Universitat Oberta de Catalunya uoc.edu

Aplicativo Web para el análisis de vulnerabilidades de una red lan Institucional integrando herramientas open source

Juan Gabriel Espinoza Calle
Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
M6.151 - Hacking_MISTIC

Manuel Jesús Mendoza Flores
Víctor García Font

01/06/2021


Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Aplicativo Web para el análisis de vulnerabilidades de una red Institucional integrando herramientas open source*
Nombre del autor:	*Juan Gabriel Espinoza Calle*
Nombre del consultor/a:	*Manuel Jesús Mendoza Flores*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	*06/2021*
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*M6.151 - Hacking_MISTIC*
Idioma del trabajo:	*ES*
Palabras clave	*Análisis de Vulnerabilidades, herramientas open source, identificador CVE*

Resumen del Trabajo*:*

La finalidad del presente TFM es la de desarrollar una aplicación web que permita realizar un análisis de vulnerabilidades en una red de área local Institucional, integrando las herramientas open source: OpenVas, NMap y vFeed, especializadas en el escaneo y gestión de vulnerabilidades de seguridad, con el objetivo de presentar al usuario final un reporte de las vulnerabilidades detectadas que permitirá visualizar gráficamente los resultados encontrados.

El proyecto inicia con la introducción teórica de diferentes conceptos de seguridad de la información, continuara con la descripción de las herramientas a utilizase conjuntamente con el proceso de instalación y configuración, dicho proceso se realizara en un ambiente virtualizado en el sistema operativo Kali Linux, seguidamente se desarrolla la interfaz web en php que correrá sobre un servidor apache, al igual se genera los diferentes scripts que permiten recolectar la información relevante del escaneo que se ejecute en segundo plano de las herramientas open source definidas previamente. Finalmente se presenta un reporte creado bajo el motor Elasticsearch que mediante Logstash recolecta los logs de las vulnerabilidades detectadas y se muestra de forma gráfica en el panel de visualización de kibana.

En este sentido, nos centramos en un proceso de prevención, que permita al

usuario final prever posibles ataques o amenazas que pueden ser explotadas por un atacante al no corregir la vulnerabilidad detectada.

Abstract:

The purpose of this TFM is to develop a web application that allows an analysis of vulnerabilities in an Institutional local area network, integrating open-source tools: OpenVas, NMap and vFeed, specialized in the scanning and management of security vulnerabilities, in order to present to the end user a report of the vulnerabilities detected that will allow the results found to be graphically visualized.

The project begins with the theoretical introduction of different information security concepts, it will continue with the description of the tools to be used together with the installation and configuration process, said process will be carried out in a virtualized environment in the Kali Linux operating system, then The web interface is developed in php that will run on an Apache server, as well as the different scripts that allow to collect the relevant information from the scan that is executed in the background of the previously defined open source tools. Finally, a report created under the Elasticsearch engine is presented that through Logstash collects the logs of the vulnerabilities detected and is displayed graphically in the kibana display panel.

In this sense, we focus on a prevention process that allows the end user to anticipate possible attacks or threats that can be exploited by an attacker by not correcting the vulnerability detected.

Índice

1. Introducción 1
1.1 Contexto y justificación del Trabajo 1
1.2 Objetivos del Trabajo 1
1.3 Enfoque y método seguido 2
1.4 Recursos y limitaciones 2
1.5 Planificación del Trabajo 3
1.6 Breve sumario de productos obtenidos 4
1.7 Breve descripción de los otros capítulos de la memoria 5
2. Marco Teórico 6
2.1 Seguridad de la información 6
2.1.1 Pilares de la seguridad de la Información 6
2.2 Ciclo de vida de la seguridad 8
2.2.1 Análisis de riesgos 8
2.3 Vulnerabilidades 10
2.3.1 Conceptos 10
2.3.2 Clasificación de vulnerabilidades 11
2.3.3 Gestión de vulnerabilidades 12
2.3.4 Etiquetado e identificación de vulnerabilidades 13
2.3.5 Bases de datos de vulnerabilidades 13
2.3.6 Evaluación de vulnerabilidades 14
2.4 Detección de vulnerabilidades 15
2.4.1 Escáneres de vulnerabilidades 16
2.4.2 Clasificación de los escáneres 16
2.5 Herramientas para la detección de vulnerabilidades 17
2.5.1 Open Vulnerability Assessment System (OpenVAS) 18
2.5.2 Network Mapper (NMap) 20
2.5.3 vFeed 22
2.6 Herramientas de Desarrollo 24
2.6.1 PHP 24
2.6.2 Apache Server 24
2.6.3 Elastic Stack (ELK) 24
3. Marco Práctico 26
3.1 Instalación y configuración de Herramientas 26
3.1.1 Escenario Inicial 26
3.1.2 Entorno de desarrollo 28
3.1.3 Servidor Apache 29
3.1.4 PHP 30
3.1.5 ELASTIC SEARCH 31
3.1.6 LOGSTASH 33
3.1.7 KIBANA 34
3.2 Diseño del sistema 35
3.2.1 Integración de componentes: 35
3.2.2 Diagrama de clases: 36
3.2.3 Mapa de navegación 37
3.3 Desarrollo 38

3.3.1 Página inicio.php 38
3.3.2 Página login.php 38
3.3.3 Página análisis_default.php 39
3.3.4 Página análisis_opciones.php 40
3.3.5 Página procesar_análisis.php 40
3.3.6 Página reporte.php 44
3.3.7 Dashboard Kibana 46
3.4 Escenario de pruebas 47
3.5 Resultados 48
4. Conclusiones 52
5. Glosario 54
6. Bibliografía 56
7. Anexos 58

Lista de figuras

Figura 1: Pilares de la seguridad de la información 7
Figura 2: Ciclo de vida de seguridad de la información 8
Figura 3: Relaciones que se crean cuando se habla de seguridad de la información 9
Figura 4: Relaciones entre ataque y amenaza 10
Figura 5: Ciclo de la seguridad 11
Figura 6: Vulnerabilidades clasificadas por tipo y año 12
Figura 7: Arquitectura de OpenVAS 18
Figura 8: Esquema vFeed 23
Figura 9: Escenario inicial 26
Figura 10: Interfaz Greenbone (OpenVas) 27
Figura 11: Entorno de desarrollo 29
Figura 12: localhost - Servidor Apache 30
Figura 13:Directorio /var/www/appWeb 30
Figura 14: PHPinfo() 30
Figura 15: Interfaz web kibana 35
Figura 16: Diagrama de componentes 35
Figura 17: Diagrama de clases 36
Figura 18: Mapa de navegación 37
Figura 19: Página inicio.php 38
Figura 20: Página login.php 39
Figura 21: Página anàlisis_default.php 39
Figura 22: Página anàlisis_opciones.php 40
Figura 23: Página processar_analisis.php 41
Figura 24: Fragmento de codigo – procesar_analisis.sh 41
Figura 25: Fragmento de código –anàlisis_nmap.sh 42
Figura 26: Fragmento de código – nmap_parser.py 42
Figura 27: Fragmento de código – analisis_openvas.sh 43
Figura 28: Fragmento de código – openvas_parser.py 44
Figura 29: Fragmento de código – vfeed_parser.pl 44
Figura 30: Página reporte.php 45
Figura 31: Página reporte_logs.php - NMap 45
Figura 32: Página reporte_logs.php - OpenVas 45
Figura 33: Página reporte_logs.php - vFeed 46
Figura 34: Dashboard Kibana 46
Figura 35: Fragmento de código – pipeline.conf 47
Figura 36: Escenario de pruebas 47
Figura 37: Información aportada por herramienta 48
Figura 38: Sistemas operativos por host 49
Figura 39: Amenazas detectadas por host 49
Figura 40: CVE asociada por host 50
Figura 41: NVT por host 50
Figura 42: Vulnerabilidades por puerto y host 51

1. Introducción

1.1 Contexto y justificación del Trabajo

En la actualidad la información es el activo más importante de las empresas o instituciones, debido a ello buscan implementar políticas de seguridad que permitan salvaguardarla y evitar comprometer la integridad, disponibilidad o confidencialidad de la misma.

En tal contexto es importante tener en cuenta que, dada la actual complejidad de los sistemas informáticos, resulta prácticamente imposible disponer de un sistema libre de vulnerabilidades y amenazas, en esta línea, el proceso de seguridad se suele percibir como un ciclo, donde se aplican medidas de prevención, detección y reacción.

Como medidas de prevención en el mercado existen herramientas open source que nos permiten mejorar la seguridad de los sistemas informáticos ya que están enfocadas a la detección de anomalías que pueden derivar en problemas para la seguridad del sistema, tales como OpenVas, NMap y vFeed, de las cuales se pretende integrar en un aplicativo web para aprovechar sus diferentes capacidades para detectar vulnerabilidades que permita al usuario final obtener información y valorar el nivel de seguridad existente en la Institución para luego proponer cambios o inclusión de medidas de seguridad que eviten que las vulnerabilidades puedan ser explotadas por atacantes informáticos, y proteger la integridad, disponibilidad y confidencialidad tanto de la información como de los dispositivos conectados a la red, mejorando así la situación actual en lo que a Seguridad Informática se refiere.

El presente documento se fundamenta en la necesidad práctica existente en empresas de modesta infraestructura y presupuesto, para las cuales la detección de vulnerabilidades basada en software libre, es una alternativa confiable y viable, que forma parte de una adecuada política de seguridad informática.

1.2 Objetivos del Trabajo

Objetivo General

Desarrollar un aplicativo web integrando herramientas open source para la detección de vulnerabilidades permitiendo presentar al usuario final un reporte grafico que le permita tomar decisiones para corregirlas o reducirlas, evitando que una vulnerabilidad pueda ser explotada por atacantes informáticos, protegiendo la integridad, disponibilidad y confidencialidad tanto de la información como de los dispositivos conectados a la red lan.

Para conseguir exitosamente el objetivo general de este proyecto, se deben cumplir antes los siguientes objetivos:

Objetivo Específicos:

- Desarrollar el aplicativo Web en PHP bajo un servidor apache.
- Instalar y configurar OpenVas, NMap y vFeed en un ambiente virtualizado en el sistema operativo Kali Linux.
- Generar scripts que permitan recolectar en segundo plano la información más relevante de las herramientas opensource.
- Mostrar resultados mediante un reporte grafico integrando Elasticsearch, Logstash y kibana.

1.3 Enfoque y método seguido

El enfoque del TFM consiste en realizar un escaneo de una red LAN Institucional, mediante un aplicativo Web que permita integrar herramientas open source para la detección de vulnerabilidades, con el fin de presentar al usuario final un reporte grafico de las vulnerabilidades encontradas, por lo tanto, se divide el proyecto en una parte teórica y otra practica:

- La *primera parte teórico* es referente a la investigación, pretende realizar un estudio del arte de seguridad informática, definiendo una serie de conceptos esenciales para el entendimiento de la memoria., Al igual que se hace un estudio conceptual de las herramientas open source a utilizar ya que conviene disponer de una base de conocimientos previo al inicio de la parte práctica.
- La *segunda parte práctica* consiste en la instalación y configuración de las herramientas OpenVas, NMap y vFeed en un ambiente virtualizado en el sistema operativo Kali Linux, se realizaran las pruebas pertinentes de cada una de las herramientas para validar su correcto funcionamiento, seguidamente se configurara un servidor Apache para el desarrollo del aplicativo web en el lenguaje de programación PHP, que presentara al usuario final una interfaz mediante el cual debe ingresar el rango de direcciones IP que corresponda a la red LAN Institucional a analizar, se crearan los diferentes scripts que generaran los logs con información relevante de la ejecución en segundo plano de las herramientas open source y finalmente se mostrara un reporte con el resultado de las vulnerabilidades detectadas de forma gráfica integrando Elasticsearch, Logstash y kibana.

1.4 Recursos y limitaciones

Recursos

Los recursos empleados para el desarrollo de este trabajo de fin de master (TFM) son los siguientes:

Recursos Materiales:

- Ordenador Portátil Acer Aspire E15 (2,4 GHz i5 / 8 Gb RAM / 500Gb HDD)
- Ordenador de sobremesa Acer Aspire M3970 (3,4 GHz i7 / 8Gb RAM / 500Gb HDD)
- Red LAN propia de mi hogar. (Router Huawey EchoLife HG8245H)

Recursos Humanos:

- El propio alumno.
- Profesor Tutor

Recursos Documentales

- Los accesibles a través de internet.
- Los accesibles a través de la biblioteca de la UOC.

Recursos Software

- Sistema operativo Windows 10 Pro 64b (Equipo anfitrión)
- Sistema operativo Kali Linux (Equipo virtualizado)
- VMWare Workstation Pro 16
- Office 2016 (Word, Excel, Project)
- OpenVas 8
- NMap 7.8
- vFeed
- Apache Server
- PHP 7.0
- Elasticsearch, Logstash y kibana.

Recursos Temporales:

- Media jornada compartida con 2 asignaturas de la UOC (Previsible 2/3 horas diarias aumentándolas los fines de semana).

Limitación

En el presente TFM nos centraremos solo en los escáneres de vulnerabilidades haciendo uso de herramientas activas open source que permiten detectar las vulnerabilidades de la red lan. No pretende presentar un análisis exhaustivo de todas las posibles vulnerabilidades de red que existen, También hay que tener en cuenta que la mayoría de las vulnerabilidades detectadas por un escáner no pueden ser reparadas por el propio escáner. Este se limita a proporcionar una serie de información e informes que, en la mayoría de los casos, requieren una intervención manual del administrador.

1.5 Planificación del Trabajo

Se dispone de un lapso de 3 meses para el desarrollo del TFM que se define en las siguientes tareas a realizar:

TFM - Hacking_MISTIC			
Entrega 1	**11,25 días**	**mié 17/02/21**	**mar 02/03/21**
Plan de Trabajo	11,25 días	mié 17/02/21	mar 02/03/21
Entrega 2	**22,5 días**	**mié 03/03/21**	**mar 30/03/21**
MARCO TEORICO			
Conceptos de seguridad informática	6,75 días	mié 03/03/21	mié 10/03/21
Conceptos de herramientas open source a utilizar	15 días	jue 11/03/21	mar 30/03/21
Entrega 3	**22,5 días**	**mié 31/03/21**	**mar 27/04/21**
MARCO PRACTICO			
Preparar el entorno de trabajo			
Instalación de kali Linux	3,38 días	mié 31/03/21	vie 02/04/21
Instalación y configuración de Herramientas Open source			
OpenVas	2,25 días?	vie 02/04/21	lun 05/04/21
NMap	4,5 días?	lun 05/04/21	jue 08/04/21
vFeed	3,38 días?	jue 08/04/21	lun 12/04/21
Desarrollo de la aplicación web			
Instalación y configuración de Servidor Apache con soporta para PHP	3,38 días?	mié 14/04/21	vie 16/04/21
Diseño de la página de inicio - Login	5,63 días?	vie 16/04/21	jue 22/04/21
Diseño de la interfaz de usuario	4,5 días?	jue 22/04/21	mar 27/04/21
Entrega 4	**28,13 días**	**mié 28/04/21**	**mar 01/06/21**
Creación de Scripts			
Script_OpenVas	4,5 días?	mié 28/04/21	lun 03/05/21
Script NMap	4,5 días?	mar 04/05/21	vie 07/05/21
Script vFee	2,25 días?	vie 07/05/21	lun 10/05/21
Pruebas de ejecución	3,38 días?	lun 10/05/21	mié 12/05/21
Generación de Reporte			
Instalación y configuración de Elasticsearch, Logstash y kibana	3,38 días?	mié 12/05/21	vie 14/05/21
Integración de logs	11,25 días?	vie 14/05/21	jue 27/05/21
Ejecución de pruebas	3,38 días?	vie 28/05/21	mar 01/06/21
Entrega 5	**5,63 días**	**mié 02/06/21**	**mar 08/06/21**
Presentación de video	5,63 días?	mié 02/06/21	mar 08/06/21

1.6 Breve sumario de productos obtenidos

El TFM se divide en un conjunto de entregables parciales que se corresponden con las fechas definidas en el plan docente, siendo las siguientes:

- **1era Entrega.** - Plan de Trabajo
- **2da Entrega.** - Marco teórico, en el cual se definen conceptos de seguridad informática y de las diferentes herramientas a utilizar.
- **3ra Entrega.** - Corresponde a la Instalación, configuración e Implementación de las diferentes herramientas, también se inicia el desarrollo de la aplicación web.

- **4ta Entrega.** - Se finaliza la aplicación web, al igual que se crean los scripts para la generación de logs con las vulnerabilidades encontradas que se integra al reporte que se presenta al usuario final.
- **5ta Entrega.** – Presentación en video del TFM

Como producto final de los diferentes entregables tenemos:

- Plan de trabajo
- Memoria
- Aplicativo Web
- Diferentes Scripts
- Video de presentación

1.7 Breve descripción de los otros capítulos de la memoria

Capítulo 1: Marco Teórico
Capítulo 2: Preparación del entorno de trabajo
Capítulo 3: Desarrollo de la aplicación Web
Capítulo 4: Creación de scripts y pruebas de integración
Capítulo 5: Generación de reporte de vulnerabilidades encontradas

2. Marco Teórico

2.1 Seguridad de la información

Según la ISO/IEC (2016), la seguridad de la información se podría definir como aquellos procesos, buenas prácticas y metodologías que busquen proteger la información y los sistemas de información del acceso, uso, divulgación, interrupción, modificación o destrucción no autorizada[1]. Esta definición básicamente significa que debemos proteger nuestros datos y nuestros recursos de infraestructura tecnológica de aquellos quiénes intentarían hacer un mal uso de ellos.

Por otro lado, en un sentido general, seguridad significa proteger nuestros activos. Esto puede significar protegerlos de atacantes que invaden nuestras redes, desastres naturales, condiciones ambientales adversas, cortes de energía, robo o vandalismo u otros estados indeseables. En última instancia, intentaremos protegernos contra las formas más probables de ataque, en la mejor medida que podamos, dado nuestro contexto.

2.1.1 Pilares de la seguridad de la Información

En la sociedad actual uno de los principales bienes a proteger, aquel que representa un mayor valor para los negocios, es la información, y de ahí la necesidad de protegerla, ya que el mundo está cada vez más conectado, y los ataques a las infraestructuras, redes y sistemas son cada vez más sofisticados. En general, el impacto de un ataque cibernético puede estar relacionado a los conceptos principales en seguridad de la información que son precisamente la confidencialidad, integridad y disponibilidad, comúnmente conocida como la tríada de la seguridad de la información (CIA) que se definen a continuación:

Confidencialidad: sólo las personas autorizadas tienen acceso a la información sensible y/o privada.

Integridad: la información y sus métodos de procesamiento son exactos y completos, y no pueden ser manipulados sin autorización.

Disponibilidad: los usuarios autorizados pueden acceder a la información cuando lo necesitan.

No obstante, en los últimos tiempos se consideran también otras dimensiones de la seguridad, contempladas por la propia legislación vigente:

Autenticidad y no repudio: existe garantía de la identidad de los usuarios o procesos que tratan la información, y de la autoría de una determinada acción.

Trazabilidad: es posible reproducir un histórico o secuencia de acciones sobre un determinado proceso y determinar quién ha sido el autor de cada acción.

[1] ISO/IEC 27000. (2016). Information technology–Security techniques–Information security management systems–Overview and vocabulary. https://www.iso.org/standard/66435.html.

También es necesario considerar la **privacidad**, que garantiza que sólo las personas autorizadas tienen acceso a información de carácter personal.

Figura 1: Pilares de la seguridad de la información
Fuente: Introducción a la seguridad de la información, PID_00275350, pág. 8 FUOC

Todas las dimensiones de seguridad son relevantes y deben ser tenidas en consideración por igual, aunque dependiendo del tipo de información que se esté tratando, tendrá mayor importancia una u otra.

Hay una cita muy conocida que dice: “El único sistema verdaderamente seguro es uno que está apagado, escondido en un bloque de hormigón y sellado en una habitación revestida de plomo con guardias armados…, y aun así tengo mis dudas[2].”

Definir el punto exacto en el que podemos ser considerados seguros presenta un desafío ¿Estamos seguros si nuestros sistemas están debidamente actualizados? ¿Estamos seguros si usamos contraseñas seguras? ¿Estamos seguros si estamos completamente desconectados de Internet? Desde cierto punto de vista, todas estas preguntas pueden responderse con un "no".

Incluso si nuestros sistemas están debidamente actualizados, siempre habrá nuevos ataques a los que seremos vulnerables. Cuando se utilizan contraseñas seguras, habrá otras vías que un ciberdelincuente puede aprovechar. Cuando estamos desconectados de Internet, se puede acceder físicamente a nuestros sistemas o ser robados si no implementamos seguridad física. En resumen, es muy difícil definir cuándo estamos realmente seguros.

Definir cuándo tenemos un ambiente inseguro es una tarea menos complicada ya que podemos enumerar una serie de elementos que nos pondrían en este estado:

- No actualizar sistemas operativos y aplicaciones.
- Usar contraseñas débiles.
- Descarga de programas de Internet de fuentes no seguras.
- Abrir archivos adjuntos de correo electrónico de remitentes desconocidos.
- Usar y desplegar redes sin cifrado, etc...

Podríamos seguir durante algún tiempo haciendo crecer esa lista, pero lo bueno es que una vez que seamos capaces de señalar los aspectos que

[2] Eugene Spafford, “Computer Recreations of Worms, Virusesand Code War”, Scientific American, marzo1998, p. 110

pueden causar que el entorno sea inseguro, se puede tomar medidas para mitigar estos problemas. Aunque es posible que nunca lleguemos a un estado que definitivamente podamos llamar "seguro", pero podemos dar pasos en la dirección correcta.

Todo ello lleva a la conclusión de que la seguridad no es un producto, sino que se trata de un proceso, una actividad que debe tener continuidad en el tiempo.

2.2 Ciclo de vida de la seguridad

El siguiente gráfico muestra cuál es el ciclo de vida correcto de la seguridad de la información:

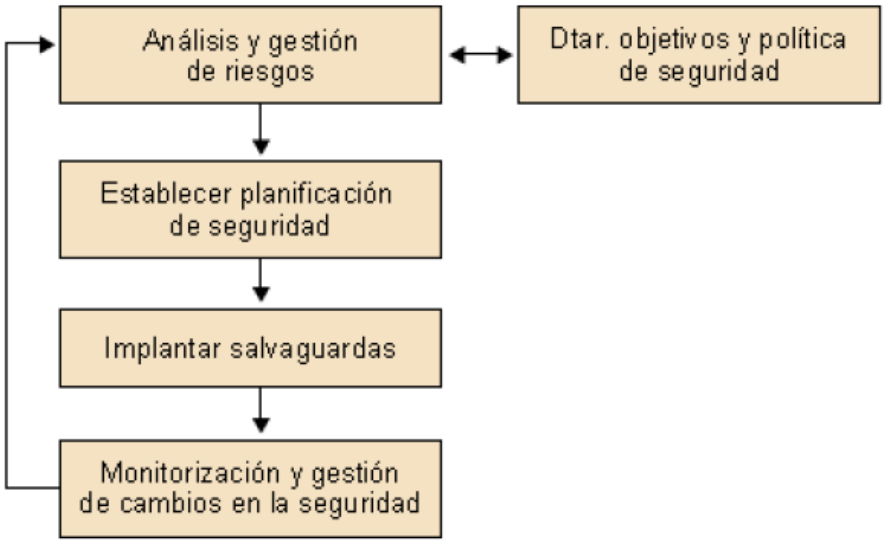


Figura 2: Ciclo de vida de seguridad de la información
Fuente: Análisis de riesgos, PID_00275346, pág. 7. FUOC

La primera y más importante de las fases corresponde al análisis de riesgos, que nos servirá para descubrir qué necesidades de seguridad tiene la organización tras detectar cuáles son nuestros agujeros en seguridad, así como las **vulnerabilidades** y **amenazas** a las que nos encontramos expuestos.

2.2.1 Análisis de riesgos

Un **análisis de riesgos** corresponde, desde el punto de vista de la seguridad, al proceso de identificación de éstos, determinando su magnitud e identificando las áreas que requieren medidas de protección.

De hecho, un análisis de riesgos nos va a permitir responder a las tres grandes preguntas:

• **¿Qué hay que proteger?** Se identifican los elementos que la organización debe tratar de asegurar.

• **¿De qué o de quién nos tenemos que proteger, y por qué?** Se identifican los peligros que pueden afectar a la organización y el motivo por el que podría producirse una incidencia.

• **¿Cómo nos queremos proteger?** Después de un análisis exhaustivo, se opta por la mejor protección para que los peligros no lleguen a materializarse.

Los elementos que se tienen en consideración en los procesos de análisis de riesgos son los siguientes:

• **Activos:** Son todos aquellos elementos que posee la organización y que serán analizados durante el proceso. Cabe destacar que por activo se entiende todo tipo de elemento que requiere la organización para poder realizar las actividades de negocio que le son propias.

• **Amenazas:** Son todas aquellas situaciones que podrían llegar a suceder en una organización y que podrían dañar a los activos, provocando que éstos no funcionen correctamente o que no puedan utilizarse del modo correcto para poder llevar a cabo la actividad de negocio de la organización.

• **Vulnerabilidades:** Son las diferentes debilidades que presentan los activos anteriormente identificados y que son aprovechados por las amenazas para provocar un daño.

• **Impactos:** Son las consecuencias que se producen en la organización cuando una amenaza aprovecha una vulnerabilidad para dañar a un activo.

A partir de éstos, se estiman los riesgos a los que se encuentra expuesta la organización. ***Los riesgos de seguridad cibernética se clasifican comúnmente como vulnerabilidades.*** Sin embargo, la vulnerabilidad y el riesgo no son lo mismo, lo que puede generar confusión.

El **riesgo** indica la probabilidad y el impacto de una vulnerabilidad que se explota. Si el impacto y la probabilidad de que una vulnerabilidad sea explotada es baja, entonces el riesgo es bajo. Inversamente, si el impacto y la probabilidad de que una vulnerabilidad sea explotada es alta, entonces existe un alto riesgo. De hecho, dentro de este proceso se crean una serie de relaciones que se indica en la siguiente figura:

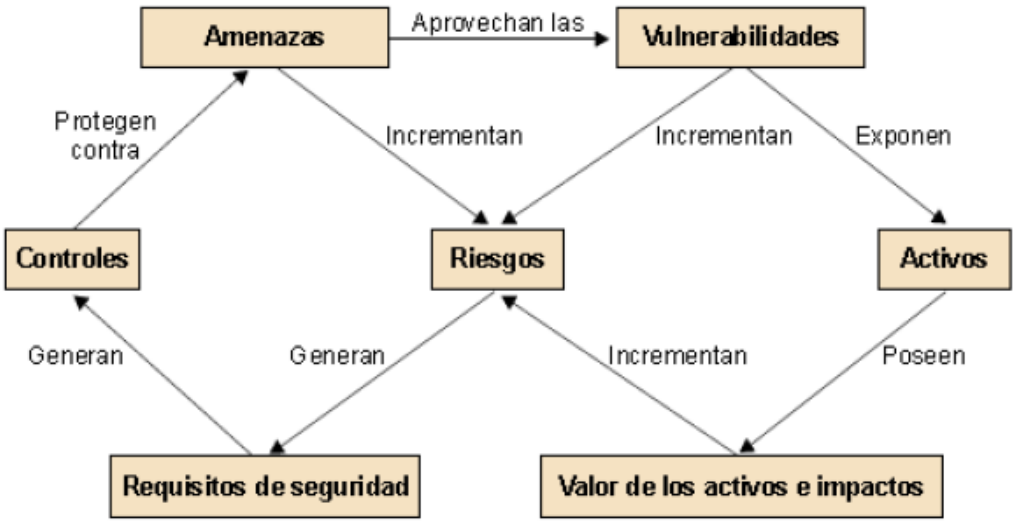


Figura 3: Relaciones que se crean cuando se habla de seguridad de la información
Fuente: Análisis de riesgos, PID_00275346, pág. 15. FUOC

Hay que tener en cuenta que una organización está expuesta a una serie de amenazas que son los causantes de los riesgos y a su vez de los posibles daños. Las amenazas aprovechan las vulnerabilidades para dañar los activos; de hecho, si no existen vulnerabilidades, las amenazas no podrán dañar a una organización.

2.3 Vulnerabilidades

2.3.1 Conceptos

Una **vulnerabilidad** de seguridad es un fallo o debilidad en el diseño, la implementación, la operación o la gestión de un sistema, que puede ser explotado con el fin de violar la política de seguridad del sistema[3].

Es importante distinguir entre ataque y amenaza. Un **ataque** es una acción intencionada realizada directa o indirectamente por un atacante al que se le atribuye cierta capacidad de acción inteligente. Por el contrario, una **amenaza** es la posibilidad de que ocurra una violación de la política de seguridad. Esta violación puede ser provocada por un ataque o por incidentes no deliberados causados de manera fortuita, como desastres naturales.

Una parte importante en la seguridad informática es **evaluar el riesgo** asociado a un servicio o sistema. Este riesgo suele ser directamente proporcional a la existencia de vulnerabilidades y amenazas. Aunque hay que tener en cuenta que no siempre a mayor número de vulnerabilidades mayor es el riesgo asociado a un sistema. El riesgo vendrá determinado también por la criticidad o gravedad de la vulnerabilidad.

Figura 4: Relaciones entre ataque y amenaza
Fuente: Introducción a las vulnerabilidades, PID_00255333, pág. 9. FUOC

Es importante tener en cuenta que, dada la actual complejidad de los sistemas informáticos, resulta prácticamente imposible disponer de un sistema libre de vulnerabilidades y amenazas. En esta línea, el proceso de seguridad se suele percibir como un ciclo, donde se aplican medidas de **prevención, detección** y **reacción**:

[3] Arribas, G. N. (2020). *Introducción a las vulnerabilidades - PID_00255333, pag8.*

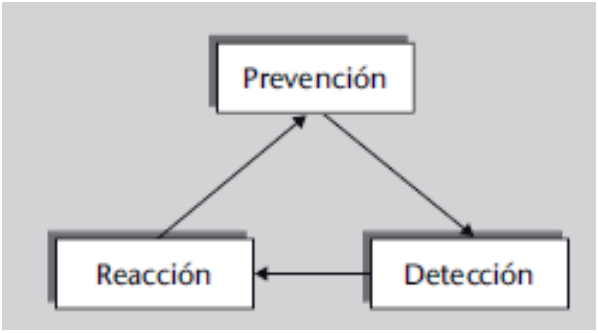


Figura 5: Ciclo de la seguridad
Fuente: Introducción a las vulnerabilidades, PID_00255333, pág. 10. FUOC

2.3.2 Clasificación de vulnerabilidades

Existen varias clasificaciones de vulnerabilidades y ninguna de ellas prevalece sobre el resto. El hecho de adoptar una clasificación u otra viene muchas veces condicionado por el propósito de dicha clasificación. En nuestro caso, y con el objetivo de proporcionar una visión global de la seguridad de la información, hemos adoptado una clasificación que se basa en el tipo de sistema al que afecta la vulnerabilidad. De esta manera, distinguimos entre:

Vulnerabilidades de bajo nivel y software malicioso. A aquí entran vulnerabilidades que afectan al sistema operativo y aplicaciones a bajo nivel propiciadas generalmente por errores en la programación, como buffer overflows o race conditions. También se incluye en este tipo de vulnerabilidades el estudio de software malicioso, como virus o gusanos que explotan este tipo de vulnerabilidades.

Vulnerabilidades de red. Son vulnerabilidades que afectan a software y componentes de red o interconexión de redes. Estas vulnerabilidades pueden ir desde redes locales a Internet. Principalmente, se observan vulnerabilidades en los diferentes protocolos de red, así como vulnerabilidades derivadas del análisis de tráfico.

Vulnerabilidades en aplicaciones web. La importancia de Internet y las aplicaciones web, desde el comercio electrónico, las redes sociales, o lo que actualmente se denomina como cloud computing, provoca que dichas aplicaciones merezcan un apartado propio. Estas son vulnerabilidades propias de aplicaciones pensadas para ser ofrecidas mediante una interfaz web y a las que generalmente tienen acceso un gran número de usuarios. Incluyen cros-site scripting, inyección de código, etc.

Vulnerabilidades de ingeniería social. Este apartado concentra las vulnerabilidades asociadas a los usuarios de los sistemas informáticos. Tienen más relación con el aspecto psicológico de dichos usuarios que con problemas puramente técnicos. Ejemplos son el spam, phising, etc.

Vulnerabilidades basadas en la identificación del servicio. Si tomamos la clasificación que tradicionalmente se aplica a los servicios de seguridad (confidencialidad, integridad y disponibilidad), tenemos vulnerabilidades que comportan:

- Pérdida de integridad.
- Pérdida de confidencialidad.
- Pérdida de disponibilidad.

Estimar el número de vulnerabilidades existentes es difícil, pero existen datos de, por ejemplo, el número de vulnerabilidades diferentes catalogadas, como las mostradas en la siguiente figura:

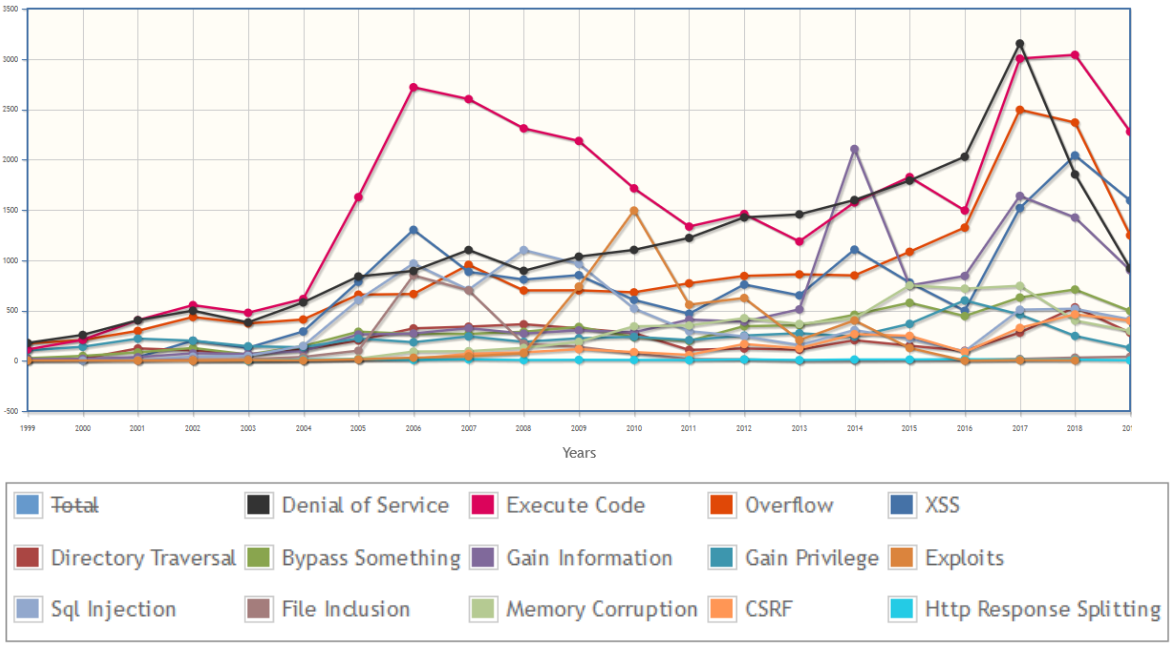


Figura 6: Vulnerabilidades clasificadas por tipo y año
Fuente: https://www.cvedetails.com/vulnerabilities-by-types.php

2.3.3 Gestión de vulnerabilidades

Los equipos encargados de la gestión de vulnerabilidades e incidentes de seguridad suelen recibir el nombre de **CERT** (*Computer Emergency Response Team*) o **CSIRT** (*Computer Security Incident Response Team*). La principal tarea de estos equipos es la gestión de incidencias de seguridad. En la práctica, sirven como medio para la difusión de vulnerabilidades de seguridad a usuarios (particulares u organizaciones), que suelen ser reportadas por los propios usuarios.

Los equipos CERT disponen de bases de datos y canales de distribución (como listas de correo) para distribuir información relativa a vulnerabilidades. Su principal tarea es recoger información sobre vulnerabilidades, clasificarlas y publicar su existencia, así como posibles medidas para mitigarlas. La información la suelen suministrar fabricantes, organizaciones o usuarios directamente a un CERT que luego suele distribuir información a otros equipos CERT con los que esté coordinado. Así mismo, la distribución puede ser directa y abierta al público, restringida a organizaciones concretas, o a suscriptores. En general, se considera una buena práctica difundir información de vulnerabilidades de modo abierto y gratuito a toda la comunidad de usuarios de Internet con el objetivo de mejorar la seguridad general de la Red. Por ello, la mayor de los CERT importantes distribuyen esta información libremente.

2.3.4 Etiquetado e identificación de vulnerabilidades

Para poder identificar vulnerabilidades, existen identificadores únicos que impiden que se publiquen duplicados y facilitan la posibilidad de hacer referencia a vulnerabilidades concretas. El sistema de identificación más importante a escala internacional es el **CVE** (***Common Vulnerabilities and Exposures***). CVE se presenta como un estándar de nombres de vulnerabilidades de seguridad informática de uso gratuito y público. Se autodefine como un diccionario de vulnerabilidades (no como una base de datos), donde cualquiera puede buscar el nombre (identificador) que recibe una vulnerabilidad concreta.

La importancia de CVE viene dada por su gran adopción a escala internacional. La mayoría de los equipos CERT, fabricantes de software y hardware, desarrolladores de sistemas operativos y organizaciones, así como productos destinados a la seguridad informática utilizan los identificadores CVE.

El **consejo editor de CVE** debe decidir entre otras cosas qué se considera como vulnerabilidad. Para ello, utiliza una definición propia. Según CVE, una vulnerabilidad es un estado de un sistema informático (o conjunto de sistemas) que cumple alguno de los siguientes casos:

- Permite a un atacante ejecutar comandos como otro usuario.
- Permite a un atacante acceder a datos violando las restricciones de control de acceso específicas para dichos datos.
- Permite a un atacante suplantar a otra entidad.
- Permite a un atacante llevar a cabo una denegación de servicio.

2.3.5 Bases de datos de vulnerabilidades

Actualmente hay varios organismos, fundaciones y empresas que se dedican a recoger, catalogar y grabar las vulnerabilidades conocidas y las dan a conocer públicamente a la comunidad, de manera que la información sobre vulnerabilidades se puede encontrar en varias bases de datos, repositorios y listas de distribución públicas open source o de iniciativa privada a través de la red. Algunos referentes en esta área son:

- **Security-Database** (http://www.security-database.com). Proporciona una base de datos de vulnerabilidades, basada en estándares abiertos para la clasificación, calificación, enumeración y explotación. También proporciona un repositorio de herramientas de seguridad y auditoría.

- **Common Vulnerabilities and Exposures (CVE)** (http://cve.mitre.org/). Es una lista de vulnerabilidades de seguridad, que tiene como objetivo proporcionar nombres comunes a problemas de conocimiento público.

- **SANS** (http://www.sans.org). Es una fuente de formación en seguridad, provee una base de datos de vulnerabilidades y también servicios de certificación en seguridad. Asimismo, desarrolla, mantiene y pone a disposición una colección de documentos de investigación sobre diversos aspectos de seguridad de la información.

- **SecurityFocus Vulnerability Database**. (https://www.securityfocus.com/) Base de datos mantenida por la empresa Symantec. Esta empresa también dispone de la lista de distribución BugTraq, que llegó a ser el principal canal de difusión de vulnerabilidades en los años noventa.

- **National Vulnerability Database (NVD)** (https://nvd.nist.gov/). Base de datos perteneciente al gobierno de Estado Unidos de acceso público.

- **Exploit DB.**(https://www.exploit-db.com/) Base de datos de vulnerabilidades que tiene la particularidad de que, además de publicar las vulnerabilidades, publica los exploits correspondientes (de ahí su nombre). Aunque sirven como prueba de concepto y para testear sistemas, su potencial uso malicioso es el motivo por el que el resto de las bases de datos y CERT no publican exploits.

Debemos tener en cuenta que cada base de datos tiene su propio sistema de codificación y de identificación de vulnerabilidades. Además, una misma vulnerabilidad se puede encontrar en varias bases de datos, pero en codificaciones diferentes, aun tratándose de la misma.

2.3.6 Evaluación de vulnerabilidades

Generalmente se define la evaluación de vulnerabilidades como el proceso de catalogar recursos, identificar, cuantificar y priorizar vulnerabilidades; En esta línea, el sistema de evaluación más extendido se conoce como **CVSS (Common Vulnerability Scoring System)**. CVSS es un intento de estandarizar una métrica común para evaluar vulnerabilidades. La idea es obtener un número (o conjunto de números) que nos den una idea del peligro potencial que supone una vulnerabilidad. CVSS distingue entre tres métricas básicas:

Métrica base, aspectos de la vulnerabilidad constantes en el tiempo y entorno descritos a continuación:

- **Vector de acceso (AV):** Cómo se explota la vulnerabilidad. Puede ser localmente (L), desde una red adyacente (A) o desde cualquier red (N).

- **Complejidad de acceso (AC):** Complejidad que requiere el atacante una vez ha accedido al sistema. Esta puede ser alta (H), media (M) o baja (L).

- **Autenticación (Au):** Número de veces que el atacante debe autenticarse contra un sistema. Pueden ser múltiples (M), una (S) o ninguna (N).

- **Impacto de confidencialidad (C), integridad (I) y disponibilidad (A):** Tres indicadores sobre el impacto que puede tener la vulnerabilidad en la confidencialidad, integridad y disponibilidad del sistema. Para cada uno los valores pueden ser: ninguno (N), parcial (P) o completo (C).

Estas métricas proporcionan un valor entre 0 y 10 que determina la gravedad de la vulnerabilidad. Esta se etiqueta como **low** (valor en [0,0,3,9]), **médium** (valor en [4,0,6,9]) o **high** (valor en [7,0,10,0]). La métrica base también se expresa como vector:

AV:[L,A,N]/**AC:**[H,M,L]/**Au:**[M,S,N]/**C:**[N,P,C]/**I:**[N,P,C]/**A:**[N,P,C].

Métrica temporal, métricas que pueden cambiar en el tiempo. Comprenden la explotabilidad (existencia de exploits y su grado de disponibilidad), nivel de curación o remediation level (existencia de soluciones y si son definitivas o temporales) y la confianza del anuncio (hasta qué nivel se ha confirmado la existencia de la vulnerabilidad). La métrica temporal se combina con la base para dar un valor entre 0 y 10.

Métricas del entorno, métricas relativas al entorno del sistema informático propiamente dicho, que incluye el riesgo que puede suponer a una organización o a personas individuales. Aquí se contempla el daño colateral potencial, que mide el daño que puede ocasionar a terceros la explotación de dicha vulnerabilidad (daño a personas, a bienes físicos, a la productividad o a los beneficios). También se incluye la distribución de objetivos o target distribution, que mide la proporción de sistemas vulnerables en el entorno. Estas métricas dan un valor entre 0 y 10, que generalmente se expresa como intervalo mínimo-máximo.

Existen aplicaciones que facilitan el cálculo del score CVSS, como las siguientes:

- CVSS SIG *(Common Vulnerability Scoring System SIG)* https://www.first.org/cvss/calculator/3.1
- NIST *(National Vulnerability Database)* https://nvd.nist.gov/Vulnerability-Metrics/Calculator-Product-Integration
- IPA, JVN *(Japan Vulnerability Notes)* http://jvnrss.ise.chuo-u.ac.jp/jtg/cvss/en/index.03.html

2.4 Detección de vulnerabilidades

Para descubrir los agujeros de seguridad en el sistema se hace uso de escáneres de vulnerabilidades. Esta es la herramienta más importante de todo el proceso, además de por ser la que aporta el grueso de información de la evaluación. Debemos realizar una distinción entre las herramientas que permiten detectar una vulnerabilidad y aquellas que permiten detectar un ataque. Si bien la distinción entre vulnerabilidad y ataque a menudo puede quedar diluida, en cuanto a su detección se considera que las herramientas que permiten la detección de vulnerabilidades quedan englobadas dentro de lo que se conoce como ***escáneres de vulnerabilidades***, mientras que aquellas herramientas que se utilizan para la detección de los ataques se incluyen dentro de la familia de ***sistemas de detección de intrusos***. Sin embargo, también puede suceder que un escáner de vulnerabilidades detecte un ataque,

ya que, por ejemplo, un ataque puede poner al descubierto o generar una nueva vulnerabilidad.

2.4.1 Escáneres de vulnerabilidades

Un **escáner de vulnerabilidades** es una aplicación diseñada para realizar análisis automáticos de cualquier aplicación, sistema o red en busca de cualquier posible vulnerabilidad. Aunque estas aplicaciones no son capaces de detectar la vulnerabilidad con total precisión, sí son capaces de detectar ciertos elementos que podrían desencadenar en una vulnerabilidad, facilitando enormemente el trabajo a los investigadores e ingenieros[4]. Los escáneres de vulnerabilidades incorporan, para cada vulnerabilidad, un conjunto de tests que deben permitir su detección.

El funcionamiento general de un escáner de vulnerabilidades se podría dividir en tres etapas:

1) Durante la **primera etapa** se realiza una extracción de muestras del conjunto de atributos del sistema para poder almacenarlas posteriormente en un contenedor de datos seguro.

2) En la **segunda etapa**, estos resultados son organizados y comparados con unas bases de datos de reglas y firmas que permiten identificar configuraciones inseguras.

3) **Finalmente**, se generará un informe con las diferencias entre ambos conjuntos de datos.

La principal ventaja de los escáneres de vulnerabilidades es que permiten la detección y solución de la vulnerabilidad antes de que esta pueda ser explotada para realizar un ataque. No obstante, hay que tener en cuenta que la mayoría de las vulnerabilidades detectadas por un escáner no pueden ser reparadas por el propio escáner. Este se limita a proporcionar una serie de información e informes que, en la mayoría de los casos, requieren una intervención manual del administrador.

2.4.2 Clasificación de los escáneres

Podemos clasificar los escáneres en función de sus habilidades de escaneo:

- **Escaneo interno y activo,** se refiere a la posibilidad de ejecutar el propio escáner dentro de la máquina que se pretende escanear. Esta característica permite un escaneo de datos de bajo nivel, como pueden ser servicios específicos de la máquina, detalles de su configuración, el propio sistema de ficheros, así como información específica del software y sistema operativo que utiliza. Permite analizar si las cuentas creadas en la máquina escaneada tienen contraseñas por defecto, o incluso si no tienen contraseñas. También permite verificar si el sistema ya ha sido atacado, analizando la existencia de ficheros sospechosos o programas en ejecución con privilegios inadecuados.

[4] Concepto: https://es.wikipedia.org/wiki/Esc%C3%A1ner_de_vulnerabilidades

- **Escaneo externo y activo,** se realiza mediante las herramientas conocidas como escáneres basados en red. En este caso, dicho escaneo se puede categorizar como un escaneo sin credenciales, en el sentido de que el actor que escanea un dispositivo no tiene acceso a él. En esta situación, la información obtenida del proceso de escaneo es una información semejante a la que puede ver un atacante. Se instala en una máquina que será la encargada de escanear distintos dispositivos de la red. A través de la red el escáner obtiene la información necesaria, mediante las conexiones que establece con el objetivo que hay que analizar. Permite la detección de cortafuegos mal configurados, servidores web vulnerables, riesgos asociados a software utilizado en los servidores, así como los riesgos asociados a una mala administración tanto de los servidores como de la red.

  Es muy importante tener en cuenta la topología de la red para el posicionamiento del escáner, de modo que el escaneo de los diferentes escáneres situados en distintos puntos de la red nos permita tener una idea clara de las vulnerabilidades de nuestro sistema dependiendo de desde dónde se accede.

- **Escaneo externo y pasivo,** combina las capacidades de escucha de los sniffers[5] con las capacidades de análisis de los escáneres de vulnerabilidades activos para detectar vulnerabilidades en los sistemas, se coloca en la red en una posición en la que se puede controlar el tráfico que viene de varios segmentos, El escáner pasivo escucha el tráfico en tiempo real y lo analiza mediante la comparación con un conjunto de reglas, como un escáner de vulnerabilidades activo, los escáneres pasivos son muy poco intrusivos y no afectan al rendimiento del sistema que se está escaneando, esta característica les permite ser utilizados en sistemas críticos en los que no se puede permitir una disminución del rendimiento o la eventual parada del sistema, que podría llegar a provocar un escaneo activo. Dado que los escáneres pasivos analizan el tráfico de la red, estos pueden detectar vulnerabilidades en la parte de la comunicación del cliente. Posibilitando el análisis de vulnerabilidades del cliente en un entorno cliente-servidor.

2.5 Herramientas para la detección de vulnerabilidades

Actualmente en el mercado tenemos una gran variedad de escáner de vulnerabilidades que están compuestos por otras herramientas para funcionalidades más específicas, cada uno de ellos se especializa en vulnerabilidades de un tipo en concreto. A continuación, se muestran unos ejemplos de las clases más usuales:

- **Escáner de vulnerabilidades de red:** Nessus, Qualys, Acunetix, OpenVAS, NMap, Nexpose, etc.

[5] **Sniffer**, del inglés sniff: olfatear, rastrear, puede entenderse como un programa con la capacidad de observar el flujo de datos en tránsito por una red, y obtener información de éste, https://en.wikipedia.org/wiki/Sniffer

- **Escáner de vulnerabilidades de aplicaciones web**: Nikto, Qualys, OWASP ZAP, w3af, Burp Suite, etc.

- **Escáner de vulnerabilidades de bases de datos:** Scuba, AppDetectivePro, McAfee Vulnerability Manager for Databases, AuditPro Enterprise, Microsoft Baseline Security Analyzer, etc.

Estas se clasifican en dos grupos: los que requieren de la adquisición de un acuerdo de licencia o pago, y los que son de código abierto o no requiere de pago alguno por su uso. Dentro de la primera categoría podemos encontrar a aplicaciones como NESSUS o SAINT, y en la segunda categoría podemos ubicar a OpenVAS o NMap, los cuales son ampliamente conocidos y difundidos en el medio. En nuestro caso nos enfocaremos en los escáneres distribuidos bajo el concepto de **código abierto**.

2.5.1 Open Vulnerability Assessment System (OpenVAS).

El sistema de evaluación de vulnerabilidad abierto, denominado inicialmente GnessUs, es un marco de diversos servicios y herramientas que ofrecen una solución completa y potente de escaneo y gestión de vulnerabilidades[6]. OpenVAS es una herramienta principal de OSSIM[7], todos los productos que la componen son software libre y la mayoría de ellos son distribuidos bajo licencia GPL[8]. En la siguiente figura se muestra la arquitectura de OpenVAS y como interactúa cada uno de sus componentes.

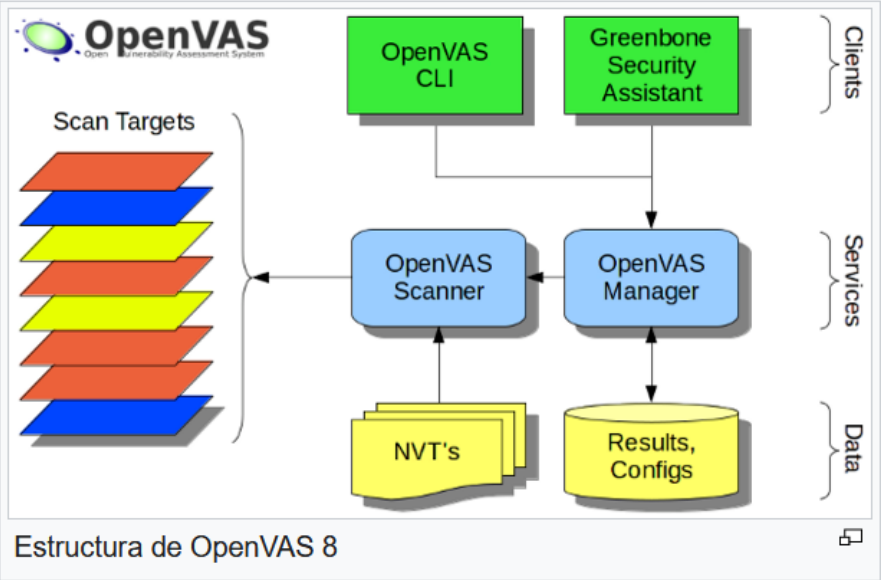


Figura 7: Arquitectura de OpenVAS
Fuente: https://es.wikipedia.org/wiki/OpenVAS

Como se observa en la figura 7, el núcleo es el **OpenVAS Scanner**, el cual ejecuta muy eficientemente las pruebas de vulnerabilidad de red (**NVTs**) que

[6] Definición de OpenVAS: https://es.wikipedia.org/wiki/OpenVAS
[7] OSSIM (Open Source Security Information Management), colección de herramientas bajo la licencia GPL, diseñadas para ayudar a los administradores de red en la seguridad de las computadoras, detección de intrusos y prevención.
[8] GNU *GPL* (General Public License) es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto.

son servidas a través de actualizaciones diarias mediante el servicio OpenVAS NVT Feed, los cuales son servidos con actualizaciones diarias mediante el **OpenVAS NVT Feed** o mediante el servicio comercial, con más de 30,000 de ellos en total. Sus características son:

- Escaneo concurrente de múltiples nodos.
- Protocolo de transferencia OpenVAS (OTP).
- Soporte SSL para OTP.
- Soporte WMI.

OpenVAS Manager es el servicio central que consolida el plan de escaneo de vulnerabilidades en una completa solución de gestión de vulnerabilidades. Toda la inteligencia se implementa en el Manager de modo que es posible implementar varios clientes ligeros que se integren consistentemente, por ejemplo, en relación con el filtrado o la clasificación de los resultados del análisis. OpenVAS Manager también controla una base de datos SQL (basada en sqlite) donde se almacenan de forma centralizada toda la configuración y los datos de resultado de la exploración. Por último, OpenVAS Manager también se encarga de la gestión de los usuarios, incluyendo el control de acceso con los grupos y roles. Sus características son:

- Protocolo de gestión OpenVAS (OMP).
- Permite tareas concurrentes de escaneo (varios OpenVAS Scanner).
- Notas de gestión para los resultados de escaneos.
- Gestión de falsos positivos.
- Escaneos programados.
- Reportes en múltiples formatos (XML, HTML, LaTeX, entre otros).
- Modo maestro – esclavo para controlar muchas instancias desde una únicacentral.
- Gestor de usuarios.

El asistente de seguridad **Greenbone (GSA)** es un servicio web ligero que ofrece una interfaz de usuario para los navegadores web. Sus características son:

- Cliente para OMP y el Protocolo de administración de OpenVAS (OAP).
- Soporte de HTTP y HTTPS.
- Servidor web propio (microhttpd), por lo tanto no requiere un servidor web adicional.
- Sistema de ayuda en línea integrada.
- Soporte multi-idioma.

OpenVAS CLI contiene la herramienta de línea de comandos que permite crear procesos por lotes para manejar el OpenVAS Manager. Otra de las herramientas de este paquete es un plugin de Nagios. Sus características son:

- Cliente para OMP.
- Corre en Windows, Linux, etc.
- Incluye plugin para Nagios.

2.5.2 Network Mapper (NMap).

Nmap ("mapeador de redes") es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes, aunque funciona muy bien contra equipos individuales. Nmap utiliza paquetes IP "crudos", es decir paquetes que no han sufrido ningún tipo de modificación, y por lo tanto son originales sea cual sea el protocolo utilizado para determinar qué equipos se encuentran disponibles en una red, qué servicios (nombre y versión de la aplicación) ofrecen, qué sistemas operativos (y sus versiones) ejecutan, qué tipo de filtros de paquetes o cortafuegos se están utilizando, así como docenas de otras características[9].

Es muy útil durante varios pasos de pruebas de penetración. Nmap no se limita a la mera recopilación y enumeración de información, sino que también es una potente utilidad que puede utilizarse como **detector de vulnerabilidades** o como **escáner de seguridad**.

Es una herramienta gratuita, de código abierto bajo licencia GPL, bien documentada, multiplataforma, disponible para consola, y que ofrece también una interfaz gráfica para facilitar su uso. Está escrita por un hacker conocido como Fyodor[10], y se beneficia de las aportaciones de una nutrida comunidad de colaboradores.

La información extraída con Nmap puede ser utilizada para múltiples usos. Los más habituales son los siguientes:

- Descubrimiento de subredes.
- Análisis de penetración de redes y equipos.
- Evaluación de la implantación de cortafuegos y de la eficacia de herramientas de detección y prevención de intrusiones.
- Descubrimiento del estado de puertos de comunicaciones.
- Descubrimiento de los servicios disponibles en un servidor, así como de sus versiones.
- Descubrimiento del tipo y versión del sistema operativo instalado en el equipo remoto.
- Obtención de información adicional acerca de servicios y equipos, a través de la ejecución de scripts convenientemente elaborados.

Incluye las siguientes herramientas:

Nping: generador de paquetes, analizador de respuestas y medidor de tiempos de respuesta. Permite generar paquetes de un gran rango de protocolos, permitiendo a los usuarios manipular virtualmente cualquier campo de las cabeceras de los protocolos.

Ncat: es una herramienta de red que transporta paquetes entre distintas redes. Se ha diseñado para ser una herramienta robusta que pueda proveer de conectividad a otras aplicaciones y usuarios.

---

[9] Concepto de NMap: https://nmap.org/man/es/

[10] Fyodor es el apodo de un conocido hacker, en honor al escritor ruso Fyodor Dostoyevsky, Ingeniero Informático y autor de Nmap así como de otras herramientas informáticas de seguridad.

Ndiff: herramienta para la comparación de diferentes análisis realizados por Nmap. A partir de los ficheros de salida de dos análisis diferentes sobre la misma red, muestra las diferencias existentes entre ellos.

Zenmap: interfaz gráfica multiplataforma y libre, soportada oficialmente por los desarrolladores de NMap. Su objetivo es facilitar a los principiantes el uso de la aplicación, mientras provee funcionalidades avanzadas para usuarios más experimentados.

En Nmap se definen las siguientes Técnicas[11]:

Técnicas de Descubrimiento de Equipos: determinación del estado de una o un conjunto de máquinas. Están encaminadas a averiguar el **estado** de una o varias máquinas, es decir, a realizar un proceso de determinación de cuales de los equipos indicados como Objetivos están activos, que equivale a decir "en línea", o remotamente alcanzables.

Técnicas de Escaneo de Puertos: determinación del estado de uno o un conjunto de puertos. Desde el punto de vista de una máquina remota, los puertos de comunicaciones tienen dos Estados: alcanzable e inalcanzable. Un puerto es **"alcanzable"** si no existe ninguna causa externa (p.ej. filtros intermedios) que evite el contacto entre los extremos. De este modo el origen tendrá información de si dicho puerto está a la escucha o está cerrado. Será **"inalcanzable"** en cualquier otro caso. Nmap distingue más Estados que se describen a continuación:

Estado Nmap de un puerto: Nmap define seis Estados distintos para recoger los distintos grados de incertidumbre en la determinación de si un puerto está abierto o cerrado, es decir, a la escucha o no de nuevas conexiones o paquetes. A diferencia del punto anterior, estos Estados no son propiedades intrínsecas de los puertos, sino que definen cómo son vistos desde el exterior. La razón de que Nmap sea considerado un escáner de puertos avanzado es, entre otros motivos, debido a esta granularidad en el Estado de dichos puertos. Los Estados son los siguientes:

- **Abierto:** existe una aplicación en la máquina objetivo que está a la escucha de nuevas conexiones o paquetes TCP o UDP.

- **Cerrado:** es un puerto alcanzable, pero no existe una aplicación a la escucha en él.

- **Filtrado:** Nmap no ha recibido respuestas a las sondas enviadas hacia un puerto. Suele significar que una herramienta intermedia (generalmente cortafuegos, sondas IDS/IPS, otros elementos de la red, o cualquier otro tipo de filtro) está bloqueando dicho puerto, respondiendo con poca o ninguna información.

- **No filtrado:** Solo aparece tras un análisis de tipo ACK[12]. Es un puerto alcanzable, pero no es posible determinar si está abierto o cerrado.

[11] Técnica: cada una de las funcionalidades, entendiendo funcionalidades como distintos tipos de escaneo, existentes en Nmap

- **Abierto | filtrado:** En este caso, Nmap no ha sido capaz de determinar si el puerto está abierto o filtrado debido a falta de respuestas.

- **Cerrado | filtrado:** Nmap no ha sido capaz de determinar si el puerto está cerrado o filtrado.

2.5.3 vFeed.

vFeed es una herramienta de código abierto que recopila información sobre vulnerabilidades utilizando multitud de fuentes confiables[13]; como:

- Estándares de seguridad:
  CVE (http://cve.mitre.org),
  CWE (http://cwe.mitre.org)
  CPE (http://cpe.mitre.org)
  OVAL (http://oval.mitre.org)
  CAPEC (http://capec.mitre.org)
  CVSS (http://www.first.org/cvss)

- Herramientas de explotación y auditoría de vulnerabilidades:
  Nessus (https://www.namicsoft.com)
  NMap, (https://nmap.org/)
  Metasploit (https://www.metasploit.com/)

- Páginas web y bases de datos de seguridad ofensiva:
  Exploit-DB (https://www.exploit-db.com/)
  Rapid7 (https://www.rapid7.com/db/)
  CXSecurity (https://cxsecurity.com/exploit/)

- Firmas de sistemas IDS, como:
  Snort (https://www.snort.org/)
  Suricata (https://suricata-ids.org/)

- Alertas de seguridad de fabricantes: Red Hat, Microsoft, Cisco, Debian, etc.

El resultado es una base de datos SQLite que agrega y relaciona toda esta información a través del identificador CVE de una vulnerabilidad determinada, de modo que contando, por ejemplo, con el identificador de un script de la herramienta OpenVas, es posible obtener toda la información disponible en el resto de fuentes de información: exploits disponibles en Metasploit o Exploit-DB, alertas de fabricantes relacionadas con la vulnerabilidad en cuestión, posibles firmas de sistemas IDS que nos permitan detectar un intento de explotación de la misma.

---

[12] ACK (del inglés acknowledgement, en español acuse de recibo o asentimiento), en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de esta para confirmar la recepción de un mensaje.

[13] vFeed: https://vfeed.io/how-it-works/#

El algoritmo de correlación de vFeed recopila y analiza una gran cantidad de macrodatos de fuentes de confianza, luego estandariza el contenido manteniendo una base de datos completa de vulnerabilidades y amenazas con respecto al esquema de nomenclatura estructurado CPE (Common Platform Enumeration). El esquema de operación se describe a continuación:

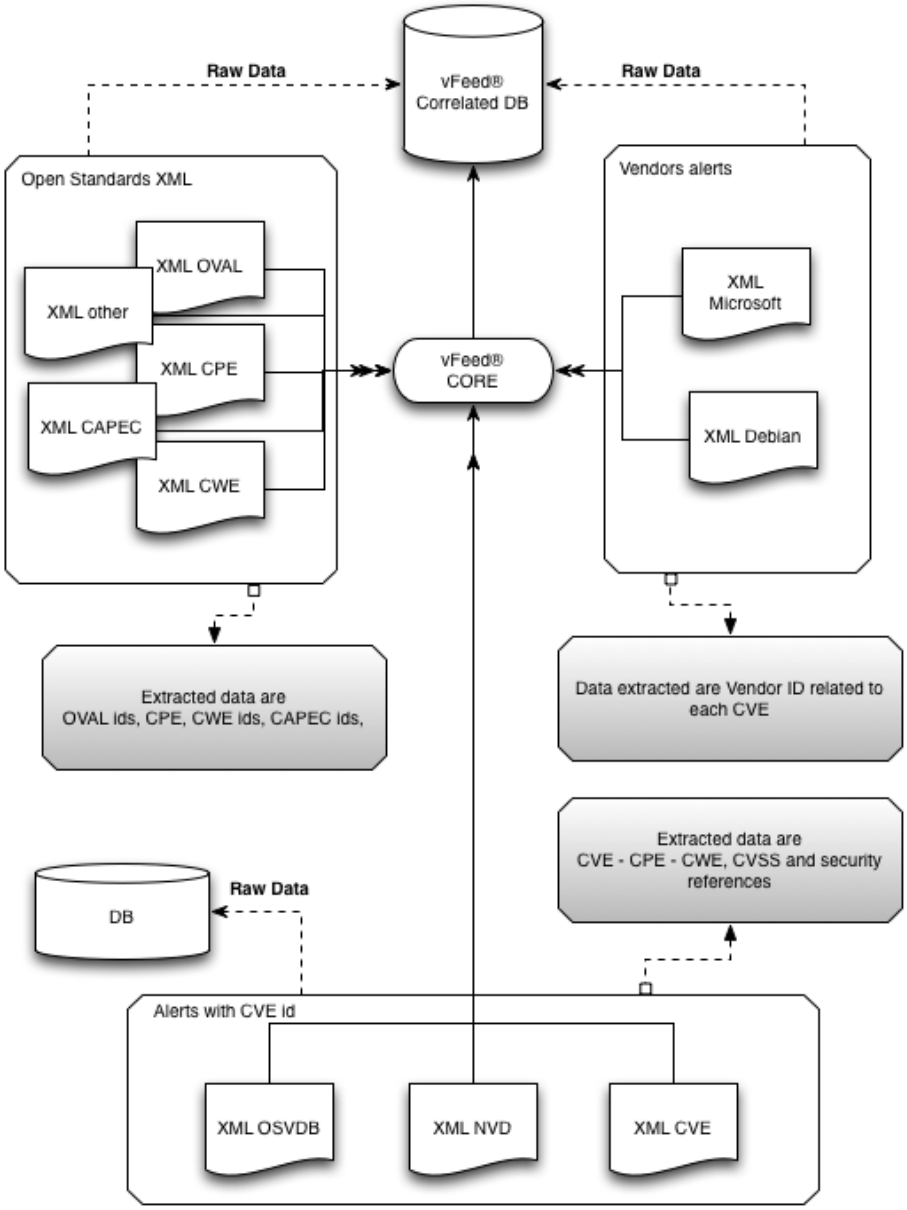


Figura 8: Esquema vFeed
Fuente: http://www.toolswatch.org/vfeed-the-open-source-correlated-cross-linked-vulnerability-xml-database/

Características:

- Utiliza el formato XML estructurado para describir las vulnerabilidades
- Provee una Base de datos de vulnerabilidades SQLite totalmente descargable para un ambiente de conexión local.
- La mejor solución para obtener información sobre la vulnerabilidad en un entorno fuera de línea.

2.6 Herramientas de Desarrollo

2.6.1 PHP

Son las siglas en inglés del acrónimo **Hypertext Pre-Processor**[14], es decir, pre-procesador de hipertexto, es un lenguaje de programación de propósito general que se ejecuta en el lado del servidor, es un lenguaje interpretado. Tiene múltiples formas de utilizarse, ya que puede utilizarse con scripts, de forma estructurada o programación en objetos. Está creado con la licencia de software libre PHPv3_01, que es una licencia Open Source.

PHP se utiliza principalmente para crear páginas web**,** con contenido dinámico, Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos.

2.6.2 Apache Server

Apache es un popular servidor web multiplataforma de fuente abierta que, permite servir contenido a las peticiones que vienen desde los clientes web (navegadores). Según los números, es el servidor web más popular que existe. Es activamente mantenido por Apache Software Foundation[15].

Entre las principales características de Apache, se encuentran las siguientes:

- Es gratuito y de código abierto.
- Instalación y configuración sencilla
- Altamente extensible y adaptable mediante módulos
- Funciones incorporadas para autentificación y validación de usuarios.
- Soporte para lenguajes como Perl, PHP y Python.

2.6.3 Elastic Stack (ELK)

Es una suite de código abierto que permite a los usuarios recoger, organizar y preparar datos con fines analíticos desde diferentes servidores (y en cualquier formato)[16]. Elastic Stack (**ELK**), es el nombre que se le otorgó al conjunto de productos: **E**lasticsearch, **L**ogstash y **K**ibana, que se describen a continuación:

- **Elasticsearch,** es el corazón del stack, siendo el motor de almacenamiento, búsqueda, procesamiento e indexación de datos, es gratuito y abierto para todos los tipos de datos, está desarrollado a partir de Apache Lucene. Se caracteriza por ser distribuido, lo que permite su crecimiento horizontal almacenando grandes cantidades de datos no estructurados mediante documentos JSON.

- **Kibana,** es una interfaz web escalable para la representación visual de datos que brinda histogramas en tiempo real, gráficos circulares y mapas. Kibana también incluye aplicaciones avanzadas, como Canvas,

[14] PHP: https://www.php.net/manual/es/intro-whatis.php

[15] Apache Server: https://apache.org/

[16] Elastic Stack (ELK): https://www.elastic.co/es/what-is/elasticsearch

que permite a los usuarios crear infografías dinámicas personalizadas con base en sus datos, y Elastic Maps para visualizar los datos geoespaciales. En el panel de control (dashboard), las diversas visualizaciones interactivas pueden combinarse para formar una imagen general dinámica que permita su filtrado y examen.

- **Logstash,** permite llevar información desde cualquier origen a Elasticsearch, se describe como un **ETL** (acrónimo del concepto Extract Transform, Load). Por ejemplo, si los sistemas almacenan su información en bases de datos estructuradas como: Oracle, SQL Server, MySQL o cualquier otro, Logtash puede recolectar y cargar la información para que puedas buscarla, analizarla y visualizarla en tiempo real. También es posible extraer información de todo tipo de archivos y orígenes de datos, algunos ejemplos son archivos de logs, XML, csv, API-REST, entre otros.

3. Marco Práctico

3.1 Instalación y configuración de Herramientas

3.1.1 Escenario Inicial

La instalación de las diferentes herramientas se realiza en un ambiente virtualizado mediante VMware Workstation 16, como primera instancia se instala el sistema operativo Kali Linux, reléase 2021.1, para continuar con la instalación de las herramientas a utiliza en el presente TFM, como se indica en el siguiente escenario:

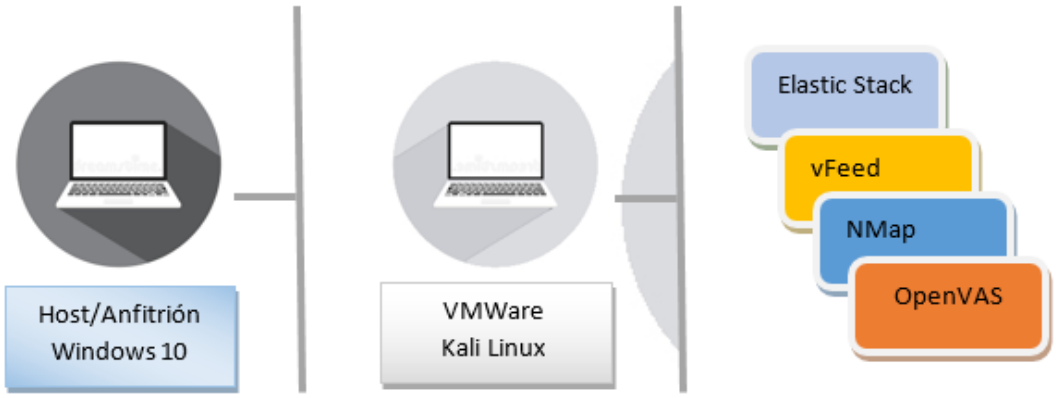


Figura 9: Escenario inicial
Fuente: Propia

OpenVAS

Previa la instalación de OpenVAS actualizamos los paquetes del sistema y validamos las nuevas actualizaciones de la distribución en general ejecutando los siguientes comandos:

apt-get update
apt-get dist-upgrade

```
┌──(root💀kali)-[~]
└─# apt-get dist-upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
```

Ejecutamos las dependencias de instalación: **apt-get install gvm***

```
┌──(root💀kali)-[~]
└─# apt-get install gvm*
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «gvm» para el global «gvm*»
Nota, seleccionando «gvm-tools» para el global «gvm*»
Nota, seleccionando «gvmd-dbgsym» para el global «gvm*»
Nota, seleccionando «gvmd» para el global «gvm*»
Nota, seleccionando «gvmd-common» para el global «gvm*»
```

Ejecutamos el instalador con el comando: **gvm-setup**

```
┌──(root💀kali)-[~]
└─# gvm-setup
Creating openvas-scanner's certificate files

sent 711 bytes  received 74,602,372 bytes  402,172.95 bytes/sec
total size is 74,582,270  speedup is 1.00
[*] Checking Default scanner
08b69003-5fc2-4037-a479-93b440211c73  OpenVAS  /var/run/ospd/ospd.sock  0  OpenVAS

[+] Done
[*] Please note the password for the admin user
[*] User created with password 'e8d57564-0c15-408e-bf8c-690cf252c8e0'.
```

En el proceso de instalación se realiza la actualización de la base de datos de los feed por lo que toma un tiempo considerado, dependiendo de la velocidad de conexión a internet.

Ejecutamos el comando para iniciar el servicio: **gvm-start**

```
┌──(root💀kali)-[~]
└─# gvm-start
[*] Please wait for the GVM / OpenVAS services to start.
[*]
[*] You might need to refresh your browser once it opens.
[*]
[*]  Web UI (Greenbone Security Assistant): https://127.0.0.1:9392
```

Automáticamente se inicializa la interfaz web en la siguiente url:

https://127.0.0.1:9392

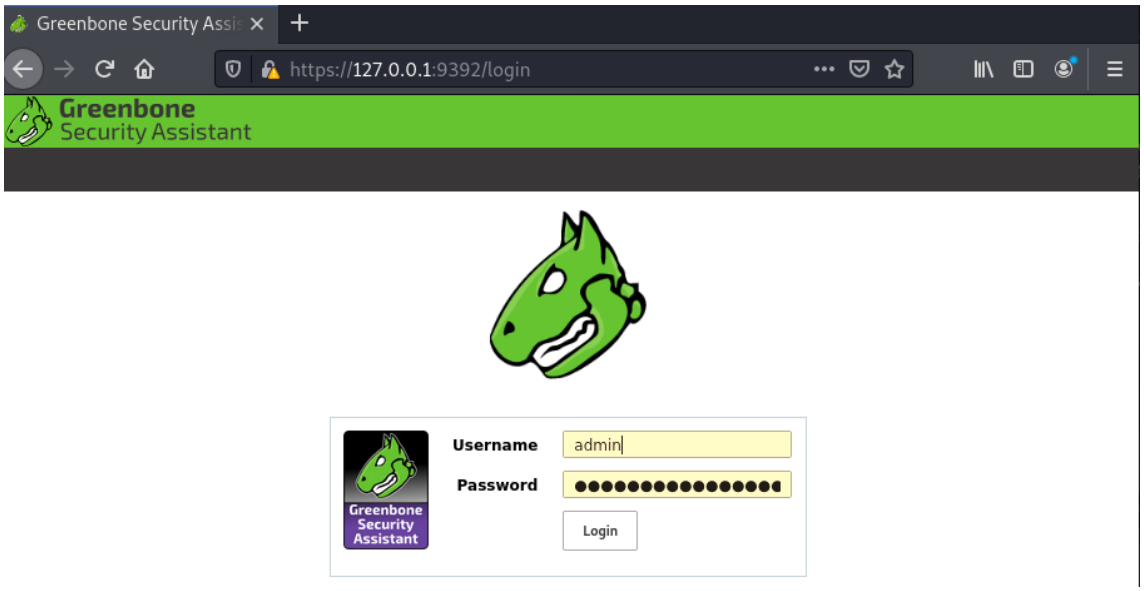


Figura 10: Interfaz Greenbone (OpenVas)
Fuente: Propia

* Nota: OpenVAS pasó a llamarse Greenbone Vulnerability Management (GVM) y ahora es solo una parte de él.[17]

[17] https://en.wikipedia.org/wiki/OpenVAS

NMap

Kali Linux ya integra NMap como parte de su repositorio de herramientas por lo que no hace falta su instalación, se realiza una prueba para validar su correcto funcionamiento:

Escaneo de la red (local), ejecutamos el comando: **nmap -sP 10.0.0.0/24**

```
┌──(root💀kali)-[~]
└─# nmap -sP 10.0.0.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-17 21:16 -05
Nmap scan report for SRV-SPS_UOC.mistic.com (10.0.0.1)
Host is up (0.00057s latency).
MAC Address: 00:0C:29:57:CB:1B (VMware)
Nmap scan report for debian.mistic.com (10.0.0.17)
Host is up (0.00037s latency).
MAC Address: 00:0C:29:92:ED:33 (VMware)
Nmap scan report for kali.mistic.com (10.0.0.15)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.05 seconds
```

Se verifica la ejecución satisfactoria de NMap.

vFeed

Para la instalación de vFeed descargamos el API del repositorio de GitHub mediante el siguiente comando:

git clone https://github.com/toolswatch/vFeed.git

```
┌──(root💀kali)-[~]
└─# git clone https://github.com/toolswatch/vFeed.git
Clonando en 'vFeed'...
remote: Enumerating objects: 578, done.
remote: Total 578 (delta 0), reused 0 (delta 0), pack-reused 578
Recibiendo objetos: 100% (578/578), 548.72 KiB | 1.25 MiB/s, listo.
Resolviendo deltas: 100% (315/315), listo.
```

Accedemos al directorio y verificamos su correcto funcionamiento:

```
┌──(root💀kali)-[~]
└─# cd vFeed

┌──(root💀kali)-[~/vFeed]
└─# ./vfeedcli.py -v
vFeed - The Correlated Vulnerability and Threat Intelligence Database API
0.7.2.1
```

3.1.2 Entorno de desarrollo

La aplicación se desarrolla en un ambiente cliente-servidor, el equipo cliente será el host anfitrión y el servidor el host virtualizado, en el que se instala y configura las diferentes aplicaciones, al igual que el medio de comunicación será la red LAN de hogar, como se indica en el siguiente entorno:

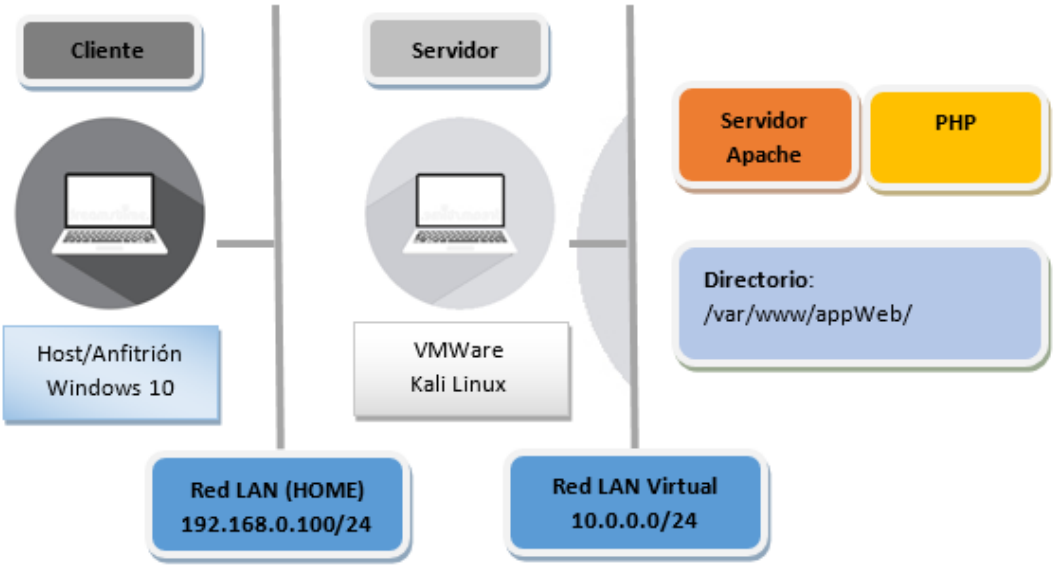


Figura 11: Entorno de desarrollo
Fuente: Propia

3.1.3 Servidor Apache

Para instalar el servidor apache, ejecutamos el siguiente comando:

apt-get install apache2

```
┌──(root💀kali)-[~]
└─# apt-get install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente (2.4.46-4).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

Habilitamos el servicio e inicializamos el servidor apache:

systemctl enable apache2
systemctl start apache2

```
┌──(root💀kali)-[~]
└─# systemctl enable  apache2
Synchronizing state of apache2.service with SysV service script with /lib/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /l
ib/systemd/system/apache2.service.

┌──(root💀kali)-[~]
└─# systemctl start apache2
```

Verificamos el correcto funcionamiento:

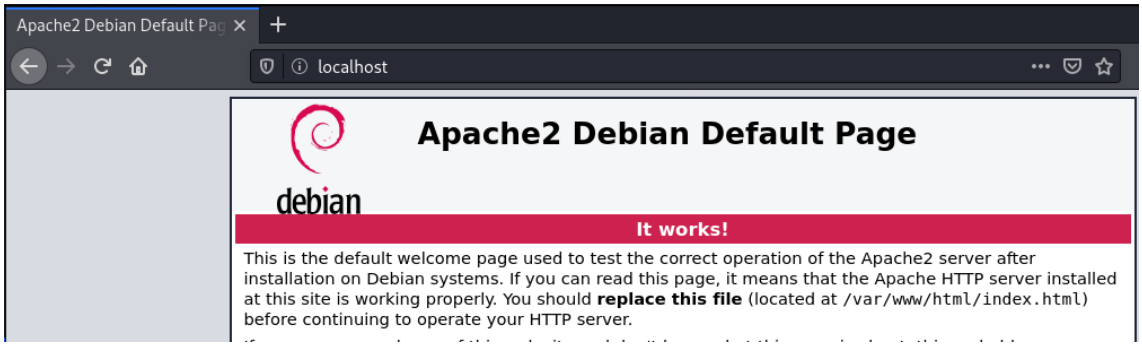


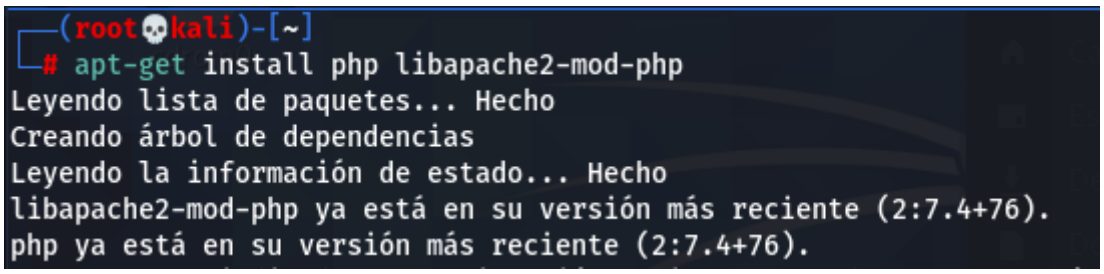
Figura 12: localhost - Servidor Apache
Fuente: Propia

Configuramos el directorio de trabajo, para lo cual editamos el archivo:
/etc/apache2/sites-available/000-default.conf

Figura 13:Directorio /var/www/appWeb
Fuente: Propia

3.1.4 PHP

Instalamos PHP ejecutando el siguiente comando:
apt-get install php libapache2-mod-php



Se verifica el correcto funcionamiento:

Figura 14: PHPinfo()
Fuente: Propia

3.1.5 ELASTIC SEARCH

Instalamos la herramienta de cifrado y firmas digitales gnupg, con el siguiente comando: **apt install -y gnupg**

```
┌──(root💀kali)-[~]
└─# apt install -y gnupg
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Instalamos la clave pública de elasticsearch para verificar las firmas de los paquetes del repositorio: **wget -qO- https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -**

```
┌──(root💀kali)-[~]
└─# wget -qO- https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key
add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (s
ee apt-key(8)).
OK
```

Creamos el archivo de repositorio y agregamos la ruta con el siguiente contenido:

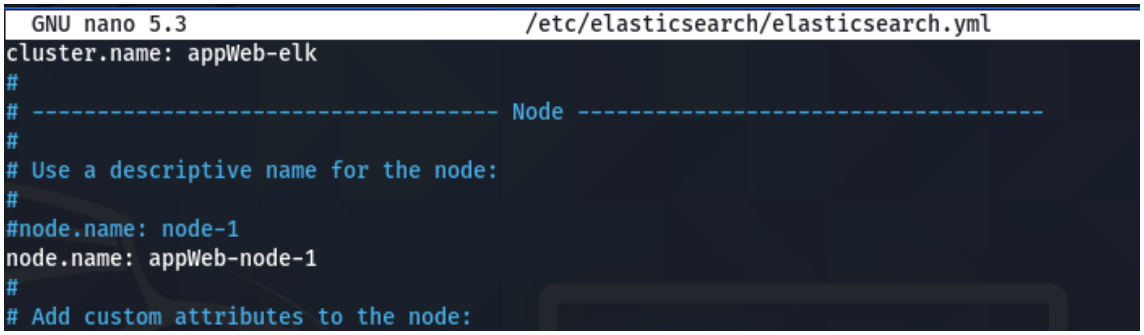
echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo tee /etc/apt/sources.list.d/elastic-7.x.list

```
┌──(root💀kali)-[~]
└─# echo "deb https://artifacts.elastic.co/packages/7.x/apt stable main" | sudo
tee /etc/apt/sources.list.d/elastic-7.x.list
deb https://artifacts.elastic.co/packages/7.x/apt stable main
```

Procedemos a instalar con el siguiente comando: **apt install -y elasticsearch**

```
┌──(root💀kali)-[~]
└─# apt install -y elasticsearch
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Preparando para desempaquetar .../elasticsearch_7.12.1_amd64.deb ...
Creating elasticsearch group... OK
Creating elasticsearch user... OK
Desempaquetando elasticsearch (7.12.1) ...
Configurando elasticsearch (7.12.1) ...
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Created elasticsearch keystore in /etc/elasticsearch/elasticsearch.keystore
Procesando disparadores para systemd (246.6-2) ...
```

Editamos el archivo de configuración **elasticsearch.yml** con los siguientes parámetros: **nano /etc/elasticsearch/elasticsearch.yml**



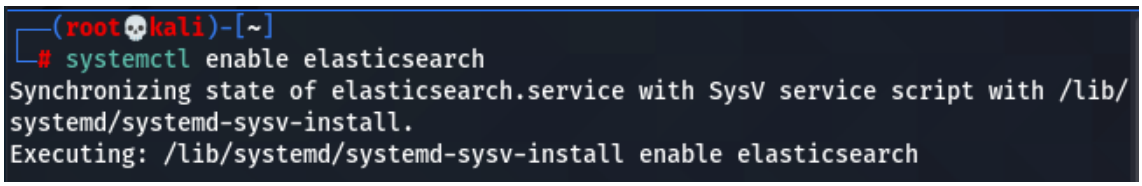
cluster.name: appWeb-elk

node.name: appWeb-node-1

network.host: localhost

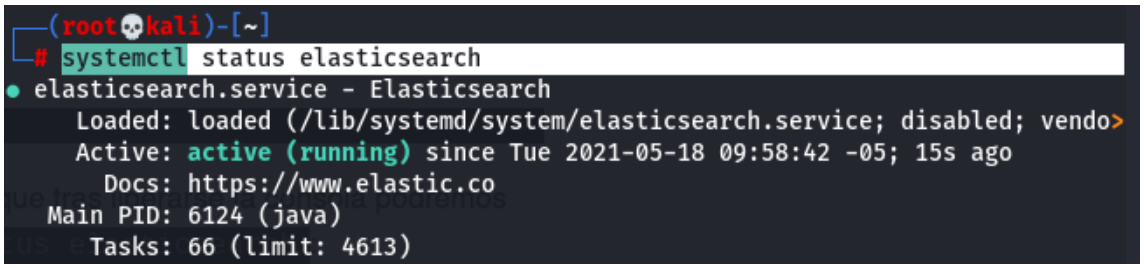
discovery.type: single-node

Habilitamos el servicio: **systemctl enable elasticsearch**



Iniciamos el servicio: **systemctl start elasticsearch**

Comprobamos que el servicio este habilitado: **systemctl status elasticsearch**



Verificamos desde la terminar que Elasticsearch se encuentra operativo:

curl -XGET http://localhost:9200/_cluster/health?pretty

```
┌──(root💀kali)-[~]
└─# curl -XGET http://localhost:9200/_cluster/health?pretty
{
  "cluster_name" : "appWeb-elk",
  "status" : "yellow",
  "timed_out" : false,
  "number_of_nodes" : 1,
  "number_of_data_nodes" : 1,
  "active_primary_shards" : 2,
  "active_shards" : 2,
  "relocating_shards" : 0,
  "initializing_shards" : 0,
  "unassigned_shards" : 1,
  "delayed_unassigned_shards" : 0,
  "number_of_pending_tasks" : 0,
  "number_of_in_flight_fetch" : 0,
  "task_max_waiting_in_queue_millis" : 0,
  "active_shards_percent_as_number" : 66.66666666666666
}
```

3.1.6 LOGSTASH

Al tener ya el repositorio y la clave publica de elasticserach, procedemos a la instalación de logstash, ejecutando el siguiente comando:

apt-get install logstash

```
┌──(root💀kali)-[~]
└─# apt-get install logstash
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
```

Habilitamos el servicio: **systemctl enable logstash**

```
┌──(root💀kali)-[~]
└─# systemctl enable logstash
Created symlink /etc/systemd/system/multi-user.target.wants/logstash.service → /etc/systemd/system/log
stash.service.
```

Iniciamos y verificamos el estado de logstach con los siguientes comandos:

systemctl start logstash

systemctl status logstash

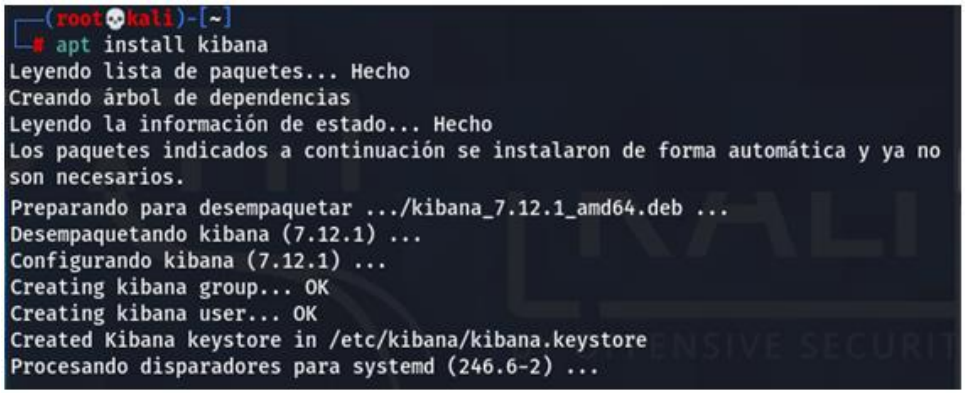
```
┌──(root💀kali)-[~]
└─# systemctl start logstash

┌──(root💀kali)-[~]
└─# systemctl status logstash
● logstash.service - logstash
     Loaded: loaded (/etc/systemd/system/logstash.service; enabled; vendor preset: disabled)
     Active: active (running) since Tue 2021-05-18 11:55:48 -05; 8s ago
   Main PID: 3441 (java)
      Tasks: 15 (limit: 4613)
     Memory: 334.3M
     CGroup: /system.slice/logstash.service
```

Los archivos de configuración de Logstash están escritos en el formato JSON y se alojan en el directorio **/etc/logstash/conf.d.**

3.1.7 KIBANA

Para la instalación de kibana ejecutamos el comando: **apt install kibana**

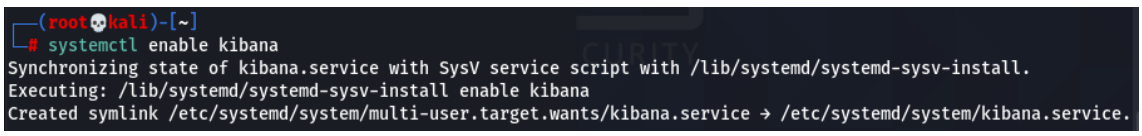


Editamos el archivo de configuración **kibana.yml** con los siguientes valores:

nano /etc/kibana/kibana.yml

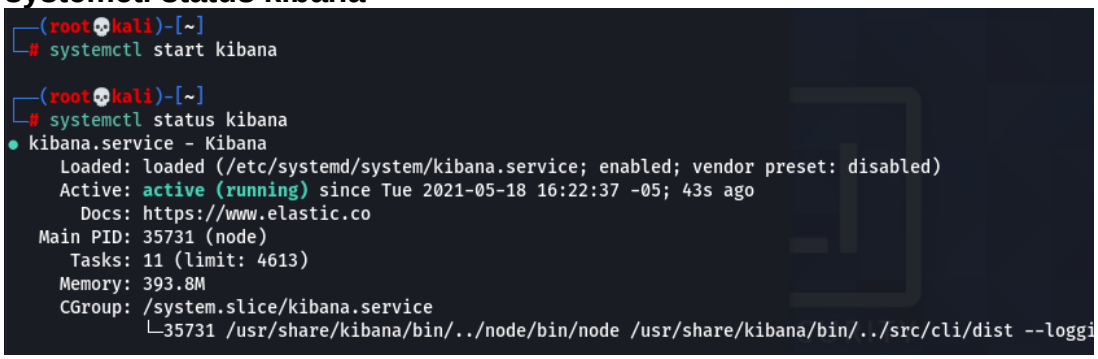
server.port: 5601
server.host: "localhost"
server.name: "appWeb-kibana"
elasticsearch.hosts: ["http://localhost:9200"]

Habilitamos el servicio de kibana: **systemctl enable kibana**



Iniciamos y verificamos qu el servico esten operativos:

systemctl start kibana
systemctl status kibana



Accedemos a la interfaz web de kibana: http://localhost:5601/

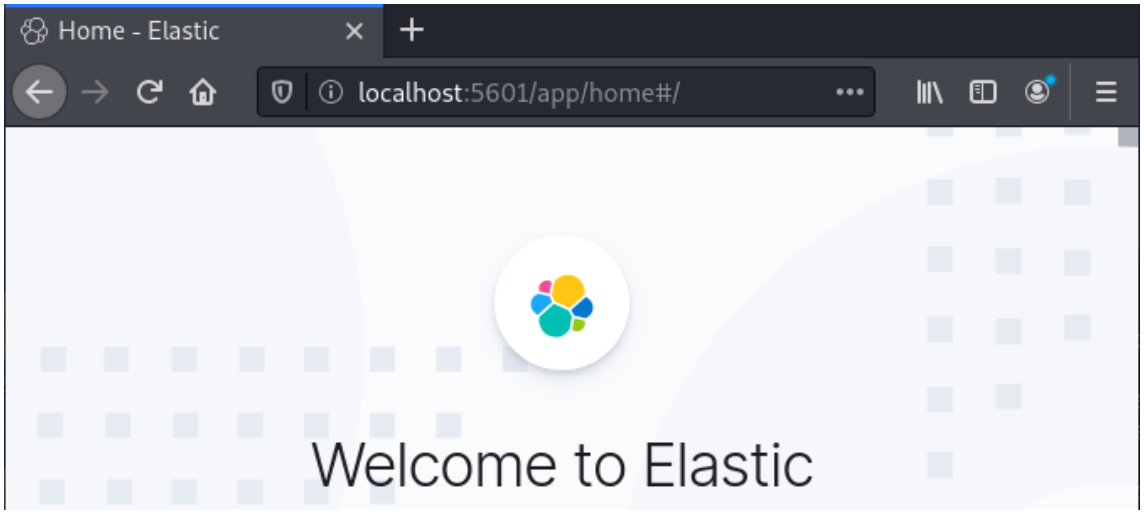


Figura 15: Interfaz web kibana
Fuente: Propia

3.2 Diseño del sistema

3.2.1 Integración de componentes:

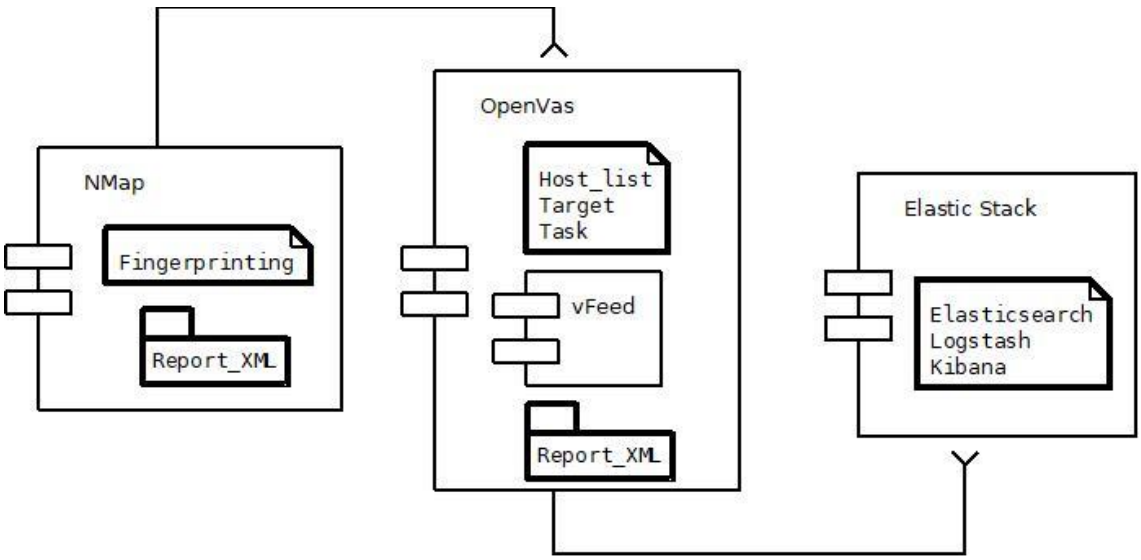


Figura 16: Diagrama de componentes
Fuente**: Propia**

En una primera instancia el componente NMap realiza un proceso de fingerprinting, con el objetivo de detectar los equipos host activos en la red, el resultado de este se almacena en un archivo XML cuya estructura será utilizada para realizar el parser a un log que contiene la información relevante de los host activos, este log alimenta al escáner de vulnerabilidades OpenVas, que mediante un Target y la creación de un Task se ejecuta por línea de comandos (CLI) el NVT (Network Vulnerability Tests), con el soporte de vFeed se recopila información de posibles vulnerabilidades detectadas, este proceso

genera un reporte XML, que también será parseado mediante consultas grook a logstash, finalmente se genera el reporte en la interfaz web kibana.

3.2.2 Diagrama de clases:

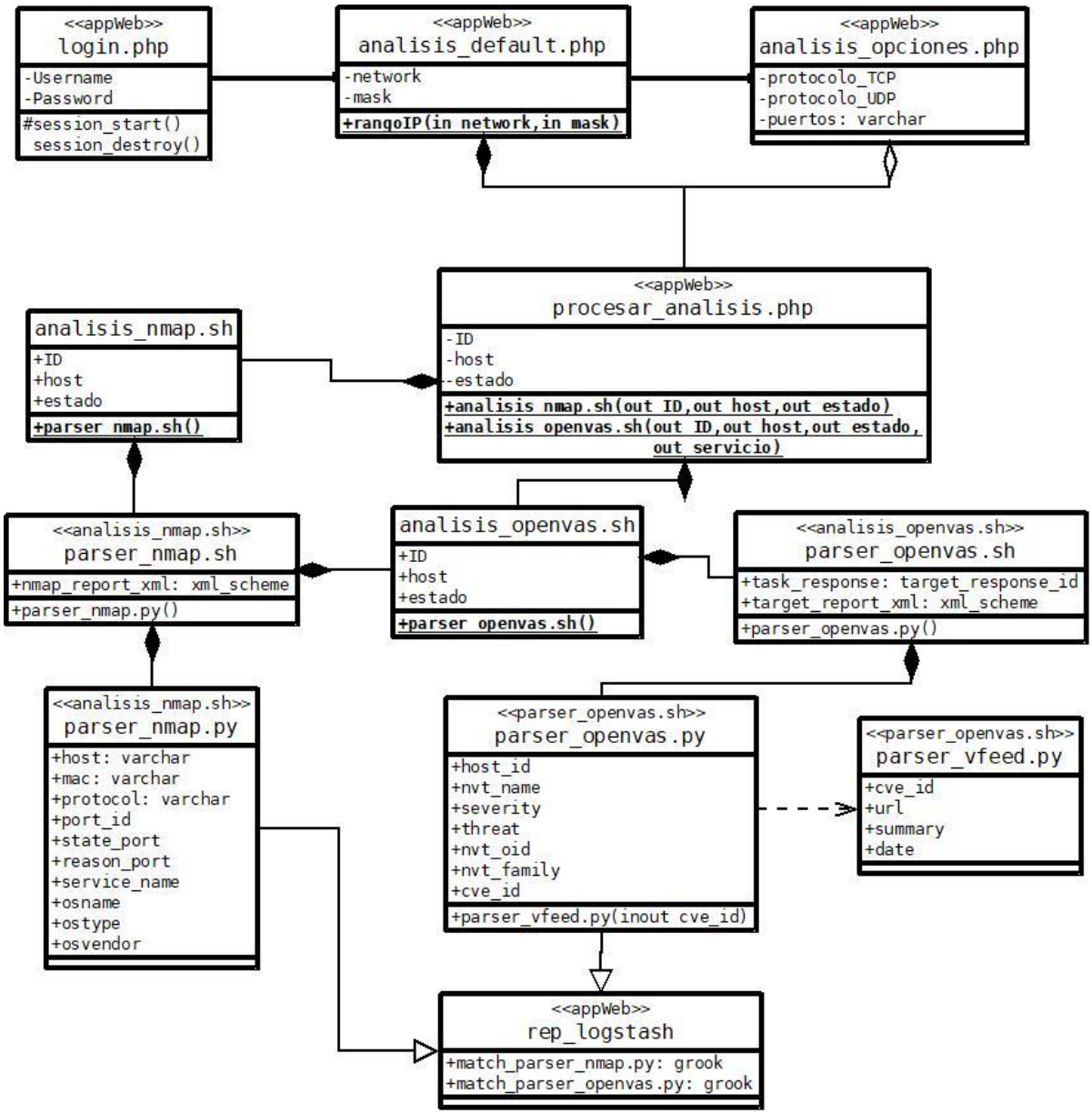


Figura 17 Diagrama de clases
Fuente**: Propia**

3.2.3 Mapa de navegación

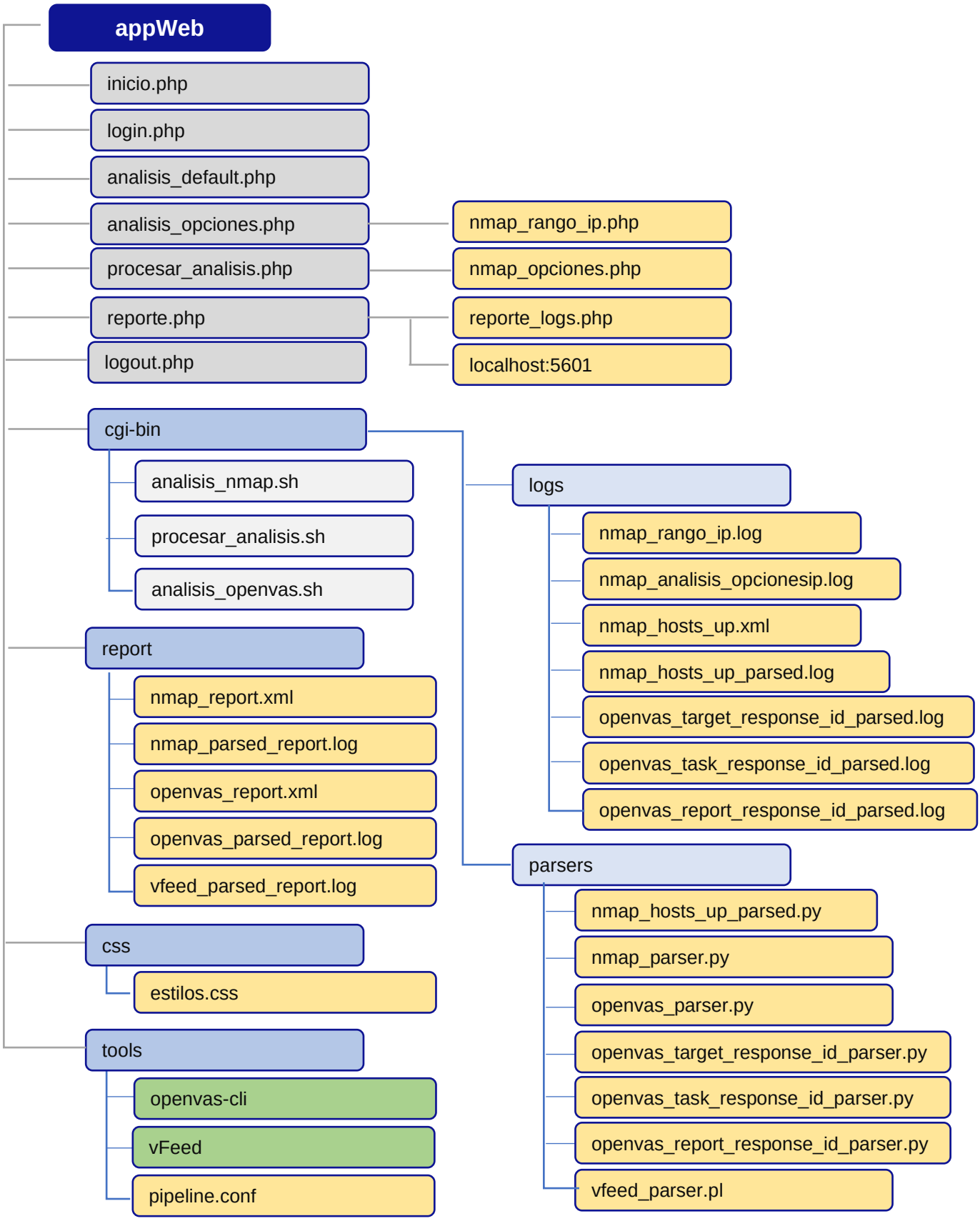


Figura 18: Mapa de navegación
Fuente**: Propia**

3.3 Desarrollo

Para el desarrollo del aplicativo se hace uso del IDE Visual Studio Code, los diferentes scripts de las herramientas open source se ejecutan mediante Shell code(.sh) incrustados en diferentes scripts, al igual que los parser de información se desarrollan en Python y Perl, los reportes se crean en un formato xml y en ficheros log.

3.3.1 Página inicio.php

Presenta información referente al TFM, el botón ingresar permite acceder al aplicativo:

Figura 19: Página inicio.php
Fuente**: Propia**

3.3.2 Página login.php

Solicita al usuario las credenciales de acceso al aplicativo con el objetivo de evitar que usuarios desautorizados tengan acceso a información que pueda comprometer la seguridad de la institución. El usuario predeterminado es: root y la contraseña: toor.

Figura 20: Página login.php
Fuente**: Propia**

3.3.3 Página análisis_default.php

Por defecto, se realiza un análisis de vulnerabilidades de los hosts activos de la red, se solicita al usuario que ingrese el numero de redes a analizar, se valida que el valor de la red ingresada corresponda al formato IPv4/mascara. Esta validación permite generar los parámetros necesarios para recopilar la información de los hosts activos en la red mediante el análisis que se ejecuta con la herramienta nmap.

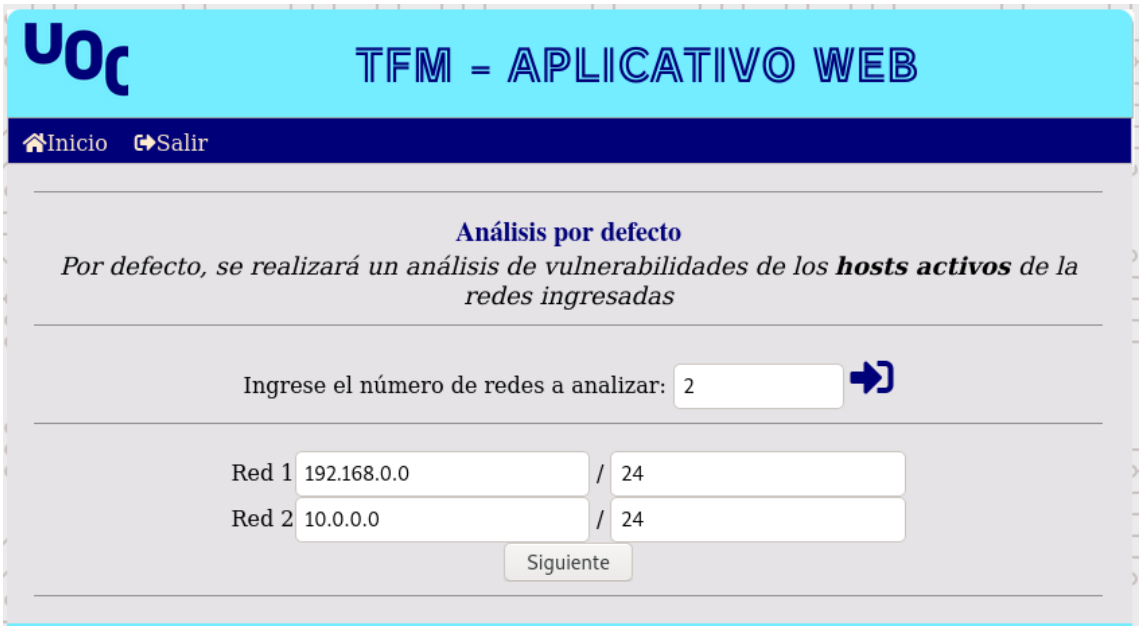


Figura 21: Página anàlisis_default.php
Fuente**: Propia**

nmap_rango_ip.php

Lee los valores para cada número de subredes introducidos por el usuario, al igual que el valor asociado a la máscara y lo define en el archivo ../appWeb/cgi-bin/logs/**nmap_rango_ip.log,** donde se escribe la información de las redes a analizar por nmap.

3.3.4 Página análisis_opciones.php

Permite al usuario seleccionar una o más de una opción para realizar un análisis de los puertos: TCP y UDP, obteniendo el ID, estado en el que se encuentra, la razón y el servicio que se ejecuta en él, además de realizar un análisis de los protocolos que se ejecutan en cada puerto. Con el objetivo de aportar con mayor información de los hosts activos para el análisis, teniendo en cuenta que el mismo será más ofensivo.

Figura 22: Página anàlisis_opciones.php
Fuente**: Propia**

nmap_opciones.php

Comprueba las opciones seleccionadas por el usuario para realizar el análisis (TCP, UDP, Análisis de protocolos) y lo define en el archivo ../appWeb/cgi-bin/logs/**nmap_analisis_opciones.log,** donde se escribe la información de las opciones seleccionadas.

3.3.5 Página procesar_análisis.php

Ejecuta el scripts shell: **procesar_analisis.sh,** que a su vez levanta los servicios y ejecuta los scripts de las herramientas nmap y openvas, mientras se visualiza un log con información del proceso, el análisis se realiza en segundo plano, dependiendo del rango de la red y las opciones seleccionadas por el usuario tomara un tiempo determinado la ejecución.

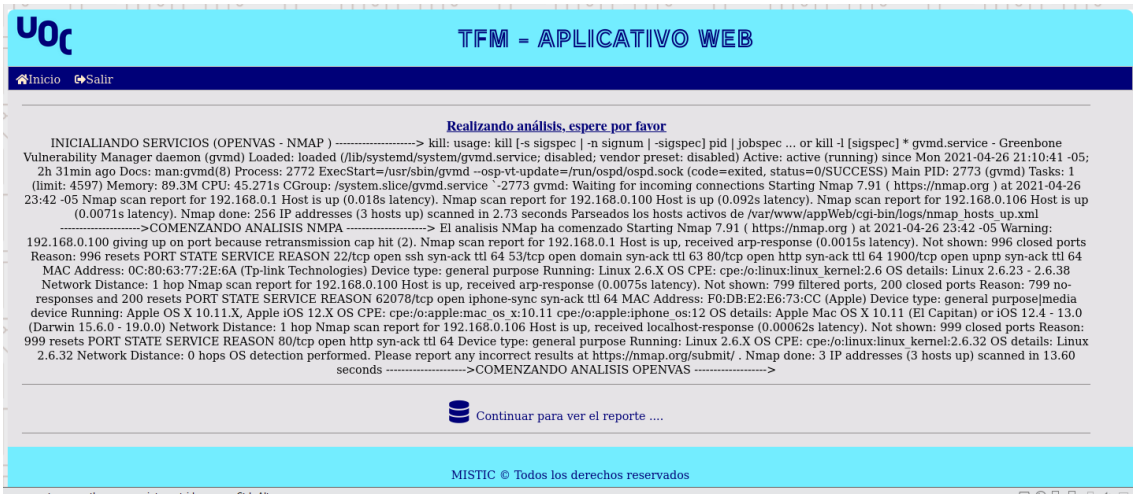


Figura 23: Página processar_analisis.php
Fuente**: Propia**

procesar_analisis.sh

En este script se borra los procesos asociados a llamadas anteriores al script e iniciamos los servicios de nmap y openvas, se elimina los archivos de logs residuales de análisis previos, leemos del archivo nmap_rango_ip.log, cada rango IP de uno en uno, comprobando si se han encontrado hosts activos ejecutando el comando: nmap -sP, mediante el archivo nmap_hosts_up_parser.py, parseamos los hosts que estén activos al fichero log: nmap_hosts_up_parsed.log

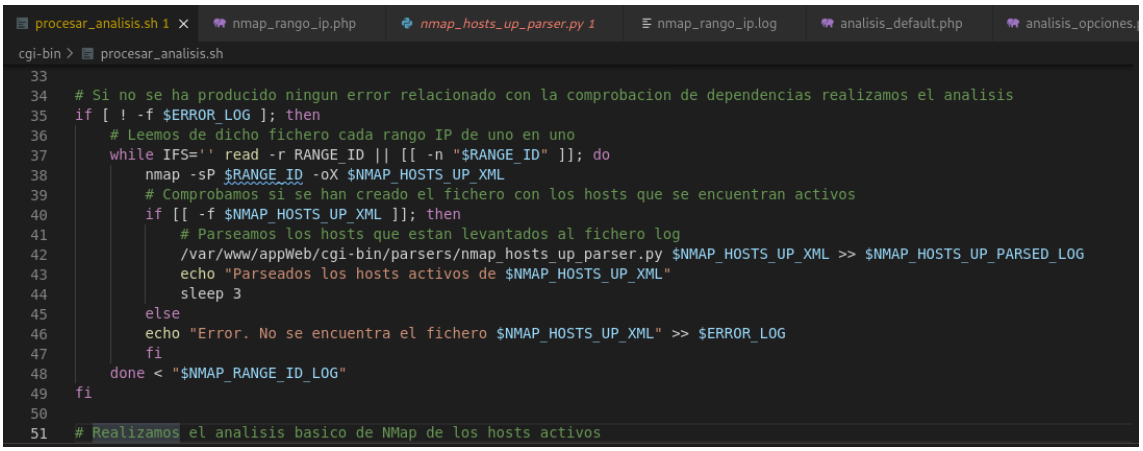


Figura 24: Fragmento de codigo – procesar_analisis.sh
Fuente**: Propia**

analisis_nmap.sh

Realiza el análisis nmap ejecutando el comando: nmap -T5-sV -O -iL. El resultado se almacena en el archivo: nmap_report.xml, parseamos la información obtenida para cada host mediante el script: nmap_parser.py y la escribimos en el archivo: nmap_parsed_report.log

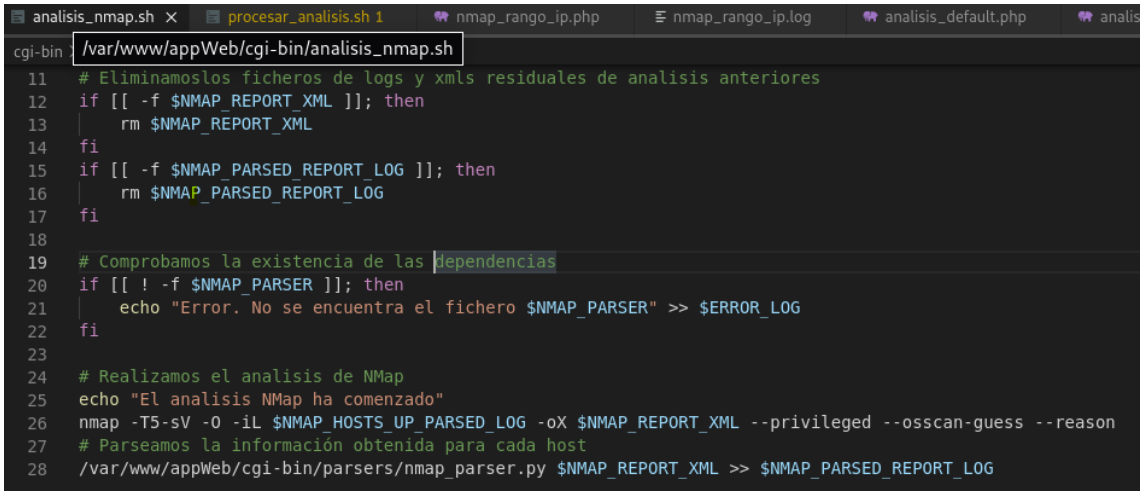


Figura 25: Fragmento de código –anàlisis_nmap.sh
Fuente**: Propia**

nmap_parser.py

Recabar la información relevante de los hosts activos, por cada etiqueta <host>, leemos el campo address, comprobamos si existe la etiqueta <port>, leemos los campos que nos interesan de cada puerto, en función de las opciones seleccionadas por el usuario, obteniendo los campos: host_id, protocol, port_id, state_port, reason_port, service_name.

También se comprueba si fue posible detectar el sistema operativo a través de la etiqueta <osmatch>, obteniendo los campos: name, vendor, osfamily, type, osgen.

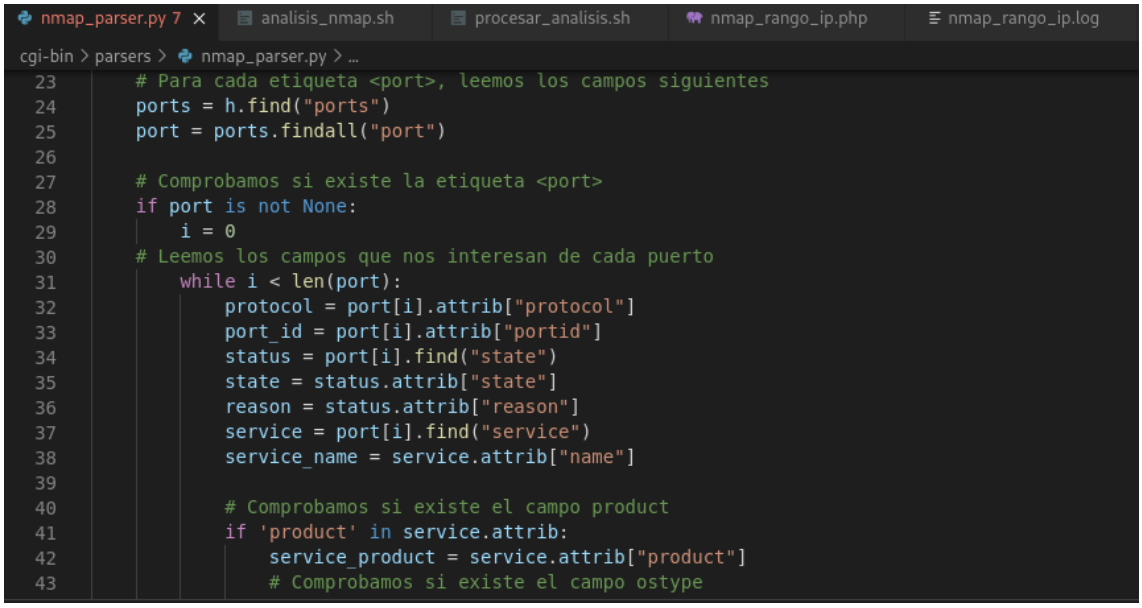


Figura 26: Fragmento de código – nmap_parser.py
Fuente**: Propia**

analisis_ openvas.sh

Define los parámetros necesarios para ejecutar el escaneo de vulnerabilidades de openvas, se establece el target por cada host activo en la red, parseamos el target_response id, necesario para crear la task, se crea la tarea de escaneo de acuerdo al objetivo preestablecido previamente (target), con el tipo de escaneo por defecto, parseamos el task_response id, necesario para la task, el resultado se escribe en un archivo en formato xml, parseamos la información obtenida para cada host, comprobamos el estado de la tarea que se está ejecutando para seguir cuando termine con el próximo host activo.

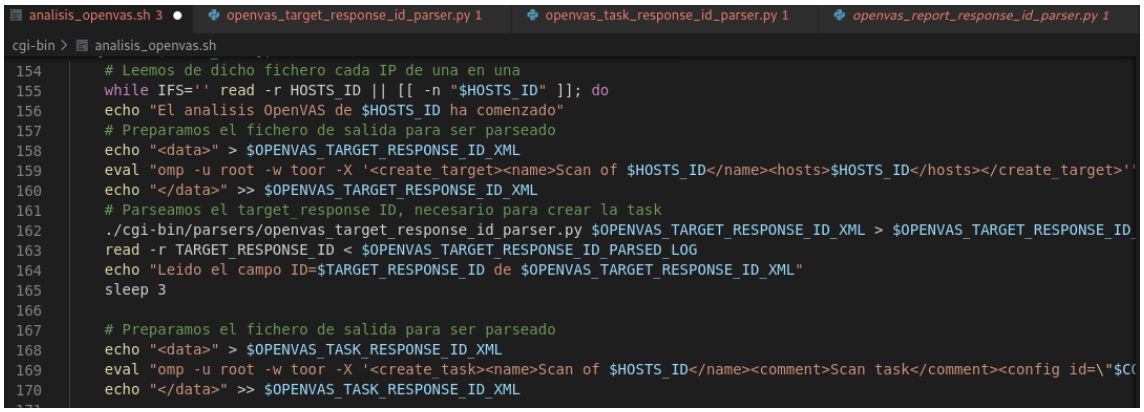


Figura 27: Fragmento de código – analisis_openvas.sh
Fuente**: Propia**

openvas_target_response_id_parser.py

Para cada etiqueta <create_target_response>, leemos el campo id

openvas_task_response_id_parser.py

Para cada etiqueta <create_task_response>, leemos el campo id

openvas_report_response_id_parser.py

Para cada etiqueta <star_task_response>, leemos el campo report_id

openvas_parser.py

Recorremos cada uno de los elementos del reporte de OpenVAS, para cada campo port, escribimos la información de los campos que hayamos encontrado: **host, severity, threat**, para cada campo result dentro de port, escribimos la información de los campos que hayamos encontrado: **host, port, nvt, oid, name, family, cve,** comprobamos si se tiene CVE asociada a la vulnerabilidad ejecutando el script vfeedcli.py, propio de la herramienta vfeed.

Definimos el archivo de destino en el cual se van a escribir los datos y parseamos toda la información al archivo de salida: ../reports/openvas_parsed_report.log.

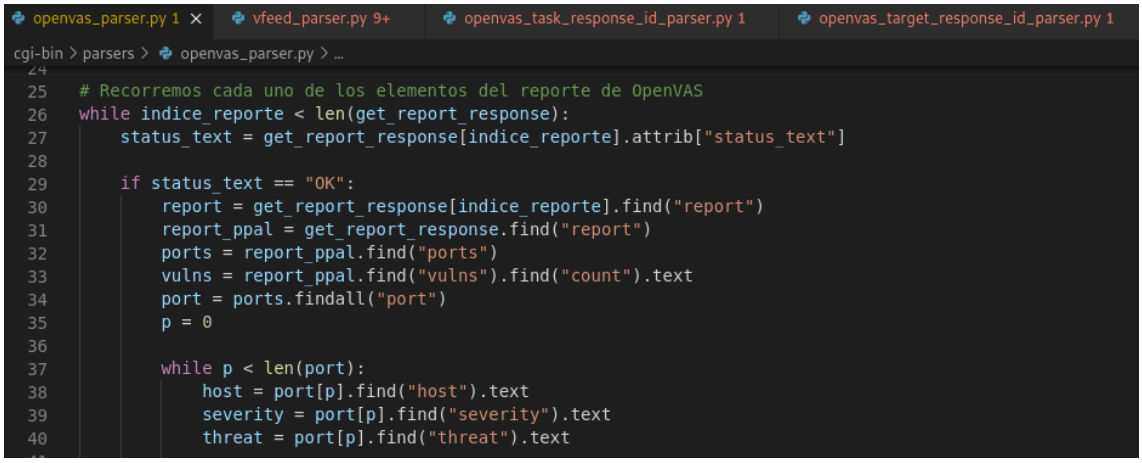


Figura 28: Fragmento de código – openvas_parser.py
Fuente**: Propia**

vfeed_parser.pl

Comprobamos si se tiene CVE asociada a la vulnerabilidad, y si aparece mas de una vez en la lista de CVE, comprobamos si existe el CVSS y el CWE en la base de datos de vFeed, se define el fichero de destino en el cual se van a escribir los datos: ../reports/ vfeed_parsed_report.log.

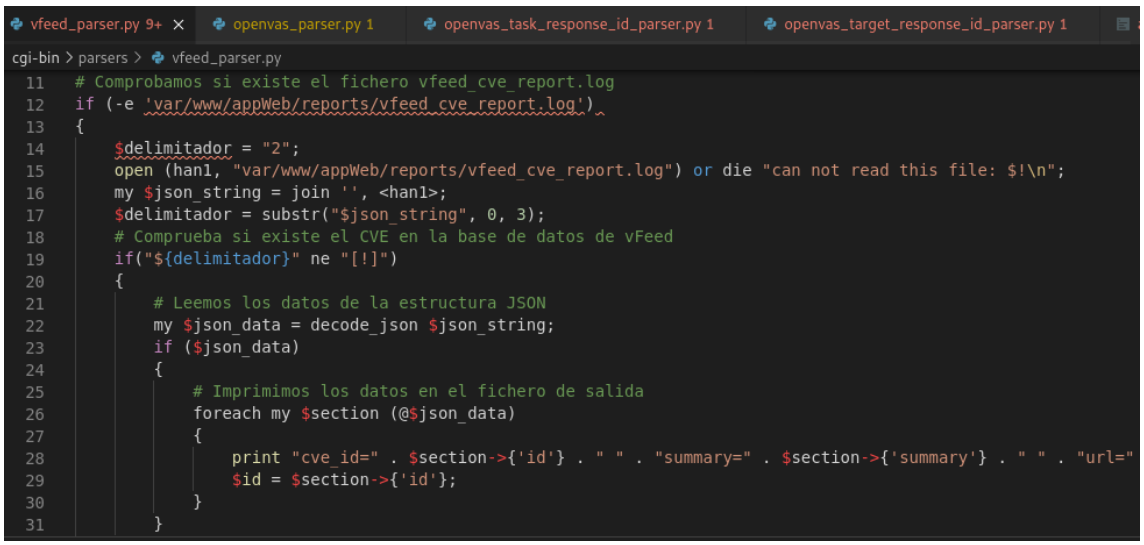


Figura 29: Fragmento de código – vfeed_parser.pl
Fuente**: Propia**

3.3.6 Página reporte.php

En una primera instancia presenta al usuario la opción **Logs**, que permite seleccionar el detalle del resultado de las diferentes herramientas a través de archivos logs que contiene los principales campos que se parsearon previamente, también presenta la opción para acceder al **Reporte** en el dashboard de kibana:

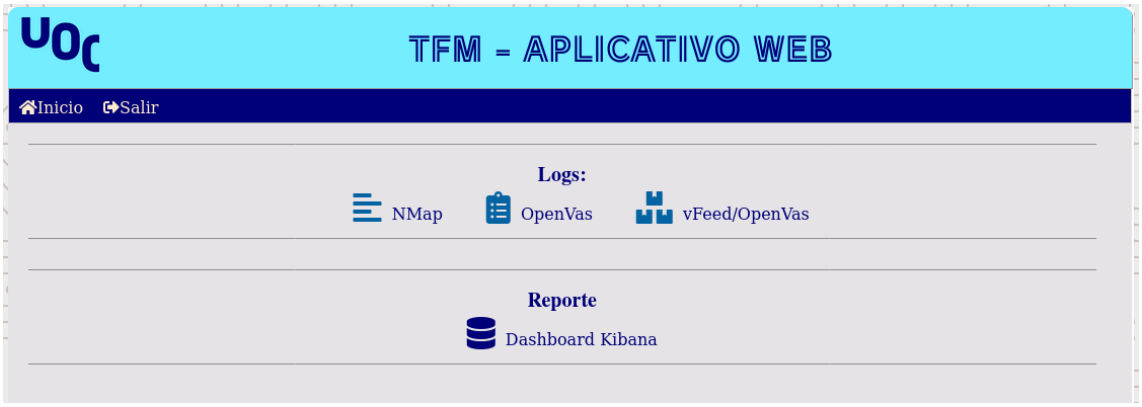


Figura 30: Página reporte.php
Fuente**: Propia**

reporte_logs.php

Logs NMap: presenta el resultado del análisis realizado con nmap, los principales campos que se listan son: host_id, protocol, port, state_port, reason_port, service_name, si fue posible detectar el sistema operativo se presentan los campos: osname, ostype, osvendor, osfamily.

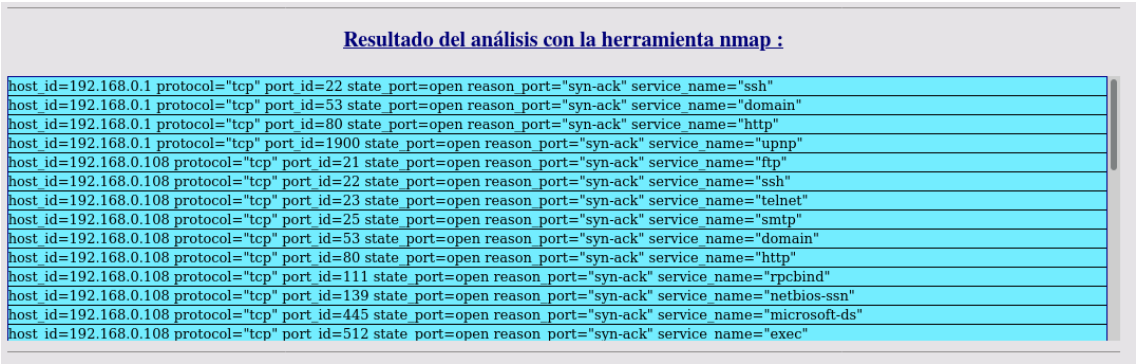


Figura 31: Página reporte_logs.php - NMap
Fuente**: Propia**

Logs OpenVas: presenta el resultado del escaneo realizado con openvas, los principales campos que se listan son: host_id, port, nvt_oid, threat, severity.

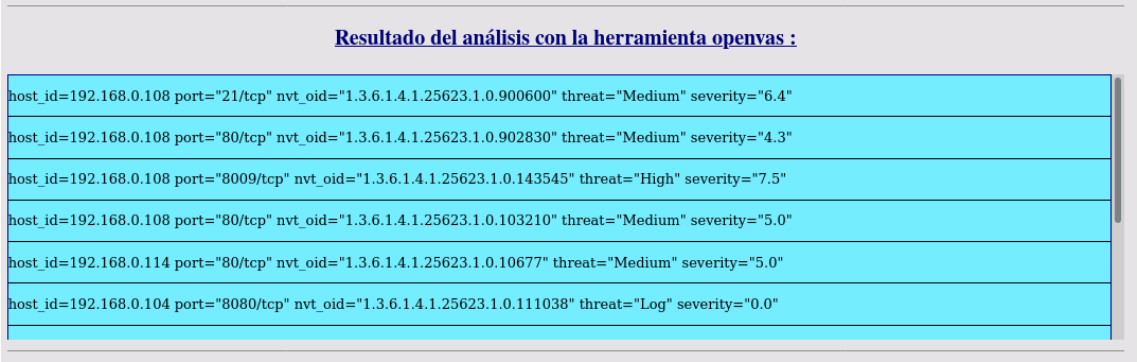


Figura 32: Página reporte_logs.php - OpenVas
Fuente**: Propia**

Logs vFeed: presenta el resultado de los cve asociados a la vulnerabilidad, los principales campos que muestran son: host_id, port, ref_type, ref_id:

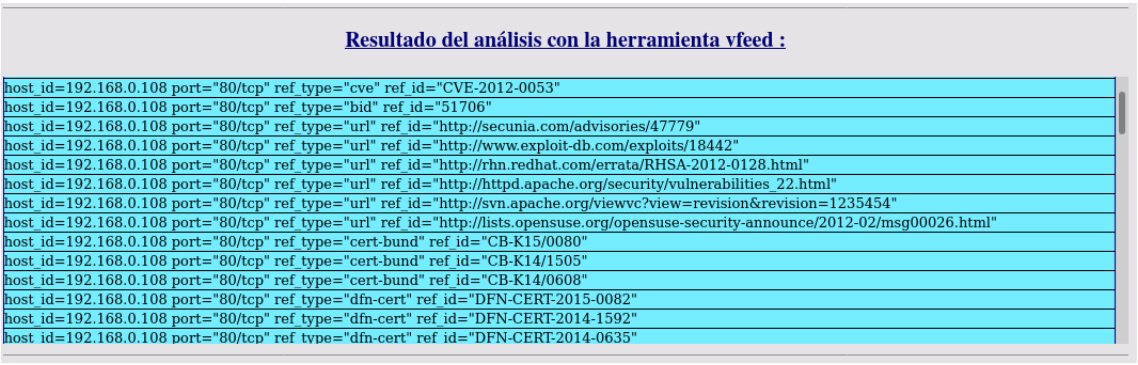


Figura 33: Página reporte_logs.php - vFeed
Fuente**: Propia**

3.3.7 Dashboard Kibana

Presenta el dashboard con el resultado de las vulnerabilidades detectadas en la interfaz web de kibana, se ejecuta en la ruta: localhost:5601.

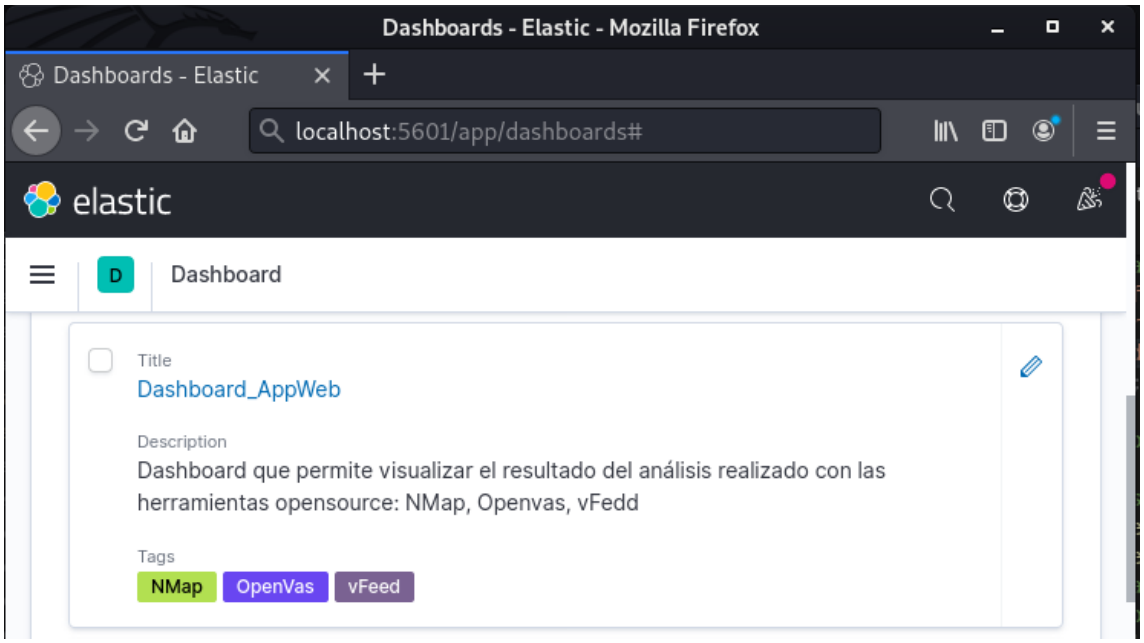


Figura 34: Dashboard Kibana
Fuente**: Propia**

pipeline.conf

Sigue un proceso ETL(Extract, Transform and Load), la ingesta de datos se realiza a través de logstash ejecutando el archivo pipeline.conf, el origen de datos son los diferentes logs que se obtuvieron como resultado de los parser implementados en los scripts, mediante consulta grok se realiza el match de los principales campos que permitan mostrar la información relevante de las vulnerabilidades detectadas, cargándolos en el motor de búsqueda de elasticsearch en el índice: **report_openvas_index**, finalmente se presenta los

resultados en el dashboard de kibana permitiendo al usuario visualizar los datos en diferentes tipos de gráficos que pueden ser consultados en tiempo real.

Figura 35: Fragmento de código – pipeline.conf
Fuente**: Propia**

3.4 Escenario de pruebas

Para el escenario de pruebas se configura un entorno virtualizado en la red lan 192.168.0.0/24 con los equipos que se describen a continuación[18]:

Equipo	Sistema Operativo	Dirección IP
1	Kali Linux	192.168.0.114
2	Windows 10 MSEdge	192.168.0.104
3	Debian 10	192.168.0.107
4	Metasploitable	192.168.0.108

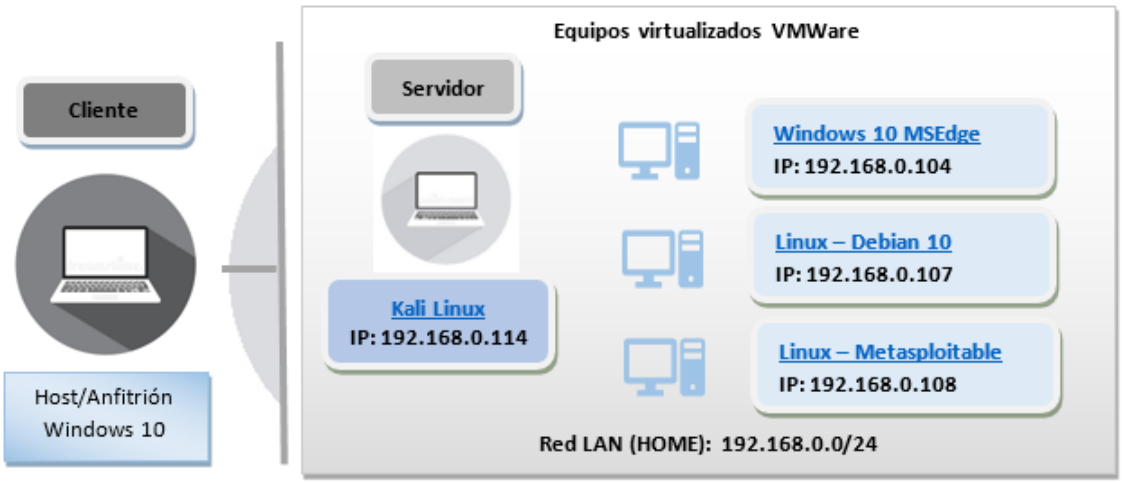


Figura 36: Escenario de pruebas
Fuente**: Propia**

[18] El entorno de pruebas se basa en la práctica realizada en la asignatura de Seguridad y pentesting de sistemas.

3.5 Resultados

Una vez realizado el escaneo de vulnerabilidades se presentan los siguientes resultados:

Datos proporcionados por las herramientas:

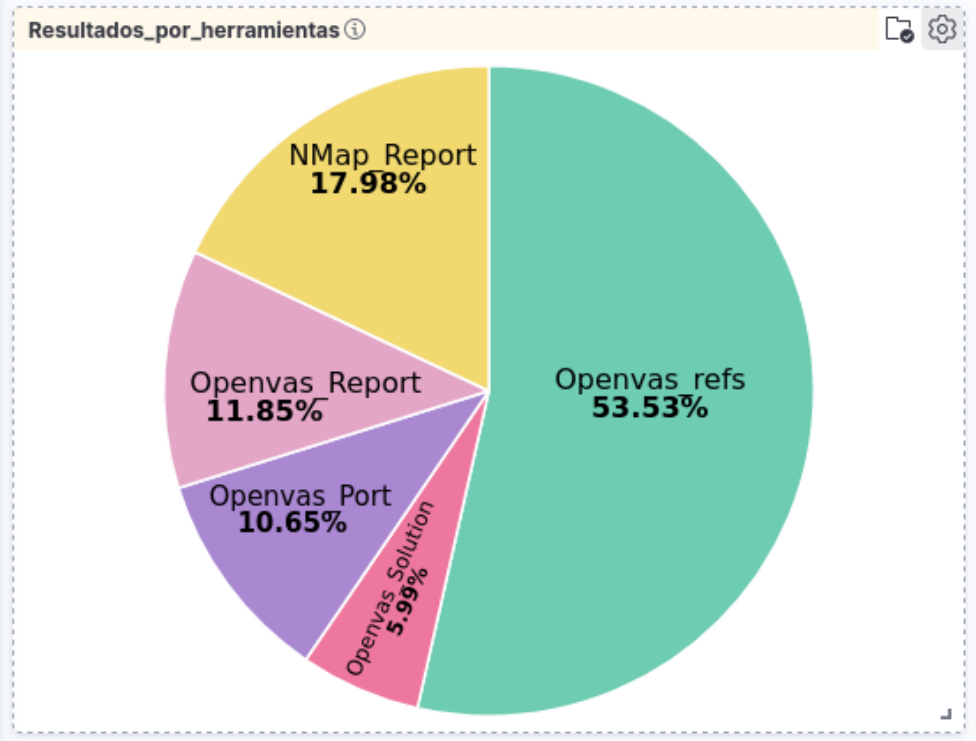


Figura 37: Información aportada por herramienta
Fuente**: Propia**

De acuerdo a los diferente match que se realizan con las consultas grok se observa que el reporte de nmap proporciona el 17.98% de información, mientras que los match de Openvas aporta con el 53.53% de información de referencias asociadas a vulnerabilidades detectadas, este valor incluye los CVE asociados con vfeed, el 11.85% indica los NVT ejecutados para la generación de reportes y que representan amenazas, de los puertos de los host activos el 10.65% presenta alguna vulnerabilidad, se observa también posibles soluciones en un 5.99% de las vulnerabilidades detectadas.

Sistemas operativos por host

Se reconoce los cuatro host activos con los que se realizó la prueba, además del sistema operativo:

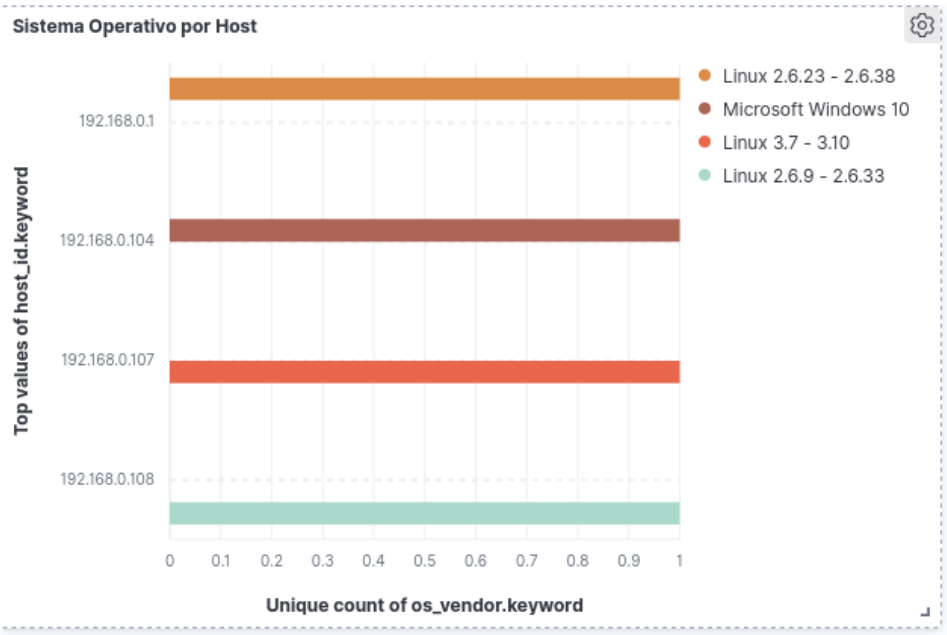


Figura 38: Sistemas operativos por host
Fuente**: Propia**

Amenazas detectadas por host:

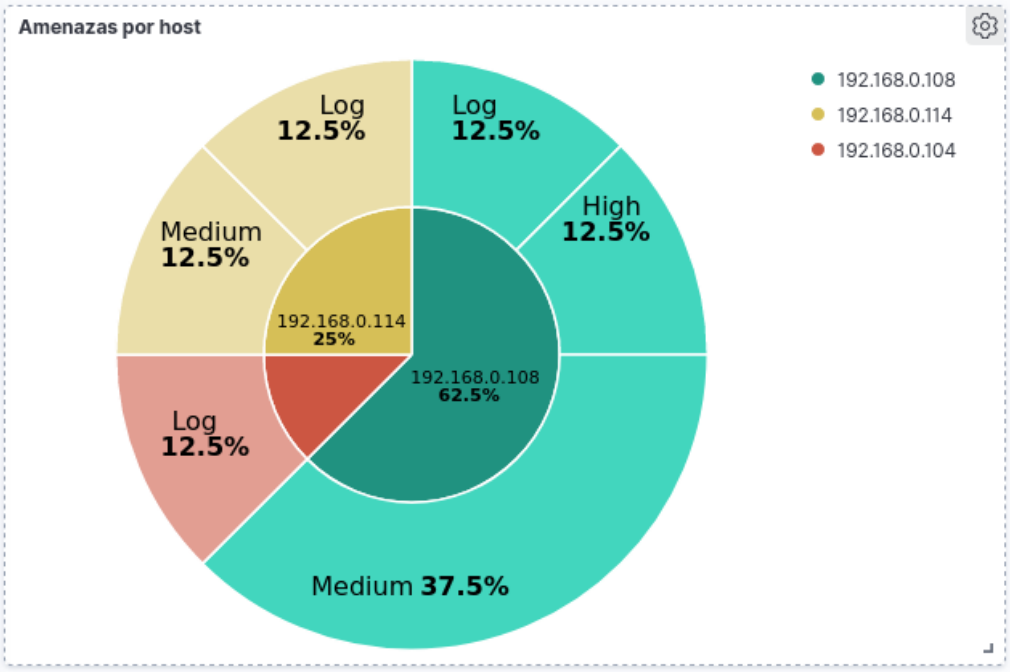


Figura 39: Amenazas detectadas por host
Fuente**: Propia**

Se observa que el host con la ip: 192.168.0.108, es el equipo con mayor número de vulnerabilidades con el 62.5%, de los cuales el 37.5% tiene una severidad media, el 12,.5 % tiene una severidad alta, el host 192.168.0.114, tiene un 25% de los cuales 12.5% son de severidad media, y el host: 192.168.0.104 solo presenta el 12.5% con unas severidad leve catalogada como log.

CVE asociados por host:

Se observa que el equipo host con la ip 192.168.0.108, presenta el mayor numero de referencias a CVE detectados y que se asocian con vfeed:

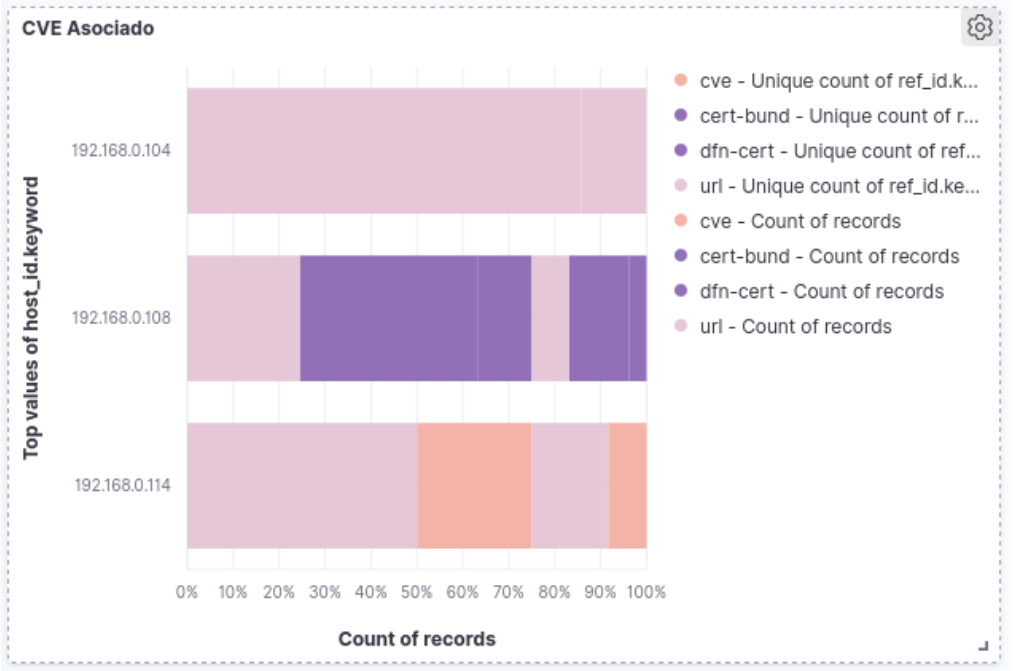


Figura 40: CVE asociada por host
Fuente**: Propia**

NVT por host:

Se observa que en el equipo host con la ip 192.168.0.108, se detectó el mayor número NVT ejecutado para la detección de vulnerabilidades:

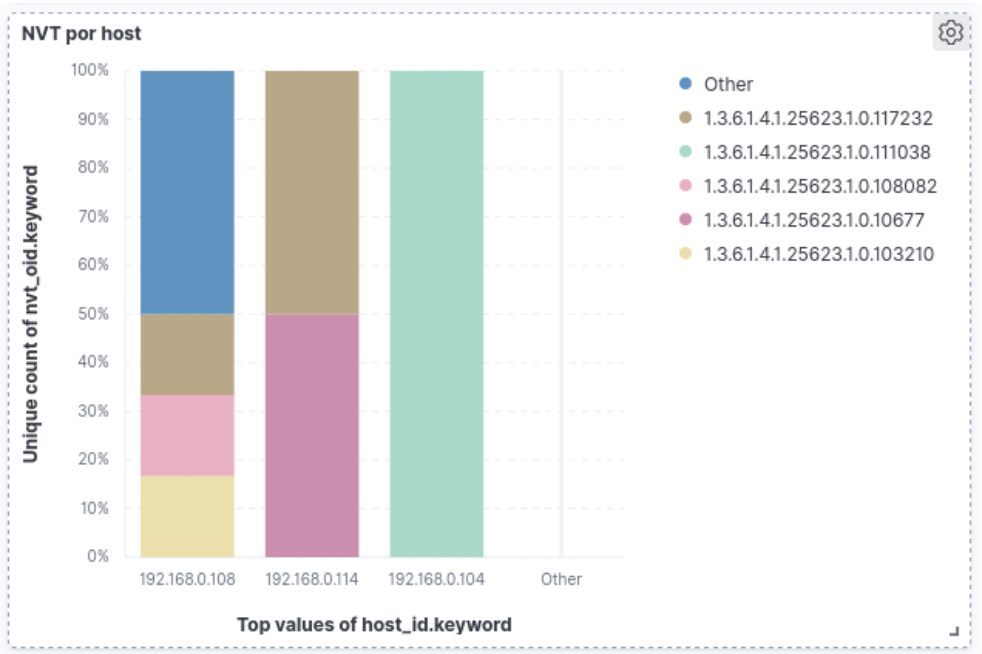


Figura 41: NVT por host
Fuente**: Propia**

Vulnerabilidades por puerto y host

El equipo host con la ip 192.168.0.108, es el que presenta el mayor número de vulnerabilidades en los puertos:

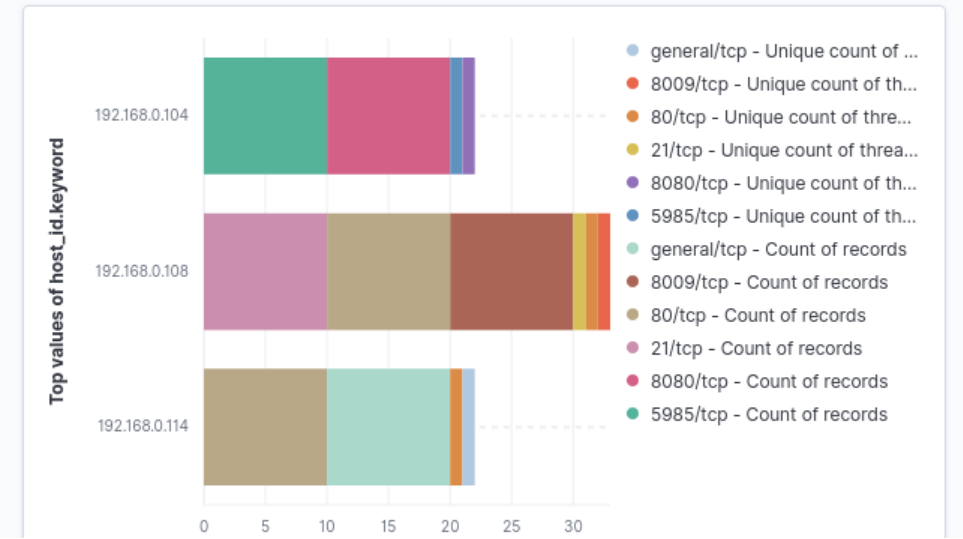


Figura 42: Vulnerabilidades por puerto y host
Fuente**: Propia**

4. Conclusiones

Como primer resultado se ha cumplido con el objetivo planteado en el TFM, se ha desarrollado una aplicación web intuitiva de fácil uso que permite realizar un escaneo de vulnerabilidades integrando las herramientas open source: nmap, openvas y vfeed, que conjuntamente con el stack ELK(Elasticsearch, logstash y kibana) presenta los resultados de las vulnerabilidades detectadas en un dashboard interactivo.

Tras completar las pruebas en el escenario que se planteó se comprobó que los resultados son coherentes y satisfactorios según lo esperado ya que al virtualizar un equipo metasploitable, este fue el que mayor vulnerabilidades presento, seguido del equipo Windows y el equipo Kali en el que se instaló el servidor apache, también notamos que no se muestra resultados del equipo Debian y es correcto ya que la prueba se integró con la práctica del módulo de Sistemas y pentesting de servidores, en el que se realizó un hardening del equipo Debian, notando que nmap refleja al equipo como un host inactivo, por lo que es omitido por el escaneo openvas.

He intentado que la interfaz sea lo más amigable posible y sobre todo que la ejecución de los diferentes scripts que permiten realizar el análisis y escaneo de la red sean transparentes para un usuario no experto, pero me llevo a un problema con el servidor web apache que se implementa en Kali ya que se debió escalar a privilegios de usuario root para poder ejecutar los Shell code y que estos no se bloqueen, este hecho fue detectado al realizar el análisis con el propio aplicativo web como una vulnerabilidad que puede comprometer la seguridad del aplicativo, además al estar en un ambiente virtualizado, la implementación del stack ELK repercutió en el rendimiento del host físico teniendo la necesidad de incrementar la memoria ram a 12Gb. Otro inconveniente que surgió fue con la herramienta vFeed, ya que a la fecha es de pago por lo que se debe suscribir a la misma para poder tener acceso a la base de datos actualizada.

A líneas futuras hay muchos aspectos de mejora para el aplicativo sobre todo en la interacción con el usuario, en la versión actual solo se ejecutan los comandos básicos con las opciones por defecto al realizar un escaneo de este tipo, pero las herramientas nmap como omp de openvas proporcionan muchos más parámetros que permiten realizar un escaneo a profundidad que permitiría conseguir información mas detallada, tomando en cuenta también que el tiempo de ejecución será mayor. Otro aspecto interesante de mejora es la implementación de librerías de reportes como OpenVAS Reporting, que permite presentar los resultados en un formato de archivo Excel

Finalmente cabe señalar que la experiencia al desarrollar el presente TFM ha sido muy positiva, ha sido necesario implementar diferentes conceptos de lo aprendido a lo largo de estos semestres, siendo gratificante reflejarlos en el desarrollo del aplicativo.

5. Glosario

amenaza *f* Violación de la seguridad en potencia, que existe en función de unas circunstancias, capacidad, acción o evento que pueda llegar a causar una infracción de la seguridad y/o causar algún daño en el sistema.

ataque *m* Agresión a la seguridad de un sistema fruto de un acto intencionado y deliberado que viola la política de seguridad de un sistema.

CERT (computer emergency response team) *m* Equipo de respuestas a emergencias informáticas. Una de sus principales tareas consiste en la gestión de vulnerabilidades.

CSIRT (computer security incident response team) *f* Equipo de respuesta a incidentes de seguridad informática. Una de sus principales tareas consiste en la gestión de vulnerabilidades.

CSV *f* Vienen del inglés "Comma Separated Values" y significan valores separados por comas.

CVE (common vulnerabilities and exposures) *f* Estándar publico para la identificación de vulnerabilidades. Asocia un identificador único a cada vulnerabilidad diferente.

CVSS (common vulnerability scoring system) *f* Marco común para la evaluación de la criticidad de vulnerabilidades.

ETL, *m* Acrónimo del concepto Extract Transform, Load.

exploit *m* Programa o script que permite explotar una o varias vulnerabilidades, es decir, programa que permite realizar un ataque aprovechando la vulnerabilidad.

GNU *GPL* (General Public License) *f* Licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto.

IDS (Intrusion Detection System) *m* Un Sistema de Detección de Intrusos, es un componente dentro del modelo de seguridad informática de una organización. Consiste en detectar actividades inapropiadas, incorrectas o anómala, desde el exterior o interior de un dispositivo o una infraestructura de red.

JSON (Javascript Object Notation) *m* Notación de Objetos de JavaScript, es un formato ligero de intercambio de datos.

NVTs (Network Vulnerability Tests) *m* Componente de OpenVas que ejecuta pruebas de vulnerabilidad de red.

OAP *m* Protocolo de administración de OpenVAS.

OMP *m* Protocolo de gestión OpenVAS.

OSSIM (Open Source Security Information Management) *m* Es una colección de herramientas bajo la licencia GPL, diseñadas para ayudar a los administradores de red en la seguridad de las computadoras, detección de intrusos y prevención.

OTP *m* Protocolo de transferencia OpenVAS.

PHP (Hypertext Pre-Processor) *m*, Es un pre-procesador de hipertexto.

política de seguridad *f* Conjunto de reglas y prácticas que definen y regulan los servicios de seguridad de una organización o sistema con el propósito de proteger sus recursos críticos y sensibles. En otras palabras, es la declaración de lo que está permitido y lo que no está permitido hacer.

riesgo *m* Expectativa de pérdida expresada como la probabilidad de que una amenaza particular explote una vulnerabilidad concreta con resultados especialmente perjudiciales.

rootkit *m* Programa que permite el acceso privilegiado a un ordenador y consigue ocultar su presencia al administrador. Suele hacer uso de varias vulnerabilidades para instalarse y conseguir su propósito.

vulnerabilidad de día-cero (zero-day vulnerability) *f* Vulnerabilidad de cuya existencia, en el momento de ser explotada, no se tiene conocimiento previo.

vulnerabilidad de seguridad *f* Fallo o debilidad en el diseño, la implementación, la operación o la gestión de un sistema, que puede ser explotado con el fin de violar la política de seguridad del sistema.

XML *f* Siglas en inglés de eXtensible Markup Language (en español, lenguaje de marcas extensible)

6. Bibliografía

Arribas, G. N. (2020). *Introducción a las vulnerabilidades - PID_00255333.* Barcelona: Fundació Universitat Oberta de Catalunya (FUOC).

Daniel Cruz Allende, A. T. (2020). *Análisis de riesgos - PID_00275346.* Barcelona: Fundació Universitat Oberta de Catalunya (FUOC).

Jordi Herrera Joancomartí, G. N. (2020). *Vulnerabilidades en redes - PID_00255332.* Barcelona: Fundació Universitat Oberta de Catalunya (FUOC).

Silvia Garre Gui, A. J. (2020). *Introducción a la seguridad de la información - PID_00275350.* Barcelona: Fundació Universitat Oberta de Catalunya (FUOC).

Parada, D., Flórez, A., Gómez, U. (2018). *Análisis de los Componentes de la Seguridad desde una Perspectiva Sistémica de la Dinámica de Sistemas*. Revista Información Tecnológica, 29(1), 27-38.

Jeong, C., Lee, S., Lim, J. (2019). *Information security breaches and IT security investments: Impacts on competitors. Information & Management*, 56(5), 681-695.

Ghanem, M. A. (2015). BackTrack System: Security against Hacking. *International Journal of Scientific and Research Publications*, *5*(2), 4.

Solarte, F. N. S., Rosero, E. R. E., & del Carmen Benavides, M. (2015). *Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001*. *Revista Tecnológica-ESPOL*, *28*(5).

Mell, P.; Scarfone, K.; Romanosky, S. (2007). A *Complete Guide to the Common Vulnerability Scoring System,* Version 2.0 [artículo en línea] http://www.first.org/cvss/cvss-guide.html

La clasificación de vulnerabilidades de seguridad informática, Edgar Vega Briceño M.Sc, Consultado el 18 de marzo de 2021, https://www.linkedin.com/pulse/la-clasificaci%C3%B3n-de-vulnerabilidades-seguridad-edgar-a-vega-brice%C3%B1o

Guía de seguridad de las TIC (CCN-STIC-954), Centro Criptológico Nacional, consultado el 20 de marzo de 2021, https://ns2.elhacker.net/timofonica/manuales/Guia_Avanzada_Nmap.pdf

Vulnerabilidades informáticas, Graciela Marker, Consultado el 20 de marzo de 2021, https://www.tecnologia-informatica.com/vulnerabilidades-informaticas/

Introducción a la seguridad informática y el análisis de vulnerabilidades, Martha Irene Romero Castro, Consultado el 20 de marzo de 2021, https://www.3ciencias.com/wp-content/uploads/2018/10/Seguridad-inform%C3%A1tica.pdf

Guía de referencia de Nmap, Gerick Toro, Consultado el 21 de marzo de 2021, https://we.riseup.net/assets/77169/Manual-de-uso-de-Nmap.pdf

Webiner OpenVas, Alonso Caballero, Consultado el 21 de marzo de 2021, http://www.reydes.com/archivos/slides/webinars/AC_WG_OpenVAS_v2.pdf

*¿Qué es y para qué sirve Elastic Stack?,*Jeyson Andrés Guzmán, Consultado el 24 de marzo de 2021, https://es.linkedin.com/pulse/qu%C3%A9-es-y-para-sirve-elastic-stack-elasticsearch-logstash-guzm%C3%A1n

¿Escaneando la red con nmap en Kali Linux, Consultado el 10 de abril de 2021, https://byte-mind.net/escaneando-la-red-con-nmap/

Dynamically create fields and automatically add the values of the filled fields, Consultado el 10 de abril de 2021, https://stackoverflow.com/questions/51322475/how-to-dynamically-create-fields-and-automatically-add-the-values-of-the-filled

Como crear, escribir, leer y eliminar archivos en PHP, Consultado el 11 de abril de 2021, https://code.tutsplus.com/es/tutorials/create-write-read-and-delete-files-in-php--cms-34950

Technical Documentation for Greenbone Technologies, Consultado el 16 de abril de 2021, https://docs.greenbone.net/API/GMP/gmp-8.0.html#command_create_task

Installation and Configuration of ElasticSearch, Consultado el 01 de mayo de 2021,https://gitlab.com/LabIT/elasticsearch/-/blob/master/elk_manual_configuration/elastic_manual_install.md

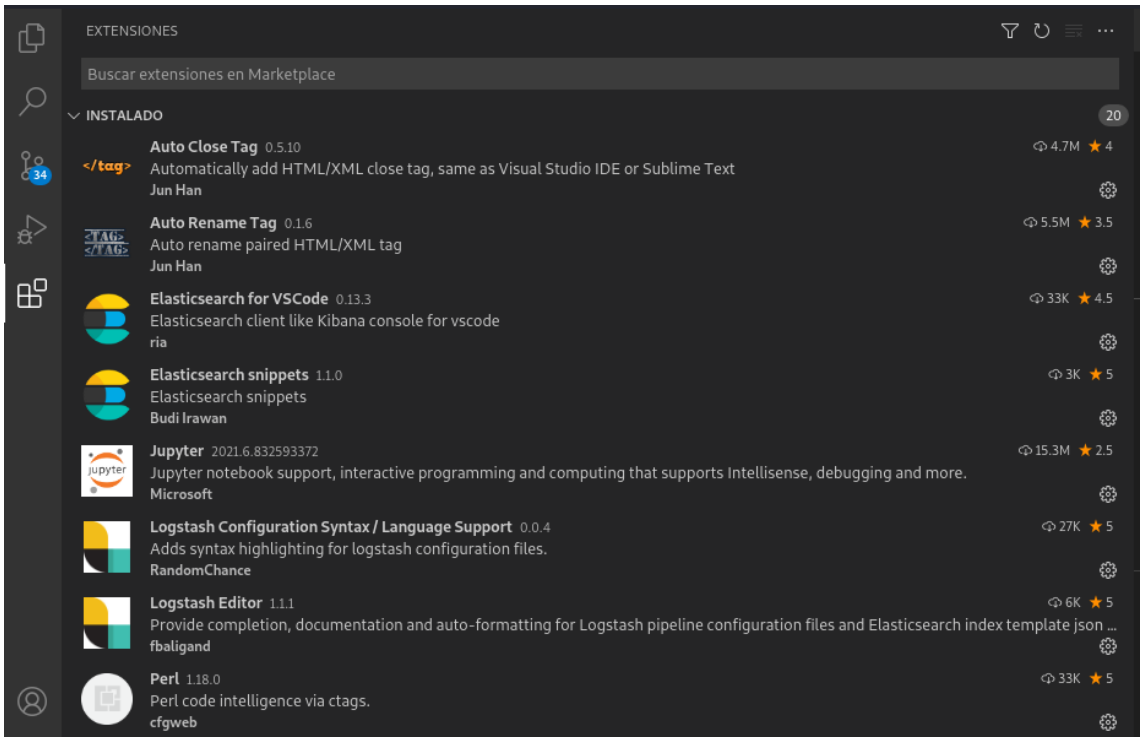
OpenVAS Management Protocol (OMP), Consultado el 04 de mayo de 2021, https://docs.greenbone.net/API/OMP/omp.html#command_start_task

Tutorial Logstash Grok Patterns, Consultado el 08 de mayo de 2021, https://coralogix.com/blog/logstash-grok-tutorial-with-examples/

7. Anexos

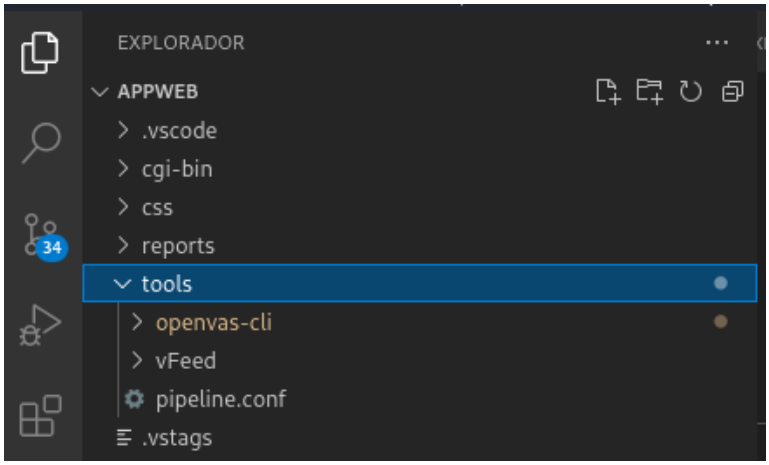
Como se indico previamente para el desarrollo de la aplicación se hizo uso del IDE Visual Studio Code, pero también se instaló las siguientes extensiones:

- Auto Close Tag
- Auto Rename Tag
- Prettier Formatter for Visual Studio Code
- Elasticsearch for VSCode
- Elasticsearch Snippets for Visual Studio Code
- Logstash Configuration Syntax
- VSCode Logstash Editor
- PHP Debug
- Perl for Visual Studio Code
- Pylance
- Python extension for Visual Studio Code
- XML Tools for Visual Studio Code
- YAML Language Support



tools

Se integro las herramientas openvas-cli y vFeed al aplicativo en la carpeta tools:



openvas-cli

Contiene el binario **omp** que es un cliente de línea de comandos que utiliza el protocolo OMP para conectarse a OpenVAS Manager y facilita el acceso a la funcionalidad completa proporcionada por OpenVAS Manager, desde la línea de comandos permitiendo una rápida integración en un entorno de script. Se lo puede descargar desde el siguiente enlace:

https://manpages.debian.org/testing/openvas-cli/omp.8.en.html

Para la instalación necesitamos cumplir con los siguientes requisitos:

* cmake >= 2.8
* glib-2.0
* gnutls (>= 2.8)
* openvas-libraries (>= 8.0.4)
* pkg-config

En mi caso se integro con el aplicativo, por lo que la librería se incluye en el mismo directorio y se define la variable de entorno que apunta a dicha librería:

$ export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:
/var/www/appWeb/tools/openvas-cli/openvas_lib

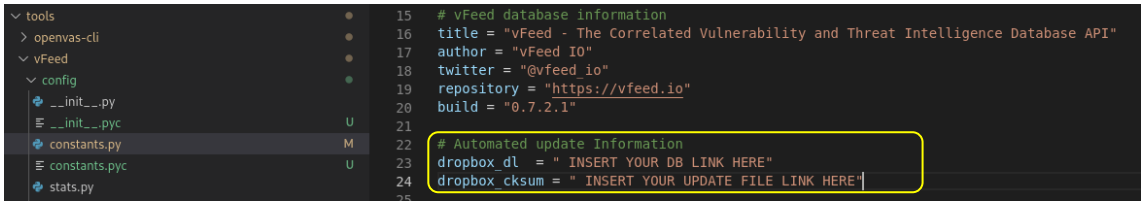
Ingresamos al directorio: /var/www/appWeb/tools/openvas-cli, y generamos el instalador del binario con cmake:

mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=/var/www/appWeb/tools/openvas-cli

vFeed

Actualmente vfeed ya no es una herramienta gratuita y requiere de una suscripción para poder hacer uso de la base de datos actualizada, en este caso debemos suscribirnos a través del siguiente link: https://vfeed.io/

Una vez suscritos se remite los enlaces DB Link que corresponden al registro de vfeed, estos demos ingresarlos en el archivo: constants.py, ubicado en el directorio: /var/www/appWeb/tools/vFeed/config/constants.py



Como parte de los resultados del escaneo de las pruebas realizadas se incluye los reportes xml con el correspondiente parser:

nmap_report.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nmaprun>
<?xml-stylesheet href="file:///usr/bin/../share/nmap/nmap.xsl" type="text/xsl"?>
<!-- Nmap 7.91 scan initiated Mon May 24 22:26:45 2021 as: nmap -T5 -O -iL /var/www/appWeb/cgi-bin/logs/nmap_hosts_up_parsed.log -oX /var/www/appWeb/reports/nmap_report.xml -&#45;privileged -&#45;osscan-guess -&#45;reason -->
<nmaprun scanner="nmap" args="nmap -T5 -O -iL /var/www/appWeb/cgi-bin/logs/nmap_hosts_up_parsed.log -oX /var/www/appWeb/reports/nmap_report.xml -&#45;privileged -&#45;osscan-guess -&#45;reason" start="1621913205" startstr="Mon May 24 22:26:45 2021" version="7.91" xmloutputversion="1.05">
<scaninfo type="syn" protocol="tcp" numservices="1000" services="1,3-4,6-7,9,13,17,19-26,30,32-33,37,42-43,49,53,70,79-85,88-90,99-100,106,109-111,113,119,125,135,139,143-144,146,161,163,179,199,211-212,222,254-256,259,264,280,301,306,311,340,366,389,406-407,416-417,425,427,443-445,458,464-465,481,497,500,512-515,524,541,543-545,548,554-555,563,587,593,616-617,625,631,636,646,648,666-668,683,687,691,700,705,711,714,720,722,726,749,765,777,783,787,800-801,808,843,873,880,888,898,900-903,911-912,981,987,990,992-993,995,999-1002,1007,1009-1011,1021-1100,1102,1104-1108,1110-1114,1117,1119,1121-1124,1126,1130-1132,1137-1138,1141,1145,1147-1149,1151-1152,1154,1163-1166,1169,1174-1175,1183,1185-1187,1192,1198-1199,1201,1213,1216-1218,1233-1234,1236,1244,1247-1248,1259,1271-1272,1277,1287,1296,1300-1301,1309-1311,1322,1328,1334,1352,1417,1433-1434,1443,1455,1461,1494,1500-1501,1503,1521,1524,1533,1556,1580,1583,1594,1600,1641,1658,1666,1687-1688,1700,1717-1721,1723,1755,1761,1782-1783,1801,1805,1812,1839-1840,1862-1864,1875,1900,1914,1935,1947,1971-1972,1974,1984,1998-2010,2013,2020-2022,2030,2033-2035,2038,2040-2043,2045-2049,2065,2068,2099-2100,2103,2105-2107,2111,2119,2121,2126,2135,2144,2160-2161,2170,2179,2190-2191,2196,2200,2222,2251,2260,2288,2301,2323,2366,2381-2383,2393-2394,2399,2401,2492,2500,2522,2525,2557,2601-2602,2604-2605,2607-2608,2638,2701-2702,2710,2717-2718,2725,2800,2809,2811,2869,2875,2909-2910,2920,2967-2968,2998,3000-3001,3003,3005-3007,3011,3013,3017,3030-3031,3052,3071,3077,3128,3168,3211,3221,3260-3261,3268-3269,3283,3300-3301,3306,3322-3325,3333,3351,3367,3369-3372,3389-3390,3404,3476,3493,3517,3527,3546,3551,3580,3659,3689-3690,3703,3737,3766,3784,3800-3801,3809,3814,3826-3828,3851,3869,3871,3878,3880,3889,3905,3914,3918,3920,3945,3971,3986,3995,3998,4000-4006,4045,4111,4125-4126,4129,4224,4242,4279,4321,4343,4443-4446,4449,4550,4567,4662,4848,4899-4900,4998,5000-5004,5009,5030,5033,5050-5051,5054,5060-5061,5080,5087,5100-
```

```
5102,5120,5190,5200,5214,5221-5222,5225-5226,5269,5280,5298,5357,5405,5414,5431-
5432,5440,5500,5510,5544,5550,5555,5560,5566,5631,5633,5666,5678-5679,5718,5730,5800-5802,5810-
5811,5815,5822,5825,5850,5859,5862,5877,5900-5904,5906-5907,5910-
5911,5915,5922,5925,5950,5952,5959-5963,5987-5989,5998-6007,6009,6025,6059,6100-
6101,6106,6112,6123,6129,6156,6346,6389,6502,6510,6543,6547,6565-6567,6580,6646,6666-
6669,6689,6692,6699,6779,6788-6789,6792,6839,6881,6901,6969,7000-
7002,7004,7007,7019,7025,7070,7100,7103,7106,7200-
7201,7402,7435,7443,7496,7512,7625,7627,7676,7741,7777-7778,7800,7911,7920-7921,7937-7938,7999-
8002,8007-8011,8021-8022,8031,8042,8045,8080-8090,8093,8099-8100,8180-8181,8192-
8194,8200,8222,8254,8290-8292,8300,8333,8383,8400,8402,8443,8500,8600,8649,8651-
8652,8654,8701,8800,8873,8888,8899,8994,9000-9003,9009-9011,9040,9050,9071,9080-9081,9090-
9091,9099-9103,9110-9111,9200,9207,9220,9290,9415,9418,9485,9500,9502-9503,9535,9575,9593-
9595,9618,9666,9876-9878,9898,9900,9917,9929,9943-9944,9968,9998-10004,10009-10010,10012,10024-
10025,10082,10180,10215,10243,10566,10616-10617,10621,10626,10628-10629,10778,11110-
11111,11967,12000,12174,12265,12345,13456,13722,13782-13783,14000,14238,14441-14442,15000,15002-
15004,15660,15742,16000-16001,16012,16016,16018,16080,16113,16992-
16993,17877,17988,18040,18101,18988,19101,19283,19315,19350,19780,19801,19842,20000,20005,20031,
20221-20222,20828,21571,22939,23502,24444,24800,25734-25735,26214,27000,27352-27353,27355-
27356,27715,28201,30000,30718,30951,31038,31337,32768-32785,33354,33899,34571-
34573,35500,38292,40193,40911,41511,42510,44176,44442-44443,44501,45100,48080,49152-
49161,49163,49165,49167,49175-49176,49400,49999-
50003,50006,50300,50389,50500,50636,50800,51103,51493,52673,52822,52848,52869,54045,54328,55055-
55056,55555,55600,56737-
56738,57294,57797,58080,60020,60443,61532,61900,62078,63331,64623,64680,65000,65129,65389"/>
<verbose level="0"/>
<debugging level="0"/>
<hosthint><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.0.108" addrtype="ipv4"/>
<address addr="00:0C:29:FA:DD:2A" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
</hosthint>
<hosthint><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.0.117" addrtype="ipv4"/>
<address addr="00:0C:29:92:ED:33" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
</hosthint>
<hosthint><status state="up" reason="arp-response" reason_ttl="0"/>
<address addr="192.168.0.1" addrtype="ipv4"/>
<address addr="0C:80:63:77:2E:6A" addrtype="mac" vendor="Tp-link Technologies"/>
<hostnames>
</hostnames>
</hosthint>
<host starttime="1621913205" endtime="1621913207"><status state="up" reason="arp-
response" reason_ttl="0"/>
<address addr="192.168.0.1" addrtype="ipv4"/>
<address addr="0C:80:63:77:2E:6A" addrtype="mac" vendor="Tp-link Technologies"/>
<hostnames>
```

```
</hostnames>
<ports><extraports state="closed" count="996">
<extrareasons reason="resets" count="996"/>
</extraports>
<port protocol="tcp" portid="22"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ssh" method="table" conf="3"/></port>
<port protocol="tcp" portid="53"><state state="open" reason="syn-
ack" reason_ttl="63"/><service name="domain" method="table" conf="3"/></port>
<port protocol="tcp" portid="80"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="http" method="table" conf="3"/></port>
<port protocol="tcp" portid="1900"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="upnp" method="table" conf="3"/></port>
</ports>
<os><portused state="open" proto="tcp" portid="22"/>
<portused state="closed" proto="tcp" portid="1"/>
<portused state="closed" proto="udp" portid="32063"/>
<osmatch name="Linux 2.6.23 - 2.6.38" accuracy="100" line="52083">
<osclass type="general purpose" vendor="Linux" osfamily="Linux" osgen="2.6.X" accuracy="100"><cp
e>cpe:/o:linux:linux_kernel:2.6</cpe></osclass>
</osmatch>
</os>
<uptime seconds="2599478" lastboot="Sat Apr 24 20:22:11 2021"/>
<distance value="1"/>
<tcpsequence index="200" difficulty="Good luck!" values="DC7EB8F1,DCED466B,DC35A161,DCDBD82E,DC4
E1938,DC7AD5E3"/>
<ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
<tcptssequence class="other" values="26E3E057,26E3E071,26E3E08A,26E3E0A4,26E3E0BD,26E3E0D6"/>
<times srtt="455" rttvar="47" to="50000"/>
</host>
<host starttime="1621913205" endtime="1621913207"><status state="up" reason="arp-
response" reason_ttl="0"/>
<address addr="192.168.0.108" addrtype="ipv4"/>
<address addr="00:0C:29:FA:DD:2A" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="977">
<extrareasons reason="resets" count="977"/>
</extraports>
<port protocol="tcp" portid="21"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ftp" method="table" conf="3"/></port>
<port protocol="tcp" portid="22"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ssh" method="table" conf="3"/></port>
<port protocol="tcp" portid="23"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="telnet" method="table" conf="3"/></port>
<port protocol="tcp" portid="25"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="smtp" method="table" conf="3"/></port>
<port protocol="tcp" portid="53"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="domain" method="table" conf="3"/></port>
```

```
<port protocol="tcp" portid="80"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="http" method="table" conf="3"/></port>
<port protocol="tcp" portid="111"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="rpcbind" method="table" conf="3"/></port>
<port protocol="tcp" portid="139"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="netbios-ssn" method="table" conf="3"/></port>
<port protocol="tcp" portid="445"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="microsoft-ds" method="table" conf="3"/></port>
<port protocol="tcp" portid="512"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="exec" method="table" conf="3"/></port>
<port protocol="tcp" portid="513"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="login" method="table" conf="3"/></port>
<port protocol="tcp" portid="514"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="shell" method="table" conf="3"/></port>
<port protocol="tcp" portid="1099"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="rmiregistry" method="table" conf="3"/></port>
<port protocol="tcp" portid="1524"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ingreslock" method="table" conf="3"/></port>
<port protocol="tcp" portid="2049"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="nfs" method="table" conf="3"/></port>
<port protocol="tcp" portid="2121"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ccproxy-ftp" method="table" conf="3"/></port>
<port protocol="tcp" portid="3306"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="mysql" method="table" conf="3"/></port>
<port protocol="tcp" portid="5432"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="postgresql" method="table" conf="3"/></port>
<port protocol="tcp" portid="5900"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="vnc" method="table" conf="3"/></port>
<port protocol="tcp" portid="6000"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="X11" method="table" conf="3"/></port>
<port protocol="tcp" portid="6667"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="irc" method="table" conf="3"/></port>
<port protocol="tcp" portid="8009"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="ajp13" method="table" conf="3"/></port>
<port protocol="tcp" portid="8180"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="unknown" method="table" conf="3"/></port>
</ports>
<os><portused state="open" proto="tcp" portid="21"/>
<portused state="closed" proto="tcp" portid="1"/>
<portused state="closed" proto="udp" portid="42117"/>
<osmatch name="Linux 2.6.9 - 2.6.33" accuracy="100" line="59346">
<osclass type="general purpose" vendor="Linux" osfamily="Linux" osgen="2.6.X" accuracy="100"><cp
e>cpe:/o:linux:linux_kernel:2.6</cpe></osclass>
</osmatch>
</os>
<uptime seconds="4614" lastboot="Mon May 24 21:09:55 2021"/>
<distance value="1"/>
<tcpsequence index="200" difficulty="Good luck!" values="F2F1ECEF,F3638EA2,F2C547AC,F3659642,F34
713DD,F3B43552"/>
```

```
<ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
<tcptssequence class="100HZ" values="70941,7094B,70956,70960,7096A,70974"/>
<times srtt="352" rttvar="123" to="50000"/>
</host>
<host starttime="1621913205" endtime="1621913209"><status state="up" reason="arp-
response" reason_ttl="0"/>
<address addr="192.168.0.117" addrtype="ipv4"/>
<address addr="00:0C:29:92:ED:33" addrtype="mac" vendor="VMware"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="1000">
<extrareasons reason="resets" count="1000"/>
</extraports>
</ports>
<os><portused state="closed" proto="tcp" portid="1"/>
<portused state="closed" proto="udp" portid="43029"/>
</os>
<distance value="1"/>
<times srtt="327" rttvar="32" to="50000"/>
</host>
<host starttime="1621913209" endtime="1621913210"><status state="up" reason="localhost-
response" reason_ttl="0"/>
<address addr="192.168.0.114" addrtype="ipv4"/>
<hostnames>
</hostnames>
<ports><extraports state="closed" count="999">
<extrareasons reason="resets" count="999"/>
</extraports>
<port protocol="tcp" portid="80"><state state="open" reason="syn-
ack" reason_ttl="64"/><service name="http" method="table" conf="3"/></port>
</ports>
<os><portused state="open" proto="tcp" portid="80"/>
<portused state="closed" proto="tcp" portid="1"/>
<portused state="closed" proto="udp" portid="43515"/>
<osmatch name="Linux 2.6.32" accuracy="100" line="55736">
<osclass type="general purpose" vendor="Linux" osfamily="Linux" osgen="2.6.X" accuracy="100"><cp
e>cpe:/o:linux:linux_kernel:2.6.32</cpe></osclass>
</osmatch>
</os>
<uptime seconds="4239487" lastboot="Mon Apr  5 20:48:43 2021"/>
<distance value="0"/>
<tcpsequence index="263" difficulty="Good luck!" values="D6571357,5F44EFE8,55AF16CA,7A398875,D8E
DB031,C5E733D2"/>
<ipidsequence class="All zeros" values="0,0,0,0,0,0"/>
<tcptssequence class="1000HZ" values="FCB16C9D,FCB16D02,FCB16D66,FCB16DCA,FCB16E2F,FCB16E93"/>
<times srtt="47" rttvar="11" to="50000"/>
</host>
```

```
<runstats><finished time="1621913210" timestr="Mon May 24 22:26:50 2021" summary="Nmap done at M
on May 24 22:26:50 2021; 5 IP addresses (4 hosts up) scanned in 5.78 seconds" elapsed="5.78" exi
t="success"/><hosts up="4" down="1" total="5"/>
</runstats>
</nmaprun>
```

Nmap_parsed_report.log

```
host_id=192.168.0.1 protocol="tcp" port_id=22 state_port=open reason_port="syn-
ack" service_name="ssh"
host_id=192.168.0.1 protocol="tcp" port_id=53 state_port=open reason_port="syn-
ack" service_name="domain"
host_id=192.168.0.1 protocol="tcp" port_id=80 state_port=open reason_port="syn-
ack" service_name="http"
host_id=192.168.0.1 protocol="tcp" port_id=1900 state_port=open reason_port="syn-
ack" service_name="upnp"
host_id=192.168.0.108 protocol="tcp" port_id=21 state_port=open reason_port="syn-
ack" service_name="ftp"
host_id=192.168.0.108 protocol="tcp" port_id=22 state_port=open reason_port="syn-
ack" service_name="ssh"
host_id=192.168.0.108 protocol="tcp" port_id=23 state_port=open reason_port="syn-
ack" service_name="telnet"
host_id=192.168.0.108 protocol="tcp" port_id=25 state_port=open reason_port="syn-
ack" service_name="smtp"
host_id=192.168.0.108 protocol="tcp" port_id=53 state_port=open reason_port="syn-
ack" service_name="domain"
host_id=192.168.0.108 protocol="tcp" port_id=80 state_port=open reason_port="syn-
ack" service_name="http"
host_id=192.168.0.108 protocol="tcp" port_id=111 state_port=open reason_port="syn-
ack" service_name="rpcbind"
host_id=192.168.0.108 protocol="tcp" port_id=139 state_port=open reason_port="syn-
ack" service_name="netbios-ssn"
host_id=192.168.0.108 protocol="tcp" port_id=445 state_port=open reason_port="syn-
ack" service_name="microsoft-ds"
host_id=192.168.0.108 protocol="tcp" port_id=512 state_port=open reason_port="syn-
ack" service_name="exec"
host_id=192.168.0.108 protocol="tcp" port_id=513 state_port=open reason_port="syn-
ack" service_name="login"
host_id=192.168.0.108 protocol="tcp" port_id=514 state_port=open reason_port="syn-
ack" service_name="shell"
host_id=192.168.0.108 protocol="tcp" port_id=1099 state_port=open reason_port="syn-
ack" service_name="rmiregistry"
host_id=192.168.0.108 protocol="tcp" port_id=1524 state_port=open reason_port="syn-
ack" service_name="ingreslock"
host_id=192.168.0.108 protocol="tcp" port_id=2049 state_port=open reason_port="syn-
ack" service_name="nfs"
host_id=192.168.0.108 protocol="tcp" port_id=2121 state_port=open reason_port="syn-
ack" service_name="ccproxy-ftp"
```

```
host_id=192.168.0.108 protocol="tcp" port_id=3306 state_port=open reason_port="syn-
ack" service_name="mysql"
host_id=192.168.0.108 protocol="tcp" port_id=5432 state_port=open reason_port="syn-
ack" service_name="postgresql"
host_id=192.168.0.108 protocol="tcp" port_id=5900 state_port=open reason_port="syn-
ack" service_name="vnc"
host_id=192.168.0.108 protocol="tcp" port_id=6000 state_port=open reason_port="syn-
ack" service_name="X11"
host_id=192.168.0.108 protocol="tcp" port_id=6667 state_port=open reason_port="syn-
ack" service_name="irc"
host_id=192.168.0.108 protocol="tcp" port_id=8009 state_port=open reason_port="syn-
ack" service_name="ajp13"
host_id=192.168.0.108 protocol="tcp" port_id=8180 state_port=open reason_port="syn-
ack" service_name="unknown"
host_id=192.168.0.114 protocol="tcp" port_id=80 state_port=open reason_port="syn-
ack" service_name="http"
host_id=192.168.0.1 osname="Linux 2.6.23 -
 2.6.38" ostype="general purpose" osvendor="Linux" osfamily="Linux" osgen="2.6.X"
host_id=192.168.0.104 osname="Microsoft Windows 10" ostype="general purpose" osvendor
=" Microsoft Windows 10" osfamily="Windows" osgen="10 1709 - 1909"
host_id=192.168.0.107 osname="Linux 3.7 -
 3.10" ostype="general purpose" osvendor="Linux" osfamily="Linux" osgen="3.X"
host_id=192.168.0.108 osname="Linux 2.6.9 -
 2.6.33" ostype="general purpose" osvendor="Linux" osfamily="Linux" osgen="2.6.X"
host_id=192.168.0.114 osname="Linux 5.0 -
 5.2" ostype="general purpose" osvendor="Linux" osfamily="Linux" osgen="5.X"
```

openva_report.xml

```
<get_reports_response status="200" status_text="OK"><report id="b9e9cf8a-1633-4356-8843-
abfa8ad5dfa1" format_id="a994b278-1f62-11e1-96ac-
406186ea4fc5" extension="xml" content_type="text/xml"><owner><name>admin</name></owner><name>202
1-05-15T21:58:55Z</name><comment></comment><creation_time>2021-05-
15T21:58:55Z</creation_time><modification_time>2021-05-
15T22:35:45Z</modification_time><writable>0</writable><in_use>0</in_use><task id="11f35a4c-3acc-
437e-aaa5-306838418deb"><name>Task_Red_LAN</name></task><report_format id="a994b278-1f62-11e1-
96ac-406186ea4fc5"><name>XML</name></report_format><report id="b9e9cf8a-1633-4356-8843-
abfa8ad5dfa1"><gmp><version>20.08</version></gmp><sort><field>name<order>ascending</order></fiel
d></sort><filters id=""><term>apply_overrides=0 min_qod=70 first=1 rows=10 sort=name</term><filt
er>High</filter><filter>Medium</filter><filter>Low</filter><filter>Log</filter><filter>Debug</fi
lter><keywords><keyword><column>apply_overrides</column><relation>=</relation><value>0</value></
keyword><keyword><column>min_qod</column><relation>=</relation><value>70</value></keyword><keywo
rd><column>first</column><relation>=</relation><value>1</value></keyword><keyword><column>rows</
column><relation>=</relation><value>10</value></keyword><keyword><column>sort</column><relation>
=</relation><value>name</value></keyword></keywords></filters><severity_class id="d4c74cda-89e1-
11e3-9c29-
406186ea4fc5"><name>nist</name><full_name>NVD Vulnerability Severity Ratings</full_name><severit
y_range><name>None</name><min>0.0</min><max>0.0</max></severity_range><severity_range><name>Low<
/name><min>0.1</min><max>3.9</max></severity_range><severity_range><name>Medium</name><min>4.0</
min><max>6.9</max></severity_range><severity_range><name>High</name><min>7.0</min><max>10.0</max
></severity_range></severity_class><scan_run_status>Done</scan_run_status><hosts><count>254</cou
nt></hosts><closed_cves><count>8</count></closed_cves><vulns><count>461</count></vulns><os><coun
t>4</count></os><apps><count>18</count></apps><ssl_certs><count>5</count></ssl_certs><task id="1
1f35a4c-3acc-437e-aaa5-
306838418deb"><name>Task_Red_LAN</name><comment>Tarea que escanea la red local VMWare</comment><
target id="9c4f261e-a93a-4599-a3a4-
7e8fd55edcf5"><trash>0</trash><name>Red_LAN</name><comment>Red_Virtual-
VMWare</comment></target><progress>100</progress></task><scan><task></task></scan><timestamp>202
1-05-15T21:43:32Z</timestamp><scan_start>2021-05-
15T21:58:55Z</scan_start><timezone>Coordinated Universal Time</timezone><timezone_abbrev>UTC</ti
mezone_abbrev><ports start="1" max="10"><count>34</count><port><host>192.168.0.108</host>80/tcp<
```

```
severity>5.0</severity><threat>Medium</threat></port><port><host>192.168.0.108</host>8009/tcp<se
verity>7.5</severity><threat>High</threat></port><port><host>192.168.0.114</host>80/tcp<severity
>5.0</severity><threat>Medium</threat></port><port><host>192.168.0.104</host>5985/tcp<severity>0
.0</severity><threat>Log</threat></port><port><host>192.168.0.108</host>general/tcp<severity>0.0
</severity><threat>Log</threat></port><port><host>192.168.0.114</host>general/tcp<severity>0.0</
severity><threat>Log</threat></port><port><host>192.168.0.108</host>21/tcp<severity>6.4</severit
y><threat>Medium</threat></port><port><host>192.168.0.104</host>8080/tcp<severity>0.0</severity>
<threat>Log</threat></port></ports><results start="1" max="10"><result id="33ebfc34-3db5-4b02-
82e5-
7ee7ddda2472"><name>Anonymous FTP Login Reporting</name><owner><name>admin</name></owner><modifi
cation_time>2021-05-15T22:19:07Z</modification_time><comment></comment><creation_time>2021-05-
15T22:19:07Z</creation_time><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-
1a76a32ff88e"/><hostname></hostname></host><port>21/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.90
0600"><type>nvt</type><name>Anonymous FTP Login Reporting</name><family>FTP</family><cvss_base>6
.4</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:P/I:P/A:N|summary=Reports if the remote FT
P Server allows anonymous logins.|insight=A host that provides an FTP service may additionally p
rovide Anonymous FTP
solution><refs><ref type="url" id="https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-1999-
0497"/></refs></nvt><scan_nvt_version>2020-08-
24T08:40:10Z</scan_nvt_version><threat>Medium</threat><severity>6.4</severity><qod><value>80</va
lue><type></type></qod><description>It was possible to login to the remote FTP service with the
following anonymous account(s):

anonymous:anonymous@example.com
ftp:anonymous@example.com
</description><original_threat>Medium</original_threat><original_severity>6.4</original_severity
></result><result id="230e59de-d853-4abc-9218-
c818801e1eba"><name>Apache HTTP Server Detection Consolidation</name><owner><name>admin</name></
owner><modification_time>2021-05-
15T22:16:26Z</modification_time><comment></comment><creation_time>2021-05-
15T22:16:26Z</creation_time><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-
1a76a32ff88e"/><hostname></hostname></host><port>general/tcp</port><nvt oid="1.3.6.1.4.1.25623.1
.0.117232"><type>nvt</type><name>Apache HTTP Server Detection Consolidation</name><family>Produc
t detection</family><cvss_base>0.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:N/I:N/A:N|
summary=Consolidation of Apache HTTP Server detections.|insight=|affected=|impact=|solution=|vul
detect=|solution_type=</tags><solution type=''></solution><refs><ref type="url" id="https://http
d.apache.org"/></refs></nvt><scan_nvt_version>2021-02-
25T13:36:35Z</scan_nvt_version><threat>Log</threat><severity>0.0</severity><qod><value>80</value
><type></type></qod><description>Detected Apache HTTP Server

Version:       2.2.8
Location:      80/tcp
CPE:           cpe:/a:apache:http_server:2.2.8

Concluded from version/product identification result:
Server: Apache/2.2.8 (Ubuntu) DAV/2
</description><original_threat>Log</original_threat><original_severity>0</original_severity></re
sult><result id="3c0e6651-6961-4bdf-a601-
569507062fcb"><name>Apache HTTP Server Detection Consolidation</name><owner><name>admin</name></
owner><modification_time>2021-05-
15T22:04:48Z</modification_time><comment></comment><creation_time>2021-05-
15T22:04:48Z</creation_time><host>192.168.0.114<asset asset_id="b72b0a96-b6fa-428d-b861-
af2873d6fde5"/><hostname></hostname></host><port>general/tcp</port><nvt oid="1.3.6.1.4.1.25623.1
.0.117232"><type>nvt</type><name>Apache HTTP Server Detection Consolidation</name><family>Produc
t detection</family><cvss_base>0.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:N/I:N/A:N|
summary=Consolidation of Apache HTTP Server detections.|insight=|affected=|impact=|solution=|vul
detect=|solution_type=</tags><solution type=''></solution><refs><ref type="url" id="https://http
d.apache.org"/></refs></nvt><scan_nvt_version>2021-02-
25T13:36:35Z</scan_nvt_version><threat>Log</threat><severity>0.0</severity><qod><value>80</value
><type></type></qod><description>Detected Apache HTTP Server

Version:       2.4.46
Location:      80/tcp
CPE:           cpe:/a:apache:http_server:2.4.46

Concluded from version/product identification result:
Server: Apache/2.4.46 (Debian)
</description><original_threat>Log</original_threat><original_severity>0</original_severity></re
sult><result id="4b730325-bfd2-4d7e-bd0a-
0d38b54cfbd1"><name>Apache HTTP Server &apos;httpOnly&apos; Cookie Information Disclosure Vulner
ability</name><owner><name>admin</name></owner><modification_time>2021-05-
15T22:32:05Z</modification_time><comment></comment><creation_time>2021-05-
15T22:32:05Z</creation_time><detection><result id="230e59de-d853-4abc-9218-
c818801e1eba"><details><detail><name>product</name><value>cpe:/a:apache:http_server:2.2.8</value
```

```
></detail><detail><name>location</name><value>80/tcp</value></detail><detail><name>source_oid</name><value>1.3.6.1.4.1.25623.1.0.117232</value></detail><detail><name>source_name</name><value>Apache HTTP Server Detection Consolidation</value></detail></details></result></detection><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-1a76a32ff88e"/><hostname></hostname></host><port>80/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.902830"><type>nvt</type><name>Apache HTTP Server &apos;httpOnly&apos; Cookie Information Disclosure Vulnerability</name><family>Web Servers</family><cvss_base>4.3</cvss_base><tags>cvss_base_vector=AV:N/AC:M/Au:N/C:P/I:N/A:N|summary=Apache HTTP Server is prone to a cookie information disclosure vulnerability.|insight=The flaw is due to an error within the default error response for
  status code 400 when no custom ErrorDocument is configured, which can be
  exploited to expose &apos;httpOnly&apos; cookies.|affected=Apache HTTP Server versions 2.2.0 through 2.2.21.|impact=Successful exploitation will allow attackers to obtain sensitive information
  that may aid in further attacks.|solution=Update to Apache HTTP Server version 2.2.22 or later.|vuldetect=|solution_type=VendorFix</tags><solution type='VendorFix'>Update to Apache HTTP Server version 2.2.22 or later.</solution><refs><ref type="cve" id="CVE-2012-0053"/><ref type="bid" id="51706"/><ref type="url" id="http://secunia.com/advisories/47779"/><ref type="url" id="http://www.exploit-db.com/exploits/18442"/><ref type="url" id="http://rhn.redhat.com/errata/RHSA-2012-0128.html"/><ref type="url" id="http://httpd.apache.org/security/vulnerabilities_22.html"/><ref type="url" id="http://svn.apache.org/viewvc?view=revision&amp;revision=1235454"/><ref type="url" id="http://lists.opensuse.org/opensuse-security-announce/2012-02/msg00026.html"/><ref type="cert-bund" id="CB-K15/0080"/><ref type="cert-bund" id="CB-K14/1505"/><ref type="cert-bund" id="CB-K14/0608"/><ref type="dfn-cert" id="DFN-CERT-2015-0082"/><ref type="dfn-cert" id="DFN-CERT-2014-1592"/><ref type="dfn-cert" id="DFN-CERT-2014-0635"/><ref type="dfn-cert" id="DFN-CERT-2013-1307"/><ref type="dfn-cert" id="DFN-CERT-2012-1276"/><ref type="dfn-cert" id="DFN-CERT-2012-1112"/><ref type="dfn-cert" id="DFN-CERT-2012-0928"/><ref type="dfn-cert" id="DFN-CERT-2012-0758"/><ref type="dfn-cert" id="DFN-CERT-2012-0744"/><ref type="dfn-cert" id="DFN-CERT-2012-0568"/><ref type="dfn-cert" id="DFN-CERT-2012-0425"/><ref type="dfn-cert" id="DFN-CERT-2012-0424"/><ref type="dfn-cert" id="DFN-CERT-2012-0387"/><ref type="dfn-cert" id="DFN-CERT-2012-0343"/><ref type="dfn-cert" id="DFN-CERT-2012-0332"/><ref type="dfn-cert" id="DFN-CERT-2012-0306"/><ref type="dfn-cert" id="DFN-CERT-2012-0264"/><ref type="dfn-cert" id="DFN-CERT-2012-0203"/><ref type="dfn-cert" id="DFN-CERT-2012-0188"/></refs></nvt><scan_nvt_version>2021-02-25T13:36:35Z</scan_nvt_version><threat>Medium</threat><severity>4.3</severity><qod><value>99</value><type></type></qod><description></description><original_threat>Medium</original_threat><original_severity>4.3</original_severity></result><result id="2eab52c5-38d6-4449-bb26-409222a85652"><name>Apache JServ Protocol (AJP) v1.3 Detection</name><owner><name>admin</name></owner><modification_time>2021-05-15T22:24:10Z</modification_time><comment></comment><creation_time>2021-05-15T22:24:10Z</creation_time><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-1a76a32ff88e"/><hostname></hostname></host><port>8009/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.108082"><type>nvt</type><name>Apache JServ Protocol (AJP) v1.3 Detection</name><family>Service detection</family><cvss_base>0.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:N/I:N/A:N|summary=The script detects a service supporting the
  Apache JServ Protocol (AJP) version 1.3.|insight=|affected=|impact=|solution=|vuldetect=|solution_type=</tags><solution type=''></solution></nvt><scan_nvt_version>2020-11-10T15:30:28Z</scan_nvt_version><threat>Log</threat><severity>0.0</severity><qod><value>80</value><type></type></qod><description>A service supporting the Apache JServ Protocol (AJP) v1.3 seems to be running on this port.
</description><original_threat>Log</original_threat><original_severity>0</original_severity></result><result id="2226fddb-81dc-4dd6-ab23-6e9a4782dcc8"><name>Apache /server-status accessible</name><owner><name>admin</name></owner><modification_time>2021-05-15T22:07:43Z</modification_time><comment></comment><creation_time>2021-05-15T22:07:43Z</creation_time><host>192.168.0.114<asset asset_id="b72b0a96-b6fa-428d-b861-af2873d6fde5"/><hostname></hostname></host><port>80/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.10677"><type>nvt</type><name>Apache /server-status accessible</name><family>Web application abuses</family><cvss_base>5.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:P/I:N/A:N|summary=Requesting the URI /server-status provides information
  on the server activity and performance.|insight=server-status is a Apache HTTP Server handler provided by the
  &apos;mod_status&apos; module and used to retrieve the server&apos;s activity and performance.|affected=- All Apache installations with an enabled &apos;mod_status&apos; module

-
 If the FreedomBox software is running on the target update the software to a later version</solution><refs><ref type="cve" id="CVE-2020-25073"/><ref type="url" id="https://httpd.apache.org/docs/current/mod/mod_status.html"/></refs></nvt><scan_nvt_version>2020-12-01T08:30:14Z</scan_nvt_version><threat>Medium</threat><severity>5.0</severity><qod><value>99</value><type></type></qod><description>Vulnerable URL: http://192.168.0.114/server-status
</description><original_threat>Medium</original_threat><original_severity>5</original_severity></result><result id="35a236f5-65a9-4a9b-a88d-
```

d747d0f3eada"><name>Apache Tomcat AJP RCE Vulnerability (Ghostcat)</name><owner><name>admin</name></owner><modification_time>2021-05-
15T22:29:58Z</modification_time><comment></comment><creation_time>2021-05-
15T22:29:58Z</creation_time><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-
1a76a32ff88e"/><hostname></hostname></host><port>8009/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.
143545"><type>nvt</type><name>Apache Tomcat AJP RCE Vulnerability (Ghostcat)</name><family>Web a
pplication abuses</family><cvss_base>7.5</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:P/I:
P/A:P|summary=Apache Tomcat is prone to a remote code execution vulnerability
  (dubbed 'Ghostcat') in the AJP connector.|insight=Apache Tomcat server has a file co
ntaining vulnerability, which can be used by
  an attacker to read or include any files in all webapp directories on Tomcat, such as webapp c
onfiguration files
  or source code.|affected=Apache Tomcat versions prior 7.0.100, 8.5.51 or 9.0.31 when the AJP c
onnector
  is enabled.

  Other products like JBoss or Wildfly which are using Tomcat might be affected as well.|impact=
|solution=Update Apache Tomcat to version 7.0.100, 8.5.51, 9.0.31 or later. For other products
  using Tomcat please contact the vendor for more information on fixed versions.|vuldetect=Sends
 a crafted AJP request and checks the response.|solution_type=VendorFix</tags><solution type='Ve
ndorFix'>Update Apache Tomcat to version 7.0.100, 8.5.51, 9.0.31 or later. For other products
  using Tomcat please contact the vendor for more information on fixed versions.</solution><refs
><ref type="cve" id="CVE-2020-
1938"/><ref type="url" id="https://lists.apache.org/thread.html/r7c6f492fbd39af34a68681dbbba0468
490ff1a97a1bd79c6a53610ef%40%3Cannounce.tomcat.apache.org%3E"/><ref type="url" id="https://www.c
haitin.cn/en/ghostcat"/><ref type="url" id="https://www.cnvd.org.cn/flaw/show/CNVD-2020-
10487"/><ref type="url" id="https://github.com/YDHCUI/CNVD-2020-10487-Tomcat-Ajp-
lfi"/><ref type="url" id="https://securityboulevard.com/2020/02/patch-your-tomcat-and-jboss-
instances-to-protect-from-ghostcat-vulnerability-cve-2020-1938-
and/"/><ref type="url" id="https://tomcat.apache.org/tomcat-7.0-
doc/changelog.html"/><ref type="url" id="https://tomcat.apache.org/tomcat-8.5-
doc/changelog.html"/><ref type="url" id="https://tomcat.apache.org/tomcat-9.0-
doc/changelog.html"/><ref type="cert-bund" id="CB-K20/0711"/><ref type="cert-bund" id="CB-
K20/0705"/><ref type="cert-bund" id="CB-K20/0693"/><ref type="cert-bund" id="CB-
K20/0555"/><ref type="cert-bund" id="CB-K20/0543"/><ref type="cert-bund" id="CB-
K20/0154"/><ref type="dfn-cert" id="DFN-CERT-2020-1508"/><ref type="dfn-cert" id="DFN-CERT-2020-
1413"/><ref type="dfn-cert" id="DFN-CERT-2020-1276"/><ref type="dfn-cert" id="DFN-CERT-2020-
1134"/><ref type="dfn-cert" id="DFN-CERT-2020-0850"/><ref type="dfn-cert" id="DFN-CERT-2020-
0835"/><ref type="dfn-cert" id="DFN-CERT-2020-0821"/><ref type="dfn-cert" id="DFN-CERT-2020-
0569"/><ref type="dfn-cert" id="DFN-CERT-2020-0557"/><ref type="dfn-cert" id="DFN-CERT-2020-
0501"/><ref type="dfn-cert" id="DFN-CERT-2020-0381"/></refs></nvt><scan_nvt_version>2020-11-
10T09:46:51Z</scan_nvt_version><threat>High</threat><severity>7.5</severity><qod><value>99</valu
e><type></type></qod><description>It was possible to read the file "/WEB-
INF/web.xml" through the AJP connector.

Result:

AB 8\x0004 Ã\x0088 \x0002OK \x0001 \x000CContent-Type \x001Ctext/html;charset=ISO-8859-
1 AB\x001FÃ¾\x0003\x001FÃ¸<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements. See the NOTICE file distributed with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version 2.0
  (the "License"); you may not use this file except in compliance with
  the License. You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  <b><a href="mailto:users@tomcat.apache.org">users@tomc
</description><original_threat>High</original_threat><original_severity>7.5</original_severity><
/result><result id="a371e237-6fa1-4fbb-a97e-
f718182d5446"><name>awiki Multiple Local File Include Vulnerabilities</name><owner><name>admin</
name></owner><modification_time>2021-05-
15T22:30:03Z</modification_time><comment></comment><creation_time>2021-05-
15T22:30:03Z</creation_time><host>192.168.0.108<asset asset_id="6a61283c-5434-4310-b934-
1a76a32ff88e"/><hostname></hostname></host><port>80/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.10
3210"><type>nvt</type><name>awiki Multiple Local File Include Vulnerabilities</name><family>Web
application abuses</family><cvss_base>5.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:P/I
:N/A:N|summary=awiki is prone to multiple local file-include vulnerabilities because
  it fails to properly sanitize user-
supplied input.|insight=|affected=awiki 20100125 is vulnerable. Other versions may also be affec
ted.|impact=An attacker can exploit this vulnerability to obtain potentially
  sensitive information and execute arbitrary local scripts in the context of the webserver proc
ess.

```
  This may allow the attacker to compromise the application and the host. Other attacks are also
 possible.|solution=No known solution was made available for at least one year
  since the disclosure of this vulnerability. Likely none will be provided anymore. General solu
tion
  options are to upgrade to a newer release, disable respective features, remove the product or
  replace the product by another one.|vuldetect=|solution_type=WillNotFix</tags><solution type='
WillNotFix'>No known solution was made available for at least one year
  since the disclosure of this vulnerability. Likely none will be provided anymore. General solu
tion
  options are to upgrade to a newer release, disable respective features, remove the product or
  replace the product by another one.</solution><refs><ref type="bid" id="49187"/><ref type="url
" id="https://www.exploit-
db.com/exploits/36047/"/><ref type="url" id="http://www.securityfocus.com/bid/49187"/><ref type=
"url" id="http://www.kobaonline.com/awiki/"/></refs></nvt><scan_nvt_version>2021-04-
16T06:57:08Z</scan_nvt_version><threat>Medium</threat><severity>5.0</severity><qod><value>99</va
lue><type></type></qod><description>Vulnerable URL: http://192.168.0.108/mutillidae/index.php?pa
ge=/etc/passwd
</description><original_threat>Medium</original_threat><original_severity>5</original_severity><
/result><result id="246a6444-bf71-44bd-aaee-
4a042bb1f2e8"><name>CGI Scanning Consolidation</name><owner><name>admin</name></owner><modificat
ion_time>2021-05-15T22:17:41Z</modification_time><comment></comment><creation_time>2021-05-
15T22:17:41Z</creation_time><host>192.168.0.104<asset asset_id="fd87efd4-6ef0-482b-b0db-
64273f7df238"/><hostname></hostname></host><port>8080/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.
111038"><type>nvt</type><name>CGI Scanning Consolidation</name><family>Web application abuses</f
amily><cvss_base>0.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:N/I:N/A:N|summary=The sc
ript consolidates various information for CGI scanning.

  This information is based on the following scripts / settings:

  - HTTP-Version Detection (OID: 1.3.6.1.4.1.25623.1.0.100034)

  - No 404 check (OID: 1.3.6.1.4.1.25623.1.0.10386)

  - Web mirroring / webmirror.nasl (OID: 1.3.6.1.4.1.25623.1.0.10662)

  - Directory Scanner / DDI_Directory_Scanner.nasl (OID: 1.3.6.1.4.1.25623.1.0.11032)

  -
 The configured &apos;cgi_path&apos; within the &apos;Scanner Preferences&apos; of the scan conf
ig in use

  -
 The configured &apos;Enable CGI scanning&apos;, &apos;Enable generic web application scanning&a
pos; and
    &apos;Add historic /scripts and /cgi-bin to directories for CGI scanning&apos; within the
    &apos;Global variable settings&apos; of the scan config in use

  If you think any of this information is wrong please report it to the referenced community por
tal.|insight=|affected=|impact=|solution=|vuldetect=|solution_type=</tags><solution type=''></so
lution><refs><ref type="url" id="https://community.greenbone.net/c/vulnerability-
tests"/></refs></nvt><scan_nvt_version>2020-11-
19T14:17:11Z</scan_nvt_version><threat>Log</threat><severity>0.0</severity><qod><value>80</value
><type></type></qod><description>The Hostname/IP &quot;192.168.0.104&quot; was used to access th
e remote host.

Generic web application scanning is disabled for this host via the &quot;Enable generic web appl
ication scanning&quot; option within the &quot;Global variable settings&quot; of the scan config
 in use.

Requests to this service are done via HTTP/1.1.

This service seems to be able to host PHP scripts.

This service seems to be able to host ASP scripts.

The User-Agent &quot;Mozilla/5.0 [en] (X11, U; OpenVAS-
VT 20.8.1)&quot; was used to access the remote host.

Historic /scripts and /cgi-
bin are not added to the directories used for CGI scanning. You can enable this again with the &
quot;Add historic /scripts and /cgi-
bin to directories for CGI scanning&quot; option within the &quot;Global variable settings&quot;
 of the scan config in use.
```

```
The following directories were used for CGI scanning:

http://192.168.0.104:8080/

While this is not, in and of itself, a bug, you should manually inspect these directories to ens
ure that they are in compliance with company security standards

</description><original_threat>Log</original_threat><original_severity>0</original_severity></re
sult><result id="14f42824-e474-4a44-ae25-
04a5aa7dc8e6"><name>CGI Scanning Consolidation</name><owner><name>admin</name></owner><modificat
ion_time>2021-05-15T22:17:41Z</modification_time><comment></comment><creation_time>2021-05-
15T22:17:41Z</creation_time><host>192.168.0.104<asset asset_id="fd87efd4-6ef0-482b-b0db-
64273f7df238"/><hostname></hostname></host><port>5985/tcp</port><nvt oid="1.3.6.1.4.1.25623.1.0.
111038"><type>nvt</type><name>CGI Scanning Consolidation</name><family>Web application abuses</f
amily><cvss_base>0.0</cvss_base><tags>cvss_base_vector=AV:N/AC:L/Au:N/C:N/I:N/A:N|summary=The sc
ript consolidates various information for CGI scanning.

  This information is based on the following scripts / settings:

  - HTTP-Version Detection (OID: 1.3.6.1.4.1.25623.1.0.100034)

  - No 404 check (OID: 1.3.6.1.4.1.25623.1.0.10386)

  - Web mirroring / webmirror.nasl (OID: 1.3.6.1.4.1.25623.1.0.10662)

  - Directory Scanner / DDI_Directory_Scanner.nasl (OID: 1.3.6.1.4.1.25623.1.0.11032)

  -
 The configured &apos;cgi_path&apos; within the &apos;Scanner Preferences&apos; of the scan conf
ig in use

  -
 The configured &apos;Enable CGI scanning&apos;, &apos;Enable generic web application scanning&a
pos; and
    &apos;Add historic /scripts and /cgi-bin to directories for CGI scanning&apos; within the
    &apos;Global variable settings&apos; of the scan config in use

  If you think any of this information is wrong please report it to the referenced community por
tal.|insight=|affected=|impact=|solution=|vuldetect=|solution_type=</tags><solution type=''></so
lution><refs><ref type="url" id="https://community.greenbone.net/c/vulnerability-
tests"/></refs></nvt><scan_nvt_version>2020-11-
19T14:17:11Z</scan_nvt_version><threat>Log</threat><severity>0.0</severity><qod><value>80</value
><type></type></qod><description>The Hostname/IP &quot;192.168.0.104&quot; was used to access th
e remote host.

Generic web application scanning is disabled for this host via the &quot;Enable generic web appl
ication scanning&quot; option within the &quot;Global variable settings&quot; of the scan config
 in use.

Requests to this service are done via HTTP/1.1.

This service seems to be able to host PHP scripts.

This service seems to be able to host ASP scripts.

The User-Agent &quot;Mozilla/5.0 [en] (X11, U; OpenVAS-
VT 20.8.1)&quot; was used to access the remote host.

Historic /scripts and /cgi-
bin are not added to the directories used for CGI scanning. You can enable this again with the &
quot;Add historic /scripts and /cgi-
bin to directories for CGI scanning&quot; option within the &quot;Global variable settings&quot;
 of the scan config in use.

The following directories were used for CGI scanning:

http://192.168.0.104:5985/

While this is not, in and of itself, a bug, you should manually inspect these directories to ens
ure that they are in compliance with company security standards
```


* Nota: Se omite los nvt ya que es mucha la información que se presenta.

Openvas_parsed_report.log

```
host_id=192.168.0.108 port="21/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.900600" th
reat="Medium" severity="6.4"
host_id=192.168.0.108 port="80/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.902830" th
reat="Medium" severity="4.3"
host_id=192.168.0.108 port="8009/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.143545"
threat="High" severity="7.5"
host_id=192.168.0.108 port="80/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.103210" th
reat="Medium" severity="5.0"
host_id=192.168.0.114 port="80/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.10677" thr
eat="Medium" severity="5.0"
host_id=192.168.0.104 port="8080/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.111038"
threat="Log" severity="0.0"
host_id=192.168.0.104 port="5985/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.111038"
threat="Log" severity="0.0"
host_id=192.168.0.108 port="general/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.11723
2" threat="Log" severity="0.0"
host_id=192.168.0.114 port="general/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.11723
2" threat="Log" severity="0.0"
host_id=192.168.0.108 port="8009/tcp" nvt_oid="1.3.6.1.4.1.25623.1.0.108082"
threat="Log" severity="0.0"
host_id=192.168.0.108 port="21/tcp" ref_type="url" ref_id="https://web.nvd.ni
st.gov/view/vuln/detail?vulnId=CVE-1999-0497"
host_id=192.168.0.108 port="general/tcp" ref_type="url" ref_id="https://httpd
.apache.org"
host_id=192.168.0.114 port="general/tcp" ref_type="url" ref_id="https://httpd
.apache.org"
host_id=192.168.0.108 port="80/tcp" ref_type="cve" ref_id="CVE-2012-0053"
host_id=192.168.0.108 port="80/tcp" ref_type="bid" ref_id="51706"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://secunia.com
/advisories/47779"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://www.exploit
-db.com/exploits/18442"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://rhn.redhat.
com/errata/RHSA-2012-0128.html"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://httpd.apach
e.org/security/vulnerabilities_22.html"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://svn.apache.
org/viewvc?view=revision&revision=1235454"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://lists.opens
use.org/opensuse-security-announce/2012-02/msg00026.html"
host_id=192.168.0.108 port="80/tcp" ref_type="cert-bund" ref_id="CB-K15/0080"
host_id=192.168.0.108 port="80/tcp" ref_type="cert-bund" ref_id="CB-K14/1505"
host_id=192.168.0.108 port="80/tcp" ref_type="cert-bund" ref_id="CB-K14/0608"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2015-0082"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2014-1592"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2014-0635"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2013-1307"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-1276"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-1112"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0928"
```

```
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0758"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0744"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0568"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0425"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0424"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0387"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0343"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0332"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0306"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0264"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0203"
host_id=192.168.0.108 port="80/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2012-0188"
host_id=192.168.0.114 port="80/tcp" ref_type="cve" ref_id="CVE-2020-25073"
host_id=192.168.0.114 port="80/tcp" ref_type="url" ref_id="https://httpd.apac
he.org/docs/current/mod/mod_status.html"
host_id=192.168.0.108 port="8009/tcp" ref_type="cve" ref_id="CVE-2020-1938"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://lists.ap
ache.org/thread.html/r7c6f492fbd39af34a68681dbbba0468490ff1a97a1bd79c6a53610e
f%40%3Cannounce.tomcat.apache.org%3E"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://www.chai
tin.cn/en/ghostcat"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://www.cnvd
.org.cn/flaw/show/CNVD-2020-10487"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://github.c
om/YDHCUI/CNVD-2020-10487-Tomcat-Ajp-lfi"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://security
boulevard.com/2020/02/patch-your-tomcat-and-jboss-instances-to-protect-from-
ghostcat-vulnerability-cve-2020-1938-and/"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://tomcat.a
pache.org/tomcat-7.0-doc/changelog.html"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://tomcat.a
pache.org/tomcat-8.5-doc/changelog.html"
host_id=192.168.0.108 port="8009/tcp" ref_type="url" ref_id="https://tomcat.a
pache.org/tomcat-9.0-doc/changelog.html"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0711"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0705"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0693"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0555"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0543"
host_id=192.168.0.108 port="8009/tcp" ref_type="cert-bund" ref_id="CB-
K20/0154"
```

```
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-1508"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-1413"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-1276"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-1134"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0850"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0835"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0821"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0569"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0557"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0501"
host_id=192.168.0.108 port="8009/tcp" ref_type="dfn-cert" ref_id="DFN-CERT-
2020-0381"
host_id=192.168.0.108 port="80/tcp" ref_type="bid" ref_id="49187"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="https://www.exploi
t-db.com/exploits/36047/"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://www.securit
yfocus.com/bid/49187"
host_id=192.168.0.108 port="80/tcp" ref_type="url" ref_id="http://www.kobaonl
ine.com/awiki/"
host_id=192.168.0.104 port="8080/tcp" ref_type="url" ref_id="https://communit
y.greenbone.net/c/vulnerability-tests"
host_id=192.168.0.104 port="5985/tcp" ref_type="url" ref_id="https://communit
y.greenbone.net/c/vulnerability-tests"
host_id=192.168.0.108 port="21/tcp" type="Mitigation"
host_id=192.168.0.108 port="80/tcp" type="VendorFix"
host_id=192.168.0.114 port="80/tcp" type="Mitigation"
host_id=192.168.0.108 port="8009/tcp" type="VendorFix"
host_id=192.168.0.108 port="80/tcp" type="WillNotFix"
```