Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

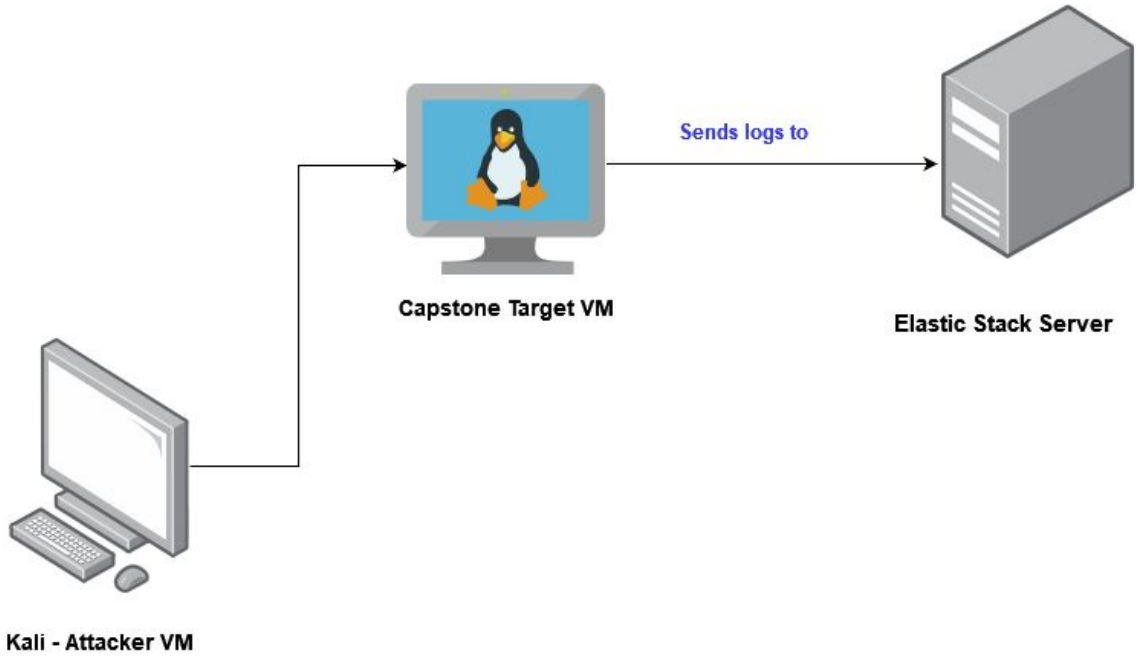
Report by Habib Slimani

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
IP Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
N/A	192.168.1.1	Gateway
Kali	192.168.1.90	Attacker Virtual Machine
ELK	192.168.1.100	Records Activity
Capstone	192.168.1.105	Target Virtual Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Remote Code Execution via Command Injection OWASP Top 10 #1 || **Critical**	Attackers can use PHP scripts to execute arbitrary shell commands.	This vulnerability allows attackers to open a reverse shell to the server.
Sensitive Data Exposure OWASP Top 10 #3 || **Critical**	The 'secret_folder' is publicly accessible, but contains sensitive data intended only for authorized personnel.	The exposure compromises credentials that attackers can use to break into the web server.
Unauthorized File Upload || **Critical**	Users are allowed to upload arbitrary files to the web server.	This vulnerability allows attackers to upload PHP scripts and other malicious files to the server.
No Brute Force Attack Mitigations || **Critical**	Attackers can brute force access to protected folders.	The lack of mitigation compromises credentials that attackers can use to break into the web server.

Exploitation: Sensitive Data Exposure

01

Tools & Processes

- `nmap` to scan network
- `dirb` to map URLs
- `Hydra` to brute force access
- `Crack Station` to crack passwords
- `msfvenom` to generate malicious payloads
- `Metasploit` for remote access
- `Firefox` to explore

02

Achievements

- The exploit revealed a `secret_folder` directory.
- This directory is password protected, but susceptible to **brute-force**.

03

Exploitation

- Publicly available information revealed that the administrator of the secret folder is `ashton`.
- This information is used to run a brute-force attack and compromise the administrator's credentials.

Exploitation: Unauthorized File Upload

01

Tools & Processes

- Cracked administrator's credentials used to connect via **WebDAV**.
- Generated malicious custom web shell with `msfvenom`.
- Uploaded shell via **WebDAV**.

02

Achievements

- Uploading a web shell allowed us to execute **arbitrary shell commands** on the target.

03

Aftermath

- Running arbitrary shell commands allows `Meterpreter` to open a full-fledged connection to the target.

Exploitation: Remote Code Execution

01

Tools & Processes

- Used `Meterpreter` to connect to uploaded web shell.
- Used shell to explore and compromise the target machine.

02

Achievements

- Leveraging the RCE allows us to open a `Meterpreter` shell to the target.
- Once on the target, the full file system is available for exploration.

03

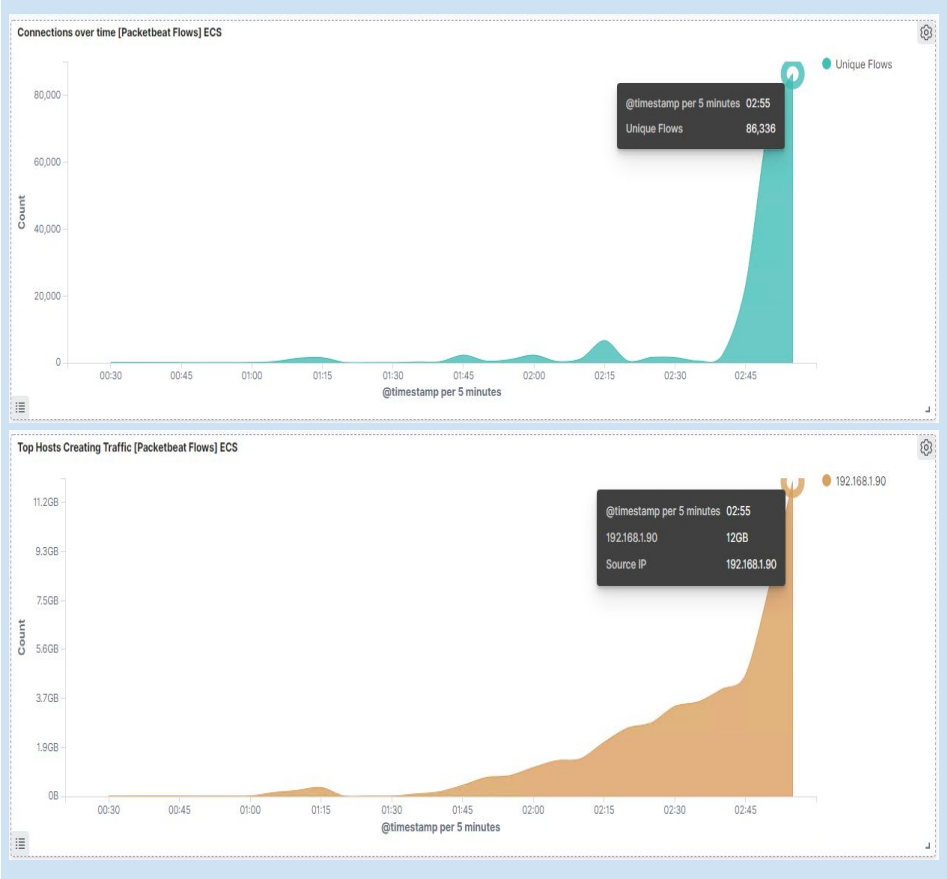
Aftermath

- Achieving a shell on the target allows us to display all files and move laterally within the network.

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



What time did the port scan occur?

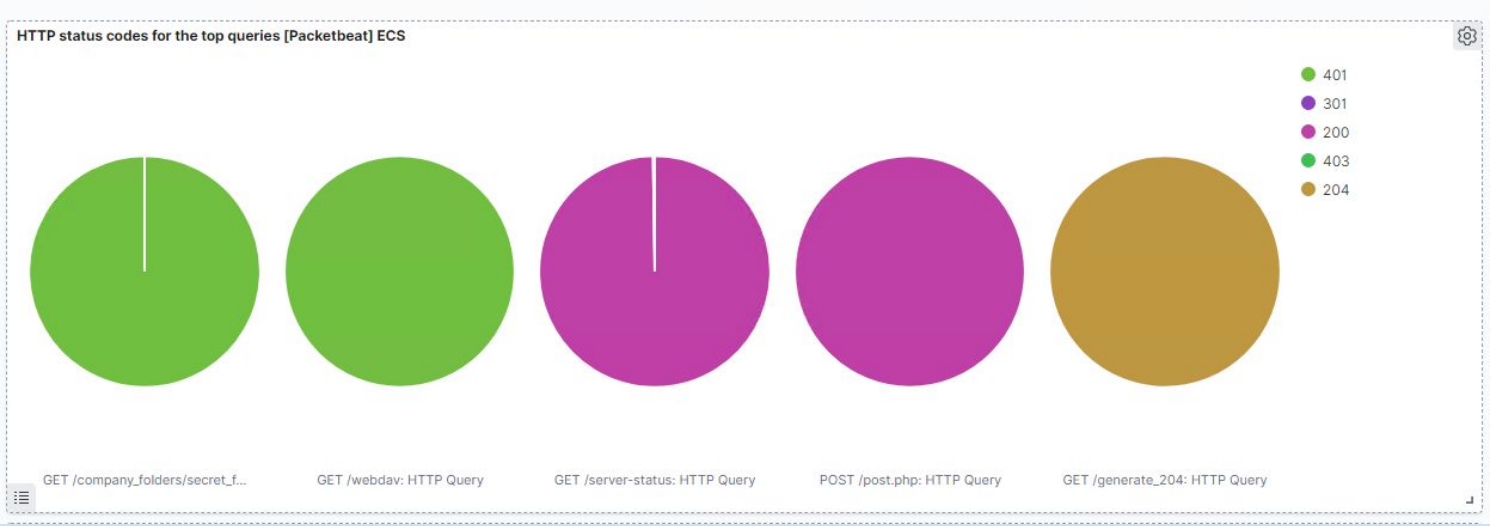
- 2:55

How many groups of packets were sent and from which IP?

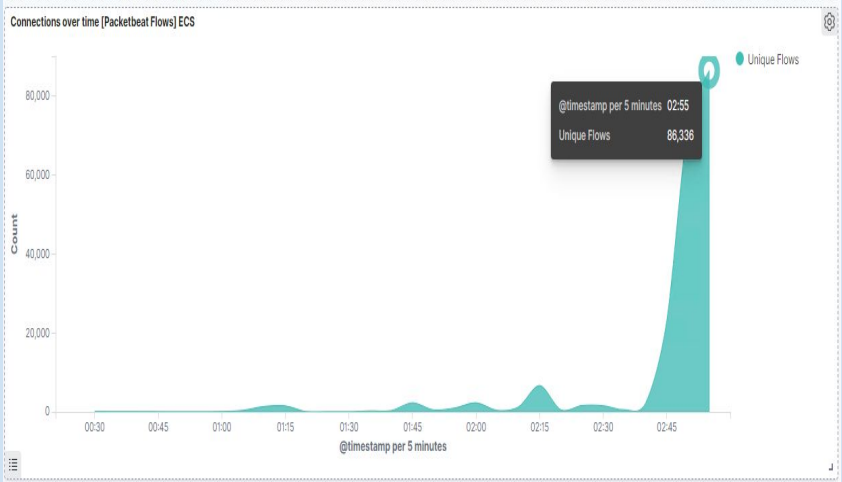
- Resting the courser at the top of the arc, we can observe **86,336.** In the second chart we can observe it's the IP address **192.168.1.90**.

On the next slide, we can observe that the victim responded back with 401 (Unauthorized), 207 (Multi-Status), 200 (OK), and 404 (Not found) responses.

Analysis: Identifying the Port Scan (cont.)



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	107,601
http://192.168.1.105/webdav	51,253
http://192.168.1.105/webdav/exploit.php	32

- In the first screenshot we can observe that the attack started at **2:55** with **86,336** requests and peaked at **107,601**.

The top three hits for directories and files that were requested were:

- `http://192.168.1.105/company_folder/secret_folder`
- `http://192.168.1.105/company_folder/webdav`
- `http://192.168.1.105/webdav/exploit.php`

Analysis: Finding the WebDAV Connection

The `secret_folder` directory was requested **107,601**.

The `exploit.php` file was requested **32 times**.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	107,601
http://192.168.1.105/webdav	51,253
http://192.168.1.105/webdav/exploit.php	32

Analysis: Uncovering the Brute Force Attack

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	107,601
http://192.168.1.105/company_folders/	17
http://192.168.1.105/company_folders/customer_info/	12
http://192.168.1.105/company_folders/secret_folder/	6

network.transport	tcp
network.type	ipv4
query	GET /company_folders/secret_folder
server.bytes	698B
server.ip	192.168.1.105
server.port	80
source.bytes	163B
source.ip	192.168.1.90
source.port	37800
status	Error
type	http
url.domain	192.168.1.105
url.full	http://192.168.1.105/company_folders/secret_folder
url.path	/company_folders/secret_folder
url.scheme	http
user_agent.original	Mozilla/4.0 (Hydra)

The logs contain evidence of a large number of requests for the sensitive data. Only **6** requests were successful. This is a telltale signature of a brute-force attack.

- Specifically, the password protected `secret_folder` was requested **107,601** times, but the file inside that directory was only requested **6** times. Out of **107,601** requests, only **6** were successful.

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

- Alarms should trigger if a given IP address sends more than **10 requests per second** for **more than 5 seconds.**

System Hardening

- Filter ICMP traffic.
- Enable an allowed IP list.
- Close unused ports or block them with a firewall.
- Proactive scan to identify running services and potential vulnerabilities to address.

Mitigation: Finding the Request for the Hidden Directory

Alarm

- Alarms should trigger if an IP that is not on the whitelist attempts to connect.

System Hardening

- Access to the sensitive file(s) can be locally restricted to a specific user.
- Move folder to server with key-based SSH access from whitelisted IPs.
- Encryption of file(s) at rest.
- Log non whitelisted IPs access to the folder.

Mitigation: Preventing Brute Force Attacks

Alarm

- Alarms should trigger when more than 100 requests per seconds for a duration of 5 seconds is detected.
- Alarms should trigger when an IP address that is not on the whitelist is trying to authenticate.

System Hardening

- Configuring `fail2ban` or a similar utility would mitigate brute force attacks.
- Limit failed login attempts.
- Limit logins to a specified IP address.
- Two factor authentication.
- Unique login URLs.
- Require authentication to upload files.
- Block upload of executable files.

Mitigation: Detecting the WebDAV Connection

Alarm

- Alarms should trigger by any read performed on files within **WebDAV** OR trigger by any unauthorized users' activity within it.

System Hardening

- Administrators must install and configure `Filebeat` on the host to monitor **WebDAV**-related activity.
- Use Restrict Access function to create an ACL that restricts access to **WebDAV**-enabled resources defining what is allowed and who can perform an allowed action.

Mitigation: Identifying Reverse Shell Uploads

Alarm

- Alarms should trigger upon receipt of any POST request containing a form or file data of an unauthorized file type, e.g., “`.php`”.

System Hardening

- Write permissions can be restricted on the host.
- Uploads can be isolated into a sandboxed partition/folder.
- `Filebeat` should be enabled and configured to monitor file uploads as well as activity in any sandboxed environment.

the End