Penetration Testing Group Project

Yogi Makadiya
Manarth Bhavsar
Akhil Jain

Table of Contents

Table of Figures

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:**

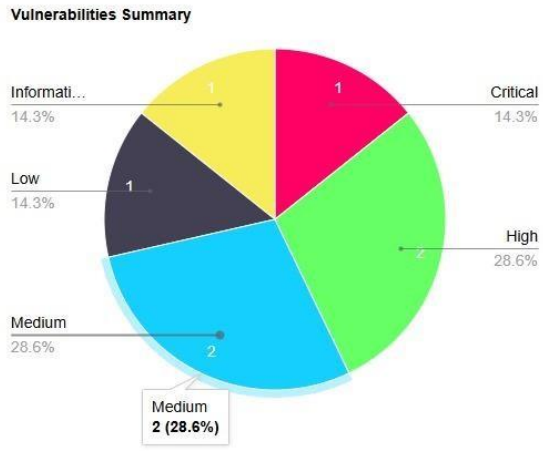

Figure 1: Vulnerability Summary According to Criticality

- **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.

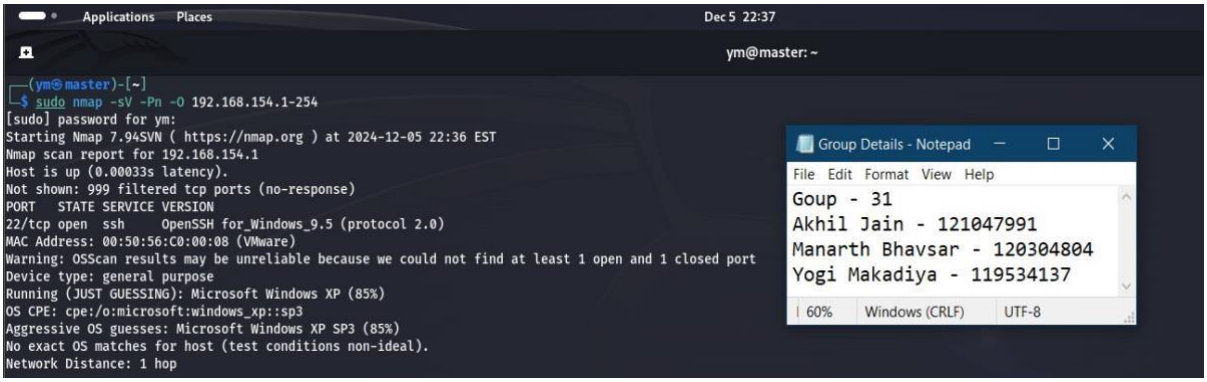

Figure 3: Scanning for IP Address Recon

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:

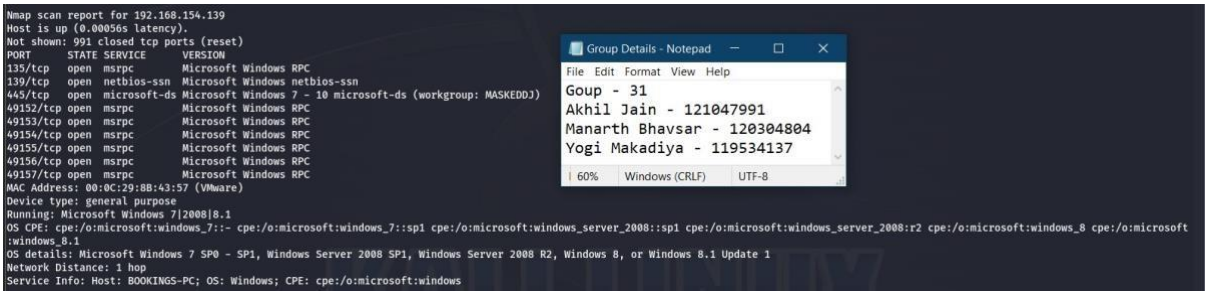

Figure 6: IP Address and Open Ports of Windows 7 Machine

IP address of Windows Server Running – 192.168.154.140:

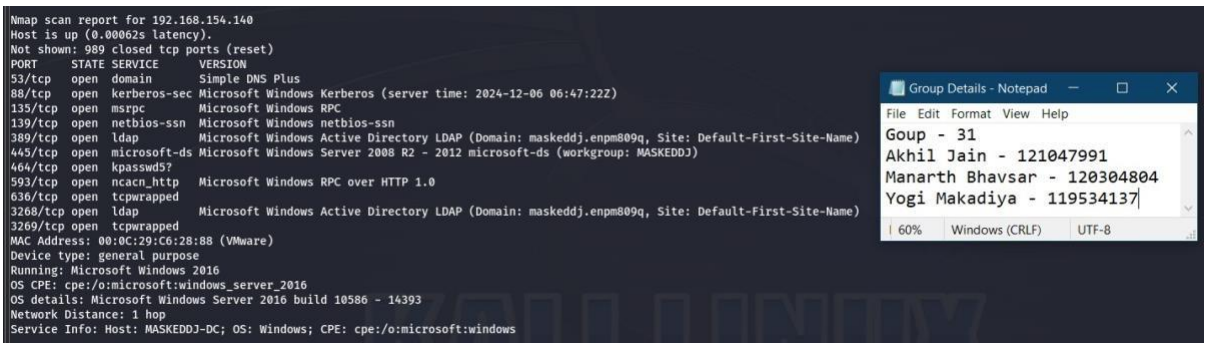

Figure 7: IP Address and Open Ports of Windows Server

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.

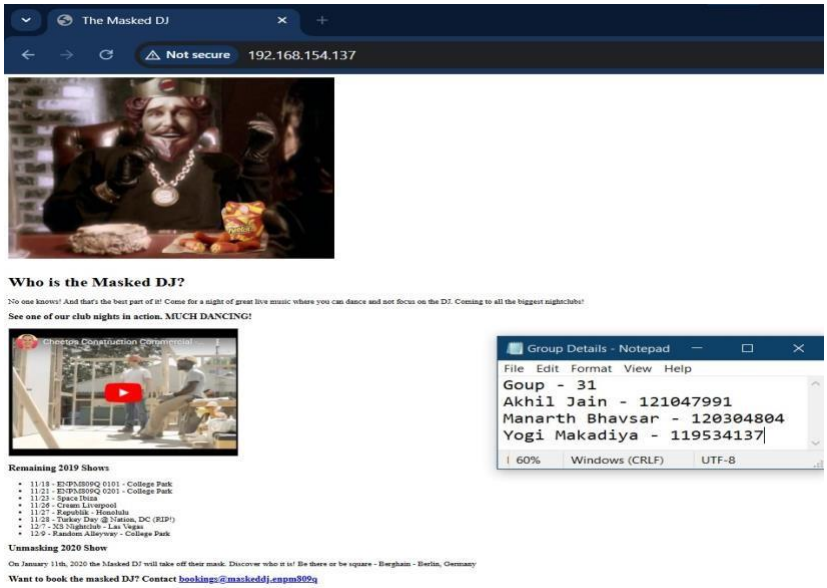

Figure 8: Homepage of the Website on Ubuntu Machine

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”.

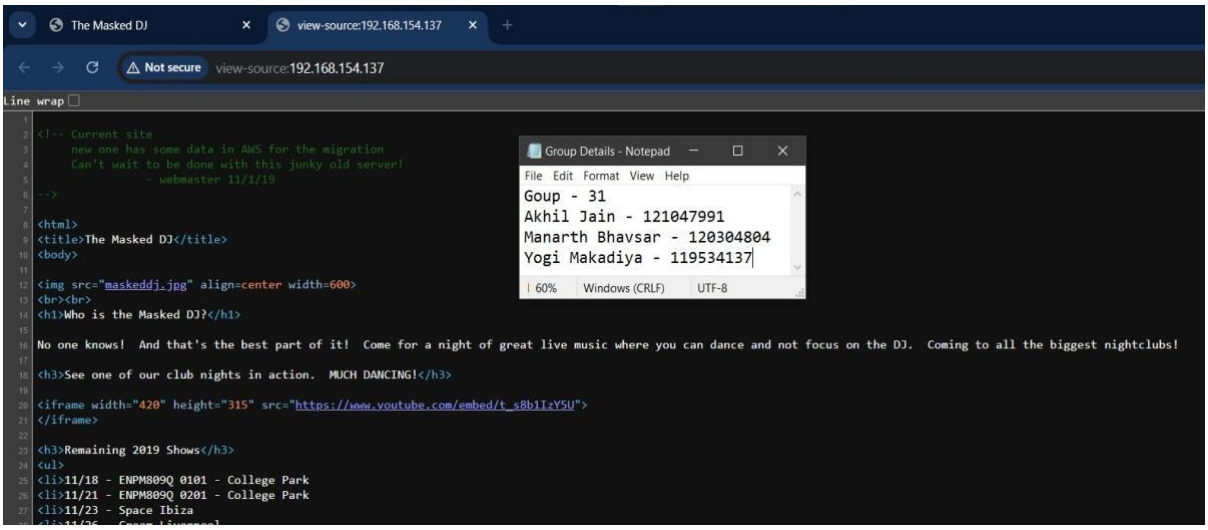

Figure 9: Page Source Code of the Homepage

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.

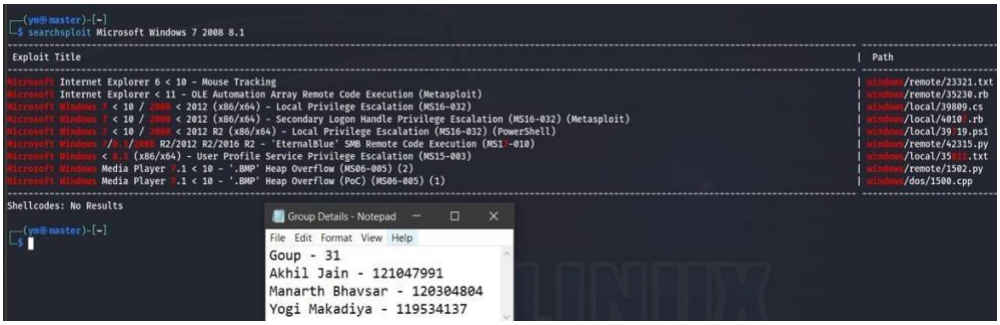

Figure 10: Searching Exploit for Windows 7 Machine

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.

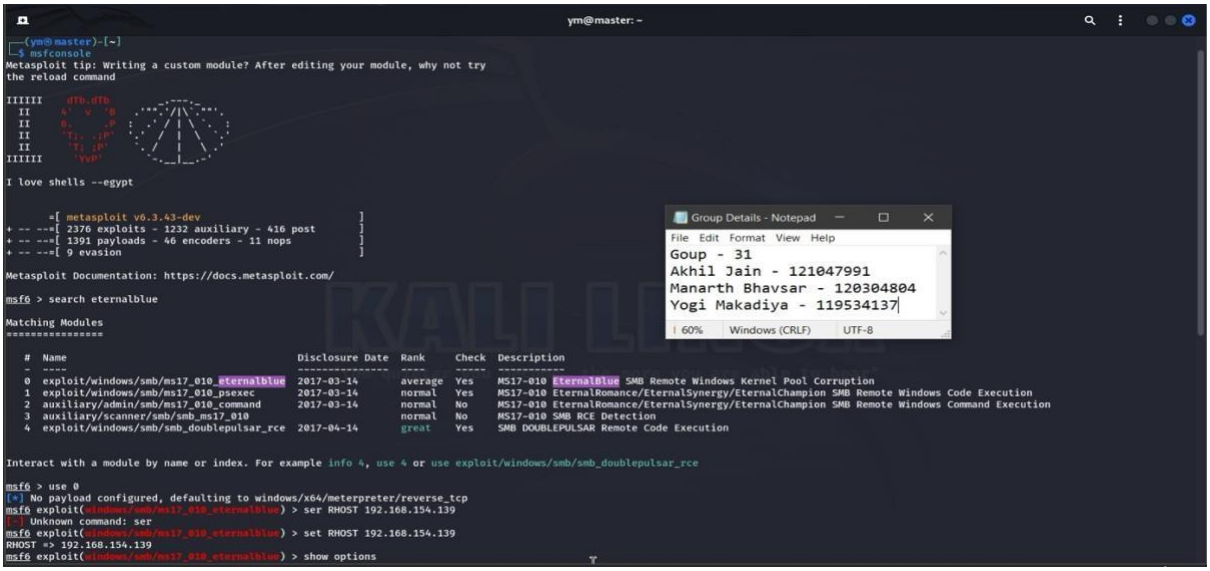

Figure 11: Metasploit used for the launch of attack

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

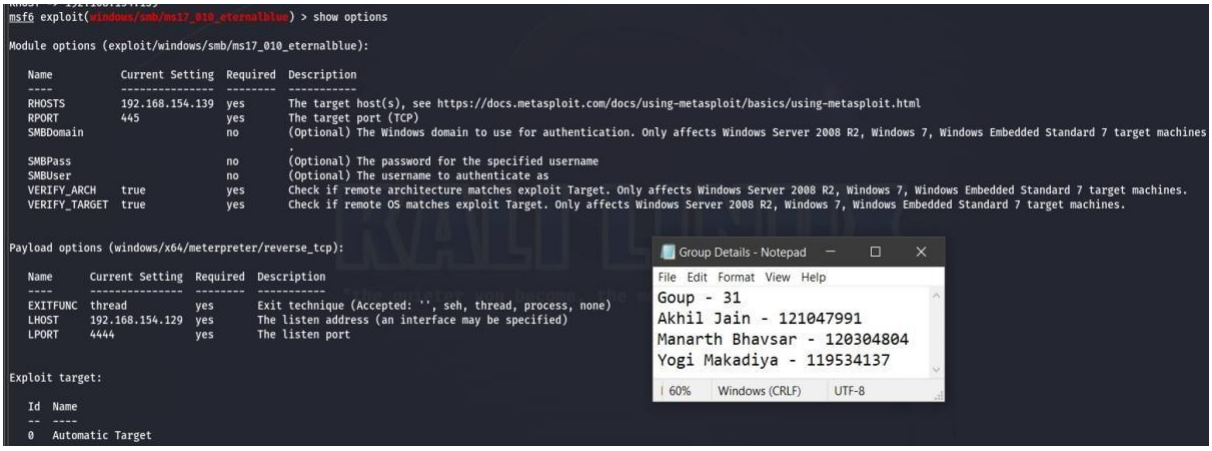

Figure 12: Configuration of exploit using "show options" command

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.

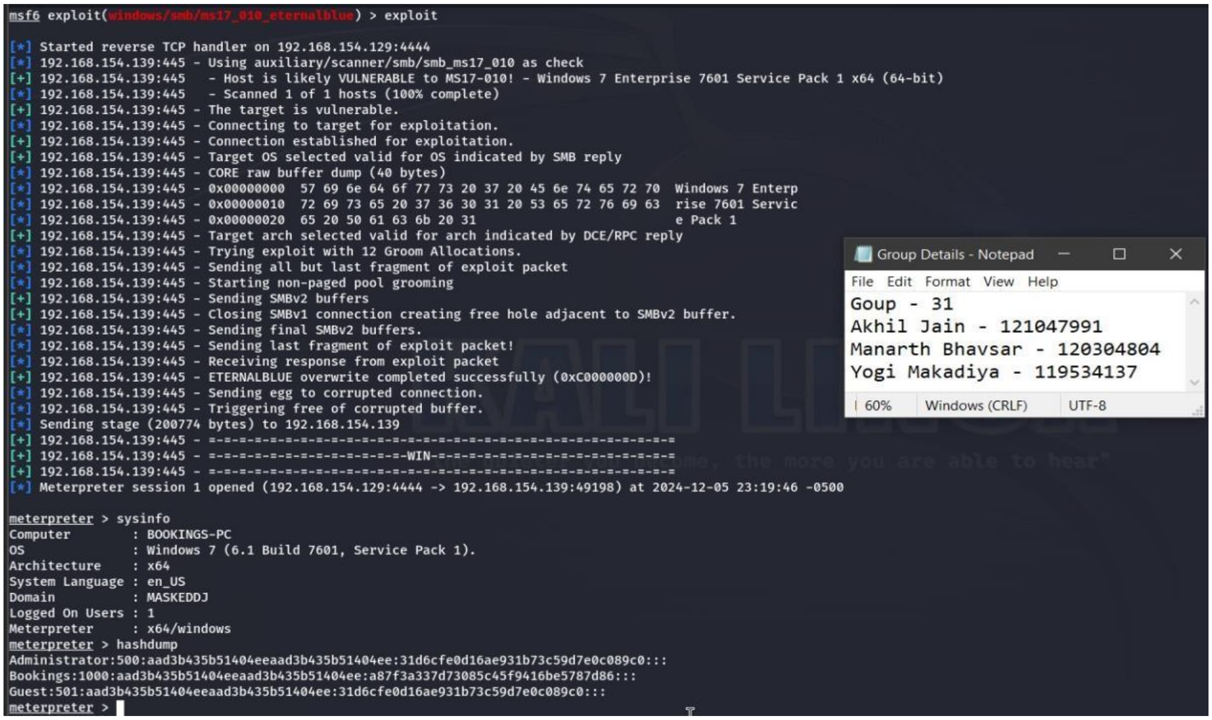

Figure 13: Launched attack on Windows 7 machine

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".

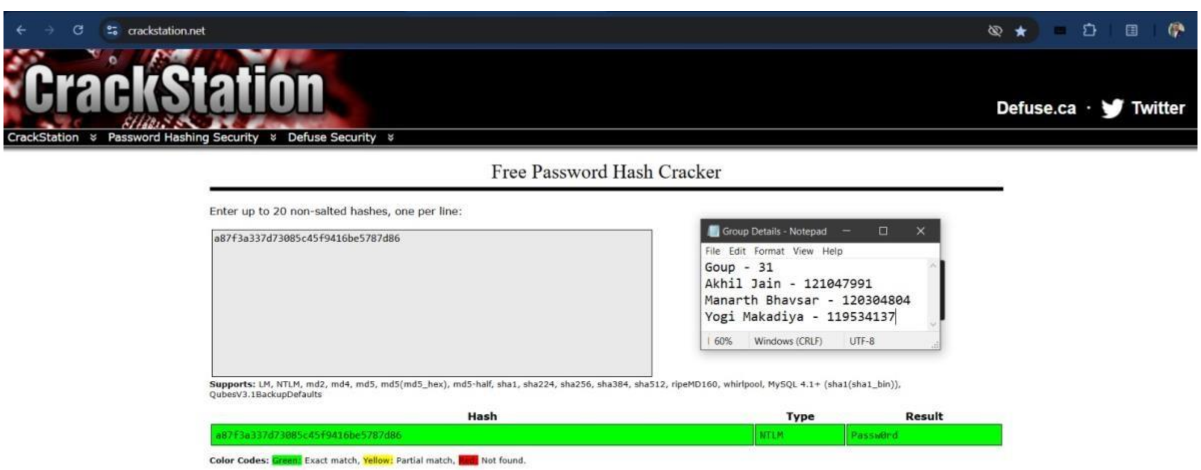

Figure 14: Password for "Bookings" account found using CrackStation

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) form 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.

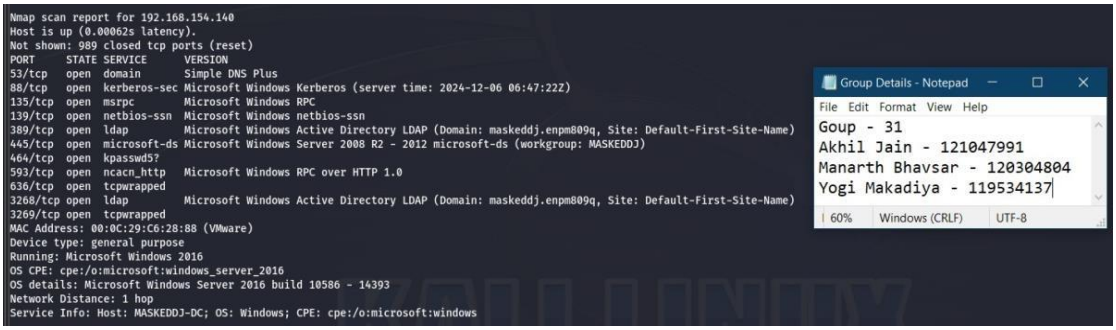

Figure 15: Open Ports of Windows Server Machine

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.

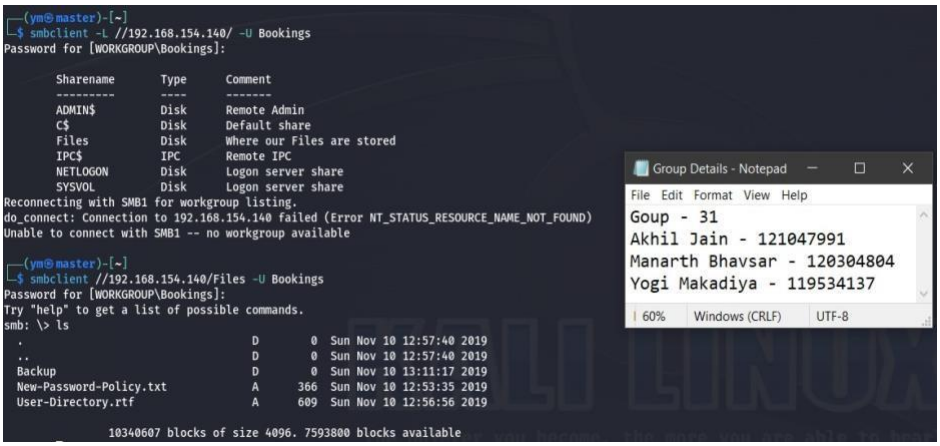

Figure 16: Listing all available services and accessing "Files"

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.

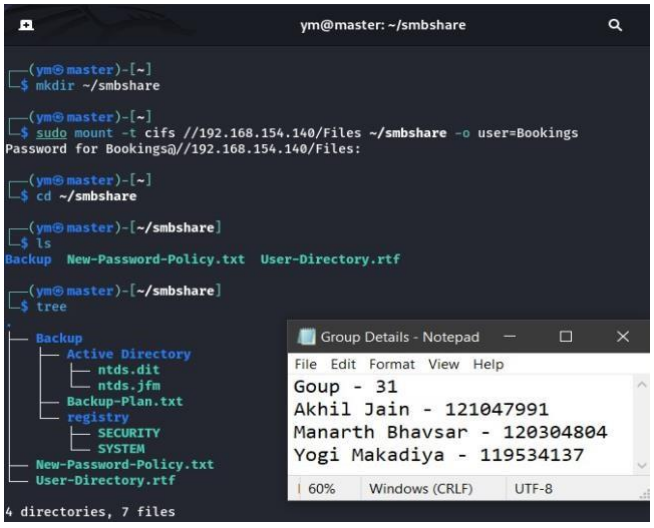

Figure 17: Mounted "Files" locally in "smbshare"

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.

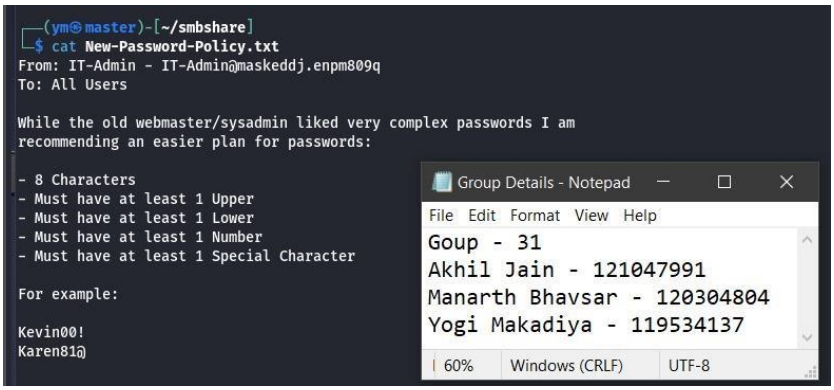

Figure 18: New-Password-Policy.txt

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.

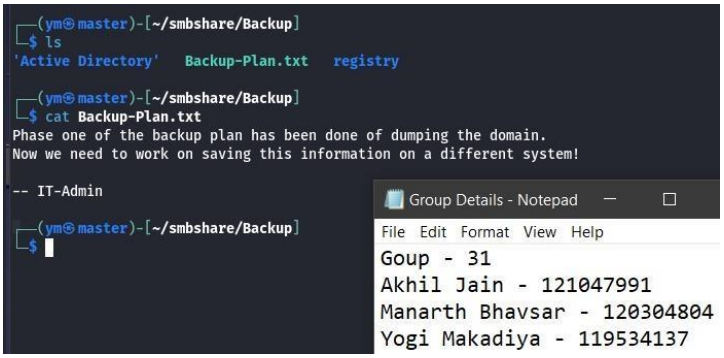

Figure 19: Accessing Backup-Plan.txt

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.

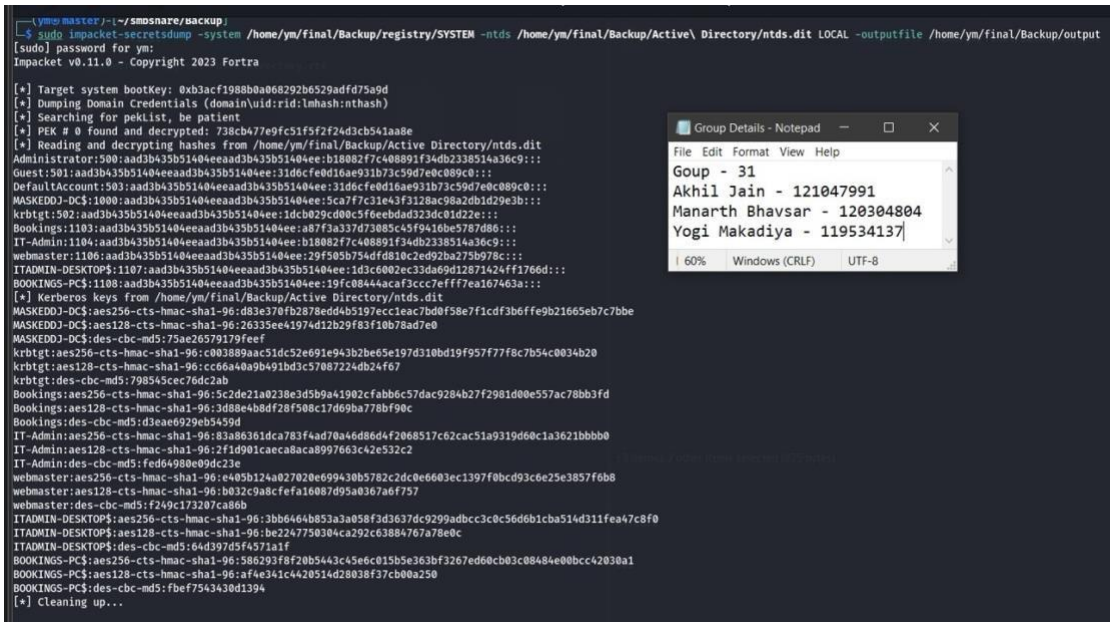

Figure 20: Extracting Information from ".dit" file

e. Cracking Hashes to Retrieve Passwords:

After extracting the hashed credentials using impacket-secretsdump, efforts were made to crack these hashes to retrieve plaintext passwords. The first attempt involved using hashcat, a powerful password recovery tool, to crack the hashes stored in the output.ntds file. The command used was “hashcat -m 1000 -a 3 output.ntds”. The -m 1000 specifies the hash type as NTLM, and -a 3 defines a brute-force attack mode. Despite running for hours, this method did not yield any meaningful results.

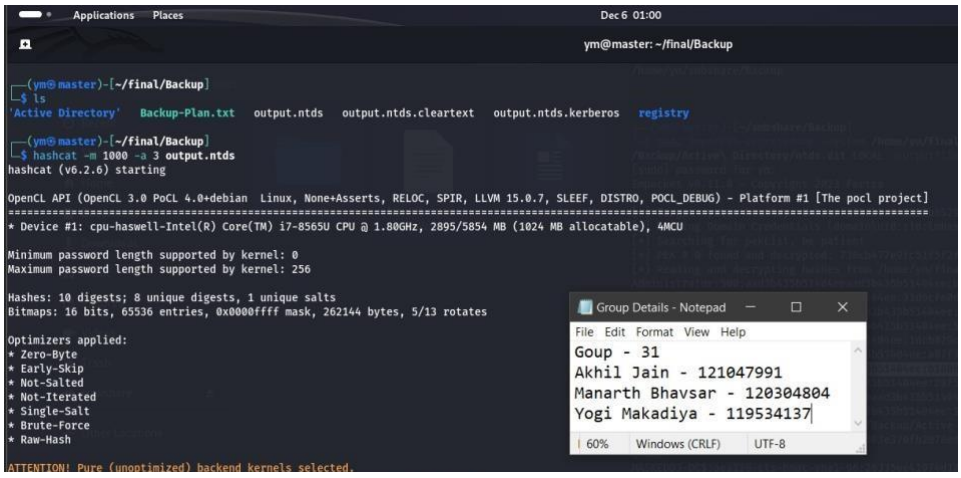

Figure 21: Attempt to crack hashes using hashcat

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.

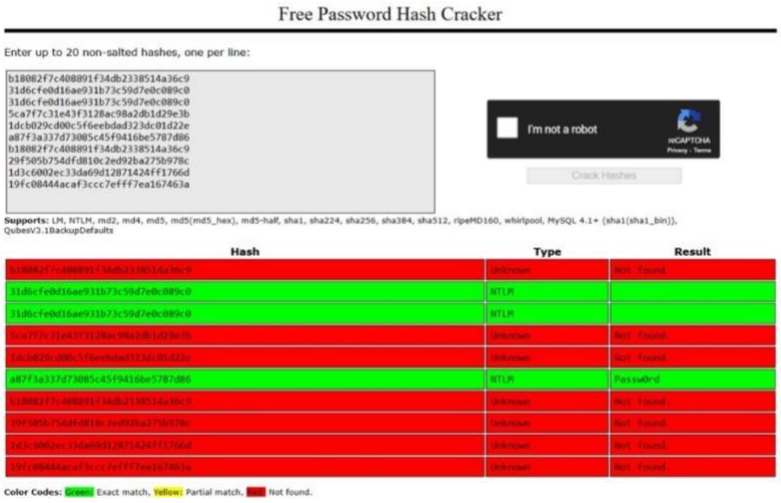

Figure 22: Attempt to crack hashes using CrackStation

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.

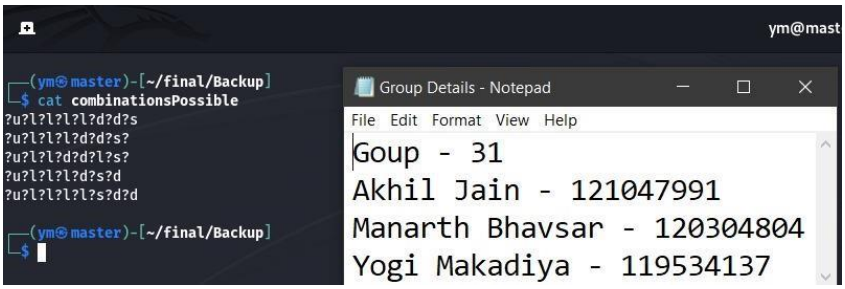

Figure 23: Combinations based on "New-Password-Policy.txt" file

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

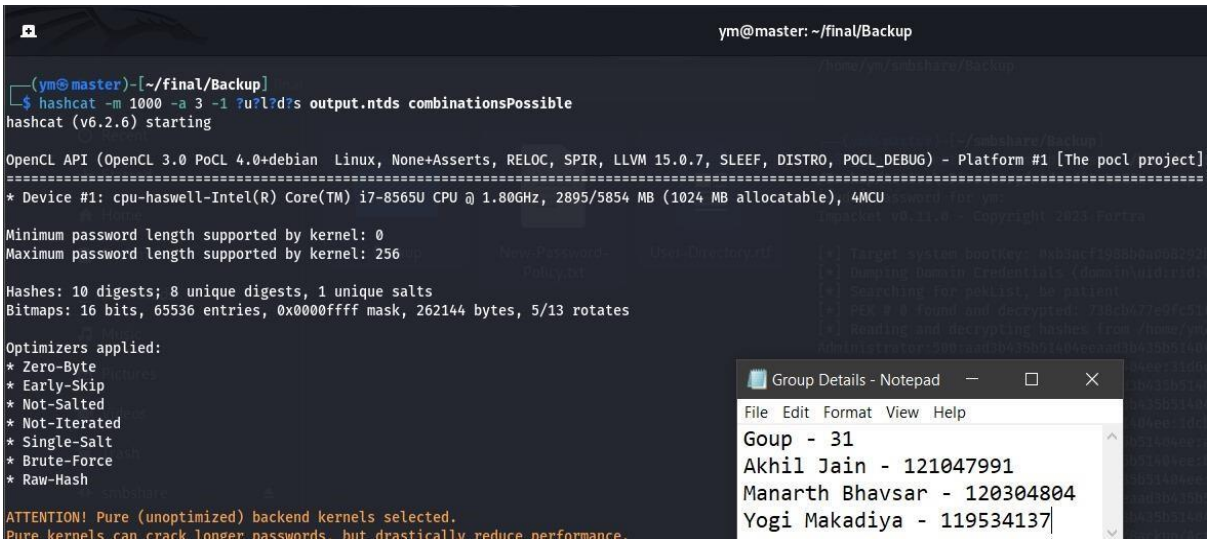

Figure 24: Cracking hashes using combinations and hashcat

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

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.

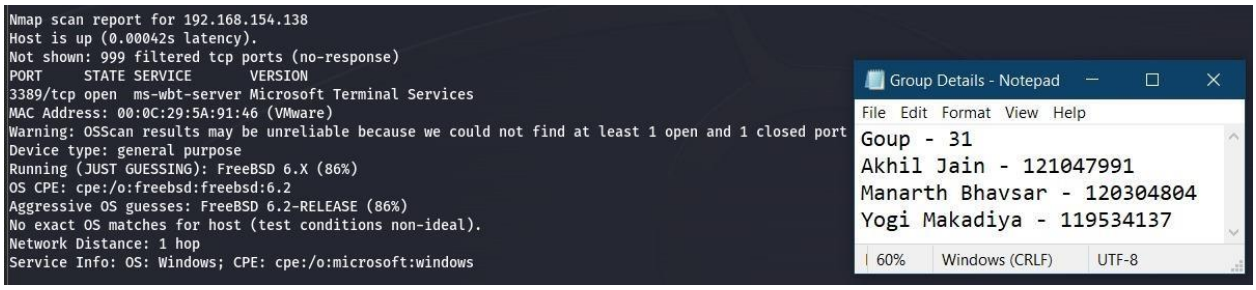

Figure 26: Open Ports Information of Windows 10 Machine

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.

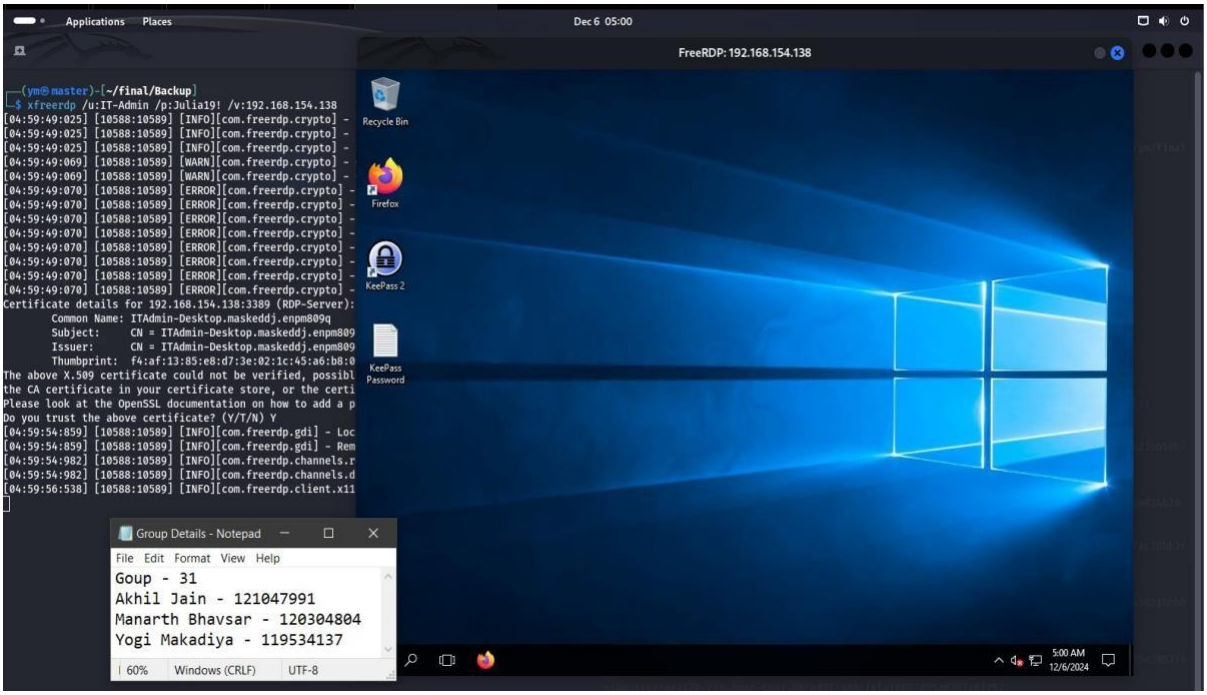

Figure 27: Getting RDP session using "xfreerdp"

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".

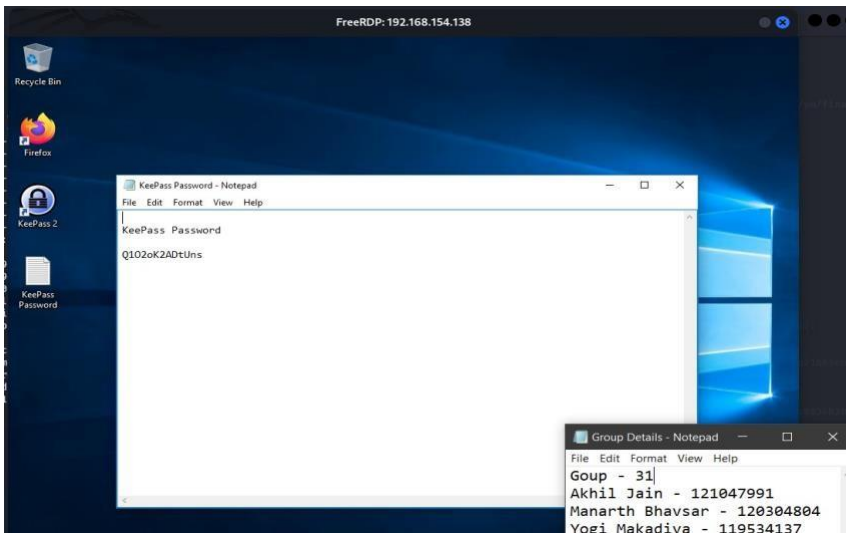

Figure 28: Found password for KeePass application

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%&".

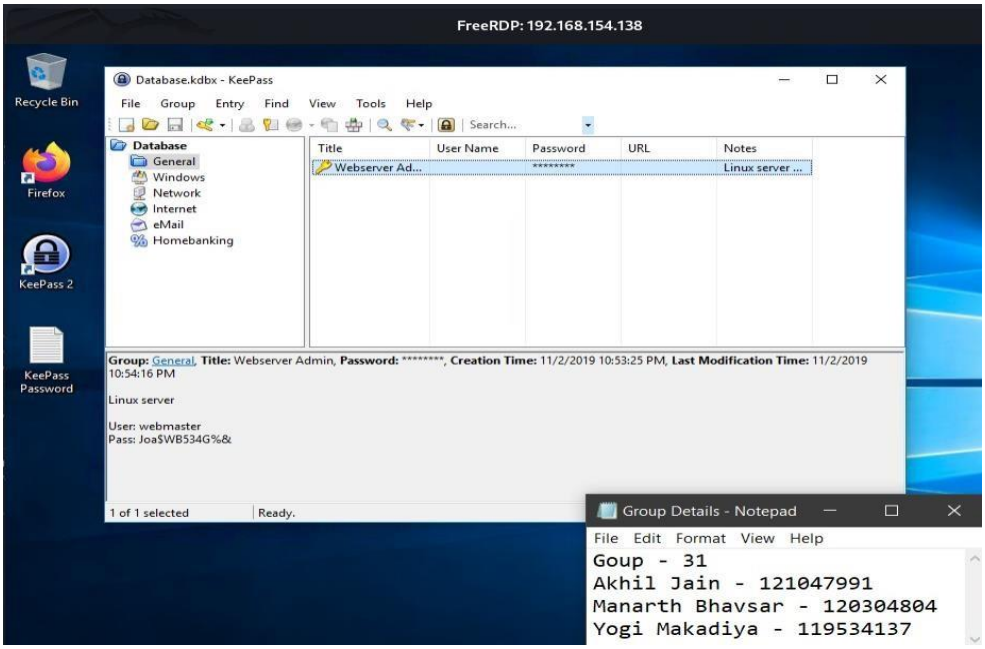

Figure 29: Found credentials for Ubuntu Machine

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.

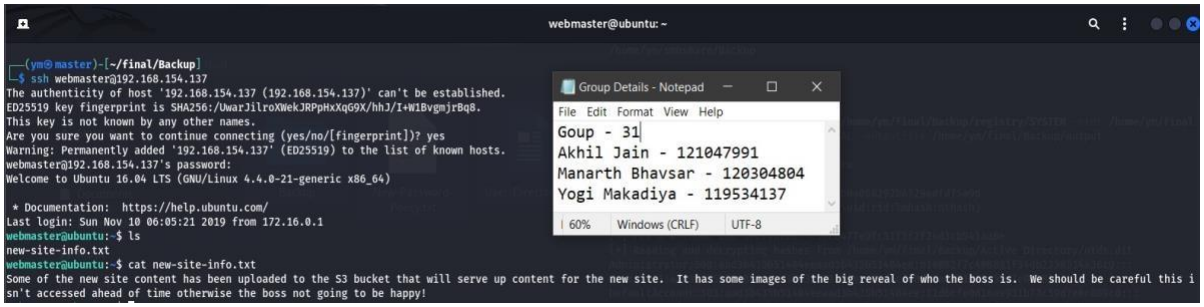

Figure 30: Getting into the Ubuntu machine and accessing "new-site-info.txt"

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).

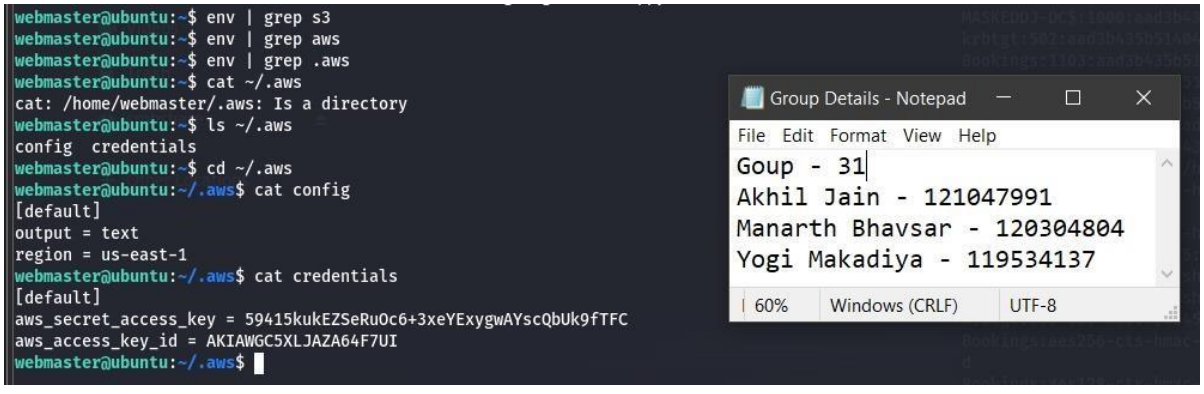

Figure 31: Accesing ".aws" directory and reading contents

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.

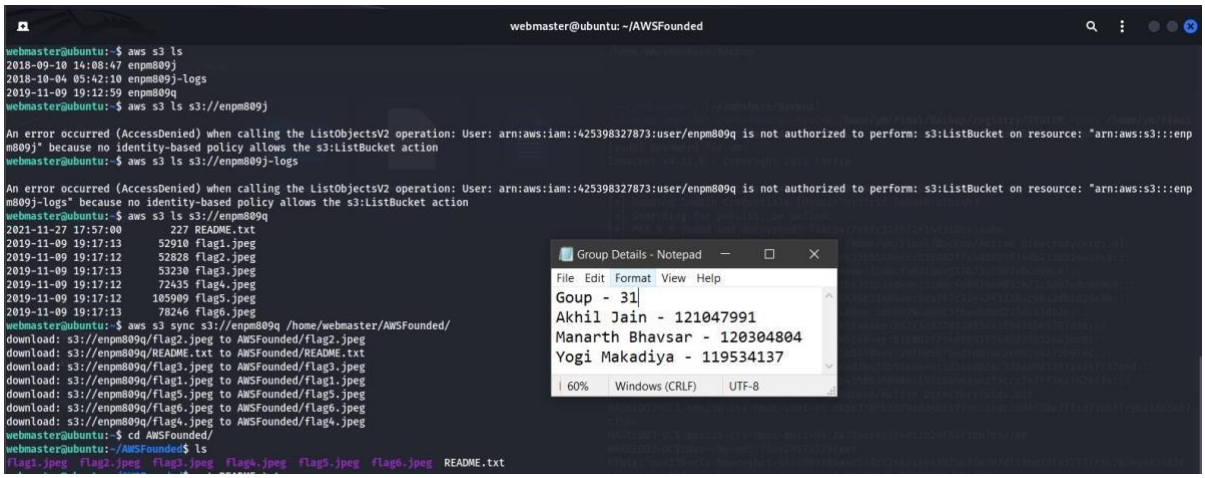

Figure 32: Getting Bucket Content into the "AWSFounded" directory

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.

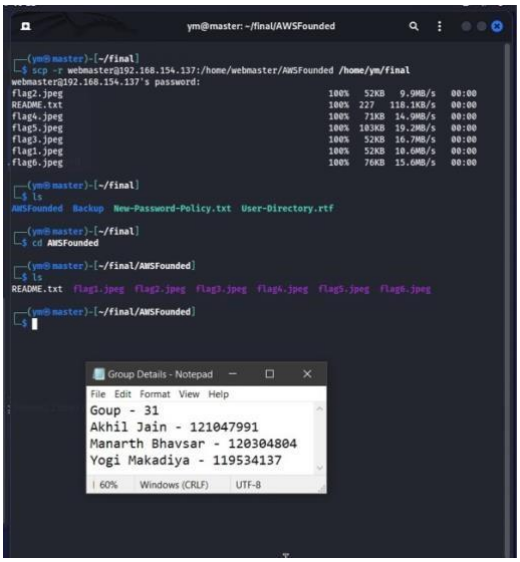

Figure 33: Getting founded contents to local machine

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.

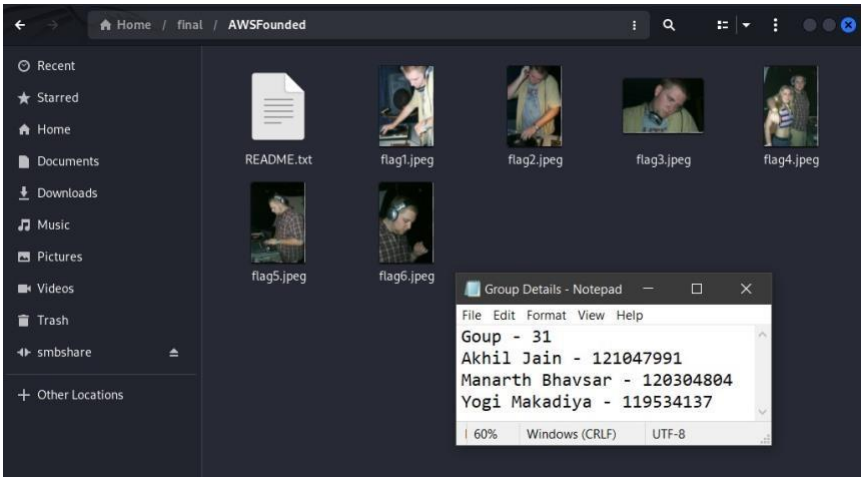

Figure 34: Files Found in S3 Bucket

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.

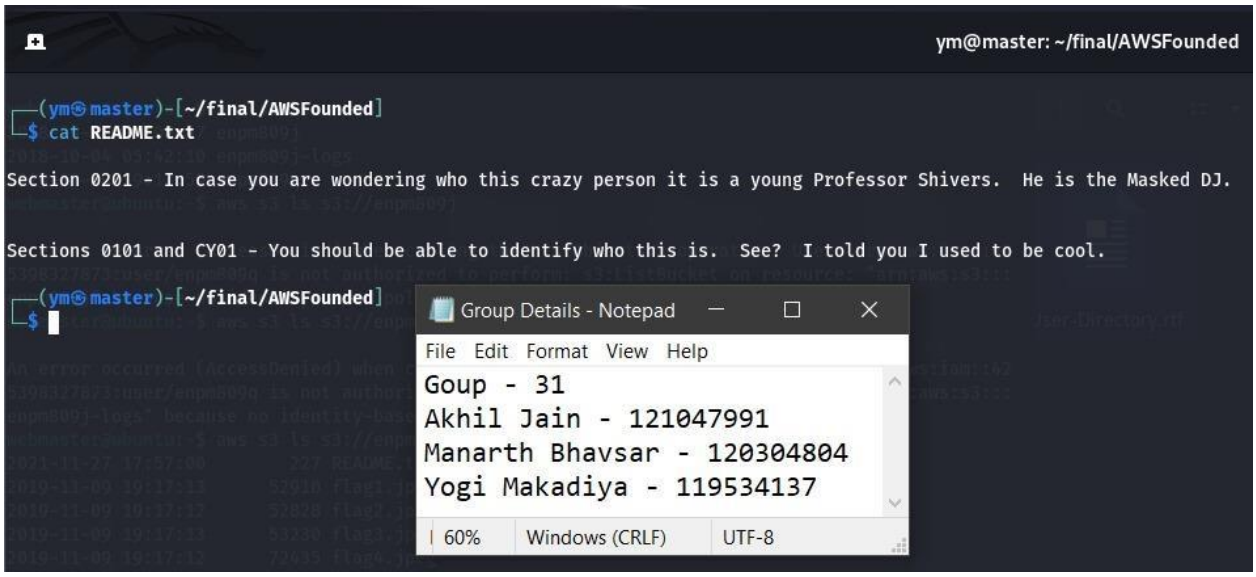

Figure 35: Message Found with flags in "README.txt"

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

Figure 36: flag1.jpeg

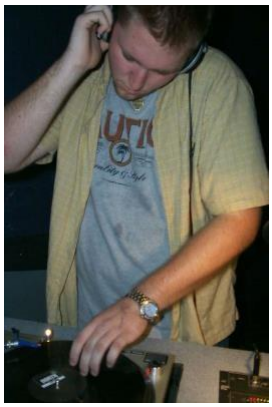

Figure 37: flag2.jpeg

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:** o 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:** o 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