



Universidad de Valladolid



Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

Implementación de un sistema de alerta temprana de amenazas de seguridad

Autor: Juan Antonio Pagés López



Universidad de Valladolid



Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
Mención en Tecnologías de la Información

Implementación de un sistema de alerta temprana de amenazas de seguridad

Autor: Juan Antonio Pagés López

Tutor: Valentín Cardeñoso Payo

A aquellos que han sido mi luz en los días oscuros, mi fuerza en los momentos de debilidad y mi inspiración en cada paso del camino. Este trabajo está dedicado con gratitud y aprecio a todos aquellos que han formado parte de mi viaje. Vuestra presencia ha sido mi mayor regalo.

*El éxito no es la clave de la felicidad. La felicidad es la clave del éxito.
Si amas lo que haces, serás exitoso. - Albert Schweitzer*

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han hecho posible la realización de este Trabajo de Fin de Grado.

En primer lugar, quiero agradecer a mi familia y seres queridos por su incondicional apoyo y paciencia a lo largo de estos años. Gracias por estar siempre a mi lado, brindándome su amor, comprensión y ánimo en los momentos más difíciles. Sin ustedes, este logro no hubiera sido posible.

A mi tutor de TFG, Valentín Cardeñoso Payo, le extiendo mi más profundo agradecimiento por su guía y asesoramiento durante todo el proceso de investigación y redacción. Su conocimiento, paciencia y dedicación han sido fundamentales para la culminación de este trabajo.

A mis amigos de la carrera, gracias por ser una fuente constante de motivación y por compartir conmigo tanto los momentos de estudio como los de ocio. Sus risas, conversaciones y apoyo han hecho que este viaje sea mucho más llevadero y agradable.

Finalmente, a mi grupo de amigos de FAINAC, gracias por ser un pilar importante en mi vida universitaria. Vuestra compañía y amistad han sido invaluable, y siempre llevaré con cariño los recuerdos de todos los momentos que hemos vivido juntos.

A todos vosotros, mil gracias.

Resumen

En un entorno cada vez más digitalizado y propenso a amenazas cibernéticas, la necesidad de contar con sistemas de alerta temprana se vuelve esencial para salvaguardar la seguridad de la información y proteger la infraestructura tecnológica de las organizaciones. Este trabajo se centra en el desarrollo e implementación de un Sistema de Alerta Temprana con el fin de detectar y responder proactivamente a posibles amenazas cibernéticas.

El objetivo principal del trabajo es diseñar, implementar y evaluar un sistema funcional y eficiente que permita monitorizar la infraestructura de forma continua, identificar comportamientos anómalos y generar alertas tempranas ante posibles ataques cibernéticos. Para lograr este objetivo, se llevarán a cabo actividades como el diseño e implementación de la arquitectura del sistema, la configuración de los componentes necesarios y la creación de reglas de detección de amenazas. Se ha optado por una implementación basada en Suricata, Elasticsearch, Kibana y Elastalert2, por ser herramientas de uso muy extendido tanto en el ámbito de la captura y señalización de tráfico de red como en la de generación de informes y gráficos de visualización.

Tras la implementación del sistema, se procederá a la evaluación de su rendimiento y efectividad en la detección y respuesta a amenazas cibernéticas. Se analizarán los resultados obtenidos y se elaborarán conclusiones sobre la viabilidad y eficacia del sistema implementado en la protección de la infraestructura de la organización contra posibles ataques. La implementación exitosa de este sistema contribuirá significativamente a fortalecer la seguridad cibernética de las organizaciones, proporcionando una capa adicional de protección y permitiendo una respuesta proactiva ante las crecientes amenazas cibernéticas.

Abstract

In an increasingly digitized environment prone to cyber threats, the need for early warning systems becomes essential to safeguard information security and protect the technological infrastructure of organizations. This work focuses on the development and implementation of an Early Warning System in order to proactively detect and respond to potential cyber threats.

The main objective of the work is to design, implement and evaluate a functional and efficient system to continuously monitor the infrastructure, identify anomalous behavior and generate early warnings of possible cyber attacks. To achieve this objective, activities such as the design and implementation of the system architecture, the configuration of the necessary components and the creation of threat detection rules will be carried out. An implementation based on Suricata, Elasticsearch, Kibana and Elastalert2 has been chosen, as they are widely used tools both in the field of network traffic capture and signaling and in the generation of reports and visualization graphs.

After the implementation of the system, its performance and effectiveness in detecting and responding to cyber threats will be evaluated. The results obtained will be analyzed and conclusions will be drawn on the feasibility and effectiveness of the implemented system in protecting the organization's infrastructure against possible attacks. The successful implementation of this system will contribute significantly to strengthening the cyber security of organizations, providing an additional layer of protection against cyber threats.

Índice general

Índice de cuadros	III
Índice de figuras	v
Índice de listados	VI
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
2. Metodología	3
2.1. Fases del proyecto	4
2.2. Costes del proyecto	6
3. Marco Conceptual	8
3.1. Ciberseguridad	8
3.1.1. Amenazas cibernéticas	8
3.2. Sistemas de Alerta Temprana	12
3.3. Sistemas de detección de intrusiones (IDS)	13
3.4. Virtualización	14
3.4.1. Máquinas virtuales	14
3.4.2. Contenedores	14
4. Soluciones Existentes	15
4.1. Sistemas de Detección de Intrusiones (IDS)	15
4.2. Recolección y envío de logs	16
4.3. Procesamiento de logs	17
4.4. Visualización y análisis de logs	18
4.5. Notificación de alertas por correo	19
4.6. Plataformas de virtualización	20
5. Herramientas de desarrollo	22
5.1. Suricata	22
5.2. Docker	23
5.3. Elasticsearch	24
5.4. Kibana	24
5.5. Elastalert2	25
5.6. Filebeat	26
5.7. Obsidian	26
5.8. Visual Studio Code	27

5.9. Microsoft Teams	28
5.10. Overleaf	29
5.11. OpenAI	30
5.12. GitHub	30
5.13. VirtualBox	31
5.14. Proxmox	32
5.15. Canva	32
6. Despliegue	34
6.1. Arquitectura	34
6.1.1. VM00 - Recolector de tráfico	34
6.1.2. VM01 - Servidor Elastic	35
6.1.3. VM02 - Kali Linux atacante	35
6.1.4. VM03 - Servidor web víctima	35
6.1.5. Subredes	36
6.2. Implementación	36
6.2.1. VM00 - Recolector de tráfico	36
6.2.2. VM01 - Servidor Elastic	49
6.2.3. VM02 - Kali atacante	66
6.2.4. VM03 - Servidor web víctima	67
7. Pruebas del sistema	71
7.1. Simulación ataque SQLI	71
7.2. Simulación ataque LFI	75
7.3. Simulación ataque RFI	79
7.4. Simulación ataque de fuerza bruta	82
7.5. Simulación ataque DDoS	90
8. Conclusiones	93
8.1. Trabajo futuro	94
Apéndices	95
Apéndice A. Repositorio del Proyecto	97
Bibliografía	98
Glosario de términos	101

Índice de cuadros

2.1. Fases de desarrollo del proyecto previstas	4
2.2. Catálogo de riesgos del proyecto	5
2.3. Plan de contingencia del proyecto	6
2.4. Presupuesto del proyecto aproximado	7
4.1. Soluciones existentes para IDS	16
4.2. Soluciones existentes para recolección y envío de logs	17
4.3. Soluciones existentes para procesamiento de logs	18
4.4. Soluciones existentes para visualización y análisis de logs	19
4.5. Soluciones existentes para notificación de alertas por correo	20
4.6. Soluciones existentes para plataformas de Virtualización	21
6.1. Descripción de las redes	36
6.2. Características VM00	36
6.3. Reglas SQLI en Suricata	42
6.4. Reglas LFI en Suricata	43
6.5. Reglas RFI en Suricata	43
6.6. Reglas de Fuerza Bruta en Suricata	44
6.7. Reglas DDoS en Suricata	44
6.8. Características VM01	49
6.9. Características VM02	66
6.10. Características VM03	67

Índice de figuras

2.1. Kanban Board de ejemplo	4
5.1. Logos de las herramientas de desarrollo empleadas en este proyecto	22
6.1. Arquitectura de red del proyecto	34
6.2. Creación de una App password para ElastAlert2 en GMAIL	58
6.3. Inicio de sesión de la página web de Kibana	65
6.4. Página principal de la web de Kibana	65
6.5. Inicio de sesión de la página web DVWA	69
6.6. Setup de la página web DVWA	69
6.7. Página principal de la página web DVWA	70
7.1. Sección de SQL Injection en la web DVWA	72
7.2. Ataque SQL Injection en la web DVWA	72
7.3. Alerta sobre SQLI en GMAIL	73
7.4. Dashboard Suricata Events sobre SQLI en Kibana	73
7.5. Dashboard Suricata Alerts sobre SQLI en Kibana	74
7.6. JSON Suricata Log sobre SQLI en Kibana	74
7.7. Sección de File Inclusion en la web DVWA	75
7.8. Ataque Local File Inclusion en la web DVWA	76
7.9. Alerta sobre LFI en GMAIL	76
7.10. Dashboard Suricata Events sobre LFI en Kibana	77
7.11. Dashboard Suricata Alerts sobre LFI en Kibana	77
7.12. JSON Suricata Log sobre LFI en Kibana	78
7.13. Ataque Remote File Inclusion en la web DVWA	79
7.14. Alerta sobre RFI en GMAIL	80
7.15. Dashboard Suricata Events sobre RFI en Kibana	80
7.16. Dashboard Suricata Alerts sobre RFI en Kibana	81
7.17. JSON Suricata Log sobre RFI en Kibana	81
7.18. Sección de Brute Force en la web DVWA	82
7.19. Configuración Proxy para Burpsuite	83
7.20. Configuración Manual Proxy para Firefox	83
7.21. Activar interceptación de solicitudes HTTP con Burpsuite	84
7.22. Envío solicitud HTTP a Burpsuite desde Firefox	84
7.23. Solicitud HTTP interceptada con Burpsuite	85
7.24. Configuración System Proxy Settings para Firefox	85
7.25. Ataque de fuerza bruta con la herramienta hydra	86
7.26. Alerta sobre Fuerza Bruta en GMAIL	87
7.27. Dashboard Suricata Events sobre Fuerza Bruta en Kibana	88
7.28. Dashboard Suricata Alerts sobre Fuerza Bruta en Kibana	88

7.29. JSON Suricata Log sobre Fuerza Bruta en Kibana	89
7.30. Ataque DDoS con la herramienta hping3	90
7.31. Alerta sobre DDoS en GMAIL	91
7.32. Dashboard Suricata Events sobre DDoS en Kibana	91
7.33. Dashboard Suricata Alerts sobre DDoS en Kibana	92
7.34. JSON Suricata Log sobre DDoS en Kibana	92

Índice de listados

6.1. Fichero configuración /etc/network/interfaces de VM00	36
6.2. Definición de redes en suricata.yaml de VM00	38
6.3. Community Flow ID en suricata.yaml de VM00	39
6.4. Interfaces de red en suricata.yaml de VM00	39
6.5. Ficheros de reglas en suricata.yaml de VM00	40
6.6. Fichero custom.rules de VM00	44
6.7. Variable dashboards en filebeat.yaml de VM00	48
6.8. Kibana setup en filebeat.yaml de VM00	48
6.9. Elasticsearch setup en filebeat.yaml de VM00	48
6.10. Fichero de configuración del módulo de Suricata de VM00	49
6.11. Fichero de configuración de red /etc/network/interfaces de VM01	50
6.12. Fichero de configuración .env de VM01	51
6.13. Fichero docker-compose.yaml de VM01	53
6.14. Fichero elastaalert2.yaml de VM01	56
6.15. Fichero smtp_auth.yaml de VM01	57
6.16. Fichero sqli_alert_rule.yaml de VM01	58
6.17. Fichero lfi_alert_rule.yaml de VM01	59
6.18. Fichero rfi_alert_rule.yaml de VM01	61
6.19. Fichero ddos_alert_rule.yaml de VM01	62
6.20. Fichero brute_force_alert_rule.yaml de VM01	63
6.21. Fichero de configuración de red /etc/network/interfaces de VM02	66
6.22. Fichero de configuración de red /etc/network/interfaces de VM03	67

Introducción

En un mundo cada vez más interconectado y digitalizado, la seguridad de la información se ha convertido en una preocupación central para empresas, gobiernos y particulares por igual. Con el avance constante de la tecnología, también han evolucionado las amenazas cibernéticas, volviéndose más complejas y sofisticadas. Por ello, surge la necesidad imperante de contar con sistemas de defensa eficaces que puedan detectar y responder rápidamente a estas amenazas.

El interés de este trabajo radica en explorar y desarrollar soluciones innovadoras que contribuyan a fortalecer la seguridad cibernética de las organizaciones. Investigar en la implementación de sistemas de alerta temprana no solo es relevante desde el punto de vista tecnológico, sino que también tiene un impacto directo en la protección de activos digitales, la preservación de la integridad de los datos y la continuidad de las operaciones empresariales.

A lo largo de este trabajo, se analizarán las características y beneficios de los sistemas de alerta temprana de amenazas, se explorarán las mejores prácticas en su implementación y se desarrollarán estrategias para su despliegue efectivo.

1.1 Motivación

En el entorno actual, marcado por la creciente complejidad y sofisticación de las amenazas cibernéticas, las organizaciones deben adoptar medidas proactivas para proteger su infraestructura de tecnologías de la información. La detección temprana de amenazas se ha convertido en un pilar fundamental de la ciberseguridad moderna, al permitir identificar y responder rápidamente a incidentes potenciales antes de que causen daños significativos. En este contexto, el despliegue de sistemas de alerta temprana de amenazas se ha vuelto esencial para la prevención y mitigación de riesgos cibernéticos, posibilitando una monitorización continua de la infraestructura de las organizaciones y la detección proactiva de posibles ataques.

1.2 Objetivos

El objetivo principal de este trabajo consiste en investigar, diseñar e implementar un sistema de alerta temprana de amenazas de seguridad basado en la integración de Suricata, Elasticsearch, Kibana y ElastAlert2, con el fin de mejorar la capacidad de detección y respuesta a incidentes de seguridad en entornos empresariales.

Así mismo, los objetivos secundarios serían los descritos a continuación:

1. Investigar, seleccionar y aprender a utilizar las herramientas y tecnologías más adecuadas para la implementación del sistema de alerta temprana.
2. Configurar y desplegar un entorno de Elasticsearch y Kibana para la recolección y visualización de datos de seguridad.
3. Integrar Suricata como sensor de red para la detección de amenazas cibernéticas en tiempo real.
4. Desarrollar reglas de detección personalizadas para Suricata con el fin de identificar comportamientos maliciosos y actividades sospechosas.
5. Configurar ElastAlert2 para generar alertas a partir de los eventos detectados por Suricata y enviar notificaciones a los responsables de seguridad.
6. Probar y validar el sistema de alerta temprana en un entorno de laboratorio simulado, evaluando su efectividad en la detección y respuesta a amenazas.
7. Documentar el proceso de diseño, implementación y pruebas del sistema de alerta temprana, proporcionando guías detalladas para su uso y mantenimiento.

Metodología

Kanban es una metodología de gestión visual que se originó en Toyota como parte de su sistema de producción Just-In-Time. Se centra en la visualización del flujo de trabajo, limitando el trabajo en progreso y maximizando la eficiencia y la calidad.

Utiliza un tablero dividido en columnas que representan diferentes etapas del proceso, y tarjetas que representan tareas individuales (Figura 2.1). Cada etapa del proceso tiene un límite de trabajo en progreso, lo que ayuda a mantener un flujo constante y evitar la sobrecarga. Ahora, vamos a describir brevemente qué implica cada etapa del proceso Kanban:

- **TO DO:** Esta es la columna inicial donde se enumeran todas las tareas que deben realizarse. Se pueden agregar nuevas tareas en cualquier momento, pero la prioridad se determina según la carga de trabajo y la necesidad del proyecto.
- **IN PROGRESS:** Cuando un elemento se mueve a esta columna, significa que se está trabajando activamente en él. Aquí es donde se aplican los esfuerzos para completar la tarea asignada.
- **BLOCKED:** Esta columna se utiliza para tareas que no pueden progresar debido a algún tipo de bloqueo. Los bloqueos pueden ser causados por dependencias externas, la necesidad de recursos adicionales o cualquier otro obstáculo que impida el avance del trabajo.
- **REVIEW:** Después de completar una tarea, pasa a esta etapa para ser revisada. Aquí se realizan controles de calidad, pruebas o cualquier otra actividad necesaria para garantizar que el trabajo se haya realizado correctamente y cumpla con los requisitos.
- **DONE:** Una vez que una tarea ha sido revisada y aprobada, se mueve a esta columna. Significa que el trabajo está completo y listo para ser entregado o implementado.

En general, la metodología Kanban proporciona una estructura flexible pero efectiva para la gestión de proyectos, lo que la hace especialmente adecuada para la realización de un Trabajo de Fin de Grado donde es importante mantenerse organizado, seguir el progreso y colaborar eficazmente en equipo. En mi caso para este TFG he utilizado la extensión comunitaria de Kanban de la herramienta Obsidian.

KANBAN BOARD



Figura 2.1: Kanban Board de ejemplo

2.1 Fases del proyecto

Una vez definida la metodología de trabajo que se va a utilizar podemos desarrollar una lista de tareas con su respectiva duración y sus posibles dependencias con otras tareas (Cuadro 2.1). Este proyecto está pensado para empezar el día 5 de febrero y finalizar el 20 de junio, sin embargo cabe destacar que para poder empezar con el proyecto se ha realizado una fase previa de investigación y la preparación de una propuesta del TFG. Además, hay que resaltar que gracias al uso de la metodología de trabajo Kanban es posible que dos tareas estén en el mismo estado simultáneamente, por lo que se podrá avanzar en la redacción de la memoria a la vez que otras tareas.

ID	Nombre de la tarea	Semanas	Dependencias
T1	Propuesta TFG	1	N/A
T2	Investigación	8	N/A
T3	Redacción memoria	19	N/A
T4	Configuración máquina Suricata (VM00)	2	N/A
T5	Configuración máquina Elastic (VM01)	2	N/A
T6	Configuración máquina Kali (VM02)	1	N/A
T7	Configuración servidor Web (VM03)	1	N/A
T8	Despliegue final en Proxmox	2	T4, T5, T6 y T7
T9	Pruebas del sistema final	2	T8

Cuadro 2.1: Fases de desarrollo del proyecto previstas

Una vez que se ha revisado la lista de tareas planificadas, se procederá a identificar y definir una serie de riesgos de alto nivel (Cuadro 2.2), proporcionando una breve descripción de cada uno, su nivel de probabilidad de ocurrencia y el impacto potencial que tendría en el proyecto en caso de materializarse.

Además, se presenta el plan de contingencia (Cuadro 2.3) elaborado para gestionar los riesgos identificados en el proyecto. Esta tabla detalla cada riesgo mediante un ID único, acompañado por su respectivo nombre y la medida específica de mitigación diseñada para contrarrestar su impacto en el desarrollo del proyecto. Esta estructura proporciona una visión clara y organizada de cómo se abordarán los riesgos a lo largo de todas las etapas del proyecto, garantizando una gestión efectiva de las posibles contingencias.

ID	Riesgo	Descripción	Probabilidad	Impacto
R01	Indisposición personal	Enfermedad o circunstancias personales que impiden trabajar en el proyecto	Media	Alto
R02	Pérdida de los ficheros del proyecto	Fallo en el respaldo o pérdida accidental de los archivos del proyecto	Baja	Alto
R03	Aumento del tiempo dedicado a la redacción de la memoria	Complejidad inesperada en la redacción o cambios de enfoque en la memoria	Media	Medio
R04	Problemas en la construcción del laboratorio	Dificultades técnicas o de recursos para establecer el laboratorio necesario para el proyecto	Media	Alto
R05	Estimación errónea del tiempo en las tareas del proyecto	Subestimación de la duración requerida para completar ciertas tareas	Alta	Medio
R06	Variación en los requerimientos del proyecto	Cambios o actualizaciones en los requisitos del proyecto	Media	Alto
R07	Falta de claridad en los requerimientos del proyecto	Ambigüedad o falta de definición en los objetivos o requisitos del proyecto	Alta	Medio
R08	Alto nivel de complejidad técnica	La implementación técnica del proyecto es más compleja de lo esperado	Alta	Alto
R09	Desconocimiento en las tecnologías usadas en el proyecto	Falta de conocimiento y experiencia en la tecnologías utilizadas en el proyecto	Media	Alto
R10	Problemas de acceso a instalaciones o recursos específicos	Puede haber dificultades para acceder a instalaciones o recursos específicos necesarios para llevar a cabo ciertas actividades del proyecto, como laboratorios de la UVa.	Media	Alto

Cuadro 2.2: Catálogo de riesgos del proyecto

ID	Riesgo	Medida para mitigar el riesgo
R01	Indisposición personal	Se dedicará tiempo extra para intentar volver a la planificación temporal original cuanto antes. De no ser posible, se retrasará la entrega a la segunda convocatoria
R02	Pérdida de los ficheros del proyecto	Todos los archivos relacionados con la elaboración del proyecto y de la memoria se almacenarán en sistemas con copia de seguridad en la nube
R03	Aumento del tiempo dedicado a la redacción de la memoria	Realizar una planificación detallada y emplear tiempo extra en la redacción de la memoria. En caso de no ser posible, se retrasará la entrega a la segunda convocatoria
R04	Problemas en la construcción del laboratorio	Se dedicará tiempo extra para intentar encontrar una solución. De no ser posible, se solicitará ayuda al tutor. Si fuera necesario, se retrasaría la entrega a la segunda convocatoria.
R05	Estimación errónea del tiempo en las tareas del proyecto	Se retrasarán las tareas posteriores. En caso de no poder finalizar a tiempo, se retrasará la entrega a la segunda convocatoria
R06	Variación en los requerimientos del proyecto	Mantener una comunicación abierta y regular con el tutor para gestionar los cambios de requisitos.
R07	Falta de claridad en los requerimientos del proyecto	Realizar reuniones periódicas con el tutor para aclarar y validar los requisitos del proyecto.
R08	Alto nivel de complejidad técnica	Se dedicará tiempo extra en investigación para llevar a cabo el proyecto. De no ser posible, se solicitará ayuda al tutor. Si fuera necesario, se retrasaría la entrega a la segunda convocatoria
R09	Desconocimiento en las tecnologías usadas en el proyecto	Para ello se realizará primero una fase de investigación para aprender y familiarizarse con las tecnologías del proyecto. Si fuera necesario, se retrasaría la entrega a la segunda convocatoria.
R10	Problemas de acceso a instalaciones o recursos específicos	Comunicarse con los responsables de las instalaciones o recursos para coordinar el acceso y resolver cualquier problema relacionado con la disponibilidad. Mientras no se pueda solucionar, se realizarán tareas que no requieran el uso de los recursos afectados.

Cuadro 2.3: Plan de contingencia del proyecto

2.2 Costes del proyecto

La tabla de costes del proyecto (Cuadro 2.4) presenta un desglose detallado de los recursos necesarios para llevar a cabo el proyecto, junto con sus correspondientes costos unitarios (C.U.) y totales (C.T.). Este presupuesto se centra principalmente en el coste de la mano de obra y en las licencias de las herramientas y tecnologías utilizadas. La mano de obra es representada por un ingeniero informático junior, cuyo coste se calcula en base a un valor por hora y al número de horas previstas para el proyecto.

Además de los costes de mano de obra, la tabla incluye las licencias de diversas herramientas tecnológicas esenciales para el desarrollo y la implementación del proyecto. Es notable que la mayoría de estas herramientas, como Suricata, Docker, Elasticsearch, Kibana, y otras, son de uso gratuito, lo cual contribuye a minimizar los costos generales del proyecto. Estas herramientas son cruciales para la creación y el mantenimiento del entorno de desarrollo y producción, así como para la gestión de datos y la automatización de tareas.

Finalmente, se presenta un coste total de **4.896€** que refleja la suma de los gastos necesarios para completar el proyecto, destacando que el principal coste económico se encuentra en la mano de obra del ingeniero informático junior. Esta estructura de costes permite tener una visión clara y precisa de los recursos financieros requeridos, facilitando así la planificación y gestión del presupuesto del proyecto.

Nombre	Descripción	C.U.	Unidades	C.T.
Ingeniero Informático Junior	Coste de la mano de obra del ingeniero informático junior para el proyecto	12€/h	408 h	4.896€
Suricata	Costo de la licencia de la herramienta Suricata (gratuita)	0€	0	0€
Docker	Costo de la licencia de la herramienta Docker (gratuita)	0€	0	0€
Elasticsearch	Costo de la licencia basic de la herramienta Elasticsearch (gratuita)	0€	0	0€
Elasticsearch	Costo de la licencia basic de la herramienta Kibana (gratuita)	0€	0	0€
Elastalert2	Costo de la licencia de la herramienta Elastalert2 (gratuita)	0€	0	0€
Filebeat	Costo de la licencia basic de la herramienta Filebeat (gratuita)	0€	0	0€
Obsidian	Costo de la licencia de la herramienta Obsidian (gratuita)	0€	0	0€
Visual Studio Code	Costo de la licencia de la herramienta Visual Studio Code (gratuita)	0€	0	0€
Overleaf	Costo de la licencia de la herramienta Overleaf (gratuita)	0€	0	0€
OpenAI	Costo de la licencia de la herramienta OpenAI (gratuita)	0€	0	0€
Github	Costo de la licencia de la herramienta Github (gratuita)	0€	0	0€
Oracle VM VirtualBox	Costo de la licencia de la herramienta Oracle VM VirtualBox (gratuita)	0€	0	0€
PROXMOX	Costo de la licencia del servidor de virtualización PROXMOX	0€	0	0€
Canva	Costo de la licencia de la herramienta Canva (gratuita)	0€	0	0€
SO Máquinas virtuales	Costo de la licencias de los sistemas operativos de las máquinas virtuales (gratuitas)	0€	0	0€
			Coste Total	4.896€

Cuadro 2.4: Presupuesto del proyecto aproximado

Marco Conceptual

En este capítulo, se explorarán y definirán los conceptos clave relacionados con la ciberseguridad, un campo esencial para la protección contra las amenazas digitales. Se discutirá la importancia de la ciberseguridad en la era digital y cómo las amenazas cibernéticas, como la inyección de comandos, el SQLI, el DDoS, entre otras, representan riesgos significativos para la integridad y la seguridad de los sistemas tecnológicos.

Además, se presentará una visión detallada de los Sistemas de Alerta Temprana, fundamentales para la detección y respuesta proactiva a posibles ataques cibernéticos, definiendo sus componentes esenciales y analizando los beneficios y desafíos de su implementación, apoyados por estudios de caso. Además, se explorarán los Sistemas de Detección de Intrusiones (IDS), explicando qué son estos sistemas, sus tipos y evolución. Finalmente, se abordará la virtualización, una tecnología clave en la infraestructura de TI moderna, definiendo sus principios básicos y aplicaciones en ciberseguridad, incluyendo el uso de contenedores y máquinas virtuales para mejorar la seguridad mediante entornos de prueba aislados y otros métodos de protección.

3.1 Ciberseguridad

La ciberseguridad se refiere al conjunto de prácticas, tecnologías y procesos diseñados para proteger los sistemas informáticos, redes, dispositivos y datos contra accesos no autorizados, ataques cibernéticos y daños, asegurando la confidencialidad, integridad y disponibilidad de la información. En un entorno cada vez más digitalizado, la ciberseguridad desempeña un papel crucial en la protección de la información sensible y la infraestructura tecnológica de organizaciones y usuarios individuales.

La creciente importancia de la ciberseguridad en la era digital se debe a la rápida expansión de las tecnologías de la información y la comunicación (TIC). Con la digitalización de procesos empresariales, la adopción de la nube, el aumento del comercio electrónico y la interconexión de dispositivos IoT (Internet de las cosas), las organizaciones se enfrentan a una creciente superficie de ataque. Las ciberamenazas, que van desde ataques de Malware y Ransomware hasta violaciones de datos y ataques de denegación de servicio (DDoS), pueden causar graves daños financieros, reputacionales y operativos a las organizaciones.

3.1.1 Amenazas cibernéticas

Las amenazas cibernéticas son acciones maliciosas diseñadas para comprometer la seguridad de sistemas informáticos, redes y datos. En el contexto de este TFG, se simularán diversas amenazas para probar el sistema de alerta temprana, incluyendo ataques como la Inclusión de Inyección de SQL (SQLI), Archivos Locales (LFI), Inclusión de Archivos Remotos (RFI), ataques de Fuerza Bruta y ataques de Denegación de Servicio (DoS).

Inyección SQL

Un ataque de inyección SQL ocurre cuando un atacante inserta o inyecta código SQL malicioso en una consulta que una aplicación web envía a una base de datos. Este tipo de ataque generalmente explota vulnerabilidades en el software que no valida adecuadamente la entrada de datos del usuario. Por ejemplo, si una aplicación permite a los usuarios ingresar datos que luego se utilizan directamente en una consulta SQL sin la debida sanitización, un atacante puede manipular esta entrada para alterar la consulta original.

Un ataque de inyección SQL puede tener múltiples impactos severos en una organización. El robo de datos confidenciales es uno de los más graves, ya que un atacante puede acceder a información personal, contraseñas, números de tarjetas de crédito y otros datos críticos. Esto no solo representa una violación de la privacidad de los usuarios, sino que también puede resultar en multas y sanciones regulatorias para la empresa afectada. Además, la manipulación de datos es otro riesgo significativo. Un atacante podría modificar, insertar o eliminar datos en la base de datos, comprometiendo la integridad de los mismos y llevando a problemas operativos y decisiones erróneas basadas en datos incorrectos.

El control del sistema es otro posible impacto, donde un atacante podría utilizar la inyección SQL para ejecutar comandos arbitrarios en el sistema operativo del servidor de base de datos, potencialmente comprometiendo todo el servidor. Esto puede llevar a una negación de servicio (DoS) si el atacante sobrecarga el servidor con consultas maliciosas, haciendo que la aplicación web se vuelva inoperable. Además, un ataque exitoso podría permitir al atacante establecer puertas traseras para mantener acceso persistente al sistema comprometido, facilitando futuros ataques sin ser detectado.

Para mitigar estos ataques, es fundamental adoptar buenas prácticas de desarrollo de software y seguridad. La validación y sanitización de entradas es esencial para asegurar que todas las entradas del usuario sean verificadas antes de utilizarlas en consultas SQL. El uso de consultas preparadas y parámetros es otra técnica eficaz, ya que separa los datos de las instrucciones SQL, evitando así que las entradas del usuario puedan alterar la lógica de la consulta. Además, la configuración de cuentas de base de datos con los mínimos privilegios necesarios puede limitar el alcance de un ataque exitoso. Finalmente, implementar auditoría y monitoreo puede ayudar a detectar y responder rápidamente a intentos de inyección SQL, mitigando potenciales daños. Estas medidas combinadas forman una defensa robusta contra los ataques de inyección SQL.

Inclusión de archivos locales (LFI)

Un ataque de inclusión de archivos locales (LFI) es un tipo de vulnerabilidad de seguridad web que ocurre cuando una aplicación permite a un usuario incluir archivos en el servidor local a través de entradas no seguras. Esto sucede generalmente cuando la aplicación no valida ni sanitiza adecuadamente las entradas proporcionadas por el usuario. Un atacante puede aprovechar esta vulnerabilidad para acceder a archivos sensibles del servidor, ejecutar código malicioso o incluso tomar control completo del sistema afectado.

En un ataque LFI, el atacante manipula las entradas para incluir archivos del servidor que no deberían ser accesibles. Por ejemplo, si una aplicación permite a los usuarios especificar qué archivo cargar mediante una URL o un parámetro de formulario, el atacante podría modificar esta entrada para incluir archivos del sistema operativo, como `/etc/passwd` en sistemas Linux, que contiene información de los usuarios del sistema. Esto se logra normalmente mediante técnicas de traversal de directorios, donde se utilizan secuencias como `../..` para navegar fuera del directorio permitido y acceder a otras partes del sistema de archivos.

Los impactos de un ataque de inclusión de archivos locales (LFI) pueden ser múltiples y severos, comenzando con el robo de datos confidenciales. A través de un ataque LFI, un atacante podría acceder a archivos sensibles del sistema, como configuraciones, contraseñas o información personal de los usuarios, comprometiendo la privacidad y seguridad de la información.

Otro impacto significativo es la ejecución de código malicioso. Si un atacante puede incluir un archivo que contiene código ejecutable, puede llegar a ejecutar comandos arbitrarios en el servidor, obteniendo el control del sistema afectado. Esto podría permitir al atacante realizar una amplia gama de acciones maliciosas, como instalar malware, crear cuentas de usuario no autorizadas o modificar archivos críticos del sistema. La capacidad de ejecutar comandos puede resultar en un control completo del servidor, convirtiendo un ataque LFI en una puerta de entrada para ataques más complejos y destructivos.

Además, un ataque LFI puede facilitar el escalado de privilegios. Un atacante que inicialmente tiene acceso limitado podría usar un ataque LFI para leer archivos del sistema que contienen información de configuración o credenciales, permitiéndole obtener mayores privilegios dentro del sistema. Esto aumenta el riesgo de comprometer otros componentes de la infraestructura de TI y afecta la integridad y disponibilidad de los servicios.

Para mitigar los ataques LFI, es esencial implementar una serie de medidas de seguridad. En primer lugar, la validación y sanitización rigurosa de todas las entradas del usuario es crucial para prevenir que datos no confiables sean utilizados para construir rutas de archivos. El uso de listas blancas para controlar qué archivos pueden ser incluidos o accedidos también es una técnica efectiva para limitar la exposición a archivos no deseados. Configurar el servidor web para deshabilitar la inclusión de archivos desde rutas no confiables y asegurar que las aplicaciones no revelen información sensible en sus mensajes de error puede reducir significativamente las posibilidades de explotación.

Adicionalmente, mantener el software y las bibliotecas actualizadas con los últimos parches de seguridad es fundamental para cerrar las vulnerabilidades conocidas que podrían ser explotadas por un ataque LFI. Implementar políticas de acceso y privilegios mínimos, junto con auditorías y monitoreo continuo del sistema, puede ayudar a detectar y responder rápidamente a intentos de inclusión de archivos locales, minimizando los daños potenciales. Estas estrategias combinadas forman una defensa robusta contra los ataques LFI, protegiendo la integridad y seguridad de la infraestructura de TI.

Inclusión de archivos remotos (RFI)

Un ataque de inclusión de archivos remotos (RFI) es una vulnerabilidad de seguridad web que ocurre cuando una aplicación permite a un usuario incluir archivos desde ubicaciones remotas a través de entradas no seguras. Este tipo de ataque es posible cuando la aplicación no valida ni sanitiza adecuadamente las entradas proporcionadas por el usuario. Un atacante puede aprovechar esta vulnerabilidad para incluir archivos maliciosos desde un servidor externo, lo que puede llevar a la ejecución de código arbitrario en el servidor afectado.

En un ataque RFI, el atacante manipula las entradas para incluir archivos ubicados en servidores remotos. Por ejemplo, si una aplicación permite a los usuarios especificar la URL de un archivo para cargar mediante un parámetro de formulario o una URL, el atacante podría modificar esta entrada para apuntar a un archivo en un servidor bajo su control que contiene código malicioso. Este código puede ser ejecutado por el servidor víctima, permitiendo al atacante tomar control del sistema, robar información sensible o causar interrupciones en el servicio.

Los impactos de un ataque RFI pueden ser extremadamente severos. Uno de los riesgos más significativos es la ejecución de código malicioso en el servidor. Al incluir y ejecutar un archivo remoto, el atacante puede introducir un shell web, malware o cualquier otro tipo de software malicioso en el servidor afectado. Esto puede llevar al compromiso total del sistema, permitiendo al atacante ejecutar comandos arbitrarios, robar datos, modificar archivos o establecer puertas traseras para acceso futuro.

Otro impacto crítico es el robo de datos sensibles. Un archivo malicioso incluido puede ser diseñado para acceder y exfiltrar información confidencial del servidor, como credenciales de usuarios, datos financieros o información personal. Esta información puede ser utilizada para realizar otros ataques, como el robo de identidad, fraudes financieros o venta en mercados ilícitos.

Además, un ataque RFI puede facilitar ataques adicionales, como la escalación de privilegios. Con acceso inicial a través de la inclusión de un archivo remoto, el atacante puede buscar otras vulnerabilidades en el sistema para aumentar sus privilegios y obtener control total sobre la infraestructura afectada.

Para mitigar los riesgos asociados con los ataques RFI, es esencial implementar varias medidas de seguridad. La validación y sanitización rigurosa de todas las entradas del usuario es crucial para evitar que datos maliciosos sean utilizados para incluir archivos no deseados. Utilizar listas blancas para controlar qué archivos pueden ser incluidos o accesibles ayuda a limitar la exposición a archivos remotos.

Configurar el servidor web para deshabilitar la inclusión de archivos desde ubicaciones remotas, asegurando que las aplicaciones no permitan la inclusión de archivos desde fuentes no confiables, es una defensa efectiva contra los ataques RFI. También es importante configurar correctamente las políticas de seguridad del servidor para restringir la capacidad de ejecutar archivos desde ubicaciones externas.

Mantener el software y las bibliotecas actualizadas con los últimos parches de seguridad es fundamental para cerrar las vulnerabilidades conocidas que podrían ser explotadas por un ataque RFI. Además, implementar políticas de acceso y privilegios mínimos, junto con auditorías y monitoreo continuo del sistema, puede ayudar a detectar y responder rápidamente a intentos de inclusión de archivos remotos, minimizando los daños potenciales.

Fuerza bruta

Un ataque de fuerza bruta es una técnica utilizada por atacantes para adivinar contraseñas o claves cifradas mediante la prueba sistemática de todas las combinaciones posibles. Este tipo de ataque se basa en el poder de cómputo y tiempo para descifrar las credenciales, y puede ser dirigido a cuentas de usuario, sistemas de autenticación, y cualquier otro mecanismo protegido por contraseñas. En un ataque de fuerza bruta, el atacante intenta ingresar diferentes combinaciones de letras, números y caracteres especiales hasta encontrar la combinación correcta que permita el acceso. Estos ataques pueden ser automatizados utilizando scripts o herramientas específicas que generan y prueban múltiples combinaciones rápidamente. Los ataques pueden ser directos, donde el atacante intenta acceder a una cuenta específica repetidamente, o distribuidos, donde el atacante utiliza múltiples IPs o dispositivos para evitar la detección y bloqueo por parte de sistemas de seguridad.

El impacto más directo de un ataque de fuerza bruta es el acceso no autorizado a cuentas de usuario. Una vez que el atacante obtiene las credenciales correctas, puede acceder a la cuenta y realizar actividades maliciosas, como robar información sensible, realizar transacciones no autorizadas, o cambiar configuraciones críticas. Si el atacante obtiene acceso a una cuenta con privilegios elevados, puede acceder a datos sensibles almacenados en el sistema, incluyendo información personal, financiera y confidencial. Esta información puede ser utilizada para cometer fraudes, robo de identidad o ser vendida en mercados ilícitos.

Los intentos repetidos de inicio de sesión pueden causar una sobrecarga en los servidores de autenticación, llevando a la degradación del rendimiento o incluso a la interrupción del servicio. Esto puede afectar la disponibilidad de servicios críticos y causar pérdidas financieras y de reputación. Con acceso a una cuenta comprometida, especialmente si es una cuenta con privilegios administrativos, el atacante puede intentar escalar sus privilegios para obtener un control más amplio sobre el sistema, comprometiendo la seguridad de toda la infraestructura. La defensa contra ataques de fuerza bruta puede consumir recursos significativos. Los sistemas de monitoreo, detección y respuesta a incidentes pueden verse sobrecargados por la cantidad de intentos de inicio de sesión fallidos, dificultando la identificación de otras amenazas.

Para mitigar los ataques de fuerza bruta, es crucial implementar políticas de contraseñas fuertes. Estas deben incluir una combinación de letras mayúsculas y minúsculas, números y caracteres especiales, y una longitud mínima, lo que dificulta los ataques de fuerza bruta. Configurar políticas de bloqueo de cuenta después de un número determinado de intentos fallidos de inicio de sesión puede detener a los atacantes y reducir la efectividad

de los ataques de fuerza bruta. Requerir múltiples formas de autenticación, como contraseñas más códigos enviados a dispositivos móviles o autenticación biométrica, añade una capa adicional de seguridad y hace que los ataques de fuerza bruta sean mucho menos efectivos.

Implementar sistemas de monitoreo que detecten y alerten sobre intentos de inicio de sesión repetidos y fallidos puede ayudar a identificar y bloquear ataques de fuerza bruta antes de que tengan éxito. Limitar la velocidad de intentos de inicio de sesión desde una misma IP puede frenar los ataques automatizados y dar tiempo a los sistemas de seguridad para responder. Implementar CAPTCHAs en los formularios de inicio de sesión puede dificultar el uso de scripts automatizados por parte de los atacantes. Educar a los usuarios sobre la importancia de utilizar contraseñas fuertes y no reutilizar contraseñas en múltiples sitios puede reducir el riesgo de compromiso de cuentas. Mantener los sistemas y aplicaciones actualizados con los últimos parches de seguridad asegura que las vulnerabilidades conocidas que podrían ser explotadas en ataques de fuerza bruta sean corregidas.

Denegación de servicio (DoS/DDoS)

Un ataque de denegación de servicio (DoS) o su versión distribuida (DDoS) es una estrategia maliciosa que tiene como objetivo interrumpir el normal funcionamiento de un servidor, servicio o red, sobrecargándolo con una cantidad abrumadora de tráfico de datos. En un ataque DoS, el atacante utiliza un solo origen para enviar una cantidad masiva de solicitudes a un sistema objetivo, consumiendo sus recursos y provocando su incapacidad para responder a solicitudes legítimas. Por otro lado, un ataque DDoS amplifica este efecto utilizando múltiples dispositivos comprometidos, conocidos como botnets, que lanzan un ataque coordinado desde diversos puntos de la red, haciendo aún más difícil defenderse contra el ataque.

Los impactos de un ataque DoS/DDoS pueden ser devastadores. Uno de los efectos más inmediatos es la interrupción del servicio. Los usuarios legítimos no pueden acceder a los servicios afectados, lo que puede provocar pérdida de ingresos, especialmente para negocios que dependen de su presencia en línea, como el comercio electrónico. Esta interrupción también puede dañar la reputación de la empresa, ya que los clientes insatisfechos pueden percibir la organización como poco confiable. Además, la sobrecarga de tráfico puede causar un aumento en los costos operativos, ya que las empresas pueden necesitar gastar más en ancho de banda y recursos adicionales para manejar el ataque. Los ataques DDoS también pueden ser utilizados como una distracción, mientras el atacante realiza otras actividades maliciosas, como la extracción de datos o la instalación de malware.

La mitigación de ataques DoS/DDoS requiere un enfoque multifacético. La implementación de dispositivos de detección y prevención de intrusiones (IDS/IPS) puede ayudar a identificar y bloquear tráfico malicioso antes de que alcance la red interna. Servicios especializados de mitigación DDoS, ofrecidos por proveedores de seguridad, pueden absorber y filtrar el tráfico malicioso antes de que llegue al servidor objetivo. Además, el uso de balanceadores de carga distribuye el tráfico entrante entre varios servidores, evitando que un solo servidor sea sobrecargado. Configurar firewalls para filtrar el tráfico no deseado y establecer límites en la tasa de solicitudes puede reducir la efectividad de los ataques. Mantener la infraestructura actualizada con los últimos parches de seguridad y configurar políticas de respuesta rápida a incidentes puede minimizar el impacto y el tiempo de inactividad en caso de un ataque.

3.2 Sistemas de Alerta Temprana

Un sistema de alerta temprana en el contexto de la ciberseguridad es un conjunto de tecnologías y procesos diseñados para detectar, alertar y, en algunos casos, responder a amenazas cibernéticas en sus etapas iniciales. Estos sistemas monitorean de manera continua el tráfico de red, los registros de eventos y las actividades del sistema para identificar patrones anómalos, comportamientos sospechosos o firmas conocidas de ataques. La

finalidad principal de un sistema de alerta temprana es proporcionar información oportuna y relevante sobre posibles incidentes de seguridad para permitir una respuesta rápida y efectiva.

En un entorno cada vez más digitalizado y complejo, las organizaciones enfrentan un número creciente de ciberamenazas que pueden comprometer la seguridad de sus datos y sistemas. La detección temprana de estas amenazas es crucial para minimizar el impacto de los ataques y evitar daños significativos. Los sistemas de alerta temprana permiten a las organizaciones identificar amenazas potenciales antes de que se conviertan en incidentes graves, facilitando una respuesta proactiva y la implementación de medidas preventivas.

Por otro lado, entre los principales beneficios de los sistemas de alerta temprana se encuentran:

- **Detección Proactiva:** Los sistemas de alerta temprana proporcionan una detección temprana de actividades maliciosas, permitiendo a los equipos de seguridad abordar las amenazas antes de que causen daños significativos.
- **Respuesta Rápida:** Al alertar inmediatamente sobre comportamientos sospechosos, estos sistemas permiten una respuesta rápida, lo que reduce el tiempo de exposición y el impacto de los ataques.
- **Mejora de la Seguridad:** La implementación de sistemas de alerta temprana contribuye a una postura de seguridad más robusta, ya que las organizaciones pueden identificar y mitigar vulnerabilidades de manera continua.
- **Reducción de Costos:** Al prevenir incidentes de seguridad mayores, las organizaciones pueden reducir los costos asociados con las violaciones de datos, incluyendo la recuperación, las multas y el daño a la reputación.
- **Cumplimiento Normativo:** Muchos estándares y regulaciones de ciberseguridad requieren la implementación de sistemas de monitoreo y alerta, por lo que estos sistemas ayudan a las organizaciones a cumplir con sus obligaciones legales y normativas.

Por lo tanto, los sistemas de alerta temprana son una herramienta esencial en la ciberseguridad moderna, proporcionando las capacidades necesarias para detectar y responder a las amenazas cibernéticas de manera eficaz y eficiente.

3.3 Sistemas de detección de intrusiones (IDS)

Un Sistema de Detección de Intrusiones (IDS) es una herramienta de seguridad que monitorea y analiza el tráfico de red o las actividades del sistema en busca de comportamientos sospechosos o maliciosos que puedan indicar una violación de seguridad. Los IDS pueden clasificarse en dos tipos principales:

- **IDS basados en host (HIDS):** Estos IDS se instalan en dispositivos individuales, como servidores o estaciones de trabajo, y supervisan las actividades y eventos en ese host específico. Los HIDS pueden detectar intrusiones locales examinando registros de sistema, archivos de registro y cambios en el sistema de archivos.
- **IDS basados en red (NIDS):** Estos IDS se implementan en la red y monitorean el tráfico de red en busca de signos de intrusiones. Los NIDS analizan los paquetes de datos que pasan por la red y buscan patrones de tráfico sospechoso, ataques conocidos o anomalías en el comportamiento de la red.

La evolución de los IDS ha sido fundamental para la seguridad de la información en la era digital. Inicialmente, los IDS se desarrollaron para abordar la creciente sofisticación de los ataques cibernéticos y la necesidad de detectar intrusiones de manera proactiva. A lo largo del tiempo, los IDS han avanzado en términos de capacidades de detección, precisión y capacidad de respuesta, adaptándose a las cambiantes amenazas cibernéticas y al

entorno de seguridad en constante evolución. Hoy en día, los IDS son una parte integral de la infraestructura de seguridad moderna, trabajando en conjunto con otras herramientas de seguridad, como Firewalls y sistemas de prevención de intrusiones (IPS), para proteger activamente los sistemas y redes contra amenazas cibernéticas. Su capacidad para identificar y responder rápidamente a las intrusiones potenciales les otorga un papel crucial en la detección temprana y la mitigación de riesgos de seguridad.

3.4 Virtualización

La virtualización es una tecnología que permite la creación de entornos virtuales o recursos informáticos virtuales, como sistemas operativos, servidores, almacenamiento y redes, sobre una infraestructura física subyacente. Esto se logra mediante el uso de software especializado llamado hipervisor o monitor de máquina virtual, que crea y administra estas instancias virtuales.

La virtualización es fundamental en la infraestructura de TI moderna debido a sus beneficios en términos de eficiencia, flexibilidad y escalabilidad. Permite la consolidación de servidores, la optimización de recursos, la implementación rápida de nuevos servicios y la mejora del rendimiento y la disponibilidad de los sistemas.

La virtualización puede mejorar la seguridad cibernética al proporcionar entornos de prueba aislados y máquinas virtuales (VMs) para ejecutar aplicaciones y cargas de trabajo. Estos entornos virtuales permiten probar y validar parches de seguridad, configuraciones de red y políticas de seguridad sin afectar la infraestructura en producción.

Las organizaciones utilizan la virtualización para mejorar su postura de seguridad al implementar políticas de aislamiento, segmentación de red y recuperación ante desastres. Por ejemplo, pueden crear redes virtuales privadas (VPN) para el acceso remoto seguro, implementar firewalls virtuales para proteger las cargas de trabajo en la nube y utilizar sistemas de respaldo y recuperación basados en VMs para garantizar la continuidad del negocio en caso de incidentes de seguridad.

3.4.1 Máquinas virtuales

Las máquinas virtuales (VMs) son entornos de software que emulan hardware físico completo y ejecutan sistemas operativos completos de forma independiente en una misma infraestructura física subyacente. Cada VM opera como una entidad aislada y puede ejecutar aplicaciones y cargas de trabajo de manera independiente.

Las máquinas virtuales ofrecen un alto grado de aislamiento y seguridad, ya que cada una opera en su propio entorno virtualizado. Esto permite a las organizaciones consolidar múltiples sistemas en un solo servidor físico, optimizando el uso de recursos y reduciendo costos operativos. Las VMs son ampliamente utilizadas en entornos de producción y desarrollo, así como en la implementación de políticas de recuperación ante desastres y continuidad del negocio.

3.4.2 Contenedores

Los contenedores son entornos de ejecución ligeros y portátiles que encapsulan aplicaciones y sus dependencias, permitiendo su ejecución de manera aislada pero compartiendo el mismo sistema operativo subyacente. Cada Contenedor es una instancia independiente que contiene todo lo necesario para ejecutar una aplicación, incluidos los archivos de código, bibliotecas y configuraciones.

Los contenedores ofrecen un gran nivel de flexibilidad y eficiencia, ya que son más livianos que las máquinas virtuales y ofrecen un inicio más rápido. Son ideales para el desarrollo, la implementación rápida de aplicaciones y la creación de entornos de pruebas aislados. Los contenedores se utilizan ampliamente en entornos de desarrollo ágil, así como en la implementación de microservicios y arquitecturas basadas en contenedores.

Soluciones Existentes

En la gestión de infraestructuras de TI y ciberseguridad, es crucial contar con herramientas y plataformas que permitan la recolección, procesamiento, análisis y visualización de datos de Logs, así como la creación y gestión de entornos virtuales. En este capítulo, exploraremos las soluciones tecnológicas más relevantes y avanzadas disponibles en el mercado para cada una de estas necesidades. Revisaremos herramientas para la detección de intrusiones, recolección y envío de logs, plataformas de procesamiento de logs, herramientas de visualización y análisis de logs, y plataformas de virtualización. Presentaremos diversas opciones, tanto de código abierto como comerciales, destacando sus características y capacidades para ayudar a determinar cuál es la más adecuada según los requerimientos específicos.

Cada sección se centrará en una categoría de herramientas, proporcionando una descripción detallada de las soluciones más prominentes. Además, justificaremos la elección de la herramienta utilizada en este Trabajo de Fin de Grado (TFG), argumentando por qué es la más adecuada entre las alternativas presentadas.

4.1 Sistemas de Detección de Intrusiones (IDS)

En el mercado actual, la seguridad de las redes y sistemas informáticos es una preocupación primordial para organizaciones de todos los tamaños. Los Sistemas de Detección de Intrusiones (IDS) son herramientas esenciales que ayudan a identificar y responder a posibles amenazas cibernéticas. Estos sistemas analizan el tráfico de red y las actividades de los sistemas en busca de comportamientos sospechosos o maliciosos, alertando a los administradores de seguridad para que puedan tomar medidas preventivas o correctivas.

Existen diversas soluciones IDS disponibles, tanto de código abierto como comerciales, cada una con características y enfoques específicos. Los IDS de código abierto son particularmente atractivos para muchas organizaciones debido a su flexibilidad, costo reducido y el apoyo de una comunidad activa de desarrolladores. Entre estas soluciones, destacan algunas por su rendimiento, capacidad de análisis y facilidad de integración con otros sistemas de seguridad.

De todas las soluciones presentadas en el Cuadro 4.1, Suricata se destaca como el IDS más adecuado para utilizar en este Trabajo de Fin de Grado (TFG) debido a su alto rendimiento, capacidad de análisis en tiempo real y arquitectura multithreading. Estas características permiten una detección eficaz y rápida de amenazas, lo que es crucial para mantener la seguridad de la red. Además, su flexibilidad en la definición de reglas y su integración con otras herramientas de seguridad hacen de Suricata una opción robusta y versátil para enfrentar los desafíos actuales de ciberseguridad.

Nombre	Descripción
Suricata	Suricata [62] es un sistema de detección y prevención de intrusiones de código abierto de alto rendimiento, capaz de realizar análisis en tiempo real del tráfico de red en busca de amenazas y comportamientos maliciosos. Utiliza reglas de detección flexibles y una arquitectura multithreading para ofrecer una protección eficaz contra una amplia gama de ataques cibernéticos.
Snort	Snort [14] es uno de los IDS/IPS de código abierto más populares y establecidos en el mercado. Utiliza reglas predefinidas para detectar y prevenir intrusiones en la red, ofreciendo flexibilidad y una amplia comunidad de usuarios y desarrolladores.
Snort++	Snort++ [63] es una versión moderna y mejorada de Snort que ofrece un rendimiento mejorado y nuevas características de detección de amenazas. Está diseñado para ser más rápido y escalable que su predecesor, manteniendo la flexibilidad y la efectividad en la detección de intrusiones.
Zeek	Zeek [53] es una plataforma de análisis de red de código abierto que se centra en la generación de registros detallados para el análisis forense de red. Además, ofrece una visión profunda del tráfico de red y puede integrarse con otros sistemas de seguridad.
Security Onion	Security Onion [60] es una distribución Linux diseñada para la monitorización de seguridad de la red, que integra herramientas como Suricata y Snort para la detección de amenazas cibernéticas. Ofrece capacidades de captura, análisis y visualización de datos de seguridad en un solo paquete, facilitando la gestión y respuesta a incidentes.

Cuadro 4.1: Soluciones existentes para IDS

4.2 Recolección y envío de logs

En el ámbito de la gestión de logs, es fundamental contar con herramientas eficientes para recopilar, enviar y procesar registros de diferentes fuentes de manera confiable y escalable. La recopilación y análisis de logs son componentes esenciales en la detección temprana de problemas de seguridad, el monitoreo del rendimiento del sistema y la resolución de problemas en entornos informáticos complejos.

Existen varias soluciones disponibles en el mercado para la recopilación y envío de logs, cada una con características y capacidades específicas. Desde herramientas de código abierto hasta soluciones comerciales, estas plataformas ofrecen diferentes enfoques para la gestión de logs, adaptándose a las necesidades y requisitos de cada organización.

Entre las soluciones presentadas en el Cuadro 4.2, Filebeat destaca como la herramienta más idónea para utilizar en este contexto, debido a su facilidad de configuración, alta escalabilidad y bajo consumo de recursos. Al formar parte del stack ELK, Filebeat se integra perfectamente con Elasticsearch, lo que facilita la indexación y análisis de logs en tiempo real. Su capacidad para enviar logs de manera eficiente y confiable lo convierte en una opción sólida para la recopilación y envío de logs en entornos distribuidos y de alta disponibilidad.

Nombre	Descripción
Filebeat	Filebeat [18] es un ligero agente de envío de logs de código abierto desarrollado por Elastic, diseñado para recopilar, enviar y procesar logs de diferentes fuentes de manera eficiente y confiable. Como parte del stack ELK (Elasticsearch, Logstash, Kibana), Filebeat se encarga de enviar los logs a Elasticsearch o Logstash para su indexación y análisis. Destaca por su facilidad de configuración, altamente escalable, bajo consumo de recursos y capacidad para enviar logs en tiempo real.
Apache Kafka	Apache Kafka [3] es una plataforma de transmisión de datos de código abierto y distribuida, diseñada para manejar flujos de datos en tiempo real a gran escala. Funciona como un sistema de mensajería distribuida que permite a los productores enviar mensajes a un conjunto de consumidores de manera eficiente y confiable.
Rsyslog	Rsyslog [30] es una herramienta de log que proporciona una amplia variedad de opciones para el procesamiento de logs y la entrega a diferentes destinos. Es ligero, altamente configurable y ampliamente utilizado en sistemas Linux para la gestión de logs en tiempo real.
Fluentd	Fluentd [51] es un agente de envío de logs de código abierto que permite la recopilación de logs de diversas fuentes y su envío a múltiples destinos. Es altamente escalable y cuenta con una arquitectura flexible que facilita su integración con una amplia variedad de sistemas y servicios.
Apache Flume	Apache Flume [2] es un sistema de recopilación y agregación de logs diseñado para mover grandes volúmenes de datos de logs de manera eficiente y confiable. Utiliza una arquitectura distribuida y escalable que permite la recopilación de logs desde múltiples fuentes y su entrega a sistemas de almacenamiento centralizados.

Cuadro 4.2: Soluciones existentes para recolección y envío de logs

4.3 Procesamiento de logs

En el ámbito del procesamiento de logs, es fundamental contar con herramientas potentes y flexibles que permitan indexar, analizar y visualizar grandes volúmenes de datos de manera eficiente y confiable. Estas plataformas son esenciales para la monitorización de sistemas, la detección de anomalías y la generación de informes sobre el rendimiento y la seguridad de los sistemas informáticos.

Existen diversas soluciones disponibles en el mercado para el procesamiento de logs, cada una con características y funcionalidades específicas. Desde motores de búsqueda y análisis distribuidos hasta servicios de gestión de logs en la nube, estas plataformas ofrecen diferentes enfoques para la gestión y análisis de logs, adaptándose a las necesidades y requisitos de cada organización.

De las soluciones presentadas en el Cuadro 4.3, Elasticsearch destaca como la plataforma más idónea para el procesamiento de logs en este contexto. Su potente motor de búsqueda y análisis distribuido, combinado con su facilidad de uso y escalabilidad, lo convierte en una opción sólida para indexar, buscar y analizar logs en tiempo real. Además, la compatibilidad con la API de Elasticsearch facilita la integración con otras herramientas y servicios, permitiendo construir soluciones completas y personalizadas para la gestión y análisis de logs en entornos distribuidos y de alta disponibilidad.

Nombre	Descripción
Elasticsearch	Elasticsearch [21] es un potente motor de búsqueda y análisis distribuido de código abierto, diseñado para indexar, buscar y analizar grandes volúmenes de datos en tiempo real. Es altamente escalable, confiable y fácil de usar, y se utiliza en una amplia variedad de aplicaciones, desde la búsqueda en sitios web hasta la monitorización de logs y la analítica de datos. Elasticsearch ofrece una API flexible y robusta que permite realizar búsquedas complejas y realizar análisis avanzados sobre los datos indexados.
Apache Solr	Apache Solr [4] es un motor de búsqueda de código abierto altamente escalable y de alto rendimiento, basado en Apache Lucene. Ofrece capacidades avanzadas de indexación y búsqueda, así como funcionalidades de facetas, resaltado y geoespaciales. Solr es especialmente adecuado para aplicaciones de búsqueda en la web y análisis de texto.
Opensearch	OpenSearch [16] es un motor de búsqueda y análisis de código abierto, creado por AWS como una bifurcación de Elasticsearch y Kibana. Ofrece capacidades avanzadas de búsqueda y análisis de datos a gran escala, manteniendo compatibilidad con la API de Elasticsearch. Es altamente escalable y robusto, ideal para aplicaciones que requieren indexación y búsqueda de grandes volúmenes de datos en tiempo real.
Splunk	Splunk [15] es una plataforma líder en el mercado para la búsqueda, análisis y visualización de datos, incluidos los logs. Ofrece una amplia gama de funcionalidades, como búsqueda en tiempo real, correlación de eventos, generación de informes y alertas, y es altamente escalable y personalizable.
Sumo Logic	Sumo Logic [61] es un servicio de gestión de logs en la nube que ofrece análisis de logs en tiempo real, detección de amenazas y visualización de datos. Proporciona una plataforma unificada para la recopilación, análisis y visualización de logs en entornos distribuidos y multinube.

Cuadro 4.3: Soluciones existentes para procesamiento de logs

4.4 Visualización y análisis de logs

En la gestión de sistemas informáticos, la visualización y análisis de logs son aspectos críticos para comprender el estado y el rendimiento de los sistemas, así como para detectar posibles problemas o anomalías. Las plataformas de visualización y análisis de logs proporcionan herramientas poderosas para explorar, analizar y visualizar datos de logs de manera eficiente y efectiva.

En el mercado actual, existen diversas soluciones disponibles para la visualización y análisis de logs, cada una con características y funcionalidades específicas. Desde plataformas de código abierto hasta servicios en la nube, estas herramientas ofrecen diferentes enfoques para la gestión y análisis de logs, adaptándose a las necesidades y requisitos de cada organización.

De todas las soluciones presentadas en el Cuadro 4.4, Kibana destaca como la plataforma más adecuada para la visualización y análisis de logs en este contexto. Su integración con Elasticsearch y su interfaz intuitiva lo convierten en una herramienta poderosa para explorar, analizar y visualizar datos de logs en tiempo real. Además, la flexibilidad y las capacidades de personalización de Kibana permiten adaptar la visualización de datos a las necesidades específicas de cada organización, facilitando la detección de problemas y la toma de decisiones informadas.

Nombre	Descripción
Kibana	Kibana [22] es una plataforma de visualización de datos de código abierto, diseñada para trabajar en conjunto con Elasticsearch y facilitar la exploración, análisis y visualización de datos almacenados en este motor de búsqueda. Con una interfaz intuitiva y fácil de usar, Kibana permite crear paneles interactivos, gráficos y tablas dinámicas para representar datos en tiempo real.
Grafana	Grafana [31] es una plataforma de visualización de datos que se centra principalmente en la visualización de series temporales, como datos de métricas y registros de tiempo. Ofrece una amplia variedad de paneles y opciones de personalización, así como integraciones con diferentes fuentes de datos, incluido Elasticsearch.
Graylog	Graylog [32] es una plataforma de gestión de logs de código abierto que incluye capacidades avanzadas de búsqueda, análisis y visualización de logs en tiempo real. Proporciona una interfaz intuitiva y flexible para explorar y analizar datos de logs, así como la generación de alertas personalizadas.
Fluentd-ui	Fluentd-ui [52] es una interfaz web para la gestión y visualización de logs en entornos que utilizan Fluentd para la recopilación y envío de logs. Permite la configuración y supervisión centralizadas de los flujos de logs, así como la generación de informes y alertas para eventos importantes.
Logz.io	Logz.io [39] es un servicio de análisis de logs en la nube que ofrece capacidades avanzadas de búsqueda, análisis y visualización de logs con una interfaz intuitiva y fácil de usar. Proporciona una variedad de herramientas y paneles preconfigurados para el análisis de logs en entornos distribuidos.

Cuadro 4.4: Soluciones existentes para visualización y análisis de logs

4.5 Notificación de alertas por correo

En el ámbito de la gestión de logs y la monitorización de eventos, la capacidad de detectar anomalías y generar alertas en tiempo real es crucial para mantener la seguridad y el rendimiento de los sistemas. Las herramientas de notificación de alertas permiten a las organizaciones monitorear sus datos y recibir notificaciones automáticas cuando se detectan eventos o condiciones específicas. Esto facilita una respuesta rápida y eficaz ante incidentes potenciales, mejorando la capacidad de reacción y mitigación de riesgos.

Existen diversas soluciones en el mercado para la notificación de alertas, cada una con características y funcionalidades particulares que se adaptan a diferentes necesidades y entornos. Estas herramientas van desde soluciones de código abierto hasta plataformas comerciales con capacidades avanzadas de personalización y gestión de alertas.

De todas las soluciones presentadas en el Cuadro 4.5, Elastalert2 se destaca como la herramienta más adecuada para utilizar en este Trabajo de Fin de Grado (TFG) debido a su integración nativa con Elasticsearch y su capacidad para configurar reglas de alerta personalizadas. La flexibilidad de Elastalert2 para definir condiciones específicas y enviar notificaciones automáticas permite una monitorización eficaz y en tiempo real de los datos indexados en Elasticsearch. Además, como herramienta de código abierto, ofrece una opción económica y adaptable para implementar un sistema robusto de notificación de alertas, ajustándose perfectamente a los requisitos del TFG.

Nombre	Descripción
Elastalert2	Elastalert2 [23] es una herramienta de código abierto diseñada para la detección de anomalías y la generación de alertas en tiempo real basadas en datos indexados en Elasticsearch. Es una evolución de Elastalert, con mejoras en rendimiento y funcionalidades. Elastalert2 permite configurar reglas flexibles y personalizadas para monitorear eventos y condiciones específicas en los datos de Elasticsearch, y enviar notificaciones automáticas, como correos electrónicos o mensajes, cuando se detectan eventos de interés.
Watcher	Watcher [20] es una característica integrada de pago en la pila ELK que permite configurar alertas basadas en condiciones específicas en los datos indexados en Elasticsearch. Permite la configuración de alertas personalizadas y la ejecución de acciones automatizadas en respuesta a eventos detectados.
Prometheus Alertmanager	Prometheus Alertmanager [5] es parte del ecosistema de Prometheus y permite configurar y gestionar alertas basadas en métricas y eventos de manera altamente personalizable. Proporciona capacidades avanzadas de deduplicación, silenciamiento y enrutamiento de alertas para una gestión eficiente.
Graylog Alerting	Graylog Alerting [33] es una funcionalidad dentro de la plataforma Graylog que permite configurar alertas basadas en reglas personalizadas sobre los datos de logs. Permite la generación de alertas en tiempo real y la integración con sistemas de notificación externos para la gestión eficiente de incidentes.
Sensu	Sensu [57] es una plataforma de monitorización de infraestructuras y aplicaciones que incluye funcionalidades avanzadas de alerta y notificación. Permite la configuración de alertas basadas en métricas y eventos, así como la integración con sistemas de notificación externos para la gestión eficiente de incidentes.

Cuadro 4.5: Soluciones existentes para notificación de alertas por correo

4.6 Plataformas de virtualización

La virtualización se ha convertido en una tecnología fundamental en la infraestructura de TI, permitiendo la creación y gestión de entornos virtuales eficientes y escalables. Las plataformas de virtualización ofrecen herramientas para crear y administrar máquinas virtuales, facilitando la consolidación de servidores, la implementación de aplicaciones y la gestión de recursos de manera flexible.

En el mercado actual, existen diversas soluciones disponibles para la virtualización, cada una con características y funcionalidades específicas. Desde hipervisores de código abierto hasta plataformas de gestión de virtualización, estas herramientas ofrecen diferentes enfoques para la creación y administración de entornos virtuales, adaptándose a las necesidades y requisitos de cada organización.

De las soluciones presentadas en el Cuadro 4.6, Proxmox VE destaca como la plataforma más idónea para la virtualización en este contexto. Al estar basado en KVM y contenedores LXC, Proxmox VE ofrece un enfoque integral para la virtualización que combina la eficiencia y escalabilidad de KVM con la flexibilidad y ligereza de los contenedores LXC. Además, su interfaz de gestión centralizada y sus capacidades de alta disponibilidad lo convierten en una opción robusta y versátil para implementaciones en entornos empresariales y de proveedores de servicios.

Nombre	Descripción
Proxmox VE	Proxmox Virtual Environment (Proxmox VE) [54] es una plataforma de virtualización de código abierto basada en KVM y contenedores LXC. Proporciona una solución completa para la virtualización, incluidos hipervisores, herramientas de gestión centralizadas y capacidades de alta disponibilidad. Es altamente flexible y adecuado para entornos empresariales y de proveedores de servicios.
VMware vSphere	VMware vSphere [9] es una plataforma de virtualización líder en el mercado que proporciona un conjunto completo de funciones para la creación y gestión de entornos virtuales. Incluye un hipervisor de alto rendimiento, herramientas de gestión centralizadas y capacidades avanzadas de automatización y seguridad. Es altamente escalable y adecuado para entornos empresariales de cualquier tamaño.
Microsoft Hyper-V	Microsoft Hyper-V [41] es una solución de virtualización incluida en el sistema operativo Windows Server. Ofrece capacidades básicas de virtualización, incluido un hipervisor de tipo 1 y herramientas de gestión integradas en el entorno de Windows Server. Es una opción popular para organizaciones que utilizan la plataforma Windows y buscan una solución de virtualización integrada y fácil de usar.
Oracle VM Virtualbox	Oracle VM VirtualBox [47] un Hipervisor de virtualización de código abierto que permite a los usuarios crear y ejecutar múltiples sistemas operativos en una misma máquina física. Es fácil de usar, compatible con diversos sistemas operativos y ofrece funciones como instantáneas y configuraciones de red flexibles. Es popular para desarrollo, pruebas y educación debido a su versatilidad y licencia gratuita.
KVM	KVM (Kernel-based Virtual Machine) [8] es un hipervisor de código abierto que se ejecuta en sistemas Linux y aprovecha las funcionalidades de virtualización integradas en el kernel de Linux. Es altamente escalable y eficiente en términos de recursos, lo que lo hace adecuado para implementaciones en servidores físicos y en la nube. KVM es una opción popular para la virtualización en entornos basados en Linux.

Cuadro 4.6: Soluciones existentes para plataformas de Virtualización

Herramientas de desarrollo



Figura 5.1: Logos de las herramientas de desarrollo empleadas en este proyecto

5.1 Suricata

Suricata [62] es un sistema de prevención de intrusiones de red (NIDS) de código abierto que proporciona detección de amenazas en tiempo real y capacidades de respuesta para proteger las redes contra ataques cibernéticos. Con una amplia comunidad de usuarios y desarrolladores en todo el mundo, Suricata es en una herramienta muy a tener en cuenta para garantizar la seguridad de las infraestructuras de red.

Además de su capacidad para detectar y prevenir intrusiones, Suricata ofrece una serie de características y funcionalidades que lo hacen altamente efectivo en la detección y respuesta a amenazas. Estas incluyen la

capacidad de inspeccionar el tráfico de red en tiempo real, la detección de firmas y anomalías, y la integración con otros sistemas de seguridad y herramientas de análisis de registro.

Ventajas:

- Suricata ofrece una detección de amenazas altamente precisa y en tiempo real, lo que permite a las organizaciones identificar y responder rápidamente a posibles ataques cibernéticos.
- Su arquitectura modular y su amplia gama de complementos permiten una fácil integración con otros sistemas de seguridad y herramientas de análisis de registro.

Desventajas:

- Configurar y mantener Suricata puede requerir un conocimiento técnico especializado, lo que puede suponer un desafío para algunos usuarios.
- El monitoreo y la gestión efectivos del tráfico de red pueden requerir una inversión significativa en recursos de hardware y ancho de banda.

En conclusión, opté por utilizar Suricata debido a su capacidad comprobada para detectar y prevenir intrusiones en tiempo real, respaldada por una amplia comunidad de usuarios. Aunque su configuración técnica puede ser exigente, su eficacia en la detección de amenazas y su capacidad de integración con otras herramientas de seguridad lo convierten en una elección sólida para garantizar la seguridad de las infraestructuras de red.

5.2 Docker

Docker [17] es una plataforma de código abierto que permite a los desarrolladores crear, implementar y ejecutar aplicaciones de manera fácil y eficiente utilizando contenedores. Con una altísima cantidad de usuarios, se ha convertido en una herramienta esencial en el desarrollo de software moderno. Docker simplifica el proceso de creación y gestión de entornos de desarrollo y producción, lo que permite a los equipos de desarrollo trabajar de manera más rápida y colaborativa.

De igual manera, Docker proporciona una serie de características y funcionalidades que hacen que el desarrollo y la implementación de aplicaciones sean más ágiles y flexibles. Esto incluye la capacidad de encapsular aplicaciones y sus dependencias en contenedores independientes, lo que garantiza la portabilidad y consistencia del entorno de ejecución. Docker también facilita la automatización del ciclo de vida de las aplicaciones mediante herramientas como Docker Compose y Docker Swarm, lo que permite a los equipos gestionar eficientemente aplicaciones distribuidas y escalables.

Ventajas:

- Docker ofrece una forma consistente y reproducible de empaquetar y distribuir aplicaciones, lo que facilita la implementación en cualquier entorno.
- Su enfoque en la virtualización a nivel de contenedor proporciona un aislamiento ligero y eficiente, lo que permite a los desarrolladores ejecutar múltiples aplicaciones en el mismo host sin comprometer el rendimiento.

Desventajas:

- Aunque Docker simplifica el proceso de desarrollo y implementación, puede requerir una curva de aprendizaje significativa para los usuarios nuevos, especialmente aquellos que no están familiarizados con los conceptos de contenedores y virtualización.

- El uso excesivo de contenedores puede aumentar la complejidad del entorno y requerir una gestión cuidadosa de los recursos para evitar la degradación del rendimiento.

En conclusión, decido utilizar Docker en mi proyecto debido a su capacidad para simplificar el desarrollo y la implementación de aplicaciones mediante contenedores. Su amplia adopción y características, como la portabilidad y la automatización, prometen agilizar mi trabajo. Aunque puede requerir una curva de aprendizaje inicial y una gestión cuidadosa de recursos, a mi parecer sus ventajas superan con creces estas dificultades potenciales.

5.3 Elasticsearch

Elasticsearch [21] es una poderosa plataforma de búsqueda y análisis distribuido diseñada para manejar grandes volúmenes de datos de manera eficiente. Utilizado por empresas de todo el mundo, Elasticsearch se ha convertido en una herramienta muy a tener en cuenta para la búsqueda de información, análisis de datos y observabilidad de sistemas.

Además de su capacidad para indexar y buscar grandes cantidades de datos, Elasticsearch tiene una serie de características y funcionalidades que lo hacen altamente efectivo en una variedad de casos de uso. Estas incluyen capacidades de búsqueda de texto completo, análisis de datos en tiempo real, y capacidades de agregación y visualización de datos.

Ventajas:

- Elasticsearch proporciona una búsqueda y análisis de datos rápidos y escalables, lo que permite a las organizaciones obtener información valiosa de sus datos de manera eficiente.
- Su arquitectura distribuida y su escalabilidad horizontal facilitan la gestión de grandes volúmenes de datos y garantizan un alto rendimiento incluso en entornos de alta carga.

Desventajas:

- Configurar y mantener un clúster de Elasticsearch puede requerir un conocimiento técnico especializado, lo que puede suponer un desafío para algunos usuarios.
- El almacenamiento y la gestión de grandes volúmenes de datos pueden requerir una inversión significativa en recursos de hardware y ancho de banda.

Decido incorporar Elasticsearch en mi proyecto debido a su eficacia en la búsqueda y análisis de grandes volúmenes de datos. Su capacidad para proporcionar información valiosa de manera eficiente lo convierte en una herramienta esencial para la búsqueda de información, análisis de datos y observabilidad de sistemas. Aunque reconozco que su configuración y mantenimiento pueden requerir conocimientos técnicos especializados, así como una inversión en recursos, las ventajas en términos de rapidez, escalabilidad y capacidad de análisis en tiempo real justifican su integración en mi proyecto.

5.4 Kibana

Kibana [22] es una plataforma de visualización de datos y análisis diseñada para trabajar en conjunto con Elasticsearch. Utilizado por corporaciones de todo el mundo, Kibana es una herramienta perfectamente diseñada para explorar, visualizar y compartir datos de manera efectiva.

Por otro lado, Kibana además de su capacidad para crear visualizaciones personalizadas y paneles de control interactivos, Kibana ofrece una amplia gama de características y funcionalidades que lo hacen altamente versátil

en diversos entornos. Estas incluyen la capacidad de crear gráficos, mapas, tablas y otros tipos de visualizaciones, así como herramientas de análisis de datos en tiempo real y capacidades de colaboración.

Ventajas:

- Kibana proporciona una interfaz intuitiva y fácil de usar para explorar y visualizar datos, lo que permite a los usuarios obtener información valiosa de manera rápida y eficiente.
- Su integración con Elasticsearch permite a los usuarios realizar análisis en tiempo real y obtener información procesable de sus datos.

Desventajas:

- Configurar y personalizar Kibana puede requerir un conocimiento técnico especializado, lo que puede suponer un desafío para algunos usuarios.
- La gestión de grandes volúmenes de datos puede afectar al rendimiento de Kibana, especialmente en entornos de alta carga.

En conclusión, decido integrar Kibana en mi proyecto debido a su capacidad para explorar y visualizar datos de manera efectiva, trabajando en conjunto con Elasticsearch. Su interfaz intuitiva y amplia gama de características lo convierten en una herramienta invaluable para empresas de todo el mundo. Aunque la configuración y personalización pueden requerir conocimientos técnicos especializados, y que la gestión de grandes volúmenes de datos puede afectar su rendimiento, las ventajas en términos de obtención rápida de información y análisis en tiempo real respaldan su elección en mi proyecto.

5.5 Elastalert2

Elastalert2 [23] es una poderosa herramienta de detección de amenazas y generación de alertas diseñada para integrarse con Elasticsearch. Con su capacidad para analizar datos en tiempo real y generar alertas basadas en reglas personalizadas, Elastalert2 se ha convertido en una herramienta esencial para la monitorización y seguridad de sistemas.

Además de su capacidad para detectar patrones y anomalías en los datos indexados en Elasticsearch, Elastalert 2 ofrece una amplia gama de funcionalidades que lo hacen altamente configurable y adaptable a diversas necesidades de detección de amenazas y generación de alertas.

Ventajas:

- Elastalert 2 es altamente configurable y permite a los usuarios definir reglas de detección de amenazas personalizadas para adaptarse a las necesidades específicas de su entorno.
- Su capacidad para generar alertas en tiempo real basadas en reglas predefinidas o personalizadas permite a los equipos de seguridad responder rápidamente a posibles incidentes.

Desventajas:

- Configurar y mantener Elastalert 2 puede requerir conocimientos técnicos especializados en Elasticsearch y lenguajes de consulta como JSON y YAML.
- En entornos de alto tráfico o con grandes volúmenes de datos, la gestión de reglas y alertas puede volverse compleja y difícil de mantener.

En conclusión, elijo incorporar Elastalert 2 en mi proyecto debido a su capacidad para detectar amenazas y generar alertas en tiempo real, lo que lo convierte en una herramienta esencial para la seguridad y monitorización de sistemas, además de su fácil integración con Elasticsearch. A pesar de que su configuración y mantenimiento pueden requerir conocimientos técnicos más concretos, su capacidad para generar una respuesta rápida a posibles incidentes justifica su incorporación en mi proyecto.

5.6 Filebeat

Filebeat [18] es un ligero y eficiente envío de registros y herramienta de recopilación de datos desarrollada por Elastic. Diseñado para trabajar en conjunto con Elasticsearch y Kibana, Filebeat facilita la recopilación, envío y análisis de registros de diferentes fuentes en tiempo real.

Asimismo, Filebeat ofrece unas excelentes capacidades para recopilar registros de una variedad de fuentes, incluidos archivos de registro, eventos del sistema y datos de aplicaciones, Filebeat proporciona una serie de características y funcionalidades que lo hacen altamente versátil en entornos de monitoreo y análisis de registros.

Ventajas:

- Filebeat es fácil de configurar y usar, lo que permite a los usuarios comenzar a recopilar registros rápidamente sin necesidad de una curva de aprendizaje prolongada.
- Su arquitectura ligera y eficiente garantiza un bajo consumo de recursos y un rendimiento óptimo, incluso en entornos de alto tráfico.

Desventajas:

- Configurar Filebeat para manejar ciertos tipos de registros o integrarlo con sistemas específicos puede requerir conocimientos técnicos especializados.
- En entornos muy grandes o complejos, la gestión de la configuración de Filebeat y la escalabilidad pueden ser un desafío.

En conclusión, decido incorporar Filebeat en mi proyecto debido a su capacidad para recopilar y enviar registros de manera eficiente, facilitando el análisis de datos en tiempo real junto con Elasticsearch y Kibana. Su facilidad de configuración y uso permite una rápida implementación, mientras que su arquitectura ligera garantiza un rendimiento óptimo incluso en entornos de alto tráfico. Aunque configurar Filebeat para manejar ciertos tipos de registros o integrarlo con sistemas específicos puede requerir una curva de aprendizaje mayor, y que la gestión en entornos muy grandes puede presentar desafíos, considero que sus beneficios en términos de agilidad y eficiencia justifican su inclusión en mi proyecto.

5.7 Obsidian

Obsidian [44] es una aplicación de gestión de conocimiento basada en Markdown que se ha convertido en una herramienta esencial para estudiantes, investigadores y profesionales del conocimiento. Con una interfaz de usuario intuitiva y altamente personalizable, permite a los usuarios capturar, organizar y conectar sus ideas de manera efectiva. Obsidian utiliza archivos de texto plano en formato Markdown, lo que facilita la creación y edición de contenido sin depender de plataformas propietarias.

Por su parte, Obsidian ofrece una serie de características y funcionalidades que mejoran la experiencia del usuario y fomentan la creación de un repositorio de conocimiento coherente. Incluye funciones como enlaces bidireccionales, grafos de conocimiento, búsqueda avanzada y soporte para complementos de terceros. Esto

permite a los usuarios gestionar grandes cantidades de información de manera eficiente y encontrar conexiones significativas entre conceptos.

Ventajas:

- Obsidian ofrece una interfaz de usuario simple y fácil de usar que permite a los usuarios concentrarse en su contenido sin distracciones.
- Su enfoque en archivos de texto plano en formato Markdown garantiza la portabilidad y la interoperabilidad de los datos, lo que permite a los usuarios acceder a su conocimiento desde una variedad de dispositivos y plataformas.
- Cuenta con una gran cantidad de extensiones comunitarias totalmente gratuitas y fácilmente instalables.

Desventajas:

- Aunque es altamente personalizable, la curva de aprendizaje inicial puede ser alta para algunos usuarios, especialmente aquellos que no están familiarizados con el formato Markdown o los conceptos de gestión de conocimiento.
- Aunque es una herramienta poderosa, Obsidian puede carecer de algunas características avanzadas que se encuentran en otras aplicaciones de gestión de conocimiento, como la colaboración en tiempo real o la integración con otras herramientas de productividad.

En conclusión, elijo usar Obsidian en mi flujo de trabajo por su interfaz intuitiva y altamente personalizable, que me permite centrarme en el contenido sin distracciones. Su enfoque en archivos de texto plano en formato Markdown asegura la portabilidad de datos, y la amplia gama de extensiones gratuitas disponibles facilita la personalización. Aunque su curva de aprendizaje inicial puede ser alta y carece de algunas características avanzadas, como la colaboración en tiempo real, considero que sus ventajas en simplicidad y portabilidad justifican su elección.

5.8 Visual Studio Code

Visual Studio Code [40] es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta fundamental para desarrolladores de software en todo el mundo. Con una amplia gama de extensiones y funcionalidades, ofrece a los usuarios una experiencia de codificación eficiente y altamente personalizable. Permite a los desarrolladores escribir, depurar y editar código en una variedad de lenguajes de programación, y cuenta con herramientas integradas para el control de versiones y la colaboración en proyectos.

Por otro lado, Visual Studio Code ofrece una serie de características y herramientas que facilitan la vida de los desarrolladores y mejoran su productividad. Incluye características como la resaltado de sintaxis, el autocompletado de código, la depuración integrada y el control de versiones integrado con Git. Además, su amplia gama de extensiones permite a los usuarios personalizar su entorno de desarrollo según sus necesidades específicas y trabajar de manera más eficiente.

Ventajas:

- Visual Studio Code ofrece una interfaz de usuario intuitiva y altamente personalizable que permite a los desarrolladores adaptar su entorno de desarrollo según sus preferencias.
- Su amplia gama de extensiones y complementos permite a los desarrolladores ampliar la funcionalidad del editor y adaptarlo a sus necesidades específicas.

- Puede ser empleado para trabajar con muchísimos lenguajes de programación distintos, gracias a su gran cantidad de extensiones.

Desventajas:

- Aunque es altamente personalizable, la amplia gama de extensiones disponibles puede ser abrumadora para algunos usuarios, especialmente aquellos que son nuevos en el desarrollo de software.
- Aunque es un editor potente, Visual Studio Code puede consumir muchos recursos del sistema, especialmente en proyectos grandes o con muchas extensiones instaladas.
- Quizás, algún editor de código especializado en un único lenguaje puede ser más cómodo y eficiente en algunos casos.

En conclusión, utilizo Visual Studio Code como mi editor principal debido a su interfaz intuitiva y altamente personalizable, que me permite adaptar mi entorno de desarrollo a mis preferencias. Su amplia selección de extensiones y complementos me permite ampliar la funcionalidad del editor según mis necesidades específicas, y su capacidad para trabajar con una variedad de lenguajes de programación lo convierte en una opción versátil. Aunque reconozco que la cantidad de extensiones disponibles puede resultar abrumadora para algunos usuarios y que puede consumir muchos recursos del sistema en proyectos grandes, valoro que sus ventajas en términos de personalización y versatilidad justifican su uso como mi editor principal de código.

5.9 Microsoft Teams

Microsoft Teams [42] es una plataforma de colaboración en línea ampliamente utilizada para la comunicación y el trabajo en equipo en entornos empresariales. Con millones de usuarios en todo el mundo, es una herramienta de vital importancia para las empresas y organizaciones que buscan mejorar la productividad y la colaboración entre sus empleados. Permite a los usuarios comunicarse a través de mensajes instantáneos, realizar llamadas de voz y video, compartir archivos y colaborar en proyectos en tiempo real.

Por otra parte, Microsoft Teams ofrece una amplia gama de características y herramientas que facilitan la colaboración efectiva entre equipos de trabajo. Además de la mensajería instantánea y las llamadas de voz y video, incluye funciones como la integración con otras aplicaciones de Microsoft Office, la creación y gestión de equipos y canales, y la posibilidad de programar reuniones y eventos. Esto permite a los usuarios trabajar de manera más eficiente y coordinada, independientemente de su ubicación geográfica o huso horario.

Ventajas:

- Microsoft Teams ofrece una experiencia de usuario intuitiva y familiar para aquellos que ya están familiarizados con otras aplicaciones de Microsoft Office.
- Su integración con otras herramientas de productividad de Microsoft, como SharePoint y OneDrive, permite a los usuarios acceder fácilmente a archivos y documentos relevantes desde un solo lugar.

Desventajas:

- Aunque es una plataforma poderosa, Microsoft Teams puede ser complejo de configurar y administrar, especialmente para organizaciones más pequeñas o menos técnicas.
- La dependencia de la conexión a Internet puede ser una limitación para aquellos que trabajan en entornos con conectividad limitada o inestable.

En conclusión, decido utilizar Microsoft Teams como plataforma principal de colaboración en mi entorno de trabajo debido a su amplia gama de características que facilitan la comunicación y el trabajo en equipo. Su familiaridad para aquellos que ya utilizan otras aplicaciones de Microsoft Office proporciona una experiencia intuitiva para los usuarios. Además, su integración con herramientas de productividad como SharePoint y OneDrive permite un acceso fácil a archivos relevantes. Sin embargo, la configuración y administración pueden ser complejas, además que la dependencia de la conexión a Internet puede ser una limitación en algunos casos, considero que las ventajas en términos de colaboración eficaz justifican su uso en mi entorno de trabajo para este proyecto.

5.10 Overleaf

Overleaf [49] es una plataforma en línea muy utilizada para la edición colaborativa de documentos LaTeX. Con millones de usuarios en todo el mundo, se ha convertido en una herramienta esencial para los científicos, investigadores y estudiantes que trabajan en documentos técnicos y académicos. Permite a los usuarios escribir, editar y compartir documentos LaTeX de manera eficiente, eliminando la necesidad de instalar y mantener software LaTeX en sus propios sistemas.

Asimismo, Overleaf ofrece un robusto editor en línea que facilita la colaboración y la edición en tiempo real de documentos LaTeX. Además, cuenta con características como el control de versiones integrado, plantillas predefinidas y una amplia gama de paquetes y bibliotecas LaTeX, lo que permite a los usuarios crear documentos complejos con facilidad y precisión.

Ventajas:

- Overleaf ofrece un entorno de edición colaborativa que permite a múltiples usuarios trabajar en un documento LaTeX simultáneamente.
- Su sistema de control de versiones integrado permite a los usuarios rastrear y gestionar cambios en sus documentos, facilitando la colaboración efectiva en proyectos de escritura.
- De base es gratuito y cuenta con una gran cantidad de funcionalidades básicas para la edición de documentos LaTeX.

Desventajas:

- Aunque es una gran plataforma, Overleaf puede tener limitaciones en la personalización y configuración avanzada en comparación con los editores de LaTeX locales.
- La dependencia de la conexión a Internet puede ser una limitación para aquellos que trabajan en entornos con conectividad limitada o inestable.
- Para proyectos muy grandes quizás requieras usar alguna de las funcionalidades de pago, como por ejemplo para reducir el tiempo de compilación del documento.

En conclusión, empleo Overleaf como mi plataforma principal para la edición colaborativa de documentos LaTeX debido a su robusto editor en línea y su sistema de control de versiones integrado, que facilita la colaboración efectiva en proyectos de escritura. Además, su gratuidad y amplia gama de funcionalidades básicas hacen que sea accesible para usuarios de todos los niveles. Aunque reconozco que puede tener limitaciones en la personalización avanzada y que la dependencia de la conexión a Internet puede ser una restricción en algunos casos, considero que las ventajas en términos de colaboración en tiempo real y facilidad de uso justifican su elección.

5.11 OpenAI

OpenAI [46] es una organización de investigación en inteligencia artificial (IA) que se enfoca en desarrollar y promover tecnologías de IA seguras y beneficiosas para la humanidad. Reconocida por su trabajo en el desarrollo de modelos de lenguaje avanzados como GPT (Generative Pre-trained Transformer), OpenAI ha sido pionera en el campo de la IA y ha generado un gran impacto en diversas áreas, desde la generación de texto hasta la robótica.

Además, OpenAI ofrece una variedad de herramientas y recursos para desarrolladores y empresas interesadas en aprovechar la IA en sus proyectos. Esto incluye APIs y plataformas de desarrollo que permiten a los usuarios integrar capacidades de IA en aplicaciones y sistemas existentes, así como una comunidad activa que comparte conocimientos y mejores prácticas en el campo de la IA.

Ventajas:

- OpenAI proporciona acceso a modelos de IA de vanguardia, como GPT, que pueden ser utilizados para una amplia gama de aplicaciones, desde la generación de texto hasta la traducción automática.
- Su enfoque en la seguridad y la ética en la IA garantiza que las tecnologías desarrolladas por OpenAI sean seguras y beneficiosas para la sociedad.

Desventajas:

- A pesar de sus avances, OpenAI enfrenta desafíos en términos de escalabilidad y accesibilidad, ya que algunos de sus modelos y herramientas pueden requerir recursos computacionales significativos.
- La comprensión y el uso efectivo de las tecnologías de IA de OpenAI pueden requerir un conocimiento técnico especializado, lo que puede limitar su adopción por parte de usuarios menos técnicos.

En conclusión, integro OpenAI en mi proyecto porque su conjunto de herramientas y modelos de inteligencia artificial, como GPT, ofrecen un soporte invaluable en una variedad de tareas. Por ejemplo, al aprovechar GPT para la generación de texto, simplifico y agilizo la redacción de contenido complejo. Además, su capacidad para comprender y generar texto coherente y relevante puede mejorar la capacidad de aprendizaje con tecnologías y herramientas nuevas. OpenAI actúa como un aliado poderoso al proporcionar soluciones inteligentes que complementan y optimizan las tareas específicas de mi proyecto.

5.12 GitHub

GitHub [28] es una plataforma en línea ampliamente usada para el control de versiones y la colaboración en el desarrollo de software. Con millones de usuarios en todo el mundo, se ha convertido en una herramienta esencial para los desarrolladores. Permite a los usuarios almacenar, administrar y compartir su código fuente y proyectos de software de manera eficiente.

Por otro lado, GitHub cuenta con un robusto sistema de control de versiones que facilita a los desarrolladores rastrear y gestionar los cambios en el código fuente. Además, cuenta con herramientas de colaboración como solicitudes de extracción, comentarios y seguimiento de problemas, lo que fomenta la colaboración efectiva entre equipos de desarrollo y la contribución a proyectos de código abierto.

Ventajas:

- GitHub ofrece una interfaz fácil de usar y herramientas de colaboración robustas, como solicitudes de extracción y comentarios, para una colaboración efectiva entre desarrolladores.

- Su sistema de seguimiento de problemas centraliza la gestión de problemas y promueve la transparencia en el proceso de resolución.

Desventajas:

- GitHub, enfocado en proyectos de código abierto, requiere que los proyectos sean públicos, lo que puede no ser adecuado para proyectos privados o comerciales.
- Su enfoque en proyectos basados en código puede limitar su utilidad para proyectos que no involucren desarrollo de software, como diseño gráfico o documentos.

En conclusión, en mi proyecto GitHub desempeña un papel fundamental al proporcionar un entorno centralizado para el control de versiones y la colaboración en el desarrollo de software. Su interfaz intuitiva y robustas herramientas de colaboración, como solicitudes de extracción y seguimiento de problemas, facilitan la comunicación y la coordinación. Además, su sistema de control de versiones permite rastrear y gestionar cambios en el código fuente de manera eficiente. Aunque reconozco que GitHub está más orientado hacia proyectos de código abierto y puede no ser adecuado para proyectos privados, su capacidad para promover la transparencia y la eficiencia en el desarrollo de software lo convierte en una herramienta de alto valor para mi proyecto.

5.13 VirtualBox

VirtualBox [47] es una herramienta de virtualización de código abierto que permite a los usuarios crear y gestionar máquinas virtuales en sus sistemas operativos. Con una amplia base de usuarios en todo el mundo, VirtualBox se ha convertido en una solución popular para el desarrollo y la prueba de software, así como para la creación de entornos de desarrollo y pruebas.

Por su parte, VirtualBox ofrece una amplia gama de características y funcionalidades que facilitan la creación y gestión de máquinas virtuales. Esto incluye soporte para una variedad de sistemas operativos huésped, capacidades de red avanzadas y una interfaz de usuario intuitiva que permite a los usuarios configurar y administrar sus máquinas virtuales de manera eficiente.

Ventajas:

- VirtualBox ofrece una interfaz de usuario intuitiva y fácil de usar que facilita la creación y gestión de máquinas virtuales.
- Su soporte para una amplia variedad de sistemas operativos huésped permite a los usuarios crear entornos de desarrollo y pruebas flexibles y personalizados.

Desventajas:

- Aunque es una herramienta potente, VirtualBox puede tener un rendimiento inferior en comparación con otras soluciones de virtualización más especializadas.
- La configuración avanzada y las características adicionales pueden requerir un aprendizaje adicional por parte del usuario, lo que puede ser una barrera para algunos usuarios menos técnicos.

En conclusión, VirtualBox juega un papel importante al proporcionar una plataforma de virtualización de código abierto que facilita la creación y gestión de máquinas virtuales para probar que todo funciona correctamente. Su interfaz intuitiva y fácil de usar permite configurar entornos de desarrollo y pruebas de manera flexible, lo que es fundamental para mi trabajo. Además, su amplio soporte para diferentes sistemas operativos huésped ofrece la versatilidad necesaria para adaptar mis entornos según las necesidades específicas del proyecto. Aunque VirtualBox puede tener un rendimiento inferior en comparación con otras soluciones más especializadas y que la configuración avanzada puede requerir cierto aprendizaje adicional, sus ventajas superan estas limitaciones para mis propósitos.

5.14 Proxmox

Proxmox [54] es una plataforma de virtualización de código abierto que permite a los usuarios crear y administrar máquinas virtuales y contenedores en un entorno de servidor. Con una sólida comunidad de usuarios y una amplia gama de características, Proxmox es una opción muy popular para implementaciones de virtualización tanto en entornos domésticos como empresariales.

Por otra parte, Proxmox proporciona una serie de características avanzadas que hacen que la creación y gestión de máquinas virtuales sea eficiente y flexible. Esto incluye soporte para tecnologías de virtualización como KVM y contenedores LXC, así como una interfaz web intuitiva que permite a los usuarios administrar sus entornos de virtualización desde cualquier lugar.

Ventajas:

- Proxmox ofrece una interfaz web intuitiva y fácil de usar que simplifica la administración de máquinas virtuales y contenedores.
- Su soporte para múltiples tecnologías de virtualización, como KVM y contenedores LXC, permite a los usuarios crear entornos de virtualización flexibles y escalables.

Desventajas:

- A pesar de su versatilidad, Proxmox puede tener una curva de aprendizaje empinada para usuarios menos técnicos, especialmente cuando se trata de configuraciones avanzadas.
- En comparación con soluciones de virtualización más especializadas, como VMware vSphere, Proxmox puede carecer de ciertas características específicas o tener un rendimiento ligeramente inferior en ciertos escenarios.

En conclusión, Proxmox es una plataforma de virtualización de código abierto que simplifica la creación y gestión de máquinas virtuales y contenedores en servidores. La interfaz web intuitiva de Proxmox facilita la administración de estos entornos, lo cual es esencial para mis necesidades. Además, su versatilidad, con soporte para tecnologías como KVM y contenedores LXC, me permite construir entornos flexibles y escalables. Aunque reconozco que puede tener una curva de aprendizaje pronunciada para usuarios menos técnicos y puede carecer de algunas características especializadas, las ventajas que ofrece Proxmox son fundamentales para el éxito de este proyecto.

5.15 Canva

Canva [12] es una herramienta en línea ampliamente utilizada para el diseño gráfico y la creación de contenido visual. Con una gran cantidad de usuarios es una plataforma esencial para diseñadores, profesionales del marketing, educadores y personas en general que desean crear diseños atractivos de manera fácil y eficiente. Permite a los usuarios diseñar y editar una amplia variedad de elementos visuales, como presentaciones, publicaciones en redes sociales, infografías, folletos y mucho más, sin la necesidad de tener habilidades avanzadas de diseño o utilizar software costoso.

De igual forma, Canva ofrece una gran cantidad de plantillas predefinidas, imágenes, ilustraciones y herramientas de edición que hacen que la creación de diseños sea accesible para usuarios de todos los niveles. Su interfaz intuitiva y amigable facilita la personalización y la experimentación con diferentes estilos y elementos visuales.

Ventajas:

- Canva proporciona una amplia variedad de plantillas y herramientas de edición que permiten a los usuarios crear diseños atractivos de manera rápida y sencilla.
- Su interfaz intuitiva y amigable hace que sea accesible para usuarios de todos los niveles de habilidad, desde principiantes hasta profesionales.
- La opción gratuita de Canva ofrece una amplia gama de funcionalidades básicas, lo que la hace accesible para aquellos con presupuestos limitados.

Desventajas:

- Aunque es una herramienta poderosa, Canva puede tener limitaciones en términos de personalización avanzada y funciones específicas que pueden ser necesarias para proyectos muy especializados.
- La dependencia de la conexión a Internet puede ser una restricción para aquellos que trabajan en entornos con conectividad limitada o inestable.
- Para acceder a características avanzadas y a una biblioteca más amplia de elementos visuales, puede ser necesario suscribirse a un plan de pago.

En conclusión, decido utilizar Canva como mi plataforma principal para el diseño gráfico y la creación de contenido visual debido a su facilidad de uso, amplia variedad de plantillas y herramientas de edición, y su opción gratuita que ofrece funcionalidades básicas. Sin embargo, reconozco que puede tener limitaciones en términos de personalización avanzada y que la dependencia de la conexión a Internet puede ser una restricción en algunos casos, considero que las ventajas en términos de accesibilidad y rapidez de creación justifican su elección como herramienta de diseño en este proyecto.

Despliegue

6.1 Arquitectura

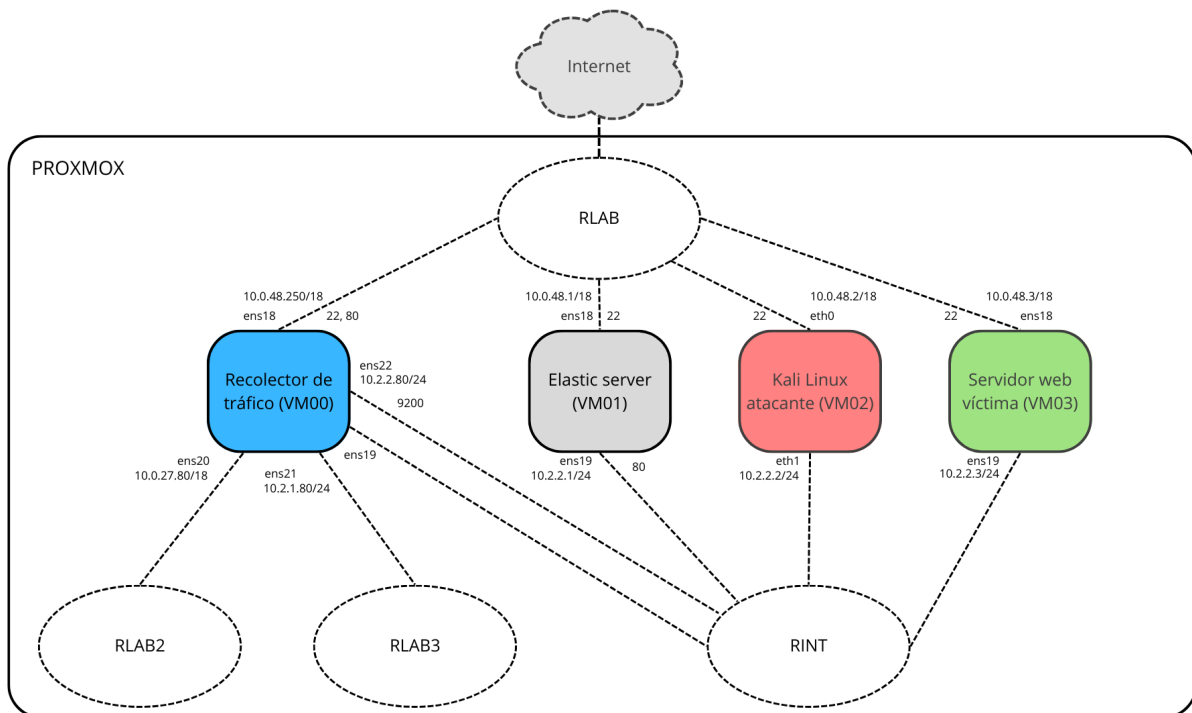


Figura 6.1: Arquitectura de red del proyecto

6.1.1 VM00 - Recolector de tráfico

▪ Interfaces y conexiones:

- **ens18**: Conectada a la subred **10.0.48.0/18**, con dirección IP **10.0.48.250**. Los puertos 22 (SSH) y 5601 están abiertos para acceso.
- **ens19**: Conectada a la subred **10.2.2.0/24**, se dedica exclusivamente a escuchar de manera pasiva el tráfico de red (no tiene IP).

Cabe destacar que para hacer que esto funcione en PROXMOX VE en la interfaz ens19, los técnicos de la Universidad de Valladolid tuvieron que aplicar un script de **Bridge Port Mirroring** para garantizar que todos los paquetes vayan a esta interfaz, todo ello viene explicado en esta página web [56].

- **ens20**: Conectada a la subred **10.0.27.0/18**.
 - **ens21**: Conectada a la subred **10.2.1.0/24**.
 - **ens22**: Conectada a la subred **10.2.2.0/24**, con dirección IP **10.2.2.80**.
- **Función**: Esta máquina recolecta y analiza el tráfico de red. Es el punto central donde se agrupan y procesan los datos de tráfico de las diferentes subredes y máquinas involucradas.

6.1.2 VM01 - Servidor Elastic

- **Interfaces y conexiones**:
- **ens18**: Conectada a la subred **10.0.48.0/18**, con dirección IP **10.0.48.1**. El puerto 22 (SSH) está abierto para acceso.
 - **ens19**: Conectada a la subred **10.2.2.0/24**, con dirección IP **10.2.2.80**. El puerto 9200 está abierto para acceso.
- **Función**: Almacena y proporciona acceso a los datos recolectados por VM00. Además, se utiliza para visualizar y analizar los datos de tráfico mediante herramientas como Kibana y mandar alertas por correo mediante Elastalert2.

6.1.3 VM02 - Kali Linux atacante

- **Interfaces y conexiones**:
- **eth0**: Conectada a la subred **10.0.48.0/18**, con dirección IP **10.0.48.2**. El puerto 22 (SSH) está abierto para acceso.
 - **eth1**: Conectada a la subred **10.2.2.0/24**, con dirección IP **10.2.2.2**.
- **Función**: Esta máquina es utilizada para ejecutar ataques de seguridad contra el servidor web víctima (VM03). Al estar en la misma subred que la víctima, puede lanzar diversos tipos de ataques y el tráfico resultante puede ser monitoreado por el recolector de tráfico (VM00).

6.1.4 VM03 - Servidor web víctima

- **Interfaces y conexiones**:
- **ens18**: Conectada a la subred **10.0.48.0/18**, con dirección IP **10.0.48.3**. El puerto 22 (SSH) está abierto para acceso.
 - **ens19**: Conectada a la subred **10.2.2.0/24**, con dirección IP **10.2.2.3**.
- **Función**: Actúa como el objetivo de los ataques lanzados desde Kali Linux (VM02). Está configurado para registrar y responder al tráfico de red, permitiendo la observación y análisis de los ataques.

6.1.5 Subredes

Nombre	Descripción	Red
RLAB	Conexión principal a Internet y a otras subredes internas	10.0.48.0/18
RLAB2	Subredes adicionales para generar tráfico propiedad de mi tutor de TFG	10.0.27.0/18
RLAB3	Subredes adicionales para generar tráfico propiedad de mi tutor de TFG	10.2.1.0/24
RINT	Red interna conectada a las subredes para comunicación entre las VMs	10.2.2.0/24

Cuadro 6.1: Descripción de las redes

6.2 Implementación

6.2.1 VM00 - Recolector de tráfico

Características VM00

Sistema Operativo	Ubuntu server 22.04.4 LTS
CPUs	2 núcleos
Memoria RAM	4 GiB
Tamaño del disco	32 GB
Número de interfaces de red	5 interfaces

Cuadro 6.2: Características VM00

Instalación de Ubuntu server 22.04.4 LTS

Primero, una vez instalada la versión de **Ubuntu server 22.04.4 LTS** en el laboratorio **PROXMOX** que me ha proporcionado la Universidad de Valladolid, procedemos a actualizar la máquina virtual con los comandos siguientes.

```
sudo apt update
sudo apt upgrade
```

Configuración de red

A continuación, para establecer la configuración de red editamos el fichero de configuración de red `/etc/network/interfaces` (Listado 6.1). En él configuraremos dos interfaces de red, la primera **ens18**, la cual es la interfaz que se conecta a la red Lab48 del laboratorio y tendrá una IP asignada por DHCP, la cual es **10.0.48.250/18**. La segunda interfaz **ens19** es la que está conectada a la red interna del laboratorio, ya que su única función consiste en escuchar el tráfico no tiene IP. La tercera interfaz **ens20**, es la que está conectada a una de las subredes de mi tutor de TFG cuya IP asignada por DHCP es **10.0.27.80/18**. La interfaz **ens21** también pertenece a una subred de mi tutor de TFG y su ip estática es **10.2.1.80/24**. Por último, la interfaz **ens22** pertenece a la red interna cuya ip estática es **10.2.2.80/18**. Hay que resaltar que las interfaces **ens19**, **ens20** y **ens21** tienen activado el modo promiscuo, lo que significa que también recibirán paquetes que no les pertenecen para facilitar la monitorización de la red con Suricata.

```
1 auto lo
2 iface lo inet loopback
3
4 # RLAB - VLAN 799 - IP Lab48 por DHCP
```

```

5 auto ens18
6 iface ens18 inet dhcp
7
8 # RINT - VLAN 798 - Red interna, tarjeta de escucha
9 auto ens19
10 iface ens19 inet manual
11 #     address 10.2.2.80
12 #     netmask 255.255.255.0
13     up ip link set ens19 promisc on
14
15 # RLAB2 - VLAN709 - IP Lab27 por DHCP
16 auto ens20
17 iface ens20 inet dhcp
18     up ip link set ens20 promisc on
19
20 # RLAB3 - VLAN 799 - LAB27 INTERNA
21 auto ens21
22 iface ens21 inet static
23     address 10.2.1.80
24     netmask 255.255.255.0
25     up ip link set ens21 promisc on
26
27 # RINT - VLAN 798 - Red interna
28 auto ens22
29 iface ens22 inet static
30     address 10.2.2.80
31     netmask 255.255.255.0

```

Listado 6.1: Fichero configuración /etc/network/interfaces de VM00

Después de guardar la configuración de red reiniciamos el servicio de red para aplicar correctamente los cambios realizados en el equipo con el comando que se muestra a continuación.

```
sudo systemctl restart networking.service
```

Por lo tanto, la configuración de red de la máquina virtual VM00 quedaría de la siguiente manera.

```

usuario@vm00:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
  1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
   link/ether 08:00:27:09:48:00 brd ff:ff:ff:ff:ff:ff
   altname enp0s18
   inet 10.0.48.250/18 brd 10.0.63.255 scope global dynamic ens18
       valid_lft 1031sec preferred_lft 1031sec
3: ens19: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
  group default qlen 1000
   link/ether 08:00:27:98:48:00 brd ff:ff:ff:ff:ff:ff
   altname enp0s19
4: ens20: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
  group default qlen 1000
   link/ether 08:00:27:09:27:80 brd ff:ff:ff:ff:ff:ff
   altname enp0s20
   inet 10.0.27.80/18 brd 10.0.63.255 scope global dynamic ens20
       valid_lft 1174sec preferred_lft 1174sec
5: ens21: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
  group default qlen 1000
   link/ether 08:00:27:99:48:00 brd ff:ff:ff:ff:ff:ff
   altname enp0s21
   inet 10.2.1.80/24 brd 10.2.1.255 scope global ens21

```

```

        valid_lft forever preferred_lft forever
6: ens22: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
    group default qlen 1000
    link/ether 08:01:27:98:48:00 brd ff:ff:ff:ff:ff:ff
    altname enp0s22
    inet 10.2.2.80/24 brd 10.2.2.255 scope global ens22
        valid_lft forever preferred_lft forever

```

Instalar Suricata

Para instalar Suricata, primero hay que añadir el repositorio de paquetes de la **Open Information Security Foundation (OISF)** al equipo con el comando siguiente.

```
sudo add-apt-repository ppa:oisf/suricata-stable
```

Después, con el posterior comando instalamos el IDS/IPS Suricata.

```
sudo apt install suricata
```

Suricata almacena su configuración en el archivo `/etc/suricata/suricata.yaml`. El modo por defecto de Suricata es el modo IDS (Sistema de Detección de Intrusos), en el que sólo se registra el tráfico y no toma acciones.

Lo primero que editaremos del fichero de configuración de Suricata será las redes y puertos que queremos que monitorice. Donde añadiremos en la variable `HOME_NET` las subredes **10.0.27.0/18**, **10.2.1.0/24**, **10.2.2.0/24**. Además, en la variable `HTTP_SERVERS` añadiremos la IP de nuestro servidor web víctima **10.2.2.3/24**. Esta declaración de variables se pueden observar en el Listado 6.2.

```

1 ##
2 ## Step 1: Inform Suricata about your network
3 ##
4
5 vars:
6 # more specific is better for alert accuracy and performance
7 address-groups:
8   HOME_NET: "[10.0.27.0/18,10.2.1.0/24,10.2.2.1,10.2.2.3,10.2.2.80]"
9
10  EXTERNAL_NET: "!$HOME_NET"
11
12  HTTP_SERVERS: "10.2.2.3/24"
13  SMTP_SERVERS: "$HOME_NET"
14  SQL_SERVERS: "$HOME_NET"
15  DNS_SERVERS: "$HOME_NET"
16  TELNET_SERVERS: "$HOME_NET"
17  AIM_SERVERS: "$EXTERNAL_NET"
18  DC_SERVERS: "$HOME_NET"
19  DNP3_SERVER: "$HOME_NET"
20  DNP3_CLIENT: "$HOME_NET"
21  MODBUS_CLIENT: "$HOME_NET"
22  MODBUS_SERVER: "$HOME_NET"
23  ENIP_CLIENT: "$HOME_NET"
24  ENIP_SERVER: "$HOME_NET"
25
26 port-groups:
27  HTTP_PORTS: "80"
28  SHELLCODE_PORTS: "!80"
29  ORACLE_PORTS: 1521
30  SSH_PORTS: 22
31  DNP3_PORTS: 20000
32  MODBUS_PORTS: 502
33  FILE_DATA_PORTS: "[$HTTP_PORTS,110,143]"
34  FTP_PORTS: 21

```

```

35 GENEVE_PORTS: 6081
36 VXLAN_PORTS: 4789
37 TEREDO_PORTS: 3544

```

Listado 6.2: Definición de redes en suricata.yaml de VM00

Por otro lado, en el fichero de configuración de Suricata activaremos el campo ID de Comunidad, el cual facilita la correlación de datos entre registros generados por distintas herramientas de monitorización y como usaremos Elasticsearch nos será útil. Para activarlo localizaremos la línea **# Community Flow ID** y estableceremos el valor de la variable **community-id** en **true**, tal y como se puede ver en el Listado 6.3.

```

1 # Community Flow ID
2 # Adds a 'community_id' field to EVE records. These are meant to give
3 # records a predictable flow ID that can be used to match records to
4 # output of other tools such as Zeek (Bro).
5 #
6 # Takes a 'seed' that needs to be same across sensors and tools
7 # to make the id less predictable.
8
9 # enable/disable the community id feature.
10 community-id: true

```

Listado 6.3: Community Flow ID en suricata.yaml de VM00

Ahora, los eventos llevarán un ID como **1:S+3BA2UmrHK0Pk+u3XH78GAFTtQ=** que podremos utilizar para hacer coincidir conjuntos de datos entre distintas herramientas de monitorización.

A continuación, definiremos las interfaces de red que queremos que Suricata monitorice, para ello buscaremos la línea **af-packet:** y justo debajo de esta estableceremos el valor y las variables de las interfaces que se van a monitorizar, quedaría como en el Listado 6.4.

```

1 af-packet:
2 - interface: ens19
3   threads: auto
4   cluster-id: 99
5   cluster-type: cluster_flow
6   defrag: yes
7
8 - interface: ens20
9   threads: auto
10  cluster-id: 98
11  cluster-type: cluster_flow
12  defrag: yes
13
14 - interface: ens21
15   threads: auto
16   cluster-id: 97
17   cluster-type: cluster_flow
18   defrag: yes
19
20 # Put default values here. These will be used for an interface that is not
21 # in the list above.
22 - interface: default
23   #threads: auto
24   #use-mmap: no
25   #tpacket-v3: yes

```

Listado 6.4: Interfaces de red en suricata.yaml de VM00

Además, quitaremos el fichero de reglas por defecto de Suricata y añadiremos unas reglas personalizadas a Suricata modificando el fichero de configuración **/etc/suricata/suricata.yaml** como se ve en el Listado 6.5.


```

1 ##
2 ## Configure Suricata to load Suricata-Update managed rules.
3 ##
4
5 default-rule-path: /var/lib/suricata/rules
6
7 rule-files:
8   - custom.rules

```

Listado 6.5: Ficheros de reglas en suricata.yaml de VM00

Por otra parte, si se quisiera ampliar las reglas de Suricata se pueden añadir más proveedores de reglas. Con el comando que podemos observar posteriormente listaremos los proveedores disponibles.

```
sudo suricata-update list-sources
```

Por ejemplo podríamos añadir el conjunto de reglas de **et/open** con el siguiente comando.

```
sudo suricata-update enable-source et/open
```

Después, se ejecutaría el comando que hay después para descargar y actualizar las nuevas reglas.

```
sudo suricata-update
```

El fichero **custom.rules** (Listado 6.6) tendrá las siguientes reglas personalizadas:

- Reglas para detectar ataques SQLI, explicadas en el Cuadro 6.3.
- Reglas para detectar ataques LFI, explicadas en el Cuadro 6.4.
- Reglas para detectar ataques RFI, explicadas en el Cuadro 6.5.
- Reglas para detectar ataques de fuerza bruta, explicadas en el Cuadro 6.6.
- Reglas para detectar ataques DDoS/DoS, explicadas en el Cuadro 6.7.

SID	Mensaje	Descripción
100001	Possible SQL Injection Attempt, SELECT FROM in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'SELECT' y 'FROM'. Estas palabras son comúnmente utilizadas en consultas SQL para seleccionar datos de una tabla, y su presencia en una URI puede indicar un intento de ataque.
100002	Possible SQL Injection Attempt, UNION SELECT in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'UNION' y 'SELECT'. La cláusula 'UNION' se utiliza para combinar los resultados de dos consultas SQL, y su uso junto con 'SELECT' en una URI puede ser un indicador de un ataque de inyección SQL.
100003	Possible SQL Injection Attempt, INSERT INTO in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'INSERT' y 'INTO'. Estas palabras se utilizan en consultas SQL para insertar nuevos registros en una tabla, y su aparición en una URI puede señalar un ataque.

100004	Possible SQL Injection Attempt, UPDATE SET in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'UPDATE' y 'SET'. Estas palabras son utilizadas en consultas SQL para actualizar registros existentes en una tabla. Su presencia en una URI puede indicar un intento de modificar datos a través de un ataque de inyección SQL.
100005	Possible SQL Injection Attempt, DELETE FROM in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'DELETE' y 'FROM'. Estas palabras son utilizadas en consultas SQL para eliminar registros de una tabla, y su uso en una URI puede ser un signo de un intento de ataque.
100006	Possible SQL Injection Attempt, ALTER in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen la palabra clave 'ALTER'. Se aplica específicamente a comandos que alteran la estructura de la base de datos, tales como 'ALTER TABLE', 'ALTER DATABASE', etc. Estos comandos pueden ser utilizados maliciosamente para cambiar la estructura de la base de datos.
100007	Possible SQL Injection Attempt, DROP in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen la palabra clave 'DROP'. Esta palabra se utiliza en consultas SQL para eliminar bases de datos, tablas u otros objetos. Su presencia en una URI puede indicar un intento de destruir datos.
100008	Possible SQL Injection Attempt, CREATE in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen la palabra clave 'CREATE'. Específicamente se refiere a comandos que crean nuevos objetos en la base de datos, tales como 'CREATE TABLE', 'CREATE DATABASE', etc. Su uso en una URI puede ser un signo de un intento de crear estructuras maliciosas en la base de datos.
100009	Possible SQL Injection Attempt, SELECT CONCAT in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'SELECT' y 'CONCAT'. La función 'CONCAT' se utiliza para concatenar cadenas en SQL. Su uso en combinación con 'SELECT' puede ser un indicador de un ataque destinado a manipular o extraer datos.
100010	Possible SQL Injection Attempt, BULK INSERT in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'BULK' y 'INSERT'. El comando 'BULK INSERT' se utiliza para importar un gran volumen de datos en una tabla. Su presencia en una URI puede indicar un intento de cargar datos masivos de forma maliciosa.
100011	Possible SQL Injection Attempt, SHOW TABLES in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'SHOW' y 'TABLES'. Este comando se utiliza para listar todas las tablas en una base de datos. Su uso en una URI puede ser un intento de enumerar la estructura de la base de datos.

100012	Possible SQL Injection Attempt, SHOW CHARACTER SET in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen las palabras clave 'SHOW', 'CHARACTER' y 'SET'. Este comando se utiliza para mostrar los conjuntos de caracteres soportados por la base de datos. Su presencia en una URI puede ser un signo de exploración de la base de datos.
100013	Possible SQL Injection Attempt, detected 'OR' Pattern in URI	Esta regla detecta intentos de inyección SQL en la URI que contienen el patrón 'OR'. La presencia de 'OR' en una URI, especialmente en el contexto de comparaciones booleanas (como OR 1=1 o OR '1'='1'), puede indicar un intento de alterar la lógica de una consulta SQL para bypassar autenticación o filtros.

Cuadro 6.3: Reglas SQLI en Suricata

SID	Mensaje	Descripción
200001	Possible Local File Inclusion Attempt, ../ or variations in URI	Esta regla detecta intentos de inclusión de archivos locales (LFI) mediante el uso de secuencias de directorios ascendentes '../' y sus variaciones en la URI. Los ataques de LFI pueden permitir a un atacante leer archivos arbitrarios en el servidor, lo que puede conducir a la revelación de información sensible o incluso a la ejecución de código malicioso.
200002	Possible Local File Inclusion Attempt, URL encoded variations of ../ in URI	Esta regla detecta variaciones de '../' codificadas en URL, como '%2e%2e', en la URI. Los atacantes pueden utilizar la codificación en URL para intentar evadir la detección de LFI.
200003	Possible Local File Inclusion Attempt, Double URL encoded variations of ../ in URI	Esta regla detecta variaciones doblemente codificadas de '../' en URL, como '%252e%252e', en la URI. Los atacantes pueden utilizar la codificación doble en URL para intentar evadir aún más la detección de LFI.
200004	Possible Local File Inclusion Attempt, /etc/passwd in URI	Esta regla detecta intentos de acceso al archivo '/etc/passwd' en la URI. El archivo '/etc/passwd' es comúnmente objetivo de ataques de LFI debido a su importancia en sistemas Unix/Linux.
200005	Possible Local File Inclusion Attempt, /proc/version in URI	Esta regla detecta intentos de acceso al archivo '/proc/version' en la URI. El archivo '/proc/version' puede revelar información sobre el kernel del sistema, lo que lo convierte en un objetivo para ataques de LFI.
200006	Possible Local File Inclusion Attempt, /etc/shadow in URI	Esta regla detecta intentos de acceso al archivo '/etc/shadow' en la URI. El archivo '/etc/shadow' contiene contraseñas cifradas en sistemas Unix/Linux, y su acceso puede conducir a la obtención de credenciales de usuario.
200007	Possible Local File Inclusion Attempt, file:// in URI	Esta regla detecta intentos de uso del protocolo 'file://' en la URI. El protocolo 'file://' permite el acceso a archivos locales en el sistema, y su uso puede indicar intentos de LFI.

200008	Possible Local File Inclusion Attempt, Hex encoded ../ in URI	Esta regla detecta secuencias ‘../’ codificadas en hexadecimal, como ‘%c0%af’, en la URI. Los atacantes pueden utilizar la codificación en hexadecimal para intentar evadir la detección de LFI.
200009	Possible Local File Inclusion Attempt, Unicode encoded ../ in URI	Esta regla detecta secuencias ‘../’ codificadas en Unicode, como ‘%u002e%u002e%u002f’, en la URI. Los atacantes pueden utilizar la codificación en Unicode para intentar evadir aún más la detección de LFI.
200010	Possible Local File Inclusion Attempt, Null Byte in URI	Esta regla detecta intentos de LFI que utilizan null bytes (‘%00’) para terminar cadenas en la URI. Los null bytes pueden ser utilizados por los atacantes para evitar la detección y manipular la interpretación de la ruta del archivo.

Cuadro 6.4: Reglas LFI en Suricata

SID	Mensaje	Descripción
300001	Possible Remote File Inclusion Attempt, http/https in URI	Esta regla detecta intentos de inclusión de archivos remotos (RFI) mediante el uso de los prefijos ‘http://’ o ‘https://’ en la URI. Los ataques de RFI pueden permitir a un atacante incluir y ejecutar código malicioso alojado en un servidor externo.
300002	Possible Remote File Inclusion Attempt, PHP include/require functions in URI	Esta regla detecta el uso de funciones de inclusión de archivos de PHP como ‘include(’, ‘require(’, ‘include_once(’, ‘require_once(’ en la URI. Estas funciones pueden ser explotadas por atacantes para incluir y ejecutar código malicioso alojado en un servidor externo.
300003	Possible Remote File Inclusion Attempt, PHP Wrapper in URI	Esta regla detecta el uso de wrappers de PHP como ‘php://’ en la URI. Los wrappers de PHP pueden ser utilizados por atacantes para incluir y ejecutar código malicioso alojado en un servidor externo.
300004	Possible Remote File Inclusion Attempt, Common Remote File Extensions in URI	Esta regla detecta intentos de RFI utilizando extensiones de archivos comunes como ‘.asp’, ‘.aspx’, ‘.jsp’ en la URI. Los atacantes pueden aprovechar estas extensiones para incluir y ejecutar código malicioso alojado en un servidor externo.
300005	Possible Remote File Inclusion Attempt, Proxy Parameters in URI	Esta regla detecta parámetros comunes de proxy en la URI como ‘proxy’, ‘url’, ‘dest’. Los atacantes pueden intentar usar parámetros de proxy para dirigir solicitudes a través de un servidor proxy y ocultar su ubicación real.

Cuadro 6.5: Reglas RFI en Suricata

SID	Mensaje	Descripción
400001	Possible Brute Force Login Attempt, 10 logins in 10 seconds	Detecta intentos de fuerza bruta en los que se realizan 10 intentos de inicio de sesión en 10 segundos en ‘/vulnerabilities/brute/’.
400002	Possible Brute Force Login Attempt, Common Usernames	Detecta intentos de fuerza bruta utilizando nombres de usuario comunes en el formulario de inicio de sesión de ‘/vulnerabilities/brute/’.

400003	Possible Brute Force Login Attempt, Common Passwords	Detecta intentos de fuerza bruta utilizando contraseñas comunes en el formulario de inicio de sesión de '/vulnerabilities/brute/'.
400004	Possible Hydra Brute Force Attack	Detecta posibles ataques de fuerza bruta realizados por la herramienta Hydra en '/vulnerabilities/brute/'.

Cuadro 6.6: Reglas de Fuerza Bruta en Suricata

SID	Mensaje	Descripción
500001	Possible SYN Flood Attack	Detecta cuando hay más de 5000 paquetes SYN en 20 segundos desde una única fuente, lo cual es típico de un ataque de inundación SYN.
500002	Possible UDP Flood Attack	Detecta cuando hay más de 5000 paquetes UDP en 20 segundos desde una única fuente, lo cual es típico de un ataque de inundación UDP.
500003	Possible SMURF Flood Attack	Detecta cuando hay más de 5000 paquetes ICMP en 20 segundos desde una única fuente, lo cual es típico de un ataque de inundación SMURF (ICMP).
500004	Possible LAND Flood Attack	Detecta posibles ataques LAND, donde los paquetes tienen la misma dirección IP de origen y destino.

Cuadro 6.7: Reglas DDoS en Suricata

```

1 # Reglas para detectar SQL Injection (SQLI)
2 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  SELECT FROM in URI"; flow:established,to_server; content:"SELECT"; nocase; http_uri;
  content:"FROM"; nocase; http_uri; classtype:web-application-attack; sid:100001; rev
  :1;)
3
4 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  UNION SELECT in URI"; flow:established,to_server; content:"UNION"; nocase; http_uri;
  content:"SELECT"; nocase; http_uri; classtype:web-application-attack; sid:100002; rev
  :1;)
5
6 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  INSERT INTO in URI"; flow:established,to_server; content:"INSERT"; nocase; http_uri;
  content:"INTO"; nocase; http_uri; classtype:web-application-attack; sid:100003; rev
  :1;)
7
8 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  UPDATE SET in URI"; flow:established,to_server; content:"UPDATE"; nocase; http_uri;
  content:"SET"; nocase; distance:0; http_uri; classtype:web-application-attack; sid
  :100004; rev:1;)
9
10 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  DELETE FROM in URI"; flow:established,to_server; content:"DELETE"; nocase; http_uri;
  content:"FROM"; nocase; http_uri; classtype:web-application-attack; sid:100005; rev
  :1;)
11
12 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
  ALTER in URI"; flow:to_server,established; uricontent:"ALTER"; nocase; pcre:"/ALTER\
  +(database|procedure|table|column)/Ui"; classtype:web-application-attack; sid:100006;
  rev:1;)
13

```

```

14 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    DROP in URI"; flow:to_server,established; uricontent:"DROP"; nocase; pcre:"/DROP\ +(
    database|procedure|table|column)/Ui"; classtype:web-application-attack; sid:100007;
    rev:1;)
15
16 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    CREATE in URI"; flow:to_server,established; uricontent:"CREATE"; nocase; pcre:"/
    CREATE\ +(database|procedure|table|column|directory)/Ui"; classtype:web-application-
    attack; sid:100008; rev:1;)
17
18 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    SELECT CONCAT in URI"; flow:established,to_server; uricontent:"SELECT"; nocase;
    uricontent:"CONCAT"; nocase; pcre:"/SELECT.+CONCAT/Ui"; classtype:web-application-
    attack; sid:100009; rev:1;)
19
20 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    BULK INSERT in URI"; flow:established,to_server; content:"BULK"; nocase; http_uri;
    content:"INSERT"; nocase; http_uri; distance:0; classtype:web-application-attack; sid
    :100010; rev:1;)
21
22 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    SHOW TABLES in URI"; flow:established,to_server; uricontent:"SHOW"; nocase;
    uricontent:"TABLES"; nocase; pcre:"/SHOW.+TABLES/Ui"; classtype:web-application-
    attack; sid:100011; rev:1;)
23
24 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt,
    SHOW CHARACTER SET in URI"; flow:established,to_server; uricontent:"SHOW"; nocase;
    uricontent:"CHARACTER"; nocase; content:"SET"; nocase; pcre:"/SHOW.+CHARACTER.+SET/Ui
    "; classtype:web-application-attack; sid:100012; rev:1;)
25
26 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible SQL Injection Attempt -
    Detected 'OR' Pattern in URI"; flow:to_server,established; content:"GET"; http_method
    ; content:"/"; http_uri; pcre:"/\bOR\b/i"; classtype:web-application-attack; sid
    :100013; rev:1;)
27
28
29 # Local File Inclusion (LFI) Rules
30 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, ../ or variations in URI"; flow:to_server,established; uricontent:"../";
    classtype:web-application-attack; sid:200001; rev:2;)
31
32 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, URL encoded variations of ../ in URI"; flow:to_server,established;
    uricontent:"%2e%2e"; nocase; classtype:web-application-attack; sid:200002; rev:1;)
33
34 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, Double URL encoded variations of ../ in URI"; flow:to_server,established;
    uricontent:"%252e%252e"; nocase; classtype:web-application-attack; sid:200003; rev
    :1;)
35
36 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, /etc/passwd in URI"; flow:to_server,established; uricontent:"/etc/passwd";
    nocase; classtype:web-application-attack; sid:200004; rev:1;)
37
38 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, /proc/version in URI"; flow:to_server,established; uricontent:"/proc/version
    "; nocase; classtype:web-application-attack; sid:200005; rev:1;)
39
40 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, /etc/shadow in URI"; flow:to_server,established; uricontent:"/etc/shadow";
    nocase; classtype:web-application-attack; sid:200006; rev:1;)
41

```

```

42 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, file:// in URI"; flow:to_server,established; uricontent:"file://"; nocase;
    classtype:web-application-attack; sid:200007; rev:1;)
43
44 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, Hex encoded ../ in URI"; flow:to_server,established; uricontent:"%c0%af";
    nocase; classtype:web-application-attack; sid:200008; rev:1;)
45
46 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, Unicode encoded ../ in URI"; flow:to_server,established; uricontent:"%u002e%
    u002e%u002f"; nocase; classtype:web-application-attack; sid:200009; rev:1;)
47
48 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Local File Inclusion
    Attempt, Null Byte in URI"; flow:to_server,established; uricontent:"%00"; classtype:
    web-application-attack; sid:200010; rev:1;)
49
50
51 # Remote File Inclusion (RFI) Rules
52 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Remote File Inclusion
    Attempt, http/https in URI"; flow:to_server,established; uricontent:"=http"; nocase;
    classtype:web-application-attack; sid:300001; rev:3;)
53
54 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Remote File Inclusion
    Attempt, PHP include/require functions in URI"; flow:to_server,established; pcre:"/(
    include\(|require\(|include_once\(|require_once\()/i"; classtype:web-application-
    attack; sid:300002; rev:1;)
55
56 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Remote File Inclusion
    Attempt, PHP Wrapper in URI"; flow:to_server,established; uricontent:"php://"; nocase
    ; classtype:web-application-attack; sid:300003; rev:2;)
57
58 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Remote File Inclusion
    Attempt, Common Remote File Extensions in URI"; flow:to_server,established; pcre:"
    /(\.asp|\.aspx|\.jsp)/i"; classtype:web-application-attack; sid:300004; rev:2;)
59
60 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Remote File Inclusion
    Attempt, Proxy Parameters in URI"; flow:to_server,established; pcre:"/(proxy|url|dest
    )/i"; classtype:web-application-attack; sid:300005; rev:2;)
61
62
63 # Brute force Rules
64 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Brute Force Login Attempt,
    10 logins in 10 seconds"; flow:to_server,established; uricontent:"/vulnerabilities/
    brute/"; uricontent:"Login"; threshold: type both, track by_src, count 10, seconds
    10; classtype:web-application-attack; sid:400001; rev:1;)
65
66 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Brute Force Login Attempt,
    Common Usernames"; flow:to_server,established; content:"/vulnerabilities/brute/";
    http_uri; pcre:"/(username=)(administrator|root|user|test)/Ui"; classtype:web-
    application-attack; sid:400002; rev:1;)
67
68 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Brute Force Login Attempt,
    Common Passwords"; flow:to_server,established; uricontent:"/vulnerabilities/brute/";
    pcre:"/(password=)(password123|123456|admin|admin123|qwerty)/Ui"; classtype:web-
    application-attack; sid:400003; rev:1;)
69
70 alert http any any -> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible Hydra Brute Force Attack";
    flow:to_server,established; content:"Hydra"; threshold: type both, track by_src,
    count 10, seconds 10; classtype:web-application-attack; sid:400004; rev:1;)
71
72
73 # Denial of Service (DoS) and Distributed Denial of Services (DDoS) Rules

```

```

74 alert tcp any any -> $HTTP_SERVERS any (msg:"Possible SYN Flood Attack"; flow:to_server;
    flags:S; threshold: type both, track by_dst, count 5000, seconds 20; classtype:
    denial-of-service; sid:500001; rev:1;)
75
76 alert udp any any -> $HTTP_SERVERS any (msg:"Possible UDP Flood Attack"; flow:to_server;
    threshold: type both, track by_dst, count 5000, seconds 20; classtype:denial-of-
    service; sid:500002; rev:1;)
77
78 alert icmp any any -> $HTTP_SERVERS any (msg:"Possible SMURF Flood Attack"; flow:
    to_server; threshold: type both, track by_dst, count 5000, seconds 20; classtype:
    denial-of-service; sid:500003; rev:1;)
79
80 alert tcp $HTTP_SERVERS $HTTP_PORTS <> $HTTP_SERVERS $HTTP_PORTS (msg:"Possible LAND
    Flood Attack"; classtype:denial-of-service; sid:500004; rev:1;)

```

Listado 6.6: Fichero custom.rules de VM00

A continuación, valido el fichero de configuración de Suricata para comprobar que todo está correctamente configurado, de la siguiente manera.

```

usuario@vm00:~$ sudo suricata -T -c /etc/suricata/suricata.yaml -v
Notice: suricata: This is Suricata version 7.0.5 RELEASE running in SYSTEM mode
Info: cpu: CPUs/cores online: 2
Info: suricata: Running suricata under test mode
Info: suricata: Setting engine mode to IDS mode by default
Info: exception-policy: master exception-policy set to: auto
Info: logopenfile: fast output device (regular) initialized: fast.log
Info: logopenfile: eve-log output device (regular) initialized: eve.json
Info: logopenfile: stats output device (regular) initialized: stats.log
Info: detect-parse: Rule with ID 500004 is bidirectional, but source and destination are
    the same, treating the rule as unidirectional
Info: detect: 1 rule files processed. 36 rules successfully loaded, 0 rules failed, 0
Info: threshold-config: Threshold config parsed: 0 rule(s) found
Info: detect: 36 signatures processed. 1 are IP-only rules, 6 are inspecting packet
    payload, 26 inspect application layer, 0 are decoder event only
Notice: suricata: Configuration provided was successfully loaded. Exiting.

```

Después, iniciamos Suricata con el comando que vemos a continuación.

```
sudo systemctl start suricata
```

Por último, comprobamos que Suricata se está ejecutando satisfactoriamente.

```

usuario@vm00:~$ sudo systemctl status suricata●
suricata.service - LSB: Next Generation IDS/IPS
  Loaded: loaded (/etc/init.d/suricata; generated)
  Active: active (running) since Wed 2024-06-12 17:55:36 CEST; 24min ago
  Docs: man:systemd-sysv-generator(8)
  Process: 25490 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
  Tasks: 12 (limit: 4557)
  Memory: 92.3M
  CPU: 54.432s
  CGroup: /system.slice/suricata.service
          └─25569 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/run/
suricata.pid --af-packet -D>

jun 12 17:55:36 vm00 systemd[1]: Starting LSB: Next Generation IDS/IPS...
jun 12 17:55:36 vm00 suricata[25490]: Starting suricata in IDS (af-packet) mode... done.
jun 12 17:55:36 vm00 systemd[1]: Started LSB: Next Generation IDS/IPS.

```

Instalar Filebeat

Primero, nos descargamos Filebeat en su versión **8.13.2** en el directorio `/home/usuario` con el siguiente comando.


```
curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-8.13.2-linux-x86_64.tar.gz
```

Acto seguido, lo descomprimimos con el comando que podemos ver a continuación.

```
tar xzvf filebeat-8.13.2-linux-x86_64.tar.gz
```

Una vez descomprimido podemos eliminar el comprimido que nos descargamos para ahorrar espacio de disco en la máquina.

```
rm filebeat-8.13.2-linux-x86_64.tar.gz
```

Accedemos a la carpeta que hemos descomprimido de Filebeat y editamos el fichero de configuración **filebeat.yml** de la siguiente manera.

Primero, ponemos el parámetro **setup.dashboards.enabled** a **true** como podemos ver en el Listado 6.7.

```
1 # ===== Dashboards =====
2 # These settings control loading the sample dashboards to the Kibana index. Loading
3 # the dashboards is disabled by default and can be enabled either by setting the
4 # options here or by using the `setup` command.
5 setup.dashboards.enabled: true
```

Listado 6.7: Variable dashboards en filebeat.yml de VM00

En la sección de Kibana del fichero de configuración de Filebeat añadimos el host con la url de Kibana, además de su usuario y contraseña para Kibana, tal y como se puede apreciar en el Listado 6.8.

```
1 # ===== Kibana =====
2
3 # Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
4 # This requires a Kibana endpoint configuration.
5 setup.kibana:
6
7 # Kibana Host
8 # Scheme and port can be left out and will be set to the default (http and 5601)
9 # In case you specify and additional path, the scheme is required: http://localhost
10 # :5601/path
11 # IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
12 host: "http://10.2.2.1:80"
13 username: "elastic"
14 password: "elastic123"
15
16 # Kibana Space ID
17 # ID of the Kibana Space into which the dashboards should be loaded. By default,
18 # the Default Space will be used.
19 #space.id:
```

Listado 6.8: Kibana setup en filebeat.yml de VM00

Por último, en el fichero de configuración en la sección de Elasticsearch (Listado 6.9) añadimos el host de Elasticsearch, el protocolo https y el usuario y contraseña de elastic, así como el certificado ssl que genera Elasticsearch y que debemos pasarnos de la máquina VM01.

```
1 # ----- Elasticsearch Output -----
2 output.elasticsearch:
3 # Array of hosts to connect to.
4 hosts: ["10.2.2.1:9200"]
5 #index: "filebeat-%{[agent.version]}-%{+yyyy.MM.dd}"
6 # Performance preset - one of "balanced", "throughput", "scale",
7 # "latency", or "custom".
8 #preset: balanced
9
```

```

10 # Protocol - either `http` (default) or `https`.
11 protocol: "https"
12
13 # Authentication credentials - either API key or username/password.
14 #api_key: "id:api_key"
15 username: "elastic"
16 password: "elastic123"
17
18 ssl.enabled: true
19 ssl.certificate_authorities: ["/home/usuario/filebeat-8.13.2-linux-x86_64/ca.crt"]

```

Listado 6.9: Elasticsearch setup en filebeat.yaml de VM00

Una vez configurado el fichero de configuración de Filebeat comprobamos que Suricata está en la lista de módulos disponibles en Filebeat.

```
./filebeat modules list
```

Después, habilitamos el módulo de Suricata con el siguiente comando.

```
./filebeat modules enable suricata
```

Dentro de la carpeta de instalación de Filebeat editamos el fichero de configuración del módulo de Suricata con la información que vemos en el Listado 6.10.

```

1 # Module: suricata
2 # Docs: https://www.elastic.co/guide/en/beats/filebeat/8.12/filebeat-module-suricata.html
3
4 - module: suricata
5   # All logs
6   eve:
7     enabled: true
8
9   # Set custom paths for the log files. If left empty,
10  # Filebeat will choose the paths depending on your OS.
11  var.paths: ["/var/log/suricata/eve.json"]

```

Listado 6.10: Fichero de configuración del módulo de Suricata de VM00

Por último, para iniciar Filebeat en segundo plano y que se comiencen a enviar los logs de Suricata a la máquina VM01 con Elasticsearch es necesario utilizar este comando.

```
./filebeat setup &
```

Es importante recordar que antes de iniciar Filebeat hay que pasar el certificado **ca.crt** que genera Elasticsearch en la máquina VM01 para que el SSL funcione correctamente.

6.2.2 VM01 - Servidor Elastic

Características VM01

Sistema Operativo	Ubuntu server 22.04.4 LTS
CPUs	2 núcleos
Memoria RAM	8 GiB
Tamaño del disco	64 GB
Número de interfaces de red	2 interfaces

Cuadro 6.8: Características VM01

Instalación de Ubuntu server 22.04.4 LTS

Primero, una vez instalada la versión de **Ubuntu server 22.04.4 LTS** en el laboratorio **PROXMOX** que me ha proporcionado la Universidad de Valladolid, procedemos a actualizar la máquina virtual con los comandos siguientes.

```
sudo apt update
sudo apt upgrade
```

Configuración de red

A continuación, para establecer la configuración de red editamos el fichero de configuración de red **/etc/network/interfaces** (Listado 6.11). En él configuraremos dos interfaces de red, la primera **ens18**, la cual es la interfaz que se conecta a la red Lab48 del laboratorio y tendrá una IP asignada por DHCP, la cual será **10.0.48.1/18**. La segunda interfaz **ens19** es la que esta conectada a la red interna del laboratorio cuya IP estática es **10.2.2.1/24**.

```
1 auto lo
2 iface lo inet loopback
3
4 # RLAB - VLAN 799 - IP Lab48 por DHCP
5 auto ens18
6 iface ens18 inet dhcp
7
8 # RINT - VLAN 798
9 auto ens19
10 iface ens19 inet static
11     address 10.2.2.1
12     netmask 255.255.255.0
```

Listado 6.11: Fichero de configuración de red /etc/network/interfaces de VM01

Después de guardar la configuración de red reiniciamos el servicio de red para aplicar correctamente los cambios realizados en el equipo con el comando siguiente.

```
sudo systemctl restart networking.service
```

Por lo tanto, la configuración de red de la máquina virtual quedaría de la siguiente manera.

```
root@vm01:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
  1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
   link/ether 08:00:27:09:48:01 brd ff:ff:ff:ff:ff:ff
   altname enp0s18
   inet 10.0.48.1/18 brd 10.0.63.255 scope global dynamic ens18
       valid_lft 1489sec preferred_lft 1489sec
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
   link/ether 08:00:27:98:48:01 brd ff:ff:ff:ff:ff:ff
   altname enp0s19
   inet 10.2.2.1/24 brd 10.2.2.255 scope global ens19
       valid_lft forever preferred_lft forever
```

Instalar Docker y Docker compose

Instalamos Docker en el equipo con el comando mostrado a continuación.

```
sudo apt install docker.io
```

Posteriormente comprobaremos, tal y como se puede ver posteriormente se ha instalado correctamente en su versión **24.0.5**.

```
root@vm01:~# docker -v
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
```

Ahora instalamos Docker compose con el comando posterior, la cual es una herramienta complementaria a Docker, la cual nos ayudará a desplegar los contenedores de una manera más cómoda.

```
sudo apt install docker-compose-v2
```

Además, comprobamos que se instaló en el sistema de manera correcta en su versión **2.20.2**.

```
root@vm01:~# docker compose version
Docker Compose version 2.20.2+ds1-0ubuntu1~22.04.1
```

Por último, añadimos al usuario usuario al grupo de Docker, para que este pueda usar Docker sin problemas. Esto se hace modificando línea **docker:x:120:usuario** en el fichero de configuración **/etc/group**.

Para aplicar los cambios en Docker reiniciamos los servicios de Docker para evitar conflictos.

```
sudo systemctl restart docker.service docker.socket
```

Despliegue de Elasticsearch, Kibana y Elastalert2 con Docker compose

Primero, desde el directorio home de usuario creamos un directorio vacío llamado **elk**.

```
mkdir elk
```

A continuación, dentro de este nuevo directorio **elk** creamos el fichero de configuración de variables de entorno para el despliegue llamado **.env** (Listado 6.12), en él podemos observar que se encuentran las contraseñas de Elastic y Kibana, la versión de Elastic y de Elastalert2, la ip donde Filebeat enviará los logs, los puertos de Elasticsearch y Kibana, entre otras cosas.

```
1 # Project namespace (defaults to the current folder name if not set)
2 #COMPOSE_PROJECT_NAME=myproject
3
4 # Password for the 'elastic' user (at least 6 characters)
5 ELASTIC_PASSWORD=elastic123
6
7 # Password for the 'kibana_system' user (at least 6 characters)
8 KIBANA_PASSWORD=kibana123
9
10 # Version of Elastic products
11 STACK_VERSION=8.13.2
12
13 # Version of Elastalert2 service
14 ELASTALERT2_VERSION=2.17.0
15
16 # Set the cluster name
17 CLUSTER_NAME=elk-cluster
18
19 # Set the Filebeat bay ip
20 FILEBEAT_BAY=10.2.2.1
21
22 # Set to 'basic' or 'trial' to automatically start the 30-day trial
23 LICENSE=basic
24 #LICENSE=trial
```

```
25
26 # Port to expose Elasticsearch HTTP API to the host
27 ES_PORT=9200
28
29 # Port to expose Kibana to the host
30 KIBANA_PORT=80
31
32 # Increase or decrease based on the available host memory (in bytes)
33 ES_MEM_LIMIT=4294967296
34 KB_MEM_LIMIT=2147483648
35
36 # SAMPLE Predefined Key only to be used in POC environments
37 ENCRYPTION_KEY=345d578f69bc0e217333076f441f120bc6bdf37101ae19eb05aac9d3cbe026c1
```

Listado 6.12: Fichero de configuración .env de VM01

Después, crearemos el fichero de despliegue **docker-compose.yaml** (Listado 6.13), donde viene como se desplegará cada servicio. Primero estarían definidos los 3 volúmenes que se van a usar, uno para almacenar los certificados llamado **certs**, otro donde se almacenará la los datos de elasticsearch que se llama **elasticsearchdata** y otro donde se guardarán los datos de kibana llamado **kibanadata**.

Siguiendo el fichero encontraremos la sección **networks** donde estará definida una red interna de Docker la cual se usará para que los servicios se comuniquen entre sí, cuyo nombre es **elastic**.

A continuación, el servicio **setup** es crucial para la configuración inicial de la pila Elastic. Utiliza la imagen de Elasticsearch de la versión especificada por la variable de entorno **STACK_VERSION**. Este servicio se encarga de crear los certificados necesarios para la comunicación segura entre los diferentes componentes del sistema. Para ello, monta un volumen llamado **certs** donde guarda estos certificados. El comando que ejecuta el contenedor realiza varias tareas: primero, verifica que las contraseñas para Elasticsearch y Kibana están definidas en las variables de entorno; luego, crea una Autoridad Certificadora (CA) si no existe y genera los certificados para Elasticsearch y Kibana. Finalmente, ajusta los permisos de los archivos y espera a que Elasticsearch esté disponible para configurar la contraseña del usuario **kibana_system**. Este servicio también incluye un **healthcheck** que verifica la existencia del archivo **elasticsearch.crt** para asegurarse de que la configuración inicial ha terminado correctamente.

El servicio **elasticsearch** depende del servicio setup, asegurándose de que el proceso de configuración inicial ha finalizado con éxito antes de iniciarse. Utiliza la misma imagen de Elasticsearch y monta dos volúmenes: uno para los certificados y otro para los datos de Elasticsearch. Expone el puerto **9200** del contenedor al host y configura diversas variables de entorno para la seguridad y el SSL. También establece límites de memoria y ajusta los límites de memoria bloqueada (memlock). Un **healthcheck** adicional se asegura de que Elasticsearch responde correctamente a las solicitudes y que la autenticación está configurada correctamente.

El servicio **kibana** depende del servicio elasticsearch para asegurarse de que Elasticsearch esté completamente operativo antes de iniciar Kibana. Utiliza la imagen de Kibana correspondiente a la versión especificada y monta volúmenes para los certificados y datos de Kibana. Expone el puerto **5601** del contenedor al host y configura las variables de entorno necesarias para conectarse a Elasticsearch de manera segura. También establece un límite de memoria para el contenedor. Un **healthcheck** verifica que Kibana responde adecuadamente a las solicitudes HTTP.

Finalmente, el servicio **elastalert2** depende tanto de Elasticsearch como de Kibana, garantizando que ambos servicios estén en estado saludable antes de su inicio. Utiliza la imagen de ElastAlert2 y está configurado para reiniciarse automáticamente a menos que se detenga explícitamente. Monta varios volúmenes locales para su configuración y reglas, permitiendo una fácil gestión y actualización de las alertas que ElastAlert2 debe manejar.

```

1 version: "2"
2
3 volumes:
4   certs:
5     driver: local
6     name: certs
7   elasticsearchdata:
8     driver: local
9     name: elasticsearchdata
10  kibana:
11    driver: local
12    name: kibana
13
14 networks:
15   default:
16     name: elastic
17     external: false
18
19 services:
20   setup:
21     container_name: setup
22     image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
23     volumes:
24       - certs:/usr/share/elasticsearch/config/certs
25     user: "0"
26     command: >
27       bash -c '
28         if [ x${ELASTIC_PASSWORD} == x ]; then
29           echo "Set the ELASTIC_PASSWORD environment variable in the .env file";
30           exit 1;
31         elif [ x${KIBANA_PASSWORD} == x ]; then
32           echo "Set the KIBANA_PASSWORD environment variable in the .env file";
33           exit 1;
34         fi;
35         if [ ! -f config/certs/ca.zip ]; then
36           echo "Creating CA";
37           bin/elasticsearch-certutil ca --silent --pem -out config/certs/ca.zip;
38           unzip config/certs/ca.zip -d config/certs;
39         fi;
40         if [ ! -f config/certs/certs.zip ]; then
41           echo "Creating certs";
42           echo -ne \
43             "instances:\n"\
44             "  - name: elasticsearch\n"\
45             "  dns:\n"\
46             "    - elasticsearch\n"\
47             "    - localhost\n"\
48             "  ip:\n"\
49             "    - 127.0.0.1\n"\
50             "    - ${FILEBEAT_HOST}\n"\
51             "  - name: kibana\n"\
52             "  dns:\n"\
53             "    - kibana\n"\
54             "    - localhost\n"\
55             "  ip:\n"\
56             "    - 127.0.0.1\n"\
57             > config/certs/instances.yml;
58           bin/elasticsearch-certutil cert --silent --pem -out config/certs/certs.zip --in
59             config/certs/instances.yml --ca-cert config/certs/ca/ca.crt --ca-key config/certs/ca
60             /ca.key;
61           unzip config/certs/certs.zip -d config/certs;

```

```

60     cp config/certs/ca/ca.crt config/certs/ca/ca.pem
61     fi;
62     echo "Setting file permissions"
63     chown -R root:root config/certs;
64     find . -type d -exec chmod 750 \{\} \;;
65     find . -type f -exec chmod 640 \{\} \;;
66     echo "Waiting for Elasticsearch availability";
67     until curl -s --cacert config/certs/ca/ca.crt https://elasticsearch:9200 | grep -
q "missing authentication credentials"; do sleep 30; done;
68     echo "Setting kibana_system password";
69     until curl -s -X POST --cacert config/certs/ca/ca.crt -u "elastic:${
ELASTIC_PASSWORD}" -H "Content-Type: application/json" https://elasticsearch:9200/
_security/user/kibana_system/_password -d "{\"password\": \"${KIBANA_PASSWORD}\"} " |
grep -q "^{}"; do sleep 10; done;
70     echo "All done!";
71     '
72     healthcheck:
73     test: ["CMD-SHELL", "[ -f config/certs/elasticsearch/elasticsearch.crt ]"]
74     interval: 1s
75     timeout: 5s
76     retries: 120
77
78     elasticsearch:
79     container_name: elasticsearch
80     depends_on:
81     setup:
82     condition: service_healthy
83     image: docker.elastic.co/elasticsearch/elasticsearch:${STACK_VERSION}
84     labels:
85     co.elastic.logs/module: elasticsearch
86     volumes:
87     - certs:/usr/share/elasticsearch/config/certs
88     - elasticsearchdata:/usr/share/elasticsearch/data
89     ports:
90     - ${FILEBEAT_BAY}:${ES_PORT}:9200
91     environment:
92     - node.name=elasticsearch
93     - cluster.name=${CLUSTER_NAME}
94     - discovery.type=single-node
95     - ELASTIC_PASSWORD=${ELASTIC_PASSWORD}
96     - bootstrap.memory_lock=true
97     - xpack.security.enabled=true
98     - xpack.security.http.ssl.enabled=true
99     - xpack.security.http.ssl.key=certs/elasticsearch/elasticsearch.key
100    - xpack.security.http.ssl.certificate=certs/elasticsearch/
101    elasticsearch.crt
102    - xpack.security.http.ssl.certificate_authorities=certs/ca/ca.crt
103    - xpack.security.transport.ssl.enabled=true
104    - xpack.security.transport.ssl.key=certs/elasticsearch/
105    elasticsearch.key
106    - xpack.security.transport.ssl.certificate=certs/elasticsearch/
107    elasticsearch.crt
108    - xpack.security.transport.ssl.certificate_authorities=certs/ca/
109    ca.crt
110    - xpack.security.transport.ssl.verification_mode=certificate
111    - xpack.license.self_generated.type=${LICENSE}
112    mem_limit: ${ES_MEM_LIMIT}
113    ulimits:
114    memlock:
115    soft: -1
116    hard: -1
117    healthcheck:

```

```

118     test:
119     [
120         "CMD-SHELL",
121         "curl -s --cacert config/certs/ca/ca.crt https://localhost:9200 | grep -q '
missing authentication credentials'",
122     ]
123     interval: 10s
124     timeout: 10s
125     retries: 120
126
127 kibana:
128     container_name: kibana
129     depends_on:
130         elasticsearch:
131             condition: service_healthy
132     image: docker.elastic.co/kibana/kibana:${STACK_VERSION}
133     labels:
134         co.elastic.logs/module: kibana
135     volumes:
136     - certs:/usr/share/kibana/config/certs
137     - kibanadata:/usr/share/kibana/data
138     ports:
139     - ${KIBANA_PORT}:5601
140     environment:
141     - SERVERNAME=kibana
142     - ELASTICSEARCH_HOSTS=https://elasticsearch:9200
143     - ELASTICSEARCH_USERNAME=kibana_system
144     - ELASTICSEARCH_PASSWORD=${KIBANA_PASSWORD}
145     - ELASTICSEARCH_SSL_CERTIFICATEAUTHORITIES=config/certs/ca/ca.crt
146     - XPACK_SECURITY_ENCRYPTIONKEY=${ENCRYPTION_KEY}
147     - XPACK_ENCRYPTEDSAVEDOBJECTS_ENCRYPTIONKEY=${ENCRYPTION_KEY}
148     - XPACK_REPORTING_ENCRYPTIONKEY=${ENCRYPTION_KEY}
149     mem_limit: ${KB_MEM_LIMIT}
150     healthcheck:
151     test:
152     [
153         "CMD-SHELL",
154         "curl -s -I http://localhost:5601 | grep -q 'HTTP/1.1 302 Found'",
155     ]
156     interval: 10s
157     timeout: 10s
158     retries: 120
159
160 elastalert2:
161     container_name: elastalert2
162     depends_on:
163         elasticsearch:
164             condition: service_healthy
165         kibana:
166             condition: service_healthy
167     image: jertel/elastalert2:${ELASTALERT2_VERSION}
168     restart: unless-stopped
169     volumes:
170     - ./elastalert2/elastalert2.yaml:/opt/elastalert/config.yaml
171     - ./elastalert2/rules:/opt/elastalert/rules
172     - ./elastalert2/smtp_auth.yaml:/opt/elastalert/smtp_auth.yaml

```

Listado 6.13: Fichero docker-compose.yaml de VM01

Para la configuración de elastalert2, dentro del directorio creado previamente `~/elk` creamos un directorio para elastalert2.


```
mkdir elastalert2
```

En este nuevo directorio creamos el fichero de configuración **elastalert2.yaml** (Listado 6.14).

```
1 # This is the folder that contains the rule yaml files
2 # This can also be a list of directories
3 # Any .yaml file will be loaded as a rule
4 rules_folder: /opt/elastalert/rules
5
6 # How often ElastAlert will query Elasticsearch
7 # The unit can be anything from weeks to seconds
8 run_every:
9   seconds: 10
10
11 # ElastAlert will buffer results from the most recent
12 # period of time, in case some log sources are not in real time
13 buffer_time:
14   minutes: 1
15
16 # The Elasticsearch hostname for metadata writeback
17 # Note that every rule can have its own Elasticsearch host
18 es_host: elasticsearch
19
20 # The Elasticsearch port
21 es_port: 9200
22
23 # Option basic-auth username and password for Elasticsearch
24 es_username: elastic
25 es_password: elastic123
26
27 # Connect with TLS to Elasticsearch
28 use_ssl: False
29
30 # Verify TLS certificates
31 verify_certs: False
32   #ca_certs: /opt/elastalert/ca.crt
33
34 # The index on es_host which is used for metadata storage
35 # This can be a unmapped index, but it is recommended that you run
36 # elastalert-create-index to set a mapping
37 writeback_index: elastalert_status
38 writeback_alias: elastalert_alerts
39
40 # If an alert fails for some reason, ElastAlert will retry
41 # sending the alert until this time period has elapsed
42 alert_time_limit:
43   days: 2
```

Listado 6.14: Fichero elastalert2.yaml de VM01

El primer parámetro del este fichero de configuración **rules_folder** especifica la carpeta donde se encuentran los archivos YAML que contienen las reglas de ElastAlert. Esta carpeta puede ser una lista de directorios y cualquier archivo con extensión **.yaml** en estas ubicaciones será cargado como una regla por ElastAlert. En este caso, las reglas se encuentran en **/opt/elastalert/rules**.

El parámetro **run_every** define la frecuencia con la que ElastAlert consultará Elasticsearch en busca de datos que coincidan con las reglas configuradas. En este archivo de configuración, se ha establecido para ejecutar cada 10 segundos. Esto significa que ElastAlert verificará los datos nuevos en Elasticsearch cada 10 segundos para identificar posibles coincidencias con las reglas definidas.

ElastAlert2 utiliza **buffer_time** para manejar retrasos en la llegada de los datos. Este parámetro define

cuánto tiempo de datos recientes se deben almacenar en el buffer para asegurar que todas las entradas de log, incluso aquellas que no llegan en tiempo real, sean procesadas. En este caso, el tiempo de buffer está configurado en 1 minuto.

El parámetro **es_host** especifica el nombre del host de Elasticsearch al que ElastAlert se conectará para escribir metadatos. En este archivo, se ha configurado como `elasticsearch`, lo que implica que ElastAlert buscará un host con este nombre para conectarse.

El parámetro **es_port** indica el puerto en el que Elasticsearch está escuchando. Aquí está configurado para usar el puerto 9200, que es el puerto predeterminado para Elasticsearch.

Para la autenticación básica, **es_username** y **s_password** se utilizan para proporcionar un nombre de usuario y una contraseña. En esta configuración, se ha establecido el usuario **elastic** y la contraseña **elastic123**.

La variable **use_ssl** determina si ElastAlert debe conectarse a Elasticsearch utilizando TLS (Transport Layer Security). En este archivo, está configurado como **False**, lo que significa que no se utilizará SSL para la conexión. Si se desea utilizar SSL, este parámetro se configuraría como **True**.

El parámetro **verify_certs** especifica si se deben verificar los certificados TLS. Aquí está configurado como **False**, lo que significa que ElastAlert no verificará los certificados cuando se conecte a Elasticsearch. Si se deseara verificar los certificados, este parámetro se configuraría como **True** y se podría proporcionar la ruta al archivo de certificados mediante el parámetro **ca_certs**.

Estos parámetros **writeback_index** y **writeback_alias** definen el índice en Elasticsearch que ElastAlert utilizará para almacenar metadatos y los alias para las alertas. El índice de writeback está configurado como **elastalert_status** y el alias para las alertas como **elastalert_alerts**. Se recomienda ejecutar **elastalert-create-index** para crear el índice con el mapeo adecuado.

Finalmente, **alert_time_limit** establece el límite de tiempo durante el cual ElastAlert intentará reenviar una alerta en caso de que falle el envío inicial. En esta configuración, está establecido en 2 días, lo que significa que ElastAlert intentará enviar la alerta durante un periodo de hasta 2 días si inicialmente falla.

Además, dentro del mismo directorio `~/elk/elastalert2/` creamos el fichero **smtp_auth.yaml** (Listado 6.15) donde están las credenciales para que se produzca la conexión al servidor de correo SMTP.

```
1 user: juapage13@gmail.com
2 password: gmailapppassword
```

Listado 6.15: Fichero `smtp_auth.yaml` de VM01

En lo que respecta a la contraseña, es necesario añadir en la cuenta de GMAIL una **App password** [26] para garantizar que Elastalert2 funcione sin ningún tipo de inconveniente. Cabe destacar que hay que activar previamente el **2-Step Verification** para poder añadir la **App password**.

Para crearla primero accederemos la cuenta de GMAIL y en la parte superior derecha de la pantalla pinchamos en nuestro perfil, después seleccionamos **Manage your Google Account**.

Se nos abrirá otra pestaña y seleccionaremos en la parte izquierda **Security** y acto seguido entraremos en **2-Step Verification**.

Una vez dentro de **2-Step Verification** bajamos al final de la página y ahí ya nos aparecerá la sección para añadir y gestionar las **App password**, tal y como se puede ver en la Figura 6.2.

Por otro lado, creamos otra carpeta dentro del directorio `~/elk/elastalert2/` para almacenar las reglas de Elastalert2.

← App passwords

App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.

[Learn more](#)

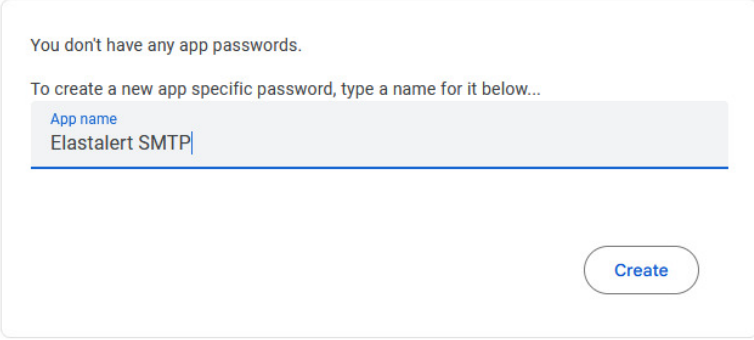


Figura 6.2: Creación de una App password para ElastAlert2 en GMAIL

`mkdir rules`

Dentro de la carpeta `~/elk/elastalert2/rules/` creamos los 5 archivos que definen las reglas para elastalert2.

El fichero de reglas `sqli_alert_rule.yaml` (Listado 6.16) se enfoca en detectar intentos de inyección SQL (SQL Injection) en el sistema. Monitorea los índices de Filebeat con el patrón `.ds-filebeat-*`. La alerta se desencadena cuando se encuentra una alerta de Suricata con un `signature_id` entre `100001` y `100013`. Esta configuración cubre las firmas relacionadas con ataques de inyección SQL. Cuando se detecta un evento que cumple estos criterios, se envía un correo electrónico a `juapage13@gmail.com` con los detalles del ataque.

```

1 name: Suricata SQLI Alert
2 index: ".ds-filebeat-*"
3 type: any
4
5 realert:
6   minutes: 5
7
8 filter:
9   - range:
10     suricata.eve.alert.signature_id:
11       from: 100001
12       to: 100013
13
14 alert:
15   - "email"
16
17 email:
18   - "juapage13@gmail.com"
19
20 smtp_host: "smtp.gmail.com"
21 smtp_port: 587
22 smtp_ssl: false
23 from_addr: "juapage13@gmail.com"

```

```

24 smtp_auth_file: "/opt/elastalert/smtp_auth.yaml"
25
26 email_format: "html"
27 alert_subject: "SURICATA SQLI ALERT"
28 alert_text_type: alert_text_only
29 alert_text_args:
30 - "@timestamp"
31 - "suricata.eve.alert.signature"
32 - "suricata.eve.alert.signature_id"
33 - "suricata.eve.alert.category"
34 - "destination.ip"
35 alert_text: |
36 <body style='font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;
padding: 0; display: flex; justify-content: center; align-items: center; height: 100
vh;'>
37   <div style='background-color: #fff; border: 2px solid #e74c3c; border-radius: 10px;
padding: 20px; box-shadow: 0 0 20px rgba(0, 0, 0, 0.1); max-width: auto; max-height:
600px; text-align: left;'>
38     <h2 style='color: #e74c3c; font-size: 2.5em; margin-bottom: 30px;'>Alerta de
Suricata!</h2>
39     <p style='font-size: 1.5em; margin-bottom: 20px;'>Se ha detectado un intento de
ataque <b>SQL Injection (SQLI)</b> en el sistema.</p>
40     <p style='font-size: 1.5em; margin-bottom: 20px;'>La información sobre el
incidente es la siguiente:</p>
41     <ul style='list-style: none; padding: 0;'>
42       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Timestamp: {0}</li>
43       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Rule name: {1}</li>
44       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Rule SID: {2}</li>
45       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Category: {3}</li>
46       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Host affected: {4}</li>
47     </ul>
48     <p style='font-size: 1.5em; margin-bottom: 20px;'>Consulta los registros desde
<i>Kibana</i> para obtener más información:</p>
49     <li><a href='http://vm4801.virtual.lab.inf.uva.es' style='color: #3498db; text-
decoration: none; font-size: 1.5em;'>http://vm4801.virtual.lab.inf.uva.es</a></li>
50   </div>
51 </body>

```

Listado 6.16: Fichero sqli_alert_rule.yaml de VM01

El fichero de reglas **lfi_alert_rule.yaml** (Listado 6.17) está diseñado para detectar intentos de inclusión de archivos locales (Local File Inclusion). Al igual que la fichero de reglas anterior, se aplica a los índices **.ds-filebeat-***. La diferencia clave es el rango de **signature_id**, que en este caso es de **200001** a **200010**. Este rango cubre las firmas de Suricata que corresponden a ataques de inclusión de archivos locales (LFI). La alerta por correo electrónico mantiene el mismo formato y destinatario que el fichero de reglas SQLI anterior.

```

1 name: Suricata LFI Alert
2 index: ".ds-filebeat-*"
3 type: any
4
5 realert:
6   minutes: 5
7
8 filter:
9 - range:
10   suricata.eve.alert.signature_id:
11     from: 200001

```

```

12     to: 200010
13
14 alert:
15 - "email"
16
17 email:
18 - "juapage13@gmail.com"
19
20 smtp_host: "smtp.gmail.com"
21 smtp_port: 587
22 smtp_ssl: false
23 from_addr: "juapage13@gmail.com"
24 smtp_auth_file: "/opt/elastalert/smtp_auth.yaml"
25
26 email_format: "html"
27 alert_subject: "SURICATA LFI ALERT"
28 alert_text_type: alert_text_only
29 alert_text_args:
30 - "@timestamp"
31 - "suricata.eve.alert.signature"
32 - "suricata.eve.alert.signature_id"
33 - "suricata.eve.alert.category"
34 - "destination.ip"
35 alert_text: |
36 <body style='font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;
37 padding: 0; display: flex; justify-content: center; align-items: center; height: 100
38 vh;'>
39     <div style='background-color: #fff; border: 2px solid #e74c3c; border-radius: 10px;
40 padding: 20px; box-shadow: 0 0 20px rgba(0, 0, 0, 0.1); max-width: auto; max-height:
41 600px; text-align: left;'>
42         <h2 style='color: #e74c3c; font-size: 2.5em; margin-bottom: 30px;'>Alerta de
43 Suricata!</h2>
44         <p style='font-size: 1.5em; margin-bottom: 20px;'>Se ha detectado un intento de
45 ataque <b>Local File Inclusion (LFI)</b> en el sistema.</p>
46         <p style='font-size: 1.5em; margin-bottom: 20px;'>La información sobre el
47 incidente es la siguiente:</p>
48         <ul style='list-style: none; padding: 0;'>
49             <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
50 px; margin-bottom: 10px; border-radius: 5px;'>Timestamp: {0}</li>
51             <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
52 px; margin-bottom: 10px; border-radius: 5px;'>Rule name: {1}</li>
53             <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
54 px; margin-bottom: 10px; border-radius: 5px;'>Rule SID: {2}</li>
55             <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
56 px; margin-bottom: 10px; border-radius: 5px;'>Category: {3}</li>
57             <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
58 px; margin-bottom: 10px; border-radius: 5px;'>Host affected: {4}</li>
59         </ul>
60         <p style='font-size: 1.5em; margin-bottom: 20px;'>Consulta los registros desde
61 <i>Kibana</i> para obtener más información:</p>
62         <li><a href='http://vm4801.virtual.lab.inf.uva.es' style='color: #3498db; text-
63 decoration: none; font-size: 1.5em;'>http://vm4801.virtual.lab.inf.uva.es</a></li>
64     </div>
65 </body>

```

Listado 6.17: Fichero lfi_alert_rule.yaml de VM01

El fichero de reglas **rfi_alert_rule.yaml** (Listado 6.18) se centra en detectar intentos de inclusión de archivos remotos (Remote File Inclusion, RFI). Monitoriza los mismos índices de Filebeat, pero con un rango de **signature_id** de **300001** a **300005**. Este rango abarca las firmas relacionadas con ataques de inclusión de archivos remotos. La notificación por correo electrónico sigue el mismo formato estándar, enviándose a **juapa-**

ge13@gmail.com.

```

1 name: Suricata RFI Alert
2 index: ".ds-filebeat-*"
3 type: any
4
5 realert:
6   minutes: 5
7
8 filter:
9   - range:
10     suricata.eve.alert.signature_id:
11       from: 300001
12       to: 300005
13
14 alert:
15   - "email"
16
17 email:
18   - "juapage13@gmail.com"
19
20 smtp_host: "smtp.gmail.com"
21 smtp_port: 587
22 smtp_ssl: false
23 from_addr: "juapage13@gmail.com"
24 smtp_auth_file: "/opt/elastalert/smtp_auth.yaml"
25
26 email_format: "html"
27 alert_subject: "SURICATA RFI ALERT"
28 alert_text_type: alert_text_only
29 alert_text_args:
30   - "@timestamp"
31   - "suricata.eve.alert.signature"
32   - "suricata.eve.alert.signature_id"
33   - "suricata.eve.alert.category"
34   - "destination.ip"
35 alert_text: |
36   <body style='font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;
37     padding: 0; display: flex; justify-content: center; align-items: center; height: 100
38     vh;'>
39     <div style='background-color: #fff; border: 2px solid #e74c3c; border-radius: 10px;
40     padding: 20px; box-shadow: 0 0 20px rgba(0, 0, 0, 0.1); max-width: auto; max-height:
41     600px; text-align: left;'>
42     <h2 style='color: #e74c3c; font-size: 2.5em; margin-bottom: 30px;'>¡Alerta de
43     Suricata!</h2>
44     <p style='font-size: 1.5em; margin-bottom: 20px;'>Se ha detectado un intento de
45     ataque <b>Remote File Inclusion (RFI)</b> en el sistema.</p>
46     <p style='font-size: 1.5em; margin-bottom: 20px;'>La información sobre el
47     incidente es la siguiente:</p>
48     <ul style='list-style: none; padding: 0;'>
49       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
50       px; margin-bottom: 10px; border-radius: 5px;'>Timestamp: {0}</li>
51       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
52       px; margin-bottom: 10px; border-radius: 5px;'>Rule name: {1}</li>
53       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
54       px; margin-bottom: 10px; border-radius: 5px;'>Rule SID: {2}</li>
55       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
56       px; margin-bottom: 10px; border-radius: 5px;'>Category: {3}</li>
57       <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
58       px; margin-bottom: 10px; border-radius: 5px;'>Host affected: {4}</li>
59     </ul>
60     <p style='font-size: 1.5em; margin-bottom: 20px;'>Consulta los registros desde

```

```

49 <i>Kibana</i> para obtener más información:</p>
    <li><a href='http://vm4801.virtual.lab.inf.uva.es' style='color: #3498db; text-
50 decoration: none; font-size: 1.5em;'>http://vm4801.virtual.lab.inf.uva.es</a></li>
51 </div>
</body>

```

Listado 6.18: Fichero rfi_alert_rule.yaml de VM01

El fichero de reglas **ddos_alert_rule.yaml** (Listado 6.19) detecta ataques de denegación de servicio distribuido (DDoS) tiene un **signature_id** que varía entre **500001** y **500004**. Esta configuración específica permite identificar las firmas asociadas con los ataques DDoS. De nuevo, la regla aplica a los índices **.ds-filebeat-***, y las alertas se envían por correo electrónico utilizando el mismo formato y destinatario.

```

1 name: Suricata DDoS Alert
2 index: ".ds-filebeat-*"
3 type: any
4
5 realert:
6   minutes: 5
7
8 filter:
9   - range:
10     suricata.eve.alert.signature_id:
11       from: 500001
12       to: 500004
13
14 alert:
15   - "email"
16
17 email:
18   - "juapage13@gmail.com"
19
20 smtp_host: "smtp.gmail.com"
21 smtp_port: 587
22 smtp_ssl: false
23 from_addr: "juapage13@gmail.com"
24 smtp_auth_file: "/opt/elastalert/smtp_auth.yaml"
25
26 email_format: "html"
27 alert_subject: "SURICATA DDoS ALERT"
28 alert_text_type: alert_text_only
29 alert_text_args:
30   - "@timestamp"
31   - "suricata.eve.alert.signature"
32   - "suricata.eve.alert.signature_id"
33   - "suricata.eve.alert.category"
34   - "destination.ip"
35 alert_text: |
36 <body style='font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;
37 padding: 0; display: flex; justify-content: center; align-items: center; height: 100
38 vh;'>
39   <div style='background-color: #fff; border: 2px solid #e74c3c; border-radius: 10px;
40 padding: 20px; box-shadow: 0 0 20px rgba(0, 0, 0, 0.1); max-width: auto; max-height:
41 600px; text-align: left;'>
42     <h2 style='color: #e74c3c; font-size: 2.5em; margin-bottom: 30px;'>Alerta de
43 Suricata!</h2>
44     <p style='font-size: 1.5em; margin-bottom: 20px;'>Se ha detectado un intento de
45 ataque <b>Distributed Denial of Service (DDoS)</b> en el sistema.</p>
46     <p style='font-size: 1.5em; margin-bottom: 20px;'>La información sobre el
47 incidente es la siguiente:</p>
48     <ul style='list-style: none; padding: 0;'>

```

```

42     <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
43     px; margin-bottom: 10px; border-radius: 5px;'>Timestamp: {0}</li>
44     <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
45     px; margin-bottom: 10px; border-radius: 5px;'>Rule name: {1}</li>
46     <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
47     px; margin-bottom: 10px; border-radius: 5px;'>Rule SID: {2}</li>
48     <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
49     px; margin-bottom: 10px; border-radius: 5px;'>Category: {3}</li>
50     <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
51     px; margin-bottom: 10px; border-radius: 5px;'>Host affected: {4}</li>
52   </ul>
53   <p style='font-size: 1.5em; margin-bottom: 20px;'>Consulta los registros desde
54   <i>Kibana</i> para obtener más información:</p>
55   <li><a href='http://vm4801.virtual.lab.inf.uva.es' style='color: #3498db; text-
56   decoration: none; font-size: 1.5em;'>http://vm4801.virtual.lab.inf.uva.es</a></li>
57 </div>
58 </body>

```

Listado 6.19: Fichero ddos_alert_rule.yaml de VM01

Por último, el fichero de reglas **brute_force_alert_rule.yaml** (Listado 6.20) se utiliza para detectar ataques de fuerza bruta. El rango de **signature_id** para esta regla es de **400001** a **400004**. Esta configuración cubre las firmas que identifican intentos de fuerza bruta. La regla opera sobre los mismos índices de Filebeat y las alertas se envían a **juapage13@gmail.com** con el formato ya establecido.

```

1 name: Suricata Brute Force Alert
2 index: ".ds-filebeat-*"
3 type: any
4
5 realert:
6   minutes: 5
7
8 filter:
9   - range:
10     suricata.eve.alert.signature_id:
11       from: 400001
12       to: 400004
13
14 alert:
15   - "email"
16
17 email:
18   - "juapage13@gmail.com"
19
20 smtp_host: "smtp.gmail.com"
21 smtp_port: 587
22 smtp_ssl: false
23 from_addr: "juapage13@gmail.com"
24 smtp_auth_file: "/opt/elastalert/smtp_auth.yaml"
25
26 email_format: "html"
27 alert_subject: "SURICATA BRUTE FORCE ALERT"
28 alert_text_type: alert_text_only
29 alert_text_args:
30   - "@timestamp"
31   - "suricata.eve.alert.signature"
32   - "suricata.eve.alert.signature_id"
33   - "suricata.eve.alert.category"
34   - "destination.ip"
35 alert_text: |
36   <body style='font-family: Arial, sans-serif; background-color: #f4f4f4; margin: 0;
37   padding: 0; display: flex; justify-content: center; align-items: center; height: 100

```



```

37     <div style='background-color: #fff; border: 2px solid #e74c3c; border-radius: 10px;
padding: 20px; box-shadow: 0 0 20px rgba(0, 0, 0, 0.1); max-width: auto; max-height:
600px; text-align: left;'>
38     <h2 style='color: #e74c3c; font-size: 2.5em; margin-bottom: 30px;'>¡Alerta de
Suricata!</h2>
39     <p style='font-size: 1.5em; margin-bottom: 20px;'>Se ha detectado un intento de
ataque de <b>Fuerza Bruta (Brute Force)</b> en el sistema.</p>
40     <p style='font-size: 1.5em; margin-bottom: 20px;'>La información sobre el
incidente es la siguiente:</p>
41     <ul style='list-style: none; padding: 0;'>
42         <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Timestamp: {0}</li>
43         <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Rule name: {1}</li>
44         <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Rule SID: {2}</li>
45         <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Category: {3}</li>
46         <li style='font-size: 1.4em; background: #e74c3c; color: #fff; padding: 10
px; margin-bottom: 10px; border-radius: 5px;'>Host affected: {4}</li>
47     </ul>
48     <p style='font-size: 1.5em; margin-bottom: 20px;'>Consulta los registros desde
<i>Kibana</i> para obtener más información:</p>
49     <li><a href='http://vm4801.virtual.lab.inf.uva.es' style='color: #3498db; text-
decoration: none; font-size: 1.5em;'>http://vm4801.virtual.lab.inf.uva.es</a></li>
50 </div>
51 </body>

```

Listado 6.20: Fichero brute_force_alert_rule.yaml de VM01

En todas las reglas, utilizamos el mismo patrón de índices, **type**, y parámetros de alerta. El servidor SMTP para enviar correos electrónicos es siempre **smtp.gmail.com** con el puerto **587** y sin **SSL**. Las credenciales de autenticación están almacenadas en el archivo **/opt/elastalert/smtp_auth.yaml**. El formato de correo es HTML, y el contenido del correo incluye detalles específicos del incidente, como el **timestamp**, el **nombre** y el **ID** de la regla, la **categoría** del ataque y la **IP** del host afectado. Esto proporciona un informe claro y detallado que se puede consultar fácilmente para tomar acciones de mitigación inmediatas.

Una vez creados todos los ficheros de configuración y despliegue necesarios, para que los contenedores se desplieguen tal y como viene definido en el fichero **docker-compose.yaml** ejecutamos el comando que veremos posteriormente. Cabe destacar que este proceso de despliegue tardará unos minutos.

```
docker compose up -d
```

Si accedemos a la página de Kibana nos aparece un panel de inicio de sesión, donde se introducirán las credenciales de Elastic, como se puede después en la Figura 6.3.

Una vez introducidas las credenciales de Elastic podemos observar en la Figura 6.4 que estamos dentro de la página principal de Kibana y todo funciona correctamente.

Por último, para copiar del contenedor Elasticsearch al host el fichero **ca.crt** que necesitamos pasar a la máquina donde estará instalado Filebeat emplearemos el siguiente comando.

```
docker cp elasticsearch:/usr/share/elasticsearch/config/certs/ca/ca.crt ./ca.crt
```

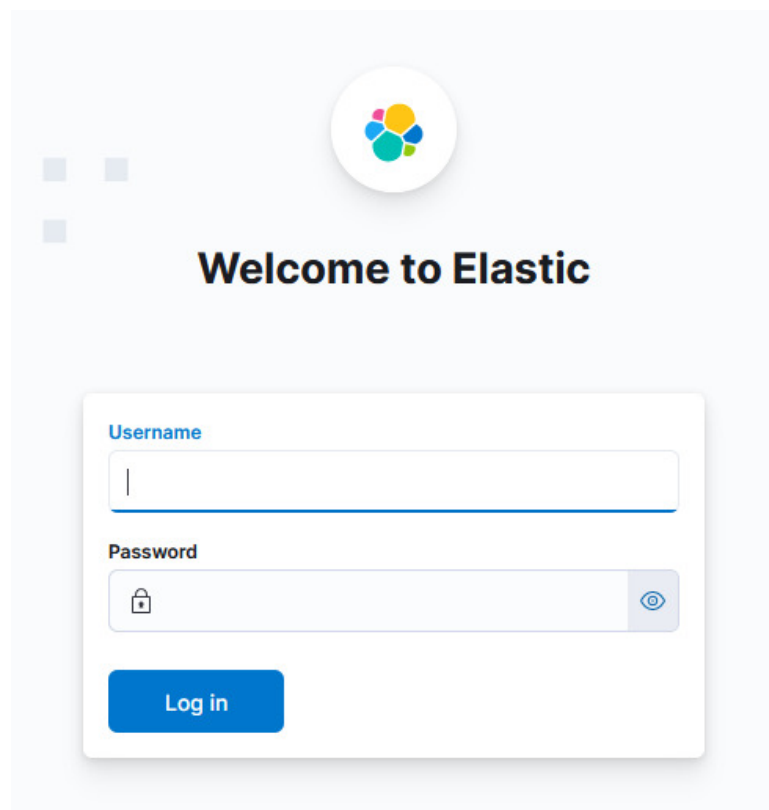


Figura 6.3: Inicio de sesión de la página web de Kibana

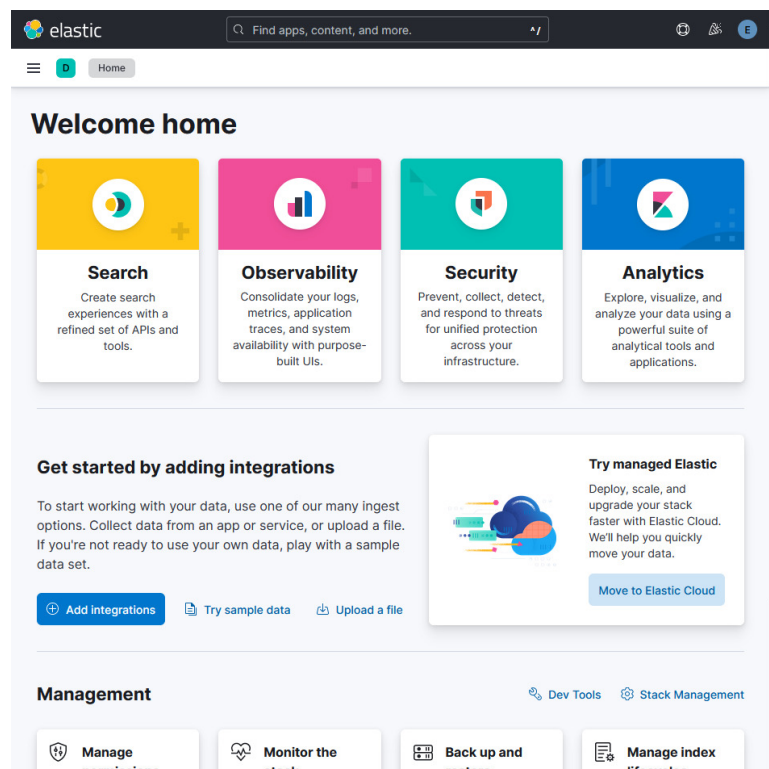


Figura 6.4: Página principal de la web de Kibana

6.2.3 VM02 - Kali atacante

Características VM02

Sistema Operativo	Kali Linux 2024.1
CPUs	2 núcleos
Memoria RAM	8 GiB
Tamaño del disco	24 GB
Número de interfaces de red	2 interfaces

Cuadro 6.9: Características VM02

Instalación de Kali Linux 2024.1

Primero, una vez instalada la versión de **Kali Linux 2024.1** en el laboratorio **PROXMOX** que me ha proporcionado la Universidad de Valladolid, procedemos a actualizar la máquina virtual con los comandos siguientes.

```
sudo apt update
sudo apt upgrade
```

Configuración de red

A continuación, para establecer la configuración de red editamos el fichero de configuración de red `/etc/network/interfaces` (Listado 6.21). En él configuraremos dos interfaces de red, la primera **eth0**, la cual es la interfaz que se conecta a la red Lab48 del laboratorio y tendrá una IP asignada por DHCP. La segunda interfaz **eth1** es la que está conectada a la red interna del laboratorio cuya IP estática es **10.2.2.2/24**.

```
1 auto lo
2 iface lo inet loopback
3
4 # RLAB - VLAN 799 - IP Lab48 por DHCP
5 auto eth0
6 iface eth0 inet dhcp
7
8 # RINT - VLAN 798 - Red interna
9 auto eth1
10 iface eth1 inet static
11     address 10.2.2.2
12     netmask 255.255.255.0
```

Listado 6.21: Fichero de configuración de red `/etc/network/interfaces` de VM02

Después de guardar la configuración de red reiniciamos el servicio de red para aplicar correctamente los cambios realizados en el equipo con el comando siguiente.

```
sudo systemctl restart networking.service
```

Por lo tanto, la configuración de red de la máquina virtual quedaría como se muestra a continuación.

```
root@vm03:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
   1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
   qlen 1000
   link/ether 08:00:27:09:48:02 brd ff:ff:ff:ff:ff:ff
   altname enp0s18
```

```

inet 10.0.48.2/18 brd 10.0.63.255 scope global dynamic ens18
    valid_lft 1304sec preferred_lft 1304sec
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether 08:00:27:98:48:02 brd ff:ff:ff:ff:ff:ff
    altname
    inet 10.2.2.3/24 brd 10.2.2.255 scope global ens19
        valid_lft forever preferred_lft forever

```

6.2.4 VM03 - Servidor web víctima

Características VM03

Sistema Operativo	Ubuntu server 22.04.4 LTS
CPUs	2 núcleos
Memoria RAM	4 GiB
Tamaño del disco	16 GB
Número de interfaces de red	2 interfaces

Cuadro 6.10: Características VM03

Instalación de Ubuntu server 22.04.4 LTS

Primero, una vez instalada la versión de **Ubuntu server 22.04.4 LTS** en el laboratorio **PROXMOX** que me ha proporcionado la Universidad de Valladolid, procedemos a actualizar la máquina virtual con los comandos siguientes.

```

sudo apt update
sudo apt upgrade

```

Configuración de red

A continuación, para establecer la configuración de red editamos el fichero de configuración de red `/etc/network/interfaces` (Listado 6.22). En él configuraremos dos interfaces de red, la primera **ens18**, la cual es la interfaz que se conecta a la red Lab48 del laboratorio y tendrá una IP asignada por DHCP, la cual es **10.0.48.3/18**. La segunda interfaz **ens19** es la que está conectada a la red interna del laboratorio cuya IP estática es **10.2.2.3/24**.

```

1 auto lo
2 iface lo inet loopback
3
4 # RLAB - VLAN 799 - IP Lab48 por DHCP
5 auto ens18
6 iface ens18 inet dhcp
7
8 # RINT - VLAN 798 - Red interna
9 auto ens19
10 iface ens19 inet static
11     address 10.2.2.3
12     netmask 255.255.255.0

```

Listado 6.22: Fichero de configuración de red `/etc/network/interfaces` de VM03

Después de guardar la configuración de red reiniciamos el servicio de red para aplicar correctamente los cambios realizados en el equipo con el comando que se muestra a continuación.

```
sudo systemctl restart networking.service
```

Por lo tanto, la configuración de red de la máquina virtual quedaría de la siguiente manera.

```
root@vm03:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
  1000
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
  link/ether 08:00:27:09:48:03 brd ff:ff:ff:ff:ff:ff
  altname enp0s18
  inet 10.0.48.3/18 brd 10.0.63.255 scope global dynamic ens18
    valid_lft 1304sec preferred_lft 1304sec
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group
  default qlen 1000
  link/ether 08:00:27:98:48:03 brd ff:ff:ff:ff:ff:ff
  altname enp0s19
  inet 10.2.2.3/24 brd 10.2.2.255 scope global ens19
    valid_lft forever preferred_lft forever
```

Instalar Docker

Instalamos Docker en el equipo con el comando mostrado a continuación.

```
sudo apt install docker.io
```

Posteriormente comprobaremos, tal y como se puede ver posteriormente se ha instalado correctamente en su versión **24.0.5**.

```
root@vm01:~# docker -v
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
```

Por último, añadimos al usuario usuario al grupo de Docker, para que este pueda usar Docker sin problemas. Esto se hace modificando línea **docker:x:120:usuario** en el fichero de configuración **/etc/group**.

Para aplicar los cambios en Docker reiniciamos los servicios de Docker para evitar conflictos.

```
sudo systemctl restart docker.service docker.socket
```


Despliegue de Servidor web vulnerable (DVWA)

Para empezar a desplegar el servidor web vulnerable con Docker utilizamos el comando que se muestra después, el cual ejecuta un contenedor Docker usando la imagen **vulnerables/web-dvwa**, que contiene la aplicación web **DVWA (Damn Vulnerable Web Application)**, diseñada para pruebas de seguridad. La opción **-rm** asegura que el contenedor se elimine automáticamente al detenerse, **-it** permite la interacción a través de una terminal, y **-p 80:80** mapea el puerto 80 del contenedor al puerto 80 del host.

```
docker run --rm -it -p 80:80 vulnerables/web-dvwa
```

Para acceder a la aplicación desde una máquina de la misma red, en este caso la máquina Kali Linux, ponemos en el navegador la url **http://10.2.2.3**.

Nos saldrá un panel de login para acceder a la aplicación web introducimos el usuario **admin** y la contraseña **password** tal y como se puede apreciar en la Figura 6.5.



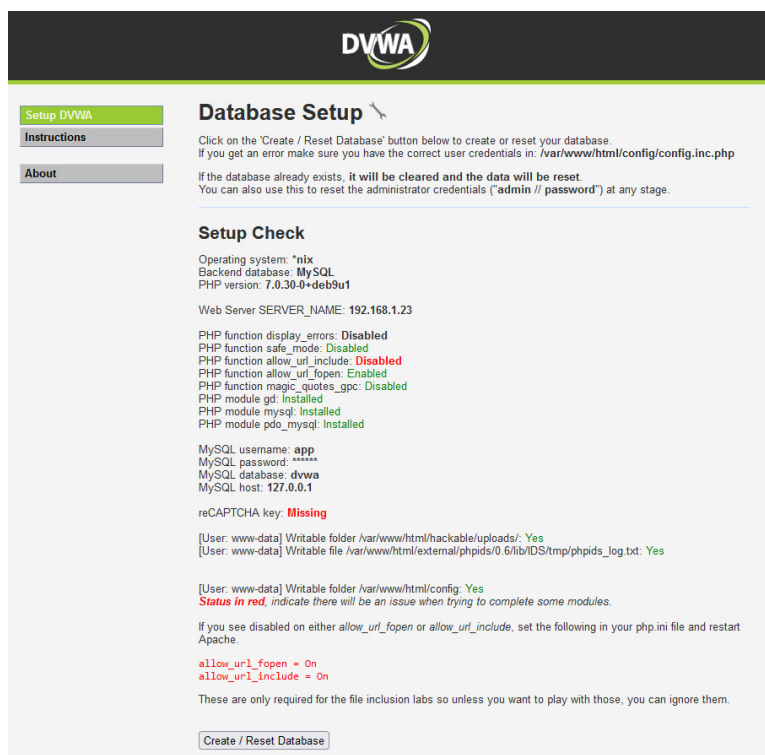
Username

Password

Login

Figura 6.5: Inicio de sesión de la página web DVWA

Entramos en la web y seleccionamos el botón que aparece abajo del todo **Create / Reset Database** para inicializar la base de datos, como se puede observar en la Figura 6.6.

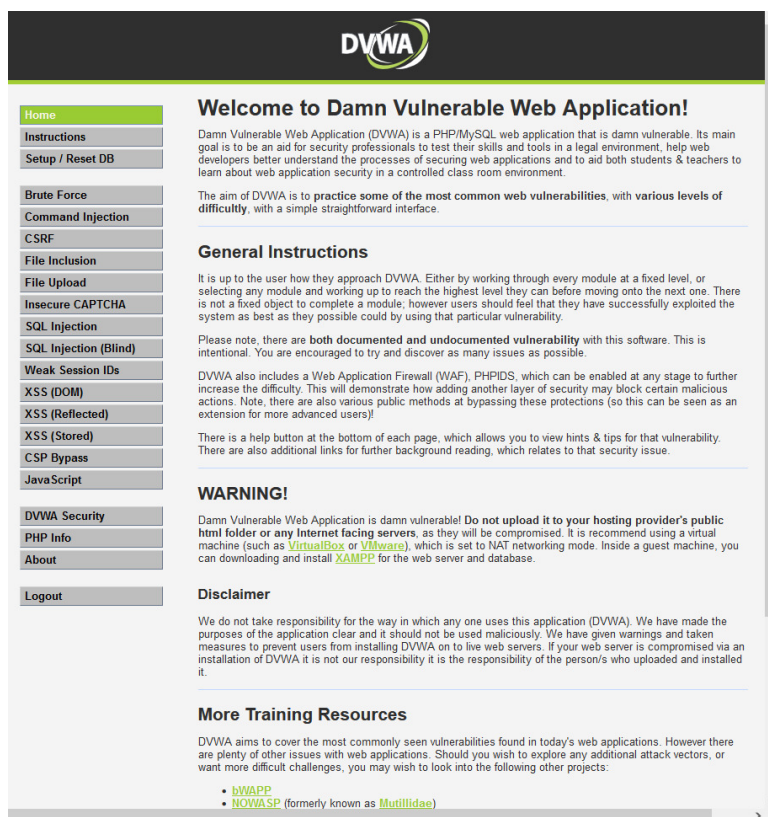


The screenshot shows the DVWA 'Database Setup' page. It includes a sidebar with 'Setup DVWA', 'Instructions', and 'About'. The main content area has a 'Database Setup' heading, instructions on how to create or reset the database, and a 'Setup Check' section. The 'Setup Check' lists system details like OS (nix), database (MySQL), PHP version (7.0.30-0+deb9u1), and server name (192.168.1.23). It also shows the status of various PHP functions and modules, with some like 'display_errors' and 'allow_url_include' being disabled. At the bottom, there is a 'Create / Reset Database' button.

Figura 6.6: Setup de la página web DVWA

Una vez pulsado el botón nos sacará de la web y nos volverá a pedir las credenciales anteriores en el panel de login como en la Figura 6.5. Una vez hecho el inicio de sesión ya estaríamos en la página principal de la web DVWA, tal y como se puede ver en la Figura 6.7.

Por último, hay que destacar que todas la pruebas del sistema en este proyecto se harán con el parámetro **security=low** en la página web DVWA, que es como viene por defecto.



DVWA

Welcome to Damn Vulnerable Web Application!

Damn Vulnerable Web Application (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goal is to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and to aid both students & teachers to learn about web application security in a controlled class room environment.

The aim of DVWA is to practice some of the most common web vulnerabilities, with various levels of difficulty, with a simple straightforward interface.

General Instructions

It is up to the user how they approach DVWA. Either by working through every module at a fixed level, or selecting any module and working up to reach the highest level they can before moving onto the next one. There is not a fixed object to complete a module; however users should feel that they have successfully exploited the system as best as they possible could by using that particular vulnerability.

Please note, there are both documented and undocumented vulnerability with this software. This is intentional. You are encouraged to try and discover as many issues as possible.

DVWA also includes a Web Application Firewall (WAF), PHPIDS, which can be enabled at any stage to further increase the difficulty. This will demonstrate how adding another layer of security may block certain malicious actions. Note, there are also various public methods at bypassing these protections (so this can be seen as an extension for more advanced users)!

There is a help button at the bottom of each page, which allows you to view hints & tips for that vulnerability. There are also additional links for further background reading, which relates to that security issue.

WARNING!

Damn Vulnerable Web Application is damn vulnerable! Do not upload it to your hosting provider's public html folder or any Internet facing servers, as they will be compromised. It is recommend using a virtual machine (such as [VirtualBox](#) or [VMware](#)), which is set to NAT networking mode. Inside a guest machine, you can downloading and install [XAMPP](#) for the web server and database.

Disclaimer

We do not take responsibility for the way in which any one uses this application (DVWA). We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

More Training Resources

DVWA aims to cover the most commonly seen vulnerabilities found in today's web applications. However there are plenty of other issues with web applications. Should you wish to explore any additional attack vectors, or want more difficult challenges, you may wish to look into the following other projects:

- [bWAPP](#)
- [NOWASP](#) (formerly known as [Mutillidae](#))

Figura 6.7: Página principal de la página web DVWA

Pruebas del sistema

En este capítulo se detallarán las pruebas realizadas para evaluar la eficacia y precisión del sistema de alerta temprana implementado utilizando Suricata, Elasticsearch, Kibana y ElastAlert2. El objetivo principal de estas pruebas es verificar que el sistema sea capaz de detectar, en tiempo real, una variedad de ciberataques comunes y generar las alertas por correo correspondientes de manera precisa y oportuna.

El sistema de alerta temprana está diseñado para identificar y notificar sobre actividades maliciosas en un entorno de red, proporcionando una defensa proactiva contra posibles intrusiones. Suricata, una herramienta de monitoreo de red y detección de intrusiones, se encarga de inspeccionar el tráfico de red y generar alertas basadas en patrones específicos de ataques. Estas alertas se envían a Elasticsearch, una potente base de datos de búsqueda y análisis, donde se almacenan y se indexan para su posterior análisis. Kibana, una plataforma de visualización de datos, se utiliza para visualizar y explorar estas alertas de manera intuitiva. Finalmente, ElastAlert2 se encarga de procesar las alertas en Elasticsearch y enviar notificaciones por correo electrónico a los administradores del sistema.

7.1 Simulación ataque SQLI

Se simulará un ataque de inyección SQL (SQL Injection), donde se intentará inyectar código SQL malicioso en las consultas a la base de datos del servidor web. Este tipo de ataque puede resultar en la manipulación o destrucción de datos.

Ahora que ya sabemos en que consiste un ataque SQLI, vamos a simularlo. Para empezar desde la máquina virtual Kali (VM02) accedemos a la página web vulnerable DVWA poniendo en el navegador la URL **http://10.2.2.3**. Nos registramos con las credenciales de inicio de sesión del laboratorio de ciberseguridad DVWA (usuario:admin y contraseña:password), como se ve en la Figura 6.5.

Una vez dentro de la web DVWA vamos a la sección de **SQL Injection** y en el campo que está pensado para buscar a un usuario según un ID introduciremos la consulta maliciosa **1' OR '1'='1'#**, como se ve en la Figura 7.1.

Después, como observamos en la Figura 7.2 el ataque SQLI ha sido exitoso, debido a que hemos conseguido que se nos muestren los datos de todos los usuarios de la base de datos con la consulta maliciosa.

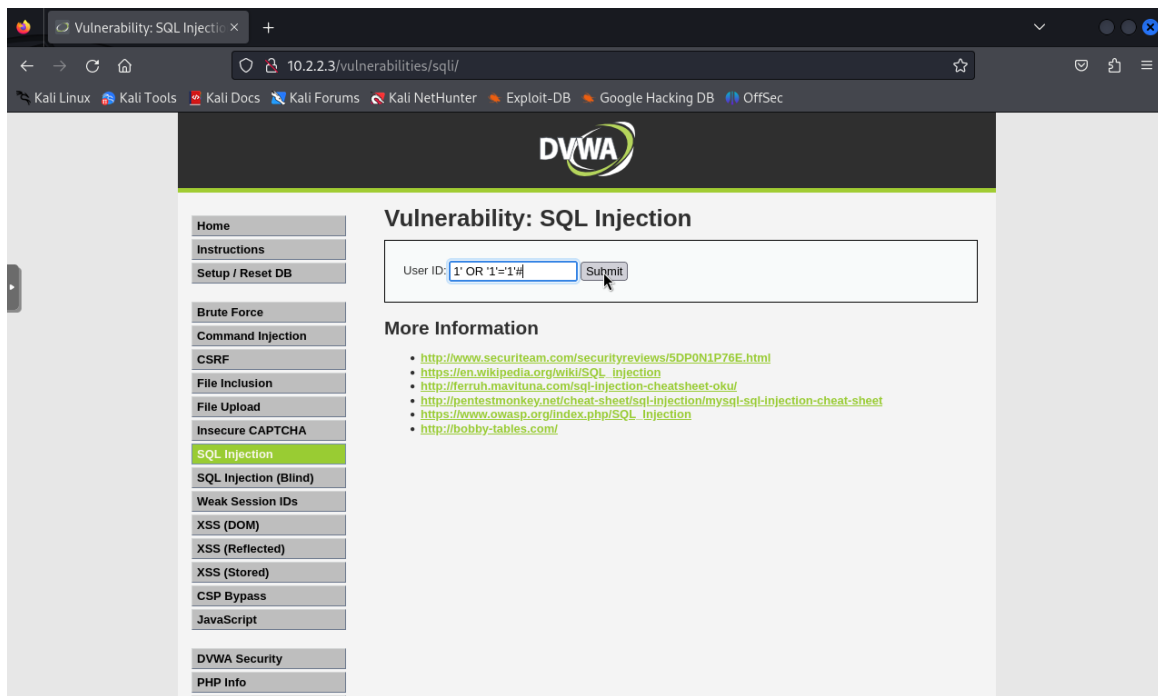


Figura 7.1: Sección de SQL Injection en la web DVWA

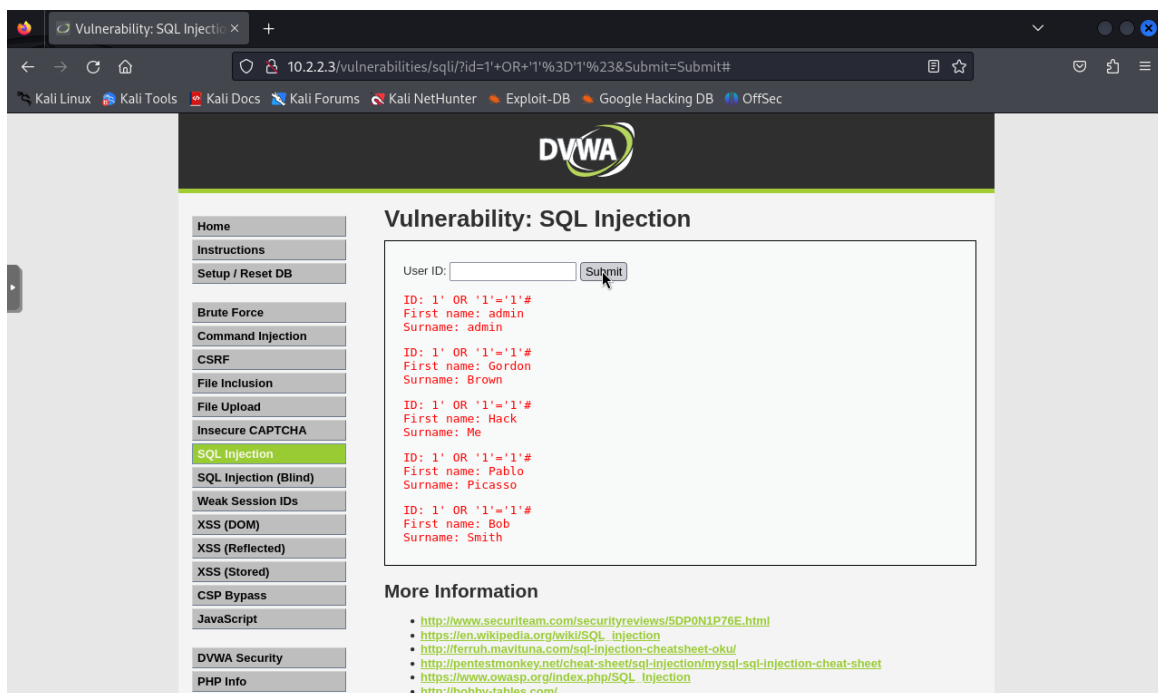


Figura 7.2: Ataque SQL Injection en la web DVWA

Unos pocos segundos después de realizar el ataque SQLI recibimos una alerta en GMAIL (Figura 7.3), donde se ven algunos datos sobre el incidente como por ejemplo cuando se realizó (**14 de junio de 2024 a las 18:09:34 UTC**), el nombre de la regla (**Possible SQL Injection Attempt - Detected 'OR' Pattern in URI**), el SID **100013**, la categoría de la alerta (**Web Application Attack**) y la IP del host afectado (**10.2.2.3**).

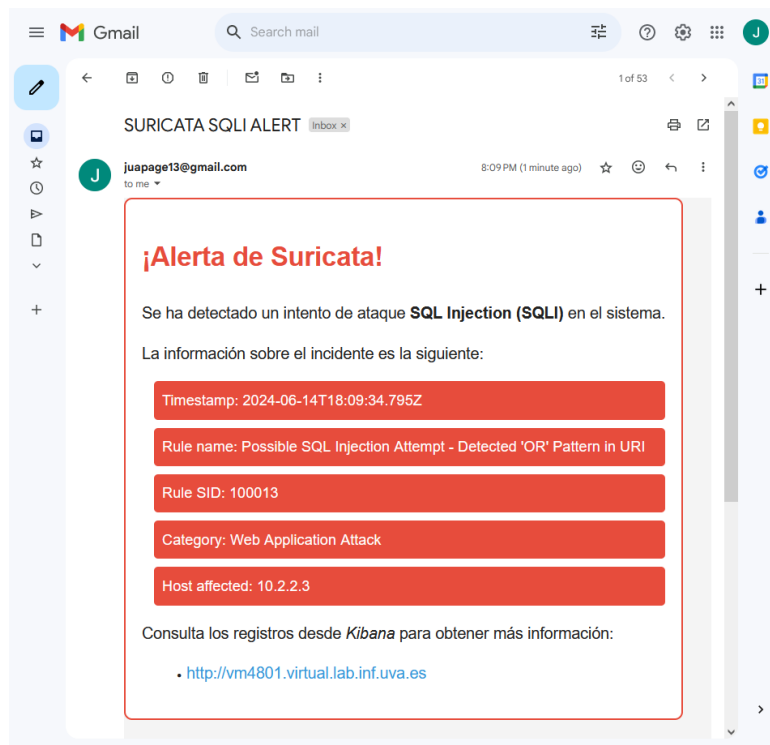


Figura 7.3: Alerta sobre SQLI en GMAIL

Para ver más información podemos pinchar en el enlace que nos lleva a la web de Kibana que está alojada en el puerto 80 de la VM01. Una vez en Kibana podemos ver el **Events Dashboard** de Suricata (Figura 7.4), el cual muestra información más detallada como el número de eventos que se han producido en los últimos 5 minutos (3.979 eventos), el número de alertas que han saltado o los distintos protocolos de red que ha detectado en ese periodo de tiempo, entre otras cosas.

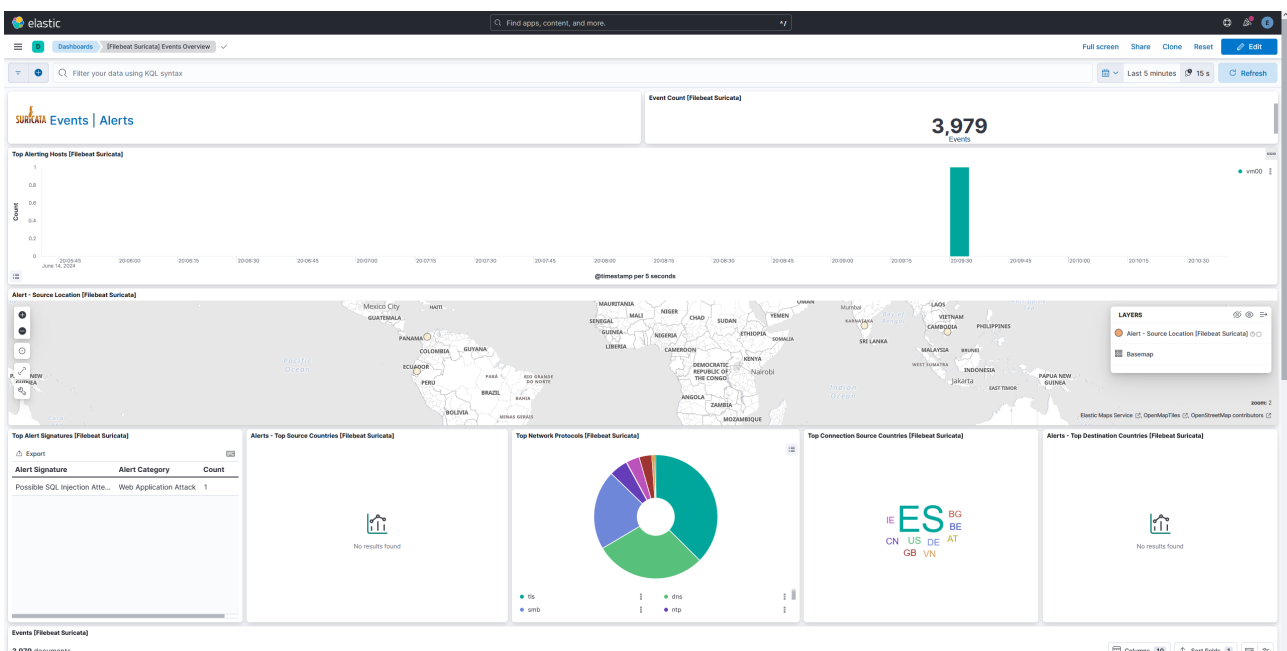


Figura 7.4: Dashboard Suricata Events sobre SQLI en Kibana

Por otro lado, estaría el **Alerts Dashboard** (Figura 7.5), donde observamos que se ha producido 1 alerta de Suricata en los últimos 5 minutos. La alerta salta porque se ha detectado la cadena **OR** en le URI de una solicitud HTTP que fue al servidor Web, lo cual es lo que llevaba la solicitud HTTP que se realizó en el ataque SQLI hecho con anterioridad.

Para terminar, si se quisiera ver en detalle más información sobre el incidente existe la posibilidad de ver el **JSON log** que se envió desde la máquina donde está Suricata, como se aprecia en la Figura 7.6.

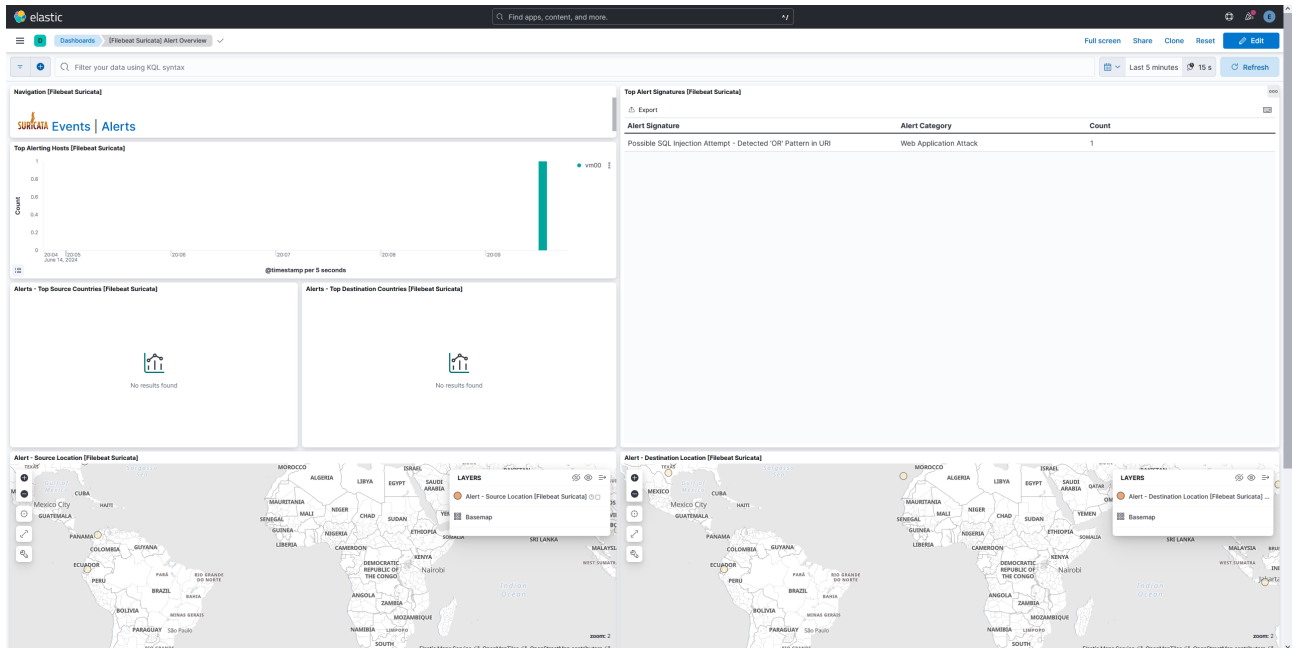


Figura 7.5: Dashboard Suricata Alerts sobre SQLI en Kibana

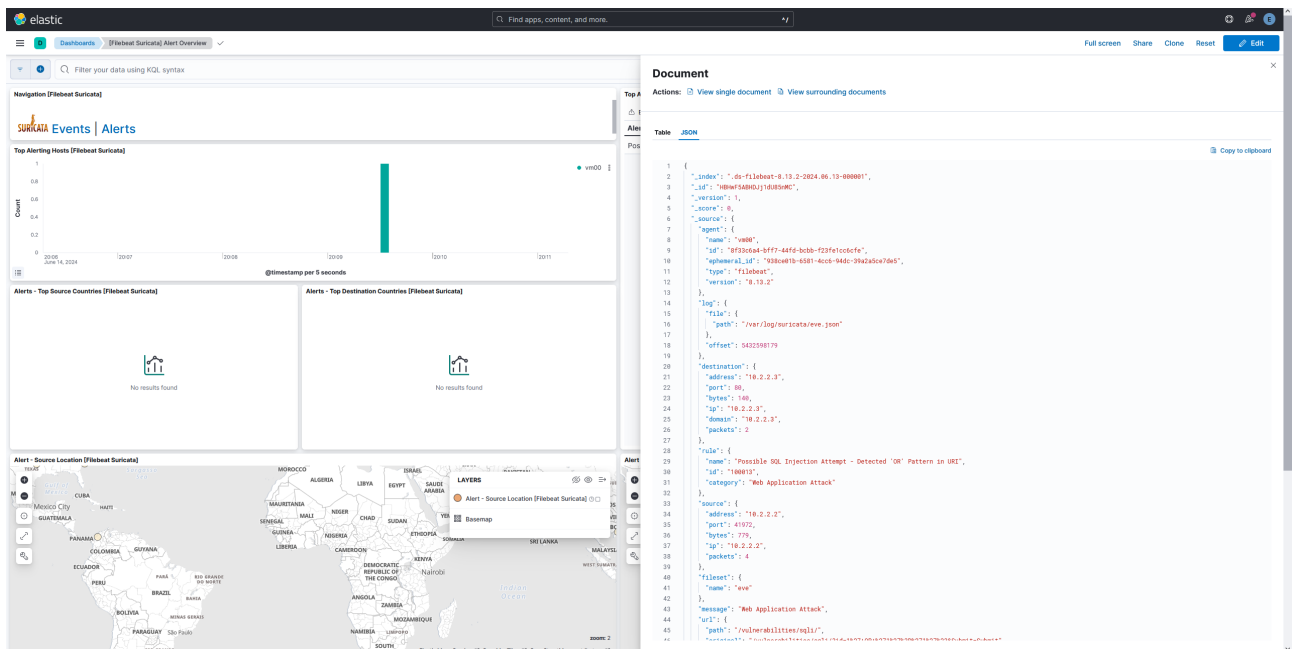


Figura 7.6: JSON Suricata Log sobre SQLI en Kibana

7.2 Simulación ataque LFI

En esta prueba, se llevará a cabo un ataque de inclusión de archivos locales (Local File Inclusion, LFI). Este tipo de ataque permite a un atacante incluir archivos locales en el servidor web, lo que puede conducir a la exposición de información sensible o la ejecución de código arbitrario.

Una vez entendido, lo que es este tipo de ataque vamos a proceder a realizarlo. Primero desde la máquina Kali (VM02) accedemos a la página web vulnerable DVWA poniendo en el navegador la URL <http://10.2.2.3>. Nos registramos con las credenciales de inicio de sesión del laboratorio de ciberseguridad DVWA (usuario:admin y contraseña:password), como se ve en la Figura 6.5.

Acto seguido vamos a la sección de File Inclusion (Figura 7.7), donde podemos ver que es una página donde nos podemos descargar 3 archivos distintos en php.

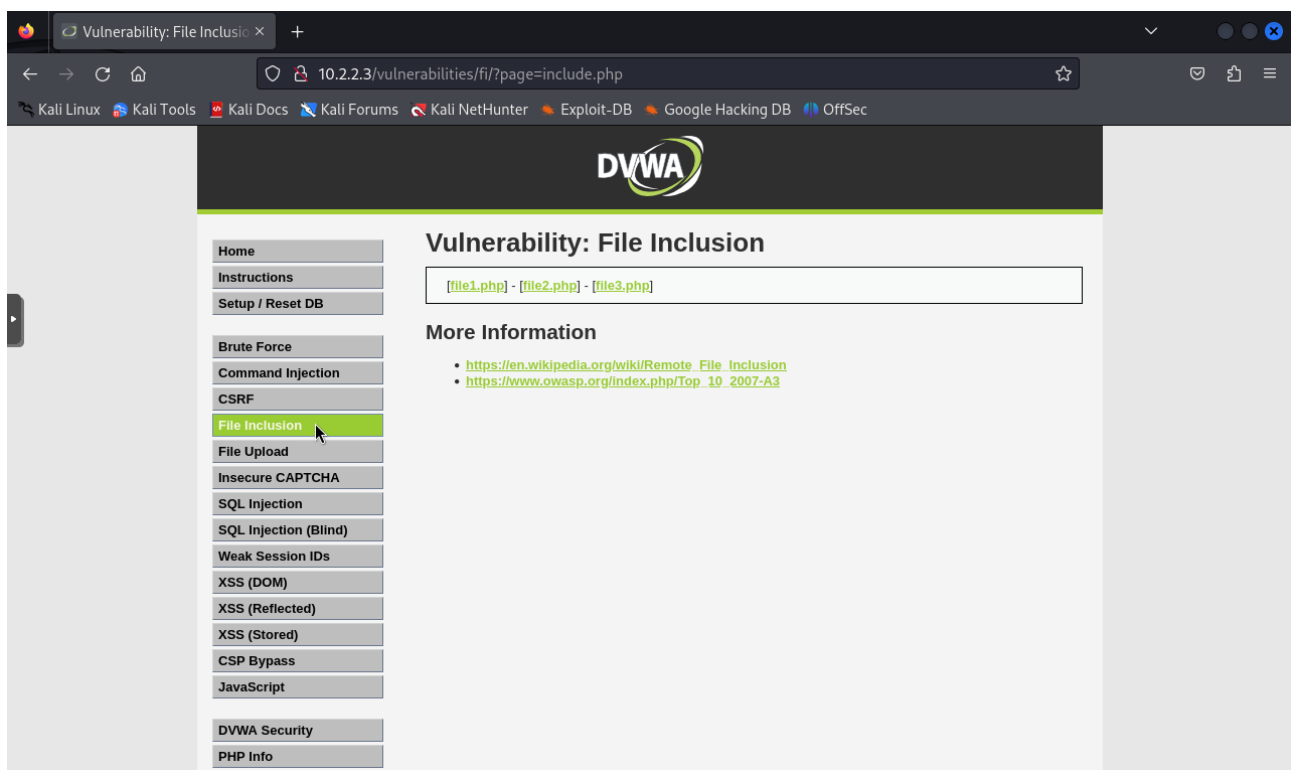


Figura 7.7: Sección de File Inclusion en la web DVWA

Para realizar el ataque LFI cambiamos el parámetro **?page=include.php** por **?page=../../../../etc/passwd** y de esta manera podemos apreciar en la Figura 7.8 que se produce una filtración de información sensible, ya que que nos muestra por pantalla el fichero **/etc/passwd** de la máquina VM03 con sus correspondientes usuarios.

Unos segundos después de haber realizado el ataque LFI sobre la página web DVWA objetivo nos llega una alerta al GMAIL (Figura 7.9), notificando que se ha producido un intento de ataque LFI el día 14 de junio de 2024 a las 17:41:38 (UTC). Además, podemos ver que la regla que ha saltado en Suricata se llama **Possible Local File Inclusion Attempt, ../ or variations in URI** con un SID de **200001**, en la categoría **Web Applications Attack** y la IP del host afectado es la **10.2.2.3**.

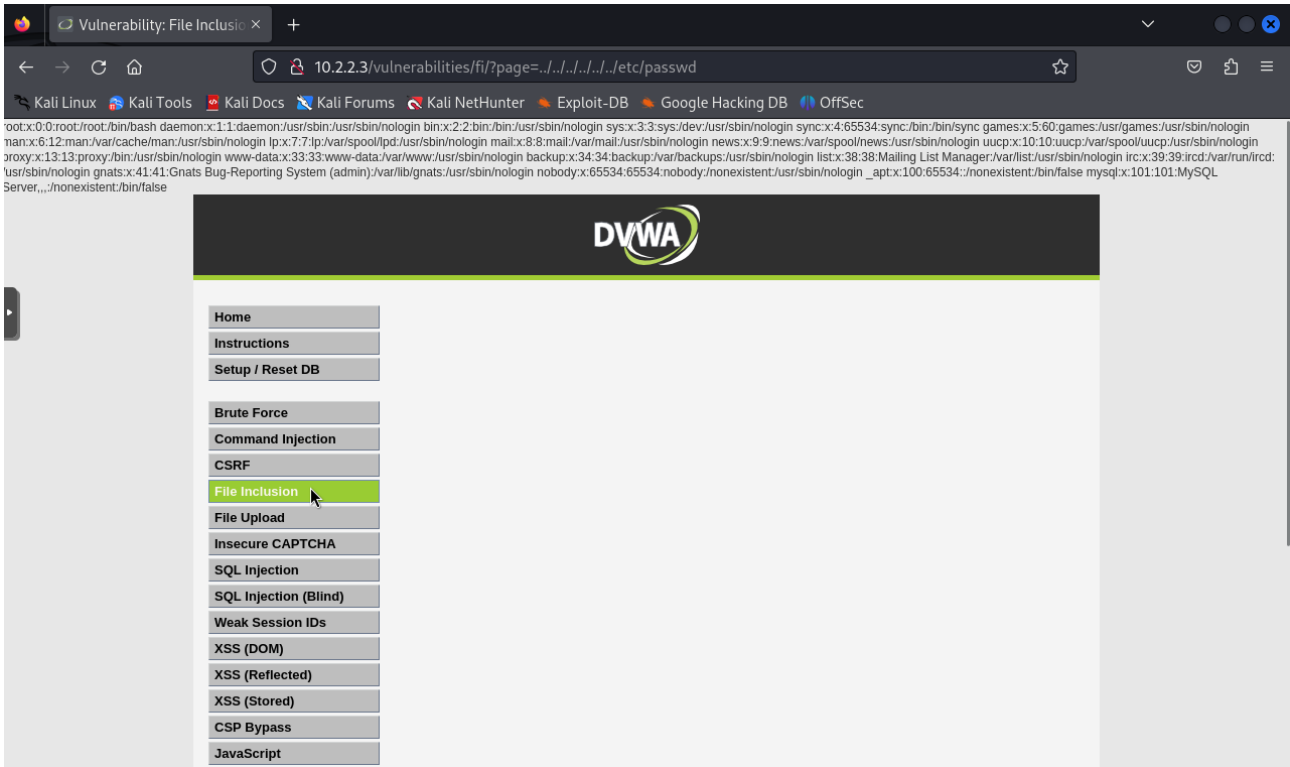


Figura 7.8: Ataque Local File Inclusion en la web DVWA

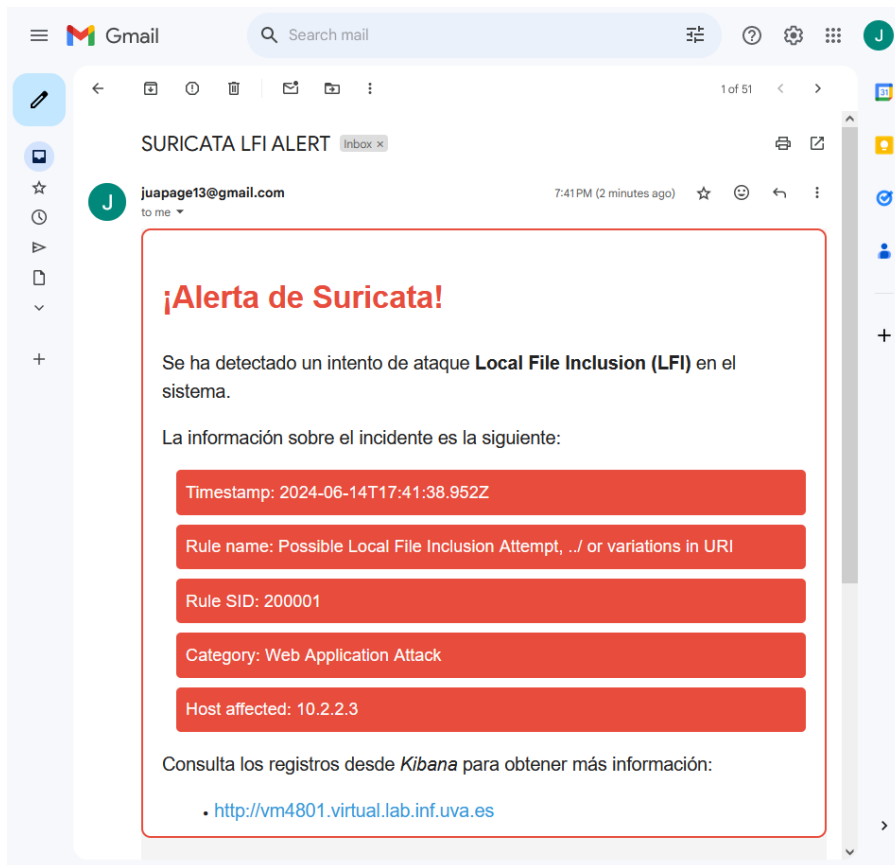


Figura 7.9: Alerta sobre LFI en GMAIL

Para ver más información podemos pinchar en el enlace que nos lleva a la web de Kibana que está alojada en el puerto 80 de la VM01. Una vez en Kibana podemos ver el **Events Dashboard** de Suricata (Figura 7.10), el cual muestra información más detallada como el número de eventos que se han producido en los últimos 5 minutos (2.229 eventos), el número de alertas que han saltado o los distintos protocolos de red que ha detectado en ese periodo de tiempo, entre otros.

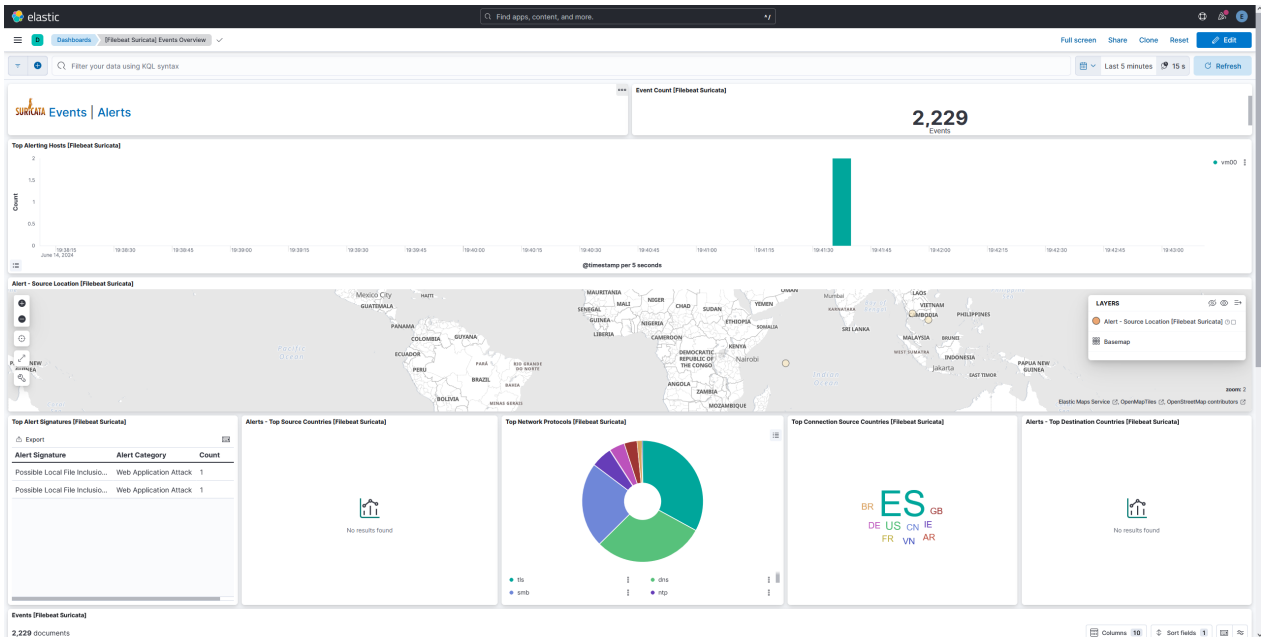


Figura 7.10: Dashboard Suricata Events sobre LFI en Kibana

Por otra parte, estaría el **Alerts Dashboard** (Figura 7.11), donde observamos que se han producido 2 alertas de Suricata en los últimos 5 minutos. La primera salta porque ha detectado `../` en le URI de una solicitud HTTP que fue al servidor Web y la segunda porque ha detectado también la cadena `/etc/passwd` en el URI, lo cual es exactamente lo que llevaba la solicitud HTTP que se realizó en el ataque LFI previo.

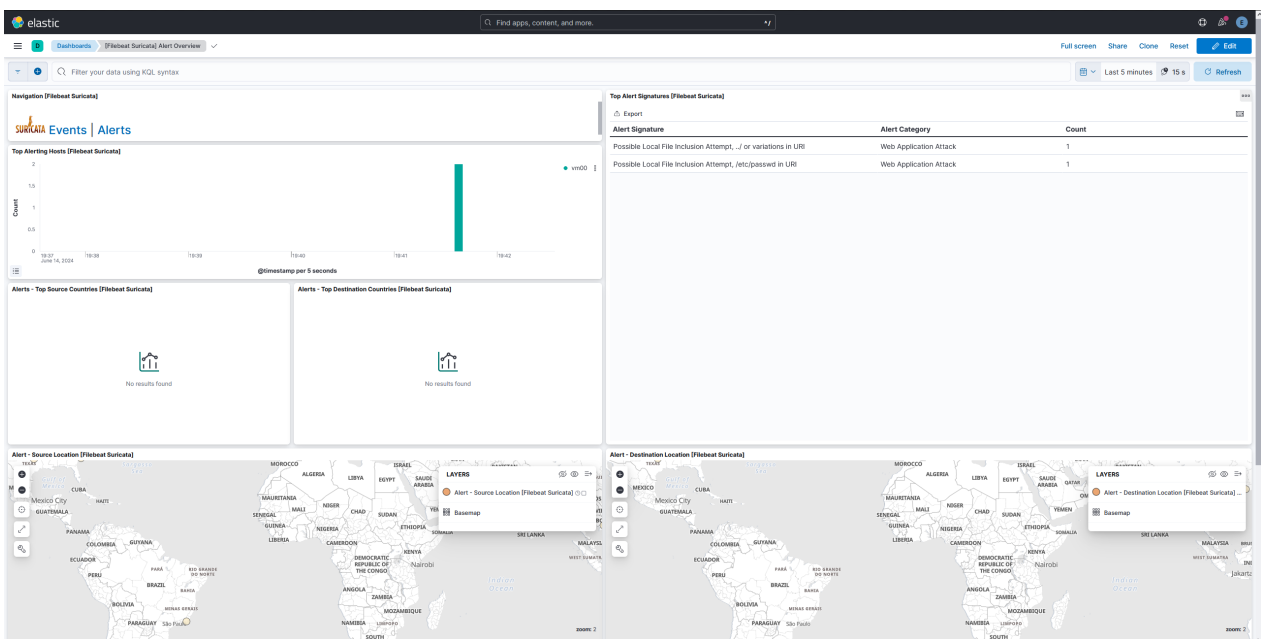


Figura 7.11: Dashboard Suricata Alerts sobre LFI en Kibana

Por último, si se quisiera ver información sobre el incidente más en profundidad existe la posibilidad de ver el **JSON log** que se envió desde la máquina donde está Suricata, tal y como podemos observar en la Figura 7.12.

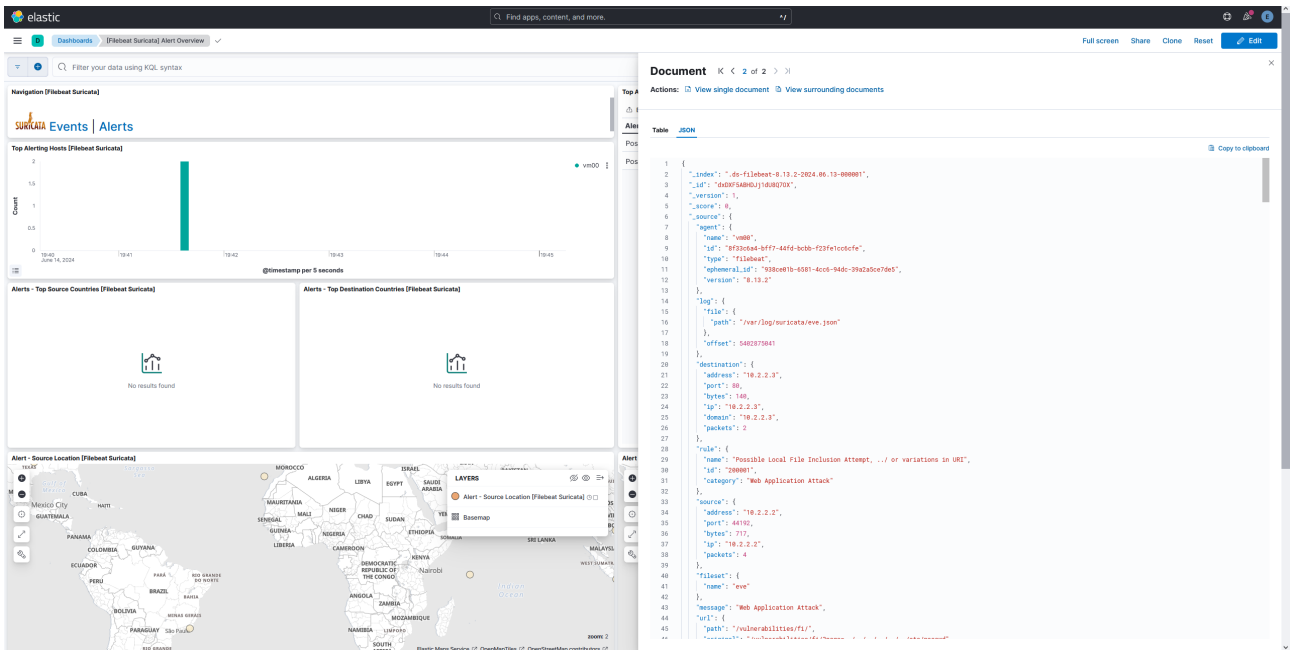


Figura 7.12: JSON Suricata Log sobre LFI en Kibana

7.3 Simulación ataque RFI

Para esta prueba, se realizará un ataque de inclusión de archivos remotos (Remote File Inclusion, RFI). Este tipo de ataque permite a un atacante incluir archivos remotos en el servidor web, lo que puede conducir a la ejecución de código arbitrario en el servidor afectado.

Para llevar a cabo este ataque, primero accedemos a la página web vulnerable DVWA desde la máquina Kali (VM02) ingresando la URL **http://10.2.2.3** en el navegador. Luego, nos registramos utilizando las credenciales de inicio de sesión del laboratorio de ciberseguridad DVWA (usuario:admin y contraseña:password), como se ve en la Figura 6.5.

Una vez registrados, navegamos a la sección de File Inclusion (Figura 7.7), donde encontramos una página que nos permite descargar tres archivos distintos en formato PHP.

Para llevar a cabo el ataque RFI, modificamos el parámetro de la URI **?page=include.php** por **?page=http://www.google.com** y observamos en la Figura 7.13 que se produce una inclusión exitosa del archivo remoto, ya que se muestra el contenido de la página web de Google en la interfaz de DVWA.

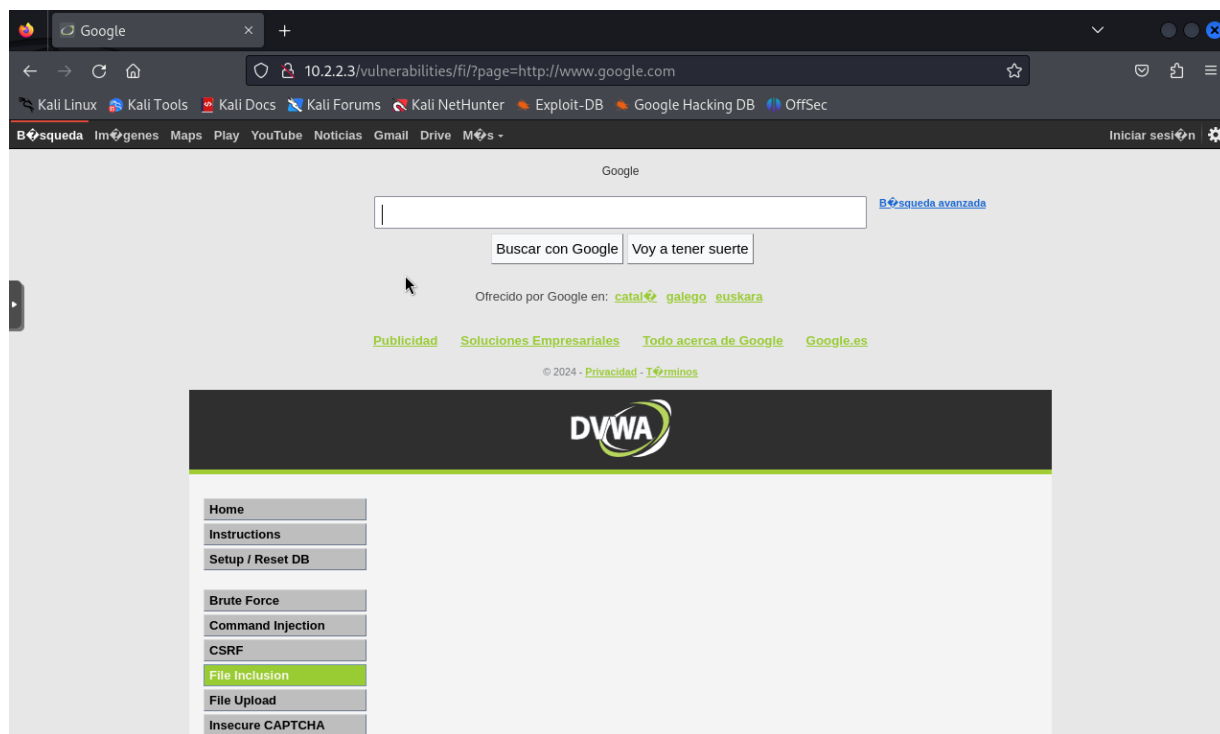


Figura 7.13: Ataque Remote File Inclusion en la web DVWA

Unos instantes después de realizar el ataque RFI sobre la página web DVWA objetivo nos llega una alerta al GMAIL (Figura 7.14), notificando que se ha producido un intento de ataque RFI el día **14 de junio de 2024 a las 17:56:17 (UTC)**. Además, podemos ver que la regla que ha saltado en Suricata se llama **Possible Remote File Inclusion Attempt, http/https in URI** con un SID de **300001**, en la categoría **Web Applications Attack** y la IP del host afectado es la **10.2.2.3**.

Para ver más información podemos pinchar en el enlace que nos lleva a la web de Kibana que está alojada en el puerto 80 de la VM01. Una vez en Kibana podemos ver el **Events Dashboard** de Suricata (Figura 7.15), el cual muestra información más detallada como el número de eventos que se han producido en los últimos 5 minutos (4.643 eventos), el número de alertas que han saltado o los distintos protocolos de red que ha detectado en ese periodo de tiempo, entre otras cosas.

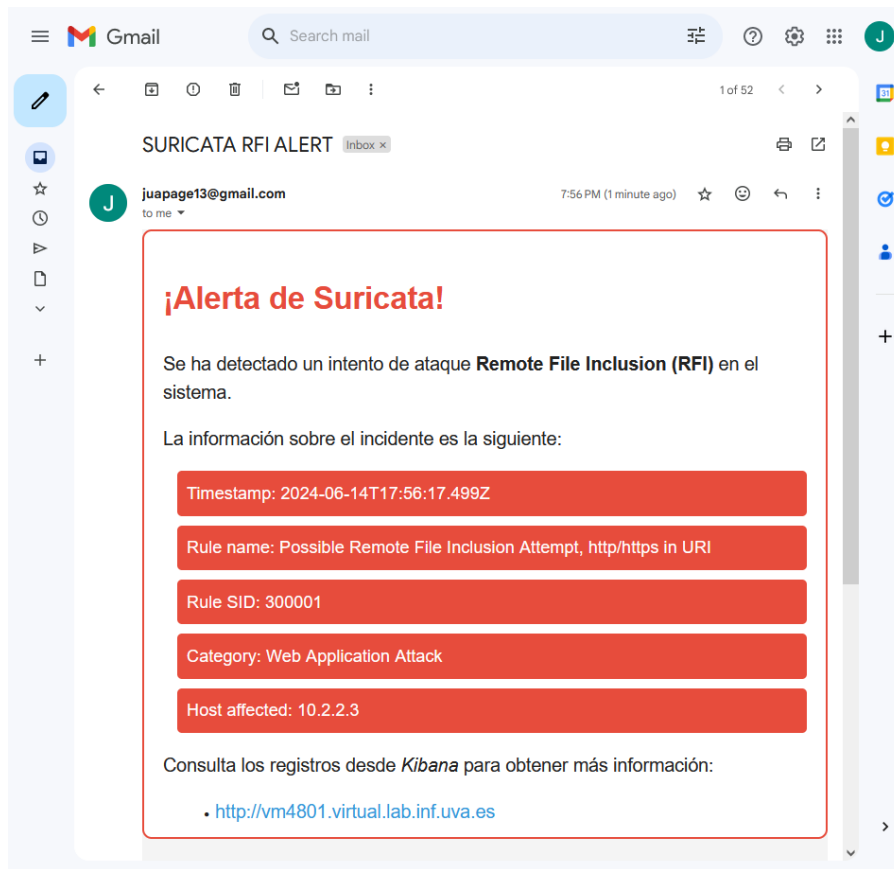


Figura 7.14: Alerta sobre RFI en GMAIL

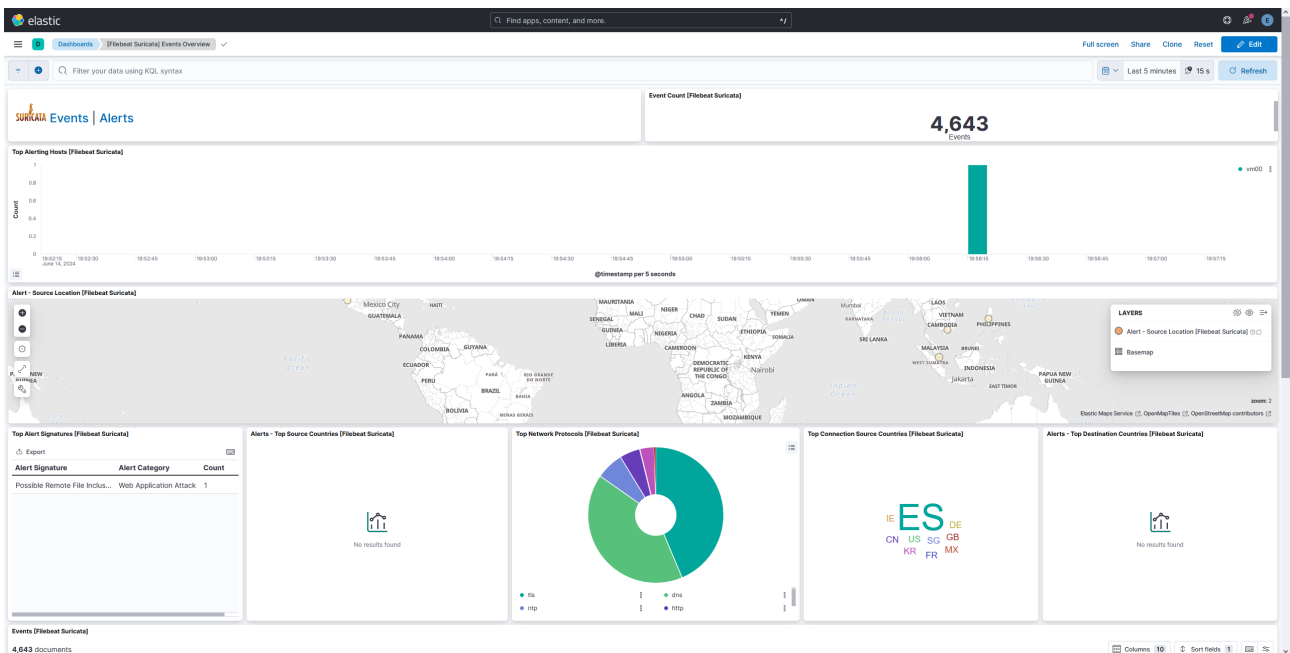


Figura 7.15: Dashboard Suricata Events sobre RFI en Kibana

Por otra parte, estaría el **Alerts Dashboard** (Figura 7.16), donde observamos que se ha producido 1 alerta de Suricata en los últimos 5 minutos. La alerta salta porque se ha detectado la cadena `=http` en le URI de una solicitud HTTP que fue al servidor Web, lo cual es justo lo que llevaba la solicitud HTTP que se realizó en el ataque RFI hecho anteriormente.

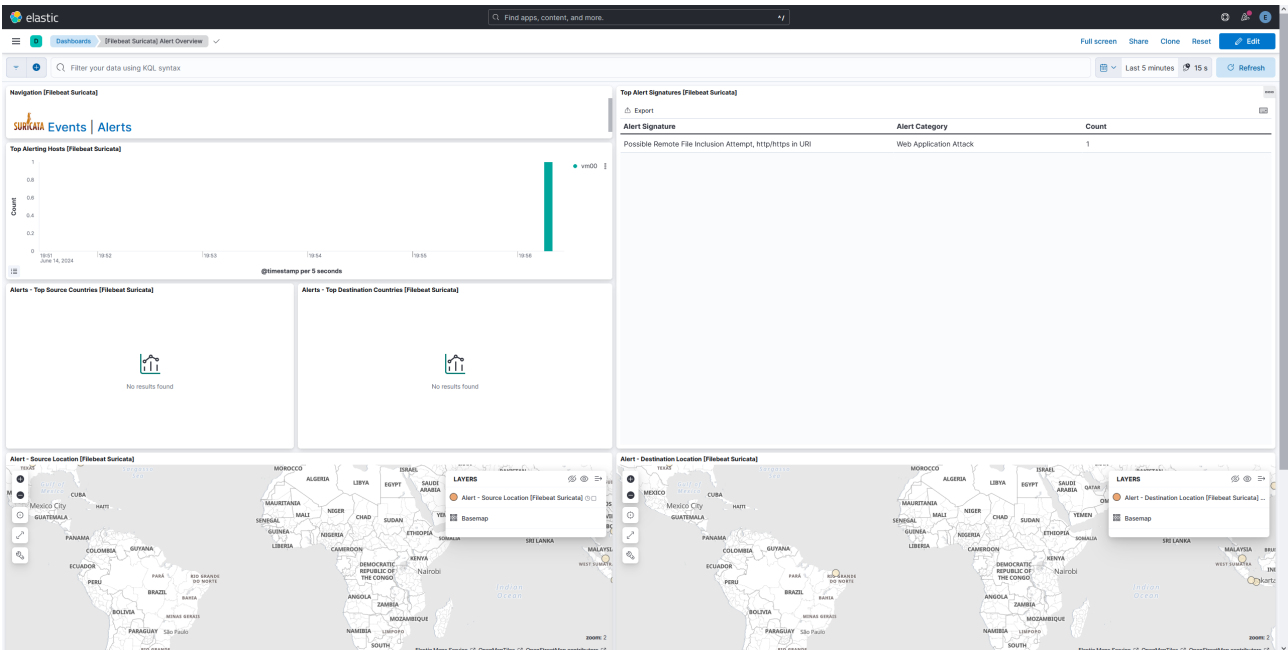


Figura 7.16: Dashboard Suricata Alerts sobre RFI en Kibana

Por último, si se quisiera ver información sobre el incidente más a fondo existe la posibilidad de ver el **JSON log** que se envió desde la máquina donde está Suricata, tal y como podemos observar en la Figura 7.17.

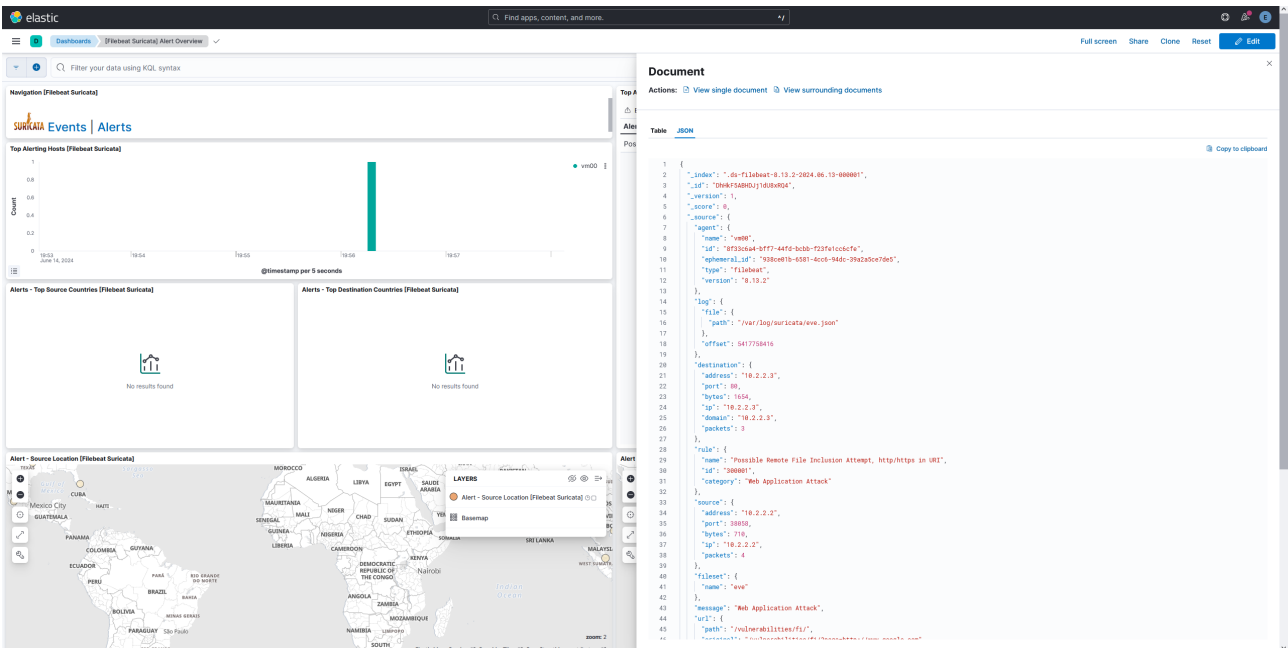


Figura 7.17: JSON Suricata Log sobre RFI en Kibana

7.4 Simulación ataque de fuerza bruta

Se ejecutará un ataque de fuerza bruta, intentando múltiples combinaciones de contraseñas para acceder a la cuenta de un usuario en el servidor web. Este tipo de ataque busca obtener acceso no autorizado a través de la repetición sistemática de contraseñas.

Ahora lo que sabemos que es este tipo de ataque vamos a proceder a realizarlo. Primero desde la máquina Kali (VM02) accedemos a la página web vulnerable DVWA poniendo en el navegador la URL **http://10.2.2.3**. Nos registramos con las credenciales de inicio de sesión del laboratorio de ciberseguridad DVWA (usuario:admin y contraseña:password), como se aprecia en la Figura 6.5.

Justo después, vamos a la sección de Brute Force (Figura 7.18), donde podemos ver que es una página donde nos aparece un panel de login con usuario y contraseña.

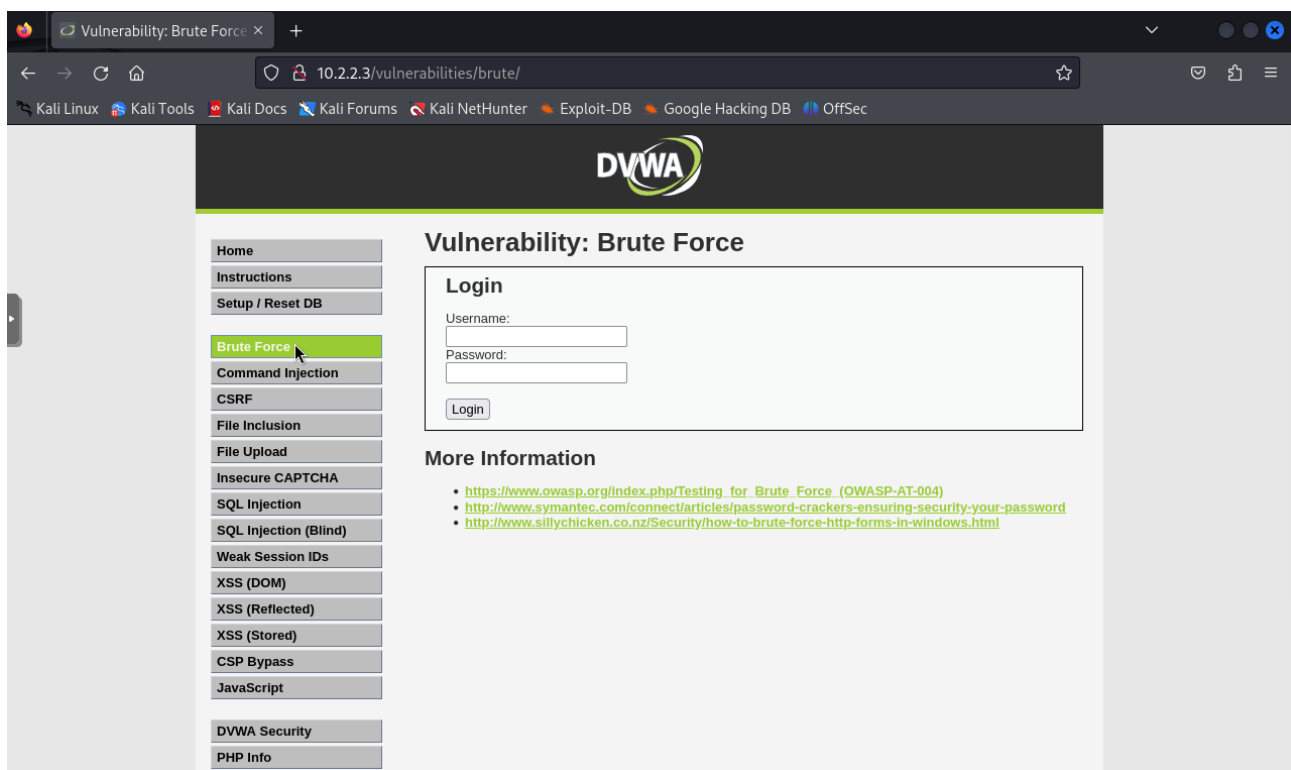


Figura 7.18: Sección de Brute Force en la web DVWA

Para poder realizar este ataque de fuerza bruta sobre el servidor web abriremos la herramienta Burpsuite para interceptar la solicitud HTTP y ver que parámetros son los que se están enviando. Dentro de Burpsuite iremos a **Proxy Settings** y como se puede ver en la Figura 7.19 el Proxy escuchará en 127.0.0.1 por el puerto 8080.

Después, nos vamos a las opciones de configuración del navegador Firefox y en **General>Network Settings** seleccionamos **Manual proxy configuration** e introducimos los mismos parámetros que hemos visto en Burpsuite (Figura 7.19). La configuración de Firefox quedaría como en la Figura 7.20.

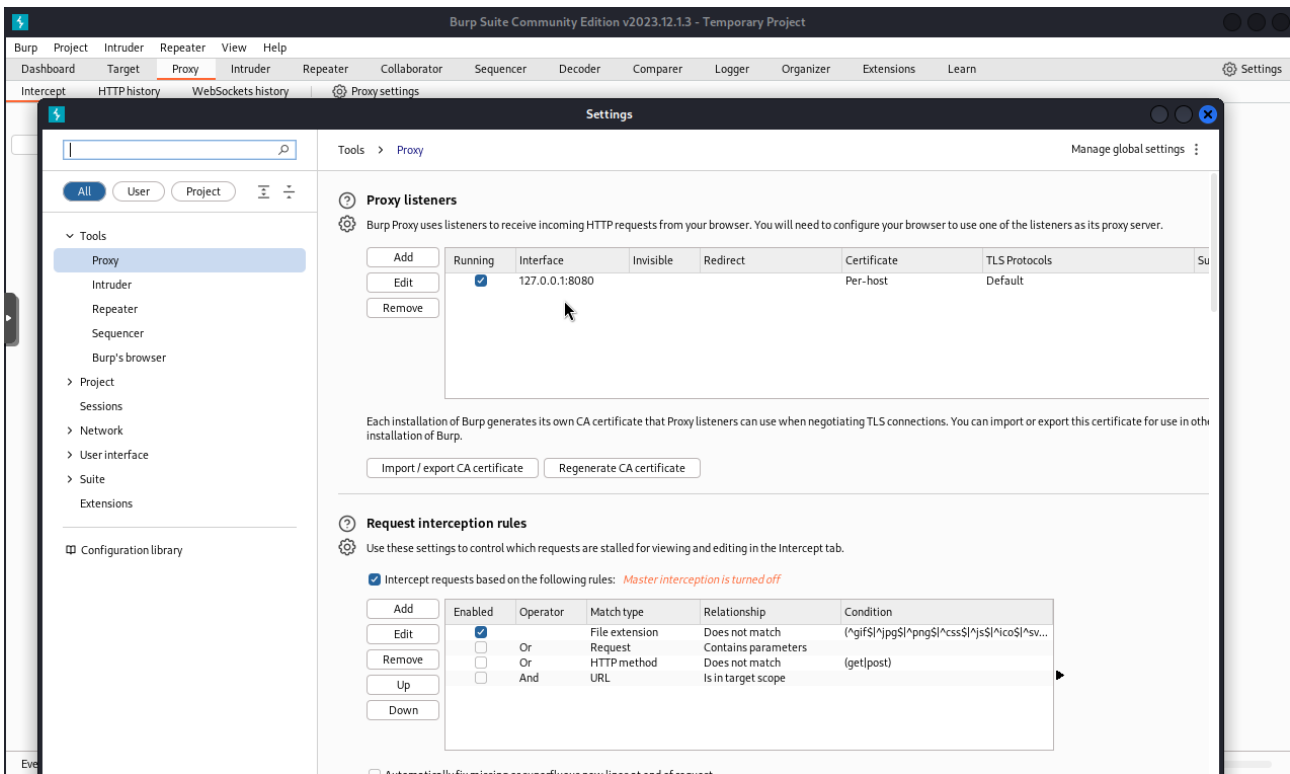


Figura 7.19: Configuración Proxy para Burpsuite

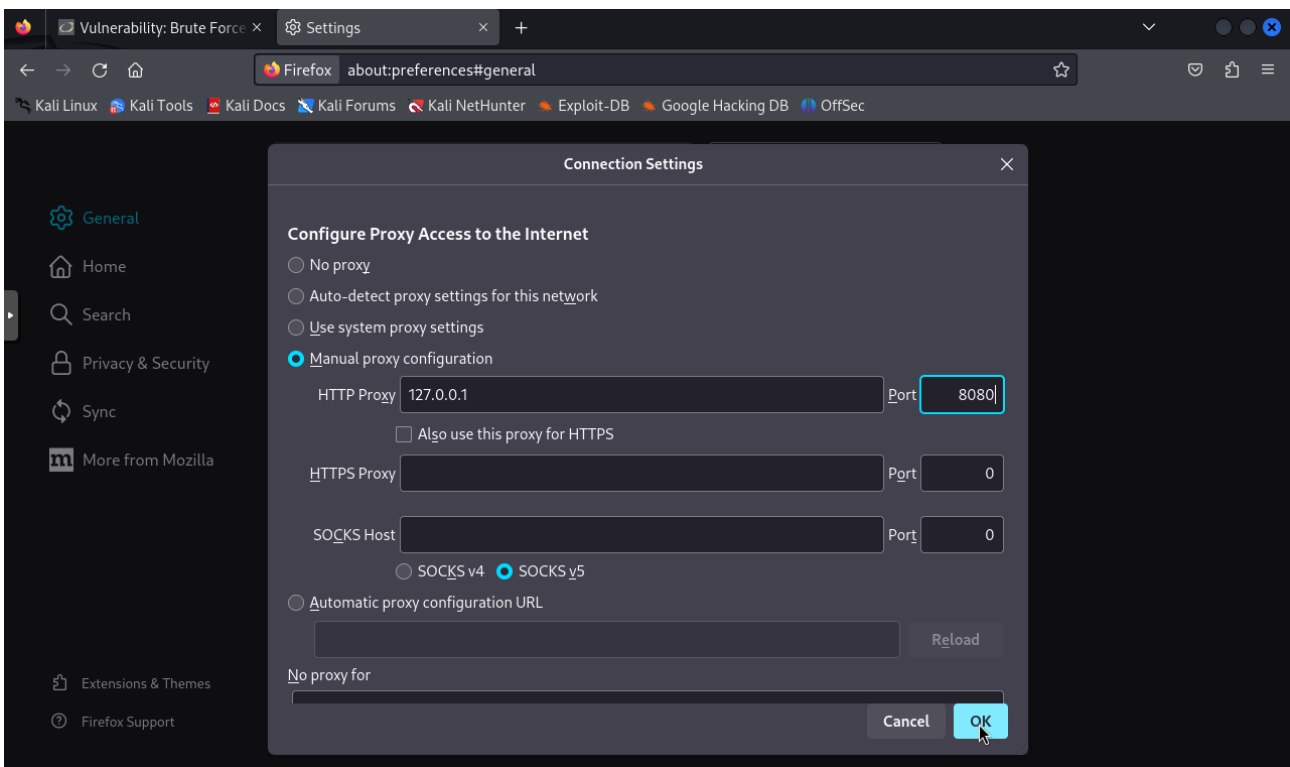


Figura 7.20: Configuración Manual Proxy para Firefox

A continuación, volvemos a Burpsuite y comenzamos a interceptar las solicitudes HTTP que se realicen en Firefox (Figura 7.21).

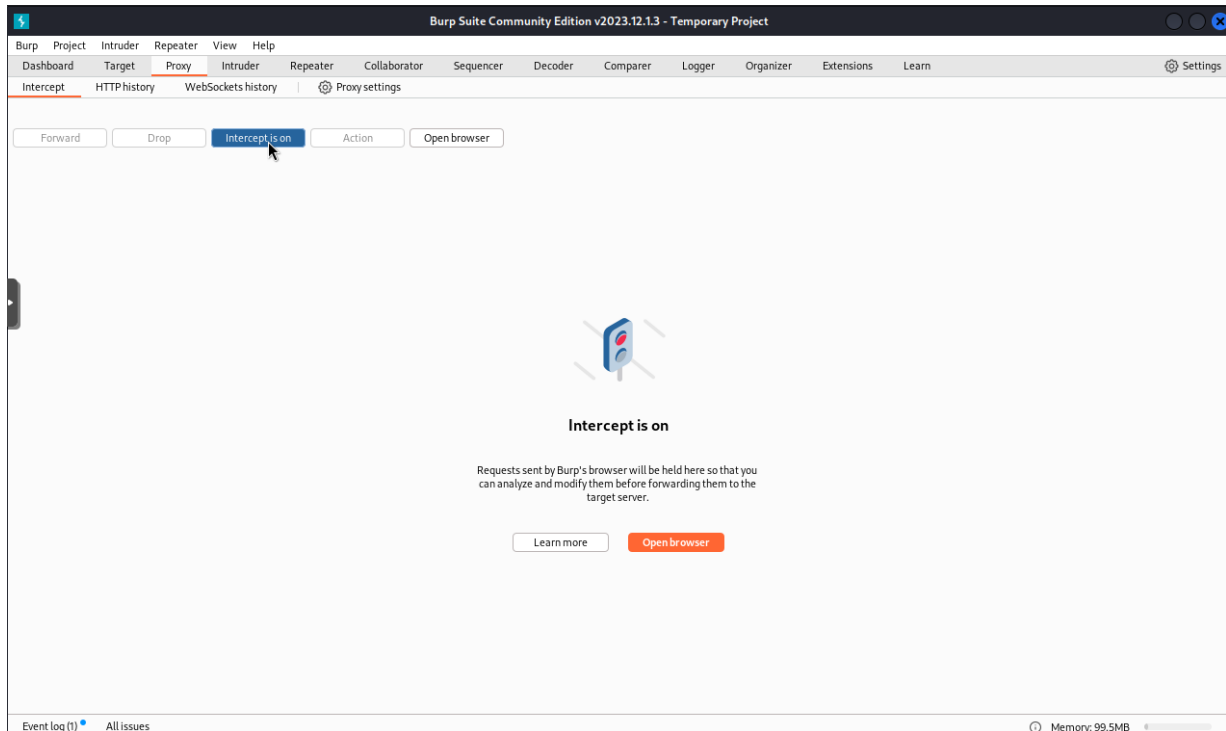


Figura 7.21: Activar interceptación de solicitudes HTTP con Burpsuite

Volvemos a la página web DVWA dentro de su sección Brute force y tal y como se observa en la Figura 7.22 enviamos una solicitud con los parámetros vacíos.

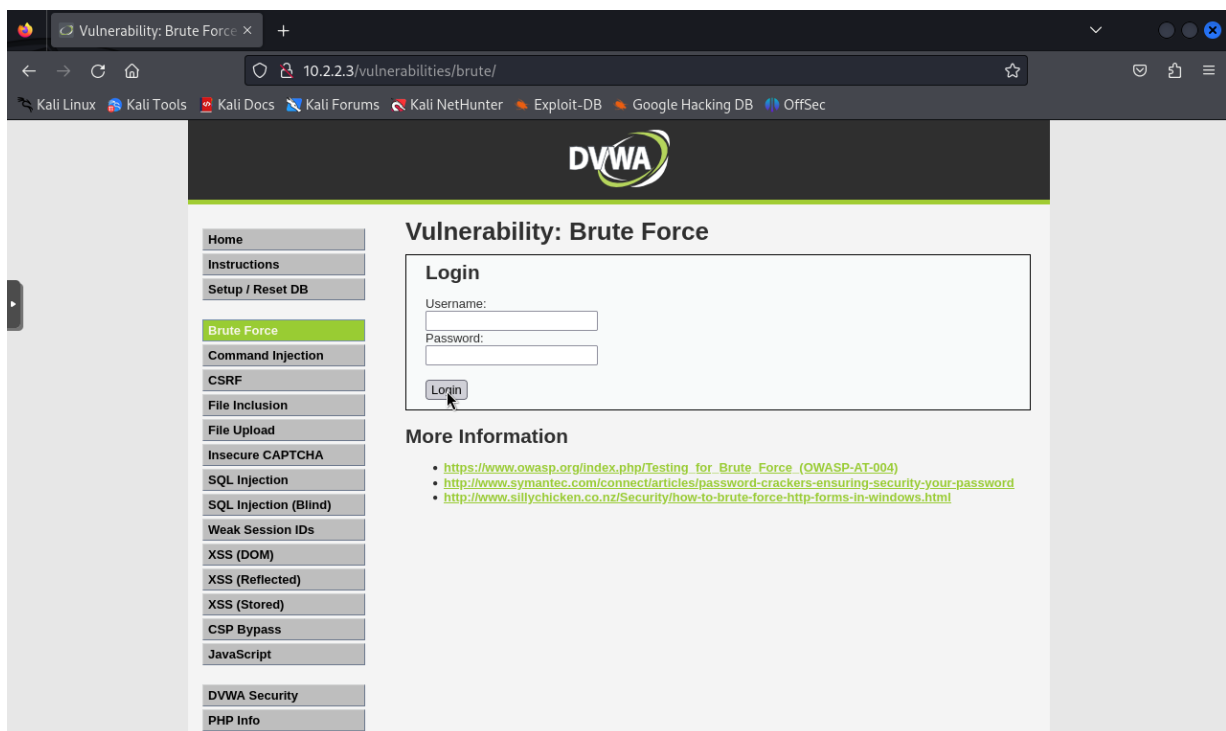


Figura 7.22: Envío solicitud HTTP a Burpsuite desde Firefox

Una vez hecho eso podremos ver información clave en la solicitud GET HTTP interceptada desde la herramienta Burpsuite (Figura 7.23):

- Los parametros de login (username=&password=&Login=Login)
- La cookie de sesión (PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low)

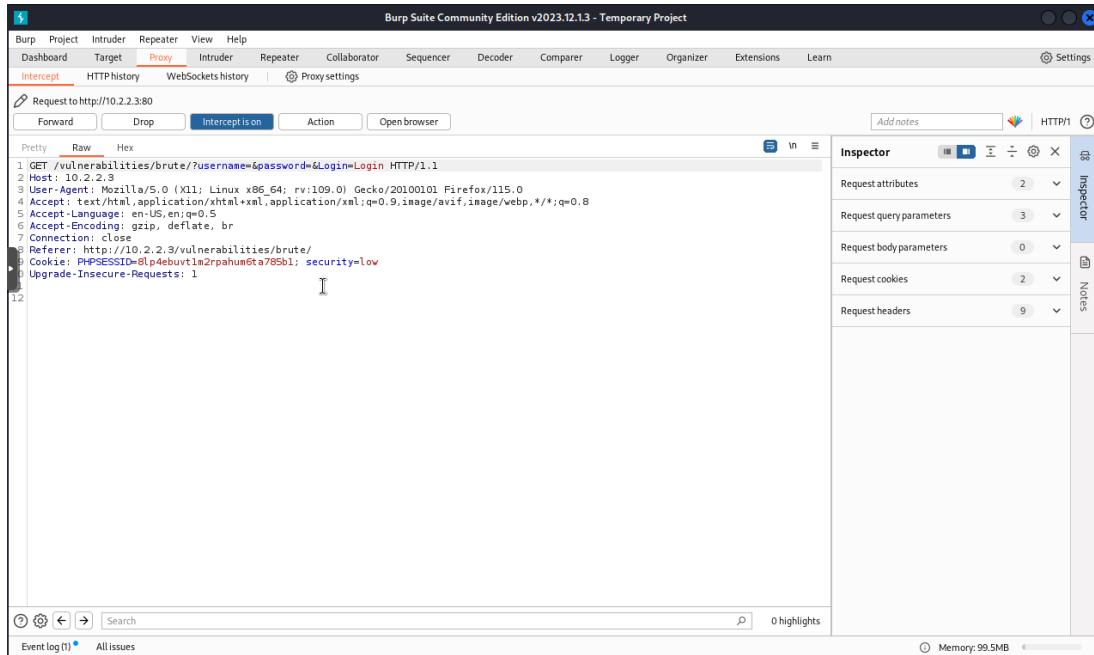


Figura 7.23: Solicitud HTTP interceptada con Burpsuite

Con esta información, podemos recrear una solicitud HTTP válida y hacer el ataque de fuerza bruta. Pero antes tendremos que volver a poner el Proxy como estaba antes en Firefox para que las solicitudes HTTP dejen de pasar por Burpsuite, como se ve en la Figura 7.24.

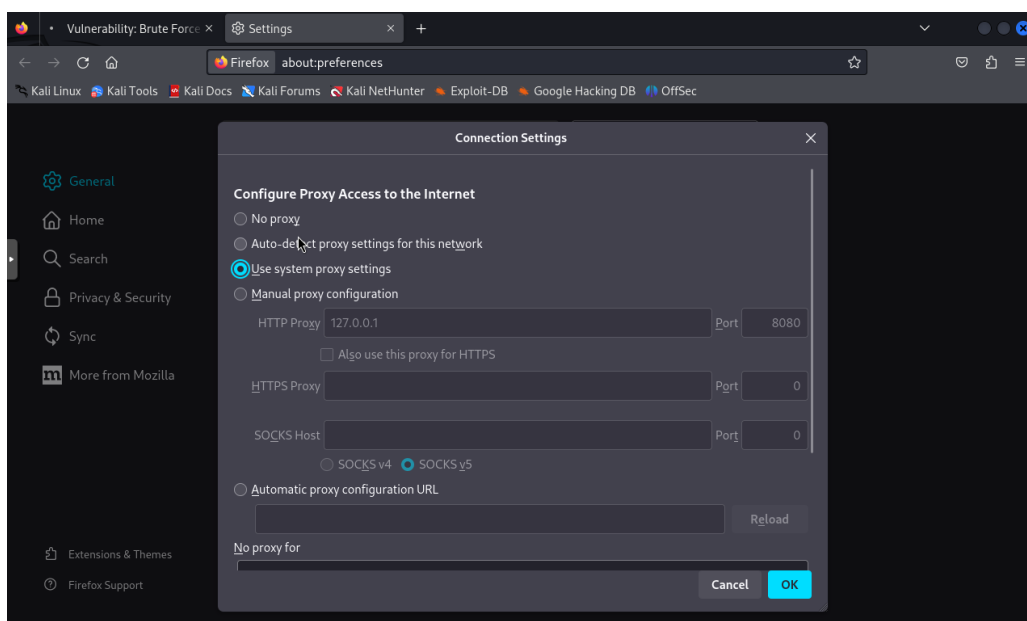


Figura 7.24: Configuración System Proxy Settings para Firefox

La herramienta que utilizaremos para hacer el ataque de fuerza bruta sobre el panel de login de la web DVWA es **THC Hydra**, la cual puede realizar un ataque de diccionario rápido y eficaz contra el servicio de autenticación.

Para realizar nuestro ataque de fuerza bruta con Hydra, necesitaremos proporcionar la siguiente información:

- **Servidor objetivo:** La dirección IP del servidor al que vamos a atacar, en este caso, [10.2.2.3](#).
- **Ruta de la URL:** La ruta específica del formulario de inicio de sesión en el servidor, que es [/vulnerabilities/brute/index.php](#).
- **Nombre de usuario:** El nombre de usuario que vamos a usar en el ataque. Vamos a suponer que conocemos que el nombre de usuario es [admin](#).
- **Diccionario de contraseñas:** La ruta al archivo que contiene una lista de posibles contraseñas [/usr/share/wordlist/fasttrack.txt](#).
- **Cookie:** Necesitamos proporcionar una cookie específica para mantener la sesión, que en este caso es `PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low`.
- **Mensaje de fallo:** La cadena que indica un intento de inicio de sesión fallido, que es `Username and/or password incorrect`.

El mensaje de fallo es la respuesta que obtenemos del formulario de inicio de sesión cuando enviamos credenciales incorrectas. Hydra busca esta cadena en el HTML de la respuesta para determinar si el inicio de sesión falló.

El comando completo que usaremos para llevar a cabo el ataque de fuerza bruta desde la máquina virtual Kali (VM02) lo podemos ver en la Figura 7.25.

```

root@vm02:~/home/usuario
# hydra 10.2.2.3 -l admin -P /usr/share/wordlists/fasttrack.txt http-get-form "/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login:H=Cookie: PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low:Username and/or password incorrect."
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-15 19:50:33
[DATA] max 16 tasks per 1 server, overall 16 tasks, 262 login tries (l:1/p:262), ~17 tries per task
[DATA] attacking http-get-form://10.2.2.3:80/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login:H=Cookie: PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low:Username and/or password incorrect.
[80][http-get-form] host: 10.2.2.3 login: admin password: password
1 of 1 target successfully completed, 1 valid password found
hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-15 19:50:36

```

Figura 7.25: Ataque de fuerza bruta con la herramienta hydra

A continuación, procederemos a explicar cada parte del comando.

```

hydra 10.2.2.3 -l admin -P /usr/share/wordlist/fasttrack.txt http-get-form "/
vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login:H=Cookie:
PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low:Username and/or password incorrect
."

```

- **hydra:** El comando para iniciar Hydra.
- **10.2.2.3:** La IP del servidor objetivo.
- **-l admin:** Especifica el nombre de usuario a utilizar en el ataque.
- **-P /usr/share/wordlist/fasttrack.txt:** Proporciona la ruta al archivo de diccionario de contraseñas, que viene por defecto instalado en Kali Linux.

- **http-get-form**: Utiliza el módulo de Hydra para atacar formularios web mediante el método HTTP GET.
- **/vulnerabilities/brute/index.php:username=^USER^&password=^PASS^&Login=Login**”: Define la ruta y los parámetros del formulario de inicio de sesión. Los marcadores ^USER^ y ^PASS^ serán reemplazados por Hydra con el nombre de usuario y las contraseñas del diccionario, respectivamente.
- **H=Cookie:PHPSESSID=v906fq0ons3ickgt4pc0gadoq0; security=low**: Proporciona una Cookie adicional necesaria para la sesión.
- **Username and/or password incorrect.**: La cadena que Hydra busca en la respuesta del servidor para determinar si las credenciales fueron incorrectas.

Una vez realizado el ataque con Hydra podemos ver en la Figura 7.25, que hemos conseguido averiguar la contraseña válida para iniciar sesión en el panel de autenticación (contraseña:password) en la web DVWA, una vez sabido el usuario admin.

Nada más ejecutar el ataque de fuerza bruta con la herramienta **hydra** sobre la página web DVWA objetivo nos llega una alerta al GMAIL (Figura 7.26), avisando que se ha producido un intento de ataque de fuerza bruta el día **15 de junio de 2024 a las 17:50:34 (UTC)**. Además, observamos que la regla que ha saltado en Suricata se llama **Possible Hydra Brute Force Attack** con un SID de **400004**, en la categoría **Web Application Attack** y la IP del host afectado es la **10.2.2.3**.

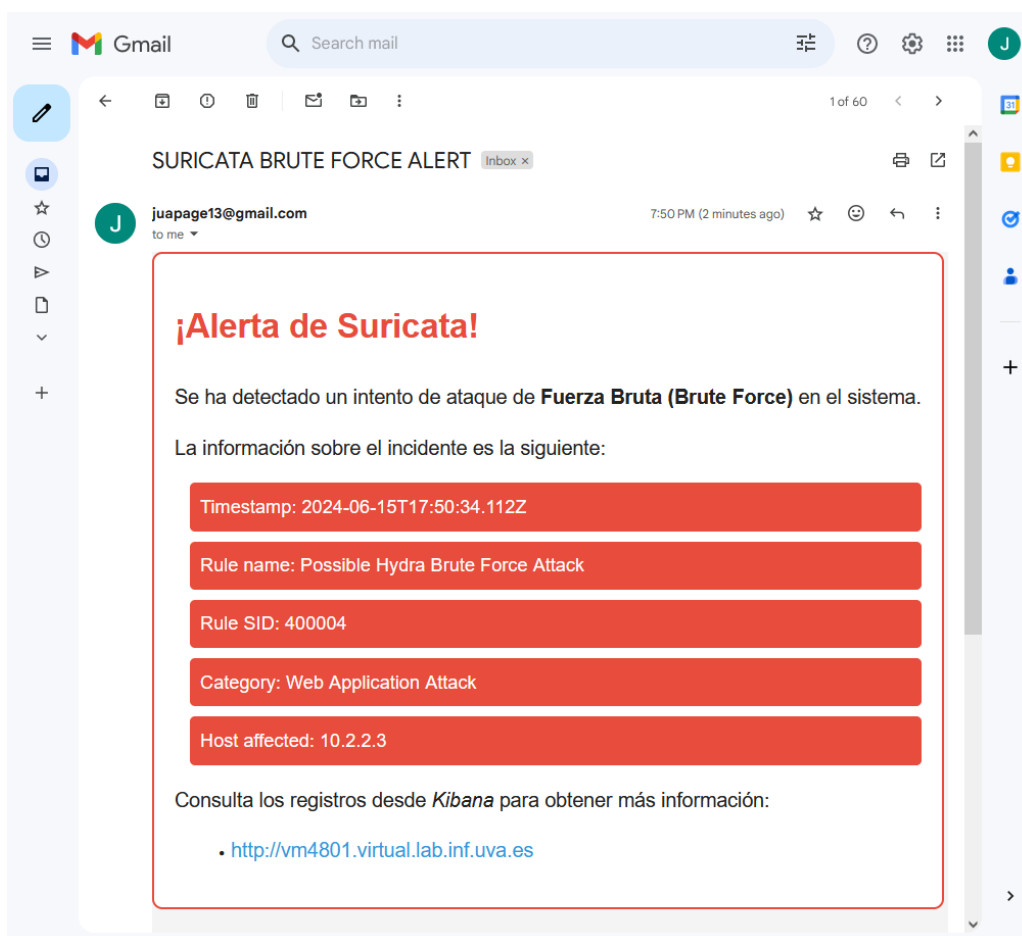


Figura 7.26: Alerta sobre Fuerza Bruta en GMAIL

Para poder ver más información podemos pinchar en el enlace que nos lleva a la web de Kibana que está alojada en el puerto 80 de la VM01. Una vez en Kibana podemos ver el **Events Dashboard** de Suricata (Figura 7.27, el cual muestra información más detallada como el número de eventos que se han producido en los últimos 5 minutos (2.876 eventos), el número de alertas que han saltado o los distintos protocolos de red que ha detectado en ese periodo de tiempo, entre otras cosas.

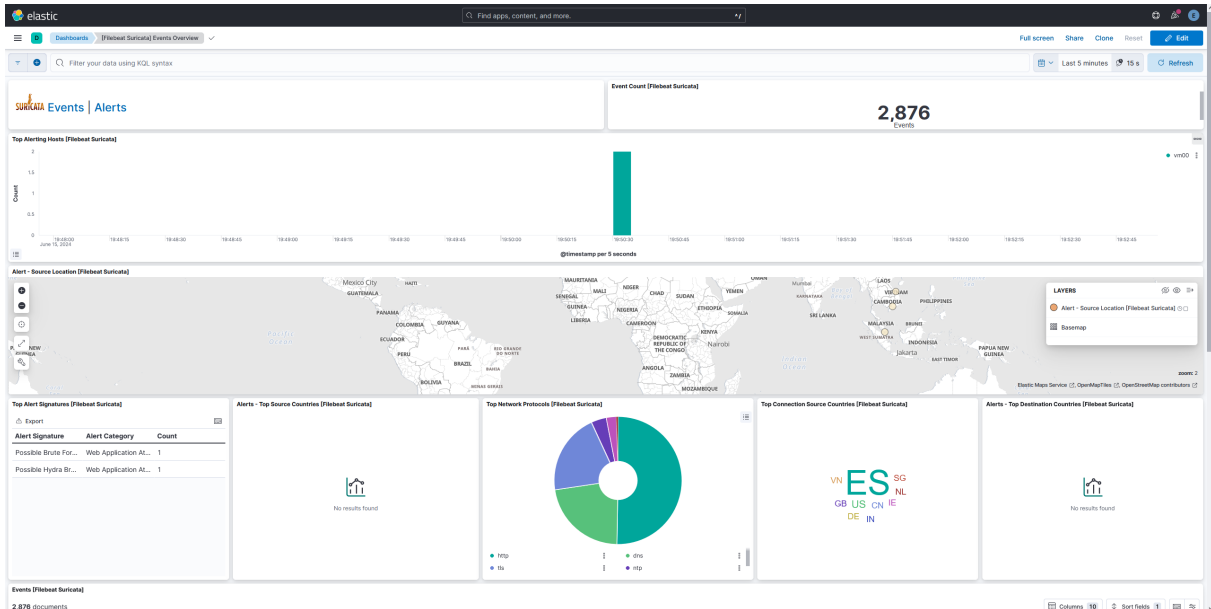


Figura 7.27: Dashboard Suricata Events sobre Fuerza Bruta en Kibana

Por otra parte, tendríamos el **Alerts Dashboard** (Figura 7.28), donde vemos que se han producido 2 alertas de Suricata en los últimos 5 minutos. La primera alerta salta porque se ha detectado la cadena **Hydra** en la solicitud HTTP con destino al servidor Web y la segunda salta porque se han producido más de 10 intentos de inicio de sesión en el formulario situado en **/vulnerabilities/brute** en un periodo de 10 segundos. Ambas alertas encajan perfectamente con el ataque de fuerza bruta previamente realizado con la herramienta Hydra.

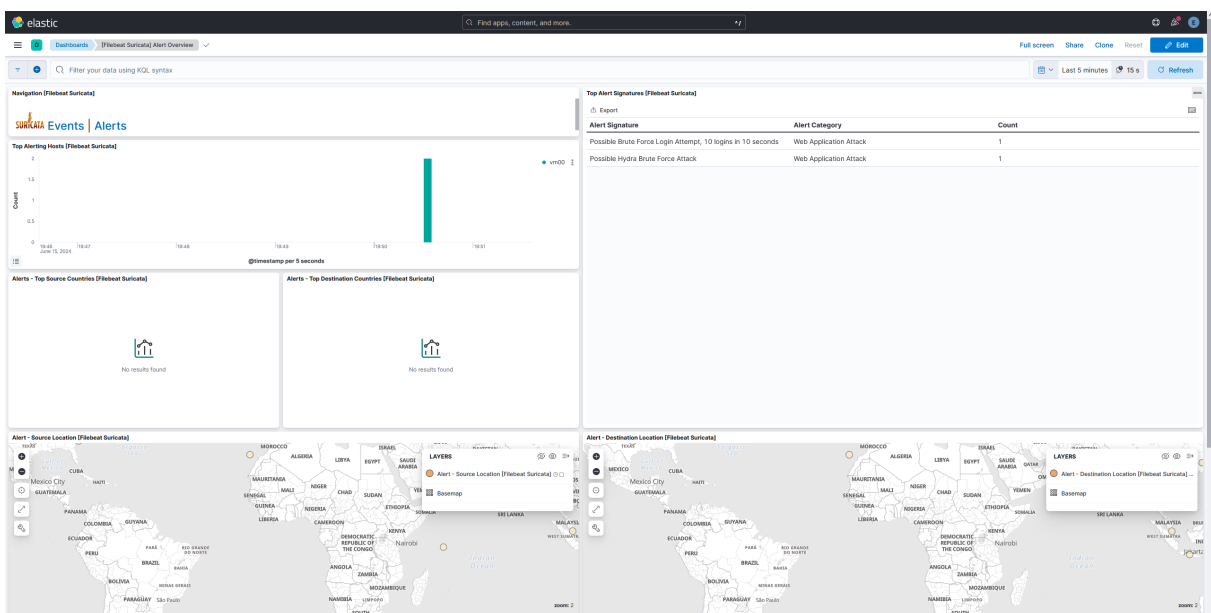


Figura 7.28: Dashboard Suricata Alerts sobre Fuerza Bruta en Kibana

Para acabar, si se quisiera ver información sobre el incidente en profundidad existe la posibilidad de ver el **JSON log** que se envió desde la máquina donde está Suricata, tal y como podemos apreciar en la Figura 7.29.

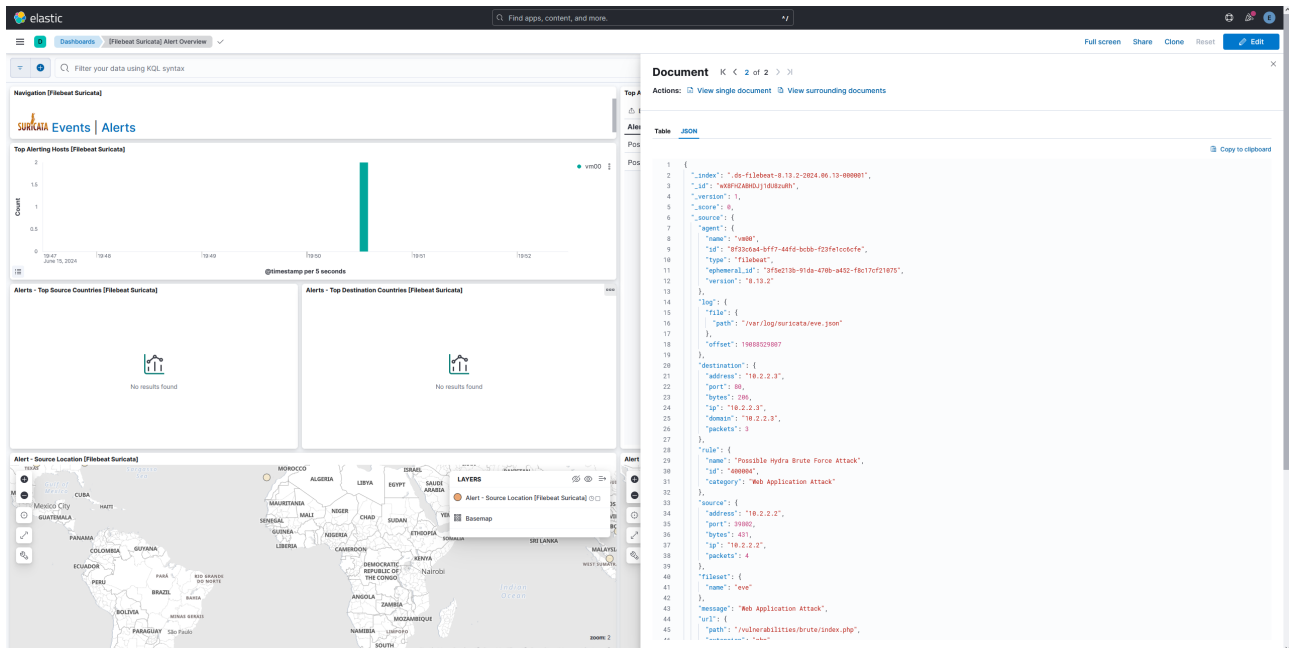


Figura 7.29: JSON Suricata Log sobre Fuerza Bruta en Kibana

7.5 Simulación ataque DDoS

En esta prueba, se llevará a cabo un ataque de denegación de servicio (Distributed Denial of Service). Este ataque tiene como objetivo hacer que el servidor web sea inaccesible para los usuarios legítimos, saturando los recursos del servidor con tráfico malicioso.

Para lanzar el ataque DDoS emplearemos la herramienta **Hping3**, la cual se sirve para enviar paquetes a través de una red y analizar las respuestas. Además, es comúnmente utilizada para pruebas de penetración y diagnóstico de red.

El comando completo que hemos utilizado en el ataque desde la máquina virtual Kali (VM02) lo podemos observar en la Figura 7.30.

```
(root@vm02)-[/home/usuario]
# timeout 15s hping3 -S --flood -V -p 80 10.2.2.3 --rand-source
using eth1, addr: 10.2.2.2, MTU: 1500
HPING 10.2.2.3 (eth1 10.2.2.3): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown

— 10.2.2.3 hping statistic —
1619432 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figura 7.30: Ataque DDoS con la herramienta hping3

A continuación, explicaremos cada componente del comando.

```
timeout 15s hping3 -S --flood -V -p 80 10.2.2.3 --rand-source
```

- **timeout 15s**: Ejecutará el comando hping3 durante 15 segundos.
- **-S**: Indica que se enviarán paquetes TCP con la bandera SYN establecida. Esto simula el inicio de una conexión TCP.
- **-flood**: Esta opción indica que se enviarán los paquetes tan rápido como sea posible, sin esperar una respuesta del servidor. Se utiliza para realizar ataques de inundación.
- **-V**: Activa el modo verbose, proporcionando información detallada sobre lo que está sucediendo mientras se ejecuta el comando.
- **-p 80**: Especifica el puerto de destino al que se enviarán los paquetes. En este caso, los paquetes se dirigirán al puerto 80, utilizado comúnmente por los servidores web para el tráfico HTTP.
- **10.2.2.3**: Es la dirección IP del servidor al que se enviarán los paquetes.
- **-rand-source**: Indica que se utilizarán direcciones IP de origen aleatorias al enviar los paquetes. Esto puede hacer que el ataque sea más difícil de detectar y mitigar.

Nada más ejecutar el ataque DDoS con la herramienta **hping3** sobre la página web DVWA objetivo nos llega una alerta al GMAIL (Figura 7.31), alertando que se ha producido un intento de ataque DDoS el día **15 de junio de 2024 a las 17:21:02 (UTC)**. Además, podemos ver que la regla que ha saltado en Suricata se llama **Possible SYN Flood Attack** con un SID de **500001**, en la categoría **Detection of a Denial of Service Attack** y la IP del host afectado es la **10.2.2.3**.

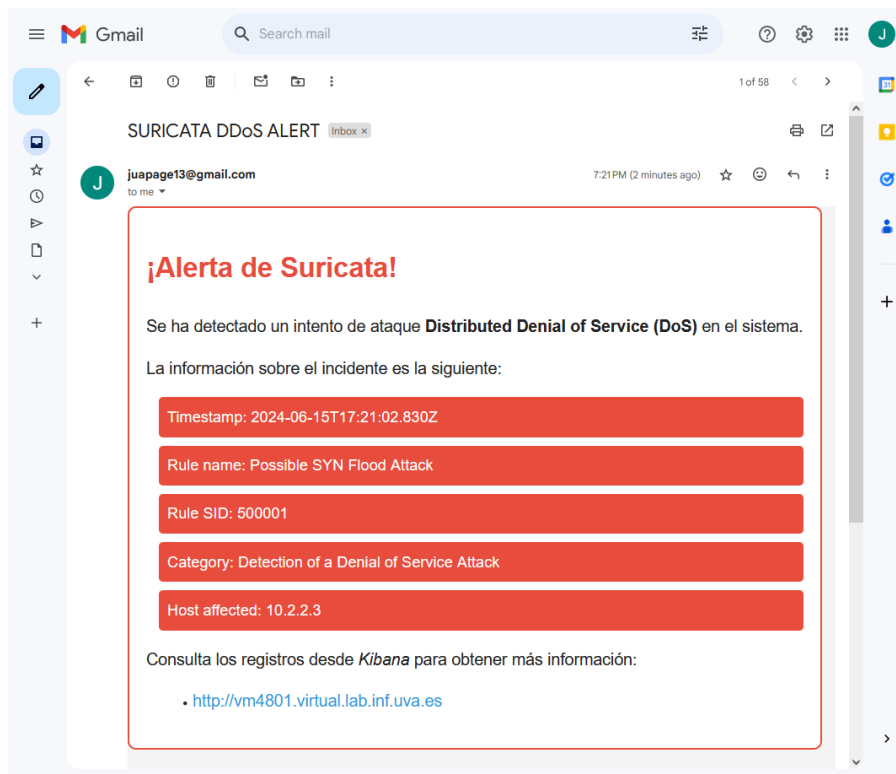


Figura 7.31: Alerta sobre DDoS en GMAIL

Para ver más información podemos pinchar en el enlace que nos lleva a la web de Kibana que está alojada en el puerto 80 de la VM01. Una vez en Kibana podemos ver el **Events Dashboard** de Suricata (Figura 7.32), el cual muestra información más detallada como el número de eventos que se han producido en los últimos 5 minutos (253.248 eventos), el número de alertas que han saltado o los distintos protocolos de red que ha detectado en ese periodo de tiempo, entre otras cosas.

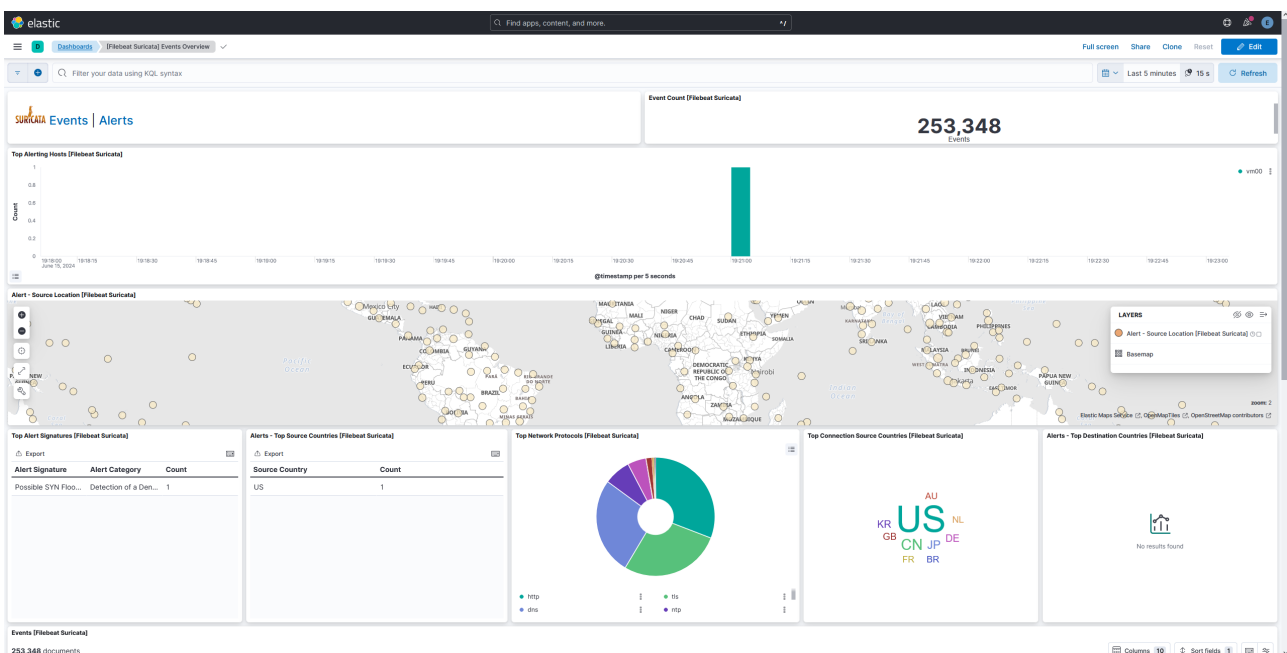


Figura 7.32: Dashboard Suricata Events sobre DDoS en Kibana

Por otro lado, tendríamos el **Alerts Dashboard** (Figura 7.33), donde observamos que se ha producido 1 alerta de Suricata en los últimos 5 minutos. La alerta salta porque se ha detectado más de 5000 solicitudes SYN con destino al servidor Web en un periodo de 20 segundos, lo cual cuadra con el ataque DDoS que hemos realizado anteriormente con hping3.

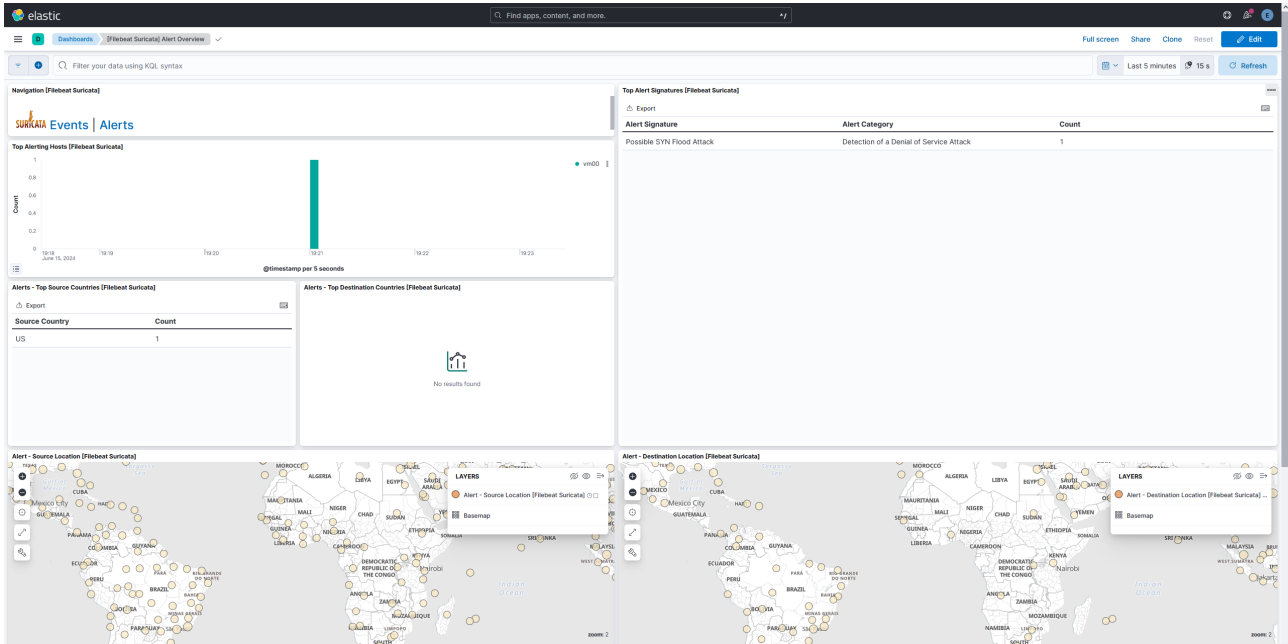


Figura 7.33: Dashboard Suricata Alerts sobre DDoS en Kibana

Para terminar, si se quisiera ver información sobre el incidente más en detalle existe la posibilidad de ver el **JSON log** que se envió desde la máquina donde está Suricata, tal y como podemos apreciar en la Figura 7.34.

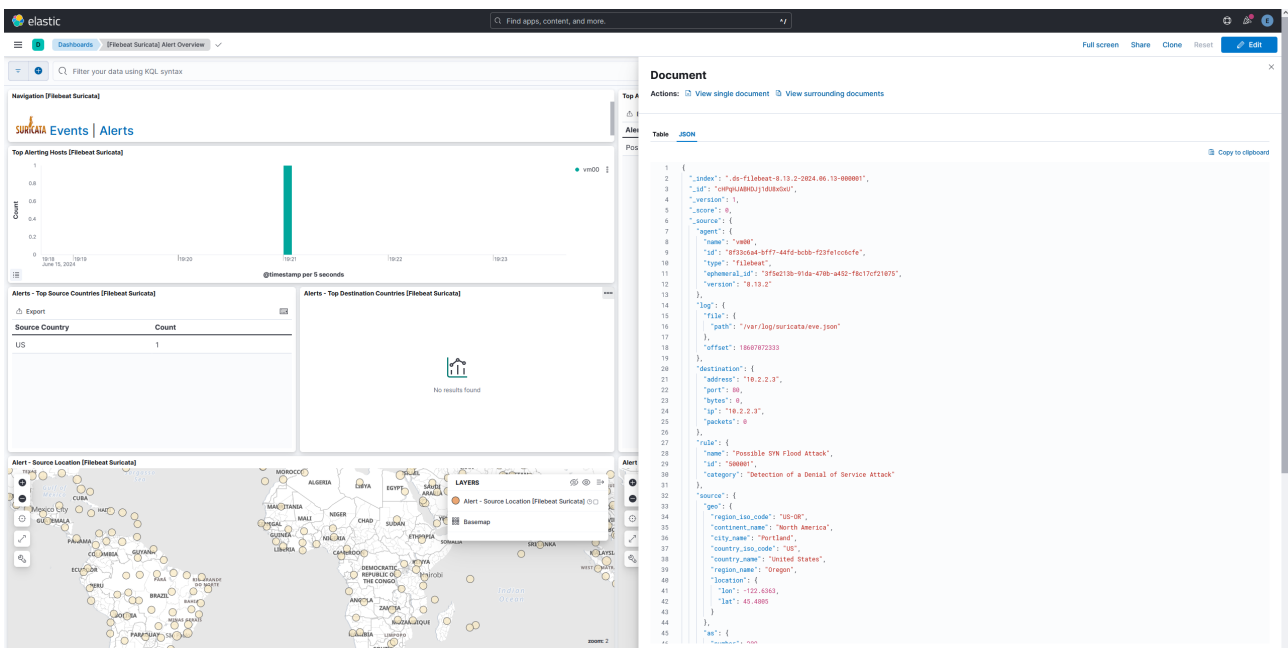


Figura 7.34: JSON Suricata Log sobre DDoS en Kibana

Conclusiones

En este Trabajo de Fin de Grado se ha llevado a cabo la investigación, diseño e implementación de un sistema de alerta temprana de amenazas de seguridad basado en la integración de Suricata, Elasticsearch, Kibana y ElastAlert2, con el objetivo de mejorar la capacidad de detección y respuesta a incidentes de seguridad en entornos empresariales.

A lo largo del proyecto, se han cumplido satisfactoriamente todos los objetivos previstos:

- Investigación y selección de herramientas: Se investigaron, seleccionaron y aprendieron a utilizar las herramientas y tecnologías más adecuadas para la implementación del sistema de alerta temprana, incluyendo Suricata, Elasticsearch, Kibana y ElastAlert2.
- Configuración y despliegue de Elasticsearch y Kibana: Se configuró y desplegó un entorno robusto de Elasticsearch y Kibana para la recolección y visualización de datos de seguridad, permitiendo un análisis detallado y en tiempo real de los eventos de seguridad.
- Integración de Suricata: Suricata se integró como sensor de red, habilitando la detección en tiempo real de amenazas cibernéticas y proporcionando una capa adicional de seguridad mediante la monitorización continua del tráfico de red.
- Desarrollo de reglas de detección personalizadas: Se desarrollaron y probaron reglas de detección personalizadas para Suricata, diseñadas específicamente para identificar comportamientos maliciosos y actividades sospechosas en el entorno empresarial.
- Configuración de ElastAlert2: Se configuró ElastAlert2 para generar alertas a partir de los eventos detectados por Suricata, asegurando que las notificaciones se envíen oportunamente a los responsables de seguridad para una respuesta rápida y efectiva.
- Pruebas y validación del sistema: El sistema de alerta temprana se probó y validó en un entorno de laboratorio simulado, evaluando su efectividad en la detección y respuesta a diversas amenazas de seguridad. Los resultados demostraron que el sistema es capaz de identificar y responder a incidentes de seguridad con alta precisión.
- Documentación del proceso: Se documentó exhaustivamente el proceso de diseño, implementación y pruebas del sistema de alerta temprana. Se proporcionaron guías detalladas que facilitan su uso y mantenimiento, garantizando que el sistema pueda ser replicado y utilizado por otros profesionales de la seguridad informática.

En conclusión, el proyecto ha logrado todos sus objetivos, proporcionando una solución efectiva y bien documentada para la detección temprana y la respuesta a amenazas de seguridad en entornos empresariales. Este trabajo no solo contribuye al campo de la seguridad informática, sino que también ofrece una herramienta práctica y accesible para mejorar la protección de las infraestructuras críticas en las organizaciones.

8.1 Trabajo futuro

A pesar de que el sistema de alerta temprana desarrollado en este trabajo ha cumplido con los objetivos previstos, existen varias áreas en las que se pueden realizar mejoras y ampliaciones para aumentar su efectividad y robustez en el futuro.

Una posible mejora significativa sería la implementación del sistema como un Sistema de Prevención de Intrusiones (IPS). Actualmente, Suricata se utiliza como un Sistema de Detección de Intrusiones (IDS), que identifica y alerta sobre actividades sospechosas, pero no interviene directamente. Transformarlo en un IPS permitiría no solo detectar, sino también prevenir automáticamente ataques en tiempo real, bloqueando o mitigando amenazas antes de que puedan causar daños.

Otra mejora crucial es el aumento de la cobertura de detección mediante el desarrollo de reglas específicas. Aunque se han creado reglas personalizadas para identificar comportamientos maliciosos, la creación y actualización continua de un conjunto más amplio de reglas permitirá detectar una gama más amplia de amenazas, incluyendo las técnicas más recientes y avanzadas utilizadas por los atacantes.

Además, la extensión del monitoreo y análisis a los endpoints representaría un avance significativo en la capacidad del sistema para detectar y responder a amenazas. Integrar herramientas de Endpoint Detection and Response (EDR) permitiría obtener una visión más completa de las actividades sospechosas en toda la infraestructura, no solo en el tráfico de red. Esto facilitaría la identificación de amenazas que puedan haber evadido las defensas de red tradicionales y proporcionaría capacidades adicionales para investigar y responder a incidentes de seguridad.

La gestión y el análisis de logs también pueden beneficiarse de mejoras. Implementar una solución centralizada de gestión de logs que permita la correlación de eventos de múltiples fuentes mejoraría considerablemente la capacidad de análisis del sistema. Además, el uso de técnicas avanzadas de análisis, como el Machine Learning, podría ayudar a identificar patrones anómalos y comportamientos sospechosos que podrían pasar desapercibidos con los métodos tradicionales.

Por otro lado, se podría implementar una política de limpieza de logs para asegurar que en un futuro cercano las máquinas no tengan el disco duro saturado. De esta manera se podría eliminar los logs más antiguos o transferirlos a un almacén persistente de datos, para su posterior consulta si fuera necesario.

Finalmente, la implementación de Backups en la nube para los logs garantizaría la integridad y disponibilidad de los datos de seguridad. Almacenar los logs en la nube no solo proporciona una solución robusta y escalable para la conservación de datos históricos, sino que también asegura que los datos están protegidos contra pérdidas o daños locales. Esto es especialmente importante para el análisis forense post-incidente y para cumplir con las normativas de retención de datos.

Estas propuestas de trabajo futuro ofrecen una hoja de ruta clara para mejorar y expandir las capacidades del sistema de alerta temprana, fortaleciendo la postura de seguridad de las organizaciones y adaptándose a las crecientes y cambiantes amenazas en el ámbito de la ciberseguridad.

Apéndices

Repositorio del Proyecto

En este manual, ofrecemos una introducción al repositorio público de GitHub conocido como TFG [38]. El repositorio contiene todos los archivos y configuraciones necesarios para implementar un sistema de alerta temprana de amenazas de seguridad en entornos empresariales.

El sistema se basa en una arquitectura que integra diversas tecnologías, incluyendo Suricata para la detección de amenazas, Elasticsearch y Kibana para el almacenamiento y visualización de datos, y ElastAlert2 para la generación de alertas. Además, se incluye una infraestructura de máquinas virtuales que simulan un entorno de prueba realista.

El repositorio está estructurado de manera organizada, facilitando la navegación y comprensión de los diferentes componentes del sistema. Se proporcionan instrucciones detalladas para configurar cada aspecto del sistema, desde la instalación de las herramientas necesarias hasta la ejecución y monitorización del sistema en un entorno simulado.

Por último, si estás interesado en contribuir al proyecto, eres bienvenido a hacerlo. Se anima a potenciales colaboradores a mejorar las funcionalidades existentes o a proponer nuevas ideas que puedan enriquecer el proyecto. El objetivo es crear un sistema robusto y efectivo que pueda ser utilizado por profesionales de la ciberseguridad para mejorar la detección y respuesta a amenazas en entornos empresariales.

Bibliografía

- [1] Sherif Abdel-Naby. *Elastic Stack on Docker*. URL: <https://github.com/sherifabdlnaby/elasticsearch/blob/main/README.md> (visitado 17-06-2024).
- [2] *Apache Flume*. URL: <https://flume.apache.org/> (visitado 17-06-2024).
- [3] *Apache Kafka*. URL: <https://kafka.apache.org/> (visitado 17-06-2024).
- [4] *Apache Solr*. URL: <https://solr.apache.org/> (visitado 17-06-2024).
- [5] Prometheus Authors. *Prometheus Alertmanager*. URL: <https://prometheus.io/docs/alerting/latest/alertmanager/> (visitado 17-06-2024).
- [6] Daniel Berman. *Installing the ELK Stack on Docker*. URL: <https://logz.io/blog/elk-stack-on-docker/> (visitado 17-06-2024).
- [7] Danny Beton. *DVWA Brute Force Tutorial (Low Security)*. URL: <https://medium.com/@dannybeton/dvwa-brute-force-tutorial-low-security-463880d53e50> (visitado 17-06-2024).
- [8] Paolo Bonzini. *KVM*. URL: https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine (visitado 17-06-2024).
- [9] Broadcom. *VMware vSphere*. URL: <https://www.vmware.com/products/vsphere.html> (visitado 17-06-2024).
- [10] Jamon Camisso. *How To Build A SIEM with Suricata and Elastic Stack on Ubuntu 20.04*. URL: <https://www.digitalocean.com/community/tutorials/how-to-build-a-siem-with-suricata-and-elastic-stack-on-ubuntu-20-04> (visitado 17-06-2024).
- [11] Jamon Camisso. *How To Install Suricata on Ubuntu 20.04*. URL: <https://www.digitalocean.com/community/tutorials/how-to-install-suricata-on-ubuntu-20-04> (visitado 17-06-2024).
- [12] *Canva*. URL: <https://www.canva.com/> (visitado 17-06-2024).
- [13] Felix Christl. *Email alerts for problems with Dockerized services using Elasticsearch and ElastAlert*. URL: <https://medium.com/@fchristl/email-alerts-for-problems-with-dockerized-services-using-elasticsearch-and-elastalert-bd26b9363881> (visitado 17-06-2024).
- [14] Cisco. *Snort*. URL: <https://www.snort.org/> (visitado 17-06-2024).
- [15] Cisco. *Splunk*. URL: <https://www.splunk.com/> (visitado 17-06-2024).
- [16] OpenSearch contributors. *Opensearch*. URL: <https://opensearch.org/> (visitado 17-06-2024).
- [17] Docker. *Docker*. URL: <https://www.docker.com/> (visitado 17-06-2024).
- [18] Elastic. *Filebeat quick start: installation and configuration*. URL: <https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-installation-configuration.html> (visitado 17-06-2024).
- [19] Elastic. *Suricata module*. URL: <https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-module-suricata.html> (visitado 17-06-2024).

- [20] Elastic. *Watcher*. URL: <https://www.elastic.co/guide/en/kibana/current/watcher-ui.html> (visitado 17-06-2024).
- [21] Elastic. *What is Elasticsearch?* URL: <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html> (visitado 17-06-2024).
- [22] Elastic. *What is Kibana?* URL: <https://www.elastic.co/guide/en/kibana/current/introduction.html> (visitado 17-06-2024).
- [23] Jason Ertel. *ElastAlert 2*. URL: <https://github.com/jertel/elastalert2> (visitado 17-06-2024).
- [24] Jason Ertel. *Getting Started*. URL: https://elastalert2.readthedocs.io/en/latest/running_elastalert.html#as-a-docker-container (visitado 17-06-2024).
- [25] MANJUNATH NAYAKA G. *Understanding File Inclusion Attack using DVWA web application*. URL: <https://medium.com/@manjuteju008/understanding-file-inclusion-attack-using-dvwa-web-application-30d06846c269> (visitado 17-06-2024).
- [26] POSITIVE GEEK. *How to Create App Password for Google Account | SMTP Configuration for Gmail Account*. URL: <https://www.youtube.com/watch?v=74QQfPrk4vE> (visitado 17-06-2024).
- [27] Gigi. *Filebeat, Elasticsearch and Kibana with Docker Compose*. URL: <https://gigi.nullneuron.net/gigilabs/filebeat-elasticsearch-and-kibana-with-docker-compose/> (visitado 17-06-2024).
- [28] Github. URL: <https://github.com/> (visitado 17-06-2024).
- [29] Erin Glass. *Cómo instalar Elasticsearch, Logstash y Kibana (Elastic Stack) en Ubuntu 20.04*. URL: <https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-20-04-es> (visitado 17-06-2024).
- [30] Adiscon GmbH. *Rsyslog*. URL: <https://www.rsyslog.com/> (visitado 17-06-2024).
- [31] Grafana. URL: <https://grafana.com/> (visitado 17-06-2024).
- [32] Graylog. URL: <https://graylog.org/> (visitado 17-06-2024).
- [33] Graylog Alerting. URL: <https://graylog.org/videos/alerts/> (visitado 17-06-2024).
- [34] Dotan Horovits. *The Complete Guide to the ELK Stack*. URL: <https://logz.io/learn/complete-guide-elk-stack/#latest-on-the-elk-stack> (visitado 17-06-2024).
- [35] Howtoforge. *Cómo instalar y configurar Suricata IDS junto con Elastic Stack en Ubuntu 22.04*. URL: <https://howtoforge.es/como-instalar-y-configurar-suricata-ids-junto-con-elastic-stack-en-ubuntu-22-04/> (visitado 17-06-2024).
- [36] Arvind Ponnarassery Jayan. *Suricata-Detect-DoS-Attack*. URL: <https://github.com/arvindpj007/Suricata-Detect-DoS-Attack/tree/master> (visitado 17-06-2024).
- [37] Anthony Lapenna. *Elastic stack (ELK) on Docker*. URL: <https://github.com/deviantony/docker-elk/blob/main/README.md> (visitado 17-06-2024).
- [38] Juan Antonio Pagés López. *Github TFG*. URL: <https://github.com/zerebritvs/TFG> (visitado 17-06-2024).
- [39] Logshero Ltd. *Logz.io*. URL: <https://logz.io/> (visitado 17-06-2024).
- [40] Microsoft. *Visual Studio Code*. URL: <https://code.visualstudio.com/> (visitado 17-06-2024).
- [41] Microsoft. *Hyper-V*. URL: <https://learn.microsoft.com/es-es/windows-server/virtualization/hyper-v/hyper-v-technology-overview> (visitado 17-06-2024).
- [42] Microsoft. *Teams*. URL: <https://www.microsoft.com/es-es/microsoft-teams/group-chat-software> (visitado 17-06-2024).

- [43] Shreetheja S N. *Centralised Logging System Using ELK and Filebeat*. URL: <https://medium.com/@snshagri/centralised-logging-system-using-elk-and-filebeat-067e1373dd71> (visitado 17-06-2024).
- [44] *Obsidian*. URL: <https://obsidian.md/> (visitado 17-06-2024).
- [45] Yeray Manuel Páez Olmedo. *Alerta Temprana de Amenazas de Seguridad con Apache Kafka y la Pila ELK*. URL: <https://uvadoc.uva.es/bitstream/handle/10324/50431/TFG-G5267.pdf?sequence=1&isAllowed=y> (visitado 17-06-2024).
- [46] *OpenAI*. URL: <https://openai.com/> (visitado 17-06-2024).
- [47] Oracle. *Virtualbox*. URL: <https://www.virtualbox.org/> (visitado 17-06-2024).
- [48] Klinsmann Öteyo. *Run Elastic stack (ELK) on Docker Containers with Docker Compose*. URL: <https://computingforgeeks.com/run-elastic-stack-elk-on-docker/> (visitado 17-06-2024).
- [49] *Overleaf*. URL: <https://www.overleaf.com> (visitado 17-06-2024).
- [50] Jérôme Petazzoni. *Introduction to Containers*. URL: <https://container.training/intro-selfpaced.yml.html#1> (visitado 17-06-2024).
- [51] Fluentd Project. *Fluentd*. URL: <https://www.fluentd.org/> (visitado 17-06-2024).
- [52] Fluentd Project. *Fluentd-ui*. URL: <https://docs.fluentd.org/deployment/fluentd-ui> (visitado 17-06-2024).
- [53] The Zeek Project. *Zeek*. URL: <https://zeek.org/> (visitado 17-06-2024).
- [54] *Proxmox*. URL: <https://www.proxmox.com/en/> (visitado 17-06-2024).
- [55] RedesPlus. *IPS IDS Linux - Instalar SURICATA y configurar las REGLAS y Alertas*. URL: <https://www.youtube.com/watch?v=vQNB7nenT2E> (visitado 17-06-2024).
- [56] Brad Searle. *Proxmox VM Bridge Port Mirror*. URL: <https://codingpackets.com/blog/proxmox-vm-bridge-port-mirror/> (visitado 17-06-2024).
- [57] *Sensu*. URL: <https://sensu.io/> (visitado 17-06-2024).
- [58] M Shulkhan. *Detection Attack using Suricata-2*. URL: <https://medium.com/@mshulkhan/detection-attack-using-suricata-2-d93d423a435> (visitado 17-06-2024).
- [59] Sameera De Silva. *How to send an email alert with html format in ElastAlert*. URL: <https://samedesilva.medium.com/how-to-send-an-email-alert-with-html-format-in-elastalert-eale5c071e1f> (visitado 17-06-2024).
- [60] Security Onion Solutions. *Security Onion 2*. URL: <https://securityonionsolutions.com/software/> (visitado 17-06-2024).
- [61] *Sumo Logic*. URL: <https://www.sumologic.com/> (visitado 17-06-2024).
- [62] *Suricata*. URL: <https://suricata.io/> (visitado 17-06-2024).
- [63] Snort 3.0 Team. *Snort++*. URL: <https://github.com/snort3/snort3> (visitado 17-06-2024).
- [64] Aayush Tiruwa. *DVWA SQL INJECTION*. URL: <https://medium.com/@aayushtiruwa120/dvwa-sql-injection-91b4efb683e4> (visitado 17-06-2024).
- [65] Carlos Prado Ventura. *Interfaz de visualización de logs en un sistema de información para la gestión de datos en investigación clínica*. URL: https://oa.upm.es/43477/1/TFG_CARLOS_PRADO_VENTURA_a.pdf (visitado 17-06-2024).
- [66] Ali Younes. *Send Email Alerts for FREE with ElastAlert2*. URL: <https://www.youtube.com/watch?v=wKeUmRuRxi8&t=305s> (visitado 17-06-2024).

Glosario de términos

A

Apache Flume

Servicio distribuido para recolectar, agregar y mover grandes cantidades de datos de logs.

Apache Kafka

Plataforma distribuida de transmisión de datos para manejar flujos de datos en tiempo real.

Apache Solr

Plataforma de búsqueda de código abierto basada en Lucene, utilizada para búsquedas empresariales.

API

Application Programming Interface, conjunto de definiciones y protocolos para construir e integrar software.

Ataque fuerza bruta

Método de prueba y error para descifrar contraseñas o claves cifradas.

B

Backups

Copias de seguridad de datos almacenadas en dispositivos físicos o en la nube, realizadas para asegurar la disponibilidad y recuperación de la información en caso de pérdida, corrupción o eliminación accidental.

Bridge Port Mirroring

Bridge Port Mirroring es una técnica de red que copia el tráfico de una o varias interfaces (puertos) a otra interfaz para monitoreo y análisis.

Burpsuite

Herramienta integrada para realizar pruebas de seguridad en aplicaciones web.

C

Canva

Herramienta en línea de diseño gráfico que permite crear contenido visual de forma sencilla.

CAPTCHAs

Un mecanismo de seguridad que distingue entre humanos y bots, generalmente mediante pruebas visuales o de audio.

Contenedor

Entorno aislado para ejecutar aplicaciones de manera consistente en diferentes entornos.

Cookie

Un pequeño archivo de texto que un sitio web guarda en el ordenador del usuario para almacenar información sobre la sesión actual o hábitos de navegación.

D**DDoS**

Distributed Denial of Service, ataque distribuido para agotar los recursos de un sistema.

DHCP

Dynamic Host Configuration Protocol, protocolo para asignar direcciones IP dinámicas a dispositivos en una red.

Docker

Plataforma para desarrollar, enviar y ejecutar aplicaciones dentro de contenedores.

DoS

Denial of Service, ataque que busca hacer que un sistema no esté disponible para sus usuarios.

DVWA

Damn Vulnerable Web Application, aplicación web vulnerable para pruebas de seguridad.

E**EDR**

Endpoint Detection and Response (EDR) se refiere a las soluciones de seguridad informática diseñadas para detectar, investigar y mitigar amenazas en tiempo real en dispositivos finales o endpoints.

Elastalert2

Herramienta de alerta para Elasticsearch basada en reglas y flexible.

Elastic

Empresa que desarrolla Elasticsearch, Kibana, Beats y Logstash.

Elasticsearch

Motor de búsqueda y análisis basado en Lucene, desarrollado por Elastic.

ELK

Suite de herramientas que incluye Elasticsearch, Logstash y Kibana para análisis de datos.

F**Filebeat**

Herramienta para enviar y centralizar archivos de log en Elasticsearch.

Firefox

Navegador web de código abierto desarrollado por Mozilla Corporation y la comunidad de software libre.

Firewalls

Sistemas de seguridad que monitorizan y controlan el tráfico de red basado en reglas de seguridad preestablecidas.

Fluentd

Herramienta de código abierto para unificar la recolección y el consumo de logs.

Fluentd-ui

Conjunto de componentes UI de código abierto para crear aplicaciones web modernas.

G**GitHub**

Plataforma para alojar y gestionar proyectos de desarrollo de software usando Git.

GMAIL

Servicio de correo electrónico proporcionado por Google.

GPT

Generative Pre-trained Transformer, modelo de lenguaje avanzado desarrollado por OpenAI.

Grafana

Plataforma de código abierto para análisis y monitoreo de métricas y datos de tiempo real.

Graylog

Herramienta de gestión y análisis de logs de código abierto.

Graylog Alerting

Sistema de alertas de Graylog para notificar eventos importantes basados en logs.

H**HIDS**

Host-based Intrusion Detection System, sistema de detección de intrusos basado en host.

Hipervisor

Software que permite crear y gestionar máquinas virtuales, permitiendo ejecutar múltiples sistemas operativos en un solo hardware.

Hping3

Herramienta para enviar paquetes TCP/IP personalizados para pruebas de red y seguridad.

HTTP

Hypertext Transfer Protocol (HTTP) es un protocolo de aplicación utilizado para la transferencia de información en la World Wide Web.

I**IA**

Inteligencia Artificial, simulación de procesos de inteligencia humana por máquinas.

IDS

Sistema de detección de intrusos, monitorea y analiza el tráfico de red en busca de actividades maliciosas.

IoT

Internet of Things, interconexión de dispositivos a través de Internet para compartir datos.

IP

Internet Protocol, protocolo principal de comunicaciones en la red de Internet.

IPS

Intrusion Prevention System, sistema de prevención de intrusos que bloquea actividades maliciosas.

J**JSON**

JavaScript Object Notation, formato de intercambio de datos ligero y de fácil lectura.

K**Kanban**

Método visual para gestionar el trabajo de un equipo, mediante tarjetas y tableros.

Kibana

Herramienta de visualización y análisis para trabajar con datos en Elasticsearch.

KVM

Kernel-based Virtual Machine, solución de virtualización en el kernel de Linux.

L**LAND Flood Attack**

Envía paquetes con dirección IP de origen y destino falsificadas iguales, causando que la víctima se bloquee al intentar responder a sí misma.

LaTeX

Sistema de preparación de documentos de alta calidad tipográfica, muy usado en academia.

LFI

Local File Inclusion, vulnerabilidad que permite la inclusión de archivos locales en un servidor.

Logs

Registros de eventos, mensajes o transacciones generados por sistemas o aplicaciones.

Logstash

Herramienta de procesamiento de datos en tiempo real que ingiere, transforma y almacena datos para su posterior análisis.

Logz.io

Plataforma de monitoreo y seguridad basada en la nube construida sobre ELK y Grafana.

LXC

Linux Containers, tecnología de virtualización a nivel de sistema operativo para ejecutar múltiples sistemas Linux aislados.

M**Machine Learning**

Un campo de la inteligencia artificial que utiliza algoritmos para permitir a las computadoras aprender automáticamente y mejorar a partir de la experiencia sin intervención humana explícita.

Malware

Software malicioso diseñado para dañar, explotar o deshabilitar computadoras y sistemas.

Microsoft Hyper-V

Tecnología de virtualización de Microsoft para crear y gestionar máquinas virtuales.

Microsoft Teams

Plataforma de comunicación y colaboración desarrollada por Microsoft.

N**NIDS**

Network-based Intrusion Detection System, sistema de detección de intrusos basado en red.

O**Obsidian**

Aplicación de toma de notas basada en Markdown con soporte para gráficos y plugins.

OISF

Open Information Security Foundation, organización que desarrolla Suricata.

OpenAI

Organización de investigación en inteligencia artificial, creadora de modelos avanzados como GPT.

Opensearch

Suite de búsqueda y análisis de código abierto derivada de Elasticsearch.

Oracle VM Virtualbox

Software de virtualización de código abierto para ejecutar sistemas operativos invitados.

Overleaf

Editor de LaTeX en línea que facilita la creación y publicación de documentos científicos.

P**PHP**

Lenguaje de scripting del lado del servidor, especialmente adecuado para desarrollo web.

Prometheus Alertmanager

Módulo de Prometheus para manejar alertas generadas por el servidor de monitoreo Prometheus.

Proxmox VE

Plataforma de virtualización de código abierto para gestionar máquinas virtuales y contenedores.

Proxy

Servidor intermedio que actúa como intermediario entre el cliente y el servidor final.

R**Ransomware**

Tipo de malware que cifra archivos y exige un rescate para devolver el acceso al usuario.

RFI

Remote File Inclusion, vulnerabilidad que permite la inclusión de archivos remotos en un servidor.

Rsyslog

Software de log en sistemas UNIX y Linux, utilizado para recolectar y enviar logs.

S**Security Onion**

Distro de Linux para monitoreo, detección y respuesta de seguridad.

Sensu

Plataforma de monitoreo de infraestructura y aplicaciones en tiempo real.

SMTP

Simple Mail Transfer Protocol, protocolo para el envío de correo electrónico.

SMURF Flood Attack

Envía ICMP Echo Request a la red de difusión con la dirección IP de la víctima, inundando su capacidad de respuesta.

Snort

Sistema de detección y prevención de intrusos de código abierto.

Snort++

Versión avanzada del IDS Snort con mejoras de rendimiento y nuevas características.

Splunk

Plataforma para buscar, monitorizar y analizar datos de máquina generados por tecnología.

SQL

Lenguaje de consulta estructurado utilizado para gestionar y manipular bases de datos relacionales.

SQLI

SQL Injection, técnica de inyección de código que permite la ejecución de comandos SQL.

SSH

Secure Shell (SSH) es un protocolo de red cifrado utilizado para la gestión segura de sistemas y la transferencia segura de datos a través de redes inseguras.

SSL

Secure Sockets Layer, protocolo para establecer enlaces cifrados en la red.

Sumo Logic

Plataforma de análisis de logs basada en la nube para monitoreo en tiempo real y seguridad.

Suricata

Motor de análisis de red de código abierto, que incluye IDS, IPS y monitoreo de seguridad.

T**TCP**

Protocolo de control de transmisión (Transmission Control Protocol) utilizado en Internet para garantizar que los datos lleguen correctamente a su destino.

TFG

Trabajo de Fin de Grado, proyecto final de estudios universitarios de grado.

THC Hydra

Herramienta de fuerza bruta para crackear contraseñas de varios protocolos de red.

TI

Tecnologías de la Información, conjunto de tecnologías usadas para la gestión y procesamiento de información.

TIC

Tecnologías de la Información y la Comunicación, conjunto de tecnologías relacionadas con la informática y la comunicación.

TLS

Transport Layer Security, sucesor de SSL para comunicaciones seguras en la red.

U**URI**

Uniform Resource Identifier, cadena de caracteres para identificar recursos en Internet.

URL

Uniform Resource Locator, tipo de URI que identifica direcciones de recursos en la web.

V**VM**

Virtual Machine, entorno de computación que simula un sistema físico mediante software.

VMware vSphere

Plataforma de virtualización de servidores de VMware para construir infraestructuras en la nube.

VPN

Red Privada Virtual que permite crear una conexión segura a otra red a través de Internet.

W

Watcher

Herramienta de Elasticsearch para crear reglas y alertas basadas en datos indexados.

Y

YAML

YAML Ain't Markup Language, formato de serialización de datos legible por humanos.

Z

Zeek

Plataforma de monitoreo de red de código abierto que ofrece análisis de tráfico de red detallado.