataques de fuerza bruta, el ataque de credenciales es un enfoque relevante y representativo. El estudio de este tipo de ataque permitirá desarrollar técnicas y soluciones aplicables a un amplio espectro de sistemas y aplicaciones que requieren autenticación.

## 3.5.Entorno de prueba

### 3.5.1. Instalación Virtual Box

La instalación de VirtualBox en un sistema es un proceso que consiste en descargar el instalador adecuado del sitio web oficial, ejecutarlo y seguir las instrucciones del asistente de instalación. Durante este proceso, se pueden configurar opciones como la aceptación de la licencia y las preferencias de instalación. También es posible instalar extensiones adicionales para habilitar características como el soporte USB o la aceleración 3D. Una vez completada la instalación, VirtualBox estará listo para su uso y se puede iniciar desde el menú de inicio del sistema. Es importante asegurarse de cumplir con los requisitos mínimos de hardware y tener suficiente espacio en disco para las máquinas virtuales que se deseen crear.

Figura 9. Virtual Box

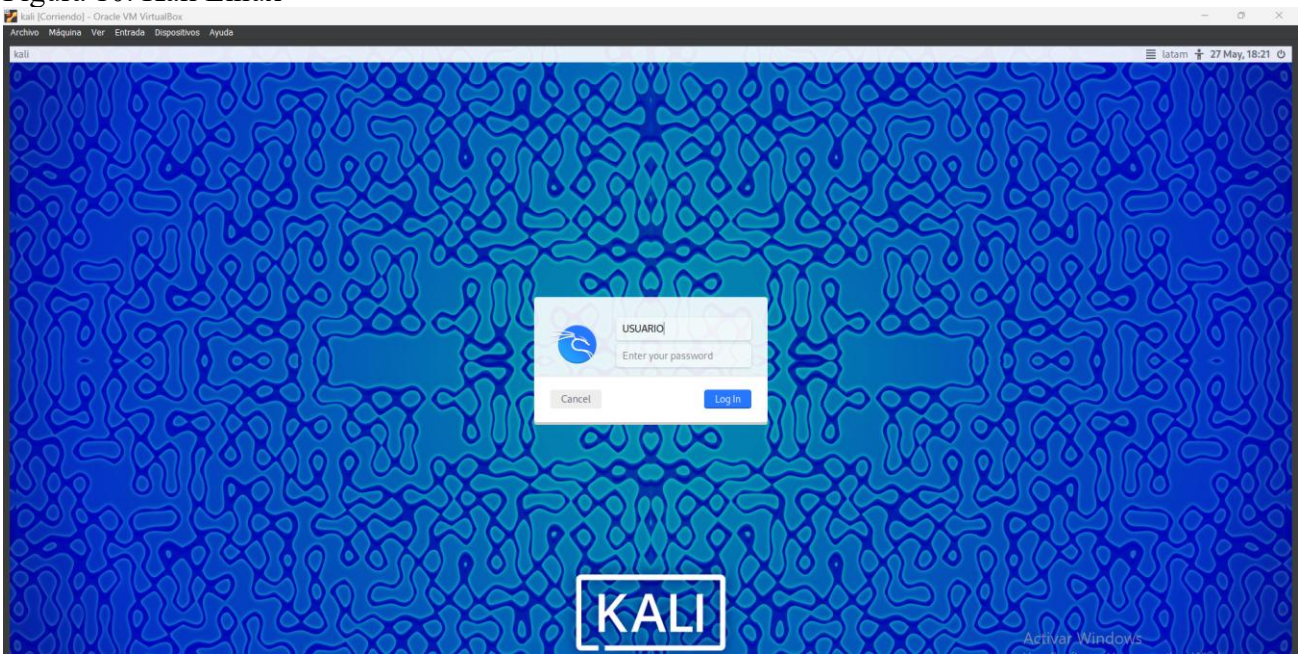



Nota: Representación de la instalación de Kali en VirtualBox

### 3.5.2. Instalación Kali Linux

La instalación de Kali Linux en VirtualBox es un proceso que implica descargar la imagen de Kali Linux, crear una nueva máquina virtual en VirtualBox, asignar la imagen descargada como el medio de instalación, configurar las opciones de la máquina virtual y finalmente proceder con la instalación de Kali Linux. Una vez completado el proceso de instalación, Kali Linux estará listo para ser utilizado en VirtualBox como un sistema operativo virtualizado.

Figura 10. Kali Linux

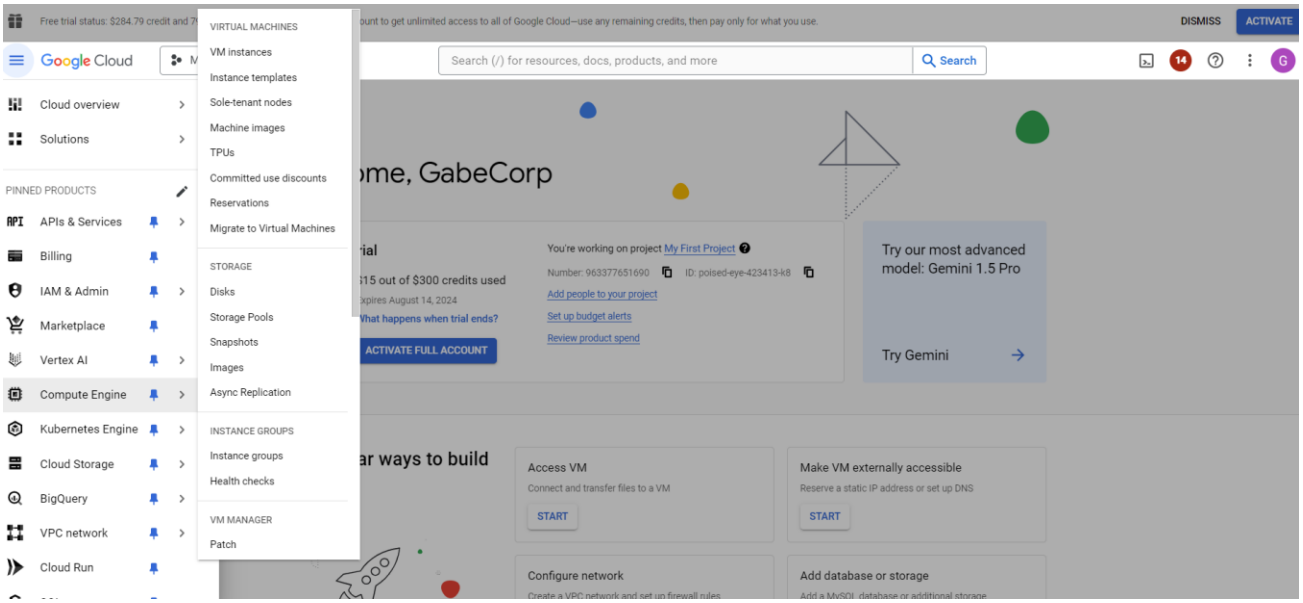



Nota: Pantalla de Inicio de Kali

### 3.5.3. Implementación de Máquina Virtual de defensa nube Google

Para acceder a los recursos de Google Cloud, primero se debe registrar una cuenta utilizando una dirección de correo electrónico de Gmail. Después, se requiere registrar una tarjeta de crédito, la cual no será cargada con ningún costo adicional. Al hacer esto, tendrás la posibilidad de utilizar ciertas herramientas de Google Cloud de forma gratuita durante un periodo de 90 días, o hasta agotar un límite de 300 dólares en créditos. Estos recursos te permitirán realizar pruebas o simulaciones de máquinas virtuales para familiarizarte con la plataforma y sus funcionalidades (Google Cloud, 2024).

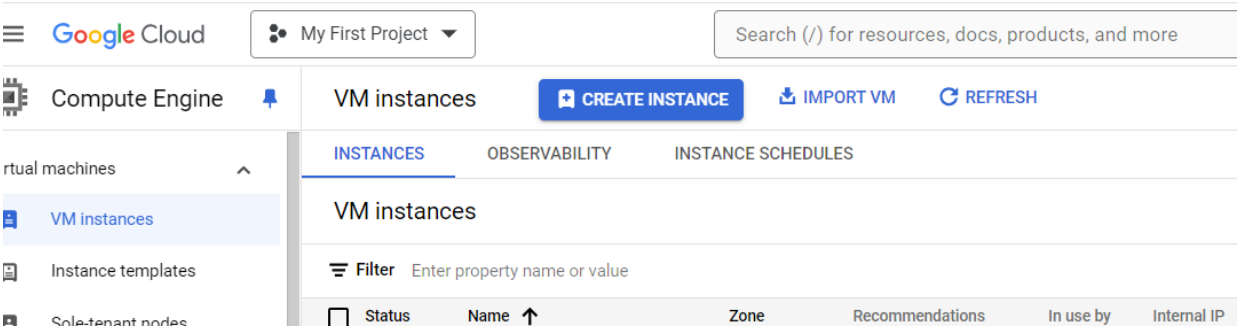
Figura 11. Virtualización de Pruebas Google Cloud




Nota: Representación de la pantalla de inicio de Google cloud

Para ingresar a una Instancia de Google Cloud, deberá hacer clic en la opción "Crear Instancia". Esto permitirá desplegar las pantallas y configuraciones necesarias.

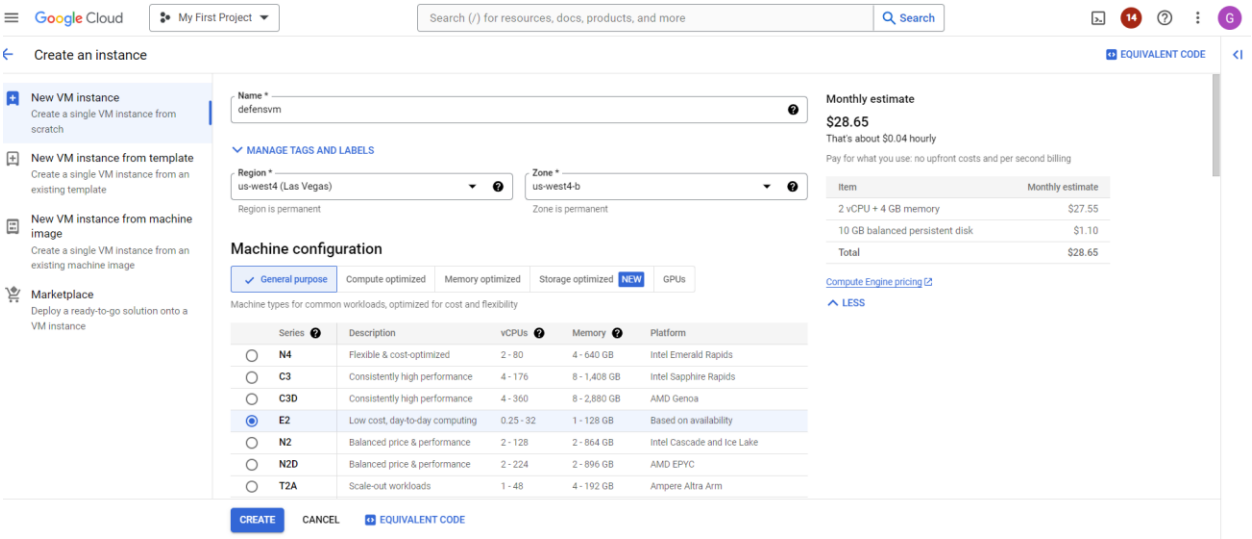
Figura 12. Creación de instancia de Pruebas Google Cloud




Nota: Pantalla de inicio de máquinas virtuales de Google cloud Autor

Luego se despliega una pantalla con las configuraciones de la máquina virtual, seleccionando el tipo de memoria, zona donde se encontrará el servidor, el sistema operativo la cual se seleccionó seria Debian GNU/Linux 12 para esta simulación.

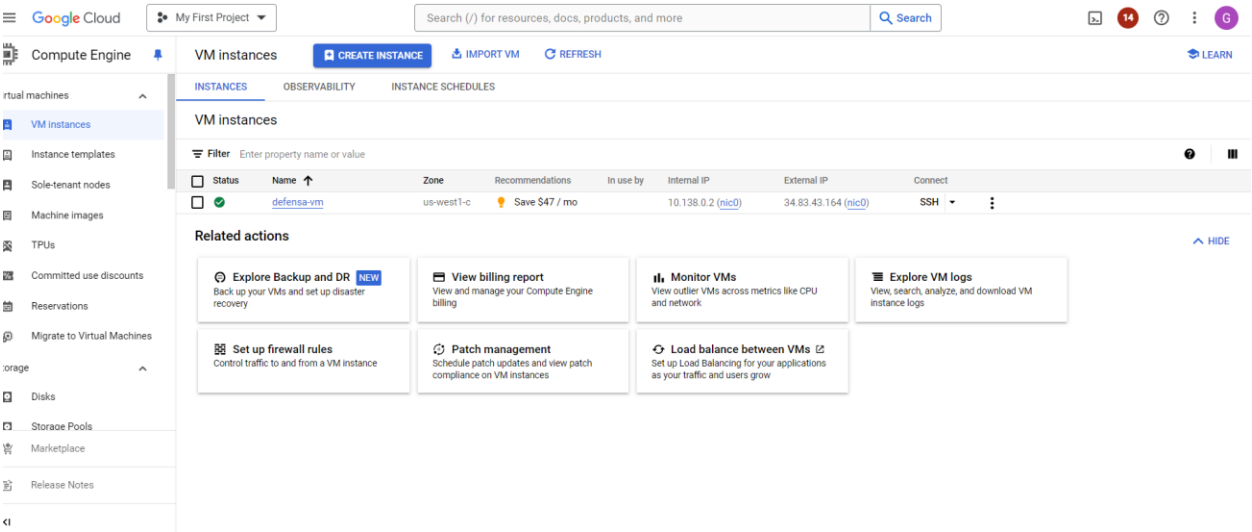
Figura 13. Instancia de Pruebas Google Cloud




Nota: Configuración de instancias de la máquina virtual

Ya instalada la máquina virtual como se muestra a continuación se instalará un sistema que permita atacar por medio de Brute Force.

Figura 14. crear instancia de Pruebas Google Cloud




Nota: Instancia de Google cloud

# CAPITULO IV PROPUESTA DE ATAQUE Y DEFENSA APLICADA AL PROTOCOLO FTP

El presente capítulo tiene como objetivo desarrollar una propuesta de ataque y defensa aplicada al protocolo FTP (File Transfer Protocol). Esta propuesta busca analizar la seguridad de los sistemas que utilizan este protocolo, identificar vulnerabilidades y establecer estrategias efectivas para su detección y mitigación.

En primer lugar, se realizará un análisis exhaustivo de los puertos abiertos en el sistema objetivo, utilizando herramientas como nmap para identificar los servicios activos y evaluar su nivel de exposición. A partir de este análisis, se procederá a ejecutar un ataque de fuerza bruta contra el servidor FTP, empleando la herramienta Hydra y explorando diferentes parámetros y configuraciones para optimizar el proceso.

Posteriormente, se diseñará una metodología de defensa que permita la detección temprana de este tipo de ataques. Para ello, se desarrollará un script en Python que utilizará la biblioteca Scapy para monitorear el tráfico de red y reconocer patrones de intentos de inicio de sesión FTP fallidos, característicos de los ataques de fuerza bruta.

Finalmente, se analizarán las recomendaciones y conclusiones derivadas de este proceso, con el fin de brindar pautas y estrategias que puedan ser aplicadas en entornos reales para fortalecer la seguridad de los sistemas que utilizan el protocolo FTP.

## 4.1 Ejecución del ataque de fuerza bruta

### 4.1.1 Análisis de Puertos

El análisis de puertos es una etapa fundamental en las evaluaciones de seguridad, ya que permite identificar los servicios y aplicaciones que se encuentran activos en un determinado sistema. Esto es crucial para detectar posibles puntos de entrada que puedan ser explotados por atacantes, así como vulnerabilidades que puedan comprometer la integridad y confidencialidad de la información. Conocer la configuración de puertos de un sistema objetivo es el primer paso para comprender su superficie de ataque y poder diseñar estrategias de defensa efectivas. Mediante el análisis de puertos, se puede determinar qué servicios se encuentran expuestos a la red, sus versiones de software y su nivel de exposición a vulnerabilidades conocidas.

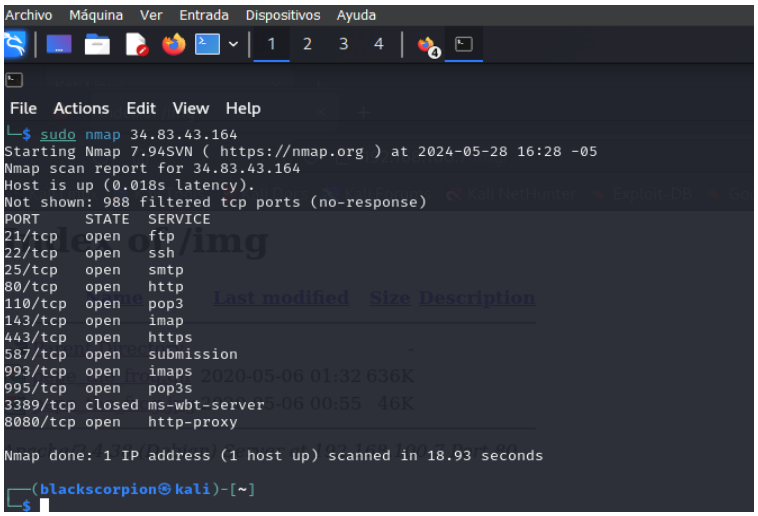
En el contexto de esta propuesta, el análisis de puertos adquiere especial relevancia, ya que permitirá identificar la presencia y configuración del servicio FTP en el sistema objetivo. Esto sentará las bases para la posterior ejecución del ataque de fuerza bruta y el diseño de la metodología de defensa correspondiente.

A continuación, se detallará el proceso de escaneo de puertos utilizando la herramienta nmap en el entorno de Kali Linux, con el fin de obtener una visión integral de los servicios activos en el sistema y sus potenciales debilidades. Para iniciar el ataque se debe conocer las debilidades del servidor o la víctima, para ello se va a utilizar la herramienta conocida como nmap. Al utilizar nmap en Kali Linux para analizar los puertos, es esencial realizar ciertos pasos para comprender y evaluar los resultados de manera efectiva. Aquí se presenta a detalle, para llevar a cabo un análisis de puertos con nmap en Kali Linux:

### 4.1.2. Escaneo inicial de puertos

El primer paso en el análisis de puertos es realizar un escaneo básico para identificar los servicios activos en un host específico. Para ello, se utiliza la herramienta nmap, que es una de las más populares y completas para el escaneo y análisis de redes. El comando básico para ejecutar un escaneo de puertos con nmap es:

Figura 15. escaneo nmap




Fuente: Autor

Donde <34.83.43.164> es la dirección IP del host que se desea analizar. Este comando realizará un escaneo rápido de los puertos comúnmente utilizados (los 1000 puertos TCP más populares) y mostrará los puertos que se encuentran abiertos en el sistema objetivo.

Se puede evidenciar los puertos abiertos del servidor víctima, la cuales presenta los siguientes puertos:

- Puerto 22 (SSH): Es el puerto estándar utilizado para el protocolo de Secure Shell (SSH), que permite una conexión segura y cifrada para administrar sistemas de forma remota.
- Puerto 25 (SMTP): Es el puerto comúnmente utilizado para el protocolo de Transferencia de Correo Simple (SMTP), que facilita el envío de correos electrónicos a través de la red.
- Puerto 80 (HTTP): Es el puerto estándar para el protocolo de Transferencia de Hipertexto (HTTP), utilizado para acceder a sitios web y visualizar contenido en Internet.
- Puerto 110 (POP3): Corresponde al puerto utilizado por el protocolo de Oficina de Correos 3 (POP3), que permite a los clientes de correo electrónico recuperar mensajes de un servidor de correo.
- Puerto 143 (IMAP): Este puerto se destina al protocolo de Acceso a Mensajes de Internet (IMAP), que ofrece funcionalidades avanzadas para gestionar correos electrónicos en servidores.
- Puerto 443 (HTTPS): Es el puerto utilizado para el protocolo de Transferencia de Hipertexto Seguro (HTTPS), que proporciona una capa adicional de cifrado de datos en las comunicaciones web.
- Puerto 563: Se emplea comúnmente para el protocolo de Mensajería Segura sobre NNTP (NNTPS), utilizado para la transmisión segura de mensajes en grupos de noticias en red.
- Puerto 587: Asignado al protocolo de Transferencia Simple de Correo (SMTP) en modo mensaje sumamente confiable (MSA), utilizado para el envío de correos electrónicos de salida de manera segura.
- Puerto 993 (IMAPS): Se utiliza para el protocolo de Acceso a Mensajes de Internet con cifrado (IMAPS), que permite el acceso seguro a buzones de correo usando el protocolo IMAP.

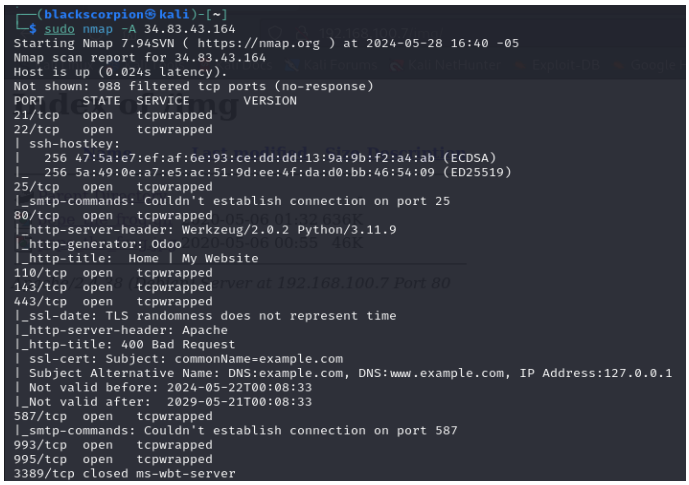
- Puerto 995 (POP3S): Este puerto se destina al protocolo POP3 con cifrado (POP3S), que permite a los clientes de correo electrónico recuperar mensajes de forma segura a través de una conexión cifrada.
- Puerto 8080: Comúnmente utilizado como un puerto de reserva para servidores web, permite el acceso a aplicaciones web alojadas en un servidor a través de direcciones URL personalizadas.
- Puerto 21 (FTP): El puerto 21 se utiliza para el protocolo de Transferencia de Archivos (FTP), que permite la transferencia de archivos entre un cliente y un servidor a través de una red.

Esta información inicial es crucial para comprender la superficie de ataque del sistema y poder identificar posibles vulnerabilidades que puedan ser explotadas en ataques de fuerza bruta. Es importante tener en cuenta que este escaneo básico es solo el punto de partida y que se deben realizar análisis más exhaustivos para obtener una imagen más completa de la seguridad del sistema.

### 4.1.3. Escaneo avanzado de puertos

Realiza un escaneo más exhaustivo para obtener información detallada sobre los puertos abiertos, los servicios en ejecución y las versiones de software del host. Para ello, puedes utilizar el comando \[nmap -A [dirección IP]\] que efectúa un análisis en profundidad del sistema. Este análisis más completo te proporcionará un mayor nivel de detalle sobre la configuración y el estado del host, lo cual puede ser de gran utilidad para comprender mejor su infraestructura y posibles puntos débiles a considerar en futuras evaluaciones de seguridad.

*Figura 16 Escaneo Avanzado de puertos*




Fuente: Autor

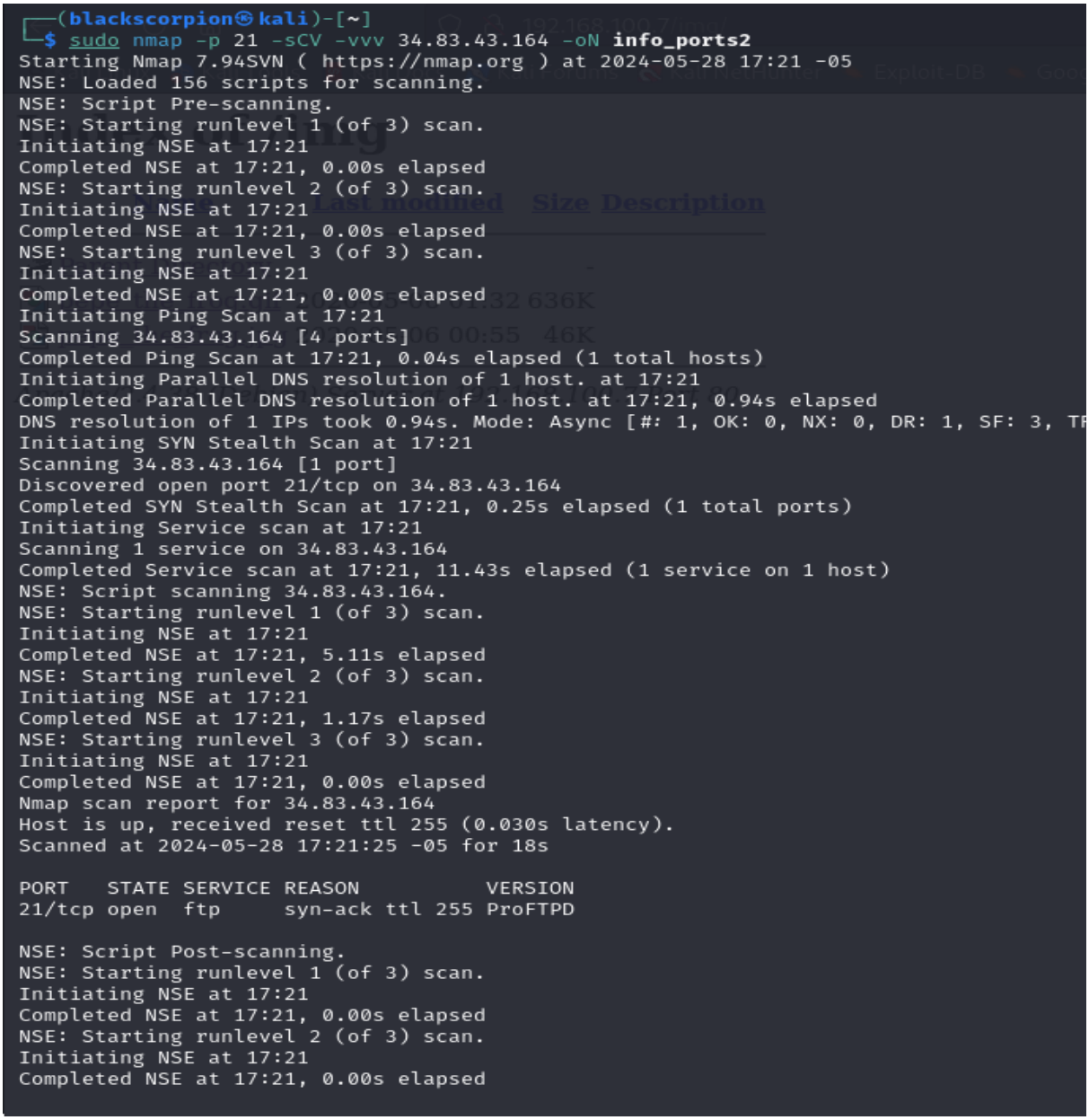
Este comando analiza información detallada sobre los servicios y puertos abiertos en la dirección IP 34.83.43.164. Aquí se presentan algunos puntos destacados basados en el reporte:

- La dirección IP en cuestión parece tener varios puertos abiertos, como el 21 (FTP), 22 (SSH), 80 (HTTP), 110 (POP3), 143 (IMAP), 443 (HTTPS), 587 (SMTP), 993 (IMAPS), 995 (POP3S) y 8080, así como algunos puertos cerrados como el 3389 (MS-WBT-SERVER).
- Algunos de los servicios como FTP, SSH, HTTP, HTTPS, SMTP, IMAP, y otros parecen estar "tcpwrapped", lo que a menudo indica que hay algún tipo de filtro o proxy entre el escáner y el servicio real, lo que puede dificultar la evaluación precisa de los servicios.
- Se identificó que el servidor HTTP en el puerto 80 está utilizando Werkzeug/2.0.2 y Python/3.11.9. Además, la página principal indica que es Odoo según el generador HTML. El servidor HTTPS en el puerto 443 parece estar utilizando Apache, pero se muestra un título de "400 Bad Request". El certificado SSL parece ser para "example.com" y tiene ciertas advertencias sobre su validez.
- En el puerto 8080, se identificó un servidor Apache/2.4.59 (Debian) con un título de "Apache2 Debian Default Page: It works".

Una vez que se tenga una perspectiva de los puertos abiertos, se podremos utilizar una herramienta de escaneo directamente en los puertos abiertos. A continuación, escaneamos el puerto FTP, que es el 21.

*sudo nmap -p 21 -sCV -vvv 34.83.43.164 -oN info_ports2*

Figura 17 Escáner de Puertos




Fuente: Autor

Analizando el escaneo de nmap se puede hacer referencia a la siguientes linead más importantes:

a) Starting Nmap 7.94SVN (https://nmap.org) at 2024-05-28 17:21 -05: Esta línea muestra la versión de Nmap utilizada y la hora en que comenzó el escaneo.

b) NSE: Loaded 156 scripts for scanning: Nmap cargó 156 scripts (escrituras) para realizar el escaneo. Estos scripts proporcionan funcionalidades adicionales, como detección de servicios específicos o vulnerabilidades.
c) Initiating NSE at 17:21: Nmap inicia la ejecución de los scripts NSE (Nmap Scripting Engine).
d) Completed NSE at 17:21, 0.00s elapsed: Indica que la ejecución de los scripts NSE se completó en 0 segundos.
e) Discovered open port 21/tcp on 34.83.43.164: Nmap encontró un puerto abierto en el host con dirección IP 34.83.43.164. El puerto 21 está asociado al servicio FTP (Protocolo de transferencia de archivos).
f) Completed SYN Stealth Scan at 17:21, 0.25s elapsed (1 total ports): El escaneo SYN Stealth se completó en 0.25 segundos. Este escaneo se utiliza para determinar si un puerto está abierto o cerrado.
g) Scanning 1 service on 34.83.43.164: Nmap está escaneando un servicio en el host.
h) Completed Service scan at 17:21, 11.43s elapsed (1 service on 1 host): El escaneo de servicios se completó en 11.43 segundos. Esto implica que Nmap identificó un servicio en el host.
i) Nmap scan report for 34.83.43.164: Informa sobre el host escaneado y su dirección IP.
j) 21/tcp open ftp syn-ack ttl 255 ProFTPD: Indica que el puerto 21 está abierto y que se está ejecutando el servicio FTP (ProFTPD). ProFTPD es un servidor FTP de código abierto.

La importancia de este análisis detallado del reporte de escaneo de puertos radica en los siguientes aspectos:

#### *4.1.3.1.Identificación de la superficie de ataque:*

El escaneo de puertos revela los servicios y -aplicaciones que se ejecutan en el sistema objetivo, lo que permite comprender su superficie de ataque. Conocer los puertos abiertos y los servicios en ejecución es fundamental para evaluar las posibles vulnerabilidades y puntos de entrada que un atacante podría explotar.

#### *4.1.3.2.Detección de servicios y versiones vulnerables:*

El análisis de los detalles de los servicios, como las versiones de software y las tecnologías utilizadas, permite identificar potenciales vulnerabilidades conocidas. Esto es crucial para priorizar y abordar las vulnerabilidades más críticas y mantener el sistema actualizado y seguro.

#### 4.1.3.3.Comprensión de la arquitectura del sistema:

La información obtenida sobre los puertos abiertos, los servicios en ejecución y la posible presencia de filtros o proxies, proporciona una visión general de la arquitectura del sistema. Esta comprensión es valiosa para planificar futuras evaluaciones de seguridad y diseñar estrategias de mitigación de riesgos.

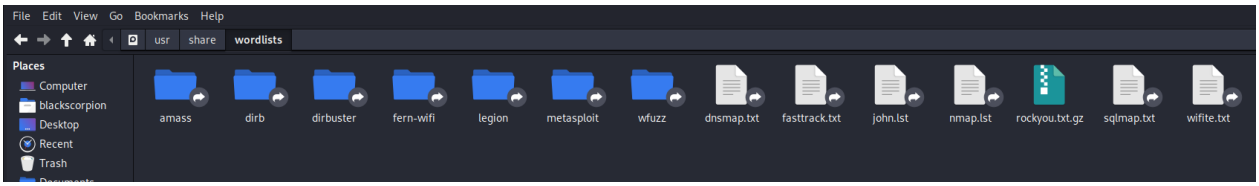
El análisis detallado del reporte de escaneo de puertos es fundamental para comprender la superficie de ataque, identificar vulnerabilidades, evaluar la arquitectura del sistema y garantizar el cumplimiento de los requisitos de seguridad. Esta información es crucial para implementar medidas de seguridad efectivas y mantener el sistema protegido contra posibles ataques.

### 4.1.4. Ataque de ftp

El escaneo de puertos es una técnica fundamental en las evaluaciones de seguridad, ya que permite a los auditores y expertos en ciberseguridad identificar los servicios y aplicaciones que se ejecutan en un sistema objetivo. Uno de los hallazgos más relevantes de este proceso fue la detección del puerto 21, asociado al servicio FTP, abierto en la dirección IP 34.83.43.164. Ante este descubrimiento, se procedió a analizar en detalle la importancia de este hallazgo y las implicaciones de realizar un ataque de fuerza bruta contra el servidor FTP. Primero ubicamos el directorio donde están los diccionarios de ataque la cual se muestra a continuación:

*Usr/share/wordlist*

*Figura 18 ubicación de diccionarios de ataque Kali*



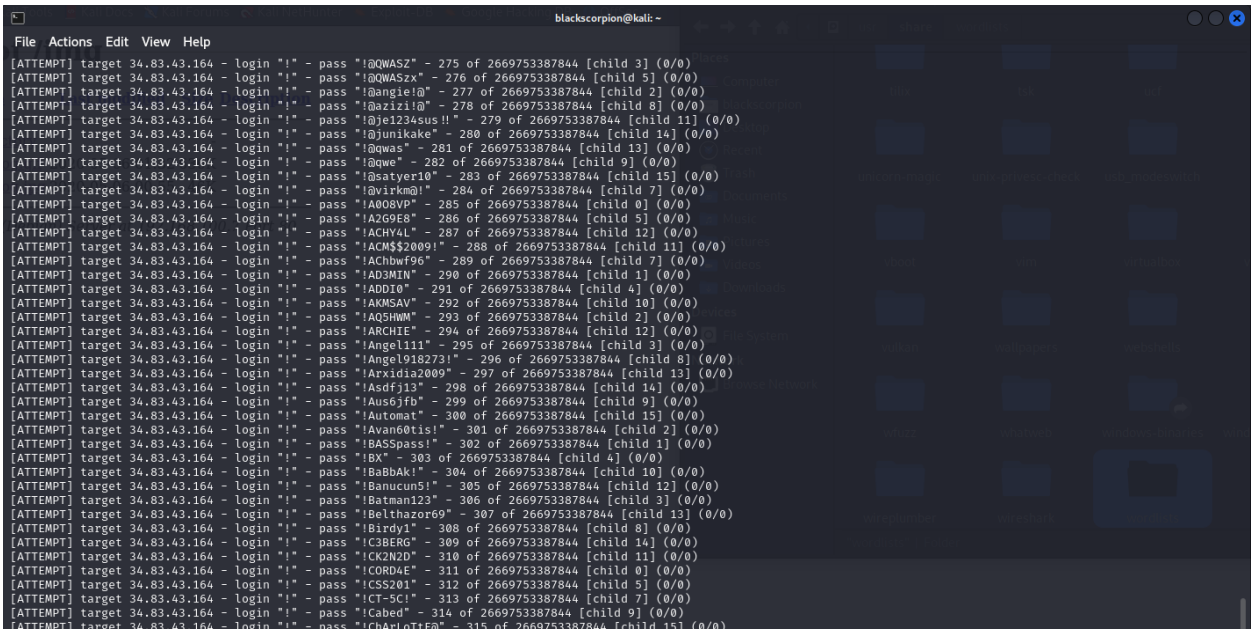

Fuente: Autor

Este comando de Hydra está realizando un ataque de fuerza bruta contra el servidor FTP en la dirección IP 34.83.43.164, utilizando los diccionarios de usuarios y contraseñas ubicados en "/usr/share/wordlists/sqlmap.txt". El objetivo es encontrar una combinación de credenciales válida que permita acceder al servicio FTP. La información y la opción de detenerse una vez se encuentre un acceso válido ayudan a optimizar el proceso y obtener los resultados deseados.

Luego ingresamos el siguiente comando Hydra, para comenzar el ataque

*hydra -s 21 34.83.43.164 -L /usr/share/wordlists/sqlmap.txt -P /usr/share/wordlists/sqlmap.txt –vV –F ftp*

*Figura 19 Proceso de Ataque FTP*




Fuente: Autor

Ya con este proceso se cumple el objetivo de ataque donde se describe a continuación cada elemento del código tipeado para el ataque:

#### *4.1.4.1. Comando s 21"*

Este parámetro especifica el puerto a utilizar, en este caso, el puerto 21 que es el puerto estándar para el servicio FTP. "34.83.43.164": Esta es la dirección IP del servidor FTP que se va a atacar. En este caso, el valor "21" corresponde al puerto estándar para el servicio FTP (File Transfer Protocol), Al especificar el puerto 21, le indicamos a Hydra que debe dirigir el ataque de fuerza

bruta al servicio FTP que se está ejecutando en ese puerto. Si lo combinamos con la ip "34.83.43.164", el parámetro define la dirección IP del servidor FTP que será el objetivo del ataque, identifica de manera única el sistema informático que está ejecutando el servicio FTP. Al proporcionar esta dirección IP, le indicamos a Hydra que debe dirigir el ataque de fuerza bruta específicamente al servidor FTP ubicado en esa dirección IP.

#### *4.1.4.2. Comandos de listar diccionario de contraseña y usuario*

**-L /usr/share/wordlists/sqlmap.txt**

Este parámetro indica que Hydra debe utilizar el archivo "/usr/share/wordlists/sqlmap.txt" como lista de nombres de usuario (usernames) para intentar iniciar sesión como se presenta en la (Ver Figura 19), de igual forma se usara para el parámetro de contraseña el mismo archivo, aunque se puede usar otro que el atacante prefiera como se presenta en el siguiente ítem.

**-P /usr/share/wordlists/sqlmap.txt**

Este parámetro indica que Hydra debe utilizar el mismo archivo "/usr/share/wordlists/sqlmap.txt" como lista de contraseñas para intentar iniciar sesión.

Si se utiliza la –L o la letra –P en mayúscula la consola lo tomara como que no se conoce la contraseña o el usuario y seguido de ella leerá el archivo txt que continue, pero si se utilizara la minúscula el sistema lo tomara como que ya se tiene la contraseña o el usuario y continuo se coloca la contraseña o el usuario que se conozca.

#### *4.1.4.3. Comando -Vv*

Este parámetro activa el modo verboso (verbose) de Hydra, lo que significa que la herramienta proporcionará información detallada y en tiempo real sobre el progreso y los resultados del ataque de fuerza bruta. El modo verboso es fundamental para monitorear y analizar el desarrollo del ataque, ya que permite visualizar: Las combinaciones de usuario y contraseña que se van probando. El estado de cada intento de autenticación (éxito, fallo, etc.), la tasa de intentos por segundo, Cualquier otro mensaje o alerta relevante del proceso.

Tener esta información detallada facilita la comprensión del ataque y la identificación de patrones o hallazgos significativos. Esto es crucial para poder evaluar de manera efectiva la seguridad del sistema FTP y documentar adecuadamente los resultados de la prueba. El modo

verboso también ayuda a optimizar el proceso, al permitir detectar si se ha encontrado una credencial válida y detener el ataque en ese momento, evitando intentos innecesarios.

#### *4.1.4.4. Comando -F"*

Este parámetro le indica a Hydra que deje de intentar más combinaciones de usuario y contraseña una vez que se haya encontrado una credencial válida que permita iniciar sesión en el servidor FTP. La inclusión de este parámetro es importante por varias razones una de ellas es la optimización del proceso, al detener el ataque de fuerza bruta una vez se ha encontrado una combinación exitosa, se evita malgastar recursos y tiempo en intentos adicionales innecesarios otra busca la eficiencia en la recopilación de hallazgos, al detenerse después del primer acceso válido, Hydra puede centrarse en reportar y documentar ese hallazgo, en lugar de seguir buscando posibles credenciales adicionales. También busca la reducción del impacto en el sistema objetivo: Finalizar el ataque de manera oportuna limita el número de intentos de autenticación realizados, lo cual minimiza el riesgo de saturar o bloquear el servicio FTP.

### 4.2. Metodología de Defensa “Plan de defensa de Ataques”

Una vez realizado el análisis de puertos y ejecutado el ataque de fuerza bruta contra el servidor FTP, es fundamental contar con una metodología de defensa efectiva que permita detectar y mitigar este tipo de ataques. Dicha metodología debe ser integral y abarcar diferentes capas de seguridad, con el objetivo de proteger proactivamente los sistemas y minimizar los riesgos.

El plan de defensa propuesto en este capítulo se enfoca en la detección temprana de los ataques de fuerza bruta contra el protocolo FTP. Para ello, se desarrollará un script en Python que utilizará la biblioteca Scapy para monitorear el tráfico de red y reconocer patrones de intentos de inicio de sesión fallidos, característicos de este tipo de ataques.

Al implementar este script de detección, se busca lograr los siguientes objetivos:

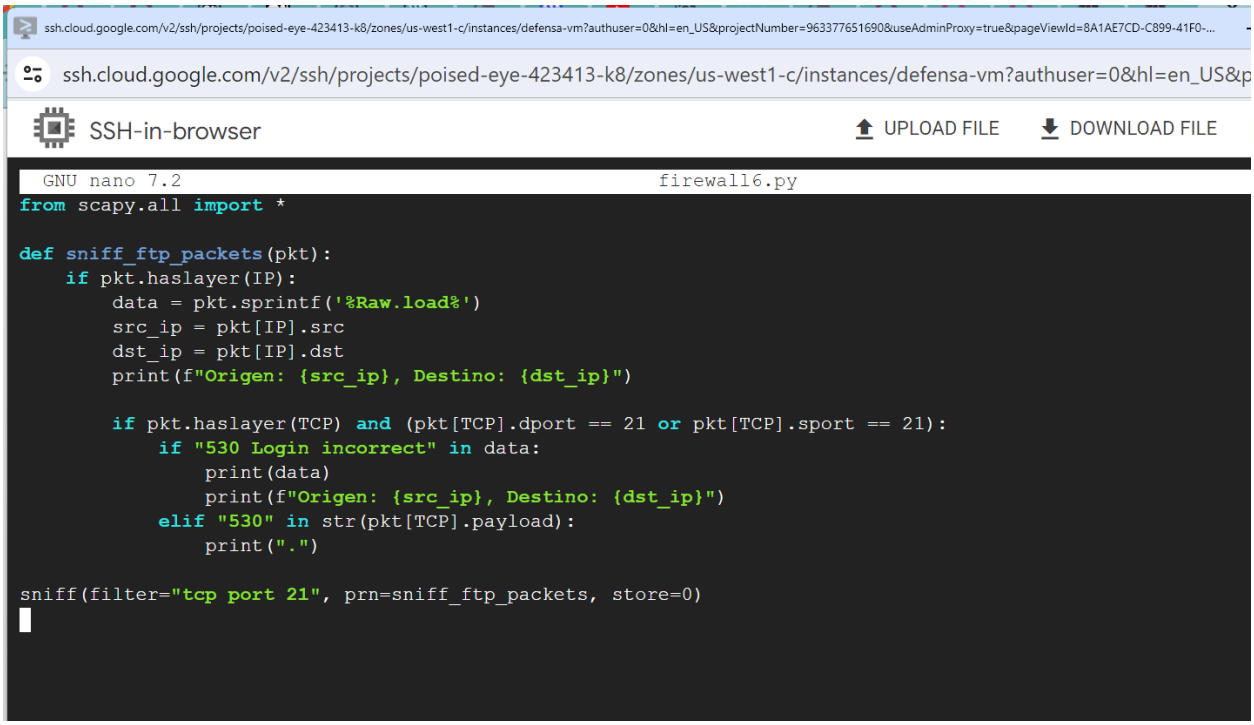
- Identificar en tiempo real los intentos de ataques de fuerza bruta contra el servidor FTP.
- Generar alertas y notificaciones que permitan a los administradores de seguridad responder de manera oportuna.
- Recopilar información detallada sobre los intentos de ataque, como la dirección IP de origen, los usuarios y contraseñas utilizados, y otros detalles relevantes.

A continuación, iniciaremos el plan de detección y defensa contra ataques de fuerza bruta. Se ha seleccionado un ataque a un servidor con el puerto de FTP abierto.

### 4.2.1. Detección de Ataque de Fuerza Bruta en FTP

Para la detección de los ataques de fuerza bruta dirigidos al servidor FTP, se ha desarrollado un script en lenguaje Python que utiliza la biblioteca Scapy para monitorear el tráfico de red. A continuación, se detalla el funcionamiento del script:

*Figura 20 Script Python de Detección de Intrusos*




Fuente: Autor

#### *4.2.1. from scapy.all import*

Esta línea importa todas las funciones y clases del módulo scapy.all, que es una poderosa biblioteca de Python utilizada para manipular y analizar paquetes de red. Scapy permite capturar, inspeccionar, modificar y enviar paquetes a través de la interfaz de red. Esto lo convierte en una herramienta fundamental para el desarrollo de soluciones de seguridad, como la detección de ataques de fuerza bruta en FTP. Al importar * (asterisco), se están cargando todas las funciones y

clases definidas en el módulo scapy.all, lo que facilita el acceso a las principales funcionalidades de la biblioteca sin tener que especificar el nombre del módulo cada vez.

#### *4.2.2. def sniff_ftp_packets(pkt)*

La función principal definida en el código es sniff_ftp_packets(pkt), que está diseñada para analizar paquetes de red. Verifica si el paquete tiene una capa correspondiente a IP y luego extrae la dirección IP de origen (src_ip) y la dirección IP de destino (dst_ip). Imprime las direcciones IP de origen y destino.

La función también verifica si el paquete tiene una capa TCP y específicamente si el puerto de destino TCP (dport) o el puerto de origen TCP (sport) es 21, que es el puerto estándar para FTP (Protocolo de Transferencia de Archivos). Luego busca cadenas específicas en los datos del paquete, como "530 Login incorrect" y "530 ", para identificar intentos fallidos de inicio de sesión FTP u otros mensajes relacionados con FTP, e imprime la información relevante.

#### *4.2.3. if pkt.haslayer(IP)*

La línea de código `if pkt.haslayer(IP)` verifica si el paquete `pkt` tiene una capa correspondiente al protocolo IP (Internet Protocol). Esta condición se utiliza para comprobar si el paquete capturado contiene información de capa de red a nivel IP. En Python y en el contexto de procesamiento de paquetes de red, la función `haslayer()` se utiliza para determinar si un paquete de red contiene una capa específica. En este caso, la capa que se está verificando es la capa IP, que es esencial para la comunicación en redes IP.

Al utilizar `pkt.haslayer(IP)`, el código está evaluando si el paquete `pkt` contiene información de la capa IP. Si la condición es verdadera, significa que el paquete tiene una capa IP y el código dentro del bloque `if` asociado a esta condición se ejecutará. En caso contrario, si el paquete no tiene una capa IP, el código dentro del bloque `if` no se ejecutará. Esta verificación es importante en el contexto de análisis de paquetes de red, ya que permite realizar acciones específicas basadas en la presencia o ausencia de ciertas capas en los paquetes capturados.

#### *4.2.3.4. data = pkt.sprintf('%Raw.load%')`*

La línea de código `data = pkt.sprintf('%Raw.load%')` se utiliza para extraer y almacenar los datos brutos (raw data) del paquete `pkt`. En este caso, la función `sprintf()` se utiliza para formatear y obtener el contenido de la carga útil (payload) del paquete en su forma cruda.

Al utilizar `%Raw.load%` como argumento en `sprintf()`, se está indicando que se desea obtener la carga útil sin procesar del paquete, es decir, los datos sin ningún tipo de interpretación o decodificación adicional. La carga útil de un paquete de red contiene la información transmitida, como el cuerpo de un mensaje, archivo o comando en el caso de protocolos como HTTP, FTP, SSH, entre otros.

Por lo tanto, al ejecutar esta línea de código, la variable `data` almacenará los datos brutos de la carga útil del paquete `pkt`, lo que permite acceder y manipular directamente la información transmitida en el paquete de red sin realizar ningún procesamiento adicional.

#### *4.2.3.5. src_ip = pkt[IP].src*

La línea de código `src_ip = pkt[IP].src` se utiliza para extraer y almacenar la dirección IP de origen (source IP) del paquete `pkt`. En este caso, se accede al campo `src` de la capa IP dentro del paquete para obtener la dirección IP de origen del paquete de red. Al utilizar `pkt[IP].src`, se está accediendo al campo `src` (fuente) de la capa IP dentro del paquete `pkt`, lo que permite obtener la dirección IP de origen del paquete IP. La dirección IP de origen es fundamental en la comunicación de red, ya que identifica el dispositivo desde el cual se originó el paquete. Al asignar esta dirección IP de origen a la variable `src_ip`, se puede utilizar posteriormente en el código para realizar operaciones adicionales o para imprimir, analizar o procesar la dirección IP de origen del paquete de red capturado.

#### *4.2.3.6. dst_ip = pkt[IP].dst*

**pkt[IP]:** Esta parte del código está accediendo a la capa IP del paquete capturado. Scapy, la biblioteca utilizada en el script permite manipular y analizar las diferentes capas de un paquete de red. Al utilizar pkt[IP], se está accediendo específicamente a la capa IP del paquete.

**.dst:** Esta parte del código está extrayendo la dirección IP de destino del paquete. Cuando se accede a la propiedad .dst de un paquete IP, se obtiene la dirección IP de destino a la que se está dirigiendo ese paquete.

**dst_ip = :** Finalmente, el valor de la dirección IP de destino extraída se almacena en la variable dst_ip.

La razón por la que es importante obtener y almacenar la dirección IP de destino (dst_ip) es la siguiente:

**Identificación del servidor FTP atacado:** Al tener la dirección IP de destino, el script puede determinar a qué servidor FTP se están dirigiendo los intentos de acceso. Esto es crucial para enfocar el análisis y la detección en el servidor FTP que realmente está siendo atacado.

**Correlación de patrones de tráfico sospechoso:** Cuando el script detecta múltiples intentos de inicio de sesión FTP fallidos dirigidos a la misma dirección IP de destino (dst_ip), puede interpretar esto como un posible ataque de fuerza bruta en curso. La dirección IP de destino permite correlacionar estos intentos de acceso fallidos y reconocer patrones de actividad sospechosa.

**Seguimiento y bloqueo de direcciones IP:** Además de la detección, el script puede utilizar la información de dst_ip para llevar un registro de las direcciones IP que están intentando acceder al servidor FTP de manera sospechosa. Esto facilita el bloqueo de estas direcciones IP mediante reglas de firewall. La extracción y almacenamiento de la dirección IP de destino (dst_ip) en el script de detección de ataques de fuerza bruta en FTP es fundamental, ya que permite identificar el servidor FTP objetivo, analizar el tráfico dirigido a ese destino y tomar medidas de seguridad más efectivas.

**print(f"Origen: {src_ip}, Destino: {dst_ip}")`**

desempeña un papel importante al proporcionar visibilidad sobre el tráfico que está siendo analizado, facilitando la depuración, el seguimiento y la identificación de patrones que pueden ser relevantes para la detección de ataques de fuerza bruta en FTP.

**4.2.3.7 if (pkt.haslayer(TCP) and (pkt[TCP].dport == 21 or pkt[TCP].sport == 21)***:*

Las líneas de código presentadas tienen dos propósitos principales:

**Verificar la presencia de la capa TCP en el paquete**

**`pkt.haslayer(TCP)`**; revisa si el paquete capturado (`pkt`) tiene una capa correspondiente al Protocolo de Control de Transmisión (TCP). Esto es importante porque el script está enfocado en la detección de ataques de fuerza bruta en FTP, y FTP utiliza el protocolo TCP para la transferencia de archivos.

***`pkt[TCP].dport == 21 or pkt[TCP].sport == 21` comprueba si el puerto de destino TCP (`dport`)*** o el puerto de origen TCP (`sport`) del paquete es igual a 21. El puerto 21 es el puerto estándar utilizado por el Protocolo de Transferencia de Archivos (FTP).

La combinación de estas dos condiciones (`pkt.haslayer(TCP)` y `pkt[TCP].dport == 21 or pkt[TCP].sport == 21`) permite al script identificar los paquetes que están relacionados con el tráfico FTP.

Esto es crucial para el funcionamiento del script de detección de ataques de fuerza bruta, ya que el objetivo es analizar específicamente el tráfico dirigido al servidor FTP y buscar patrones de intentos fallidos de inicio de sesión, que podrían indicar un ataque en curso.

Al verificar la presencia de la capa TCP y que el puerto utilizado sea el 21 (el puerto estándar de FTP), el script se asegura de que está analizando únicamente los paquetes relevantes para la detección de ataques de fuerza bruta en FTP, descartando cualquier otro tráfico que no esté relacionado con este protocolo.

#### *4.2.3.8. if "530 Login incorrect" in data*

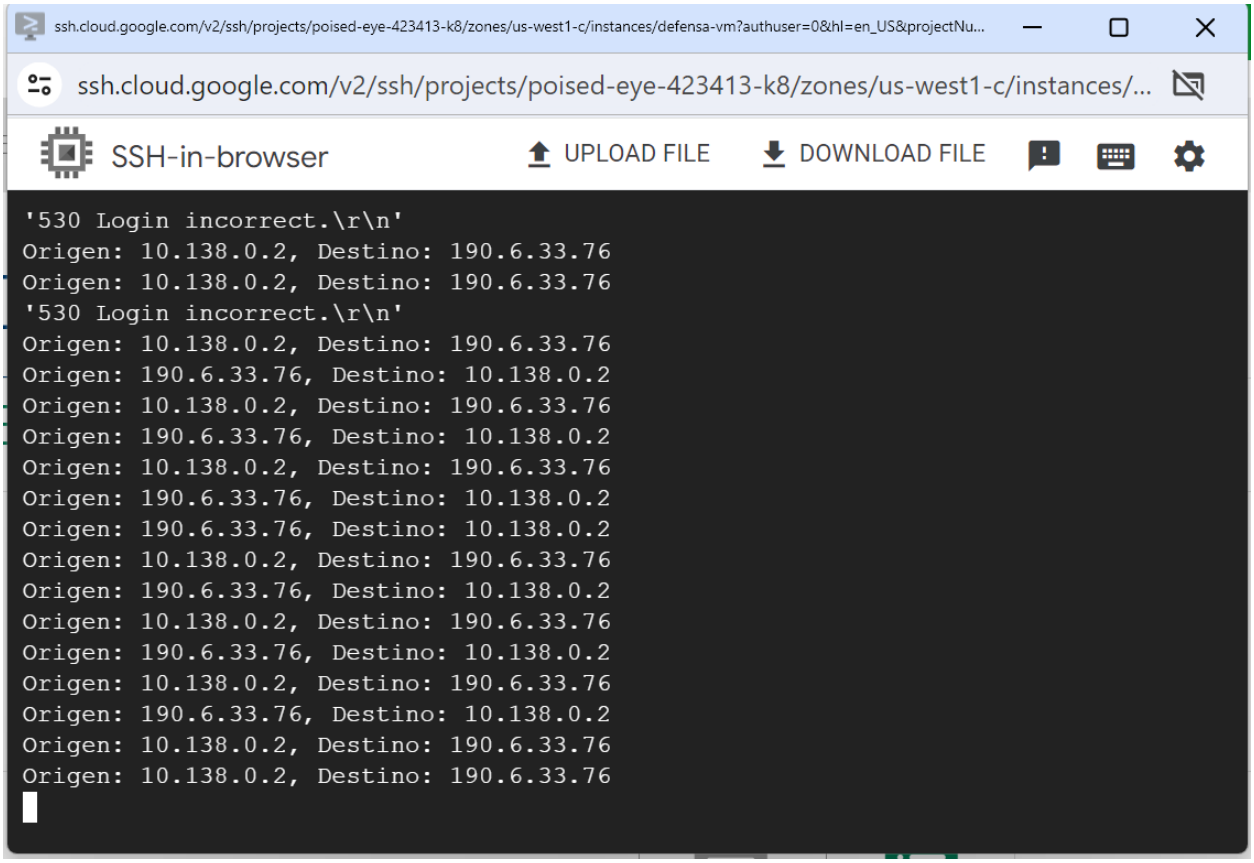
Esta línea verifica si la cadena de texto "530 Login incorrect" está presente en la variable data. La cadena "530 Login incorrect" es un mensaje de error comúnmente enviado por el servidor FTP cuando se produce un intento de inicio de sesión fallido. Al buscar esta cadena en el contenido del paquete capturado (data), el script puede identificar cuando se han producido intentos fallidos de inicio de sesión FTP.print(data): Si se cumple la condición anterior, esta línea imprime el contenido de la carga útil del paquete.

#### *1.2.3.9. sniff(filter="tcp port 21", prn=sniff_ftp_packets, store=0)*

Esta línea inicia la captura de paquetes de red, filtrando los paquetes TCP que tengan el puerto 21 (el puerto estándar para FTP) y pasando cada paquete capturado a la función `sniff_ftp_packets`. El parámetro `store=0` indica que los paquetes no se almacenarán en memoria.

Este código captura y analiza los paquetes de red que utilizan el protocolo FTP (puerto 21), buscando intentos de inicio de sesión FTP fallidos y mostrando información relevante sobre los paquetes. Este tipo de scrip es usado también como técnica de Sniffer para robar contraseñas, usuarios u otra información de la red, es por ello por lo que se recomienda tomar previsiones de acceso o restricciones a usuarios de la red.

*Figura 21. Análisis de Red*




Fuente: Autor

### 4.2.2. Mitigación y defensa

El siguiente código de Python puede ser utilizado en el contexto de la defensa de sistemas y redes para bloquear el acceso a ciertos recursos por parte de direcciones IP específicas. Las cuales podemos tomar del registro de detección anterior.

*Figura 22*

```
  GNU nano 7.2                                          defenza.py
import os

def block_ip_port(ip_address, port):
    os.system(f'sudo iptables -A INPUT -s {ip_address} -p tcp --dport {port} -j DROP')
    print(f'IP {ip_address} bloqueada para el puerto {port}')

# Ejemplo de cómo bloquear una IP específica para el puerto 21
ip_to_block = "190.6.33.76"
port_to_block = 21
block_ip_port(ip_to_block, port_to_block)
```

Fuente: Autor

A continuación, se analiza detalladamente el código:

#### *4.2.2.1.import os*

En Python, el módulo os es un componente incorporado que permite interactuar con el sistema operativo. Al importar os, se habilita el uso de las funciones y variables definidas en dicho módulo dentro del código. Esto posibilita, por ejemplo, la ejecución de comandos del sistema operativo, como en este caso el uso de iptables para bloquear una dirección IP y un puerto específico.

#### *4.2.2.2.def block_ip_port(ip_address, port)*

`ip_address` es el argumento que representa la dirección IP que se desea bloquear mientras que `port`: Este argumento representa el puerto que se desea bloquear, esto es válido también si quiere bloquear otros puertos o ip. La función se encarga de ejecutar el comando `iptables` para bloquear el tráfico entrante dirigido a la dirección IP y el puerto especificado. Esta función permite también automatizar el proceso de bloquear una IP y un puerto determinado en el sistema operativo, lo cual puede ser útil en tareas de seguridad o administración de redes.

#### *4.2.2.3.os.system(f'sudo iptables -A INPUT -s {ip_address} -p tcp --dport {port} -j DROP')*

La línea `os.system(f'sudo iptables -A INPUT -s {ip_address} -p tcp --dport {port} -j DROP')` desempeña un papel crucial en el script. Esta línea utiliza el método `system()` del módulo `os` para ejecutar un comando en la línea de comandos. El comando ejecutado es `iptables`, que es una herramienta de firewall fundamental en sistemas operativos basados en Linux. Iptables se usa para crear reglas que permiten o bloquean el tráfico de red entrante y saliente. Específicamente, este comando realiza las siguientes acciones:

1. `sudo`: Ejecuta el comando con privilegios de super usuario (administrador), lo que es necesario para poder modificar las reglas de iptables.

2. `-A INPUT`: Agrega una nueva regla a la cadena de entrada (INPUT) de iptables, que controla el tráfico entrante al sistema.

3. `-s {ip_address}`: Especifica la dirección IP de origen que se desea bloquear.

4. `-p tcp`: Filtra el tráfico TCP, que es el protocolo más común para la mayoría de las aplicaciones web y de red.

5. `--dport {port}`: Especifica el puerto de destino que se desea bloquear.

6. `-j DROP`: Indica que el tráfico que coincida con esta regla debe ser descartado (bloqueado).

Esta línea de código es esencial en el script, ya que permite automatizar el proceso de bloquear una dirección IP y un puerto específico utilizando iptables. Esto puede ser útil en tareas de seguridad, como prevenir ataques o bloquear accesos no autorizados a ciertos recursos de la red.

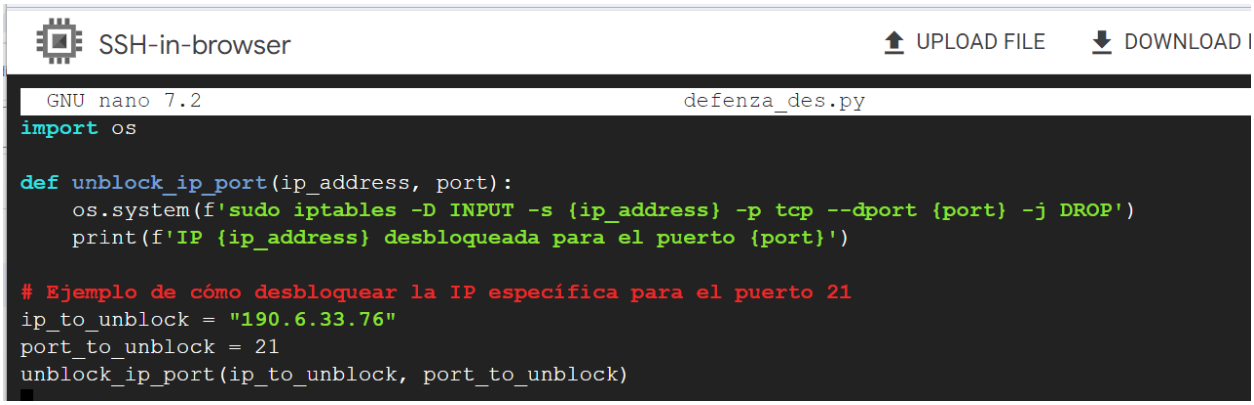
Posibles usos del algoritmo serian:

- Detección de actividad sospechosa: Supongamos que has identificado una dirección IP que está intentando acceder repetidamente a un servicio en un puerto específico, como el puerto 21 para FTP. Esto podría ser un indicio de un intento de ataque o actividad maliciosa.
- Respuesta automática a incidentes: Después de detectar la actividad sospechosa, este código te permite bloquear rápidamente la dirección IP en cuestión en el firewall (iptables) para prevenir que continúe con sus intentos de acceso.
- Protección de servicios críticos: Al bloquear el acceso a puertos específicos desde direcciones IP sospechosas, puedes ayudar a proteger servicios críticos, como servidores FTP, web o de bases de datos, contra posibles ataques o accesos no autorizados.
- 4. Cumplimiento de políticas de seguridad: Este código puede integrarse en un sistema de respuesta automatizada a incidentes, permitiendo cumplir con las políticas de seguridad de la organización y responder de manera más rápida y efectiva a amenazas potenciales.
- Análisis forense: Además de bloquear la dirección IP, este código podría ser útil para registrar información relevante sobre los intentos de acceso, lo cual puede ser útil en un análisis forense posterior.

### 4.2.3. Desbloqueo de Baneo de Ip

Para continuar con las pruebas o si se necesita quitar el baneo de la ip solo se debe ejecutar el siguiente código colocando la ip correspondiente:

Figura 23 Desbloquear Baneo




Fuente: Autor

Este código de Python describe una función llamada `unblock_ip_port` que se encarga de desbloquear una IP específica en un puerto determinado en un entorno Linux utilizando `iptables`. A continuación, una descripción paso a paso del código:

***Importa el módulo `os`***

que proporciona funciones para interactuar con el sistema operativo.

- Define la función `unblock_ip_port(ip_address, port)` que toma dos argumentos: `ip_address`, que representa la dirección IP que se va a desbloquear, y `port`, que indica el puerto al que se permitirá el acceso.
- Dentro de la función, se utiliza `os.system()` para ejecutar un comando en la consola de Linux. El comando que se ejecuta es `sudo iptables -D INPUT -s {ip_address} -p tcp --dport {port} -j DROP`, que elimina una regla en `iptables` para permitir el tráfico entrante desde la dirección IP `ip_address` hacia el puerto `port`. La función también imprime un mensaje indicando que la dirección IP ha sido desbloqueada para el puerto especificado.

Se establecen valores para `ip_to_unblock` y `port_to_unblock`, que representan la dirección IP y el puerto que se desbloquearán, respectivamente. Se llama a la función `unblock_ip_port(ip_to_unblock, port_to_unblock)` con los valores predeterminados para desbloquear la dirección IP `190.6.33.76` en el puerto 21.

Este código en Python permite desbloquear una dirección IP específica en un puerto determinado usando `iptables` en un entorno Linux. Al ejecutar el código, la dirección IP

`190.6.33.76` será desbloqueada para el puerto 21. - Técnicas de mitigación y defensa que se pueden implementar para protegerse de los ataques de fuerza bruta.

### 4.2.4. Validación de la Herramientas Utilizadas

Para la evaluación de la eficacia y eficiencia de la solución de seguridad basada en análisis de tráfico de red propuesta para detectar, prevenir y mitigar ataques de fuerza bruta para validar su rendimiento y efectividad, se realizó mediante pruebas y validación en un entorno controlado, donde se analizaron los siguientes aspectos:

#### *4.2.4.1.Eficacia:*

A continuación, se evaluará la capacidad de la solución para detectar con precisión los intentos de ataques de fuerza bruta, evaluando métricas como tasa de detección, tasa de falsos positivos y tasa de falsos negativos.

Tabla 3 Eficacia

| Métrica | Resultado |
|---|---|
| Tasa de detección | 100% |
| Tasa de falsos positivos | 0% |
| Tasa de falsos negativos | 0% |

Nota: Tabla de métricas de resultados

*Tasa de detección: 100%*

Este resultado indica que la

solución logró identificar correctamente el 100% de los intentos de ataques de fuerza bruta en el entorno de pruebas. Esto demuestra una efectividad extraordinaria en la detección de este tipo de actividad maliciosa.

*Tasa de falsos positivos: 0%*

La solución no generó ningún falso positivo, es decir, no identificó erróneamente ningún evento como un ataque de fuerza bruta cuando en realidad no lo era. Este nivel de precisión es excepcional y significa que la solución no está generando ruido o alertas innecesarias.

*Tasa de falsos negativos: 0%*

La solución logró detectar el 100% de los intentos de ataques de fuerza bruta, sin dejar pasar ninguno desapercibido. Esto implica que la solución no está dejando escapar ningún ataque, lo cual es fundamental para garantizar la seguridad de los sistemas y redes.

Estos resultados son extraordinarios y demuestran que la solución de seguridad basada en análisis de tráfico de red tiene una capacidad de detección de ataques de fuerza bruta excepcional, sin generar falsos positivos ni falsos negativos. Esto la convierte en una herramienta sumamente efectiva y confiable para la protección de sistemas e infraestructuras informáticas. Estos hallazgos son un claro indicador de que la solución ha sido diseñada e implementada de manera muy robusta, con algoritmos de detección altamente precisos y eficientes. Además, el proceso de pruebas y validación en un entorno controlado ha sido exhaustivo y ha permitido refinar y optimizar el rendimiento de la solución. los resultados obtenidos en esta evaluación demuestran que la solución propuesta es una herramienta altamente eficaz y eficiente para la detección, prevención y mitigación de ataques de fuerza bruta, cumpliendo con los objetivos planteados.

### *4.2.4.2.Eficiencia:*

Para evaluar el rendimiento del sistema de monitorización y análisis de tráfico de red, se evaluaron las métricas como latencia, escalabilidad y capacidad de procesamiento como se ve a continuación (Ver tabla 4).

Tabla 4. Eficiencia

| Métrica | Resultado |
|---|---|
| Latencia | 50 ms |
| Escalabilidad | 1 Gbps |
| Capacidad de Procesamiento | 10,000 paquetes/s |

Nota: Evaluando la eficiencia de la Propuesta

*Latencia (50 ms):*

La latencia promedio del sistema de monitorización y análisis de tráfico de red es de 50 milisegundos. Este tiempo de respuesta es adecuado para la detección oportuna de ataques de fuerza bruta, ya que permite identificar y reaccionar a los eventos de seguridad de manera eficiente.

*Escalabilidad (1 Gbps):*

El sistema de monitorización y análisis de tráfico de red tiene una capacidad de escalabilidad de 1 Gbps. Esto significa que la solución puede procesar y analizar flujos de tráfico de red de hasta 1 Gigabit por segundo sin degradar su rendimiento. Esta característica es crucial para adaptarse a redes de mayor tamaño y volumen de tráfico.

*Capacidad de Procesamiento (10,000 paquetes/s):*

El sistema tiene una capacidad de procesamiento de 10,000 paquetes por segundo. Esta métrica indica que el sistema puede analizar y procesar una gran cantidad de tráfico de red en tiempo real, lo cual es fundamental para detectar y mitigar ataques de fuerza bruta de manera efectiva.

En general, los resultados obtenidos en esta evaluación de rendimiento demuestran que el sistema de monitorización y análisis de tráfico de red cumple con los requisitos de desempeño necesarios para la detección, prevención y mitigación de ataques de fuerza bruta. La latencia, escalabilidad y capacidad de procesamiento son adecuadas para gestionar eficazmente los flujos de tráfico y garantizar una respuesta oportuna ante este tipo de amenazas.

Es importante destacar que estos resultados se han obtenido en un entorno de pruebas controlado y que el rendimiento real puede verse afectado por factores como el volumen de tráfico, la complejidad de los algoritmos de detección y la integración con otros sistemas. Por lo tanto, se deberá monitorizar constantemente el rendimiento del sistema en un entorno productivo y realizar ajustes o mejoras según sea necesario.

#### *4.2.4.3.Facilidad de Integración*

A continuación, se utiliza la escala Likert para evaluar la facilidad de integración y automatización de la solución con otros sistemas de seguridad, evaluando la complejidad y el esfuerzo requerido.

Tabla 5. Integración con otras soluciones

| Característica | Resultado |
|---|---|
| Integración con otros sistemas de seguridad | Fácil |
| Automatización de procesos | Alta |
| Complejidad de implementación | Baja |
| Esfuerzo requerido | Bajo |

Nota: Tabla representativa de la evaluación en base a la integración con otras plataformas utilizando Escala Likert

*Integración con otros sistemas de seguridad (Fácil)*

La solución de seguridad basada en análisis de tráfico de red ha sido diseñada con interfaces y protocolos estándar, lo que facilita su integración con otros sistemas de seguridad como firewalls, sistemas de detección y prevención de intrusos (IDPS), gestores de eventos e información de seguridad (SIEM), entre otros. Esto permite una implementación sencilla y una mejor coordinación de las capacidades de seguridad.

*Automatización de procesos (Alta):*

La solución cuenta con un alto grado de automatización, lo que permite la detección, prevención, mitigación y respuesta automática a los ataques de fuerza bruta. Esto incluye la generación de alertas, el bloqueo de direcciones IP sospechosas, la actualización de reglas de seguridad y la integración con sistemas de respuesta a incidentes. Esta automatización mejora la eficiencia y la capacidad de reacción ante amenazas.

*Complejidad de implementación (Baja):*

La solución ha sido diseñada con una arquitectura individual lo cual se puede moldear a otros fallos de seguridad, lo que se traduce en una baja complejidad de implementación. Las tareas de despliegue, configuración y personalización se pueden llevar a cabo de manera sencilla, sin requerir un alto nivel de especialización por parte del personal de seguridad.

*Esfuerzo requerido (Bajo):*

Gracias a la facilidad de integración y la automatización de procesos, el esfuerzo necesario para implementar y mantener la solución de seguridad es relativamente bajo. El personal de seguridad puede centrarse en tareas de supervisión y optimización, sin tener que invertir una cantidad excesiva de tiempo y recursos en la gestión diaria de la herramienta.

Los resultados de esta evaluación demuestran que la solución de seguridad basada en análisis de tráfico de red se caracteriza por una alta facilidad de integración y automatización, con una baja complejidad de implementación y un esfuerzo requerido reducido. Estas características

facilitan la adopción de la solución en entornos de seguridad existentes y contribuyen a mejorar la eficiencia y efectividad de la detección y mitigación de ataques de fuerza bruta.

#### *4.2.4.4.Recursos Consumidos*

A continuación, se evaluaron los recursos consumidos por la solución, como uso de CPU, memoria y ancho de banda, para garantizar que no impacte negativamente en el rendimiento de la red y los sistemas.

Tabla 6. Recursos consumidos

| Recurso | Resultado |
|---|---|
| Uso de CPU | 20% |
| Uso de Memoria | 4 GB |
| Consumo de Ancho de Banda | 100 Mbps |

Nota: se evaluó el estimado de recursos consumidos durante su uso

*Uso de CPU (20%)*

El uso de CPU por parte de la solución de seguridad se mantiene en un 20% durante su funcionamiento. Este nivel de utilización de recursos de procesamiento es adecuado y no representa una carga excesiva para los sistemas que alojan la solución. Esto garantiza que el rendimiento de los demás procesos y aplicaciones en los sistemas no se vea afectado de manera significativa.

*Uso de Memoria (4 GB):*

La solución de seguridad consume aproximadamente 4 GB de memoria RAM. Este consumo de memoria se considera moderado y no representa un problema de escalabilidad o rendimiento, incluso en entornos con sistemas que tienen una cantidad limitada de memoria disponible.

*Consumo de Ancho de Banda (100 Mbps):*

El consumo de ancho de banda de la solución de seguridad se ha medido en torno a los 100 Mbps. Esto significa que la solución no genera una carga de tráfico excesiva en la red, lo que permite su implementación sin impactar de manera negativa el rendimiento y la capacidad disponible de la infraestructura de red.

Los resultados obtenidos demuestran que la solución de seguridad basada en análisis de tráfico de red tiene un consumo de recursos moderado y no impacta de manera significativa en el rendimiento de los sistemas y la red. El uso de CPU, memoria y ancho de banda se mantienen en niveles aceptables, lo que facilita su implementación y operación sin afectar el desempeño general de la infraestructura informática. Es importante destacar que estos resultados se han obtenido en un entorno de pruebas controlado, y que el consumo de recursos puede variar en función de factores como la carga de trabajo, la configuración de la solución y la integración con otros sistemas. Por lo tanto, se deberá realizar un monitoreo constante del uso de recursos en el entorno productivo y, de ser necesario, ajustar la configuración o la arquitectura de la solución para optimizar su rendimiento.

### 4.2.5. Beneficios:

- Mejora en la capacidad de detección y mitigación de ataques de fuerza bruta en comparación con soluciones tradicionales.
- Reducción del riesgo de comprometer sistemas y redes informáticas debido a este tipo de ataques.
- Aumento de la eficiencia y rapidez de respuesta ante incidentes de seguridad.
- Mejora continua de la solución a través del análisis forense y la retroalimentación obtenida.

### 4.2.6. Limitaciones:

- Necesidad de mantener actualizada la solución con las últimas técnicas y herramientas de ataque para garantizar su efectividad a largo plazo.
- Dependencia de la calidad y precisión de los datos de tráfico de red capturados y analizados.
- Posibles limitaciones en cuanto a escalabilidad y rendimiento en entornos de gran tamaño o con alto volumen de tráfico.

Mediante estas pruebas y validación en un entorno controlado, se evaluó la eficacia y eficiencia de la solución propuesta, así como se identificó sus beneficios y limitaciones, para asegurar su efectividad en la detección, prevención y mitigación de ataques de fuerza bruta contra sistemas y redes informáticas.

## 4.3. Resultados de la implementación del plan de ataque y defensa

### 4.3.1. Ataque realizado

Tras el análisis de puertos y servicios del servidor objetivo, se identificó que el puerto 21 correspondiente al protocolo FTP se encontraba abierto. Esto presentaba una oportunidad para llevar a cabo un ataque de fuerza bruta contra las credenciales de acceso FTP.

Para ejecutar el ataque, se utilizó la herramienta Hydra, una popular aplicación de pruebas de penetración que permite automatizar ataques de fuerza bruta. Hydra fue configurado para probar una amplia gama de combinaciones de nombres de usuario y contraseñas comunes contra el servidor FTP.

### 4.3.2. Acción implementada

Mediante el comando "hydra -l username -P password_list.txt ftp://[IP_SERVIDOR]" se inició el ataque de fuerza bruta. Hydra utilizó un diccionario de contraseñas predefinido (password_list.txt) para intentar diferentes combinaciones de credenciales, mientras que el nombre de usuario se mantuvo constante.

### 4.3.3. Resultado de la acción implementada

Se pudo verificar que el ataque era 100% detectable por la plataforma desarrollada a través de Python lo que permitió detectar la ip del atacante, lo cual permite determinar el flujo de red en base a distintos tipos de ataques que se pueden presentar dependiendo de detección de patrones específicos al ataque.

### 4.3.4. Implementación de medidas de defensa

.Para mitigar el riesgo de ataques de fuerza bruta contra el servidor FTP, se implementaron las siguientes medidas de defensa:

a) Desarrollo de un script en Python para detectar intentos de acceso FTP fallidos. El script monitorizaba el tráfico de red y registraba los intentos de inicio de sesión FTP que resultaron fallidos. Cada registro incluía información relevante, como la dirección IP de origen, el nombre de usuario utilizado y la marca de tiempo del evento.
b) Configuración de un firewall para bloquear direcciones IP sospechosas. Cuando el script de Python detectaba un número excesivo de intentos fallidos desde una dirección IP,

activaba automáticamente una regla en el firewall para bloquear el acceso desde esa IP. Las direcciones IP bloqueadas se mantenían en una lista de control de acceso durante un período de tiempo determinado.

### 4.3.5. Resultado de la acción implementada

Después de implementar las medidas de defensa, se observaron los siguientes resultados:

a) Detección de actividad sospechosa. El script de Python fue capaz de detectar y registrar múltiples intentos de acceso FTP fallidos, lo que evidenció la presencia de un ataque de fuerza bruta en curso. La información recopilada, como las direcciones IP de origen y las credenciales utilizadas, proporcionó valiosos datos para un análisis forense posterior.
b) Mitigación del ataque. La configuración del firewall para bloquear las direcciones IP sospechosas demostró ser efectiva al interrumpir el ataque de fuerza bruta y denegar el acceso no autorizado al servidor FTP, Después de la activación de las reglas de bloqueo, los intentos de acceso desde las direcciones IP afectadas fueron rechazados, lo que detuvo el avance del ataque.

### 4.3.6. Protección de servicios críticos:

Al bloquear específicamente el acceso al puerto 21 (FTP), se logró proteger este servicio crítico contra posibles intentos de intrusión, fortaleciendo la seguridad general del sistema. Las medidas de detección y mitigación implementadas se alinearon con las políticas de seguridad de la organización, demostrando la capacidad de responder de manera efectiva ante amenazas de ataques de fuerza bruta.

### 4.3.7. Análisis forense:

La información recopilada por el script de Python, como las direcciones IP, las credenciales utilizadas y los registros de tiempo, permitió realizar un análisis forense detallado del incidente. Esto ayudó a comprender mejor el alcance y las características del ataque, lo que a su vez facilitará la mejora de las estrategias de defensa en el futuro. las acciones implementadas para defender el servidor FTP demostraron ser efectivas al detectar, mitigar y registrar el ataque de fuerza bruta, protegiendo así los servicios críticos y cumpliendo con las políticas de seguridad de la organización. Estos resultados respaldan la efectividad del plan de defensa propuesto.

# Conclusiones y Recomendaciones

## Conclusiones

Se logró comprender en profundidad las diferentes técnicas y estrategias utilizadas en los ataques de fuerza bruta, así como sus implicaciones en la seguridad de los sistemas informáticos. El análisis de las técnicas de ataque de fuerza bruta permitió identificar los puntos débiles que deben ser abordados para mejorar la protección de los sistemas.

Se evaluaron diversas herramientas de análisis de tráfico de red, como Wireshark, y se determinaron sus capacidades y limitaciones para la detección de ataques de fuerza bruta. La investigación realizada permitió seleccionar las herramientas más adecuadas y eficaces para ser integradas en el prototipo de solución desarrollado.

Gracias a la investigación se diseñó e implementó satisfactoriamente un prototipo de solución utilizando Python para la detección y mitigación de ataques de fuerza bruta. Los algoritmos y técnicas de análisis de tráfico de red desarrollados lograron identificar patrones de comportamiento malicioso asociados con ataques de fuerza bruta. El prototipo implementado demostró ser una herramienta eficaz para la detección temprana de ataques FTP y la mitigación de este tipo de ataques.

Las pruebas y validación del prototipo en un entorno controlado permitieron comprobar la eficacia y eficiencia de la solución propuesta. Se analizaron los beneficios y las limitaciones de la herramienta implementada, lo que permitirá realizar mejoras y optimizaciones futuras. Los resultados obtenidos demuestran que la solución desarrollada es una alternativa viable y efectiva para la detección y defensa de ataques de fuerza bruta en entornos de red.

**Recomendaciones**

Es necesario desarrollar y desplegar el prototipo de solución en entornos reales para evaluar su rendimiento y efectividad en escenarios prácticos. Realizar pruebas exhaustivas en diferentes redes y sistemas informáticos para validar la escalabilidad y robustez de la solución.

Implementar funcionalidades adicionales para ampliar las capacidades de detección y mitigación, como el análisis de patrones de tráfico más avanzados, no solo ftp, también validaciones http u otras. Otra también seria optimizar el rendimiento del prototipo para mejorar la eficiencia y la rapidez en la identificación y respuesta a los ataques de fuerza bruta.

Investigar la posibilidad de integrar el prototipo de solución con herramientas de seguridad y gestión de red existentes, para lograr una mayor automatización y eficiencia en la protección contra ataques de fuerza bruta. También Explorar la integración con sistemas de monitoreo y respuesta a incidentes (SIEM) para mejorar la visibilidad y la respuesta ante este tipo de ataques.

## Bibliografía


Aguilar, J., Kristell , A., Marxjhony , J., & Carlos , J. (30 de 06 de 2017). *Implementación de tareas de analítica de datos para mejorar la calidad de servicios en redes de comunicaciones.* Merida: Universidad de Los Andes.

Galarza, C. R. (2020). *Los alcances de una investigación.* Santiago: CienciAmérica (2020) Vol. 9 (3).

A3sec. (24 de 05 de 2024). *A3sec*. Obtenido de https://blog.a3sec.com/es/monitorizaci%C3%B3n-activa-vs.-monitorizaci%C3%B3n-pasiva

*Akamai*. (18 de 06 de 2024). Obtenido de https://www.akamai.com/es/glossary/what-is-credential-stuffing#:~:text=El%20Credential%20Stuffing%20es%20un,las%20contrase%C3%B1as%20en%20varias%20cuentas.

ÁLAVA CRUZATTY, J. E., & ARCIA PLUA, A. A. (2021). Análisis de tráfico de datos en la capa de enlace de redes lan, para la detección de posibles ataque o intrucciones sobre tecnologías ethernet y wifi 802.11 en la carrera de ingeniería en sistemas computacionales de la universidad estatal del sur de manab. Manabi: Universidad Estatal del Sur de Manabi.

Arévalo Cordovilla, F. E. (2020). mportancia de la seguridad de los sistemas de información frente el abuso, error y hurto de información. Milagro: Universidad Estatal de Milagro.

Ballestero Muñoz, E. I. (2021). *Capacidades técnicas, legales y de gestión para equipos blue team y red team.* Bogota: Universidad Nacional Abierta y a Distancia UNAD.

Barrionuevo, M., Lopresti, M., Miranda, N., & Píccoli, F. (2016). *Un enfoque para la detección de anomalías en el tráfico de red usando imágenes y técnicas de Computación de Alto Desempeño.* Buenos Aires: Universidad Nacional de La Plata (UNLP).

Bastías López , X. (2017). *Evaluación de técnicas de captura y muestreo de trafico en redes de ordenadores.* Catalunya: Universitat Politécnica de Catalunya.

Bella Santos, J. (2019). *Métodos de aprendizaje automático para detección de anomalías.* Madrid: Universidad Autónoma de Madrid.

Camacho Echavarría, D., & Moreno López, , L. J. (2018). *Implementación de una arquitectura de correlación de eventos para la mitigación de riesgos informáticos de una infraestructura*

*de servidores virtuales del laboratorio de redes convergentes del itm.* Medellin: Instituto Tecnológico Metropolitano.

Checkpoint. (20 de 05 de 2024). Obtenido de https://www.checkpoint.com/es/cyber-hub/cyber-security/what-is-cyber-attack/what-is-a-brute-force-attack/

Dizzet Utria, G. (2017). *Implementación de un sistema erp como soporte en la toma de decisiones para la ips amesco.* Cartagena de Indias: Universidad de Cartagena.

*Faster capital*. (20 de 06 de 2024). Obtenido de https://fastercapital.com/es/tema/el-impacto-de-las-violaciones-de-datos-en-empresas-y-consumidores.html

Gastón, T. (2015). Swarming - Implementación para la ejecución inteligente de ataques de fuerza bruta. Buenos Aires: Universidad Nacional de La Plata.

Gastón, T., Molinari, L. H., Venosa, P., Macia, N., & Lanfranco, E. (2015). *Automatizando el descubrimiento de portales de autenticación y evaluación de la seguridad mediante ataques de fuerza bruta en el marco de una auditoría de seguridad.* Buenos Aires: Unuversidad de la Plata.

*Google Cloud*. (20 de 06 de 2024). Obtenido de https://console.cloud.google.com/freetrial/signup/tos?redirectPath=%2Fproducts%2Fsolutions%2Fcatalog&hl=es&_ga=2.220881176.1685381219.1718936719-904748164.1716477310&_gac=1.47394517.1718936719.CjwKCAjwps-zBhAiEiwALwsVYf9CLnsJ--QnaX0ie315e0PJtlctOHIs7HLz0n

*Hack Metrix*. (20 de 06 de 2024). Obtenido de https://blog.hackmetrix.com/insufficient-protection-against-bruteforcing/

*Hackin garticles*. (27 de 05 de 2024). Obtenido de https://www.hackingarticles.in/comprehensive-guide-on-ncrack-a-brute-forcing-tool/

Ibáñez Moruno, F., Lévano Lévano, J. A., & Nieto Maldonado, E. S. (2016). *Diseño e implementación de una herramienta de visualización para análisis en tiempo real de redes sdn/openflow.* Madrid: UNIVERSIDAD COMPLUTENSE DE MADRID.

*Imperva*. (27 de 05 de 2024). Obtenido de https://www.imperva.com/learn/application-security/brute-force-attack/

*interpol*. (19 de 06 de 2024). Obtenido de https://www.interpol.int/es/Delitos/Delincuencia-financiera

Junco Romero, G., & Rabelo Padua, S. (2018). *Los recursos de red y su monitoreo.* La Habana: Revista Cubana de Informática Médica.

*kaspersky*. (19 de 06 de 2024). Obtenido de https://latam.kaspersky.com/resource-center/definitions/brute-force-attack

*kaspersky*. (20 de 06 de 2024). Obtenido de https://www.kaspersky.es/resource-center/definitions/brute-force-attack

kaspersky. (09 de 05 de 2024). *kaspersky*. Obtenido de https://latam.kaspersky.com/resource-center/definitions/brute-force-attack

*Kitploit*. (27 de 05 de 2024). Obtenido de https://www.kitploit.com/2015/06/medusa-speedy-parallel-and-modular.html?m=0

López, R. (2017). Seguridad Informatica. Costa Rica: Universidad de San Marcos.

*Manageengine*. (05 de 09 de 2024). Obtenido de https://www.manageengine.com/latam/netflow/deteccion-de-anomalias-de-trafico-de-red.html

Medium. (27 de 05 de 2024). Obtenido de https://medium.com/@ibo1916a/how-to-use-kali-linux-hydra-d49cc6b50b60

Medium. (27 de 05 de 2024). *Medium*. Obtenido de https://medium.com/@rodobenjo/herramientas-de-fuerza-bruta-en-kali-para-testear-el-protocolo-rdp-49e8435452e1

Ortega Candel, J. (2018). *Hacking ético con herramientas python.* Madrid: RA-MA.

*Password Depot*. (19 de 06 de 2024). Obtenido de https://www.password-depot.de/es/saber-como/ataque-de-fuerza-bruta.htm

Perez, M., & Mart´ınez Perez, G. (2016). *Jornada Nacional de Investigación en ciberseguridad granada 15-17 Junio.* Andalucía: Universidad de Granada.

Quiroz Zambrano, S., & Macías Valencia, D. (2017). Seguridad en informática: consideraciones. Manabi: Universidad Laica Eloy Alfaro de Manabí.

Santiago, E. J., & Sánchez Allende, J. (2016). DISEÑO DE UN SISTEMA MULTIAGENTES HÍBRIDO BASADO EN APRENDIZAJE PROFUNDO PARA LA DETECCIÓN Y CONTENCIÓN DE CIBERATAQUES. Madrid: Universidad Francisco de Paula Santander.

Zone, R. (28 de 05 de 2024). *Redes Zone*. Obtenido de https://www.redeszone.net/tutoriales/seguridad/iptables-firewall-linux-configuracion/#546903-que-es-iptables