**Irked walkthrough**

# **Index**

[Index 1](#_Toc181737791)

[List of pictures 1](#_Toc181737792)

[Disclaimer 2](#_Toc181737793)

[Reconnaissance 2](#_Toc181737794)

[Initial foothold 2](#_Toc181737795)

[User flag 3](#_Toc181737796)

[Privilege escalation 5](#_Toc181737797)

[Personal comments 6](#_Toc181737798)

[Appendix A – CVE-2010-2075 6](#_Toc181737799)

[Appendix B – Steganography 6](#_Toc181737800)

[References 6](#_Toc181737801)

# **List of pictures**

[Figure 1 - nMap scan results 2](#_Toc181737781)

[Figure 2 - nMap IRC scripts 3](#_Toc181737782)

[Figure 3 - IRC exploited 3](#_Toc181737783)

[Figure 4 - Password found 4](#_Toc181737784)

[Figure 5 - steghid tool to extract data from images 4](#_Toc181737785)

[Figure 6 - Lateral movement 4](#_Toc181737786)

[Figure 7 - User flag 5](#_Toc181737787)

[Figure 8 - Information to escalate privileges 5](#_Toc181737788)

[Figure 9 - Privilege escalation 5](#_Toc181737789)

[Figure 10 - Root flag 5](#_Toc181737790)

# **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:

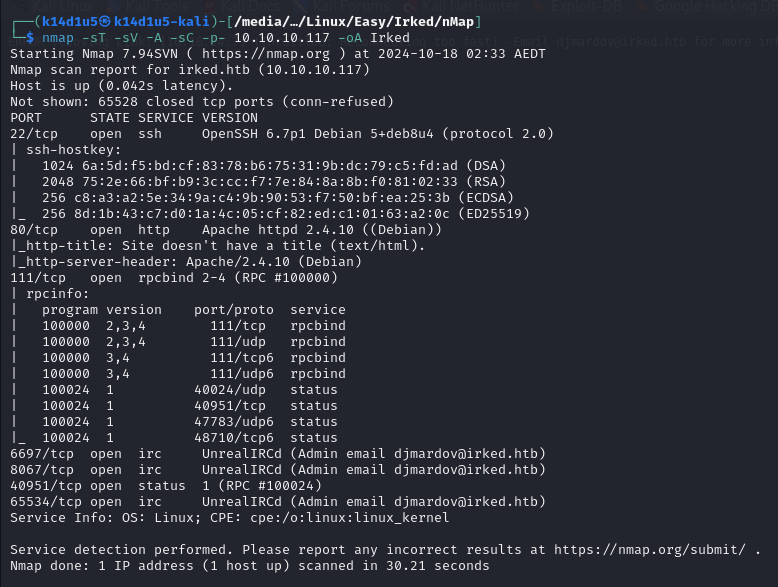


Figure 1 - nMap scan results

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:

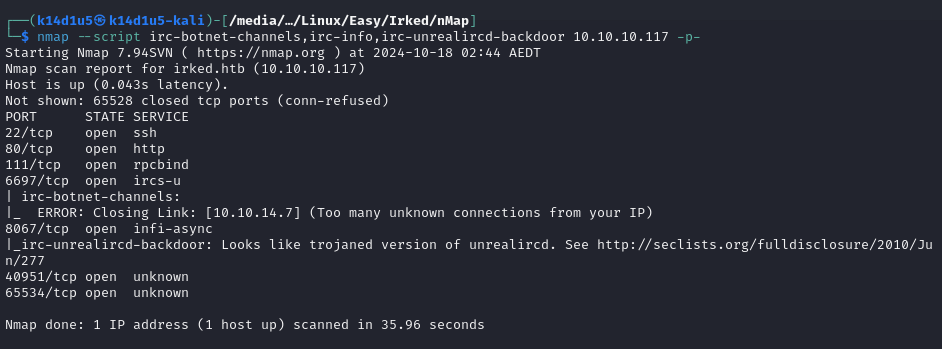


Figure 2 - nMap IRC scripts

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:

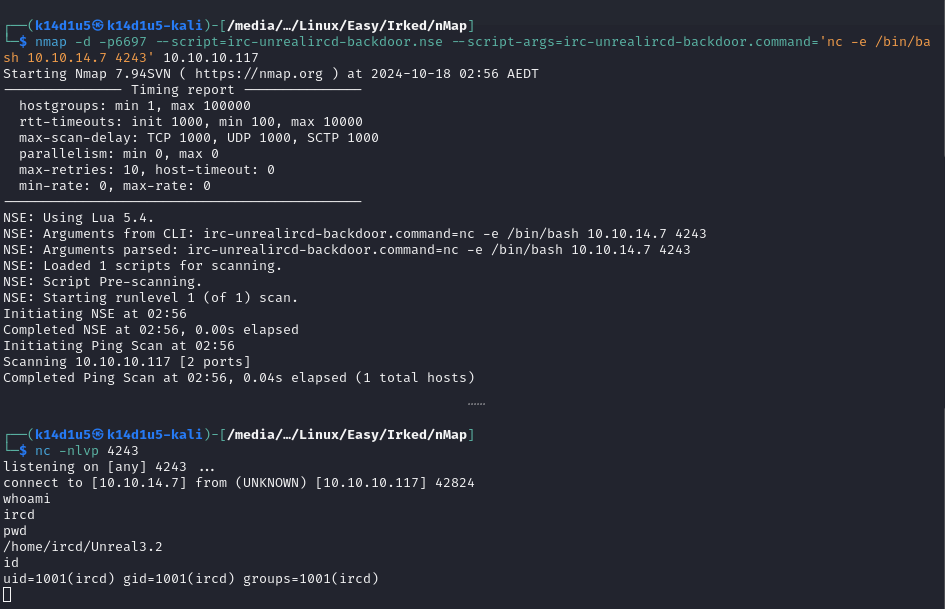


Figure 3 - IRC exploited

# **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 file in the path. I tried to read this file and I found out a password, as shown in the following:

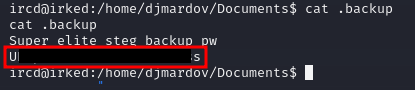


Figure 4 - Password found

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 tool to extract some information from the image, as shown in the following:

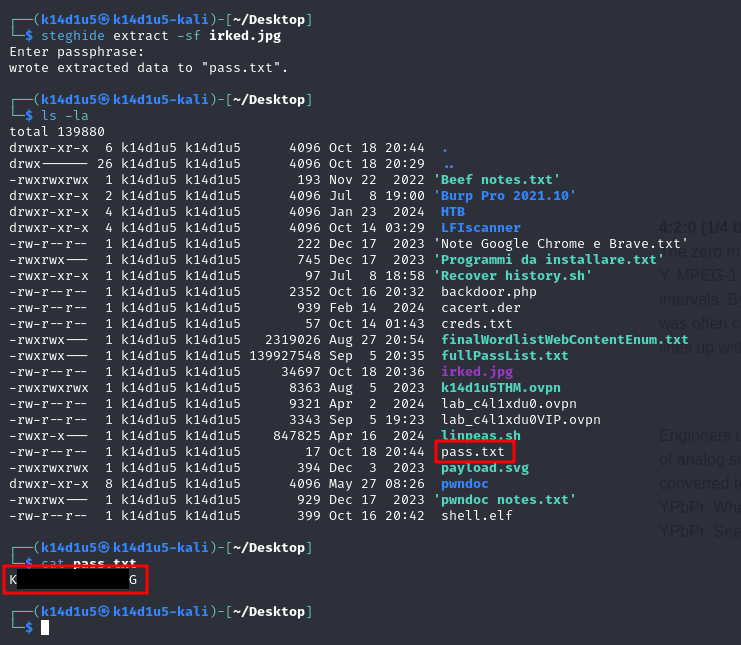


Figure 5 - steghid tool to extract data from images

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

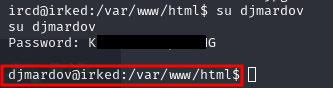


Figure 6 - Lateral movement

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

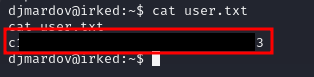


Figure 7 - User flag

# **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 binary had an unknown SUID settings. So, I investigate more on it and I run it:

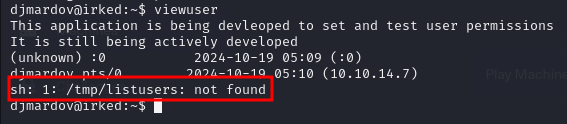


Figure 8 - Information to escalate privileges

It was very interesting. This binary tried to use a file named 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” file and tried to run the program:

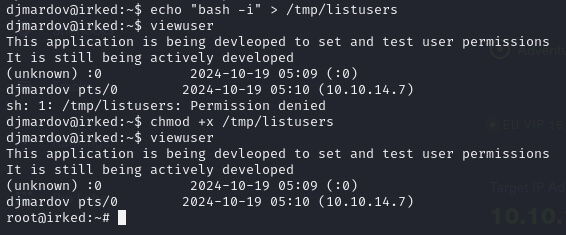


Figure 9 - Privilege escalation

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

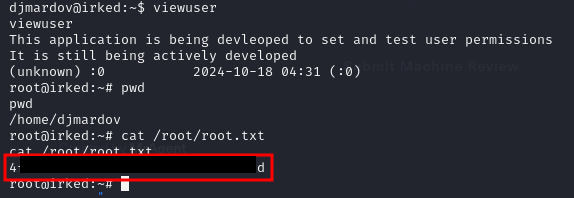


Figure 10 - Root flag

# **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 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