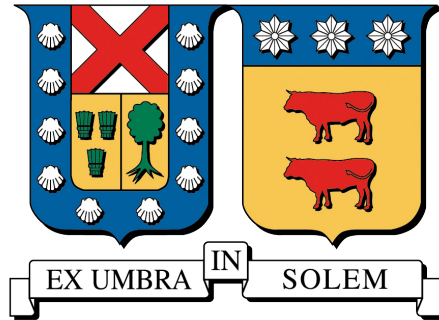


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



IMPLEMENTACIÓN DE UN SISTEMA SIEM CON TECNOLOGÍAS OPEN-SOURCE

LISETTE WILCKENS BUCHHEISTER

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN ELECTRÓNICA**

PROFESOR GUÍA : BERIOSKA CONTRERAS
PROFESOR CORREFERENTE : JOSÉ GABRIEL TORRES

AGOSTO 2024

Agradecimientos

En primer lugar quiero agradecer a mis amigos más cercanos: Andrés, Fabián, Vision, Sofia P. y a mis tíos Astrid y Gary por estar a mi lado durante los últimos 10 años de mi vida. Ellos fueron un pilar fundamental en mi vida y en mis estudios universitarios, brindándome la motivación y el apoyo en cada desafío que debía enfrentar. Sin ellos, no sería la persona que soy actualmente.

También, quiero agradecer a la selección de atletismo de la universidad, ya que desempeñaron un papel crucial en mi formación tanto de persona como deportista. Fue dentro de este deporte donde encontré a mis amigos más cercanos de la universidad y de la vida, con quienes compartí momentos muy especiales.

Asimismo, menciono y agradezco la ayuda de José Gabriel Torres durante el desarrollo de este proyecto.

No puedo dejar de mencionar a los amigos que he conocido durante este período: Eliseo, Dante, Felipe D, Fonchi, Kevin, Valentina y Sofia L.

Finalmente, quiero agradecer a mi familia que me ha apoyado durante este período. Gracias por recibirme siempre y brindarme su ayuda cuando más lo necesitaba.

Resumen

Lo que comprende este trabajo de memoria, se demostró la realización de ataques de denegación de servicio (DDoS) y la detección de este por parte de un SIEM generado a base de tecnologías open-source. Utilizando 10 contenedores orquestados por Docker, instanciando cada uno la prueba de carga JMeter, se logró saturar un servicio expuesto en el puerto 80, aumentando el tiempo de respuesta en 7020 veces por parte del servicio y generando códigos de error HTTP 000 durante el ataque.

Para detectar este ataque, se propuso la implementación de un Sistema de Gestión de Información y Eventos de Seguridad (SIEM). Una herramienta de seguridad defensiva de código abierto, como el basado en la pila ELK, permite centralizar, y analizar los registros de la red, identificar patrones de comportamiento anómalo, correlacionar y detectar posibles ataques.

Al integrar el SIEM con el marco MITRE ATT&CK, se podrá correlacionar los eventos de seguridad con tácticas, técnicas y procedimientos (TTPs) conocidas de los atacantes, aumentando la visibilidad y mejorando la detección de amenazas.

En conclusión, la implementación de un SIEM es fundamental para aumentar la visibilidad de la red, la capacidad de respuesta ante incidentes de seguridad y la mitigación, protegiendo a la organización de futuros ataques.

Abstract

This paper demonstrated the execution of a Denial-of-Service (DDoS) attack using 10 Docker containers, each running a JMeter load test. The attack successfully saturated a service on port 80, increasing response times by 7020 times and generating HTTP error code 000.

To detect such attacks, the implementation of a Security Information and Event Management (SIEM) system was proposed. An open-source SIEM, like one based on the ELK stack, can centralize, manage and analyze network logs, identifying anomalous behavior patterns to detect potential threats.

By integrating the SIEM with the MITRE ATT&CK framework, security events can be correlated to known adversary tactics, techniques, and procedures (TTPs), enhancing threat detection, visibility and allowing correct incident response.

In conclusion, implementing a defensive tool like a SIEM is crucial for improving network visibility, enabling effective incident response, mitigation and bolstering overall organizational security against future attacks.

Keywords: Security, threat detection, network visibility, incident response, vulnerabilities, SIEM, MITRE ATT&CK, log management, anomalous activities, DDoS attacks, correlation, defensive tools, mitigation.

Índice general

1. Glosario	6
2. Introducción	9
2.1. Descripción de capítulos	10
2.2. Motivación	11
2.3. Objetivos	11
3. Marco Teórico	12
3.1. Red and Blue team	12
3.2. Sistemas SIEM	13
3.3. Amenazas	14
3.4. Modelamiento de amenaza específica	15
3.4.1. Actor malicioso (Atacante)	15
3.4.2. Fases de la amenaza	15
3.4.3. Indicadores del ataque	17
3.5. Matriz MITRE ATT&CK	18
3.6. Correlación y precisión	18
4. Estado del Arte	21
4.0.1. Alternativas del mercado	21
4.0.2. Sistemas con licencias pagadas	21
4.0.3. Sistemas de código abierto (open-source)	24
4.0.4. Evaluación Gartner	26
4.1. Análisis de trabajos desarrollados y temas relevantes	28
4.1.1. Protocolo Syslog	28
4.1.2. Protocolo SNMP	28
4.1.3. Ryuk Malware	29
4.1.4. A SIEM Architecture for Advanced Anomaly Detection	30
4.1.5. Evaluation Report	30
4.1.6. Siem Pattern	31
4.1.7. SIEM:Analysis, Trends, and Usage in Critical Infrastructures	31
4.1.8. Extended Detection and Response	31
5. Método de investigación	32
5.1. Arquitectura del sistema	33
5.2. ELK Stack	34
5.3. Manejo de datos	34

6. Modelamiento de red	36
7. Modelamiento de amenaza	45
7.1. Modelo OWASP	45
7.1.1. Beneficios:	45
7.1.2. Proceso:	45
7.2. Modelo de amenazas para la red	46
7.2.1. Vectores de ataque	46
7.2.2. Pasos del ataque	47
7.2.3. Impacto	47
7.2.4. Prevención	47
8. Resultados	50
8.1. Implementación de ataque DDoS	50
8.2. Resultados SIEM	54
8.2.1. SIEM estado inicial	54
8.3. Ejecución y detección del ataque	56
8.4. Análisis de resultados	63
9. Conclusiones y trabajo futuro	67
10. Bibliografía	69

1 | Glosario

1. **SIEM:** Security Information and Event Management.
2. **ELK:** Elasticsearch Logstash and Kibana.
3. **DDoS:** Distributed Denial of Service.
4. **TTPs:** Tactics, Techniques and Procedures.
5. **APT:** Advanced Persistent Threat.
6. **RDP:** Remote Desktop Protocol.
7. **EOL:** End Of Life.
8. **CISO:** Chief Information Security Officer.
9. **STRIDE:** Spoofing, Tampering, Repudiation, Information disclosure, Denial of service and Elevation of privilege.
10. **DREAD:** Damage, Reproducibility, Exploitability, Affected Users, and Discoverability.
11. **MFA:** Multi Factor Authentication.
12. **API:** Application Programming Interface.
13. **IaaS:** Infrastructure as a Service.
14. **SaaS:** Software as a Service.
15. **SNMP:** Simple Network Management Protocol.
16. **TCP:** Transmission Control Protocol.
17. **OS:** Operating System.
18. **DNS:** Domain Name System.
19. **LAN:** Local Area Network.
20. **UDP:** User. Datagram Protocol.
21. **GUI:** Graphical User Interface.
22. **SSL:** Secure Sockets Layer.
23. **SMB:** Server Message Block.
24. **FTP:** File Transfer Protocol.

Índice de figuras

3.1. Matriz MITRE para organizaciones(MITRE, n.d)	18
4.1. Splunk Enterprise Security	22
4.2. IBM QRadar	22
4.3. LogRhythm	22
4.4. McAfee Enterprise Security Manager	23
4.5. ELK stack	24
4.6. OSSIM (Open Source Security Information and Event Management)	24
4.7. Graylog	25
4.8. Security Onion	25
4.9. Wazuh	25
4.10. Cuadrante mágico Gartner	26
5.1. ELK Stack	34
6.1. Arquitectura de la red implementada	36
6.2. Parte 1 filtro pfSense	39
6.3. Parte 2 filtro pfSense	39
6.4. Parte 3 filtro pfSense	40
6.5. Filtro syslog	40
6.6. Llegada de eventos a Kibana	41
6.7. Filtro de eventos en Kibana	42
6.8. Dashborad con información centralizada	43
7.1. Modelamiento de amenazas	46
7.2. Diagrama lógico de la red	48
8.1. Tiempo de respuesta con 1 contenedor instanciado	51
8.2. Tiempo de respuesta con 2 contenedores instanciados	51
8.3. Tiempo de respuesta con 3 contenedores instanciados	52
8.4. Tiempo de respuesta con 4 contenedores instanciados	52
8.5. Tiempo de respuesta con 5 contenedores instanciados	53
8.6. Dashboard Mitre inicial	54
8.7. Dashboard Impact	55
8.8. Containers instanciados	56
8.9. Lista de containers	57
8.10. Consumo de recursos	58
8.11. Apache access logs	59

8.12. Tiempos de respuesta	60
8.13. Estado de respuesta	60
8.14. Dashboard Mitre post ataque	61
8.15. Histograma post ataque	61
8.16. Tabla y tag cloud post ataque	62
8.17. Raw logs post ataque	62
8.18. Direcciones IP de origen post ataque	63

2 | Introducción

Tras leer y analizar distinta documentación y noticias sobre el estado actual de la ciberseguridad, hay que establecer cuál es su relevancia en el mundo actual de la tecnología y qué papel desempeña la herramienta SIEM en esto.

La ciberseguridad es una practica de proteger los recursos de un sistema. La necesidad de esto se aprecia cuando se hacen análisis del aumento de volumen de ataques como plantea Laue, T.(2022) o de los casos como el de Target y Ryu, que, a pesar de ser extremos, son realidades que muestran como evoluciona un ataque, pudiendo ser un incidente pequeño o un APT. Complementariamente a los casos mencionados, se observó durante la presentación de proyectos de las clases lo sencillo que es realizar un ataque (independiente de las intenciones) con pocas herramientas. Por otra parte, se observó también la cantidad de ataques existentes fuera de los proyectos, es decir, al crear ambientes seguros para poder medir la cantidad de ataques recibidos de externos, la cantidad de eventos medidos es alarmante, considerando que los ambientes creados no representaban marcas u organizaciones grandes que se vean como objetivos deseables. Con los datos obtenidos, se llega fácilmente a la conclusión que la seguridad es necesaria, por muy poca que sea, para cualquier tipo de usuario que se mueva con información en las redes.

Dependiendo de las organizaciones y la legislación de la zona, existe distinta información que es crítica que se debería proteger. Al buscar proteger la información y considerando que ésta no se encuentra en servidores pequeños de fácil acceso ¿por qué es relevante enfocarse en servidores y componentes perimetrales de la red, además de los centrales?

Para responder la pregunta hay que considerar que, muchas veces dentro de las organizaciones, la seguridad que se ve desde afuera es mucho más robusta que la que se tiene en la red interna. Esto implica que si se logra penetrar algún servicio vulnerable desde 'afuera', se puede lograr llegar a la información crítica que está guardada al interior de la red. Esto generalmente se logra gracias a los movimientos laterales que realizan los atacantes (explotación de vulnerabilidades, robo de credenciales, RDP, pivotar a través de sistemas comprometidos, entre otros), que permiten llegar a información extremadamente crítica, a través de algo tan simple como un correo de phishing masivo a un proveedor externo, llegando a tener la información personal y bancaria de clientes de la organización más grande, como se ve en el caso Target (Plachkinova & Maurer, 2018).

2.1. Descripción de capítulos

- **Marco teórico:** En el primer capítulo que incluye desarrollo se plantea el marco teórico bajo el que se realiza la memoria. Esto incluye los conceptos Red and Blue Team y sus respectivas funciones, los sistemas SIEM y sus características, la definición de amenazas, el modelamiento de la amenaza específica, la matriz MITRE ATT&CK y la relevancia de correlación y precisión.
- **Estado del arte:** El segundo capítulo analiza el estado del arte respecto al tema seleccionado. Se incluyen las alternativas del mercado, tanto las con licencia pagadas como las opciones open-source, la evaluación Gartner de las distintas opciones disponibles y el análisis de trabajos desarrollados relevantes al tema.
- **Método de investigación:** El tercer capítulo describe el método de investigación a usar, describiendo la arquitectura de la red a usar y la estructura de la herramienta ELK Stack a usar. El desarrollo final se resume en modelar e implementar una red, implementar un sistema SIEM ELK, mapearlo a la matriz MITRE ATT&CK, alimentar el sistema SIEM con logs de distintas fuentes, ejecutar un ataque y detectarlo exitosamente.
- **Modelamiento de red:** El cuarto capítulo modela la red que se tuvo que implementar para poder realizar las pruebas necesarias de comprobación de la eficacia del SIEM diseñado. Se incluyen las componentes a usar, la descripción del tráfico en esta red, los filtros utilizados en el SIEM, cómo se observa el movimiento al interior de la red en la herramienta final a usar y finalmente los criterios aplicados respecto al framework.
- **Modelamiento de amenaza:** El quinto capítulo, al igual que el capítulo anterior, presenta un modelo. En este caso es el de amenaza específico respecto a la red diseñada en el capítulo anterior. En este se describe el modelo a usar, sus beneficios, el proceso, como se modeló la red implementada, los efectos de los resultados iniciales y la implementación/componentes de la amenaza específica.
- **Resultados:** El penúltimo capítulo describe los resultados obtenidos durante el desarrollo del proyecto completo y el análisis de estos. Se tienen los resultados de la red implementada, que incluyen el proceso del ataque DDoS, su efecto en los servicios (tiempos de respuesta y disponibilidad) y el efecto final. Se observó que se pudo efectivamente denegar el servicio. Además, se mostró como la herramienta detectó el ataque y centralizó la información de lo ocurrido. Finalmente, se incluyó la sección de análisis de resultados. Se plantean mejoras a realizar a la herramienta y respecto a las implicancias del método utilizado para desplegar el ataque DDos y su relación con detección de ataques ejecutados con botnets.
- **Conclusiones y trabajos futuros:** El último capítulo describe las conclusiones obtenidas del trabajo realizado y los trabajos futuros a realizar para mejorar el proyecto ya desarrollado. Se plantea que no es necesario incurrir en soluciones costosas, ya que el desarrollo de este trabajo demostró que se puede mejorar la situación con soluciones open-source que puede ser implementado en redes productivas. Finalmente se proponen trabajos futuros que se pueden realizar para el trabajo desarrollado y mejorarlo.

2.2. Motivación

En la actualidad se han observado casos que fácilmente puede resultar en compromiso de datos, tanto para los usuarios como para las organizaciones, que pueden ser frenados o eliminados de manera sencilla si existieran ciertas herramientas para detectar y responder a los eventos iniciales. Por ejemplo, se reciben correos de phishing con apropiación de credenciales. Dentro de lo que es una organización, no se debería tener mayor problema en guardar la información crítica en servidores seguros de pequeños ataques como estos, pero se ha observado una tendencia clara que, muchas veces, se deja vulnerable a estos mismos. Dentro de las alertas más críticas que se han visto, es que una cantidad de servidores están corriendo sobre sistemas operativos y hardware que ya han llegado a su EOL o simplemente no han sido actualizados en períodos largos de tiempo. Esto significa que no se realiza una actualización sobre el software, que es la encargada de parchar distintas vulnerabilidades que aparecen a través del tiempo, y/o la marca ya no ofrece soporte en caso presentar errores físicos en lo que son las máquinas. Esta des-actualización de los servidores implica una acumulación constante de vulnerabilidades, de las que puede no haber consciencia hasta el momento que se tiene un ataque activo.

Considerando estos factores, uno esperaría que se encontrara habilitado un sistema de monitoreo constante y centralizado con fácil acceso a la información crítica, que esté siendo revisado y actualizado bajo la supervisión del CISO, tratando de mejorar constantemente los servicios para velar por el bien de los usuarios.

Es debido lo último que se plantea el proyecto de esta memoria. La idea es probar un despliegue de una herramienta SIEM a escala más baja con tecnologías open-source, probar si es efectivo en la detección de amenazas y posibles ataques específicos, ver qué cosas exponen estos resultados, y dejar el núcleo del sistema disponible al CISO para la implementación a mayor escala dentro de la organización, con reglas de correlación que él considera necesarias para agregar las fuentes de información en un futuro.

2.3. Objetivos

Dentro de los objetivos a realizar durante el trabajo, se tiene:

- Modelar una red adecuada para realizar las pruebas.
- Implementar la red modelada.
- Modelar amenaza DDoS de acuerdo con la red utilizada.
- Implementar un sistema SIEM para centralizar la información.
- Implementar una manera para ejecutar el ataque a la red.
- Ejecutar al ataque.
- Detectar el ataque ejecutado.
- Evaluar la necesidad del sistema SIEM.

3 | Marco Teórico

3.1. Red and Blue team

En el ámbito de la ciber-seguridad, los roles de Red Team y Blue Team son fundamentales para evaluar y fortalecer las defensas de una organización frente a ataques reales.

- Red team: El NIST (National Institute of Standards and Technology) define al Red Team como un grupo de profesionales de seguridad autorizados que emulan a adversarios para simular ataques y vulnerabilidades en los sistemas de una organización. El objetivo principal del Red Team es identificar debilidades en la postura de seguridad y mejorar las capacidades de respuesta a incidentes. (National Institute of Standards and Technology, n.d.)
- Blue team: El NIST define al Blue Team como un grupo de profesionales de seguridad de la información responsables de defender los sistemas de una organización contra ataques simulados o reales. El Blue Team trabaja para detectar, contener y responder a incidentes de seguridad, utilizando herramientas y técnicas de defensa cibernética. (National Institute of Standards and Technology, n.d.)

Funciones del Red Team

- El Red Team debe diseñar escenarios de ataque realistas que reflejen las tácticas, técnicas y procedimientos (TTPs) utilizados por los ciber-atacantes actuales. (Cole et al., 2020)
- El Red Team utiliza técnicas de hacking ético para encontrar y explotar vulnerabilidades en los sistemas y redes de la organización. (Cole et al., 2020)
- El Red Team evalúa la capacidad del Blue Team para detectar, contener y responder a sus ataques simulados. (Cole et al., 2020)
- El Red Team documenta sus hallazgos y recomendaciones para mejorar la postura de seguridad de la organización. (Cole et al., 2020)

Funciones del Blue Team

- El Blue Team implementa y mantiene firewalls, sistemas de detección de intrusiones (IDS), sistemas de prevención de intrusiones (IPS) y otras herramientas de seguridad

para proteger los sistemas y redes de la organización. (Cole et al., 2020)

- El Blue Team analiza registros de seguridad, eventos y alertas para detectar posibles incidentes de seguridad. (Cole et al., 2020)
- El Blue Team toma medidas para contener el impacto de los incidentes de seguridad, erradicar la amenaza y recuperar los sistemas afectados. (Cole et al., 2020)
- El Blue Team trabaja para restaurar los sistemas afectados y volver a la normalidad tras un incidente de seguridad. (Cole et al., 2020)

La colaboración entre Red Team y Blue Team es esencial para mejorar la postura de seguridad de una organización. El Red Team proporciona información valiosa sobre las vulnerabilidades existentes y las tácticas de los atacantes, mientras que el Blue Team utiliza esta información para fortalecer las defensas y mejorar sus capacidades de respuesta a incidentes.

Las pruebas Red Team y Blue Team ofrecen varios beneficios, incluyendo la mejora continua de la postura de seguridad ya que al identificar y corregir vulnerabilidades de forma regular, estas pruebas pueden reducir el riesgo de ataques exitosos. Además aumenta la capacidad de respuesta a incidentes, dado que estas pruebas permiten al Blue Team practicar sus habilidades de detección, contención, erradicación y recuperación en un entorno controlado y mejora de la comunicación y coordinación, considerando que las pruebas Red Team y Blue Team fomentan la comunicación y la coordinación entre los equipos de seguridad, lo que es crucial para una respuesta efectiva a incidentes reales.

Por último, se promueve la cultura de seguridad, ya que estas pruebas pueden aumentar la conciencia sobre la ciber-seguridad en toda la organización, lo que puede ayudar a los empleados a identificar y reportar actividades sospechosas.

3.2. Sistemas SIEM

Un sistema de SIEM (Security Information and Event Management) cumple el rol de detección de amenazas al interior de las organizaciones a partir de eventos especificados de distintas fuentes también especificadas, por lo que se puede clasificar como una herramienta defensiva, ya que se busca una opción transversal a las capas de seguridad para implementarlo en una organización. Lo bueno de esta herramienta, como se dijo anteriormente, es que es transversal a las capas de seguridad, por lo que resulta útil tanto a niveles principiantes de desarrollo de seguridad, como a niveles avanzados. Esto se debe a que otorga visibilidad en el entorno, característica esencial para poder accionar una respuesta cuando la herramienta cumple su función defensiva de dar aviso sobre los sucesos acontecidos.

Por otra parte, la implementación técnica debe cumplir con las siguientes condiciones:

- Cumplimiento (*Compliance*): implica que se depende completamente de los requerimientos dados por el CISO. Como esto es individual a cada organización, no se puede afirmar que, por haber podido implementar un SIEM con las mismas tecnologías en otra organización, sea factible en este caso.

- **Gestión de eventos (*Log management*):** depende de la implementación de la tecnología que se haga, ya que ahí se determina como se normalizan los datos (parsing), almacena y gestiona la información obtenida y por cuánto tiempo se retendrá la información gestionada y procesada (dependiendo de los requerimientos del CISO). La manera de montar e implementar el sistema está estandarizada por guías de las mismas marcas, pero los cambios y detalles fino se ajustan de acuerdo con los requerimientos de la organización.
- **Detección de Amenazas (*Threat detection*):** se debe tener en cuenta qué se considera una amenaza y definir el marco de acuerdo con los parámetros adecuados. En este caso se aplicará el marco conceptual de MITRE ATT&CK (The Mitre Corp., 2023). MITRE ATT&CK es una base de conocimiento accesible de manera global sobre tácticas y técnicas adversarias basadas en observaciones del mundo real. La base de conocimiento ATT&CK es usada como base para el desarrollo de modelos de amenazas específicas y metodologías en el sector privado, gubernamental y en los productos y servicios de la comunidad de la ciber-seguridad. A partir de esta matriz, se podrán clasificar los eventos recibidos y analizados como el tipo de amenazas definidas. La manera de procesar estos datos depende completamente de las reglas de correlación definidas al momento de implementar el sistema, que también deben ser especificadas según los requerimientos de la organización

Para lograr implementar un sistema SIEM de manera exitosa y efectiva, se deben cumplir las siguientes condiciones (Herédia, & Valença, 2019):

- **Recopilación y normalización de datos:** Un SIEM debe ser capaz de recopilar datos de una amplia gama de fuentes, incluyendo dispositivos de red, servidores, bases de datos y aplicaciones. Estos datos deben ser normalizados para que sean comparables y puedan ser analizados de forma eficaz.
- **Correlación de eventos y análisis de comportamiento:** Un SIEM debe ser capaz de correlacionar eventos de seguridad de diferentes fuentes para identificar patrones de comportamiento maliciosos. Esto puede ayudar a identificar ataques que podrían pasar desapercibidos de otra manera. Por ejemplo, un SIEM puede correlacionar intentos fallidos de login de múltiples usuarios para identificar un potencial ataque de fuerza bruta.
- **Monitoreo en tiempo real, notificación y alertas:** Un SIEM debe ser capaz de monitorear la actividad de la red en tiempo real y generar alertas cuando se detectan eventos sospechosos o maliciosos. Esto puede ayudar a los equipos de seguridad a responder a las amenazas de forma rápida y eficaz.

3.3. Amenazas

El Instituto Nacional de Estándares y Tecnología (NIST) define una amenaza como:

- Cualquier circunstancia o evento con el potencial de afectar adversamente las operaciones organizacionales (incluyendo la misión, funciones, imagen o reputación), activos organizacionales o individuos por un sistema de información a través de accesos no autorizados, destrucción, divulgación, modificación de la información y/o denegación de servicio. También el potencial de exitosamente explotar una vulnerabilidad particular de

un sistema de información por parte de una fuente de amenaza. Adaptado de THREAT de CNSSI 4009 y Threat de FIPS 200 en NIST SP 1800-15B (NIST,2018) y (NIST, 2017).

Esta definición de amenaza se basa en los conceptos de THREAT de CNSSI 4009 y Threat de FIPS 200, y se adapta en el NIST SP 1800-15B para proporcionar una comprensión más completa de lo que constituye una amenaza para la seguridad de la información(NIST, 2016).

3.4. Modelamiento de amenaza específica

Dentro del trabajo a desarrollar, se debe modelar la amenaza que se utilizará para atacar al sistema diseñado. A continuación se presenta un ciclo de respuesta a incidente genérico para un ataque del tipo DoS/DDoS, durante el desarrollo del trabajo, se debe realizar un modelamiento del ataque tipo STRIDE, DREAD, etc. para acotarlo a la arquitectura del sistema/servicio.

Este modelo abarca todas las fases del ataque DoS/DDoS, desde la planificación y ejecución por parte del atacante hasta la detección, mitigación y respuesta por parte del defensor, además del proceso de análisis post-ataque y aprendizaje para la mejora continua de la seguridad. La prevención efectiva de ataques DoS/DDoS implica una combinación de medidas técnicas, procedimientos y capacitación del personal de seguridad.

3.4.1. Actor malicioso (Atacante)

El actor malicioso puede ser un individuo, un grupo de hackers o incluso una organización que tiene un motivo o beneficio por realizar este tipo de ataque. Su objetivo es inutilizar o perturbar un sistema o red por períodos de tiempo específicos o en momentos específicos que son críticos para el servicio.

3.4.2. Fases de la amenaza

Reconocimiento

El atacante identifica el objetivo y su infraestructura, incluyendo servidores, servicios y recursos críticos. Además se puede buscar vulnerabilidades conocidas o debilidades en la configuración del objetivo.

Planificación

El atacante selecciona el tipo de ataque DoS/DDoS que utilizará (por ejemplo, inundación SYN, amplificación DNS, ataque HTTP GET)y determina la cantidad de recursos necesarios y la estrategia para coordinar el ataque.

Preparación

El atacante adquiere los recursos necesarios, que pueden incluir botnets, servidores comprometidos, herramientas de ataque y malware, los cuales configura para lanzar el ataque de manera efectiva.

Ejecución del ataque

El atacante lanza el ataque DoS/DDoS enviando una gran cantidad de tráfico malicioso al objetivo. Puede utilizar múltiples fuentes de ataque, incluyendo dispositivos comprometidos y servidores zombies.

Control y mantenimiento (DDoS)

En el caso de un ataque DDoS, el atacante controla y coordina las múltiples fuentes de ataque para mantener la presión sobre el objetivo. Puede cambiar las tácticas o recursos para evadir la detección y mitigación.

Detección y respuesta del defensor

Los defensores monitorean el tráfico en busca de indicadores de un ataque DoS/DDoS. para esto utilizan sistemas de detección de intrusiones, sistemas de mitigación de DDoS y herramientas de análisis de tráfico para identificar el ataque.

Mitigación y respuesta

Los equipos de seguridad responden implementando contra-medidas para mitigar el ataque. Esto puede incluir el filtrado de tráfico malicioso, la adición de ancho de banda adicional o la redistribución de carga que deberían evitar estos ataques a futuro.

Análisis post-ataque

Después del ataque, se lleva a cabo un análisis forense para identificar las fuentes del ataque, los vectores utilizados y cualquier vulnerabilidad explotada. Respecto a estos resultados se toman medidas para fortalecer las defensas y prevenir ataques futuros.

Aprendizaje y mejora continua

Los defensores aprovechan el conocimiento adquirido para mejorar las defensas y respuestas contra futuros ataques DoS/DDoS.

3.4.3. Indicadores del ataque

La identificación de un ataque DoS o DDoS es crucial para tomar medidas de mitigación y proteger un sistema o red. Algunos de los indicadores que se monitorean para identificar estos tipos de ataques son:

Aumento inusual del tráfico de red

Uno de los indicadores más evidentes de un ataque DDoS es un aumento significativo y anormal en el tráfico de red entrante. Esto puede medirse mediante la monitorización de la cantidad de paquetes por segundo (pps) o el ancho de banda.

Picos de tráfico repentinos

Un ataque DDoS a menudo causan alzas de tráfico repentinos y masivos. Estas alzas pueden sobrecargar los recursos de red y servidores, lo que resulta en una disminución del rendimiento o la indisponibilidad de los servicios.

Reducción en el rendimiento del sistema

Un ataque DDoS exitoso puede provocar una disminución significativa en el rendimiento del sistema o la red. Esto puede manifestarse como tiempos de respuesta lentos o errores de conexión.

Variación en los patrones de tráfico

Los ataques DDoS a menudo generan patrones de tráfico inusuales, como un gran número de solicitudes repetitivas desde múltiples direcciones IP, lo que difiere de los patrones de tráfico normales.

Análisis de logs de servidor

Los registros de servidores web y aplicaciones pueden mostrar una gran cantidad de solicitudes de entrada desde múltiples direcciones IP, lo que es indicativo de un ataque DDoS.

Monitoreo de anomalías de tráfico

El monitoreo constante de anomalías en el tráfico, como un aumento repentino en las solicitudes HTTP o DNS, puede ayudar a detectar un ataque DDoS en curso.

Informes de usuarios o clientes

Los usuarios legítimos pueden informar problemas de acceso o rendimiento inusual de servicios, lo que puede ser un indicio de un ataque DDoS.

Alertas de sistemas de seguridad

Los sistemas de seguridad, como firewalls y sistemas de detección de intrusos, pueden generar alertas cuando detectan tráfico malicioso o comportamiento anómalo.

3.5. Matriz MITRE ATT&CK

MITRE ATT&CK es una base de conocimiento con acceso global sobre tácticas y técnicas adversarias basados en observaciones reales.

La base de conocimiento ATT&CK se usa como base para el desarrollo de modelos de amenaza específicas y metodologías en el mundo privado, gubernamental y en la comunidad y los productos de ciber-seguridad.

Con la creación de ATT&CK, MITRE esta cumpliendo la misión de resolver problemas para un mundo más seguro, juntando comunidades para desarrollar ciber-seguridad más efectiva. ATT&CK es de acceso abierto y está disponible para cualquier persona u organización para utilizarla sin cargos.



Figura 3.1: Matriz MITRE para organizaciones(MITRE, n.d)

3.6. Correlación y precisión

En esta sección, se realiza un análisis sobre los conceptos de correlación y precisión en el contexto de los sistemas de detección de intrusiones (IDS) y prevención de intrusiones (IPS),

así como en los sistemas de gestión de información y eventos de seguridad (SIEM).

La precisión es un indicador crucial del rendimiento de las herramientas IDS/IPS, tal como lo señalan Ptacek y Newsham (1998). Se define como la capacidad del sistema para clasificar correctamente los paquetes de red como normales o de ataque. Existen cuatro categorías de clasificación:

- **Verdadero Positivo (TP):** Un paquete de ataque identificado correctamente como tal.
- **Verdadero Negativo (TN):** Un paquete normal identificado correctamente como tal.
- **Falso Positivo (FP):** Un paquete normal incorrectamente clasificado como un paquete de ataque.
- **Falso Negativo (FN):** Un paquete de ataque incorrectamente clasificado como un paquete normal.

La precisión se calcula utilizando la siguiente fórmula:

$$P = \frac{TP}{TP + FP} \quad (3.1)$$

Por otra parte, la precisión total (Average Accuracy Rate) se calcula de la siguiente manera:

$$AAR = \frac{TP + TN}{TP + FN + FP + TN} \quad (3.2)$$

Las dos categorías críticas que afectan la precisión son los falsos positivos y los falsos negativos, como lo destacan Valdes y Mukhopadhyay (2004). Ambos impactan el valor de la precisión, pero la cantidad de falsos negativos es más relevante que la de falsos positivos:

- **Falsos Positivos:** Un exceso de falsos positivos genera una inundación de alarmas, lo que puede ocasionar que se pasen por alto los verdaderos positivos. Esto implica un análisis manual de eventos, desvirtuando el propósito de la herramienta automatizada.
- **Falsos Negativos:** La existencia de falsos negativos permite que los ataques pasen desapercibidos, lo que representa un riesgo significativo para la seguridad de la red.

Para optimizar la precisión y el rendimiento, es necesario buscar un equilibrio entre las verdaderas y falsas alarmas, tal como lo sugieren Ptacek y Newsham (1998). Esto implica ajustar la sensibilidad de la herramienta para minimizar los falsos positivos sin comprometer la detección de verdaderos positivos.

Una estrategia para reducir los falsos positivos es generar una base de datos con reglas de comportamiento malicioso, como lo proponen Gonizzi y Caviglione (2008). Esto permite adaptar las condiciones que generan alertas y evitar la clasificación errónea de paquetes normales como ataques.

Los sistemas SIEM complementan a los IDS/IPS al correlacionar eventos de seguridad de diversas fuentes, proporcionando una visión holística de la actividad de la red. Esta correlación permite identificar patrones y comportamientos anómalos que podrían indicar un ataque en

curso.

La integración de IDS/IPS con SIEM potencia la precisión y la detección de amenazas. La información de los IDS/IPS, que incluye los paquetes clasificados como TP, FP, TN y FN, puede ser utilizada por el SIEM para enriquecer su análisis y mejorar la correlación de eventos.

La precisión total (Average Accuracy Rate - AAR) representa la proporción de eventos clasificados correctamente por el sistema, considerando tanto los verdaderos positivos como los verdaderos negativos. Ho et al. (2003) señalan que la AAR es una medida importante para evaluar el rendimiento general del sistema.

Para mejorar la AAR, se deben implementar estrategias que reduzcan tanto los falsos positivos como los falsos negativos. Esto implica optimizar la configuración de los IDS/IPS, generar bases de datos de firmas y comportamientos maliciosos, e integrar adecuadamente los sistemas IDS/IPS con el SIEM.

La precisión y la AAR son indicadores fundamentales para la efectividad de los sistemas IDS/IPS y SIEM en la protección de las redes contra ataques cibernéticos. La integración de estas herramientas y la implementación de estrategias para mejorar la precisión y la AAR permiten obtener una visión completa de la actividad de la red y responder de manera oportuna a las amenazas.

4 | Estado del Arte

Un SIEM (Security Information and Event Management) es un sistema de seguridad de la información que permite recolectar, analizar y correlación de eventos de seguridad en tiempo real, con el fin de detectar y responder a posibles amenazas y vulnerabilidades. A continuación se presentan algunas características y tendencias actuales del estado del arte de los SIEM:

4.0.1. Alternativas del mercado

Dentro de lo que es el mercado de los sistemas SIEM permiten recopilar, correlacionar y analizar eventos de seguridad en tiempo real desde diversos dispositivos y aplicaciones para detectar y responder a posibles amenazas, siendo sistemas que, bien implementados en organizaciones con definiciones delimitadas, son excelentes herramientas defensivas/mitigadoras (Firewalls, IPS, IDS, antivirus y anti-malware, SIEM, MFA, encriptación de datos, entre otros), ayudado a detectar, prevenir, mitigar y responder a incidentes de seguridad, ya que al ofrecer visibilidad al interior de la red, ya se puede considerar herramienta defensiva. Dentro del mercado, se identifican sistemas SIEM comerciales con licencias pagadas y de código abierto.

4.0.2. Sistemas con licencias pagadas

Splunk Enterprise Security

Splunk Enterprise Security: Splunk es una plataforma líder en gestión de datos y seguridad. Splunk Enterprise Security es su solución SIEM que proporciona análisis avanzados, correlación de eventos, detección de amenazas y cumplimiento normativo. (Kinney Group. (2022, September 2))



Figura 4.1: Splunk Enterprise Security

IBM QRadar

QRadar es una solución SIEM de IBM que ofrece capacidades de seguridad avanzadas, como detección de amenazas en tiempo real, análisis de comportamiento, correlación de eventos y automatización de respuestas. (IBM Security QRadar SIEM - Pricing. (n.d.)).



Figura 4.2: IBM QRadar

LogRhythm

LogRhythm es otra solución SIEM popular que combina análisis de seguridad, gestión de eventos y gestión de la información. Ofrece detección de amenazas en tiempo real, respuesta automatizada y análisis de comportamiento de usuarios.



Figura 4.3: LogRhythm

McAfee Enterprise Security Manager

McAfee ofrece una solución SIEM integral que permite la gestión de eventos de seguridad, detección de amenazas, análisis de riesgos y cumplimiento normativo.



Figura 4.4: McAfee Enterprise Security Manager

Las soluciones SIEM comerciales suelen ofrecer características y funcionalidades más avanzadas, soporte técnico especializado y actualizaciones regulares. Al adquirir una licencia pagada, los usuarios obtienen acceso a estas ventajas adicionales. Por esto mismo, aplicar estas soluciones es mucho más sencillo a nivel organizacional, porque se tiene el apoyo de los expertos de la misma marca dentro del soporte para administrar y realizar las mantenciones o arreglos necesarios.

4.0.3. Sistemas de código abierto (open-source)

Existen varias soluciones SIEM de código abierto (open-source) que ofrecen funcionalidades similares a las soluciones comerciales. Estas herramientas pueden ser una opción asequible y flexible para organizaciones que buscan implementar un sistema SIEM sin incurrir en costos de licencia.

ELK Stack

ELK es una combinación de tres herramientas de código abierto: Elasticsearch, Logstash y Kibana. Elasticsearch se utiliza para almacenar y buscar datos, Logstash se utiliza para recopilar, analizar y enriquecer los datos de registro, y Kibana proporciona una interfaz de usuario para visualizar y analizar los datos. El conjunto ELK puede ser configurado para funcionar como un sistema SIEM eficiente (Elastic Security. (n.d.)).



Figura 4.5: ELK stack

OSSIM (Open Source Security Information and Event Management)

OSSIM es una solución SIEM de código abierto desarrollada por AlienVault. Ofrece capacidades de gestión de eventos, detección de intrusiones, análisis de seguridad, correlación de eventos y cumplimiento normativo (AlienVault. (2023, May 12)).



Figura 4.6: OSSIM (Open Source Security Information and Event Management)

Graylog

Graylog es una plataforma de registro y análisis de registros de código abierto que puede utilizarse para construir un sistema SIEM personalizado. Ofrece capacidades de búsqueda y visualización de registros, así como alertas y notificaciones.



Figura 4.7: Graylog

Security Onion

Security Onion es una distribución de Linux diseñada específicamente para la monitorización de seguridad de redes. Combina varias herramientas de código abierto, como Elasticsearch, Logstash, Kibana y Snort, para proporcionar capacidades de detección de intrusiones, análisis de tráfico de red y gestión de eventos de seguridad.



Figura 4.8: Security Onion

Wazuh

Wazuh es una solución SIEM y de detección de intrusiones de código abierto. Proporciona capacidades de análisis de logs, detección de amenazas, correlación de eventos y gestión de incidentes de seguridad.



Figura 4.9: Wazuh

Estas son solo algunas de las opciones de sistemas SIEM de código abierto disponibles. Es importante evaluar las características y requerimientos específicos de seguridad de la organiza-

ción antes de seleccionar una solución, y considerar aspectos como la comunidad de soporte, la facilidad de implementación y la escalabilidad de la herramienta.

4.0.4. Evaluación Gartner

La empresa Gartner es una consultora estadounidense que entrega constantemente información respecto a las distintas tecnologías disponibles y como se desempeñan en distintos aspectos. A la fecha de octubre 2022, Gartner publicó el siguiente gráfico, conocido como cuadrante mágico.



Figura 4.10: Cuadrante mágico Gartner

El cuadrante se divide en 4 secciones, donde se evalúa la habilidad de ejecutar y qué tan completa es la visión. Con esto se obtienen las categorías de líderes, retadores, jugadores nicho y visionarios (Shoard, Davis,& Schneider, M. (2022, October 10)). Las marcas que componen el cuadrante actual son:

- Microsoft
- IBM
- Splunk
- Securonix
- Exabeam
- Guncul
- Sumo Logic
- Elastic
- Micro Focus

- Logpoint
- Huawei
- ManageEngine
- Devo
- Fortinet
- Rapid7
- LogRhythm

El criterio de inclusión o exclusión representa los atributos específicos que los analistas de Gartner consideran necesarios para que un proveedor sea incluido en el cuadrante. Para calificar como incluido, el proveedor necesita cumplir con los siguientes criterios:

- Un producto que proporciona capacidades SIM y SEM a clientes que sean usuarios finales vía software nativo a cloud y/o como SaaS.
- Características, funcionalidad y soluciones añadidas, al menos dos de las que Gartner enumera como necesarias.
- Security orchestration, automation and response (SOAR).
- Threat intelligence platform (TIP)
- User and entity behavior analytics (UEBA).
- Almacenamiento y reporte de data a largo plazo (más de 365 días).
- Un producto que soporta la captura y análisis de data de una tercera fuente heterogénea vía API en adición a data-streaming o colección de logs (distintos al producto entregado por el proveedor), incluyendo tecnologías líderes en el mercado, endpoints/servidores, cloud (IaaS o SaaS) y aplicaciones de negocios. Esto debe incluir asociaciones bidireccionales y reconocidas con al menos 10 proveedores importantes de tecnologías de seguridad.
- Licencias de plataformas cloud/SaaS y mantención (excluyendo servicios gestionados) con ganancias superiores a \$ 50 millones por 12 meses antes del primero de febrero 2022 o tengan 100 distintos clientes productivos con contratos directos a servicios nativos de cloud o plataformas SaaS al final del mismo período.
- En los 12 meses anteriores al 1 de febrero 2022, deben haber recibido ganancias del 15 % de SIEM nativo a cloud/SaaS de compradores con centrales fuera de la región geográfica de las centrales del proveedor o tener por lo menos 20 clientes productivos, todos con centrales fuera de la zona geográfica de las centrales del proveedor.
- Operaciones de marketing y ventas promoviendo los productos SIEM, cubriendo al menos dos regiones geográficas a la fecha 1 de febrero 2022.

Al leer la lista y los otros factores expuestos en el estudio, se pudo determinar que de todas las soluciones planteadas en las secciones anteriores, la única que es open-source incluida en el cuadrante mágico es Elastic. Esto se debe a que puede generar ganancias con licencias específicas de soporte y otros servicios (por lo que califica para Gartner), pero se puede implementar un SIEM funcional y competitivo entre las mejores marcas del mercado, sin tener que incurrir en costos extra (al ser bien implementado).

4.1. Análisis de trabajos desarrollados y temas relevantes

4.1.1. Protocolo Syslog

El protocolo Syslog es una arquitectura por capas que separa los contenidos del transporte del mensaje, permitiendo extensibilidad para cada capa. Describe el formato estándar para mensajes del tipo syslog y delimita el concepto de asignaciones de transporte. Además, describe elementos de datos estructurados, que se pueden usar para transmitir información estructurada, fácilmente parseable y permite extensiones de proveedores.

En el contexto de SIEM, se puede apreciar que este protocolo va muy entrelazado con el funcionamiento fundamental de la herramienta, dado que aparece en la recopilación de registros, donde el protocolo syslog es utilizado para recopilar registros de eventos y actividades de seguridad de diferentes dispositivos y aplicaciones en una red. Además, permite que los dispositivos generen mensajes de registro y los envíen a un servidor syslog centralizado, donde se almacenan y administran. Esto facilita la recopilación y centralización de los datos de registro en un único repositorio, lo cual es fundamental para el análisis y la correlación de eventos en un sistema SIEM. Dentro de los que es la normalización y el parsing, los mensajes syslog son recibidos por el servidor syslog centralizado o el recolector de SIEM, pasando por un proceso de normalización y parsing. La normalización implica la estandarización de los mensajes syslog para asegurarse de que todos los datos relevantes se encuentren en un formato coherente y uniforme. Luego, el proceso de parsing se encarga de desglosar y extraer información específica de los mensajes syslog, como el tipo de evento, la fuente, la severidad, la hora, la dirección IP origen y destino, y cualquier otro detalle relevante.

La información extraída se utiliza posteriormente para el análisis y correlación de eventos en el sistema SIEM. Con esta información ya disponible y de fácil acceso, los sistemas SIEM analizan los registros syslog en busca de patrones, comportamientos anómalos y eventos de seguridad que puedan indicar una amenaza. Al correlacionar y analizar los eventos syslog con otros eventos de seguridad, como alertas de intrusiones o actividades sospechosas, un SIEM puede detectar incidentes de seguridad y proporcionar una visibilidad integral de la postura de seguridad de una organización.

Finalmente, dentro de lo que es el cumplimiento normativo, ya que los registros syslog son esenciales para el cumplimiento de las regulaciones y normativas de seguridad. Los sistemas SIEM utilizan los registros syslog para generar informes de cumplimiento y demostrar que se han tomado las medidas adecuadas para garantizar la seguridad de los datos y el cumplimiento normativo.

4.1.2. Protocolo SNMP

El propósito de SNMP es proveer un protocolo simple y extensible para gestionar dispositivos en una red. SNMP funciona definiendo un conjunto de funciones gestoras que pueden ser utilizadas para monitorear y controlar elementos de la red (hosts, gateways y servidores terminales). Las estaciones de administración de red ejecutan aplicaciones de gestión que se comunican

con agentes de gestión en los elementos de la red usando SNMP. El protocolo es usado para intercambiar información de gestión entre las estaciones de gestión y agentes, permitiendo a los administradores de red de monitorear y controlar equipos de red desde una localización central.

La relación del protocolo SNMP con los sistemas SIEM se encuentra en el monitoreo de dispositivos de red, donde SNMP es ampliamente utilizado para monitorear y administrar dispositivos de red, como routers, switches, firewalls y otros equipos de infraestructura. Los sistemas SIEM pueden aprovechar el protocolo SNMP para recopilar información sobre el estado y el rendimiento de estos dispositivos. Esto incluye datos como la utilización de ancho de banda, el estado de las interfaces, las estadísticas de tráfico, entre otros. Al integrar el monitoreo SNMP con un sistema SIEM, se obtiene una visibilidad completa de la infraestructura de red, lo que ayuda a detectar eventos de seguridad y anomalías.

Para la recopilación de registros de eventos SNMP, los dispositivos de red generan eventos y notificaciones a través del protocolo SNMP. Estos eventos pueden incluir cambios de configuración, fallas de dispositivos, eventos de autenticación y otros eventos relevantes desde el punto de vista de la seguridad. Un sistema SIEM puede recopilar y analizar estos eventos SNMP para identificar posibles amenazas o comportamientos anómalos.

Por otra parte permite integración con herramientas de administración de red, ya que las soluciones SIEM se integran con herramientas de administración de red que utilizan SNMP para monitorear y administrar dispositivos. Al aprovechar la información recopilada por estas herramientas, un SIEM puede tener y mostrar una visión más completa y precisa del estado de la red y los eventos de seguridad.

Al recopilar información SNMP como estadísticas de tráfico y eventos de dispositivos de red, un sistema SIEM puede identificar patrones de tráfico sospechosos, escaneos de puertos, comportamientos anómalos y posibles ataques, lo que permite una detección temprana de amenazas y una respuesta más rápida a los incidentes de seguridad.

Finalmente, para la correlación de eventos, los eventos SNMP pueden correlacionarse con otros eventos de seguridad recopilados por el SIEM, como registros de firewall, eventos de autenticación, eventos de IDS/IPS, entre otros. Esto proporciona una visión de los eventos de seguridad en la red y ayuda a identificar posibles correlaciones y patrones que indican actividades maliciosas.

4.1.3. Ryuk Malware

Ryuk es un malware de tipo ransomware que se hizo conocido el año 2018. Se cree que es operado por un grupo de ciber-delincuentes llamado Wizard Spider, quienes están asociados con el grupo de ransomware TrickBot.

Este malware se dirige principalmente a organizaciones y empresas grandes que pueden pagar grandes sumas de dinero para recuperar sus datos una vez vulnerados. A diferencia de otros

ransomware que se propagan indiscriminadamente, Ryuk se despliega de manera dirigida y cuidadosamente planificada, utilizando técnicas de spear phishing, exploits de vulnerabilidades y acceso remoto a sistemas comprometidos.

Una vez que Ryuk infecta un sistema, cifra los archivos y muestra una nota de rescate que exige un pago en cripto-monedas a cambio de la clave de descifrado. Los rescates exigidos por Ryuk suelen ser muy altos, lo que indica que los atacantes están dirigidos a objetivos rentables.

Se cree que Ryuk está relacionado con otro malware llamado Emotet, que se utiliza para infiltrarse en las redes de las organizaciones objetivo. Una vez que Emotet establece una presencia dentro de la red, se utiliza como punto de entrada para el despliegue de Ryuk.

Ryuk ha sido responsable de numerosos ataques de alto perfil en todo el mundo, afectando a sectores como el gobierno, la salud y las instituciones financieras. Sus objetivos principales son los sistemas Windows y, una vez que infecta una red, se propaga rápidamente a través de la red, cifrando archivos y causando interrupciones significativas en las operaciones del objetivo.

La evolución de Ryuk ha sido continua, con los operadores del malware actualizando y mejorando sus capacidades a lo largo del tiempo. Continúa siendo una amenaza significativa en el panorama de la ciber-seguridad y se requieren medidas de seguridad sólidas, como copias de seguridad regulares, parcheo de vulnerabilidades y conciencia de seguridad, para protegerse contra este tipo de amenaza.

4.1.4. A SIEM Architecture for Advanced Anomaly Detection

En este paper, se describe y analiza la arquitectura SIEM requerida y se propone el proyecto GLACIER, un sistema SIEM desarrollado e implementado por los mismos autores del paper, con ciertas modificaciones respecto a los sistemas tradicionales del mercado y los casos en los que el proyecto puede ser aplicado de manera pertinentes de acuerdo con el contexto y los resultados/funciones buscadas. Se plantea este proyecto con el que se han realizado pruebas para comprobar su funcionamiento, pero aún no está afinado como se desea. Sin embargo, se siguen realizando estudios respecto al proyecto finalizado para poder lograr afinar detalles del sistema realizado (Lau et al.2022).

4.1.5. Evaluation Report

Se realiza una comparación de los servicios disponibles en el mercado para desempeñar las funciones de sistemas SIEM, ya que no todas las tecnologías están diseñadas para ser utilizadas de esta manera, pero se pueden manipular de manera que funcionen. El problema comparativamente con este trabajo desarrollado es que la mayoría de los sistemas probados implican licencias pagadas. Uno de los utilizados del tipo open-source es Elastic security, pero sin embargo no resulta adecuado para el estudio que se realizó. Sin embargo, el estudio no llega a una sola solución que cumpliera todos los requisitos, si no que tres de ellas (Splunk, IBM QRadar y Palo Alto Cortex XDR) (Leung et al.2021), todas con licencias pagadas, cosa que tiene que decidir la empresa compradora por depender del presupuesto de la misma.

4.1.6. Siem Pattern

En este paper se analizan los posibles patrones que se requiere que tenga un sistema SIEM para poder operar de manera correcta y la arquitectura que debe presentar. Se evalúa que el sistema más avanzado disponible en el mercado es el entregado por IBM QRadar, pero este es nuevamente una licencia pagada por cualquiera de sus opciones disponibles SaaS y software, siendo la opción software mucho más barata de que la SaaS, pero aún así, pagada. Por el lado open-source, se menciona la existencia de AlienVault OSSIM, y una parte de este que también está sujeta al pago de licencia, AlienVault USM (Vielberth et al., 2018).

4.1.7. SIEM:Analysis, Trends, and Usage in Critical Infrastructures

En este paper, se presenta la solución que ofrece una arquitectura de un sistema SIEM, las características de lo que deben tener incorporados, las funciones que cumplen sus características, la evolución de los sistemas actuales disponibles en el mercado, las limitaciones de estos sistemas, el futuro y como se calcula que evolucionarán y posibles mejoras a aplicar. En general se determinan las soluciones más avanzadas que ofrece el mercado: ArcSight, QRadar, McAfee, LogRhythm, USM-OSSIM, RSA, Splunk y SolarWinds (González-Granadillo et al.,2021). No solo se considera la facilidad del uso de estos sistemas para los usuarios, si no que la evolución de la facilidad de acceder a la información disponible para tener mejor entendimiento de las situaciones que se están desarrollando al interior de los sistemas. Además, se incluye el factor de que tan completos son los sistemas (si requieren extensiones extras para poder desempeñar las tareas de manera correcta) y los costos asociados a las licencias y la capacidad de las maquinas para almacenar la información.

4.1.8. Extended Detection and Response

En este paper se describe el funcionamiento de un sistema EDR, un sistema SIEM y finalmente los sistemas XDR y como estos se comparan con los sistemas SIEM (Brandao & Nunes, 2021).

5 | Método de investigación

Para el desarrollo del trabajo, se propone la siguiente metodología. En un ambiente acotado y controlado se realiza el despliegue de una herramienta del tipo SIEM open-source.

Al sistema desplegado se le deben configurar las características para que efectivamente presente una función y el comportamiento de una herramienta del tipo SIEM, utilizando una fuente de información, en este caso es la matriz MITRE ATT&CK, para poder detectar comportamientos maliciosos que podrían resultar en incidentes de seguridad de mayor escala.

Teniendo implementada la herramienta, se procede a alimentarla con información de distintas fuentes. En el caso específico se utilizarán logs de firewall y de sistema específicos de un servicio asignado. El sistema SIEM, al recibir información de distintas fuentes respecto al mismo servicio, tiene capacidad de realizar correlación entre los eventos y determinar qué comportamientos son maliciosos al compararlo con la fuente de información. Con la característica anterior satisfecha, se busca someter el servicio específico a una amenaza determinada, en este caso DOS/DDOS. Estos ataques, a pesar de ser antiguos, son muy simples en su desarrollo y presentan una alta frecuencia en los ambientes actuales (Verizon, 2023).

Se busca verificar que el sistema SIEM pueda identificar el ataque y generar la alerta correspondiente, es decir, clasificarlo como crítica.

Lo que se busca con el trabajo de memoria, se basa en determinar la capacidad de detectar amenazas por parte de la herramienta implementada, por ejemplo DDoS y/o DoS (dada la frecuencia con los que suceden y no es necesario entrar a la red para el ataque, (Verizon, 2023). Se plantea que la implementación de un sistema SIEM es indispensable para el aumento de visibilidad y la capacidad de respuesta a amenazas (DDoS/DoS)(no se puede responder a algo que no se sabe que está ocurriendo).

Se busca evaluar si es posible implementar un sistema SIEM del tipo open-source, configurarlo de acuerdo a los requerimientos dados, suministrarle información relevante respecto al sistema monitoreado (logs de firewall y de sistema), que debería ser suficiente para aumentar la visibilidad de los movimientos para saber qué está sucediendo al interior del sistema y si esto es se puede catalogar como un SIEM.

De esta manera, al juntar la información en un sistema central, se puede establecer un com-

portamiento normal dentro de lo que es el sistema, y con esto mismo, se pueden detectar movimientos que se pueden definir como anómalos.

Estos movimientos anómalos pueden efectivamente tratarse de una amenaza al sistema y a la organización.

Se quiere verificar que el sistema SIEM (de código abierto con el motor de reglas implementados por el alumno), teniendo información basal y alimentación de distintas fuentes respecto al mismo sistema, puede realizar una correlación correcta y ayudar a detectar amenazas, presentando la información de tal manera que se puedan tomar acciones, resultando en una herramienta defensiva/mitigadora.

5.1. Arquitectura del sistema

Para poder desplegar de manera correcta el proyecto planteado, se debe tener una configuración de red para la implementación de SIEM y detección de amenazas específicas.

Para demostrar eficazmente las capacidades de una solución SIEM, se requiere una configuración de red que incluya por lo menos una máquina cliente vulnerable bajo el control de un firewall (pfSense), ambas enviando registros a un ELK Stack que no es cliente del firewall, y la máquina cliente teniendo acceso a Internet y siendo susceptible a ataques. Esta configuración permite la generación de registros de seguridad tanto desde el firewall como desde la máquina cliente, proporcionando al SIEM una visión completa de la actividad de la red y las amenazas potenciales, pudiendo generar correlación a partir de esta información de distintas fuentes.

La estructura de la red se puede describir de la siguiente manera:

1. **Máquinas cliente:** Una o varias máquinas cliente, como una computadora portátil o una estación de trabajo, están conectadas a la red. Estas máquinas representan los objetivos de posibles ataques cibernéticos.
2. **Firewall:** Un firewall actúa como puerta de enlace de red, controlando el tráfico entre la red interna e Internet. El firewall genera registros del tráfico de red, incluidos eventos de firewall e intentos de detección de intrusiones.
3. **ELK Stack:** Un ELK Stack, que consta de Elasticsearch, Logstash y Kibana, se implementa en un servidor separado. Este servidor no es un cliente del firewall, lo que garantiza que no genere eventos de firewall respecto a su actividad. El ELK Stack recibe registros tanto de las máquinas cliente (syslogs) como del firewall para el almacenamiento, análisis y visualización centralizada.
4. **Internet:** Las máquinas cliente tienen acceso a Internet, lo que les permite interactuar con recursos externos y potencialmente encontrarse con actores maliciosos. Esta exposición aumenta la probabilidad de ataques, proporcionando datos de registro valiosos para que el SIEM los analice.

Esta configuración de red proporciona un escenario realista para probar la efectividad de una

solución SIEM en la detección a ciber-ataques. El firewall genera registros del tráfico de red, mientras que las máquinas cliente vulnerables proporcionan un objetivo para ataques y generan registros de actividad sospechosa. El ELK Stack centraliza y analiza estos registros, lo que permite a los analistas de seguridad identificar amenazas potenciales y tomar las medidas adecuadas.

5.2. ELK Stack

El despliegue de la herramienta a usar corresponde al ELK stack. Se compone de tres aplicaciones esenciales para el funcionamiento de cualquier opción de la herramienta.

En primer lugar se tiene el Logstash, aplicación encargada de recibir los logs y procesarlos (parsear, filtrar, realizar correlación y clasificación de los eventos) para enviarlos en un solo formato a almacenamiento.

Luego se tiene la aplicación Elasticsearch, cuya función es recibir la información procesada desde el Logstash y almacenarla por los tiempos establecidos. Esta aplicación facilita las queries y la búsqueda de datos necesaria.

Finalmente, se tiene acceso a la información de forma condensada y simple en la interfaz gráfica entregada por la aplicación Kibana, la que se instala y corre desde la web. En esta aplicación se escoge la información que se quiere visualizar y la distribución de ésta.

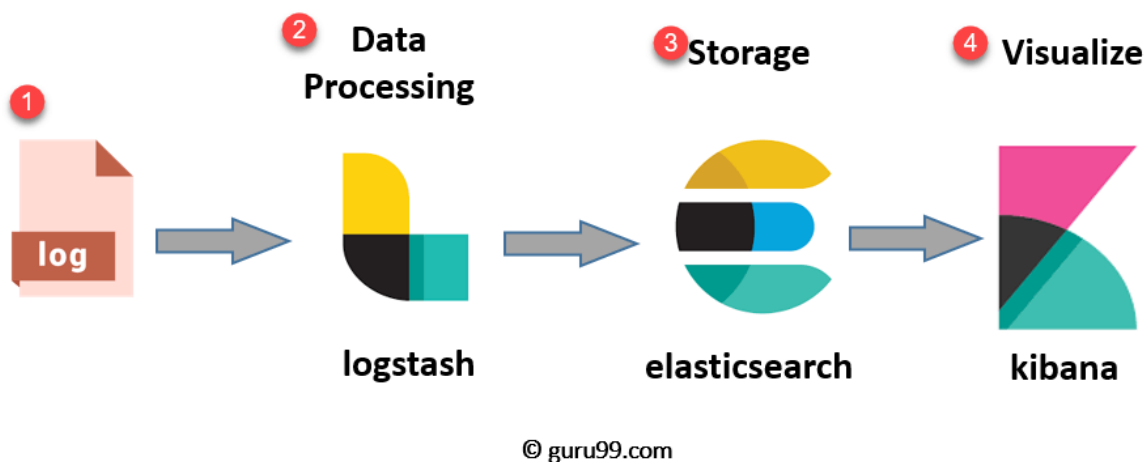


Figura 5.1: ELK Stack

5.3. Manejo de datos

Al momento de recibir datos, la herramienta Logstash procesa inmediatamente los logs y los parsea de manera adecuada dependiendo de su origen.

El formato como los debe parsear y analizar, va a depender completamente de la fuente que envía los logs. En el caso aplicado, los logs de firewall proceden de una herramienta pfSense, por lo que se debe analizar el mensaje que es enviado y se busca cómo separar esto y asignarlo a campos de variables estandarizadas que utiliza el sistema ELK. Los logs provenientes de los sistemas tienen un formato estándar simple, fácil de parsear y con poca información altamente relevante. Correlacionar estos datos solo va a implicar agregar filtros que busquen información específica en campos completados durante el parseo.

6 | Modelamiento de red

Dado que el firewall inicial a usar no registra eventos TCP, se tuvo que diseñar e implementar toda una red nueva y desplegar las componentes necesarias para lograr los objetivos deseados de detección.

A continuación se inserta la figura 6.1 que describe la arquitectura de la red implementada diseñada.

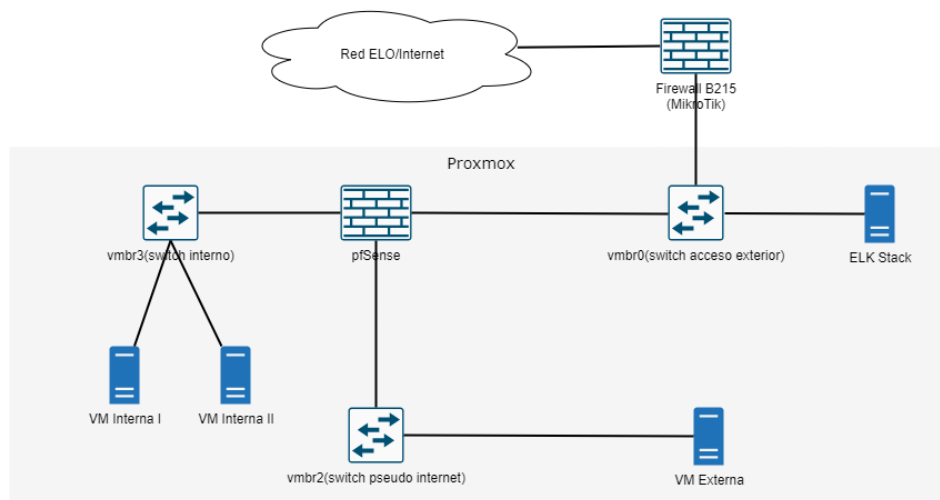


Figura 6.1: Arquitectura de la red implementada

La red implementada ilustrada en la imagen está compuesta por diversos componentes interconectados que trabajan en conjunto para simular el acceso a internet a las máquinas virtuales internas de forma segura y monitoreada.

Las componentes principales con las que se tiene que trabajar son:

- **MikroTik:** El firewall MikroTik es el firewall efectivo de la red interna, encargado de dirigir el tráfico entre las distintas subredes y hacia internet desde las redes internas. Este firewall es el de la capa más externa y en este caso, habría imposibilitado la ejecución de ataques a realizar para verificar el funcionamiento del sistema ELK como SIEM.
- **Firewall pfSense:** El firewall pfSense se tiene que montar para poder cumplir con el requisito del envío de logs de firewall para poder realizar correlación. Además de lo anterior, se encarga de dirigir el tráfico de la red para que las VM internas registren

los eventos de la VM externa como si fuera internet. Así mismo, funciona bloqueando intrusiones y permitiendo conexiones legítimas. La presencia del firewall permite la implementación de reglas personalizadas y la detección de intrusiones.

- **Máquina virtual externa:** Esta máquina virtual actúa como internet simulado hacia las máquinas virtuales internas. De esta manera, se pueden realizar las pruebas y ataques de manera segura, sin comprometer la red oficial. Además, tener una máquina dentro de la red interna que es vista por los servicios como internet, permite modificar la dirección IP de manera manual para usar direcciones IP públicas. Esta es la máquina que genera eventos hacia los clientes de la red.
- **Máquinas virtuales internas:** Estas máquinas virtuales representan los dispositivos que requieren acceso a internet dentro de la red privada. Estas máquinas, que actúan como clientes del pfSense, tienen como OS un Centos 7 y son las que envían los eventos de sistema al ELK para poder correlacionar eventos y detectar ataques.
- **Servidor ELK:** El servidor ELK centraliza la recopilación y análisis de registros de red y seguridad, permitiendo la visualización de registros, la detección de anomalías y la toma de acciones preventivas.

El flujo de tráfico es regulado al interior de esta red de la siguiente manera:

- Las máquinas virtuales internas envían sus solicitudes al firewall pfSense.
- El firewall pfSense analiza las solicitudes y permite solo el tráfico legítimo y autorizado. Si las direcciones de destino de las máquina son internet, dirige el tráfico hacia la máquina virtual que simula ser internet.
- El tráfico permitido se redirige a la máquina virtual externa o al router MikroTik, según su destino.
- El tráfico destinado a internet se envía a la máquina virtual externa.
- La máquina virtual externa responde como si fuera internet.
- El pfSense tiene acceso a internet para poder realizar resoluciones de DNS, entre otros.
- Las solicitudes del PfSense y del ELK son enviadas a internet a través del MikroTik.
- Las respuestas recibidas se envían al MikroTik.
- El MikroTik dirige las respuestas a las máquinas correspondientes que tienen acceso a internet.
- Las máquinas virtuales internas envían syslogs al ELK a través del pfSense.
- El pfSense envía todos los logs pertinentes al ELK.
- El ELK correlaciona e informa respecto a la actividad observada.

Las ventajas de haber desarrollado una arquitectura que emula el funcionamiento de una red con acceso a internet son:

- **Acceso controlado a internet:** Las máquinas virtuales internas no pueden acceder a internet, lo que permite un control granular del tráfico, el análisis exhaustivo de la información respecto a la actividad y la prevención de accesos no autorizados o ataques reales.

- **Protección contra amenazas externas:** El firewall pfSense dirige el tráfico entrante y saliente de la red, mientras que el Mikrotik actúa bloqueando intentos de intrusión y protegiendo la red interna de ataques cibernéticos reales, pero dejando espacio para realizar las pruebas sin comprometer la red real.
- **Monitoreo centralizado:** El servidor ELK centraliza la recopilación de registros de la red y el firewall, permitiendo a los administradores detectar comportamientos anormales y tomar medidas preventivas.
- **Aislamiento de la red interna:** Dado el contexto del sistema, se puede realizar una configuración muy específica y de fácil verificación, ya que no hay información irrelevante circulando, cosa que podría alterar las pruebas al momento de hacerlas.

Al momento de plantear e implementar la arquitectura de la red deseada, se tiene que considerar distintos factores que influyen en las redes internas. En primer lugar, la complejidad de la solución, dado que la configuración y gestión de esta arquitectura de red puede ser compleja y requiere conocimientos técnicos específicos. Además se tiene que verificar que las componentes están funcionando solo de la manera deseada. Se tiene que considerar toda la configuración básica de red, como los son las direcciones IPs necesarias, el tráfico permitido en las distintas LANs y el despliegue de un firewall que cumpla las necesidades básicas de una red y entregue la información pertinente, sin estorbar ni modificarla.

Por otra parte, los recursos necesarios de hardware y software para soportar la infraestructura y el tráfico de la red no son despreciables. Desde el punto de vista de hardware, el laboratorio original a utilizar cuenta con suficientes recursos para desplegar la red, y desde el punto de vista del software, se hace uso de los recursos open-source que existen en el mercado.

La dificultad más alta al momento de implementar esto es la implementación efectiva y las interconexiones de esta arquitectura, ya que se requieren conocimientos técnicos altos de y un conocimiento base de la red ya existente, para poder así crear las redes e interconectar las componentes a los distintos niveles de manera correcta y poder recibir la información para poder procesarla. Una vez establecida e implementada la arquitectura de la red propuesta, se procedió a enviar los logs desde las distintas fuentes y se verificó que la información enviada estuviera correcta y fuera pertinente a lo que estaba sucediendo al interior de la red.

Para poder procesar esta información de manera correcta, se tuvieron que implementar distintos filtros al momento de montar los pipelines en el Logstash.

Las fuentes envían distinta información en distintos formatos, por lo que hay que parsearla de manera correcta para extraer los datos adecuados y relevantes. En ambos casos se enviaron a través del protocolo syslog, pero se asignaron distintos puertos de destino en el Logstash por temas de orden (verificación de la información enviada) y simplicidad (levantar los pipelines por separado para realizar diagnósticos de red y/o comunicación). En el caso de los logs de sistemas enviados desde los clientes, son formato syslog. Los eventos de firewall pfSense son enviados en formato BSD. En las figuras 6.2, 6.3, 6.4 y 6.5 se muestran los filtros implementados:


```

    }
    if "matched" in [tags] {
      mutate {
        remove_field => [ "message" ]
      }
    }
    mutate {
      lowercase => [ "proto" ]
    }
    if "matched" in [tags] {
      geoip {
        add_tag => [ "GeoIP" ]
        source => "src_ip"
        target => "geoip"
      }

      translate {
        source => "dest_ip"
        target => "C&C"
        dictionary_path => "/etc/logstash/blacklist_ips.yaml"
      }
    }
  }
}

output {
  if [type] == "syslog" {
    stdout { codec => rubydebug }
    elasticsearch {
      hosts => [ "localhost:9200" ]
      user => "elastic"
      password => "NERsp6HPaAqG2bYIyJlv"
      ssl => true
      cacert => "/usr/share/logstash/config/ca.crt"
      cacert => "/etc/pki/ca-trust/source/anchors/http_ca.crt"
      ssl_certificate_verification => false
      index => "logstash-pfsense-%{+YYYY.MM.dd}"
    }
  }
}

```

Figura 6.4: Parte 3 filtro pfSense

```

input {
  udp {
    host => "172.16.0.91"
    port => 5145
    type => syslog
  }
}

filter {
  if [type] == "syslog" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:syslog_hostname} %{DATA:syslog_program}(?:\\(%{POSINT:syslog_pid}\\))?: %{GREEDYDATA:syslog_message}" }
      add_field => [ "received_at", "%{timestamp}" ]
      add_field => [ "received_from", "%{host}" ]
    }
    mutate {
      remove_field => [ "syslog_hostname", "syslog_message" ]
    }
    date {
      match => [ "syslog_timestamp", "MMM d HH:mm:ss", "MMM dd HH:mm:ss" ]
      timezone => "America/Santiago"
      tag_on_failure => [ "_dateparsefailure_2" ]
    }
  }
  if [syslog_program] == "login" {
    mutate {
      split => { "message" => " " }
    }
    grok {
      match => { "[message]" => "%{WORD:login_result}" }
    }
  }
}

output {
  if [type] == "syslog" {
    stdout { codec => rubydebug }
    elasticsearch {
      hosts => [ "localhost:9200" ]
      user => "elastic"
      password => "NERsp6HPaAqG2bYIyJlv"
      ssl => true
      cacert => "/usr/share/logstash/config/ca.crt"
      cacert => "/etc/pki/ca-trust/source/anchors/http_ca.crt"
      ssl_certificate_verification => false
      index => "logstash-syslog-%{+YYYY.MM.dd}"
    }
  }
}

```

Figura 6.5: Filtro syslog

Dentro de los datos técnicos es relevante aclarar:

- La dirección IP del Logstash es: "172.16.0.91".
- El protocolo de input en ambos filtros es UDP.
- En el caso del filtro pfSense el puerto a escuchar es port:5140 y en el caso de los clientes

es port:5145.

- El output es enviado al Elasticsearch a "localhost:9200".
- Para acceder a la GUI del Kibana se accede a "localhost:5601"
- Se indica el índice en el que se ubican los eventos enviados, en ambos casos, Logstash*.

Para tener el funcionamiento correcto del Stack, los 3 servicios tienen que estar levantados y funcionando correctamente simultáneamente. El Kibana no funciona si el Elasticsearch no está en funcionamiento.

Para que la transferencia de datos desde Logstash a Elasticsearch sea correcta y lo más segura posible, se debe activar el protocolo SSL y localizar el certificado en la carpeta indicada.

Una vez completado el paso anterior, ya se puede acceder a la GUI del Kibana con las credenciales indicadas y se puede verificar la llegada de eventos en Home → Analytics → Discover. Se debe verificar que el índice que se está revisando corresponde a Logstash. En la figura 6.6 se muestra la vista de lo mencionado:

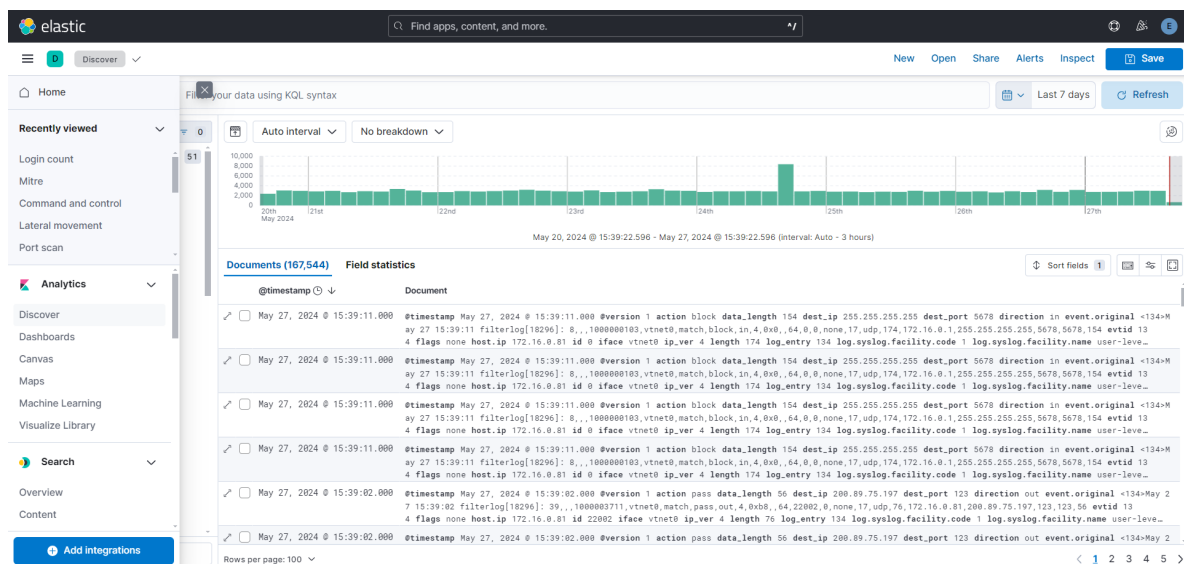


Figura 6.6: Llegada de eventos a Kibana

En la sección anterior se pueden filtrar los logs ingresando los valores de los campos existentes, el intervalo de tiempo que se quiere analizar y todos los logs asociados a todas las condiciones puestas, como se muestra en la figura 6.7:

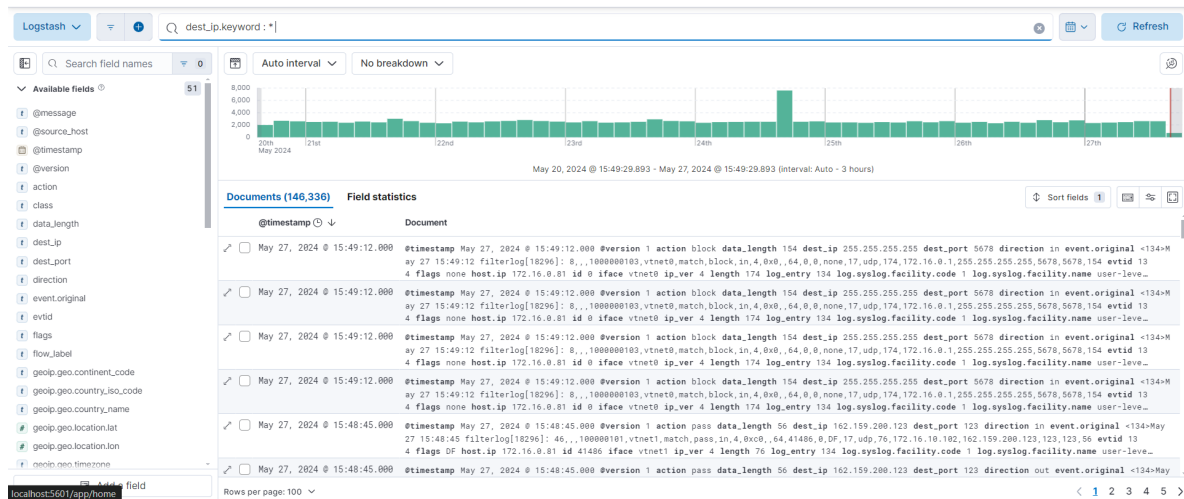


Figura 6.7: Filtro de eventos en Kibana

Teniendo ya una ingesta de datos correcta, se realizó la revisión de campos e información que deben ser analizados para poder detectar distintos ataques.

A continuación se enumeran las tácticas que se usaron para poder detectar los ataques:

1. **Reconnaissance:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de escaneos de IP dentro de la red interna.
2. **Initial access:** La técnica que se utilizará en este caso para detectar este tipo de ataques es el conteo de intentos de login en las máquinas clientes.
3. **Execution:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de ejecuciones de comandos como superusuario (sudo) en los mensajes de sistema las máquinas clientes.
4. **Persistence:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de modificaciones a tareas programadas (crontab) en los mensajes de sistema las máquinas clientes.
5. **Privilege escalation:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de creación o modificaciones a procesos de sistema (systemd) en los mensajes de sistema las máquinas clientes.
6. **Defense evasion:** Las técnicas que se utilizará en este caso para detectar este tipo de ataques son:
 - a) **Abuso de mecanismos de elevación de control:** Detección de mensajes con el campo chmod o setenforce 0 en los mensajes de sistema de las máquinas clientes.
 - b) **Obstaculización de defensas:** Detección de modificación o des-habilitación del firewall de sistema en los mensajes de sistema de las máquinas clientes.
7. **Credential access:** Las técnicas que se utilizará en este caso para detectar este tipo de ataques son:

- a) **Intentos de login fallidos:** Detección de la cantidad de intentos de login fallidos en las máquinas clientes.
 - b) **Volcado de credenciales del sistema operativo:** Detección de la presencia de `/etc/passwd` o `/etc/shadow` en los mensajes de sistema de las máquinas clientes.
 - c) **Sniffing de red:** Detección de la presencia de `tcpdump` en los mensajes de sistema de las máquinas clientes.
8. **Discovery:** Las técnicas que se utilizará en este caso para detectar este tipo de ataques son:
 - a) **Descubrimiento de cuentas:** Detección de la presencia de `/etc/passwd` en los mensajes de sistema de las máquinas clientes.
 - b) **Descubrimiento de servicios de red:** Detección de presencia de escaneos de puertos dentro de la red interna.
9. **Lateral movement:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de tráfico asociado al protocolo SMB dentro de la red interna.
10. **Collection:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de DHCP spoofing (`dhclient+DHCPACK`) en el tipo de programa y los mensajes de sistema las máquinas clientes.
11. **Command and control:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de tráfico asociado a puertos poco comunes dentro de la red interna. Además, se agregó una base de datos que identifica si la dirección IP de destino es conocida como maliciosa.
12. **Exfiltration:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de tráfico asociado al protocolo FTP dentro de la red interna.
13. **Impact:** La técnica que se utilizará en este caso para detectar este tipo de ataques es la presencia de conexiones repetidas a las mismas IPs dentro de la red interna.

Para tener completamente centralizada la información, se creó un dashboard que resume la información más relevante y se agregaron links a dashboards completos con la información detallada de cada caso. Esto se ve en la figura 6.8:

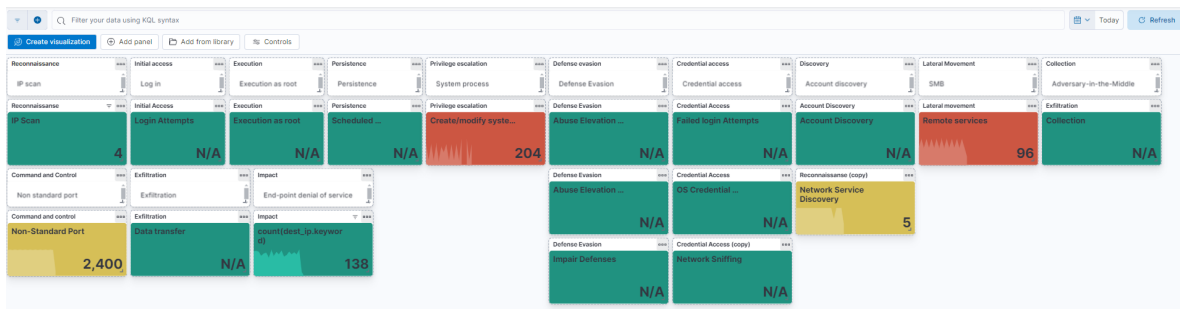


Figura 6.8: Dashborad con información centralizada

Como se observa en la figura 6.8, se aprecia que la información numérica entregada tiene asociado un color de fondo variable de acuerdo a rangos de valores establecidos. Estos colores van varian-

do a medida que se registra mayor cantidad numérica de eventos del tipo analizado. Cuando la cantidad de eventos sobrepasa el rango establecido se cambia el color, indicando que se podría estar detectando algún tipo de ataque. Los rangos de cambio de color se tienen que establecer por cada tipo de eventos analizados y el volumen correcto que aplica a la red. En este caso solo se tienen dos máquinas cliente, por lo que los valores de los límites entre rangos son bajos. A medida que aumenta la cantidad de clientes en la red, aumentan los valores límites de los rangos.

La herramienta Kibana tiene integrado en su sección de seguridad la creación automática de un dashboard con las técnicas MITRE, pero para poder implementarlo, se deben realizar los filtros con los campos establecidos por ellos, y el dashboard MITRE no permite modificaciones. Por lo mismo, se tiene menos control sobre la información, ya que si hay actualizaciones o cambios realizados por ellos, no se traduce inmediatamente al sistema propio.

Teniendo ya completamente lista la arquitectura del ambiente controlado y desplegado el SIEM, se debe proceder a implementar y ejecutar el ataque DoS/DDoS para verificar la detección de este por parte del SIEM.

La correlación que se utiliza en este caso es el volumen de tráfico detectado, información entregada por el firewall, y si la dirección IP de destino corresponde a alguna de las direcciones IPs internas, se extrae de la información enviada por los clientes.

7 | Modelamiento de amenaza

7.1. Modelo OWASP

El modelado de amenazas es un proceso clave para la seguridad de aplicaciones. Se trata de analizar un sistema desde la perspectiva de un atacante para identificar vulnerabilidades potenciales, en este caso un DoS/DDoS. Se realiza idealmente al inicio del ciclo de desarrollo de software y se actualiza constantemente (Open Web Application Security Project (OWASP) (n.d.)).

Según el Manifiesto del Modelo de Amenazas, el proceso de modelado de amenazas debe responder a las siguientes cuatro preguntas:

- ¿En qué estamos trabajando?
- ¿Qué puede salir mal?
- ¿Qué vamos a hacer al respecto?
- ¿Hicimos un trabajo lo suficientemente bueno?

7.1.1. Beneficios:

- **Detección temprana de riesgos:** Permite identificar y mitigar problemas de seguridad desde el diseño, por ejemplo, puertos expuestos.
- **Mayor conciencia de seguridad:** Ayuda a los desarrolladores a pensar como atacantes y aplicar medidas de seguridad, detectando tempranamente posibles vulnerabilidades.
- **Mejora la visibilidad del sistema:** Requiere un profundo entendimiento del sistema y sus flujos de datos.

7.1.2. Proceso:

1. **Modelado del Sistema:** Se crea un diagrama visual (DFD) que muestra los componentes del sistema y su interacción. En caso de DoS/DDoS, cualquier servicio expuesto a internet es un posible objetivo.
2. **Identificación de Amenazas:** Se identifican posibles amenazas usando técnicas como STRIDE (Suplantación, Alteración, Repudio, Divulgación de Información, Denegación

de Servicio, Elevación de Privilegios). En el caso de un DoS/DDoS, se deben buscar los servicios internos expuestos a tráfico desde internet.

3. **Respuesta y Mitigaciones:** Se definen acciones para mitigar las amenazas identificadas (ej. medidas de seguridad). Para evitar DoS/DDoS, se busca implementar análisis y restricciones de tráfico entrante, a nivel de aplicación y/o internet.
4. **Revisión y Validación:** Se revisa el modelo de amenazas para asegurar que sea completo y las mitigaciones sean adecuadas.

El modelado de amenazas es un proceso continuo que debe actualizarse junto con el sistema (Open Web Application Security Project (OWASP) (n.d.)). No existe un estándar único, pero la mayoría de enfoques siguen pasos similares a los descritos. Un modelo de amenazas bien hecho ayuda a construir sistemas más seguros.

7.2. Modelo de amenazas para la red

De acuerdo a la red modelada y simulada, se tiene el siguiente modelo de amenazas, como se ve en la figura 7.1:

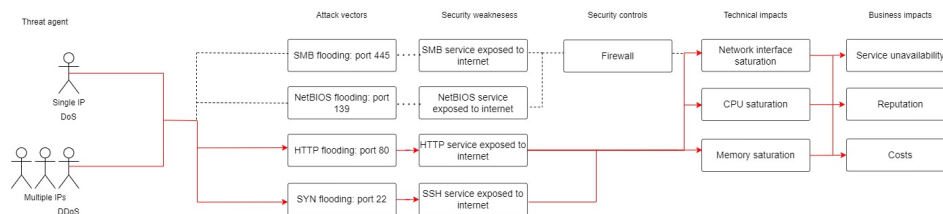


Figura 7.1: Modelamiento de amenazas

Descripción: Los ataques DoS/DDoS buscan saturar un sistema con un flujo masivo de tráfico, con el objetivo de indisponibilizar el servicio para usuarios legítimos.

7.2.1. Vectores de ataque

Dados los servicios/puertos expuestos a internet, se tienen los posibles vectores de ataque:

- **Inundaciones SYN:** Envían solicitudes SYN falsas al servidor objetivo, agotando los recursos de conexión. Son dirigidas al puerto TCP 22 (SSH) para agotar las conexiones disponibles.
- **Ataques HTTP:** Envían solicitudes HTTP masivas al servidor objetivo, agotando recursos de CPU y memoria. Son dirigidas al puerto TCP 80 (HTTP) para sobrecargar el servidor web.
- **Ataques NetBIOS:** Dirigidos al puerto TCP 139 (NetBIOS) para sobrecargar el puerto expuesto.
- **Ataques SMB:** Dirigidos al puerto TCP 445 (SMB) para sobrecargar el puerto expuesto.

7.2.2. Pasos del ataque

Los pasos del ataque se describen a continuación:

1. **Reconocimiento:** El atacante identifica un sistema objetivo y sus vulnerabilidades. Un simple escaneo de puertos permite identificar un o varios objetivos.
2. **Preparación:** El atacante prepara herramientas, bot-nets o recursos para lanzar el ataque. Esto depende si es un DoS, donde solo se requiere una dirección IP para lanzar el ataque o en caso de un ataque DDoS, se requieren bot-nets o varias máquinas con direcciones IP distintas para poder efectivamente calificarlo como distribuido.
3. **Lanzamiento:** El atacante inicia el ataque, utilizando uno o varios vectores de ataque, dependiendo de sus recursos y de la identificación de los vectores.
4. **Sobrecarga:** El sistema objetivo se satura con tráfico, causando interrupciones del servicio.
5. **Explotación:** El atacante puede aprovechar la interrupción del servicio para robar datos, extorsionar o dañar la reputación de la víctima.

7.2.3. Impacto

El impacto de un ataque exitoso de este tipo se observa en:

- **Disponibilidad:** Los usuarios legítimos no pueden acceder al servicio.
- **Reputación:** La reputación de la organización se ve afectada.
- **Pérdidas Financieras:** La organización incurre en gastos de mitigaciones, respuesta a incidente y re-disponibilizar los servicios.

7.2.4. Prevención

Las medidas de prevención que se deberían aplicar a la red descrita son:

Controles de seguridad de red

- **Fortalecer el firewall:**
 - Configurar reglas de firewall restrictivas.
 - Limitar el número de conexiones simultáneas.
 - Habilitar la protección contra ataques DDoS.
 - Filtrar el tráfico malicioso.
- **Implementar medidas de seguridad adicionales:**
 - Firewall de aplicaciones web (WAF).
 - Sistema de detección de intrusos (IDS) o Sistema de prevención de intrusos (IPS).

- Servicios de mitigación DDoS.
- **Actualizar el software y los servicios:**
 - Aplicar parches de seguridad puntualmente.
 - Mantener el software actualizado con las últimas versiones.
- **Realizar pruebas de penetración periódicas:**
 - Identificar y corregir vulnerabilidades antes de que sean explotadas.
 - Evaluar la eficacia de las medidas de seguridad.

El modelado de amenazas es una herramienta esencial para la respuesta a incidentes efectiva y medidas de seguridad generales. Al comprender las amenazas potenciales a los activos de un sistema, las organizaciones pueden tomar medidas pro-activas para detectar, contener, mitigar y recuperarse de incidentes de seguridad.

Debido a los cambios que se tuvieron que realizar en la arquitectura original para ejecutar de manera correcta un ataque DoS/DDoS, se implementó el gestor de contenedores Docker en la VM MAL (Máquina Virtual que simula internet) para instanciar la imagen pre-configurada con la aplicación JMeter (JMeter es una herramienta de prueba de carga que, con ciertas modificaciones, va a permitir ejecutar ataques DoS y DDoS) en las máquinas internas, buscando verificar que se puede implementar el DoS a partir de un solo contenedor ejecutando el JMeter y luego convertirlo en un DDoS, ya que se ejecutan simultáneamente las mismas pruebas desde varios contenedores con direcciones IP distintas.

El diagrama lógico de la red implementada se puede observar en 7.2.

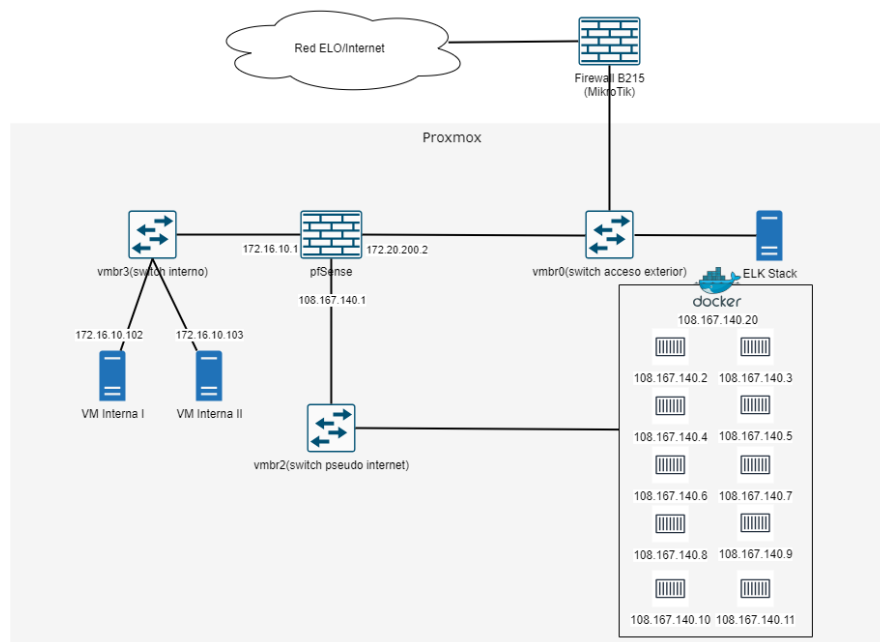


Figura 7.2: Diagrama lógico de la red

En primera instancia, se instaló y configuró el gestor de contenedores Docker en la VM MAL para que pudiera ejecutar contenedores. Se tuvo que configurar una red macvlan en el Docker para permitir que los contenedores tuvieran direcciones IP distintas entre ellos y así, al momento de ejecutar el ataque, se cumpliera con la condición de distribución.

Luego se descargó desde Dockerhub una imagen de contenedor de Alpine Linux con la aplicación JMeter en el Docker.

A continuación, se configuró el JMeter para ejecutar un ataque DoS. Esto implicó especificar los parámetros de la prueba, la dirección IP del objetivo (172.16.10.102), la cantidad de solicitudes y la duración del ataque. Para que cada contenedor tuviera una dirección IP distinta al momento de instanciarse, se tuvo que configurar la red macvlan del Docker.

La configuración de la cantidad de solicitudes se realizó definiendo la cantidad de threads a realizar en el JMeter. Con 200 threads configurados, cada prueba ejecutada por contenedor desplegado realiza un aproximado de 800 queries en el tiempo asignado. El tiempo dado para la prueba es de un minuto. Por lo observado al realizar la prueba con los contenedores, estos valores presentan variaciones. Las variaciones se atribuyen a que se están instanciando 10 contenedores simultáneos que se instanciaron en serie, consumiendo recursos de manera distinta a la que está diseñado el JMeter ejecutado desde un solo contenedor.

Para el caso del DDoS, se ejecutaron ataques DoS desde 10 contenedores diferentes instanciados simultáneamente. Cada contenedor tenía una dirección IP diferente, lo que significa que las solicitudes registradas en los logs provenían de múltiples máquinas, convirtiéndolo en un DDoS.

8 | Resultados

8.1. Implementación de ataque DDoS

Durante el ataque, se monitorizó la disponibilidad del servidor web de la máquina atacada (S1LW) desde la otra máquina cliente (S2LW). Esto permitió observar el impacto de los ataques en los tiempos de respuesta del servicio web.

Una vez desplegados los contenedores necesarios para el ataque DDoS, generados desde el interior de la red sin acceso a internet, se pudo comprobar que lograron sobrecargar el servidor web en la VM cliente objetivo (S1LW).

La VM cliente no atacada (S2LW) y sin acceso al servicio no se vió afectada por los ataques, pero, tal como se quería, no pudo acceder al servicio solicitado, logrando comprobar que el ataque efectivamente denegó el servicio.

Durante una sesión posterior, se determinó el umbral de sesiones concurrentes que resiste el servicio web atacado (S1LW en el puerto 80).

El valor se determinó realizando las pruebas con, en un inicio, un contenedor programado con 800 queries, para luego ir agregando un contenedor más con 800 pruebas más cada vez. De esta manera, se miden los tiempos de respuesta con el código de estado http y se determinó la cantidad de queries totales programadas que el servicio resiste sin perjudicar el tiempo de respuesta en gran medida.

En la figura 8.1 se observan los tiempos de respuesta y el código de estado de parte del servicio http al instanciar 1 contenedor que tiene programado realizar 800 queries al servicio web durante la prueba.

```

[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.004
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.004
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.003
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.004
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.003

```

Figura 8.1: Tiempo de respuesta con 1 contenedor instanciado

Respecto a la prueba anterior, se pudo observar que 1 contenedor que realiza 800 queries totales programadas no afecta realmente el servicio. Los tiempos de respuesta observados son todos similares y muy bajos (todos menores a 0.005 ms), análogos al caso donde no hay contenedores realizando pruebas de carga. Todos los casos presentan código de estado http 200 (exitoso).

En la figura 8.2 se observan los tiempos de respuesta y el código de estado de parte del servicio http al instanciar 2 contenedores que tienen cada uno programado realizar 800 queries al servicio web durante la prueba.

```

[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.002
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.002
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 2.770
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 2.735
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 3.720
[fw@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.106

```

Figura 8.2: Tiempo de respuesta con 2 contenedores instanciados

Respecto a la prueba anterior, se pudo observar que 2 contenedores que realizan 1600 queries totales programadas no afectan significativamente al servicio. Los tiempos de respuesta observados son bajos en un principio, para luego observar un alza leve (todos bajo los 4.000 ms). El servicio se ve levemente afectado en temas de tiempo de respuesta, pero se observa que el servidor logra responder de manera exitosa después de un breve tiempo de espera.

En la figura 8.3 se observan los tiempos de respuesta y el código de estado de parte del servicio http al instanciar 3 contenedores que tienen cada uno programado realizar 800 queries al servicio web durante la prueba.

```

[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.005
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 2.212
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 1.390
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.002
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 10.726
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 13.767
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.974

```

Figura 8.3: Tiempo de respuesta con 3 contenedores instanciados

Respecto a la prueba anterior, se pudo observar que 3 contenedores que realizan 2400 queries totales programadas afectan levemente al servicio. Los tiempos de respuesta observados son bajos en un principio, para luego observar un alza leve (en torno a 13.800 ms). El servicio se ve levemente afectado en temas de tiempo de respuesta, pero se observa que logra responder de manera exitosa después de un tiempo de espera moderado. Este tiempo no es en sí alto, pero ya es suficiente para que un usuario que solicita el servicio observe que la disponibilidad está afectada.

En la figura 8.4 se observan los tiempos de respuesta y el código de estado de parte del servicio http al instanciar 4 contenedores que tienen cada uno programado realizar 800 queries al servicio web durante la prueba.:

```

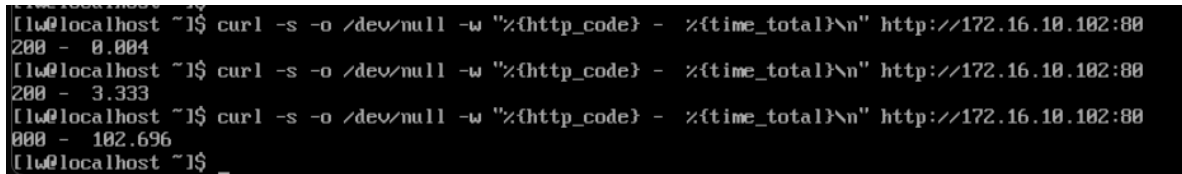
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.002
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 2.701
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 14.425
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 35.137
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.002

```

Figura 8.4: Tiempo de respuesta con 4 contenedores instanciados

Respecto a la prueba anterior, se pudo observar que 4 contenedores que realizan 3200 queries totales programadas comienzan a afectar significativamente al servicio. Los tiempos de respuesta observados son altos (llegando a 35.200 ms). El servicio se ve afectado en temas de tiempo de respuesta, pero se observa que logra responder de manera exitosa después de un tiempo de espera alto. Este tiempo de espera ya es significativo para un usuario que solicita el servicio. Con el tiempo de espera máximo observado en este caso, el usuario probablemente se impacienta y reitera la solicitud en vez de esperar la resolución. En este caso la disponibilidad está afectada.

En la figura 8.5 se observan los tiempos de respuesta y el código de estado de parte del servicio http al instanciar 5 contenedores que tienen cada uno programado realizar 800 queries al servicio web durante la prueba.



```
[lwe@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.004
[lwe@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 3.333
[lwe@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
000 - 102.696
[lwe@localhost ~]$
```

Figura 8.5: Tiempo de respuesta con 5 contenedores instanciados

Respecto a la prueba anterior, se pudo observar que 5 contenedores que realizan 4000 queries totales programadas afectan significativamente al servicio. Los tiempos de respuesta observados son muy altos y se observa un código de estado http 000. El código mencionado no está definido en la tabla de códigos de estado. El retorno de este código puede significar que no se recibió ningún código http por un error de red o que el cliente se desconectó antes de completar la solicitud. Observando el código anterior, se puede afirmar que el servicio está denegado en el momento de la solicitud. Este tiempo de espera no es aceptable para un usuario que solicita el servicio, menos si no se logra una respuesta exitosa. Este tiempo de espera ya implica que el usuario va a reiterar las solicitudes, empeorando el problema ya observado y probablemente no logre nada mientras esté activa la prueba.

Con los resultados de las pruebas anteriores, se observó que la cantidad de sesiones simultáneas máximas que aguanta el servidor web es de aproximadamente 3200 queries programadas.

Una vez ejecutados los ataques, se puede afirmar que es posible ejecutar ataques DoS y DDoS efectivos desde el interior de una red sin acceso a internet utilizando herramientas como JMeter y Docker.

Los ataques DDoS son más difíciles de mitigar que los ataques DoS dada la característica de distribución de la amenaza, pero ambos pueden ser útiles para realizar pruebas como estas.

Observando la simplicidad con la que se montó y desplegaron distintos ataques, las redes internas deben contar con medidas de seguridad adecuadas para protegerse contra este tipo de ataques externos.

Desde el punto de vista del ataque, la efectividad de los ataques DoS/DDoS puede variar dependiendo de la configuración del sistema objetivo y de los recursos disponibles para el atacante.

Desde el punto de vista de la defensa, es indispensable utilizar herramientas de monitoreo y análisis de tráfico para detectar y prevenir este tipo de ataques.

Finalmente, se debe agregar que este experimento se realizó en un entorno controlado con fines educativos y de investigación. No se debe intentar replicar en redes reales sin la debida autorización y conocimiento técnico.

8.2. Resultados SIEM

Una vez que se implementaron y ejecutaron las componentes necesarias para el ataque específico y se verificó el funcionamiento esperado de las componentes involucradas en el ataque, se tuvo que verificar que el sistema SIEM detectara efectivamente el ataque y entregara la información de manera de fácil acceso e interpretación.

8.2.1. SIEM estado inicial

Los dashboards sin un ataque activo presentan la siguiente vista inicial:

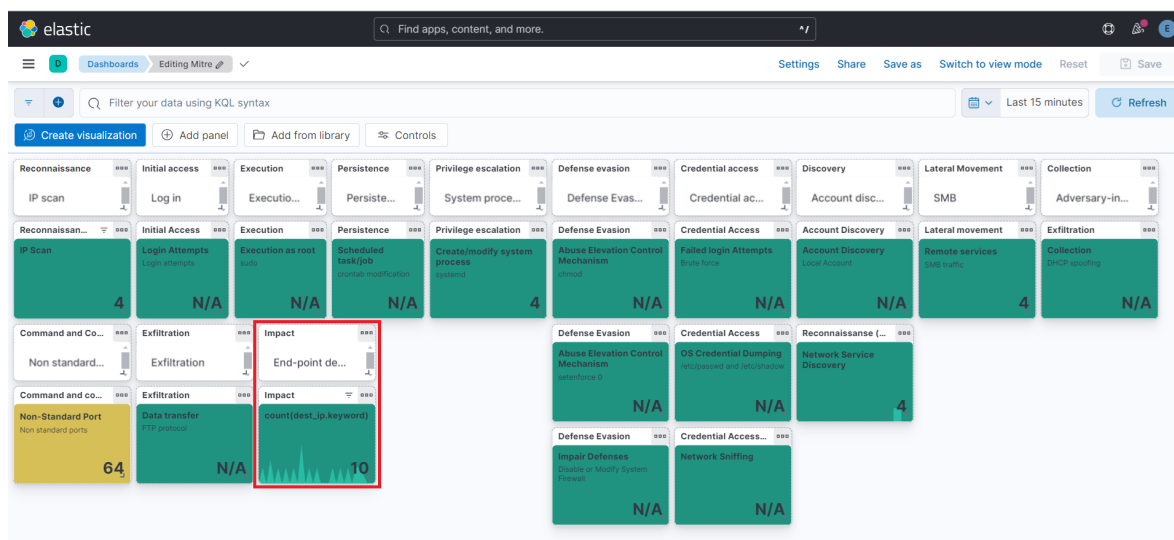


Figura 8.6: Dashboard Mitre inicial

En la imagen 8.6 los dos bloques marcados en el recuadro rojo son los relevantes para estas pruebas. El recuadro inferior cambia de color cuando se detecta un exceso de conexiones por minuto a una misma dirección IP que esté dentro de las redes internas en un intervalo de tiempo acotado. El intervalo analizado se va ajustando en el filtro superior derecho.

En el bloque superior destacado en la imagen 8.6, se tiene un link hacia el dashboard específico relacionado a la actividad sospechosa indicadora de ataques de impacto.

En caso de verificar que el bloque inferior ha cambiado a color rojo, se selecciona el link y se obtiene información acotada respecto a la actividad registrada en distintos momentos.

Al seguir el link, se despliega el dashboard asociado a impacto, visto en 8.7.

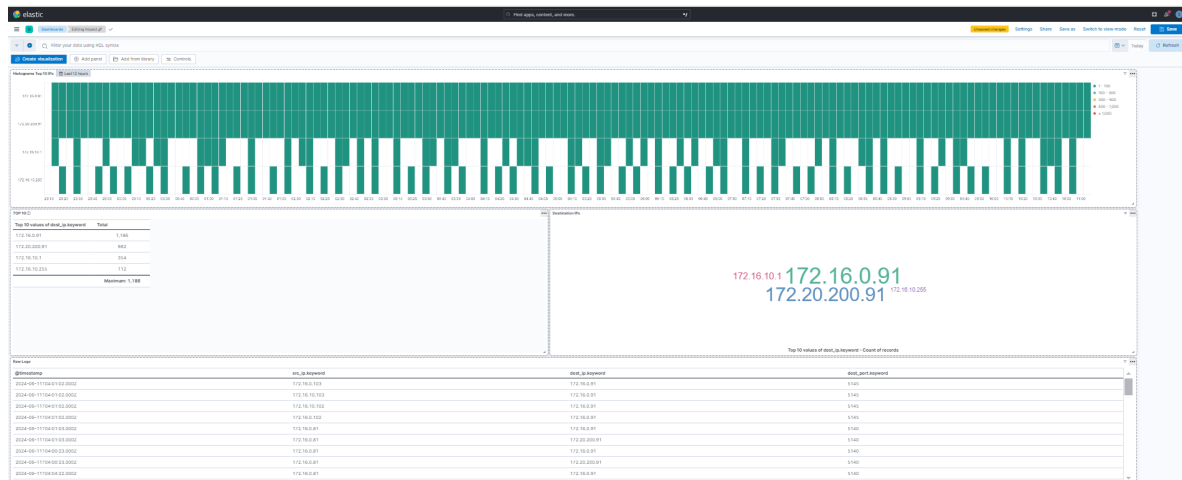


Figura 8.7: Dashboard Impact

La información que entrega este dashboard es, en primer lugar, el tráfico detectado hacia alguna dirección IP interna en las últimas 12 horas y su volumen en rangos de 5 minutos. Esta información permite detectar rápidamente si se ha observado algún comportamiento anómalo en las últimas 12 horas, teniendo de esta manera referencias de lo que ha estado sucediendo en la red.

Luego se tiene una tabla que indica el total de conexiones a las 10 distintas direcciones IP internas más activas. El intervalo de tiempo se puede ir ajustando en el filtro de tiempo.

Junto a la tabla se tiene una gráfica llamada "Tag cloud" que indica el parámetro, en este caso es la dirección IP de destino que mayor actividad presenta, facilitando la detección visual de un destino normal o anómalo.

A continuación se tiene un registro de Raw logs que entrega información de logs relevantes para el caso analizado. En esta instancia se entrega el timestamp del log, la dirección IP de origen, la dirección IP de destino y el puerto de destino.

Finalmente se agregó un gráfico que muestra la dirección IP de origen. Este gráfico es más bien de información complementaria, pero indica si la actividad se relaciona a un DoS o un DDoS.

Toda la información entregada por este dashboard es información condensada que se considera relevante para la detección y el análisis de este tipo de ataques.

Algunos datos, como la dirección IP de destino, son más relevantes para el caso, pero se requiere información complementaria, como la dirección IP de origen, para determinar detalles. La identificación entre un DoS y un DDoS depende completamente de la o las direcciones IP de origen.

Ambos dashboards descritos en 8.6 y 8.7 cambian radicalmente en la siguiente sección una vez que se ejecutó el ataque.

8.3. Ejecución y detección del ataque

El día martes 11 de junio, entre las 14:00 y 14:30 se realizó y documentó el ataque DDoS para verificar la detección del SIEM.

Para realizar el ejercicio completo, se utilizaron todas las máquinas de la red que se crearon.

La máquina VM LWMAL, con dirección IP: 108.167.140.20, es la que tiene instalada el gestor de contenedores Docker que instancia 10 contenedores que están ejecutando la aplicación JMeter en la red macvlan.

Durante la ejecución del ataque, se ejecutó el comando: "sudo ./test.sh" desde la consola de la máquina. Este es el que instancia 10 containers distintos que comienzan a ejecutar la aplicación JMeter. El output se aprecia en la imagen 8.8, donde se observan los IDs de los contenedores instanciados.

```

./test.sh: line 27: -e: command not found
[lw@108-167-140-20 ~]$ sudo ./test.sh
[sudo] password for lw:
c508c7c8b8af53ad6ae630b25ef356f6c423ab038f5e15ef5d3b2f5d8efdc17e
3750083c03f6f0789669350a4ab539379146ae08a90e1f98941bd2f25d71acb7
46b22f856973daf13c03db0b257b8bf848bb7ffd0e98e5b01604bb1ee9ad8c68
7c9eb305c5d9ee27cfda7c83032f16aec053a6da679596176e68831a7e281d5e
687ddbaba313168611d425d86b7d51b145c3db65dadce15e02b722818a388d2f
5b28da6cb246551c1fc19602bb675ca0d1e563926cbcbcb24493616724b588c09
324a20af58bb588c7775abacb2b9f48d4270168009abd9f26988847b47b2b2ad
3bc633c2190dfc8f008f213c617002e9b696ad8ee1b5c887a92ff3e64784f1ce
0f4a8e21b203578a84caeecf793d7d8ddf228a6d921fc8cbff07ed37069250fd
3b7adfbb6f1630cd2e192ae68451fc7abc77bd7457a48dac81923c7be0d1900dc
./test.sh: line 27: -e: command not found
[lw@108-167-140-20 ~]$

```

Figura 8.8: Containers instanciados

En otra consola de la misma máquina, se ejecutó el comando: "sudo docker container ls". El output de este comando, que se observa en la imagen 8.9, indica el ID del container, la imagen que está utilizando, el comando, la creación del container, el estado del container y su nombre. En este caso, una vez que el contenedor se corre, se destruye, por lo que la información entregada por created y status es prácticamente la misma. Este comando se utilizó para monitorear que los 10 containers estuvieran instanciados y que estuvieran corriendo alrededor de un minuto para asegurarse que la máquina objetivo estuviera en pleno ataque.

```
[lwm@188-167-148-20 ~]$ sudo docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3b7adfb6f163	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-10
0f4a8e21b203	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-9
3bc633c2190d	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-8
324a20af58bb	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-7
5b28da6cb246	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-6
687ddbeba313	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-5
7c9eb305c5d9	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-4
46b22f856973	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-3
3750883c03f6	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-2
c508c7c8b8af	alpine/jmeter	"/entrypoint.sh -Dlog"	About a minute ago	Up About a minute		jmeter-1

```
[lwm@188-167-148-20 ~]$ sudo docker container ls
```

Figura 8.9: Lista de containers

Finalmente, se verificó el consumo de recursos de la máquina LWMAL con el comando: *"top"* en una tercera consola, cuya respuesta se observa en la figura 8.10.

```
top - 17:14:11 up 19 days, 1:16, 3 users, load average: 4.30, 1.69, 0.79
Tasks: 245 total, 1 running, 244 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.5 us, 1.2 sy, 0.0 ni, 93.1 id, 0.0 wa, 0.0 hi, 1.1 si, 0.1 st
KiB Mem : 16264304 total, 6902076 free, 8871112 used, 491116 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used, 7100508 avail Mem
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4750	root	20	0	10.4g	836684	18076	S	26.5	5.1	0:38.73	java
5885	root	20	0	8884980	588600	18084	S	20.8	3.6	0:32.56	java
5471	root	20	0	9698108	607184	18084	S	16.3	3.7	0:32.98	java
3198	root	20	0	12.9g	901528	18172	S	12.1	5.5	0:40.53	java
5122	root	20	0	9943.5m	628732	18080	S	11.5	3.9	0:34.93	java
4259	root	20	0	11.2g	859928	18088	S	10.2	5.3	0:40.62	java
3997	root	20	0	11.6g	866272	18084	S	8.3	5.3	0:38.57	java
3462	root	20	0	12.5g	888724	18096	S	7.7	5.5	0:39.17	java
3024	root	20	0	13.3g	912320	18172	S	4.2	5.6	0:43.38	java
6584	lw	20	0	162252	2364	1512	R	3.8	0.0	0:01.69	top
2849	root	20	0	13.4g	919744	18108	S	3.2	5.7	0:43.65	java
9	root	20	0	0	0	0	S	1.9	0.0	0:46.53	rcu_sched
651	root	20	0	3401064	91260	27824	S	1.9	0.6	15:24.72	dockerd
24	root	20	0	0	0	0	S	1.3	0.0	0:17.35	ksoftirqd/3
6	root	20	0	0	0	0	S	1.0	0.0	0:30.00	ksoftirqd/0
44	root	20	0	0	0	0	S	1.0	0.0	0:15.49	ksoftirqd/7
14	root	20	0	0	0	0	S	0.6	0.0	0:14.25	ksoftirqd/1
19	root	20	0	0	0	0	S	0.6	0.0	0:19.10	ksoftirqd/2
29	root	20	0	0	0	0	S	0.6	0.0	0:17.13	ksoftirqd/4
34	root	20	0	0	0	0	S	0.6	0.0	0:15.72	ksoftirqd/5
39	root	20	0	0	0	0	S	0.6	0.0	0:16.61	ksoftirqd/6
49	root	20	0	0	0	0	S	0.6	0.0	0:19.15	ksoftirqd/8
64	root	20	0	0	0	0	S	0.6	0.0	0:15.47	ksoftirqd/11
69	root	20	0	0	0	0	S	0.6	0.0	0:14.38	ksoftirqd/12
508	root	20	0	0	0	0	S	0.6	0.0	0:28.85	xfsaild/dm-0
14874	root	20	0	0	0	0	S	0.6	0.0	6:25.99	kworker/0:0
59	root	20	0	0	0	0	S	0.3	0.0	0:15.96	ksoftirqd/10
74	root	20	0	0	0	0	S	0.3	0.0	0:14.05	ksoftirqd/13
79	root	20	0	0	0	0	S	0.3	0.0	0:11.98	ksoftirqd/14
2661	root	20	0	0	0	0	S	0.3	0.0	0:00.06	kworker/0:1
5245	root	20	0	1236448	9288	4332	S	0.3	0.1	0:00.07	containerd-shim
1	root	20	0	193708	6672	4072	S	0.0	0.0	0:29.99	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.58	kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:03.01	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	lru-add-drain
11	root	rt	0	0	0	0	S	0.0	0.0	0:33.33	watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:24.17	watchdog/1
13	root	rt	0	0	0	0	S	0.0	0.0	0:01.77	migration/1
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/1:0H
17	root	rt	0	0	0	0	S	0.0	0.0	0:23.88	watchdog/2
18	root	rt	0	0	0	0	S	0.0	0.0	0:03.42	migration/2

Figura 8.10: Consumo de recursos

Observando los resultados mostrados en la figura 8.10, se puede afirmar que la máquina está sobre-dimensionada para la prueba, ya que solo consume un aproximado de 10 Gb de memoria, teniendo un total de 16 Gb disponibles.

Una vez que se comprobó que los containers se crearon y se estaban ejecutando, se abrió una consola de la máquina víctima (VM interna 1, dirección IP:102.16.10.102) y se accedió a los

logs del apache (servicio HTTP bajo ataque) con el comando: "sudo tailf /etc/httpd/logs/access_log". En la imagen 8.11 se pueden observar los logs de acceso registrados por el apache durante el ataque.

```

100.167.140.8 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.4 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.10 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.8 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.7 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.8 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.3 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.11 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.7 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.3 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.10 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.11 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.2 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.10 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.8 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.4 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.6 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.6 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.6 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.7 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.3 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.10 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.8 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.5 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.4 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.2 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.10 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.2 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.9 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.3 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.7 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.2 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.11 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"
100.167.140.6 - - [11/Jun/2024:14:16:58 -0400] "GET / HTTP/1.1" 200 202 "-" "Apache-HttpClient/4.5.14 (Java/1.8.0_402)"

```

Figura 8.11: Apache access logs

Al analizar los logs, se puede apreciar que está recibiendo un gran volumen de solicitudes desde direcciones IPs distintas, todas del rango que se le asignó a la red de los contenedores.

Habiendo verificado que el servicio se encontraba saturado, se procedió a la consola de la tercera máquina (VM interna 2, dirección IP:102.16.10.103), y se ejecutó el comando: "curl -s -o/dev/null -w '%time_total\n' http://172.16.10.102:80". Este comando entrega el tiempo que se demora el comando: "curl http://172.16.10.102:108" en recibir la respuesta de la aplicación web del servidor. En la imagen 8.12 se observan los tiempos de demora (en ms) en 3 casos distintos.

```

[lu@localhost ~]$ curl -s -o /dev/null -w "%{time_total}\n" http://172.16.10.102:80
0.009
[lu@localhost ~]$ curl -s -o /dev/null -w "%{time_total}\n" http://172.16.10.102:80
44.639
[lu@localhost ~]$ curl -s -o /dev/null -w "%{time_total}\n" http://172.16.10.102:80
63.184
[lu@localhost ~]$

```

Figura 8.12: Tiempos de respuesta

El primer tiempo listado de respuesta es el tiempo normal que se demora en responder el servidor, de 0.009 [ms]. El segundo tiempo de respuesta que se observó es el registrado al principio del ataque, con 44.639 [ms] y finalmente el último tiempo de respuesta observado es durante el ataque en su peak, igual a 63.184 [ms].

El tiempo de respuesta que arrojó el servidor debe ser confirmado como tiempo de espera o tiempo de respuesta, por lo que se modificó el comando a: `curl -s -o /dev/null -w "%{time_total} - %{time_total} \n" http://172.16.10.102:80`. Esta modificación entrega el código de estado por parte del servidor. En el caso de los ataques generados durante los procesos de prueba (uno nuevo, ya que faltó verificarlo en primera instancia), se retornaron los códigos vistos en la imagen 8.13.

```

[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 0.010
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 38.451
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
000 - 13.909
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 37.571
[lu@localhost ~]$ curl -s -o /dev/null -w "%{http_code} - %{time_total}\n" http://172.16.10.102:80
200 - 4.828

```

Figura 8.13: Estado de respuesta

Como se observó, en la mayoría de las respuestas se tiene el código 200 (satisfactorio), pero con un tiempo de espera largo. En una de las instancias se observa el código 000, explicado en el caso de prueba de 5 contenedores para determinar la cantidad de sesiones que resiste el servicio.

Esto mostró que el ataque efectivamente denegó el servicio, afectado el rendimiento normal del servidor. La proporción que se observó ($\frac{63.183}{0.009} = 7020.4$) indicó un aumento de tiempo de respuesta de aproximadamente 7000 veces. Si un ataque similar al realizado se ejecutara durante un mayor rango de tiempo sobre una red productiva con características similares a la implementada, los servicios estarían seriamente perjudicados durante el tiempo en el que el ataque se encuentre activo.

Con el ataque ya ejecutado, se verificó que el sistema SIEM registrara correctamente el ataque.

En primer lugar, se revisó el dashboard Mitre, donde se verificó que el bloque que indica la actividad asociada a ataque de impacto se encontrara en rojo. Como se observa en la

imagen 8.14, el bloque efectivamente cambió de color e indicó el máximo de conexiones que se registraron en un minuto (12334) en el rango de tiempo establecido (últimos 15 minutos).

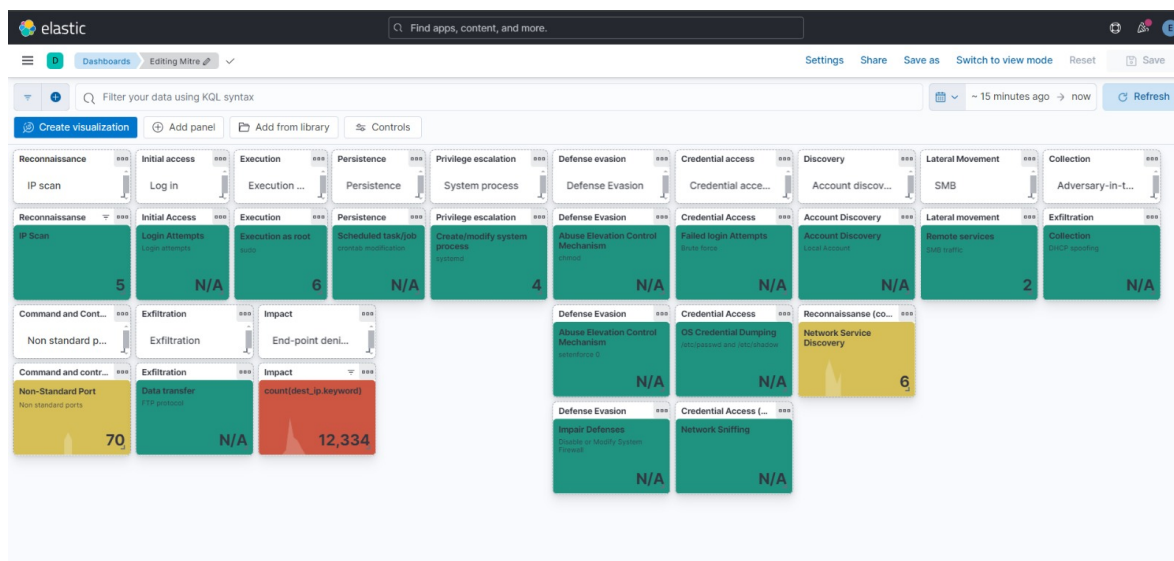


Figura 8.14: Dashboard Mitre post ataque

En segundo lugar, con el primer dashboard verificado, se siguió el link al dashboard Impact para observar la actividad en mayor detalle.

Lo primero que se observó fue el histograma presentado en la imagen 8.15.

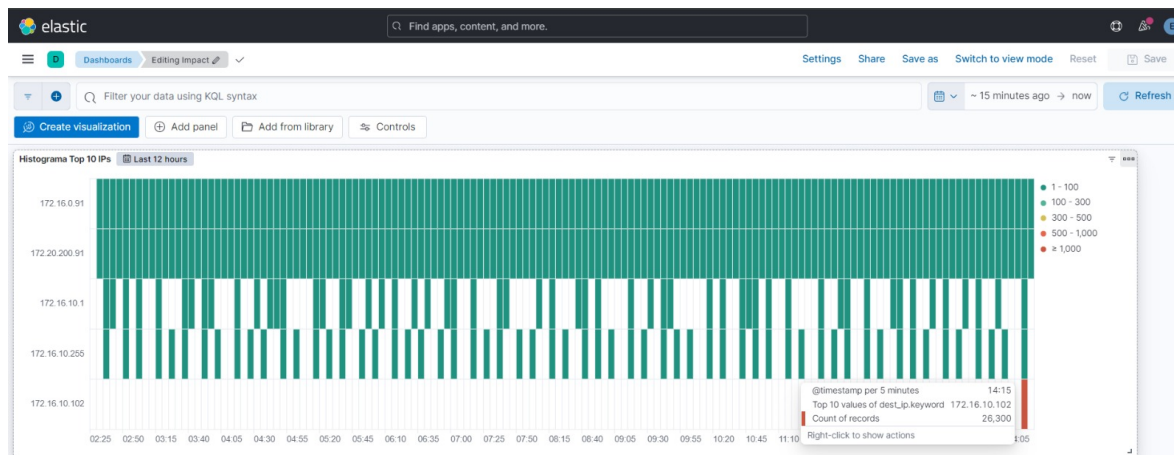


Figura 8.15: Histograma post ataque

De esa imagen, se puede observar la actividad registrada en las últimas 12 horas en rangos de 5 minutos. Las direcciones IPs indicadas a la izquierda son las que mayor actividad presentaron. La colorimetría definida en la esquina derecha superior indica la cantidad de conexiones asociado a la criticidad del color asignado. Esto depende de los rangos que se establecieron de acuerdo con la dimensión de la red. Se pudo verificar que la dirección IP 172.16.10.102 registró un total

de 26300 conexiones en un rango de 5 minutos a las 14:15, detectando exitosamente el DDoS ejecutado.

Para verificar la información que se observó en el histograma, se revisó la tabla de información y el tag cloud. La información observada en la imagen 8.16 es congruente con lo indicado por el histograma y el dashboard Mitre. La dirección IP atacada es la con mayor actividad registrada y ambas gráficas lo representan.

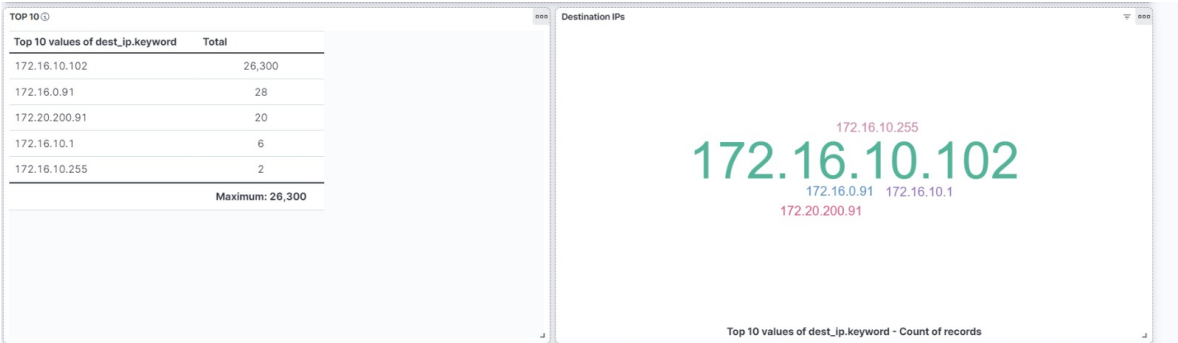


Figura 8.16: Tabla y tag cloud post ataque

La revisión de los raw logs indicó que las direcciones IP de origen son distintas entre si y el puerto de destino es el 80, información congruente con todo lo que se observó y programó durante el ataque. La información se corrobora en la imagen 8.17.

The figure shows a 'Raw Logs' table with four columns: '@timestamp', 'src_ip.keyword', 'dest_ip.keyword', and 'dest_port.keyword'. The table contains multiple rows of log entries, all showing a destination port of 80 and a destination IP of 172.16.10.102, which is consistent with the DDoS attack data.

@timestamp	src_ip.keyword	dest_ip.keyword	dest_port.keyword
2024-06-11T18:15:24.000Z	108.167.140.3	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.4	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.3	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.5	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.4	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.5	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.2	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.2	172.16.10.102	80
2024-06-11T18:15:24.000Z	108.167.140.2	172.16.10.102	80

Figura 8.17: Raw logs post ataque

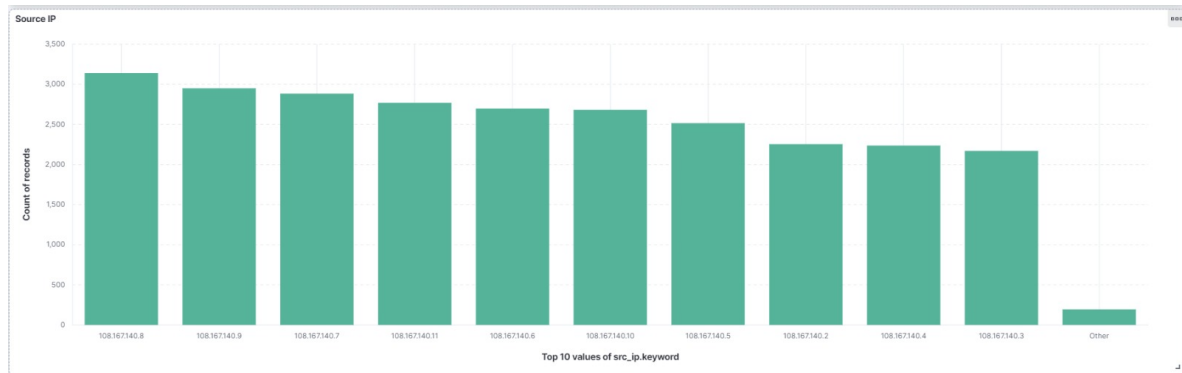


Figura 8.18: Direcciones IP de origen post ataque

Finalmente, como información complementaria, se analizó la gráfica vista en la figura 8.18 para observar gráficamente las direcciones IP de origen (en la tabla anterior ya estaban enumeradas, pero había que buscarlas) y permitir un análisis más expedito del tipo de ataque observado. En el caso documentado se observó una distribución de recursos, por lo que se puede afirmar, sin saber nada de la orquestación del ataque, que se trata de un DDoS.

8.4. Análisis de resultados

La simulación exitosa de un ataque DDoS en una red simulada y el diseño de un SIEM que detectó dicho ataque demuestran el papel fundamental de las herramientas SIEM en la detección y mitigación de amenazas. Sin la debida visibilidad de la actividad de la red, las organizaciones se ven obligadas a hacer suposiciones sobre lo que está sucediendo en la red.

La capacidad de simular ataques DDoS en un entorno controlado proporciona información valiosa sobre el comportamiento y el impacto de estas amenazas. Permite a los equipos de seguridad probar sus defensas, identificar debilidades y refinar sus procedimientos de respuesta a incidentes. Además, demuestra lo simple que es ejecutar dicho ataque a pesar de tener pocos recursos comparativamente. Considerando que la prueba de ataque que se realizó solo requiere una máquina con 10 Gb de memoria y 32 Gb de disco duro, se demostró que dentro de todo no solo fue simple aplicar el ataque, si no que fue poco costoso. Tener acceso a este tipo de infraestructura no es difícil y con el conocimiento y acceso de manuales y configuración de internet, es sencillo ejecutarlo con contenedores.

Cabe destacar que la herramienta JMeter es una herramienta de prueba de carga, pero que el método con el que fue usado durante este trabajo es poco visto. Actualmente, para generar un ataque DDoS, se puede instanciar desde servicios cloud, pero eso requiere acceso a internet. La metodología con containers de JMeter en Docker permitió circundar esta condición y abrir distintas posibilidades respecto a su uso. Al ser una red aislada sin acceso a internet, se pudo utilizar un rango de direcciones IP de origen válidas que pueden ser instaladas y usadas por máquinas de pocos recursos. Con esto se pueden probar distintas herramientas defensivas en ambientes aislados y seguros, preparando y midiendo correctamente las respuestas de los equipos defensivos. También se puede utilizar para medir la efectividad de ciertas herramientas

y su capacidad de detección y/o defensa.

Las implicancias que se pudieron detectar debido a la metodología de la ejecución del ataque se pueden relacionar directamente con los servicios de Bulletproof hosting (BPH).

Los Bulletproof hosting services (BPH) y los contenedores Docker con capacidades de orquestación DDoS están relacionados en la facilitación de actividades maliciosas. Alrwais et al. (2017) discuten la naturaleza cambiante de los servicios BPH y los desafíos que presentan. Tradicionalmente, estos servicios los ofrecían proveedores independientes. Sin embargo, en la actualidad, los proveedores de BPH revenden infraestructura de proveedores de servicios de gama baja (Alrwais et al., 2017). Esto dificulta su detección, ya que se mezclan con el tráfico legítimo.

En relación a la orquestación de ataques DDoS, los autores del estudio mencionado proponen un nuevo método para detectar los servicios BPH mediante el análisis de bloques de red. Su método utiliza datos de Whois, DNS pasivo y listas de reputación de AS (Alrwais et al., 2017). Con este método, lograron una tasa de recuperación del 98 % y una tasa de falsos positivos del 1.5 % (Alrwais et al., 2017).

Si bien el artículo de Alrwais et al. (2017) no se centra específicamente en los contenedores Docker, la relación entre los servicios BPH y la capacidad de orquestar ataques DDoS como se realizó durante este trabajo se pueden relacionar desde varios aspectos.

Los servicios BPH proporcionan anonimato y protección a los actores maliciosos. Al ofrecer servicios de alojamiento con una supervisión mínima o sin aplicación de políticas, dificultan el rastreo de los atacantes (Alrwais et al., 2017). Los contenedores Docker son ideales para implementar herramientas de DDoS. Su naturaleza ligera y portátil permite a los atacantes desplegarlos y escalarlos rápidamente (ver técnicas descritas en la sección resultados).

Combinando BPH y Docker se crea un entorno favorable para ataques DDoS. Los atacantes pueden alojar contenedores Docker con herramientas de orquestación DDoS en la infraestructura BPH. Esto les permite lanzar ataques DDoS potentes con anonimato y escalabilidad. El servicio BPH oculta la ubicación real y la identidad del atacante, mientras que los contenedores Docker facilitan el despliegue y la gestión de los bots utilizados en el ataque.

La combinación de servicios BPH y contenedores Docker con herramientas de orquestación DDoS representa una amenaza significativa para la seguridad en línea. Al comprender esta relación, los profesionales de seguridad pueden desarrollar mejores estrategias para detectar y mitigar ataques DDoS, que siempre van evolucionando y cambiando.

Por otra parte, el método utilizado durante el desarrollo del trabajo tiene otros usos asociados a la detección de botnets.

El artículo "Botnet Detection Based on Machine Learning Techniques Using DNS Query Data" por Li et al. (2019) se centra en la detección de botnets utilizando técnicas de aprendizaje automático y datos de consulta DNS. Los autores del estudio discuten las dificultades para detectar botnets debido a su naturaleza cambiante.

Proponen un método para detectar botnets basado en el análisis de patrones de consulta de DNS. Su método utiliza técnicas de aprendizaje automático como kNN, árboles de decisión, bosques aleatorios y Bayes ingenuo para clasificar el tráfico de la red como legítimo o malicioso. El método logró una alta precisión y una baja tasa de falsos positivos en los experimentos realizados por los autores (Li et al., 2019).

Si bien lanzar ataques DDoS reales es ilegal y poco ético, las simulaciones controladas dentro de un entorno seguro (como la utilizada en este trabajo) ofrecen valiosas ventajas para probar la eficacia de los métodos de detección de botnets basados en DNS, como el descrito en el artículo anterior (Li et al., 2019).

La primera ventaja es que los contenedores Docker proporcionan una forma liviana y portátil de simular la actividad de botnets. Los investigadores pueden crear escenarios realistas donde los contenedores Docker comprometidos se comunican con un servidor C&C simulado, imitando el comportamiento de botnets del mundo real. Esto permite una evaluación más precisa de la capacidad del método de detección para identificar patrones de tráfico malicioso en un entorno controlado (Li et al., 2019).

Otro factor a considerar es que las simulaciones de Docker se pueden replicar y escalar fácilmente. Esto permite a los investigadores realizar múltiples pruebas con parámetros variables, como la cantidad de bots o los tipos de ataques simulados. Esta repetibilidad asegura la consistencia y facilita las comparaciones entre diferentes métodos de detección (Li et al., 2019).

Además, a diferencia de los ataques del mundo real, las simulaciones basadas en Docker ocurren en un entorno controlado. Esto minimiza el riesgo de daños colaterales o consecuencias no deseadas que podrían afectar a los sistemas del mundo real. Los investigadores pueden experimentar y analizar libremente el comportamiento del método de detección sin causar interrupciones (Li et al., 2019).

Finalmente, las simulaciones de Docker se pueden adaptar para centrarse en aspectos específicos del comportamiento de la botnet o los patrones de comunicación. Esto permite a los investigadores identificar las fortalezas y debilidades del método de detección para identificar diferentes tipos de ataques DDoS potencialmente orquestados con contenedores Docker.

Finalizado el proceso, se logró determinar que el daño que se logró realizar, considerando que solo se instanciaron 10 contenedores en una máquina de pocos recursos, es significativo y, si se escala, podría ser altamente perjudicial para cualquier servidor objetivo .

El diseño e implementación de una solución SIEM detectó rápida y eficazmente el ataque DDoS

simulado, destacando la importancia de la gestión centralizada de la información y eventos de seguridad. Las herramientas SIEM agregan y analizan datos de diversas fuentes, lo que permite al personal de seguridad identificar anomalías, investigar posibles amenazas y tomar medidas oportunas.

9 | Conclusiones y trabajo futuro

Finalizando la implementación y detección del ataque por un sistema SIEM, se puede concluir que el uso de simulaciones DDoS alojadas en Docker ofrece un enfoque seguro, controlado y versátil para probar la eficacia de los métodos de detección de botnets basados en DNS. Este enfoque permite a los investigadores crear escenarios realistas, realizar pruebas replicables y obtener información valiosa sobre la eficacia de las técnicas de detección contra posibles ataques orquestados por Docker.

Se enfatiza el papel crucial de la visibilidad de la red en la ciber-seguridad. Sin una visibilidad completa o parcial del tráfico de la red, los equipos de seguridad se ven obligados a hacer suposiciones sobre posibles amenazas, lo que puede conducir a una detección y respuesta tardías. Además, no permite ni siquiera una acción reactiva, dado que al no poder observar cosas como la dirección IP de origen, ni siquiera se puede implementar un bloqueo a nivel de firewall de la dirección IP en caso de un DoS.

La simulación exitosa de un ataque DDoS y el desarrollo de un SIEM que lo detectó subrayan la importancia de las medidas de seguridad pro-activas y el papel fundamental de las herramientas SIEM en el refuerzo de la seguridad. Al invertir en soluciones SIEM e implementar estrategias de visibilidad de la red, la organización puede fortalecer sus defensas contra las amenazas cibernéticas y proteger sus activos. Además, el desarrollo de este trabajo demuestra que no necesariamente se necesita invertir en soluciones costosas, si no que se puede realizar una solución open-source diseñada para servir de mejor manera a los requerimientos específicos de la organización.

Se debe recalcar que los umbrales o límites puestos para que la detección de eventos sea precisa, dependen completamente de las características y el tráfico de la misma red monitoreada. No es posible implementar esta solución de manera efectiva sin un conocimiento de la propia red y el comportamiento de ésta, por lo que tanto esta solución como todas las otras disponibles en el mercado, no son automatizables y deben ser adaptadas a los ambientes que se van a monitorear por los administradores de red para tener detecciones precisas.

Finalmente, considerando los umbrales apropiados, la solución desarrollada es completamente aplicable a ambientes productivos de organizaciones.

Respecto a trabajos futuros para el SIEM desarrollado durante este proyecto, se recomiendan los siguientes objetivos:

- **Estandarizar campos de Elasticsearch**
- **Refinar los filtros del SIEM para una mejor detección de eventos:**
- **Integrar técnicas de detección de la matriz MITRE ATT&CK Enterprise faltantes**

Al implementar estas mejoras, el sistema SIEM ELK desarrollado estará mejor equipado para identificar y priorizar eventos de seguridad con mayor precisión, detectar ataques alineados con la matriz MITRE ATT&CK Enterprise y proporcionar una representación de datos de seguridad más completa, estandarizada y precisa. Mejorar los filtros implica una mejor detección de TP y TN y disminuyendo los FP y FN que podrían ser detectados, bajando la AAR. Se debe mencionar que, para poder tener la mejor correlación posible en estas herramientas, se requiere trabajar en conjunto con algún administrador de la red o alguien que tenga conocimientos respecto al flujo de datos en esta. Toda solución a implementar se debe escalar a la red y fijar umbrales que tengan sentido con el volumen de tráfico normal. La implementación de una herramienta de este tipo no tiene la capacidad de detectar y ajustar automáticamente el factor mencionado, por lo que siempre se requiere el refinamiento experto humano.

Otro factor que se puede mejorar desde el punto de vista del ataque es respecto a la fidelidad de los datos programados. Se sabe que se instanciaron 10 contenedores que, cada uno, realiza 800 queries totales durante el tiempo de prueba programado (1 minuto). Sin embargo, al realizar la prueba final del ataque, se observó que se realizaban muchas más queries, 26300 en vez de 8000, en un tiempo de más de 2 minutos en vez del minuto programado. Dado que cada contenedor JMeter se instanciaba de manera serial, es posible que el tiempo de prueba oficial comenzara a correr un vez que los 10 contenedores estuvieran instanciados, pero los primeros contenedores instanciados ya llevan mucho más tiempo arriba, realizando queries desde un principio. Esta incongruencia implica una gran diferencia entre la prueba dimensionada y la efectivamente realizada.

Considerando que la metodología de la prueba puede ser muy útil a nivel productivo, se recomienda analizar si es posible corregir esta diferencia y efectivamente cambiarla de manera que los resultados ejecutados sean iguales a los programados, teniendo menos riesgos de dañar de manera permanente los servicios puestos a prueba.

10 | Bibliografía

- AlienVault. (2023, May 12). OSSIM: The Open Source SIEM. AlienVault. Retrieved from <https://cybersecurity.att.com/products/ossim>
- Alrwais, S., Liao, X., Mi, X., Wang, P., Wang, X., Qian, F., ... & McCoy, D. (2017, May). Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks. In 2017 IEEE Symposium on Security and Privacy (SP) (pp. 805-823). IEEE.
- Brandao, P. R., & Nunes, J. (2021). Extended detection and response. Retrieved from <https://www.kriativ-tech.com/wp-content/uploads/2021/10/ExtendedDetectionResponse.pdf>
- Cole, R., Rioux, A., & Ruehl, R. (2020). Red Teams - Pentesters, APTs or Neither. In A. Rocha, H. Chen, & A. Salah (Eds.), **Cyber Deception** (pp. 25-42). Springer, Cham.
- Elastic. (n.d.). SIEM & Security Analytics. Elastic Security. Retrieved from <https://www.elastic.co/security/siem>
- González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): analysis, trends, and usage in critical infrastructures. **Sensors**, 21(14), 4759.
- Herédia, C. S., & Valença, G. M. (2019). Security Information and Event Management (SIEM) Analysis Trends and Usage in Critical Infrastructures. In *International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 223-232). Springer, Cham.
- IBM Security QRadar SIEM - Pricing. (n.d.). IBM Security QRadar SIEM - Pricing . IBM. Retrieved from <https://www.ibm.com/products/qradar-siem/pricing>
- Kinney Group. (2022, September 2). Splunk Licenses: What Size Do You Need? Here's How to Estimate It. Kinney Group. Retrieved from <https://kinneygroup.com/blog/splunk-license-estimations/>
- Laue, T., Klecker, T., Kleiner, C., & Detken, K. O. (2022). A SIEM architecture for advanced anomaly detection. **Open Journal of Big Data**, 6(1), 26-42.
- Leung, B. (2021). **Security information and event management (SIEM) evaluation report**. Doctoral dissertation, California State University San Marcos.
- Li, X., Wang, J., Zhang, X., & Zou, Z. (2019). Botnet Detection Based on Machine Learning Techniques Using DNS Query Data. **MDPI, Sensors**, 19(4), 55.

- MITRE Corporation. (n.d.). ATT&CK® for Enterprise. Retrieved from <https://attack.mitre.org/matrices/enterprise/>
- National Institute of Standards and Technology (NIST). (n.d.). Computer Security Resource Center: Glossary. Retrieved from https://csrc.nist.gov/glossary/term/red_team_blue_team_approach
- National Institute of Standards and Technology (NIST). (2016). *NIST SP 800-18 Revision 2: Guide for Developing an Information Security Incident Response Plan*. Retrieved from <https://csrc.nist.gov/pubs/sp/800/18/r1/final>
- National Institute of Standards and Technology (NIST). (2017). *NIST SP 800-30 Revision 5: Risk Management Guide for Information Systems*. Retrieved from <https://csrc.nist.gov/pubs/sp/800/30/r1/final>
- National Institute of Standards and Technology (NIST). (2018). *NIST SP 800-53 Revision 5: Security and Privacy Controls for Federal Information Systems and Organizations*. Retrieved from <https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final>
- Open Web Application Security Project (OWASP). (n.d.). OWASP Testing Guide. Retrieved from <https://owasp.org/www-project-web-security-testing-guide/>
- Open Web Application Security Project (OWASP). (2024, May 29). Threat Modeling Cheat Sheet. OWASP Cheat Sheet Series. Retrieved from https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html