

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DE TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Trabajo Fin de Grado

Estudio comparativo entre OpenVas y Wazuh



Autor: Antonio Barquero Pastor

Director: María Dolores Cano Baños

Codirector: Igor Alexander Bello Tasic

Agradecimientos

A mi familia, por soportar momentos duros durante todo mi trayecto universitario, pero sobre todo a mi padre por alentarme a seguir a pesar de las circunstancias que se han ido sucediendo.

A mis amigos, que han estado apoyándome a lo largo de toda la carrera y animándome a seguir.

A mi directora de proyecto Lola por preocuparse por mis todos estos meses.

Han sido unos últimos años complicados debido al covid.

Mil gracias.

Contenido

Capítulo 1: Introducción y Objetivos.....	7
1.1 Introducción	7
1.2 Objetivo del TFG.....	8
1.3 Resumen del proyecto	8
Capítulo 2. Base teórica.....	9
2.1 Conceptos teóricos – SIEM y OSSIM	9
2.2 Funcionamiento y Arquitectura de OpenVas.....	10
2.3 Funcionamiento y Arquitectura de Wazuh	13
2.4 Características principales de Wazuh.....	14
Capítulo 3. Entorno de pruebas	17
3.1 Instalación VirtualBox.....	17
3.2 Instalación de los equipos virtuales	17
3.2.1 Metasploitable 3	18
3.2.2 Kali Linux.....	19
3.2.3 Windows 10, Android 8.1 y Windows Server 2008.....	20
3.2.4 Configuración del entorno de pruebas	21
3.3 Instalación y configuración de OpenVas	23
3.4 Instalación y configuración de Wazuh	33
3.4.1 Instalación de los agentes (Windows y Linux).....	34
3.4.2 Configuración del detector de vulnerabilidades	36
Capítulo 4. Resultados y comparación entre OpenVas y Wazuh	39
4.1 Resultados del estudio de OpenVas.....	39
4.1.1 Vulnerabilidades graves	41
4.1.2 Vulnerabilidades medias	43
4.1.3 Vulnerabilidades leves	46
4.2 Resultados del estudio de Wazuh	47
4.2.1 Vulnerabilities	47
4.2.2 Integrity monitoring	50
4.2.3 VirusTotal	52
4.2.4 MITRE ATT&CK	54
4.3 Diferencias y similitudes entre Wazuh y OpenVas.....	55
4.4 Conclusiones.....	56
Referencias.....	58

Índice de imágenes

Imagen 1: Herramientas de OSSIM	10
Imagen 2: Actualización de la información de OpenVas.....	11
Imagen 3: Arquitectura de OpenVas.....	12
Imagen 4: Arquitectura de Wazuh	14
Imagen 5: Paquete de instalación de VirtualBox	17
Imagen 6: Topología de la Red Virtual	17
Imagen 7: Instalación del plugin de Vagrant.....	18
Imagen 8: Creación de la carpeta metasploitable3-workspace.....	18
Imagen 9: Descarga del archivo metasploitable3	19
Imagen 10: Interfaz de comandos metasploitable3	19
Imagen 11: Paquete de instalación de Kali Linux.....	20
Imagen 12: Interfaz de Kali Linux	20
Imagen 13: Interfaz de VirtualBox con todas las maquinas instaladas.....	21
Imagen 14: Proceso de configuración de red (1)	21
Imagen 15: Proceso de configuración de red (2)	22
Imagen 16: Proceso de configuración de red (3)	22
Imagen 17: Actualización de los paquetes del sistema.....	23
Imagen 18: Validación de las actualizaciones del sistema	23
Imagen 19: Validación de las actualizaciones del sistema	24
Imagen 20: Proceso de obtención de la contraseña de acceso a OpenVas.....	24
Imagen 21: Contraseña de acceso a OpenVas	24
Imagen 22: Inicio de OpenVas mediante comandos.....	25
Imagen 23: Interfaz de acceso con credenciales a OpenVas	25
Imagen 24: Interfaz de Greenbone (OpenVas)	26
Imagen 25: Configuración de escaneo de vulnerabilidades (1)	26
Imagen 26: Configuración de escaneo de vulnerabilidades (2)	27
Imagen 27: Configuración de escaneo de vulnerabilidades (3)	27
Imagen 28: Configuración de escaneo de vulnerabilidades (3)	28
Imagen 29: Configuración de escaneo de vulnerabilidades (4)	29
Imagen 30: Resultados de todas las maquinas escaneadas.....	30
Imagen 31: Listado de vulnerabilidades (1)	30
Imagen 32: Fecha y duración del escaneo	31
Imagen 33: Listado de vulnerabilidades (2)	31
Imagen 34: Información sobre la vulnerabilidad	31
Imagen 35: Información sobre el host	32
Imagen 36: Información sobre los puertos	32
Imagen 37: Aplicaciones escaneadas con CPE	32
Imagen 38: Sistema operativo escaneado	32
Imagen 39: CVEs.....	32
Imagen 40: Certificados TLS	33
Imagen 41: Mensajes de error	33
Imagen 42: Interfaz del servidor Wazuh	34
Imagen 43: Interfaz principal de Wazuh	34
Imagen 44: Instalación del agente Ubuntu	35

Imagen 45: Activación del agente Ubuntu	35
Imagen 46: Registro del Agente de Windows	35
Imagen 47: Instalador de Wazuh	36
Imagen 48: Mensaje de actualización de la base de datos de un proveedor	38
Imagen 49: Mensaje de final de actualización	38
Imagen 50: Final del escaneo	38
Imagen 51: Interfaz de resultados del análisis de vulnerabilidades (metasploitable3).....	38
Imagen 52: Test Windows 10	40
Imagen 53: Test Windows Server.....	40
Imagen 54: Test Android 8.1	40
Imagen 55: Test Metasploitable 3.....	40
Imagen 56: PDF generado por OpenVas mostrando una vulnerabilidad	47
Imagen 57: Visualización de las alertas Wazuh.....	48
Imagen 58: Descripción, impacto y referencias sobre una de las vulnerabilidades de Wazuh ..	48
Imagen 59: Id y nivel de la regla.....	48
Imagen 60: Alertas del módulo Integrity Monitoring	50
Imagen 61: Contraseña de activación VirusTotal.....	52
Imagen 62: Resultados de VirusTotal (1)	53
Imagen 63: Resultados de VirusTotal (2)	53
Imagen 64: Alertas de seguridad del módulo MITRE ATT&CK	54
Imagen 65: Descripción del ataque por fuerza bruta	54

Capítulo 1: Introducción y Objetivos

1.1 Introducción

Lo primero que tenemos que preguntarnos es ¿Qué es la seguridad informática o ciberseguridad?

En la actualidad uno de los temas que más relevancia ha cobrado en los últimos años para las empresas es este concepto, el cual podemos definir como todos los procesos que se pueden llevar a cabo para proteger la privacidad e integridad de la información, tanto de manera física como en la nube.

Hoy en día, cualquier empresa relacionada con cualquier sector, debe proteger su información de cualquier acceso no autorizado. Las pequeñas y grandes empresas están invirtiendo mucho dinero en este ámbito, pero sigue sin ser suficiente. Se está anteponiendo la rapidez de procesos y la productividad a cualquier medida de seguridad, creando agujeros susceptibles de ser usados para acceder a información valiosa. Esto produce grandes pérdidas económicas afectando a la reputación de cualquier empresa al verse afectados datos de procesos, bienes de socios y clientes, así como de trabajadores, siendo esta información manipulada y por consiguiente susceptible de ser comercializados los datos y puestos a la venta a cualquier postor.

Según el responsable de servicios de seguridad del Instituto Nacional de Ciberseguridad Español (INCIBE), Marco Lozano, “en 2020 este organismo gestionó 133.155 incidentes, un 24% más que en el año anterior. Entre los más afectados figuran las pequeñas empresas y los autónomos. Para blindarse ante amenazas que pueden costar miles de euros, un empresario debería invertir entre 500 y 800 euros al mes en la gestión de estos servicios”.

La pandemia actual ha hecho que las empresas se adapten de manera muy rápida al mundo digital sin realizar ninguna infraestructura de seguridad. Según la empresa **Nuvix Strategic Consulting**, radicada en 14 países y experta en integración tecnológica de las empresas, “las pérdidas económicas sufridas por ataques informáticos supondrían la pérdida de entre los 3000 y los 75000 euros por siniestro declarado”.

La clave es conseguir detectar la fragilidad del sistema antes de que se produzca el ataque, y es aquí donde entran los detectores de vulnerabilidades. Hay múltiples escáneres que gracias a diferentes bases de datos cruzadas se consigue priorizar el nivel de riesgo del sistema y actuar en consecuencia, sugiriendo una solución para mitigar los riesgos.

Tales medidas deberían ser implementadas por todas las empresas y entidades que manejen información relevante, que son la gran mayoría, evitando accesos de terceros y pérdida de información.

1.2 Objetivo del TFG

El objetivo de este TFG es estudiar de forma experimental las prestaciones que ofrecen las herramientas OpenVas y Wazuh para así conseguir hacer una comparativa entre ambas.

Aprenderemos a crear nuestra propia red de ordenadores virtual a partir la herramienta VirtualBox, conseguiremos manejar perfectamente el entorno de las herramientas OpenVas y Wazuh para así en un futuro poder implementarlas en proyectos para las empresas y por último perfeccionaremos el nivel de manejo en Linux.

1.3 Resumen del proyecto

En un primer lugar, explicaremos todos los conceptos teóricos necesarios para comprender todo el desarrollo del proyecto. Conoceremos de manera general el funcionamiento y características de las herramientas de detección de vulnerabilidades que nos ocupan, conocidas como Wazuh y OpenVas, así como sus diferencias más importantes.

A continuación, desarrollaremos más en profundidad ambas herramientas. Crearemos un entorno virtual con distintos sistemas operativos, explicando toda su configuración y seguiremos paso a paso la instalación y configuración de las mismas.

Por último, evaluaremos los resultados obtenidos y sacaremos las conclusiones adecuadas, terminando con una comparativa de resultados.

Capítulo 2. Base teórica

2.1 Conceptos teóricos – SIEM y OSSIM

Antes de explicar las diferencias entre OpenVas y Wazuh hay que comprender los siguientes términos.

SIEM (Security Information and Event Management)

Herramienta que permite interpretar de forma centralizada las disposiciones de seguridad pertinentes, permite reunir, regular y relacionar eventos. Ofrece inteligencia de seguridad, elimina los falsos positivos, evalúa el impacto de los ataques, concentra la información e incorpora herramientas de detección de amenazas.

Las amenazas detectadas por esta herramienta se pueden dividir en:

- Vulnerabilidades y protocolos sensibles
- Error de configuración del administrador
- Introducción de errores por el usuario de forma consciente o no
- Amenazas internas y externas

Uno de los SIEM más relevantes es OSSIM.

OSSIM (Open Source Security Information Management)

Es un conjunto de herramientas de licencia pública bajo la versión de AlienVault (empresa privada del campo de la ciberseguridad). OSSIM está diseñado para asistir al administrador a proteger los dispositivos, detectar y prevenir intrusiones, siendo útil para controlar equipos de red comprometidos. Puede correlacionar datos de todas las herramientas para identificar patrones y actuar sobre ellos.

Posee su propio motor de correlación que cruza los eventos de seguridad que le llegan de sus herramientas productoras de detección de intrusos (Snort y OSSEC)

La imagen 1 muestra algunas de las herramientas incluidas en OSSIM.

Una de las herramientas que integra OSSIM es **OpenVas (Escáner de Vulnerabilidades)**.

Herramienta	Uso
Arpwatch	Anomalías en direcciones MAC
P0f	Identificación pasiva de OS
Pads	Detección de anomalías en servicios
OpenVas	Escaneo de vulnerabilidades
Snort	Sistema de detección de intrusos (IDS)
Spade	Detección de anomalías en paquetes
Tcptrack	Obtención de la información de las sesiones
Ntop	Detección de anomalías en el comportamiento
Nagios	Disponibilidad de host y servicios
nfSen	Visor de flujos de red para detección de anomalías en la red
Osiris	Sistema de detección de intrusos basado en host (HIDS)
Snare	Colección de logs en Windows
OSSEC	HIDS

Imagen 1: Herramientas de OSSIM

2.2 Funcionamiento y Arquitectura de OpenVas

OpenVAS es un escáner de vulnerabilidades completo. Sus características incluyen, verificación de autenticación, autenticación cero, ajuste de rendimiento para escaneo a gran escala, un poderoso lenguaje de programación interno que le permite verificar cualquier tipo de agujeros de seguridad y una amplia variedad de protocolos industriales y de internet de alto y bajo nivel.

Es una herramienta gratuita de evaluación de vulnerabilidades y servicios que se puede usar sola o como parte del kit de herramientas de seguridad incorporadas en OSSIM. Como resultado, podemos analizar PCs, servidores locales o remotos y generar varios tipos de informes sobre las vulnerabilidades detectadas. También agrega un motor de correlación para analizar lo que se haya identificado y encontrar las soluciones correctas

El corazón de esta arquitectura es el Escáner OpenVas orientada al servicio. Esto está asegurado con el protocolo SSL (Secure Sockets Layer). El escáner ejecuta los NVTs – Network Vulnerability Test (Pruebas de Vulnerabilidad de Redes) los cuales tienen actualizaciones diarias (Imagen 2). Las NVTs son semejantes a plugins que son descargados y permiten actualizar la base de datos de OpenVas para así poder detectar las vulnerabilidades más recientemente descubiertas.

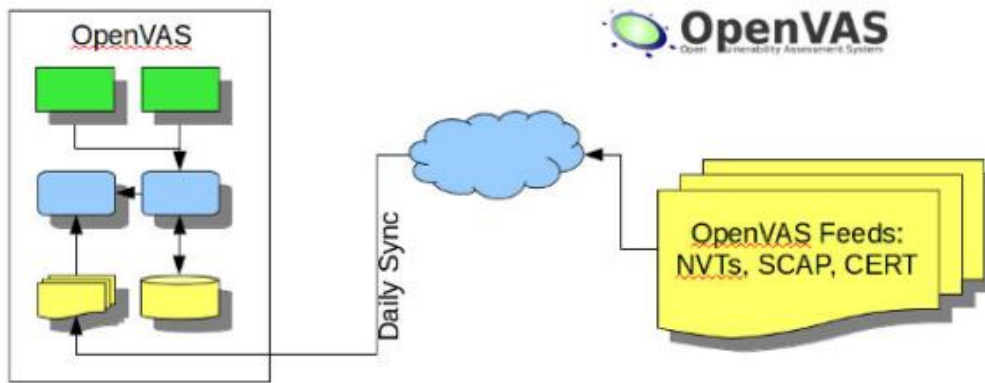


Imagen 2: Actualización de la información de OpenVas

Actualmente hay más de 34000 tests (NVTs) en dicha base de datos que van creciendo diariamente. Además de los NVTs, OpenVas cruza otras bases de datos como pueden ser los CVE (Common Vulnerabilities and Exposures), para obtener mejores resultados y mejorar la calidad de la herramienta

Las características clave que encontramos en OpenVAS incluyen:

- Compatibilidad con el protocolo SSL
- Podemos hacer escaneos programados
- Podemos reiniciar o detener los escaneos en cualquier momento
- Admite varios idiomas
- Compatibilidad con HTTP y HTTPS
- Multiplataforma
- Informes claros y completos
- Podemos escanear varias computadoras simultáneamente
- Podemos gestionar usuarios desde el panel de control

Se desarrollan las funciones a continuación.

OpenVAS funciona principalmente con 3 módulos:

- Un módulo de escaneo (**OpenVas Scanner**), el cual se encarga de ejecutar los análisis de las vulnerabilidades.
- Un módulo cliente, usado como interfaz gráfica (**OpenVas CLI**) necesario para configurar OpenVAS y mostrar los resultados obtenidos o ejecutar informes.
- Otro cliente (**Greenbone Security Assistant (GSA)**), es un servicio que ofrece una interfaz para navegadores y convierte trazas del protocolo OMP directamente a HTML.
- Un módulo administrador (**OpenVas Manager**), que es el responsable de interactuar con todos los demás (Scanner, Cliente, Framework, CLI...) a través de OMP (**OpenVAS Management Protocol**)

OpenVas Scanner

- El escáner ejecuta una prueba de vulnerabilidad en tiempo real de manera muy eficiente
- Puede manejar más de un host de destino a la vez
- Se proporciona el protocolo de transferencia OpenVas (OTP)
- SSL es compatible con OTP

OpenVas Manager

- Maneja la base de datos SQL donde se almacenan todos los resultados y configuraciones del escaneo
- Controla el escáner a través de OTP y ofrece el protocolo de administración OpenVas (OMP) basado en XML
- Puede detener, pausar o reanudar las operaciones de escaneo
- Se encarga de la gestión de usuarios, incluida la gestión a nivel de grupo y la gestión del control de acceso
- Se permite el uso del plugin Nagios, que es un sistema de monitorización de red.

OpenVas CLI

- Es una herramienta de línea de comandos y actúa como cliente para OMP. Puede ejecutarse en Windows o Linux

La siguiente imagen (Imagen 3) muestra la arquitectura de OpenVas.

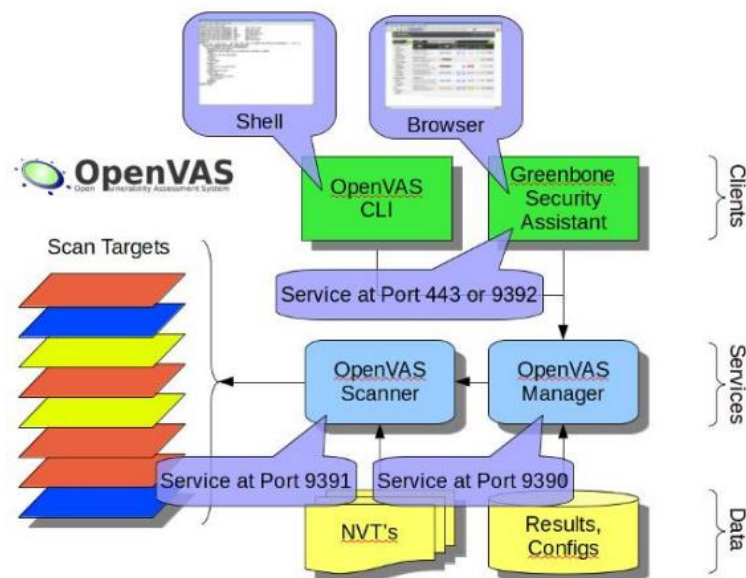


Imagen 3: Arquitectura de OpenVas

2.3 Funcionamiento y Arquitectura de Wazuh

Wazuh comenzó a evolucionar como una bifurcación de OSSEC, uno de los SIEM de código abierto más populares. Y ahora esta es su propia solución única con nueva funcionalidad, corrección de errores y arquitectura optimizada.

OSSEC es un sistema de detección de intrusos (HIDS) de código abierto y gratuito. Es un software que se adapta a las necesidades de seguridad. Su tarea es analizar los registros de eventos del sistema operativo, verificar los registros de dispositivos Windows, verificar su integridad, detectar alertas y rootkits en tiempo real, así como respuesta proactiva a los ataques.

Wazuh se utiliza para la prevención, detección y respuesta de amenazas. Protege las cargas de trabajo en entornos locales, virtualizados, en contenedores y basados en la nube y es ampliamente utilizado por miles de organizaciones en todo el mundo, desde pequeñas empresas hasta grandes empresas.

Wazuh está estructurado en 3 partes:

1. Un **agente** implementado en los sistemas monitorizados (portátiles, máquinas virtuales...), instalado en puntos finales como computadoras portátiles, computadoras de escritorio, servidores, instancias en la nube o máquinas virtuales, proporciona capacidades de prevención, detección y respuesta. Es compatible con las plataformas Windows, Linux, MacOS, HP-UX, Solaris y AIX.
2. Un **servidor** de administración que analiza los datos recibidos por los agentes, los cuales son procesados a través de decodificadores y reglas. Usa la inteligencia de amenazas para buscar indicadores de compromiso (IOC) conocidos. Un solo servidor puede analizar datos de cientos o miles de agentes y escalar horizontalmente cuando se configura como un clúster.
3. **Elastic Stack** : Elastic Stack es un conjunto de software formado por Filebeat, Elasticsearch y Kibana. Indexa y almacena las alertas generadas por el servidor de Wazuh. Además, la integración entre Wazuh y Kibana proporciona una potente interfaz de usuario para la visualización y el análisis de datos. Esta interfaz también se usa para administrar la configuración de Wazuh y monitorear su estado.

Además del monitoreo basado en agentes, la herramienta Wazuh también puede monitorear dispositivos sin ellos, como firewalls, conmutadores, enrutadores, IDS de red, etc. Por ejemplo, los datos de un registro del sistema se pueden recopilar a través de Syslog (protocolo de registro del sistema), y su configuración se puede monitorizar a través del sondeo periódico de sus datos (por ejemplo, a través de SSH o mediante una API).

La Imagen 4 muestra la arquitectura de Wazuh.

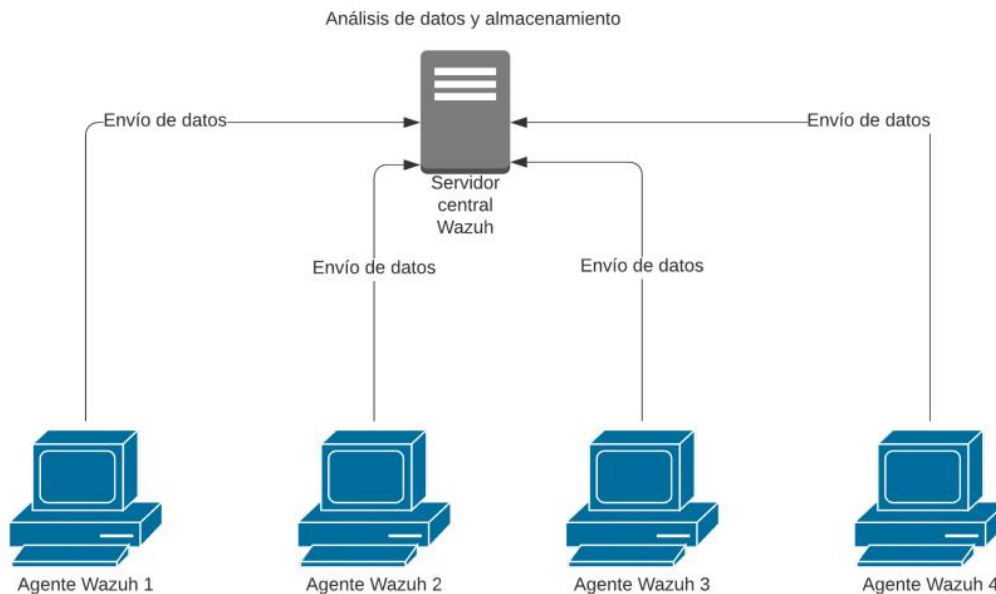


Imagen 4: Arquitectura de Wazuh

2.4 Características principales de Wazuh

Análisis de seguridad: Recopila, añade, indexa y analiza información, lo que ayuda a las clientes a detectar intrusos, amenazas y comportamientos inusuales.

Detección de intrusos: Se monitorea el sistema en busca de malware, rootkits, anomalías sospechosas, respuestas inconsistentes y archivos ocultos.

Análisis de los datos de logs: Wazuh está capacitado para leer los registros de diferentes sistemas operativos y aplicaciones, enviando estos a un administrador central para su análisis.

Monitorización de la integridad de los archivos: Monitoriza su sistema de archivos, en busca de cambios en el contenido, permisos, atributos de los archivos que necesitemos controlar, así como, identifica de forma propia los usuarios y aplicaciones utilizadas para crear o modificar cualquier archivo.

Escaneo de vulnerabilidades: Los agentes de Wazuh envían información a un servidor, donde se asocia con una base de datos denominada CVE (Common Vulnerabilities Exposure) que se mantiene actualizada, para identificar aplicaciones susceptibles de ser atacadas. Las evaluaciones de vulnerabilidad automatizadas lo ayudan a encontrar debilidades en sus activos críticos y tomar medidas correctivas antes de que los atacantes las utilicen en detrimento a fin de robar datos confidenciales.

Evaluación de la configuración: Supervisa los ajustes de configuración del sistema y de las aplicaciones para garantizar que cumplan con sus políticas de seguridad. Los agentes escanean periódicamente aplicaciones conocidas como vulnerables, no parcheadas o que están configuradas para ser inseguras.

Respuesta a incidentes: Proporciona varias respuestas proactivas “listas para usar”, que activan diversas contramedidas que gestionan amenazas activas. Así mismo, Wazuh puede llevar a cabo la ejecución de comandos de manera remota o consultas al sistema, identificando indicadores de compromiso (IOCs) y asistiendo con otras labores forenses o de respuesta automatizada a incidentes.

Cumplimiento normativo: Se proporcionan diversos controles de seguridad que son necesarios para obedecer con los estándares y regulaciones de la industria.

Seguridad en la nube: Wazuh proporciona ayuda para monitorizar infraestructuras en la nube, integrando módulos que pueden transmitir información desde proveedores como pueden ser Azure, Google Cloud o Amazon AWS.

Seguridad de los contenedores: La herramienta provee seguridad en los host Docker y contenedores, monitorizando su comportamiento y detectando amenazas y anomalías.

2.5 Resumen Wazuh vs OpenVas

Con la explicación de cada una de las herramientas descritas se puede tener una idea generalizada de las diferencias, adjuntando a modo de resumen los pros y contras entre OSSIM (engloba a OpenVas) y Wazuh.

OSSIM

Ventajas:

- Basado en proyectos probados de código abierto.
- Gran comunidad de usuarios y desarrolladores.

Desventajas

- Las plataformas en la nube como Amazon AWS y Azure no son compatibles.
- No se pueden gestionar logs, ni automatizar, ni visualizar, ni integrar nada con otras herramientas.
- Problemas de escalabilidad.
- Arquitectura de servidor único.

Wazuh

Ventajas:

- Basado en OSSEC.
- Admite implementaciones Docker, Puppet, Chef y Ansible.
- Estándares de cumplimiento normativo como PCI DSS
- Conjunto de normas para detectar algunos ataques usuales.
- Admite el monitoreo de plataformas en la nube (AWS y Azure).
- Escaneo de vulnerabilidades.
- Se integra con Splunk para visualizar alertas y datos de API.

- Excelente documentación online actualizada.

Desventajas

- Requiere una implementación completa de ELK Stack además de los componentes del servidor Wazuh.

Capítulo 3. Entorno de pruebas

3.1 Instalación VirtualBox

Para poder estudiar Wazuh y OpenVas se ha utilizado la máquina virtual de VirtualVox, con el objetivo de crear una red local en la que instalar diferentes equipos con sus correspondientes sistemas operativos, que comentaremos más adelante.

El equipo de trabajo utilizado para montar la red virtual tiene estas especificaciones:

- Windows 10 Home 64 bits
- Intel Core I7-4510U 2GHz
- 12GB de RAM

Se ha instalado la versión 6.1.32 de VirtualBox (Imagen 5). El proceso de instalación es muy sencillo.

VirtualBox 6.1.32 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

Imagen 5: Paquete de instalación de VirtualBox

3.2 Instalación de los equipos virtuales

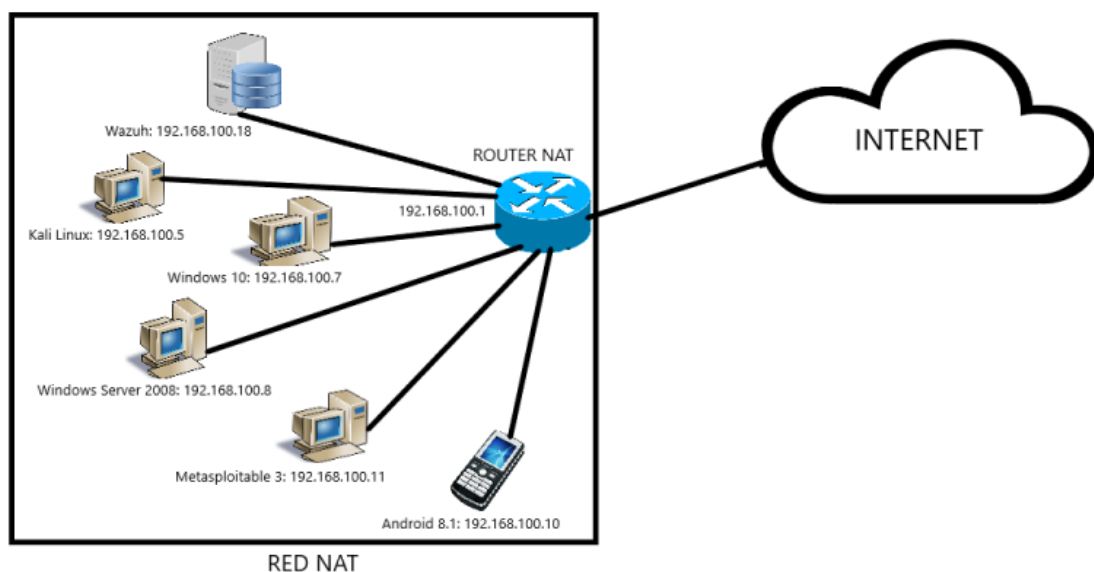


Imagen 6: Topología de la Red Virtual

3.2.1 Metasploitable 3

Metasploitable3 es una máquina virtual preconfigurada basada en Ubuntu 14.04 y en Windows Server 2008 (en nuestro caso utilizaremos únicamente la versión de Ubuntu), que cuenta con múltiples vulnerabilidades por defecto que nos van a ayudar a tener una perspectiva más amplia de lo que puede suceder en un equipo. La gran ventaja que tiene es que podemos explotar la máquina sin comprometer nuestro equipo personal.

Para hacer la instalación es necesario utilizar PowerShell, debido a que disponemos solo de unos scripts.

1. Instalamos Vagrant: Herramienta que permite crear entornos de desarrollo virtualizados. Crea y configura máquinas virtuales a partir de ficheros de configuración
2. Instalamos Packet: Herramienta para la creación de imágenes para múltiples sistemas.

Para que Vagrant funcione correctamente es necesario instalar un plugin llamado reload con el comando **vagrant plugin install vagrant-reload** (Imagen 7).

```
PS C:\Users\barka> vagrant plugin install vagrant-reload
Installing the 'vagrant-reload' plugin. This can take a few minutes...
Fetching vagrant-reload-0.0.1.gem
Installed the plugin 'vagrant-reload (0.0.1)'
```

Imagen 7: Instalación del plugin de Vagrant

A continuación, accedemos al directorio descargas y creamos la carpeta metasploitable3-workspace, y accedemos a ella (Imagen 8).

```
PS C:\Users\barka> cd Downloads
PS C:\Users\barka\Downloads> mkdir metasploitable3-workspace

Directorio: C:\Users\barka\Downloads

Mode                LastWriteTime         Length Name
----                -
d-----          12/09/2021   13:15             metasploitable3-workspace
```

Imagen 8: Creación de la carpeta metasploitable3-workspace

Por último, descargamos el archivo metasploitable3 con el comando **Invoke-webRequest -Uri <https://raw.githubusercontent.com/rapid7/metasploitable3/master/Vagrantfile> -OutFile "Vagrantfile"** y hacemos la instalación **con vagrant up**. Esto iniciara la máquina virtual y ejecutara todos los scripts tanto de instalación como de configuración (Imagen 9).

```

PS C:\Users\barka\Downloads\metasploitable3-workspace> Invoke-WebRequest -Uri "https://raw.githubusercontent.com/rapid7/metasploitable3/master/Vagrantfile" -OutFile "Vagrantfile"
PS C:\Users\barka\Downloads\metasploitable3-workspace> vagrant up
Bringing machine 'ub1404' up with 'virtualbox' provider...
Bringing machine 'win2k8' up with 'virtualbox' provider...
==> ub1404: Box 'rapid7/metasploitable3-ub1404' could not be found. Attempting to find and install.
..
  ub1404: Box Provider: virtualbox
  ub1404: Box Version: >= 0
==> ub1404: Loading metadata for box 'rapid7/metasploitable3-ub1404'
  ub1404: URL: https://vagrantcloud.com/rapid7/metasploitable3-ub1404
==> ub1404: Adding box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for provider: virtualbox
  ub1404: Downloading: https://vagrantcloud.com/rapid7/boxes/metasploitable3-ub1404/versions/0.1.12-weekly/providers/virtualbox.box
  ub1404:
==> ub1404: Successfully added box 'rapid7/metasploitable3-ub1404' (v0.1.12-weekly) for 'virtualbox'

```

Imagen 9: Descarga del archivo metasploitable3

Una vez finalizado el proceso de instalación ya tendremos la máquina preparada en VirtualBox para trabajar con ella. Para iniciar sesión nos pedirá unas credenciales predeterminadas:

Nombre de usuario: vagrant

Contraseña: vagrant

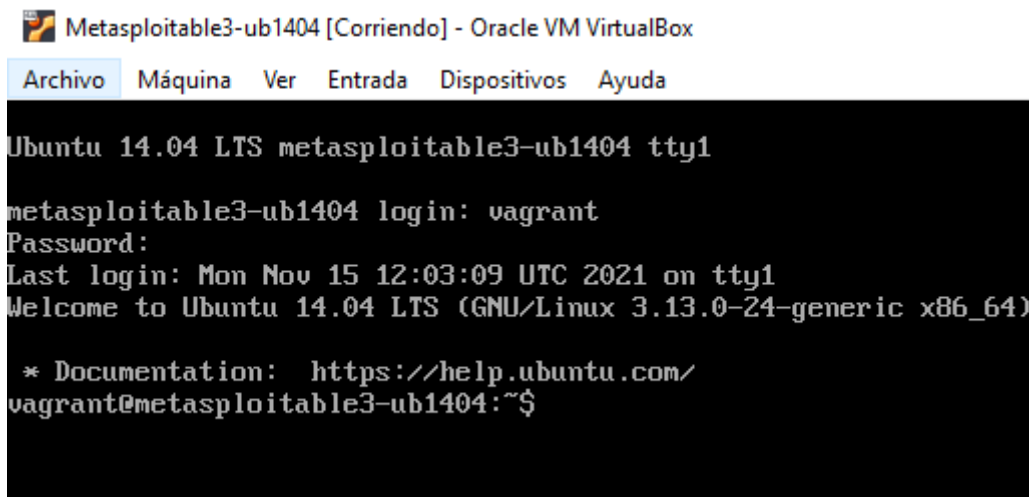


Imagen 10: Interfaz de comandos metasploitable3

3.2.2 Kali Linux

Kali Linux es una distribución de Linux basada en Debian preparada para una gran variedad de temas de seguridad, como ataques inalámbricos, análisis de red, recopilación de información, *sniffing* (herramienta maliciosa para analizar y monitorizar paquetes en la red) y *spoofing* (técnica de hacking para suplantar identidades o personas en la red), hacking de hardware... Es una de las distribuciones de seguridad más usadas, ya que vienen preinstaladas múltiples herramientas de análisis y da soporte a instalar otras externas que son las que utilizaremos.

En nuestra red virtual la maquina principal será Kali Linux, es decir, todos los resultados de nuestro estudio se harán ahí. La versión instalada es la 2021.4a (Imagen 11).

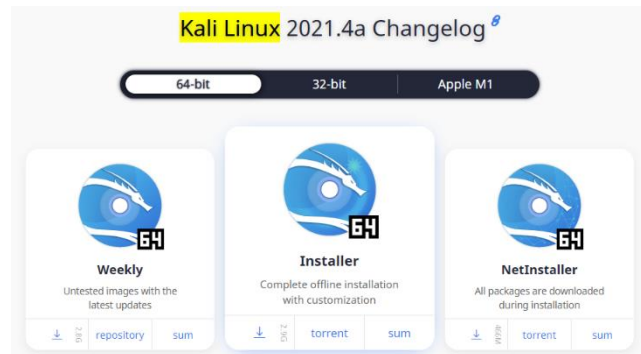


Imagen 11: Paquete de instalación de Kali Linux

El proceso de instalación es automático. Todo viene ya preconfigurado para hacer el montaje en VirtualBox. Nos pedirá un usuario y contraseña que estableceremos nosotros.

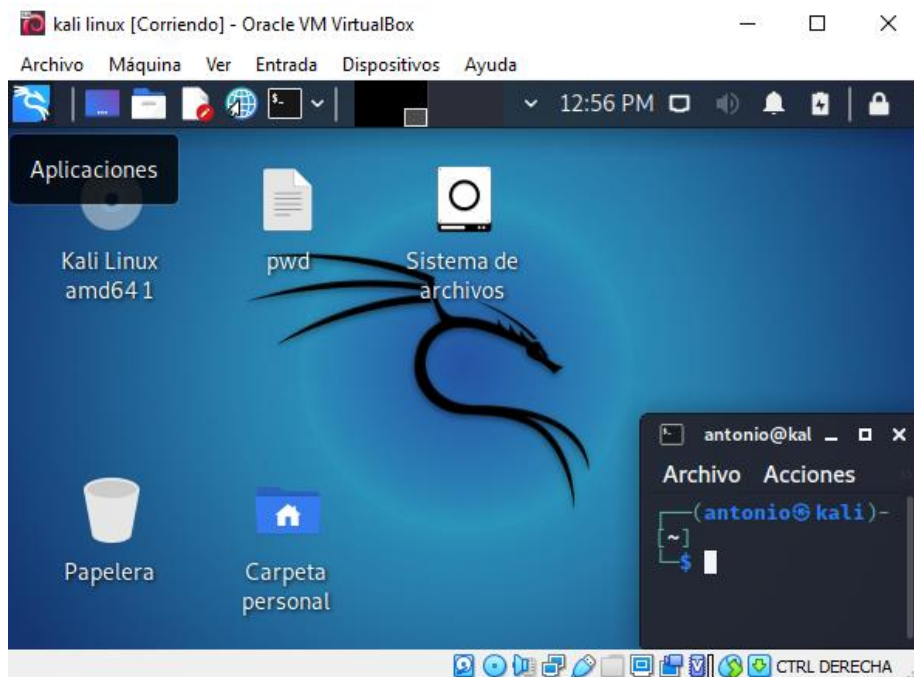


Imagen 12: Interfaz de Kali Linux

3.2.3 Windows 10, Android 8.1 y Windows Server 2008

Para completar nuestro laboratorio y tener más opciones de estudio, se han instalado 3 máquinas más con Windows 10, Android 8.1 y Windows Server 2008. Se irán utilizando o no en función de la compatibilidad de las herramientas de OpenVas y Wazuh.

Tras la instalación de todas las máquinas nos queda la interfaz de VirtualBox así (Imagen 13).

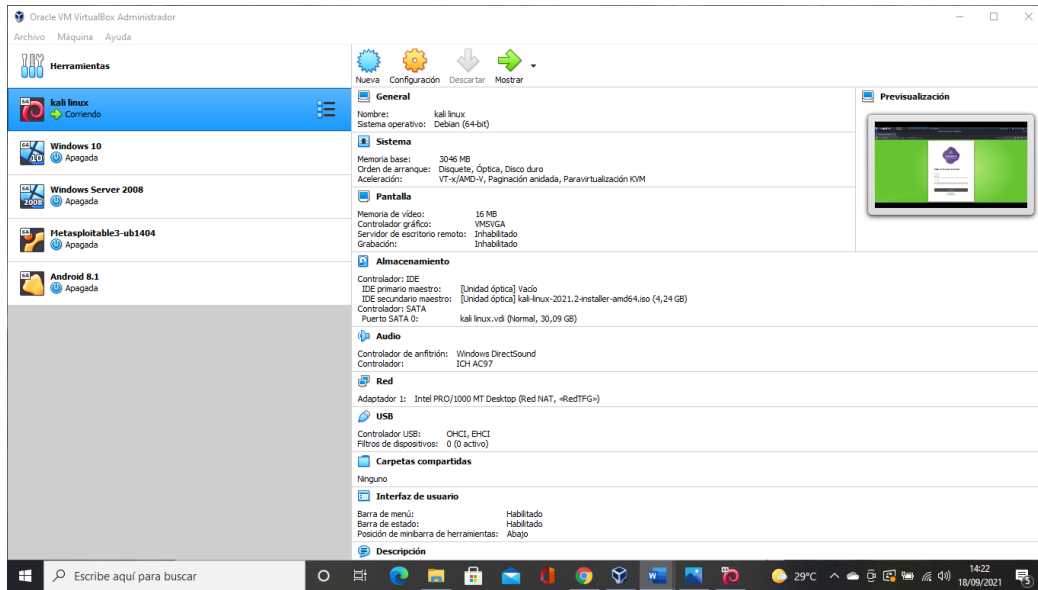


Imagen 13: Interfaz de VirtualBox con todas las maquinas instaladas

3.2.4 Configuración del entorno de pruebas

VirtualBox nos ofrece la opción de crear distintos tipos de redes. En nuestro caso para simular el propio router de nuestra casa, crearemos una **Red Nat**. Todas las máquinas estarán conectadas entre sí y tendrán acceso a internet.

Para activar este tipo de conexión, lo primero que tendremos que hacer es irnos a la ventana principal de VirtualBox, seleccionar en la parte superior “**Archivo**” y clicar sobre “**Preferencias**” y en el apartado “**Red**”, pulsamos en el icono situado a la derecha para crear la nueva red (Imagen 14).

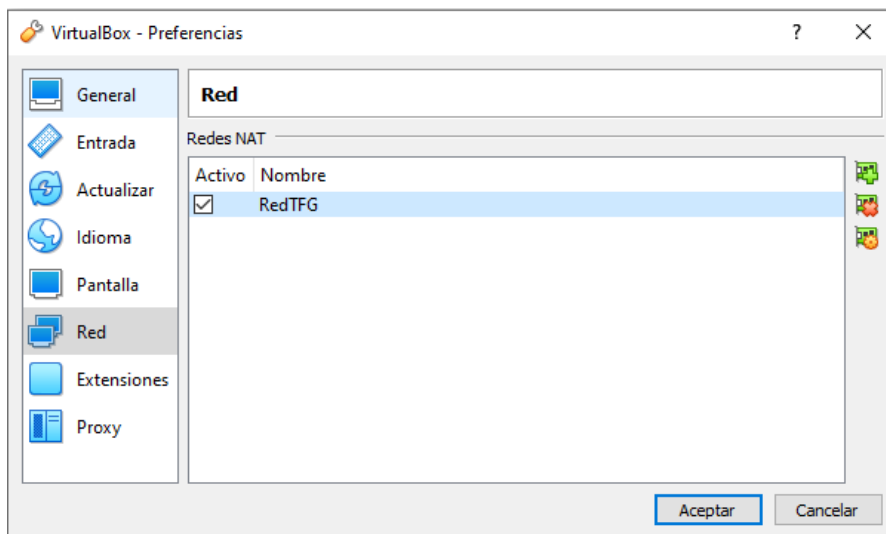


Imagen 14: Proceso de configuración de red (1)

Sobre el nombre de la red, hacemos doble clic para editarla. Por defecto VirtualBox asigna IPs del tipo 10.0.2.x/24. En nuestro caso se ha elegido la IP **192.168.100.0/24** para la red CIDR (enrutamiento entre dominios sin clases), y se ha seleccionado el soporte DHCP.

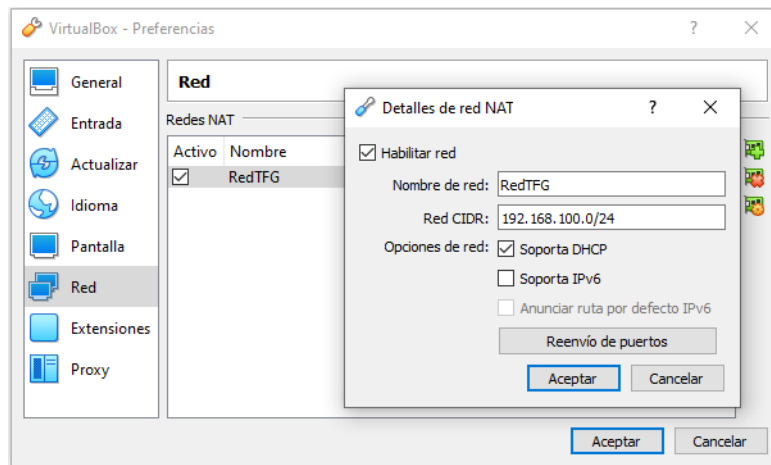


Imagen 15: Proceso de configuración de red (2)

Para terminar la configuración, nos dirigiremos de manera individual a las máquinas virtuales, y les diremos que se conecten mediante uno de los adaptadores a la **RedTFG** que se he creado previamente.

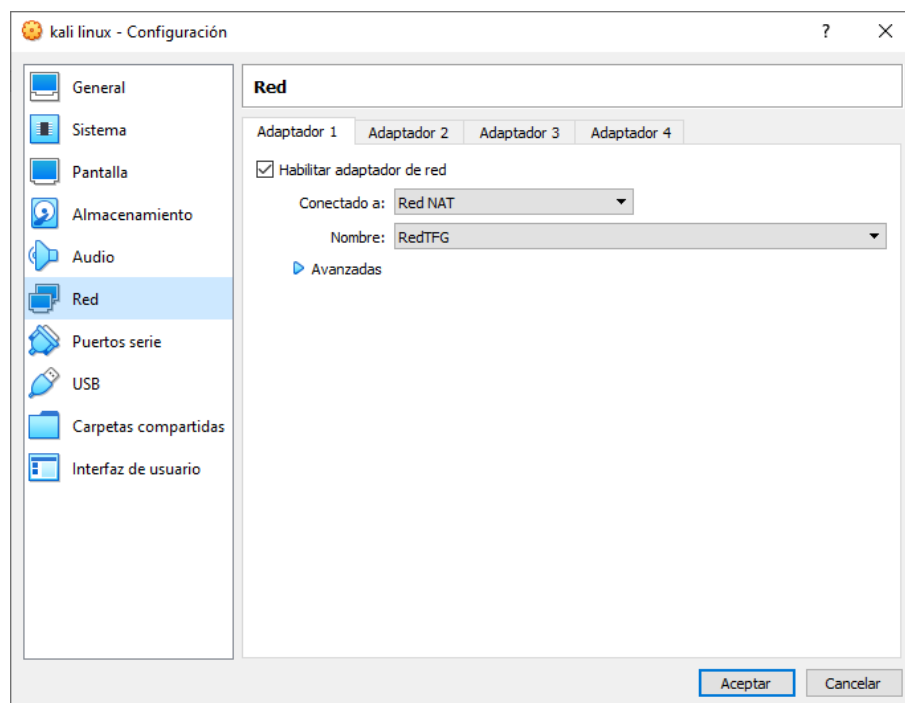


Imagen 16: Proceso de configuración de red (3)

Automáticamente cada equipo coge las siguientes IPs:

Kali Linux - 192.168.100.5

Windows 10 - 192.168.100.7

Windows Server 2008 - 192.168.100.8

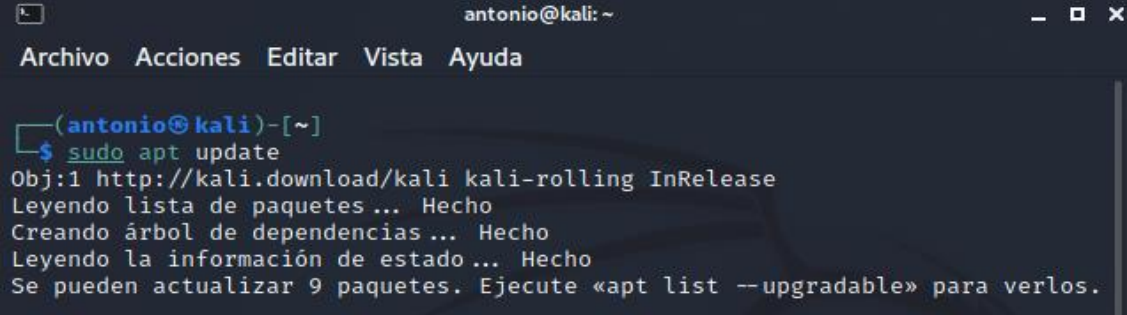
Metasploitable 3 ubuntu14.04 - 192.168.100.11

Android 8.1 - 192.168.100.10

3.3 Instalación y configuración de OpenVas

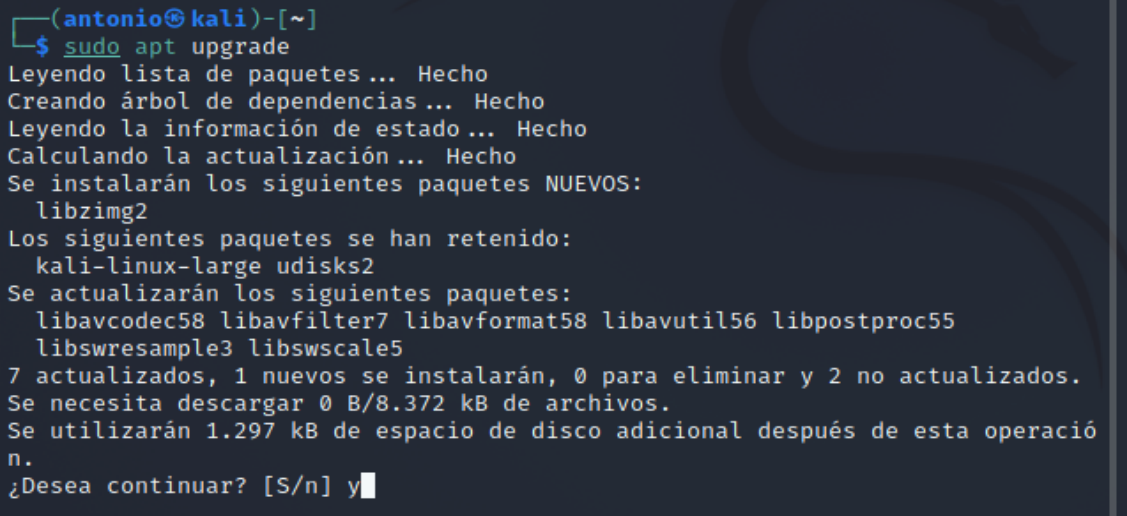
Para proceder a la instalación de OpenVas tenemos que iniciar Kali Linux y acceder al terminal de comandos.

Actualizamos los paquetes del sistema y validamos las actualizaciones mediante los comandos **sudo apt update** y **sudo apt upgrade**.



```
antonio@kali: ~  
Archivo Acciones Editar Vista Ayuda  
  
antonio@kali)~  
$ sudo apt update  
Obj:1 http://kali.download/kali kali-rolling InRelease  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Se pueden actualizar 9 paquetes. Ejecute «apt list --upgradable» para verlos.
```

Imagen 17: Actualización de los paquetes del sistema



```
antonio@kali)~  
$ sudo apt upgrade  
Leyendo lista de paquetes ... Hecho  
Creando árbol de dependencias ... Hecho  
Leyendo la información de estado ... Hecho  
Calculando la actualización ... Hecho  
Se instalarán los siguientes paquetes NUEVOS:  
  libzim2  
Los siguientes paquetes se han retenido:  
  kali-linux-large udisks2  
Se actualizarán los siguientes paquetes:  
  libavcodec58 libavfilter7 libavformat58 libavutil56 libpostproc55  
  libswresample3 libswscale5  
7 actualizados, 1 nuevos se instalarán, 0 para eliminar y 2 no actualizados.  
Se necesita descargar 0 B/8.372 kB de archivos.  
Se utilizarán 1.297 kB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [S/n] y
```

Imagen 18: Validación de las actualizaciones del sistema

Una vez tengamos todo actualizado, iniciamos la instalación de OpenVas con el comando **sudo apt-get install openvas**.

```
(antonio@kali)-[~]
└─$ sudo apt-get install openvas
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
 openvas
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 5.072 B de archivos.
Se utilizarán 12,3 kB de espacio de disco adicional después de esta operación
.
Des:1 http://kali.download/kali kali-rolling/main amd64 openvas all 21.4.2.0
[5.072 B]
Descargados 5.072 B en 1s (8.661 B/s)
Seleccionando el paquete openvas previamente no seleccionado.
(Leyendo la base de datos ... 350289 ficheros o directorios instalados actual
mente.)
Preparando para desempaquetar ... /openvas_21.4.2.0_all.deb ...
Desempaquetando openvas (21.4.2.0) ...
Configurando openvas (21.4.2.0) ...
```

Imagen 19: Validación de las actualizaciones del sistema

A continuación, iniciamos el proceso de configuración con el comando **sudo gvm-setup**. Este proceso tardará unos minutos y nos proporcionará una contraseña para poder acceder a OpenVas.

```
(antonio@kali)-[~]
└─$ sudo gvm-setup
[>] Starting PostgreSQL service
[>] Creating GVM's certificate files
[>] Creating PostgreSQL database
[*] Creating database user
[*] Creating database
[*] Creating permissions
CREATE ROLE
```

Imagen 20: Proceso de obtención de la contraseña de acceso a OpenVas

```
[+] Done
[*] Please note the password for the admin user
[*] User created with password 'db8ffef8-549c-4099-a5a2-2dc7d03bd525'.

[>] You can now run gvm-check-setup to make sure everything is correctly conf
igured
```

Imagen 21: Contraseña de acceso a OpenVas

Tras completar el proceso anterior, ya podemos iniciar OpenVas mediante el comando **sudo gvm-start**. Es recomendable comprobar que la instalación se ha realizado correctamente con el comando **sudo gvm-check-setup**.

```
(antonio@kali)-[~]
└─$ sudo gvm-start
[sudo] password for antonio:
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>] Web UI (Greenbone Security Assistant): https://127.0.0.1:9392

● greenbone-security-assistant.service - Greenbone Security Assistant (gsad)
   Loaded: loaded (/lib/systemd/system/greenbone-security-assistant.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2021-09-12 00:17:25 CEST; 154ms ago
     Docs: man:gsad(8)
           https://www.greenbone.net
   Process: 115580 ExecStart=/usr/sbin/gsad --listen=127.0.0.1 --port=9392 (code=exited, status=0/SUCCESS)
    Main PID: 115581 (gsad)
       Tasks: 3 (limit: 3471)
      Memory: 2.4M
```

Imagen 22: Inicio de OpenVas mediante comandos

Semanalmente hay que actualizar las bases de datos de los NVTs (*Network Vulnerability Tests*) mediante el comando **sudo gvm-feed-update**. Si no se van actualizando periódicamente pueden surgir muchos falsos positivos en el proceso de análisis de vulnerabilidades.

Al finalizar este proceso automáticamente Kali Linux abrirá su navegador con la url <https://127.0.0.1.9392>. Se mostrará la interfaz de administración con su usuario y contraseña e iniciaremos sesión.

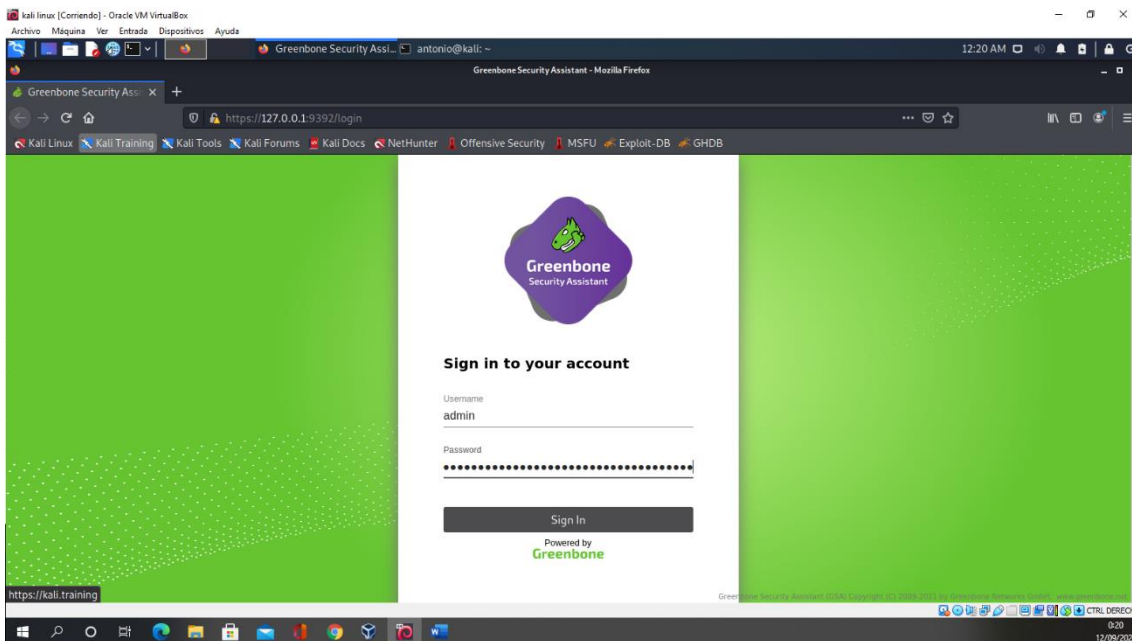


Imagen 23: Interfaz de acceso con credenciales a OpenVas

A continuación, se desplegará Greenbone Security Assistant que es la interfaz web de OpenVas.

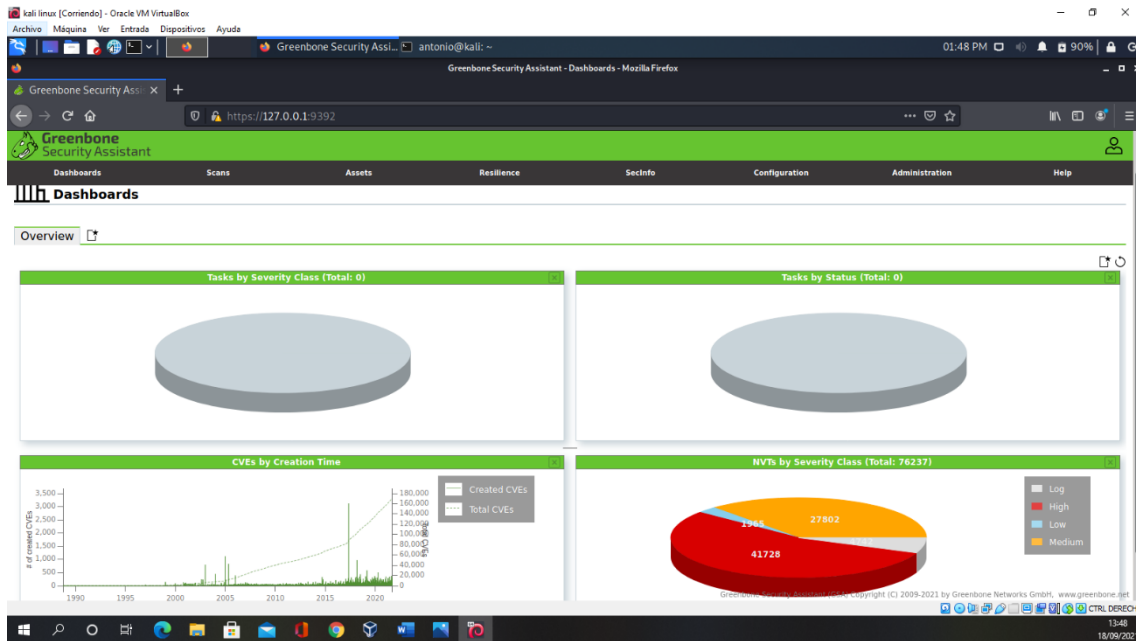


Imagen 24: Interfaz de Greenbone (OpenVas)

El proceso de instalación ha concluido, por lo que ya podemos hacer un escaneo de cada una de las máquinas. Vamos a ir mostrando el proceso de configuración y explicando el significado de varios términos que nos irán saliendo.

Los pasos a seguir para analizar cualquiera de las maquinas son los siguientes:

1. Nos dirigimos a la pestaña de configuración y damos clic en “**targets**”.

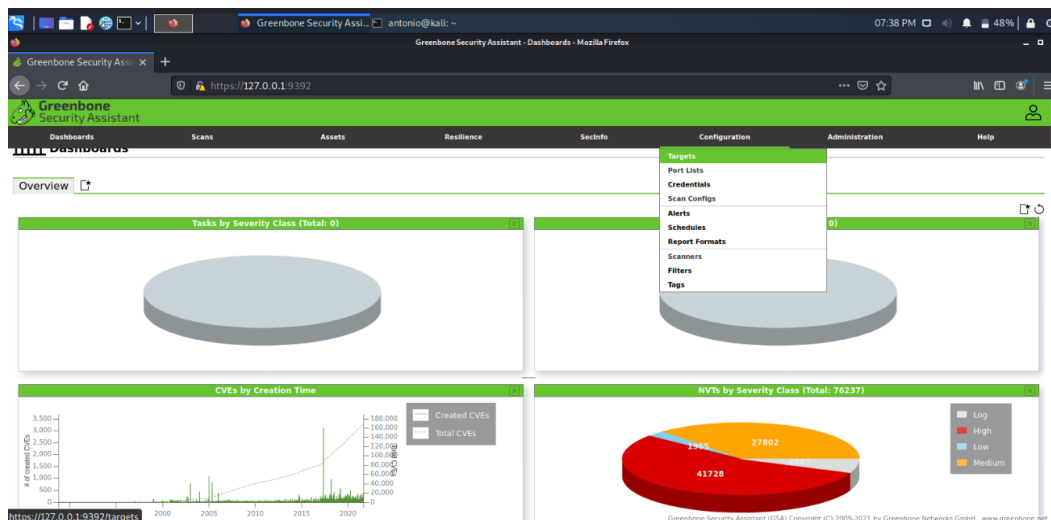


Imagen 25: Configuración de escaneo de vulnerabilidades (1)

2. A continuación, hacemos clic arriba a la izquierda en la hoja con una estrella y aparecerá un desplegable (Ver flecha roja)

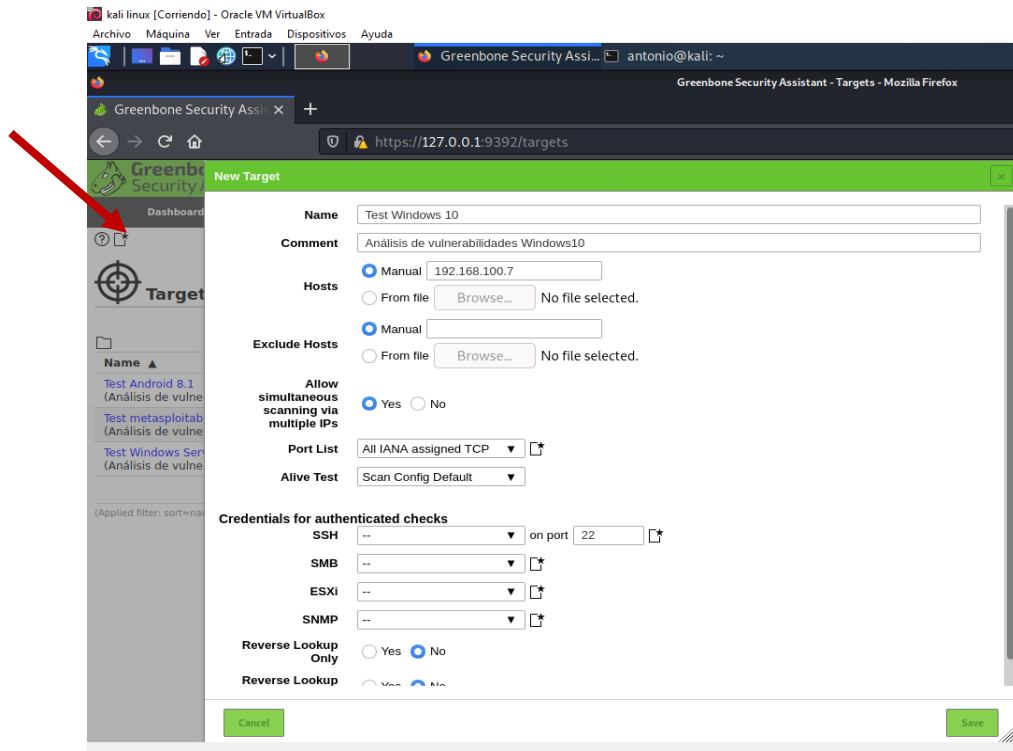


Imagen 26: Configuración de escaneo de vulnerabilidades (2)

3. En nuestro desplegable ponemos el nombre y un comentario para identificarlo de escaneos que hagamos en un futuro, con la máquina que estamos preparando. En este caso se está configurando la máquina de Windows 10 cuyos campos rellenos serán los mismos para las demás máquinas. En el campo host elegimos la IP o un rango de IPs del equipo que queremos escanear. En el apartado “**Port List**” elegiremos el tipo de puertos a analizar y el “**Alive Test**” lo dejamos tal cual viene. Podemos agregar también las credenciales de nuestro sistema para permitir que OpenVas verifique las vulnerabilidades locales (en nuestro caso no lo haremos).
4. Guardamos la configuración y nos vamos a la pestaña “**Scans**” y seleccionamos “**Tasks**”.

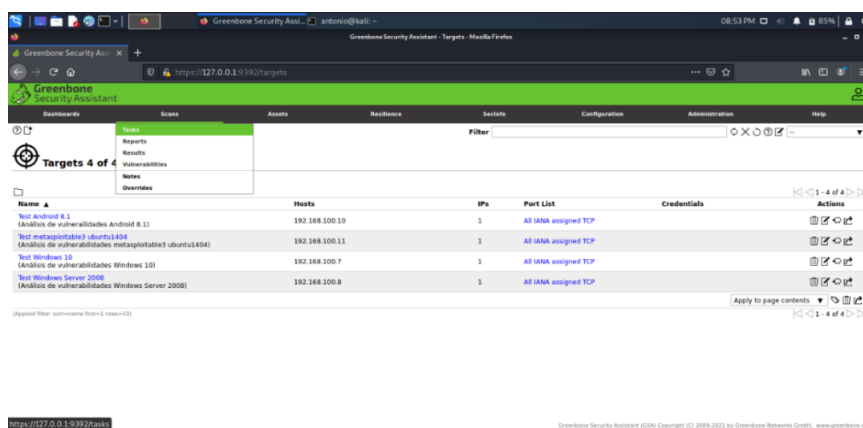


Imagen 27: Configuración de escaneo de vulnerabilidades (3)

5. En la pantalla siguiente pantalla veremos de nuevo la hoja con la estrella por lo que volvemos a hacer clic en ella y seleccionando “New tasks”. Nos aparecerá otro desplegable con los siguientes campos:

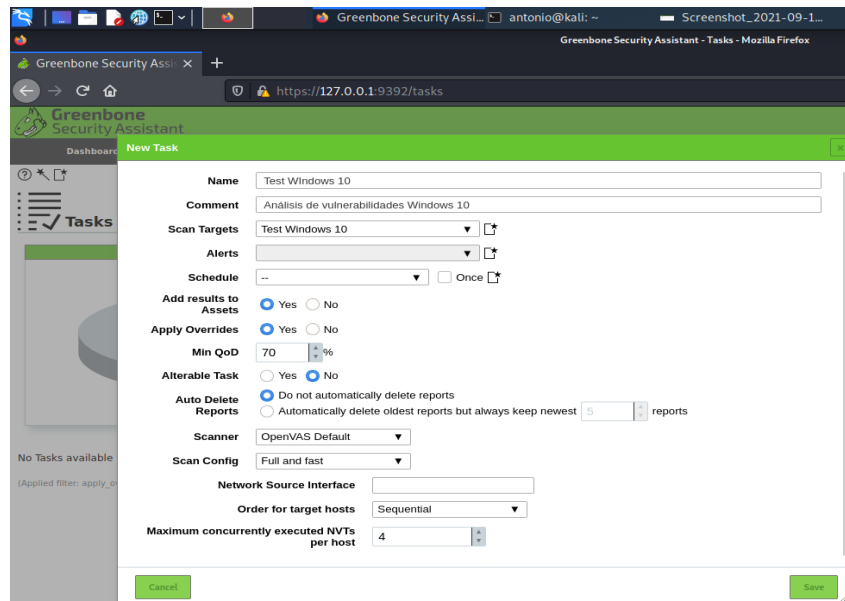


Imagen 28: Configuración de escaneo de vulnerabilidades (3)

- **Scan Targets:** seleccionaremos el sistema que queremos escanear, el cual hemos creado previamente en el apartado targets.
- **Alerts:** envía notificaciones bajo ciertas condiciones determinadas.
- **Schedule:** podemos determinar un horario preconfigurado para el escaneo.
- **Apply Overrides:** con esta opción activada se consigue modificar la gravedad de un resultado. Es importante para gestionar las vulnerabilidades que se han encontrado como falsos positivos, las cuales se les proporciona un nivel de gravedad que no deberían tener.
- **Min QoD:** «Mínima calidad de detección», OpenVas aconseja dejar por defecto la opción en un 70% para así evitar amenazas no reales.
- **Alterable Task:** permite modificar la tarea incluso después de haber generado un informe.
- **Auto Delete Reports:** nos permite eliminar o sobrescribir informes antiguos. Se puede seleccionar la cantidad de informes guardados por tarea.
- **Scanner:** la herramienta nos ofrece dos tipos de escáneres a elegir, los cuales son el de OpenVas predeterminado (el que aconseja el manual) y el CVE. Se pueden crear otros tipos de escáneres manualmente aparte de estos.

- **Scan Config:** seleccionaremos uno de los siete tipos de escaneo que OpenVas nos proporciona. Se detectarán más o menos vulnerabilidades dependiendo la opción elegida. Un escaneo con la mejor opción puede tardar días en terminar.
 - **Network Source Interface:** podemos determinar el nombre de la interfaz de origen.
 - **Order for target hosts:** OpenVas nos da a elegir el orden en el que se procesarán los sistemas analizados (sequential, random o reverse).
 - **Maximum concurrently executed NVTs per host/ Maximum concurrently scanned hosts:** podemos establecer el máximo de NVTs ejecutados a la vez en un sistema, así como el número máximo de sistemas analizados simultáneamente. Hay que tener cuidado con establecer valores altos porque pueden surgir varios errores.
6. Guardamos la configuración y procedemos a iniciar el escaneo presionando el botón blanco de reproducción. Realizamos todo el proceso de configuración igual para las demás máquinas.

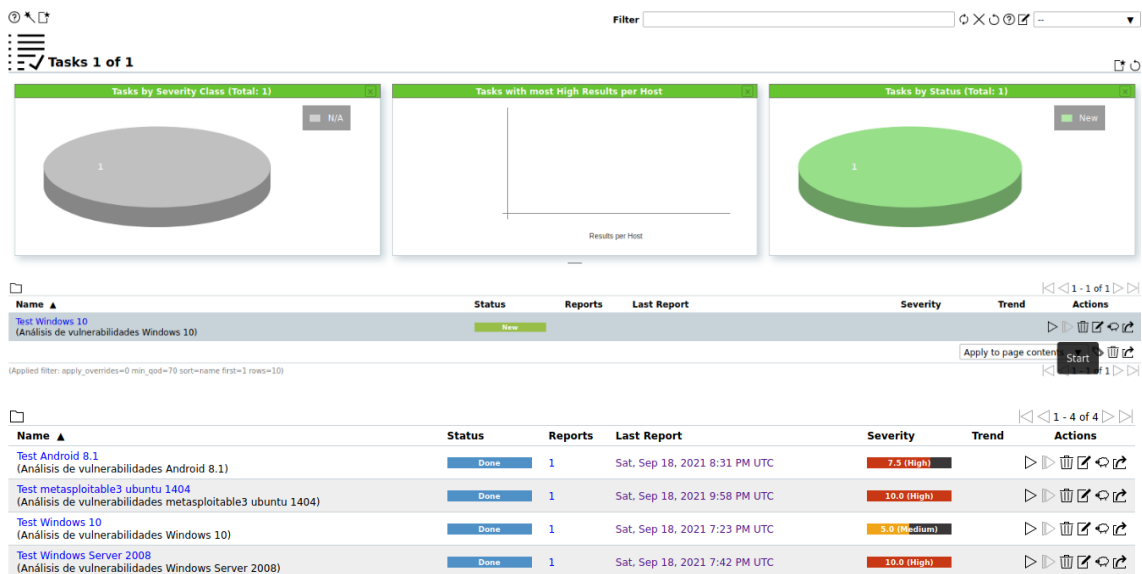


Imagen 29: Configuración de escaneo de vulnerabilidades (4)

Una vez concluido el escaneo nos vamos a “**Scan**” y hacemos clic en “**Report**” para observar la severidad de los resultados.



Imagen 30: Resultados de todas las maquinas escaneadas

Distinguimos lo grave que es la vulnerabilidad por lo siguiente:

Vulnerabilidades altas y medias High Medium. Estas deben abordarse con prioridad. Primero los altos y después los medios, aunque hay casos excepcionales en los que los hallazgos de nivel alto deben considerarse menores porque no se puede acceder al servicio a través del firewall.

Vulnerabilidades bajas y logs Low Log. Son interesantes para comprender los detalles. Estos resultados se filtran de manera predeterminada. Se requiere un conocimiento más profundo del host para su comprensión.

Falsos positivos False Pos.. Son hallazgos que describen un problema que realmente no existe.

Una vez conocemos la severidad de las vulnerabilidades hacemos clic en **done** Done para saber todos los detalles.

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Drupal Coder Remote Code Execution	10.0 (High)	95 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 11:01 PM UTC
ProFTPD `mod_copy` Unauthenticated Copying Of Files Via SITE CPFR/CPFO	10.0 (High)	99 %	192.168.100.11		21/tcp	Sat, Sep 18, 2021 10:53 PM UTC
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95 %	192.168.100.11		22/tcp	Sat, Sep 18, 2021 11:32 PM UTC
FTP Brute Force Logins Reporting	7.5 (High)	95 %	192.168.100.11		21/tcp	Sat, Sep 18, 2021 11:32 PM UTC
Test HTTP dangerous methods	7.5 (High)	99 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 11:30 PM UTC
Drupal Core SQL Injection Vulnerability	7.5 (High)	98 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 11:01 PM UTC
jQuery < 1.9.0 XSS Vulnerability	6.1 (Medium)	80 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 10:40 PM UTC
jQuery < 1.9.0 XSS Vulnerability	6.1 (Medium)	80 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 10:40 PM UTC

Imagen 31: Listado de vulnerabilidades (1)

Nos encontramos con la siguiente información:

Information: Información general sobre el escaneo correspondiente.

Task Name [Test metasploitable3 ubuntu 1404](#)
Comment [Análisis de vulnerabilidades metasploitable3 ubuntu 1404](#)
Scan Time Sat, Sep 18, 2021 9:59 PM UTC - Sat, Sep 18, 2021 11:33 PM UTC
Scan Duration 1:34 h
Scan Status Done
Hosts scanned 1
Filter [apply_overrides=0 levels=hml min_qod=70](#)
Timezone Coordinated Universal Time (UTC)

Imagen 32: Fecha y duración del escaneo

Results: Lista de todas las vulnerabilidades (severidad, calidad de detección (QoD), IP del host, en que puerto se localiza y fecha y hora de detección).

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Drupal Coder Remote Code Execution	10.0 (High)	95 %	192.168.100.11		80/tcp	Sat, Sep 18, 2021 11:01 PM UTC
ProFTPD 'mod_copy' Unauthenticated Copying Of Files Via SITE CPFR/CPTO	10.0 (High)	99 %	192.168.100.11		21/tcp	Sat, Sep 18, 2021 10:53 PM UTC
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95 %	192.168.100.11		22/tcp	Sat, Sep 18, 2021 11:32 PM UTC

Imagen 33: Listado de vulnerabilidades (2)

Si hacemos clic en una de las vulnerabilidades obtendremos información de como se ha detectado y una pequeña explicación de lo que sucede, así como el método de detección y como resolver la vulnerabilidad.

Summary

It was possible to login into the remote FTP server using weak/known credentials.

As the VT 'FTP Brute Force Logins' (OID: 1.3.6.1.4.1.25623.1.0.108717) might run into a timeout the actual reporting of this vulnerability takes place in this VT instead.

Detection Result

It was possible to login with the following credentials <User>:<Password>

vagrant:vagrant


Detection Method

Reports weak/known credentials detected by the VT 'FTP Brute Force Logins' (OID: 1.3.6.1.4.1.25623.1.0.108717).

Details: [FTP Brute Force Logins Reporting OID: 1.3.6.1.4.1.25623.1.0.108718](#)

Version used: 2021-01-21T10:06:42Z

Solution

Solution Type:  Mitigation

Change the password as soon as possible.

Imagen 34: Información sobre la vulnerabilidad

Hosts: Se muestran los sistemas operativos detectados, el número de vulnerabilidades encontradas y la gravedad más alta encontrada en el escaneo.



IP Address	Hostname	OS	Ports	Apps	Distance	Auth	Start	End	High	Medium	Low	Log	False Positive	Total	Severity ▼
192.168.100.11		 Ubuntu 14.04	4	17			Sat, Sep 18, 2021 10:00 PM UTC	Sat, Sep 18, 2021 11:33 PM UTC	6	12	2	0	0	20	10.0 (High)

Imagen 35: Información sobre el host

Ports: Se muestran los puertos escaneados, el número de hosts y la mayor gravedad encontrada por el escaneo.

Port	Hosts	Severity ▼
21/tcp	1	10.0 (High)
80/tcp	1	10.0 (High)
22/tcp	1	7.5 (High)
631/tcp	1	5.0 (Medium)

Imagen 36: Información sobre los puertos

Applications: Aplicaciones escaneadas con CPE (norma para especificar que versión de software ejecuta un sistema) de la aplicación, número de hosts, número de ocurrencias de resultados que detectaron este CPE y la mayor gravedad encontrada por el escaneo.

Application CPE	Hosts	Occurrences	Severity ▼
cpe:/a:proftpd:proftpd:1.3.5	1	1	10.0 (High)
cpe:/a:phpmyadmin:phpmyadmin:3.5.8	1	1	N/A

Imagen 37: Aplicaciones escaneadas con CPE

Operating Systems: Sistemas operativos analizados con el nombre del sistema, el nombre de host, la cantidad de hosts analizados y la gravedad más alta detectada por el análisis.

Operating System	CPE	Hosts	Severity ▼
 Ubuntu 14.04	cpe:/o:canonical:ubuntu_linux:14.04	1	10.0 (High)

Imagen 38: Sistema operativo escaneado

CVEs: CVE encontrados con el escaneo.

*CVE: Catálogo de vulnerabilidades, con una notación normalizada para identificarlas.

CVE	NVT	Hosts	Occurrences	Severity ▼
CVE-2015-3306	ProFTPD 'mod_copy' Unauthenticated Copying Of Files Via SITE CFR/CP/TO	1	1	10.0 (High)
CVE-2014-3704	Drupal Core SQL Injection Vulnerability	1	1	7.5 (High)
CVE-2012-6708	jQuery < 1.9.0 XSS Vulnerability	1	2	6.1 (Medium)
CVE-2016-2183 CVE-2016-6329 CVE-2020-12872	SSL/TLS: Report Vulnerable Cipher Suites for HTTPS	1	1	5.0 (Medium)
CVE-2011-3730	Drupal Information Disclosure Vulnerability	1	1	5.0 (Medium)

Imagen 39: CVEs

Closed CVEs: CVEs de vulnerabilidades detectadas originalmente que ya se confirmaron como resueltas durante el análisis.

TLS Certificates: Certificados TLS encontrados con el escaneo.

Issuer DN ▲	Serial	Activates	Expires	IP	Hostname	Port	Actions
CN=ubuntu	00B4195557D4D940BF	Thu, Oct 29, 2020 6:28 PM UTC	Sun, Oct 27, 2030 6:28 PM UTC	192.168.100.11		631	↓

Imagen 40: Certificados TLS

Error Messages: Mensajes de error que ocurrieron durante el escaneo.

Error Message ▲	Host	Hostname	NVT	Port
NVT timed out after 320 seconds.	192.168.100.11		Apache Struts Detection (HTTP)	general/tcp
NVT timed out after 900 seconds.	192.168.100.11		Backup File Scanner (HTTP)	general/tcp

Imagen 41: Mensajes de error

User Tags: Etiquetas asignadas.

En el siguiente capítulo se analizarán los resultados de los análisis de cada uno de los equipos en profundidad.

3.4 Instalación y configuración de Wazuh

Para instalar Wazuh se ha optado por una máquina preconstruida cuyo archivo `.ova` se ha descargado de su página oficial. Se ha importado directamente en VirtualBox.

En un principio se optó por una instalación manual pero debido a la complejidad y a la multitud de problemas que iban sucediendo se fue descartando.

La máquina virtual contiene lo suficiente para trabajar con ella:

- CentOS 7
- Administrador de Wazuh: 4.2.4
- Abrir distribución para Elasticsearch: 1.13.2
- Filebeat-OSS: 7.10.2
- Kibana: 7.10.2
- Complemento Wazuh Kibana: 4.2.5-7.10.2

La instalación y configuración de red es como las anteriores máquinas virtuales. Se le ha asignado la dirección IP 192.168.100.18, la cual tendremos que poner en el navegador de Kali Linux para iniciar Wazuh.

La instalación no es la aplicación de Wazuh, sino un servidor basado en CentOS 7, del cual recogeremos los datos. Al abrir la máquina virtual nos pedirá unas credenciales:

Usuario: Wazuh
Contraseña: Wazuh

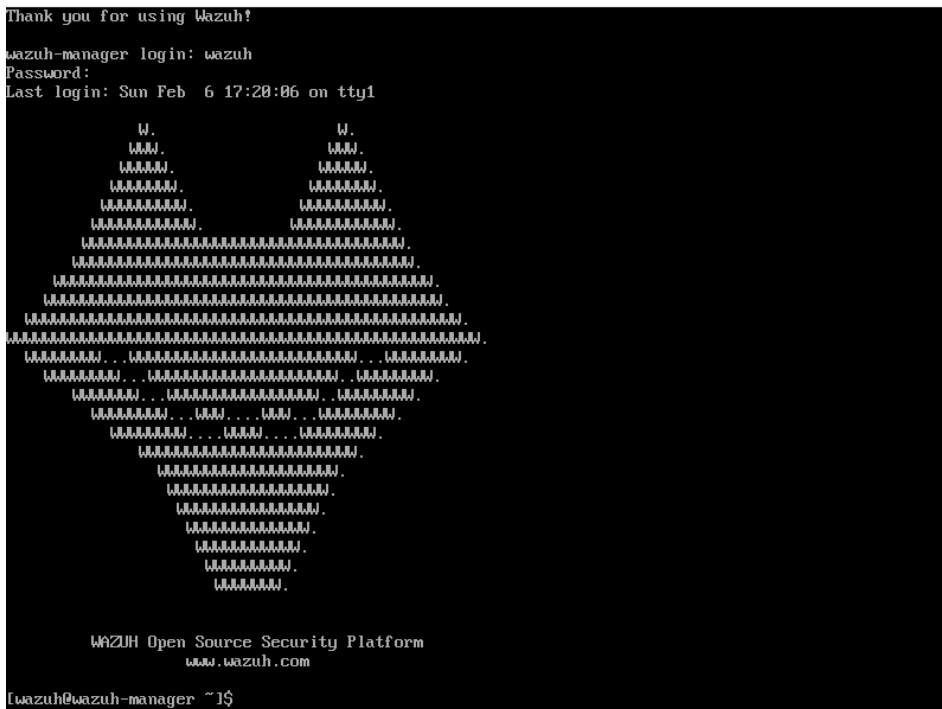


Imagen 42: Interfaz del servidor Wazuh

Accedemos a Kali Linux y en su navegador escribimos la url <https://192.168.100.18>. De esta manera se abrirá la interfaz y nos pedirá de nuevo unas credenciales que serán las mismas que las del servidor.

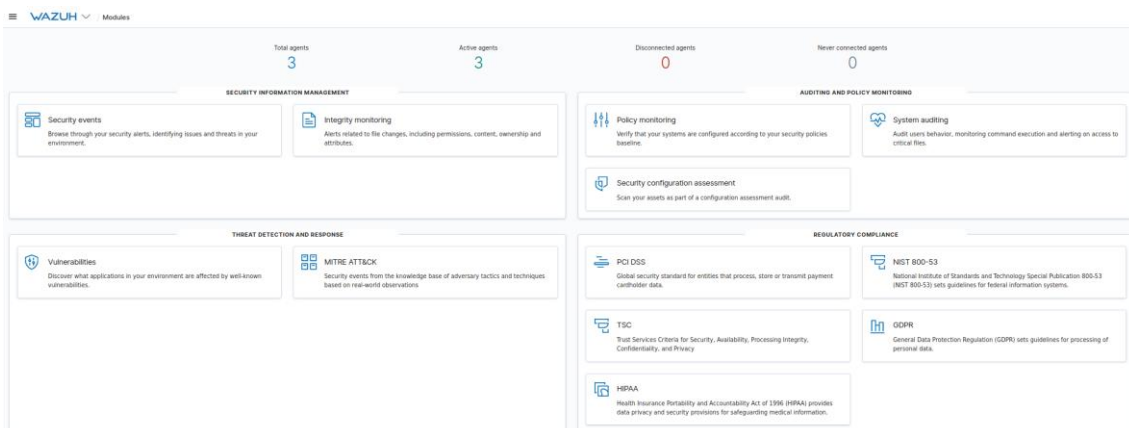


Imagen 43: Interfaz principal de Wazuh

3.4.1 Instalación de los agentes (Windows y Linux)

En el capítulo 2 se explicó que los agentes Wazuh solo eran compatibles con Linux, Windows, HP-UX, MacOS, AIX y Solaris. Por lo tanto, para esta herramienta se

utilizarán los equipos Windows 10, Windows Server 2008 y metasploitable3, todos ellos basados en Windows y Linux. Estos serán los equipos que monitorizaremos.

Linux

Accedemos a la interfaz de Wazuh y nos vamos al menú **Agents**, donde seleccionaremos la opción **Deploy New Agents**. Se abrirá un desplegable que rellenaremos con información del agente a registrar. Este nos proporcionará los comandos adecuados para el registro.

```
curl -so wazuh-agent-4.2.4.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.2.4-1_amd64.deb && sudo WAZUH_MANAGER='192.168.100.18' dpkg -i ./wazuh-agent-4.2.4.deb
```

Imagen 44: Instalación del agente Ubuntu

```
sudo update-rc.d wazuh-agent defaults 95 10  
sudo service wazuh-agent start
```

Imagen 45: Activación del agente Ubuntu

La ejecución de comandos se hará en la máquina que queremos registrar.

Windows

Ejecutaremos un instalador que proporciona Wazuh llamado **Wazuh Agent Manager**. En él tendremos que rellenar la dirección IP del servidor de Wazuh (192.168.100.18) y conseguir una contraseña, que se nos dará al realizar la siguiente configuración:

1. Abrimos una sesión en el administrador de PowerShell y accedemos con el comando a la ruta de instalación con el comando “**cd C:\Program Files (x86)\ossec-agent**”. Una vez ahí, registramos el agente mediante el comando “**.\agent-auth.exe -m 192.168.100.18**”

```
PS C:\Windows\system32> cd "C:\Program Files (x86)\"  
PS C:\Program Files (x86)> cd .\ossec-agent\  
PS C:\Program Files (x86)\ossec-agent> .\agent-auth.exe -m 192.168.100.18  
2022/02/08 01:35:07 agent-auth: INFO: Started (pid: 6508).
```

Imagen 46: Registro del Agente de Windows

2. Volvemos al instalador y le damos al botón “**Refresh**” proporcionándonos la contraseña. Por último, guardamos y reiniciamos el agente.

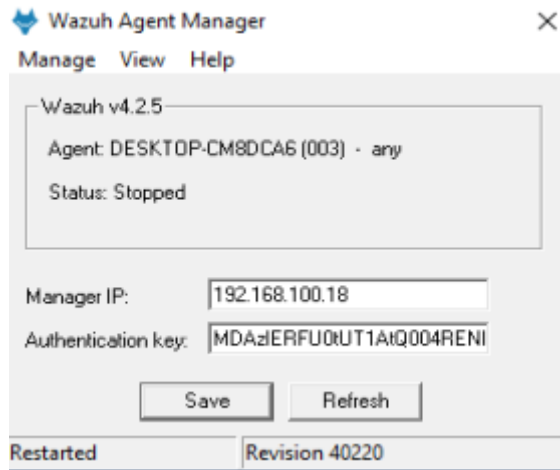


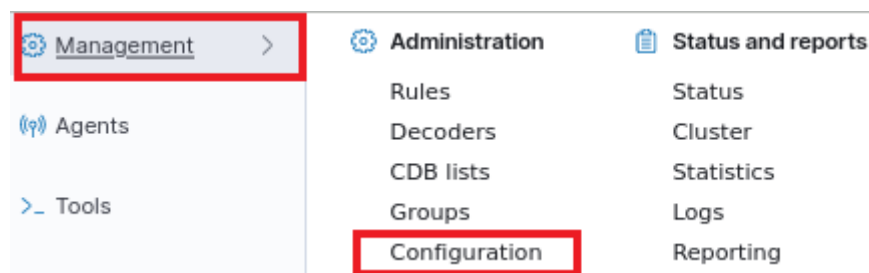
Imagen 47: Instalador de Wazuh

3.4.2 Configuración del detector de vulnerabilidades

Por defecto Wazuh no tiene activado el detector de vulnerabilidades por lo que hay que configurarlo manualmente.

Hay 2 formas de realizar la configuración:

1. Desde el servidor, acceder al fichero `/var/ossec/etc/ossec.conf` y modificar el archivo. Esta manera es un poco compleja, debido a que pide muchos permisos para la modificación de este y es muy poco intuitiva. Esta forma la descartamos.
2. Desde la propia aplicación de Wazuh accedemos a Management (Configuration).



Editamos el archivo XML `ossec.conf` para que funcione el detector de vulnerabilidades de la siguiente manera:

```
<vulnerability-detector>
  <enabled>yes</enabled>
  <interval>5m</interval>
  <ignore_time>6h</ignore_time>
  <run_on_start>yes</run_on_start>
```

El **<enabled>** hace referencia a la activación del detector, el **<interval>** es el tiempo transcurrido entre análisis, **<ignore_time>** es el tiempo durante el cual las vulnerabilidades que ya generaron una alerta no generaran otra y para que el análisis y las actualizaciones de la base de datos se ejecuten inmediatamente después del inicio de Wazuh activamos **<run_on_start>**.

En el propio archivo también editaremos los códigos que hacen referencia a los proveedores que utilizaremos:

```
<!-- Ubuntu OS vulnerabilities -->
<provider name="canonical">
  <enabled>yes</enabled>
  <os>trusty</os>
  <os>xenial</os>
  <os>bionic</os>
  <os>focal</os>
  <update_interval>1h</update_interval>
</provider>
```

```
<!-- Aggregate vulnerabilities -->
<provider name="nvd">
  <enabled>yes</enabled>
  <update_from_year>2008</update_from_year>
  <update_interval>1h</update_interval>
</provider>
```

Hemos activamos el proveedor canonical que hace referencia a Ubuntu. Para ahorrar espacio en el disco duro y evitar descargas, en la configuración final se ha borrado el código que contiene los nombres en clave de las distintas versiones de Ubuntu (xenial, bionic y focal).

La configuración **<update_from_year>** que hace referencia al año a partir del cual se van a descargar las vulnerabilidades indexadas, se ha establecido en el año 2008 en adelante.

Una vez editado el archivo **ossec.conf**, reiniciamos el servicio del administrador de Wazuh.

Para comprobar que el escaneo está en marcha accedemos al archivo **ossec.log** desde el menú Management (Logs). Ahí observaremos la actualización de la base de datos de cada proveedor, el punto en el que la base de datos ha sido actualizada y final del escaneo.

Capítulo 4. Resultados y comparación entre OpenVas y Wazuh

4.1 Resultados del estudio de OpenVas

A la hora de escanear cualquier equipo, OpenVas nos da la opción de elegir el tipo de escaneo, que dependiendo de cual se escoja detectara más o menos vulnerabilidades. En la imagen 28 (capítulo 3), en la opción “**Scan config**” distinguimos 7 tipos de escaneo preconfigurados:

Discovery: No detecta vulnerabilidades, simplemente proporciona información general del sistema a analizar.

Host Discovery: No detecta vulnerabilidades. Solo informa de la lista de sistemas descubiertos.

System Discovery: Solo se utilizan NVT que descubren los sistemas a analizar, incluidos los sistemas operativos instalados y el hardware en uso.

Full and Fast: Utiliza casi todos los NVT, exceptuando los que pueden dañar el sistema. Las NVT están optimizadas de manera que la tasa potencial de los falsos positivos sea muy baja. El escaneo es rápido.

Full and Fast Ultimate (Opción de pago*): Similar a Full and Fast pero utilizando NVTs que podrían interrumpir servicios e incluso provocar cierres.

Full and Very Deep (Opción de pago*): Incluye la configuración de Full and Fast Ultimate, pero los resultados del escaneo de puertos y la detección de servicios no tienen impacto en la selección de los NVT. Es un escaneo muy lento.

Full and Very Deep Ultimate (Opción de pago*): Incluye la configuración del anterior tipo de escaneo, pero utilizando NVTs peligrosos que podrían colapsar el sistema. Es una configuración muy lenta y poco útil debido a que suelen encontrar las mismas vulnerabilidades que las configuraciones que no son ultimate.

*Para este proyecto se ha utilizado la plantilla Full and Fast debido a que la versión utilizada es la *community* gratuita de OpenVas. Las opciones disponibles en esta versión aparte de Full and Fast son: Discovery, Host Discovery y System Discovery.

Tras realizar el proceso mencionado en el capítulo 3, podemos observar que se ha obtenido una única vulnerabilidad en Windows 10 de nivel medio (Imagen 52), ya que lleva al día todas sus actualizaciones correspondientes. También nos muestra 10 logs (nos ofrece información que la aplicación no la considera una amenaza, pero es lo suficientemente importante para tenerlo en cuenta). Para las demás máquinas observamos el número de vulnerabilidades y su gravedad de manera gráfica (Imagen 53, 54 y 55).

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.
Sat, Sep 18, 2021 7:23 PM UTC	Done	Test Windows 10	5.0 (Medium)	0	1	0	10	0

Imagen 52: Test Windows 10

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.
Sat, Sep 18, 2021 7:42 PM UTC	Done	Test Windows Server 2008	10.0 (High)	3	1	1	16	0

Imagen 53: Test Windows Server

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.
Sat, Sep 18, 2021 8:31 PM UTC	Done	Test Android 8.1	7.5 (High)	1	0	1	6	0

Imagen 54: Test Android 8.1

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.
Mon, Feb 21, 2022 4:18 PM UTC	Done	Test metasploitable3 ubuntu1404-DEFINITIVO	10.0 (High)	6	12	2	72	0

Imagen 55: Test Metasploitable 3

OpenVas ha detectado un total de 28 vulnerabilidades entre todas las máquinas de la red.

Hay que tener en cuenta que OpenVas de forma predeterminada solo muestra las vulnerabilidades con un QoD del 70% o superior. Esto asegura que apenas haya falsos positivos.

La herramienta genera un informe desde la propia página web en PDF detallando cada una de las vulnerabilidades. Todo lo que está en el informe se puede ver en la propia aplicación, pero por comodidad se ha optado por esta opción.

A continuación, se irán explicando las vulnerabilidades detectadas entre todos los equipos y se indicará la solución a estas. Vamos a clasificarlas según su gravedad.

4.1.1 Vulnerabilidades graves

NVT: Vulnerabilidad de ejecución remota de código del protocolo de negociación SMB2 de Microsoft Windows - PUERTO 445/ tcp - WINDOWS SERVER 2008

Información sobre la vulnerabilidad:

Un atacante puede explotar este problema para ejecutar código con privilegios de nivel de SISTEMA. Es probable que los intentos de explotación fallidos provoquen condiciones de denegación de servicio.

Posibles soluciones:

«<https://docs.microsoft.com/en-us/security-updates/securitybulletins/2009/ms09-050>»

NVT: Vulnerabilidades múltiples del servidor SMB de Microsoft Windows: remoto (4013389) – PUERTO 445/ tcp - WINDOWS SERVER 2008

Información sobre la vulnerabilidad:

Existen múltiples fallos debido a la forma en que el servidor Microsoft Server Message Block 1.0 (SMBv1) maneja ciertas solicitudes.

Posibles soluciones:

«<https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010>»

NVT: Vulnerabilidad de omisión de autenticación de sesión nula SMB/NETBIOS de Microsoft Windows – PUERTO 445/ tcp - WINDOWS SERVER 2008

Información sobre la vulnerabilidad:

Un recurso compartido SMB permite el acceso completo a los usuarios invitados. Si la cuenta de invitado está habilitada, cualquier persona puede acceder al pc sin una cuenta de usuario o contraseña válida.

Posibles soluciones:

«<https://seclab.cs.ucdavis.edu/projects/testing/vulner/38.html>»

*SMB (Server Message Block): Es un protocolo cliente-servidor que controla el acceso a archivos y directorios, así como otros recursos de red.

*NETBIOS: Protocolo estándar de IBM que proporciona servicios de comunicación entre redes locales.

NVT: Puente de depuración de Android (ADB) accesible sin autenticación - PUERTO 5555/ tcp - ANDROID 8.1

Información sobre la vulnerabilidad:

La secuencia de comandos comprueba si el host de destino está ejecutando un servicio compatible con el protocolo Android Debug Bridge (ADB) sin una autenticación habilitada.

Posibles soluciones:

«<https://www.mageni.net/vulnerability/android-debug-bridge-adb-accessible-without-authentication-108450>»

«<https://doublepulsar.com/root-bridge-how-thousands-of-internet-connected-android-devices-now-have-no-security-and-are-b46a68cb0f20>»

«<https://nelenkov.blogspot.com/2013/02/secure-usb-debugging-in-android-422.html>»

NVT: ProFTPD `mod_copy` Copia no autenticada de archivos a través del SITIO CPTO/CPTO - PUERTO 21/ tcp - METASPLOITABLE 3

Información sobre la vulnerabilidad:

ProFTPD es propenso a una vulnerabilidad de copia no autenticada de archivos.

Posibles soluciones:

«http://bugs.proftpd.org/show_bug.cgi?id=4169»

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.105254>»

«<https://nvd.nist.gov/vuln/detail/CVE-2015-3306>»

* ProFTPD: Servidor nft de Linux

NVT: Informes de inicios de sesión de fuerza bruta FTP - PUERTO 21/ tcp - METASPLOITABLE 3

Información sobre la vulnerabilidad:

Es posible iniciar sesión en el servidor FTP remoto usando credenciales débiles/conocidas.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.108718>»

«<https://null-byte.wonderhowto.com/how-to/brute-force-ftp-credentials-get-server-access-0208763/>»

NVT: Inicios de sesión de fuerza bruta SSH con informes de credenciales predeterminadas - PUERTO 22/ tcp - METASPLOITABLE 3

Información sobre la vulnerabilidad:

Es posible iniciar sesión en el servidor SSH remoto utilizando las credenciales predeterminadas.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.103239>»

NVT: Ejecución remota de código de Drupal Coder – PUERTO 80/ tcp - METASPLOITABLE 3**Información sobre la vulnerabilidad:**

El módulo Coder no valida suficientemente las entradas del usuario en un archivo de script que tiene la extensión php. Un usuario malicioso no autenticado puede realizar solicitudes directamente a este archivo para ejecutar código php arbitrario.

Posibles soluciones:

«<https://www.drupal.org/node/2765575>»

NVT: Probar métodos peligrosos HTTP – PUERTO 80/ tcp - METASPLOITABLE 3**Información sobre la vulnerabilidad:**

Servidores web mal configurados que permiten al cliente remotos métodos http peligrosos como ELIMINAR o PONER.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.10498>»

NVT: Vulnerabilidad de inyección SQL Core de Drupal - METASPLOITABLE 3**Información sobre la vulnerabilidad:**

Drupal no desinfecta suficientemente los datos proporcionados por el usuario antes de usarlos en una consulta SQL.

Posibles soluciones:

«<https://www.incibe-cert.es/alerta-temprana/vulnerabilidades/cve-2014-3704>»

4.1.2 Vulnerabilidades medias

NVT: Informes de enumeración de servicios DCE/RPC y MSRPC – PUERTO 135/ tcp - WINDOWS 10 / WINDOWS SERVER 2008**Información sobre la vulnerabilidad:**

Entorno informático distribuido/llamadas a procedimientos remotos (DCE/RPC) o servicios MSRPC que se ejecutan en el host remoto se pueden enumerar conectándose al puerto 135 y haciendo las consultas apropiadas.

Posibles soluciones:

«<http://www.securityspace.com/es/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.10736>»
«<https://answers.microsoft.com/es-es/windows/forum/all/widnows-server-mitigar-vulnerabilidad-dcerpc-and/5ab3f7b2-eaf5-4168-a103-3442e323b7a2>»

NVT: inicio de sesión de texto claro sin cifrar de FTP – PUERTO 21/ tcp – METASPLOITABLE 3

Información sobre la vulnerabilidad:

El host remoto está ejecutando un servicio FTP, que permite inicios de sesión de texto claro a través de conexiones sin cifrar.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.108528>»

NVT: algoritmos de cifrado débil SSH compatibles – PUERTO 22/ tcp – METASPLOITABLE 3

Información sobre la vulnerabilidad:

El servidor SSH remoto está configurado para permitir o admitir algoritmos de cifrado débiles.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.105611>»

NVT: jQuery < 1.9.0 Vulnerabilidad XSS / jQuery < 1.6.3 Vulnerabilidad XSS – PUERTO 80/ tcp – METASPLOITABLE 3

Información sobre la vulnerabilidad:

jQuery es propenso a una vulnerabilidad de secuencias de comandos entre sitios (XSS) a través del método de carga.

*jQuery es una librería de JavaScript usado para desarrollo web.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.143968>»
«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.141637>»

NVT: Instaladores de aplicaciones web desprotegidos (HTTP) – PUERTO 80/ tcp – METASPLOITABLE 3

Información sobre la vulnerabilidad:

El script intenta identificar las páginas de instalación de varias aplicaciones web que son de acceso público y no están protegidas por restricciones de cuenta.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.107307>»

NVT: Divulgación de archivos confidenciales (HTTP) – PUERTO 80/ tcp – METASPLOITABLE 3**Información sobre la vulnerabilidad:**

El script intenta identificar archivos que contienen datos confidenciales en el servidor web remoto.

Posibles soluciones:

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.107305>»

NVT: transmisión de texto sin cifrar de información confidencial a través de HTTP – PUERTO 80/ tcp – METASPLOITABLE 3**Información sobre la vulnerabilidad:**

El host/aplicación transmite información confidencial (nombre de usuario, contraseñas) en texto claro a través de HTTP.

Posibles soluciones:

«<https://cwe.mitre.org/data/definitions/319.html>»

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.108440>»

NVT: SSL/TLS: Información de conjuntos de cifrado vulnerables para HTTPS – PUERTO 631/ tcp – METASPLOITABLE 3**Información sobre la vulnerabilidad:**

Esta rutina informa todos los conjuntos de cifrado SSL/TLS aceptados por un servicio donde los vectores de ataque existen solo en los servicios HTTPS.

Posibles soluciones:

«<http://www.securityspace.com/es/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.108031>»

NVT: SSL/TLS: Detección de protocolo TLSv1.0 y TLSv1.1 en desuso – PUERTO 631/ tcp – METASPLOITABLE 3**Información sobre la vulnerabilidad:**

Protocolo obsoleto TLSv1.0 y TLSv1.1 con fallos criptográficos

Posibles soluciones:

«<https://www.mageni.net/vulnerability/ssl-tls-deprecated-tls-v1-0-and-tls-v1-1-protocol-detection-117274>»

*SSL: Secure Sockets Layer (capa de sockets seguros). Protocolo para mantener una conexión de red segura.

*TLS: Transport Layer Security (seguridad de la capa de transporte). Versión actualizada y mejorada de SSL.

4.1.3 Vulnerabilidades leves

NVT: marcas de tiempo TCP - METASPLOITABLE 3 / WINDOWS SERVER 2008 / ANDROID 8.1

Información sobre la vulnerabilidad:

El servicio remoto implementa marcas de tiempo TCP. Es decir, se puede calcular el tiempo de actividad del host remoto.

Posibles soluciones:

«<https://beyondsecurity.com/scan-pentest-network-vulnerabilities-tcp-timestamps-retrieval.html>»

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.80091>»

NVT: algoritmos MAC débiles de SSH admitidos - METASPLOITABLE 3

Información sobre la vulnerabilidad:

El servidor SSH remoto está configurado para permitir o admitir algoritmos MAC débiles.

Posibles soluciones:

«<https://www.ibm.com/support/pages/ssh-weak-mac-algorithms-enabled-vulnerability-mitigation-security-network-ips-appliances>»

«<http://www.securityspace.com/smysecure/catid.html?id=1.3.6.1.4.1.25623.1.0.105610>»

2.1.1 High 21/tcp

High (CVSS: 10.0) NVT: ProFTPD 'mod_copy' Unauthenticated Copying Of Files Via SITE CPFR/CPTO
Product detection result cpe:/a:proftpd:proftpd:1.3.5 Detected by ProFTPD Server Version Detection (Remote) (OID: 1.3.6.1.4.1.25623.1.1.0.900815)
Summary ProFTPD is prone to an unauthenticated copying of files vulnerability.
Vulnerability Detection Result The target was found to be vulnerable
Impact Under some circumstances this could result in remote code execution
Solution: Solution type: VendorFix Ask the vendor for an update
Vulnerability Detection Method Try to copy /etc/passwd to /tmp/passwd.copy with SITE CPFR/CPTO Details: ProFTPD 'mod_copy' Unauthenticated Copying Of Files Via SITE CPFR/CPTO OID:1.3.6.1.4.1.25623.1.0.105254 Version used: 2021-04-16T06:57:08Z
Product Detection Result Product: cpe:/a:proftpd:proftpd:1.3.5 Method: ProFTPD Server Version Detection (Remote) OID: 1.3.6.1.4.1.25623.1.0.900815)
References cve: CVE-2015-3306 url: http://bugs.proftpd.org/show_bug.cgi?id=4169 cert-bund: CB-K15/0791 cert-bund: CB-K15/0553 dfn-cert: DFN-CERT-2015-0839 dfn-cert: DFN-CERT-2015-0576

Imagen 56: PDF generado por OpenVas mostrando una vulnerabilidad

4.2 Resultados del estudio de Wazuh

4.2.1 Vulnerabilities

Para saber cómo interpretar los resultados hay que tener claro el funcionamiento del módulo de vulnerabilidades de Wazuh.

El administrador Wazuh crea una base de datos (única para cada agente) a partir de repositorios CVE disponibles que proporcionan los agentes. A partir de aquí se van generando alertas cuando un CVE afecta a un paquete que se sabe que está instalado en alguno de los agentes.

Tras 6 horas de análisis se han obtenido 209 alertas críticas, 582 alertas altas, 680 alertas y 102 alertas leves. Todas ellas son del equipo Metasploitable 3, ya que no ha detectado ninguna alerta en la máquina de Windows 10, no es compatible con Android y a pesar de dejar el registro del agente de Windows Server 2008, no es compatible con el módulo de vulnerabilidades.

Critical Severity Alerts 209	High Severity Alerts 582	Medium Severity Alerts 680	Low Severity Alerts 102
--	------------------------------------	--------------------------------------	-----------------------------------

Time	data.vulnerability.package.name	data.vulnerability.cve	data.vulnerability.severity
> Feb 6, 2022 @ 19:24:11.221	linux-image-generic	CVE-2017-17886	High
> Feb 6, 2022 @ 19:24:11.180	linux-image-generic	CVE-2017-17885	High
> Feb 6, 2022 @ 19:24:11.018	openssl	CVE-2016-0798	High
> Feb 6, 2022 @ 19:24:10.987	openssl	CVE-2016-0797	High
> Feb 6, 2022 @ 19:24:10.957	linux-image-generic	CVE-2017-16939	High
> Feb 6, 2022 @ 19:24:10.857	linux-image-generic	CVE-2018-9518	High
> Feb 6, 2022 @ 19:24:10.737	linux-image-generic	CVE-2017-12154	High
> Feb 6, 2022 @ 19:24:10.656	bash	CVE-2016-7543	High

Imagen 57: Visualización de las alertas Wazuh

Cada alerta nos va a proporcionar mucha información relevante. Además de proporcionar el CVE, la fecha y hora de la alerta, la severidad y el paquete al que afecta la vulnerabilidad (Imagen 57), si expandimos la alerta podemos ver más información. Se observa una breve descripción junto al impacto que puede provocar y múltiples referencias para solucionarla (Imagen 58). Además, a pesar de que Wazuh nos dice la gravedad de la vulnerabilidad (Critical, High, Medium y Low), sin una regla establecida, la cual aparece en la información proporcionada (Imagen 59), no podemos comprender bien el significado. El manual de la herramienta proporciona una tabla sobre el nivel de la regla.

«<https://documentation.wazuh.com/current/user-manual/ruleset/rules-classification.html>»

```

f data.vulnerability.rationale
>
The fmtstr function in crypto/bio/b_print.c in OpenSSL 1.0.1 before 1.0.1s and 1.0.2 before 1.0.2g improperly calculates string length, which allows remote attackers to cause a denial of service (overflow and out-of-bounds read) or possibly have unspecified other impact via a long string, as demonstrated by a large amount of ASN.1 data, a different vulnerability than CVE-2016-2842.

f data.vulnerability.references
>
https://git.openssl.org/?p=openssl.git;a=commit;h=578b956fe741bf8e84055547b1e83c28dd902c73, http://openssl.org/news/secadv/20160301.txt, https://kb.pulsesecure.net/articles/Pulse_Security_Advisories/SA40168, http://www.oracle.com/technetwork/security-advisory/cpua-pr2016v3-2985753.html, https://h20566.www2.hp.com/portal/site/hpsc/public/kb/docDisplay?docId=emr_na-c05157667, https://h20566.www2.hp.com/portal/site/hpsc/public/kb/docDisplay?docId=emr_na-c05119617, https://h20566.www2.hp.com/portal/site/hpsc/public/kb/docDisplay?docId=emr_na-c05131085, https://h20566.www2.hp.com/portal/site/hpsc/public/kb/docDisplay?docId=emr_na-c05150800, http://rhn.redhat.com/errata/RHSA-2016-0722.html, ht

```

Imagen 58: Descripción, impacto y referencias sobre una de las vulnerabilidades de Wazuh

† rule.id	23506
# rule.level	13

Imagen 59: Id y nivel de la regla

Se ha hecho una selección de las vulnerabilidades más comunes (CVEs) que han aparecido en el análisis y de los paquetes más afectados.

Vulnerabilidad en telnetd en FreeBSD (CVE-2011-4862)

Información sobre la vulnerabilidad:

Copia de búfer sin comprobación del tamaño de entrada (Desbordamiento de búfer clásico)

Posibles soluciones:

«<https://www.incibe-cert.es/alerta-temprana/vulnerabilidades/cve-2011-4862>»

«<https://vuldb.com/es/?id.4504>»

CVE-2017-11103

Información sobre la vulnerabilidad:

Vulnerabilidad en Heimdal (es una implementación libre de Kerberos 5, el cual es un protocolo de autenticación de red) que conlleva una suplantación de identidad

Posibles soluciones:

«<https://access.redhat.com/security/cve/cve-2017-11103>»

«<https://www.samba.org/samba/security/CVE-2017-11103.html>»

CVE-2017-3135

Información sobre la vulnerabilidad:

Fallo de denegación de servicio en la forma en que BIND (servidor DNS común en sistemas Unix) manejó las respuestas de consulta cuando se usaron tanto DNS64 como RPZ

Posibles soluciones:

«<https://kb.isc.org/docs/aa-01453>»

Vulnerabilidad en ISC BIND (CVE-2015-5477)

Información sobre la vulnerabilidad:

Fallo en la gestión de las peticiones TKEY, lo que lleva a un ataque de denegación de servicio y por consiguiente el fallo completo de BIND

Posibles soluciones:

«<https://kb.isc.org/docs/aa-01272>»

Principalmente hay 5 paquetes que se ven afectados:

- **Libcurl3-gnutls**: Paquete de transferencia de URL del lado del cliente.
- **Ntpdate**: Paquete utilizado para la sincronización de relojes a través de la red.
- **Openssl**: Paquete que forma la biblioteca criptográfica que implementa los protocolos SSL y TLS para la comunicación segura entre redes.
- **Tcpdump**: El paquete forma parte de la herramienta tcpdump, cuya función es capturar y mostrar en tiempo real los paquetes transmitidos y recibidos por la red.

- **Linux-image-generic:** Es un metapaquete para la última versión de Kernel (núcleo del sistema operativo Linux).

4.2.2 Integrity monitoring

Integrity monitoring es uno de los módulos que Wazuh proporciona para detectar cambios en el sistema de ficheros (cambios en los permisos de un archivo, cambios en el contenido de un fichero...)

Automáticamente Wazuh no va a detectar nada si no configuramos el directorio que queremos que se vaya analizando en tiempo real. Eso lo haremos añadiendo el siguiente comando en el archivo `/var/ossec/etc/ossec.conf` del agente:

```
<directories check_all="yes" realtime="yes" report_changes="yes">C:\Antonio</directories>
```

Para probar este módulo se ha creado en Windows 10 un archivo llamado “prueba.txt”, el cual se ha modificado, se ha cambiado un permiso y se ha borrado. Wazuh nos ha generado las siguientes alertas:

Time	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
> Mar 4, 2022 @ 12:55:54.346	c:\antonio\prueba.txt	deleted	File deleted.	7	553
> Mar 4, 2022 @ 12:55:42.688	c:\antonio\prueba.txt	modified	Integrity checksum changed.	7	550
> Mar 4, 2022 @ 12:54:26.307	c:\antonio\prueba.txt	modified	Integrity checksum changed.	7	550
> Mar 4, 2022 @ 12:54:06.454	c:\antonio\prueba.txt	modified	Integrity checksum changed.	7	550
> Mar 4, 2022 @ 12:53:37.462	c:\antonio\prueba.txt	added	File added to the system.	5	554

Imagen 60: Alertas del módulo Integrity Monitoring

Nos ha dado como resultado 4 alertas en las que se puede observar cuando se ha añadido y cuando se ha borrado el archivo en cuestión. Vemos también que tenemos 3 alertas de modificación de archivos, pero no sabemos qué tipo de cambio se ha hecho. Para ello tendremos que expandir las alertas y ver con más detalle los datos que nos dan. Vamos a ir enumerando la información más interesante:

1. Tamaño antes y después de modificar el archivo:

```
# syscheck.size_after      31
# syscheck.size_before     0
```

2. Hash antes y después del cambio en el archivo. Wazuh también nos proporciona las funciones hash SHA-256 y MD5. La utilización del hash es la forma en la que la herramienta sabe que el archivo ha cambiado. Cuando se crea el archivo se genera un hash que se guarda para compararlo y ver si ha habido algún cambio.

```
t syscheck.sha1_after          9c959a880e8fa56413f6372623f2909b5a4d00d4
t syscheck.sha1_before        da39a3ee5e6b4b0d3255bfef95601890afd80709
```

3. El cambio exacto del archivo antes y después de ser modificado:

```
t syscheck.diff                < Tarjeta de credito: 123456789
                               ---
                               > Tarjeta de credito: 987654321
```

4. Técnica MITRE utilizada. Hablaremos más adelante del módulo específico de Wazuh MITRE ATT&CK:

```
t rule.mitre.technique        Stored Data Manipulation
t rule.mitre.technique        File Deletion, Data Destruction
```

5. Breve descripción del evento identificado:

```
t rule.description            File added to the system.
t rule.description            Integrity checksum changed.
```

6. Información sobre cómo se relaciona con los diversos capítulos de estándares de cumplimiento:

```
t rule.gdpr                   II_5.1.f
t rule.hipaa                   164.312.c.1, 164.312.c.2
t rule.nist_800_53             SI.7
t rule.pci_dss                 11.5
t rule.tsc                     PI1.4, PI1.5, CC6.1, CC6.8, CC7.2, CC7.3
```

PCI DSS: es un estándar de seguridad de la información patentado para organizaciones que manejan tarjetas de crédito de las principales compañías (Visa, MasterCard...). Se creó para reducir el fraude de datos del titular de la tarjeta.

GDPR: reglamento general de protección de datos de la Unión Europea.

HIPAA: ley de transparencia y responsabilidad del seguro médico.

NIST-800-53: publicación del Instituto Nacional de Estándares y Tecnología que recomienda controles de seguridad para organizaciones y sistemas de información federales.

TSC: criterios de servicios de confianza

4.2.3 VirusTotal

VirusTotal es un software externo creado por una empresa española (Hispanic sistemas), cuya funcionalidad es escanear archivos y URLs en busca de contenido malicioso. Detecta virus, gusanos, troyanos y otros tipos de contenido sospechoso mediante motores antivirus y escáneres web. Va comparando los hashes de los archivos con una base de datos de hashes conocidos maliciosos. Cualquier archivo que tengamos en un equipo se puede analizar en tiempo real.

Es una herramienta gratuita, aunque hay una versión premium.

Este módulo no forma parte de Wazuh, pero sí tenemos la opción de añadirlo. Para implementarlo hay que seguir una serie de pasos:

- Hay que registrarse en la web de VirusTotal.
(www.virustotal.com/gui/home/upload)
- Una vez completado el registro se solicitará una key gratuita que integraremos en el servidor Wazuh.

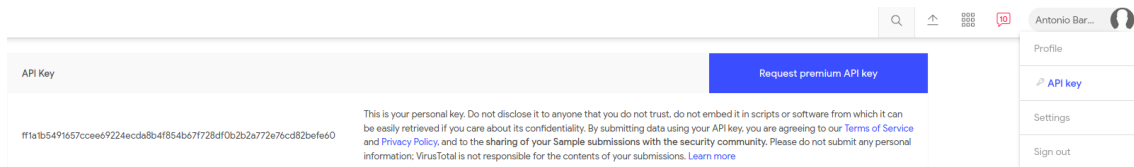


Imagen 61: Contraseña de activación VirusTotal

- Dentro del archivo **/var/ossec/etc/ossec.conf** añadiremos una serie de líneas de código:

```
<!--Módulo VirusTotal -->
<integration>
  <name>virustotal</name>
  <api_key>ff1a1b5491657ccee69224ecda8b4f854b67f728df0b2b2a772e76cd82bef60</api_key>
  <group>syscheck</group>
  <alert_format>json</alert_format>
</integration>
```

- Por último, configuraremos en el cliente el directorio donde queremos que Wazuh analice los archivos en tiempo real. Utilizaremos el mismo que se ha configurado para el módulo *integrity monitoring*, por lo tanto, la línea de código será la misma.

Tras realizar la configuración se han descargado 2 archivos infectados por virus genéricos (<https://dasmalwerk.eu>) para analizar el funcionamiento de la herramienta.

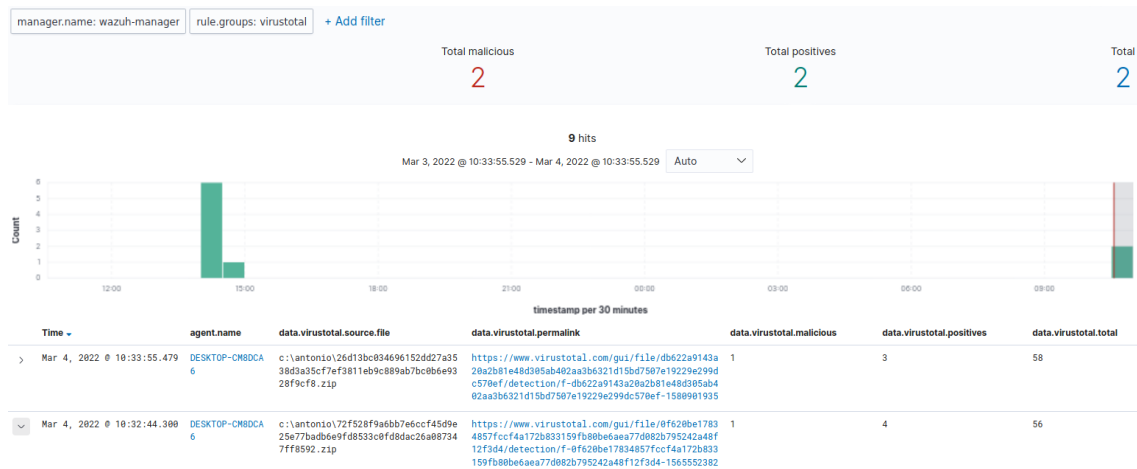


Imagen 62: Resultados de VirusTotal (1)

Analizando la información que nos da el módulo vemos que efectivamente Wazuh ha detectado 2 alertas de archivos maliciosos. Nos da una información similar a la interfaz del módulo *integrity monitoring*, pero nos proporciona además un enlace a la página oficial de virustotal, donde nos da mucha más información del tipo de malware.

«<https://www.virustotal.com/gui/file/db622a9143a20a2b81e48d305ab402aa3b6321d15bd7507e19229e299dc570ef/detection>»

«<https://www.virustotal.com/gui/file/0f620be17834857fcc4a172b833159fb80be6aea77d082b795242a48f12f3d4/detection/f-0f620be17834857fcc4a172b833159fb80be6aea77d082b795242a48f12f3d4-1565552382>»

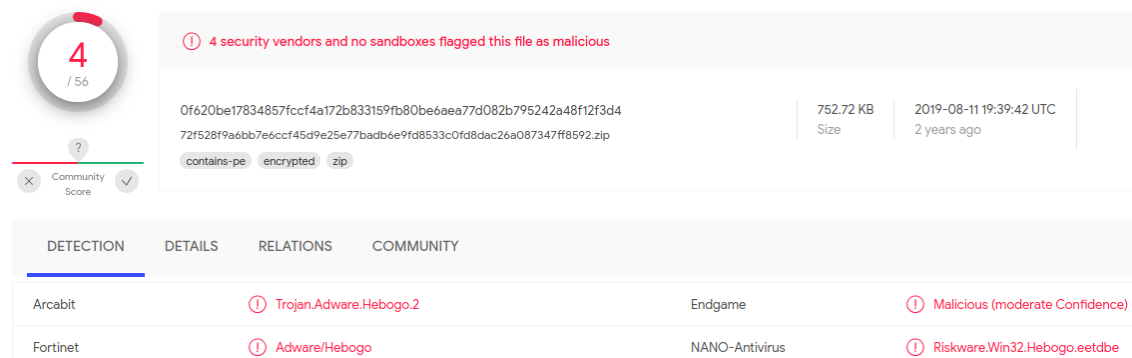


Imagen 63: Resultados de VirusTotal (2)

Wazuh además es capaz de iniciar un procedimiento de respuesta activa para actuar ante una amenaza.

4.2.4 MITRE ATT&CK

MITRE ATT&CK es una base de datos de tácticas y técnicas accesibles a nivel mundial que están fundamentadas en observaciones de las amenazas a la seguridad. Se muestran en matrices organizadas por etapas de ataque. Esto esta implementado como un módulo en Wazuh.

Para probar el módulo, vamos a realizar un ataque ssh de fuerza bruta desde nuestro terminal de Kali Linux mediante la herramienta ncrack que ya viene preinstalada en el equipo. Se hará con el comando: **ncrack -v -user admin 192.168.100.11:22**

Veremos cómo van saltando muchas alertas y observamos que la regla 5710 es la que más se repite, la cual corresponde a un intento de inicio de sesión con un usuario inexistente. En 5 min se ha obtenido 100 alertas de esta regla.

Time	rule.mitre.id	rule.mitre.tactic	rule.description	rule.level	rule.id
> Mar 6, 2022 @ 18:41:25.592	T1110	Credential Access	sshd: Attempt to login using a non-existent user	5	5710
> Mar 6, 2022 @ 18:41:25.587	T1110	Credential Access	sshd: Attempt to login using a non-existent user	5	5710
> Mar 6, 2022 @ 18:41:25.579	T1110	Credential Access	sshd: Attempt to login using a non-existent user	5	5710
> Mar 6, 2022 @ 18:41:25.577	T1110	Credential Access	sshd: Attempt to login using a non-existent user	5	5710

Imagen 64: Alertas de seguridad del módulo MITRE ATT&CK

Si queremos observar exactamente qué es lo que está pasando, pulsaremos sobre el número de la técnica mitre que se nos da (T1110). Este nos dirá que se está produciendo un ataque de fuerza bruta y detallará qué es lo que puede estar ocurriendo.

Brute Force

ID
T1110 [es](#)

Tactic
Credential Access

Platform
Azure AD, Linux, Office 365, SaaS, Windows, macOS

Data sources
Office 365 account logs, Authentication logs

Version
2.0

Description
Adversaries may use brute force techniques to attempt access to accounts when passwords are unknown or when password hashes are obtained. [Credential Dumping](#) is used to obtain password hashes, this may only get an adversary so far when [Pass the Hash](#) is not an option. Techniques to systematically guess the passwords used to compute hashes are available, or the adversary may use a pre-computed rainbow table to crack hashes. Cracking hashes is usually done on adversary-controlled systems outside of the target network. [\[1\]](#)

Adversaries may attempt to brute force logins without knowledge of passwords or hashes during an operation either with zero knowledge or by attempting a list of known or possible passwords. This is a riskier option because it could cause numerous authentication failures and account lockouts, depending on the organization's login failure policies. [\[2\]](#)

A related technique called password spraying uses one password (e.g. "Password01"), or a small list of passwords, that matches the complexity policy of the domain and may be a commonly used password. Logins are attempted with that password and many different accounts on a network to avoid account lockouts that would normally occur when brute forcing a single account with many passwords. [\[3\]](#)

Typically, management services over commonly used ports are used when password spraying. Commonly targeted services include the following:

- SSH (22/TCP)
- Telnet (23/TCP)
- FTP (21/TCP)
- NetBIOS / SMB / Samba (139/TCP & 445/TCP)
- LDAP (389/TCP)
- Kerberos (88/TCP)
- RDP / Terminal Services (3389/TCP)
- HTTP/HTTPS Management Services (80/TCP & 443/TCP)
- MSSQL (1433/TCP)
- Oracle (1521/TCP)
- MySQL (3306/TCP)
- VNC (5900/TCP)

In addition to management services, adversaries may "target single sign-on (SSO) and cloud-based applications utilizing federated authentication protocols," as well as externally facing email applications, such as Office 365. [\[4\]](#)

In default environments, LDAP and Kerberos connection attempts are less likely to trigger events over SMB, which creates Windows "login failure" event ID 4625.

References

1. Wikipedia Password cracking
2. Cylance Cleaver
3. BlackHillsInfosec Password Spraying
4. US-CERT TA18-068A 2018

Imagen 65: Descripción del ataque por fuerza bruta

4.3 Diferencias y similitudes entre Wazuh y OpenVas

Dejando fuera el módulo de detección de vulnerabilidades, las funciones que ofrece Wazuh no se pueden comparar con OpenVas. Son abrumadoras. Desde analizar cambios en archivos, pasando por detectar malwares, hasta identificar si nos están atacando y con qué método e incluso iniciar una respuesta. No sería justo comparar las dos aplicaciones si no nos centramos más en la funcionalidad que comparten.

OpenVas es una herramienta gratuita y de código abierto, aunque hay una versión de pago que mejora mínimamente las funciones de escaneo. Wazuh también es una herramienta de código abierto y gratuita, pero tiene una versión premium que brinda al cliente un mejor soporte. Las versiones gratuitas son suficientemente potentes como para no depender de las otras.

OpenVas es únicamente un escáner de vulnerabilidades y además proporciona información general del host que se va a analizar. Está basado principalmente en scripts denominados NVTs. Se centra en la búsqueda de vulnerabilidades en los puertos del host. Cada NVT OpenVas lo puede organizar en varios CVEs. No todos los NVTs tienen un identificador CVE asociado. Wazuh funciona un poco distinto. Se centra en buscar paquetes vulnerables y los resultados se presentan como alertas con un identificador CVE.

Continuamente se están actualizando los NVTs y los CVEs en ambas herramientas, por lo que se está al tanto de las últimas brechas de seguridad que pueden ir apareciendo.

Las dos herramientas cuentan con extensos manuales, pero quizás debido a que la comunidad de OpenVas es mucho más amplia se ha conseguido realizar un estudio más preciso de esta.

La compatibilidad en cuanto a sistemas operativos soportados es uno de los puntos más fuertes en ambas aplicaciones. Si bien es cierto que se han tenido problemas con Android 8.1 en Wazuh debido a que no acepta S.O Android, y con Windows Server 2008 únicamente en el módulo de detección de vulnerabilidades.

OpenVas es una herramienta con una interfaz muy intuitiva, por lo que es fácil de manejar para las personas que no tienen un amplio conocimiento en el ámbito de la seguridad, en cambio Wazuh es muy compleja, por lo que es necesario amplios conocimientos técnicos para su uso, pero sobre todo para interpretar los resultados. La configuración con los agentes en Wazuh para que sean detectados por la aplicación muchas veces se puede convertir en un problema sin un adecuado manejo.

Wazuh se puede instalar en cualquier sistema operativo cotidiano, por lo que está disponible para un público más amplio, en cambio para OpenVas se necesita un equipo con Kali Linux o por excepción Centos 7, aunque su instalación sería más compleja.

El tiempo de detección de vulnerabilidades en un sistema es similar en Wazuh y OpenVas. Eso sí, todo depende de las características técnicas que tengas en el pc en el que se ha montado la red virtual. Se obtienen mejores resultados si en el escaneo se proporcionan las credenciales del sistema a analizar.

Wazuh descubre muchas más vulnerabilidades que OpenVas, pero una de las causas es que este solo presenta las vulnerabilidades más críticas y con un grado de fiabilidad superior a un 70% de QoD (se puede configurar para que muestre un QoD por debajo del 70% pero no sería útil). Wazuh no proporciona un QoD, por lo que no podemos determinar la calidad del análisis pudiendo surgir muchos falsos positivos. Otra de las causas es la manera en la que Wazuh detecta las vulnerabilidades y como las presenta.

OpenVas es más completo en cuanto a detector de vulnerabilidades se refiere, ya que ofrece la información de una manera más clara y menos confusa que Wazuh.

Wazuh no informa sobre como mitigar la vulnerabilidad, simplemente muestra referencias externas. OpenVas aparte de aportar referencias, en el propio informe sí hace distinción de cual podría ser la solución.

OpenVas no puede integrar herramientas externas, a diferencia de Wazuh, como se ha podido comprobar con la implementación de VirusTotal.

Aunque no se ha podido comprobar debido a que solo está disponible en la versión de pago, sabemos que Wazuh proporciona servicios en la nube y OpenVas está empezando a implementarlas ahora.

Todas estas diferencias conllevan a que OpenVas sea una herramienta más destinada a empresas pequeñas, aunque en general si se quiere un software que brinde un soporte completo a la red Wazuh no tiene competencia.

Ambas son herramientas muy potentes, pero se necesitan la una a la otra para ser perfectas.

4.4 Conclusiones

Este proyecto se ha centrado en estudiar las prestaciones en profundidad de dos herramientas para garantizar la seguridad en la red, como son OpenVas y Wazuh. En un primer momento se pretendía analizar solo las funcionalidades comunes entre ambas herramientas, pero conforme avanzaba el proyecto vi que Wazuh proporcionaba todas las funciones necesarias para un análisis completo de la red e incluso se podían integrar funciones adicionales externas, por lo que se amplió el estudio de esta herramienta.

A lo largo de este TFG, ha quedado claro, que a pesar de las diferencias notables entre Wazuh y OpenVas, estas son herramientas muy útiles para el análisis de vulnerabilidades y se complementan perfectamente.

Como conclusión, una vez finalizado el desarrollo del trabajo, podemos afirmar que se han conseguido alcanzar todos los objetivos planteados desde el inicio e incluso llegando a lograr algunos adicionales.

Para futuros proyectos, sería interesante seguir profundizando en la funcionalidad de Wazuh, ya que hay opciones y configuraciones más ocultas que no se han llegado a ver cómo podría ser el uso del módulo de integración para extraer datos de la nube, la

configuración de mecanismos de respuesta antes diferentes situaciones o la visualización de resultados en Kibana.

Referencias

- A, D. (s.f.). *OpenVAS, instala este explorador de vulnerabilidades en Ubuntu 16.04*.
Obtenido de UbuLog: <https://ubunlog.com/openvas-instala-este-explorador-vulnerabilidades-ubuntu-16-04/>
- A2Secure. (12 de Febrero de 2019). *Sistemas IDS, IPS, HIDS, NIPS, SIEM ¿Qué son?*
Obtenido de <https://www.a2secure.com/blog/ids-ips-hids-nips-siem-que-es-esto/>
- Altube, R. (5 de Noviembre de 2021). *Kali Linux: Qué es y características principales*.
Obtenido de OpenWebinars: <https://openwebinars.net/blog/kali-linux-que-es-y-caracteristicas-principales/>
- Álvarez Huerta, L. (30 de Mayo de 2014). *OpenVas en Linux: Explorando nuestros sistemas*. Obtenido de OpenWebinars:
<https://openwebinars.net/blog/openvas-en-linux-explorando-nuestros-sistemas/>
- Castillo, J. A. (25 de Noviembre de 2018). *Como instalar Android en VirtualBox*.
Obtenido de Profesionalreview:
<https://www.profesionalreview.com/2018/11/25/instalar-android-virtualbox/>
- Castillo, J. A. (16 de Diciembre de 2018). *Formas de conectar dos máquinas virtuales en red VirtualBox*. Obtenido de Profesional review:
<https://www.profesionalreview.com/2018/12/16/conectar-maquinas-virtuales-en-red-virtualbox/>
- Diego, C. (Agosto de 2020). *Wazuh - Plataforma de seguridad*. Obtenido de <https://diegochecha.hashnode.dev/wazuh-plataforma-de-seguridad-cke8zoumv00ptx3s1agtlbo4r>
- GSM. (2 de Diciembre de 2021). *Greenbone Security Manager (GSM)*. Obtenido de Greenbone: <https://docs.greenbone.net/GSM-Manual/gos-20.08/en/reports.html#quality-of-detection-concept>
- Ibili, R. (11 de Junio de 2019). *Vulnerabilidades: OpenVas*. Obtenido de Security Labs:
<https://securitylabs.es/elementor-1138-2-2-2-3-2-2-2/>
- Ichasco. (8 de Febrero de 2015). *Ossim: Instalación y configuración*. Obtenido de Blog ichasco: <https://blog.ichasco.com/ossim/>
- Kali Linux: Qué es y características principales*. (25 de Febrero de 2020). Obtenido de Solvetic: <https://www.solvetic.com/tutoriales/article/8278-como-instalar-openvas-en-kali-linux/>

Kifarunix. (11 de Julio de 2021). *Detecting Malicious Files with Wazuh and VirusTotal*.
Obtenido de Kifarunix: <https://kifarunix.com/detecting-malicious-files-with-wazuh-and-virustotal/>

Mancomun. (3 de Noviembre de 2017). *OSSEC: Sistema de detección de intrusos*.
Obtenido de Mancomun: <https://www.mancomun.gal/es/artigo-tic/ossec-sistema-de-deteccion-de-intrusos/>

Metasploitable 3. (Enero de 2022). Obtenido de Github:
<https://github.com/rapid7/metasploitable3>

Mitre Att&ck. (s.f.). Obtenido de <https://attack.mitre.org/>

Morales, F. (16 de Febrero de 2022). *Servidor Wazuh (SIEM)*. Obtenido de Sysadminsdecuba: <https://www.sysadminsdecuba.com/2022/02/servidor-wazuh-siem/>

Nguyen, T. (3 de Mayo de 2019). *5 Open Source SIEM Solutions*. Obtenido de Logdna: <https://www.logdna.com/blog/open-source-siem-tools>

Olmedo, J. (11 de Junio de 2017). *Metasploitable3: Crea una máquina vulnerable para probar tus ataques*. Obtenido de Hackpuntos: <https://hackpuntos.com/metasploitable3-crea-una-maquina-vulnerable-para-probar-tus-ataques/>

Rodríguez, A. (7 de Mayo de 2022). *Using Wazuh for Windows vulnerability detection*. Obtenido de Wazuh blog: <https://wazuh.com/blog/using-wazuh-for-windows-vulnerability-detection/>

Sanjinez, V. (12 de Febrero de 2022). *Wazuh - Defensa de Infraestructuras Tecnológicas*. Obtenido de Youtube: <https://www.youtube.com/watch?v=2Z-NvytPgJU>

Solvetic Seguridad. (22 de Noviembre de 2016). *OpenVAS suite de seguridad para análisis de vulnerabilidades*. Obtenido de Solvetic: <https://www.solvetic.com/tutoriales/article/2085-openvas-suite-de-seguridad-para-analisis-de-vulnerabilidades/>

Songer, A. (17 de Junio de 2021). *How to Install & Register Wazuh Agent on Windows and Linux (Debian-Based)*. Obtenido de Songer: <https://songer.pro/how-to-install-register-wazuh-agent-on-windows-and-linux-debian-based/>

Velasco, R. (26 de Agosto de 2016). *OSSEC Wazuh, un monitor de seguridad para redes de ordenadores*. Obtenido de Redeszone: <https://www.redeszone.net/2016/08/26/ossec-wazuh-monitor-seguridad-redes-ordenadores/>

Virtual Machine (OVA). (2022). Obtenido de Wazuh: <https://documentation.wazuh.com/current/virtual-machine/virtual-machine.html>

VirtualBox. (s.f.). Obtenido de <https://www.virtualbox.org/wiki/Downloads>

Wazuh. (2022). *Deploying Wazuh agents on Linux systems*. Obtenido de <https://documentation.wazuh.com/current/installation-guide/wazuh-agent/wazuh-agent-package-linux.html>

Wazuh. (2022). *Enhancing with MITRE*. Obtenido de <https://documentation.wazuh.com/current/user-manual/ruleset/mitre.html>

Wazuh. (2022). *File integrity monitoring*. Obtenido de <https://documentation.wazuh.com/current/user-manual/capabilities/file-integrity/index.html>