Informe de Pentesting

A continuación se describirán los pasos seguidos al realizar este informe de pentesting.

Objetivo

Realizar escaneos y pruebas de penetración en la máquina vulnerada, con el fin de poder encontrar otras vulnerabilidades diferentes a la explotada, para así corregirlas y aumentar la seguridad del sistema.

Herramientas utilizadas

Máquina vulnerada (Debian)
Máquina de ataque para el escaneo (Kali)

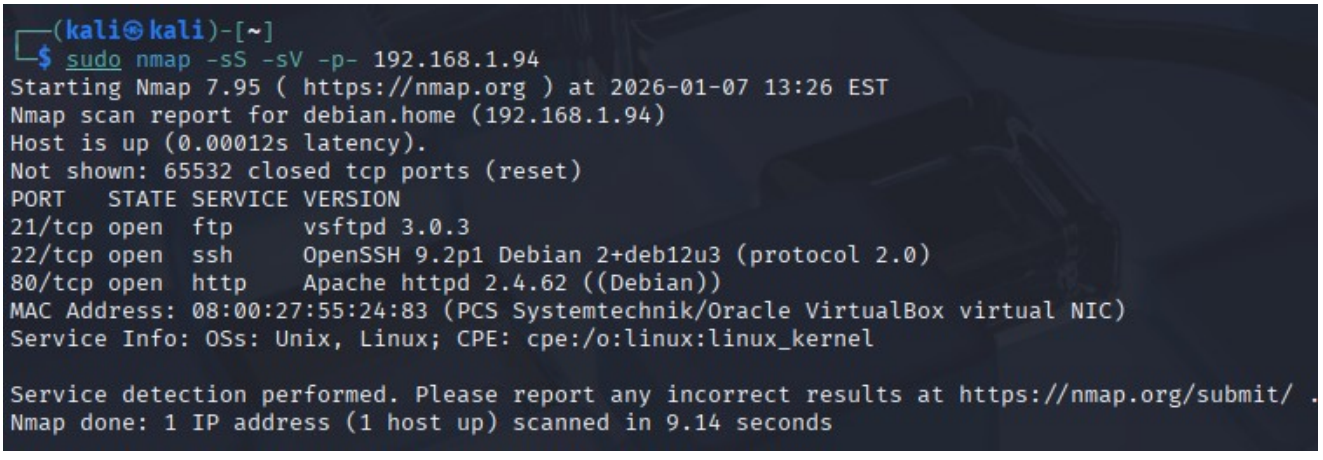
Escaneo completo del sistema

Paso 1. Escaneo completo con Nmap

Primero en este caso, ya que ambas máquinas están en la misma red se realiza un escaneo para descubrir la IP objetivo con:
"nmap -sn 192.168.1.0/24"

Con la IP ahora conocida (La IP es 192.168.1.94) se realiza un escaneo completo con
"nmap -sS -sV -p- 192.168.1.94"
Con ésto se buscó hacer un escaneo:

- -sS: Escaneo sigiloso.
- -sV: Detección de versiones de servicios.
- -p-: Escaneo de todos los puertos.

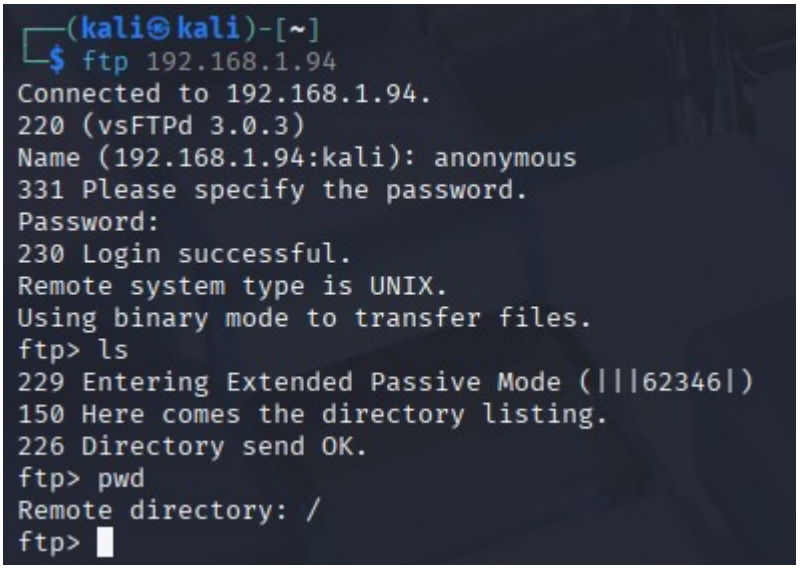


Observaciones:

El escaneo muestra que están expuestos los servicios FTP, SSH y Apache.
Se escogió explotar la vulnerabilidad FTP a través del puerto 21, ya que es un servicio que transmite credenciales en claro, es un gran riesgo ya que transmite información sin cifrar.

Explotación vulnerabilidad FTP

Para explotar esta vulnerabilidad se intentará conectar al servicio FTP para demostrar la baja seguridad y buscar soluciones.



Se estableció conexión remota al servicio FTP como usuario anonimo y sin necesidad de contraseña.
Ha sido posible ejecutar los comandos "ls" y "pwd", confirmando no sólo el acceso al servicio y directamente a la raiz, si no también la posibilidad de ejecutar comandos.

Corrección de la Vulnerabilidad

Para corregir la vulnerabilidad explotada se realizaron los siguientes pasos en la máquina vulnerada (Debian).

Mediante el comando "sudo systemctl status vsftpd" se confirma que el servicio está activo y en ejecución.

Como primera opción se deshabilitará el acceso anónimo al servicio.

Con el comando "sudo nano /etc/vsftpd.conf" se accedió al archivo de configuración de ftp:

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=YES
#
```

Se modificó a "NO" para bloquear el acceso como usuario anonimo:

```
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
```

Reinicio del servicio FTP con el comando "sudo systemctl restart vsftpd".

Para comprobar el cambio en la configuración se vuelve a intentar conectar desde Kali como se hizo anteriormente.

```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.1.94
Connected to 192.168.1.94.
220 (vsFTPd 3.0.3)
Name (192.168.1.94:kali): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp>
```

Se comprueba que el acceso como usuario anonimo no está disponible, consiguiendo así solucionar esa debilidad.

Se solucionó esta debilidad teniendo en cuenta que el servidor FTP sea necesario en esta ocasión. De no ser así, lo mas seguro es deshabilitar el servicio FTP y cerrar así la posibilidad de conexión a través de él.

El escaneo inicial mostró una debilidad de seguridad en el servicio Apache.

```
80/tcp open  http    Apache httpd 2.4.62 ((Debian))
```

Como medida de seguridad recomendada en caso de no necesitar el servicio, es desactivarlo.
En este caso establecemos que es necesario por lo que se endurecerá la seguridad.

Se modificaron los siguientes dos opciones dentro del archivo de configuración de apache:

- ServerTokens OS -> ServerTokens Prod
- ServerSignature ON -> ServerSignature Off

```
debian@debian:~$ sudo nano /etc/apache2/conf-available/security.conf
```

```
#ServerTokens Minimal
ServerTokens OS
#ServerTokens Full
```
->
```
#ServerTokens Minimal
ServerTokens Prod
#ServerTokens Full
```

```
#ServerSignature Off
ServerSignature On
```
->
```
#ServerSignature Off
ServerSignature Off
```

A continuación se modificó el archivo de configuración de virtualhost para restringir el acceso solo a localhost o la red interna y evitar conexiones remotas:

- AllowOverride All -> Require ip 127.0.0.1

```
debian@debian:~$ sudo nano /etc/apache2/sites-available/000-default.conf
```

```
DocumentRoot /var/www/html
 <Directory /var/www/html>
        AllowOverride All
</Directory>
```
->
```
DocumentRoot /var/www/html
 <Directory /var/www/html>
        Require ip 127.0.0.1
</Directory>
```

Tras esto se reinició el servicio para comprobar las medidas tomadas.

Firewall

Como medida adicional se instaló firewall mediante el comando:

- sudo apt install ufw

Una vez instalado se modificaron las reglas de permisos para denegar cualquier intento de conexión por el puerto 80 a direcciones externas.

- sudo ufw deny 80
- sudo ufw deny 80/tcp
- sudo ufw allow from 127.0.0.1 to any port 80

```
debian@debian:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     DENY IN     Anywhere
[ 3] 443                        ALLOW IN    Anywhere
[ 4] 80                         DENY IN     Anywhere
[ 5] 80                         ALLOW IN    127.0.0.1
[ 6] 21/tcp                     ALLOW IN    Anywhere
[ 7] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] 80/tcp (v6)                DENY IN     Anywhere (v6)
[ 9] 443 (v6)                   ALLOW IN    Anywhere (v6)
[10] 80 (v6)                    DENY IN     Anywhere (v6)
[11] 21/tcp (v6)                ALLOW IN    Anywhere (v6)
```

A continuación desde la máquina Kali se escaneó el puerto 80 para comprobar la configuración, apareciendo "filtered" en el servicio.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p 80 192.168.1.94
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-08 17:14 EST
Nmap scan report for debian.home (192.168.1.94)
Host is up (0.00057s latency).

PORT   STATE    SERVICE
80/tcp filtered http
MAC Address: 08:00:27:55:24:83 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.74 seconds
```

Observaciones y acciones tomadas

Durante el desarrollo de las tareas de análisis y pruebas de seguridad sobre el sistema Debian, se llevaron a cabo acciones de reconocimiento, explotación controlada y endurecimiento de servicios con el objetivo de reducir la superficie de ataque sin interrumpir el funcionamiento de los servicios activos.

Inicialmente, se realizó un escaneo de puertos y servicios desde una máquina Kali Linux, identificándose varios servicios expuestos, entre ellos el servicio FTP en el puerto 21 y el servicio HTTP Apache en el puerto 80. Ambos servicios se encontraban accesibles desde la red, lo que motivó un análisis más detallado de su configuración.

En el caso del servicio FTP, se detectó una mala configuración que permitía el acceso anónimo sin necesidad de autenticación. Esta vulnerabilidad fue confirmada mediante una conexión exitosa utilizando el usuario “anonymous”, desde la cual fue posible ejecutar

comandos básicos de enumeración. Como medida correctiva, se modificó la configuración del servidor FTP para deshabilitar el acceso anónimo, manteniendo el servicio operativo únicamente para usuarios autenticados. Tras aplicar los cambios y reiniciar el servicio, se verificó desde la máquina atacante que el acceso anónimo ya no era posible, mitigando así la vulnerabilidad detectada.

Respecto al servicio Apache, se observó que el servidor web estaba activo y accesible desde la red, exponiendo información del sistema que podía facilitar tareas de reconocimiento por parte de un atacante. Para reducir este riesgo, se aplicaron medidas de endurecimiento a nivel de configuración, tales como la ocultación de información del servidor y la limitación de los datos expuestos en las respuestas HTTP. Estas acciones permitieron minimizar la información revelada sin deshabilitar el servicio ni cerrar el puerto correspondiente, manteniendo la disponibilidad del servidor web.

Aunque tras la aplicación de estas medidas los puertos continuaron apareciendo como abiertos en los escaneos de red, se considera que el riesgo fue mitigado mediante el endurecimiento de las configuraciones, reduciendo significativamente el impacto potencial de una explotación. Este enfoque permite mejorar la seguridad del sistema manteniendo la funcionalidad de los servicios expuestos.

En conclusión, las acciones tomadas se centraron en corregir configuraciones inseguras y fortalecer los servicios existentes, logrando un equilibrio entre seguridad y disponibilidad, y reduciendo la probabilidad de explotación de vulnerabilidades conocidas.