Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

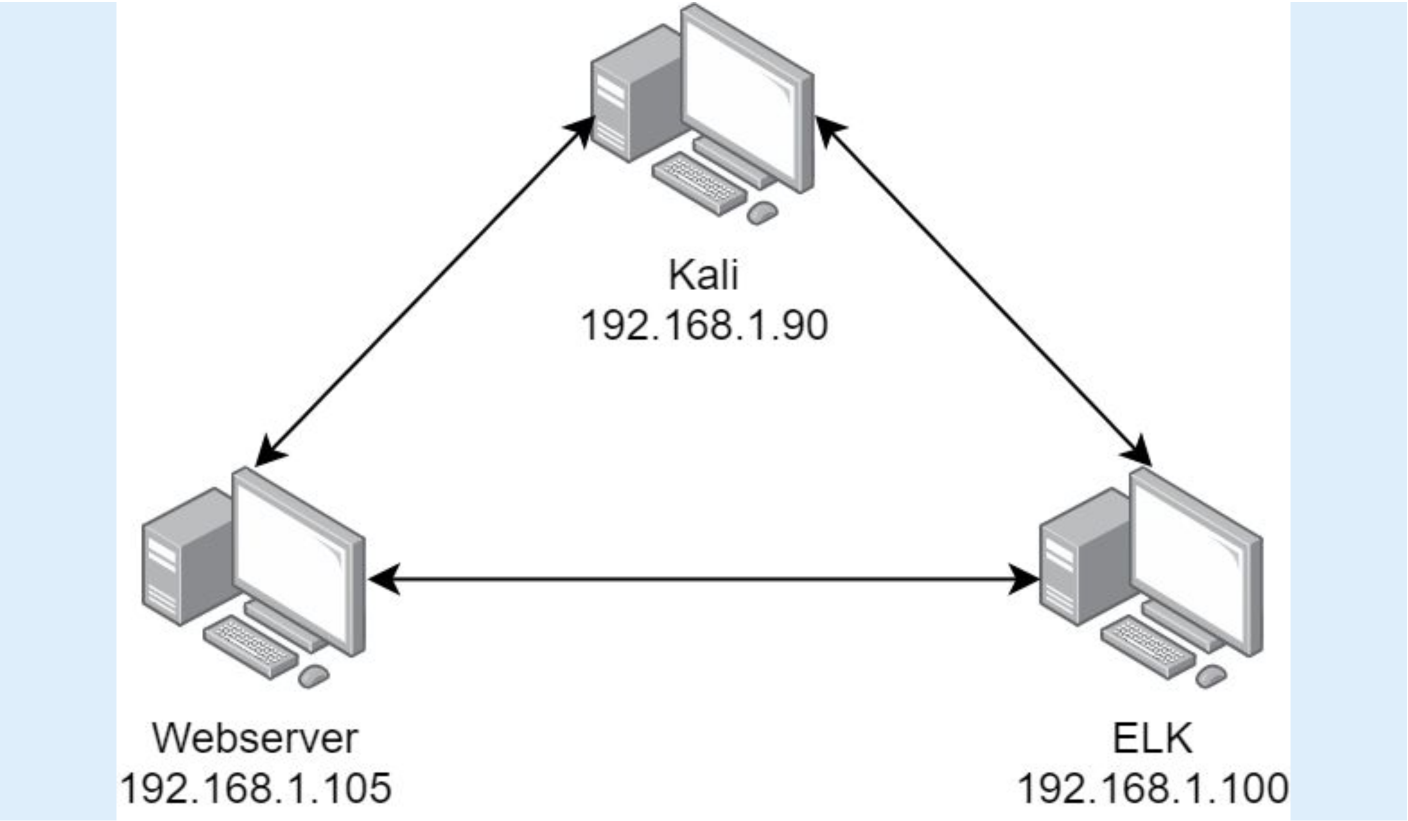
02 **Red Team**: Security Assessment

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:
Hostname: Gateway

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Webserver

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Gateway	192.168.1.1	Network Gateway
Kali	192.168.1.90	Attacker Machine
ELK	192.168.1.100	Logging
Webserver	192.168.1.105	Target Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Brute force	*An attacker can flood the server with login request using different passwords until they find a password that works.*	*This should be considered a severe vulnerability, as it is relatively easily mitigated, and can allow attackers to gain potentially any level of access.*
Weak cryptographic functions	A password found on the web server was hashed using MD5, which is outdated and vulnerable	This should be considered a critical vulnerability due to the privileges an attacker could gain after decrypting the hashed password.
Remote code execution	A php file was able to be uploaded onto the server, which when executed allowed a reverse shell connection.	This should be considered a critical vulnerability as it allowed for reverse shell access to the webserver.

Exploitation: Brute Force

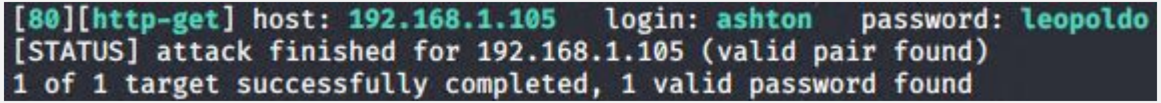
01

Tools & Processes: I used Hyda to target the secret_folder login page using aston's username and the rockyou.txt wordlist.

02

Achievements: I was able to gain access to Ashton's user account and use it to access secret_folder.

03



```
[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
```

Exploitation: Weak Cryptographic Functions

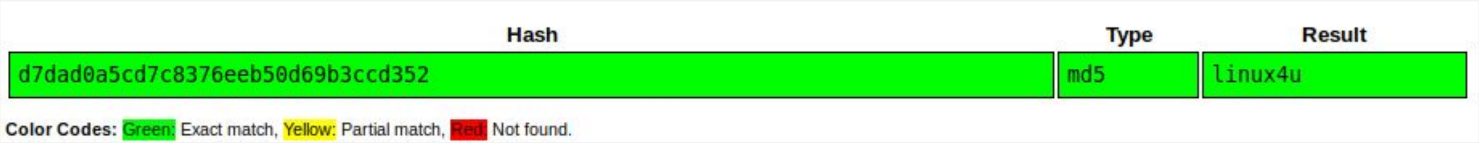
01

Tools & Processes: Once I had access to the connect_to_corp_server file, I used crackstation to decrypt the MD5 hash found in it.

02

Achievements: I was able to access Ryan’s account, which allowed me to access the webdav folder with both read and write privileges.

03



Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Exploitation: Remote Code Execution

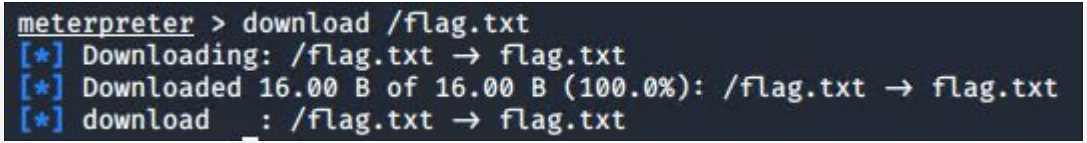
01

Tools & Processes: Using msfvenom I made a shell.php file which I uploaded to webdav, and was able to gain a meterpreter reverse shell after executing the shell.php file.

02

Achievements: Once I had reverse shell access I was able to search the target machine and download the flag.

03

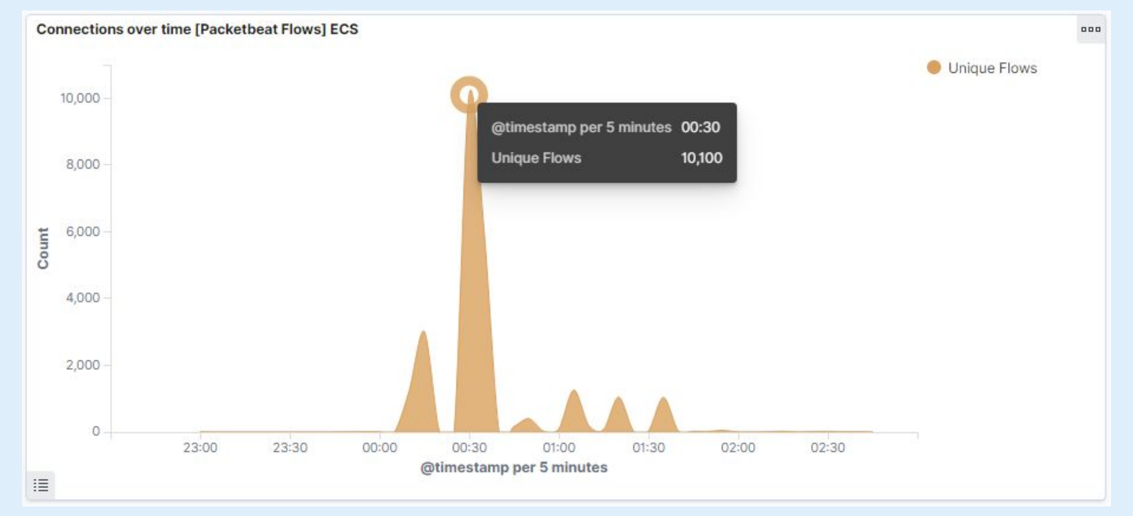


```
meterpreter > download /flag.txt
[*] Downloading: /flag.txt → flag.txt
[*] Downloaded 16.00 B of 16.00 B (100.0%): /flag.txt → flag.txt
[*] download   : /flag.txt → flag.txt
```

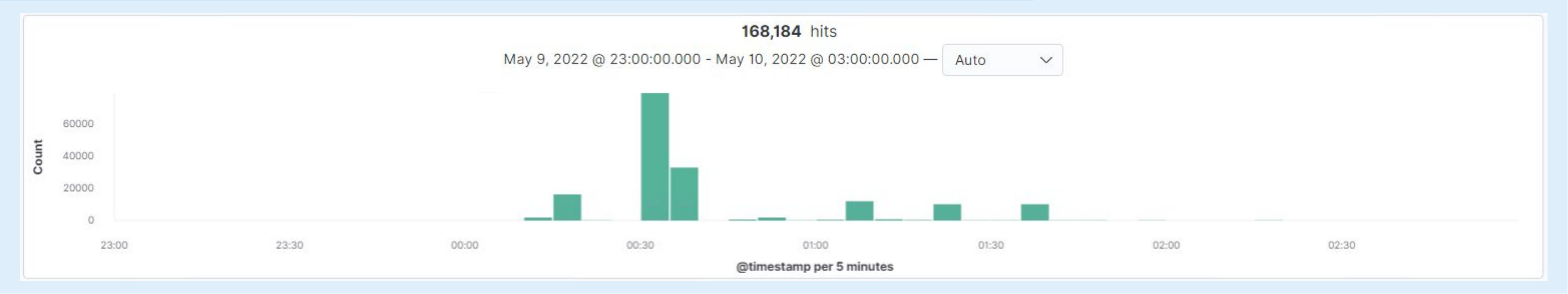
Blue Team

Log Analysis and Attack Characterization

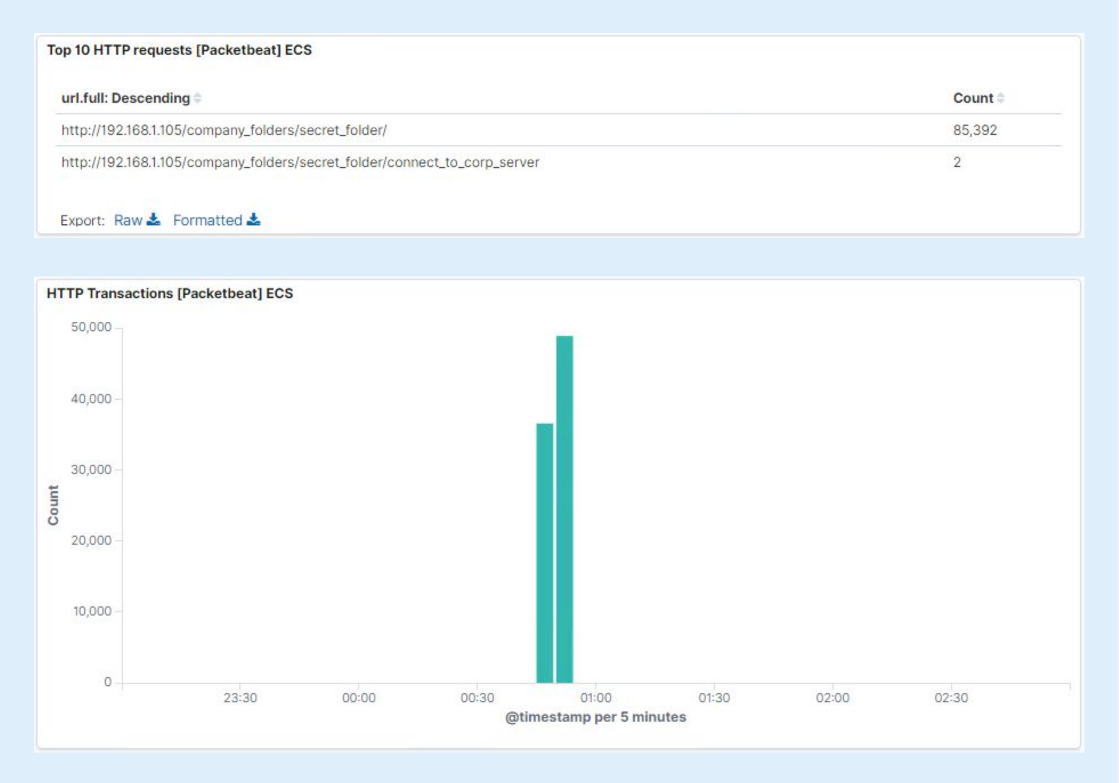
Analysis: Identifying the Port Scan



- The port scan occurred at 12:30 am on May 10th, with other network scanning happening before and after
- 10,100 packets were sent during the port scan, from 192.168.1.90
- We can detect the port scan by filtering for number of network packets, because all connections during this scan contained exactly 2 network packets



Analysis: Finding the Request for the Hidden Directory



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	85,392
http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server	2

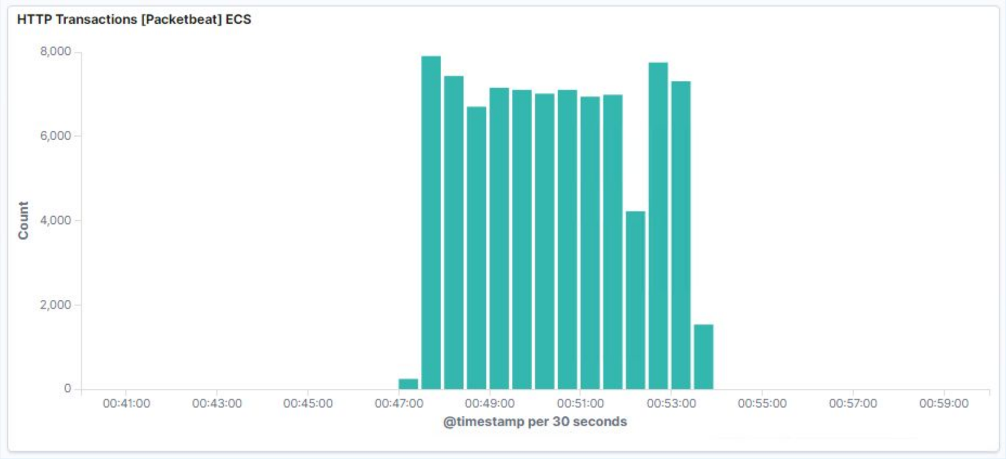
Export: Raw Formatted

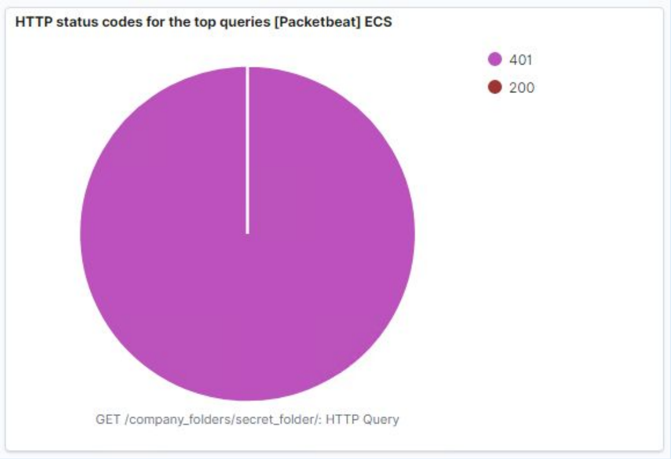
HTTP Transactions [Packetbeat] ECS

- 85,394 requests were made, all requests were from 192.168.1.90 between 12:40 am and 12:50 am only 6 requests did not receive an error code.

- The connect_to_corp_server file was requested. This file contains a hashed password to Ryan’s user account as well as directions to access the webdav folder.

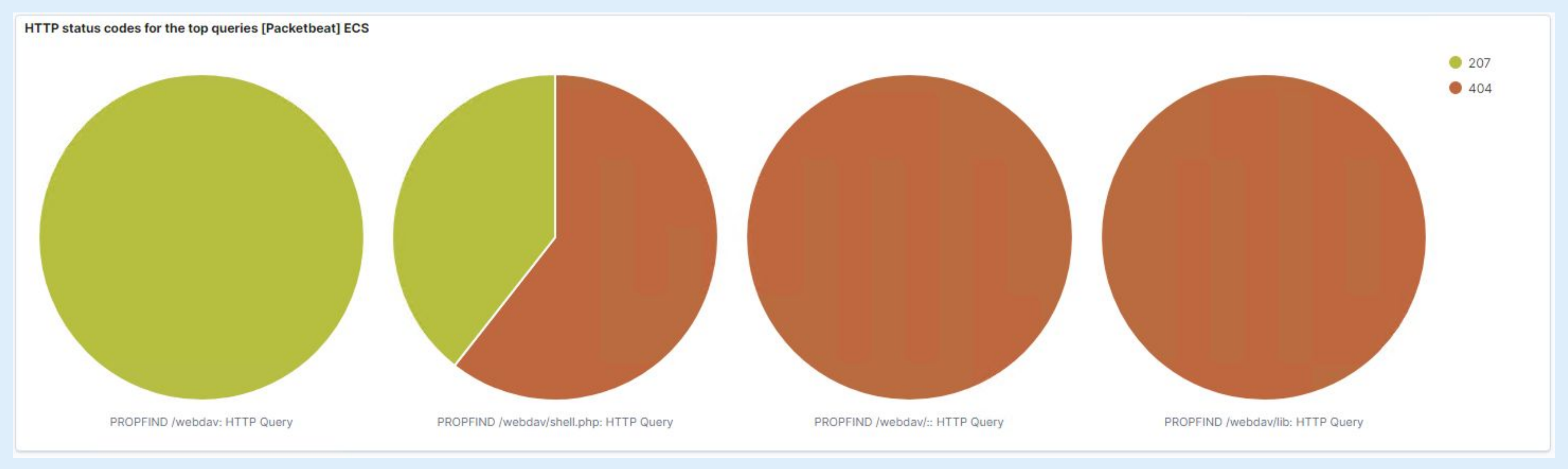
Analysis: Uncovering the Brute Force Attack


HTTP Transactions [Packetbeat] ECS


HTTP status codes for the top queries [Packetbeat] ECS

- 85,384 total requests were made during the brute force attack.
- 85,382 of those requests resulted in a 401 forbidden error

Analysis: Finding the WebDAV Connection



- 3,057 requests were made to /webdav/ 907 of which were successful, the rest resulted in errors.
- Due to the use of the davtest tool there were a large number of files requested. The most crucial files in this attack were the already existing file of passwd.dav, as well as the shell.php file which was uploaded and accessed.

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

Syn scans can be detected by checking for log data with the network.packets value of 2

The baseline does not go above about 4180, so I would put a threshold of 6000 on the alarm.

System Hardening

We can set a firewall rule to block connections to non-essential ports by default

```
sudo ufw default deny incoming
```

Mitigation: Finding the Request for the Hidden Directory

Alarm

As the secret folder contains very sensitive data we should set an alarm to alert any access from IPs that are not strictly authorized

A threshold of 1 would be appropriate

System Hardening

We can set up a firewall rule to block connections from unauthorized IPs or implement a company VPN and allow only connections from within the VPN

Whitelisting specific IP addresses or in network connections, while blocking all others, would be a strong mitigation

Mitigation: Preventing Brute Force Attacks

Alarm

We can monitor for 401 errors returned from secret_folder to monitor brute force attacks

As secret_folder should not be accessed often I would set an alarm at more than 50 failed connections

System Hardening

What configuration can be set on the host to block brute force attacks?
We can implement account lockouts after a threshold of failed login attempts

A more specific rule would be to lockout the account for 15 minutes after 10 failed attempts, and have counter reset after 15 minutes

Mitigation: Detecting the WebDAV Connection

Alarm

As webdav is not often accessed, we can set an alarm to monitor any connections to webdav

As we are monitoring any access to the webdav server the threshold would be 1

System Hardening

We could set a firewall rule to block access to webdav from outside of the company network or any not specifically whitelisted IP addresses

Implementing a company VPN and allowing only connections coming from within VPN would also potentially be a strong mitigation

Mitigation: Identifying Reverse Shell Uploads

Alarm

We can set an alarm to monitor for any put requests into the webdav folder

A threshold of 1 would be appropriate, as we should monitor for any file uploads into webdav

System Hardening

We can set the webdav configuration to disallow file uploads from all users who do not need those permissions.

Depending on the use case within the company we can allow only certain users write privileges, and potentially allow even fewer users to have upload privileges.

The End