Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01 **Network Topology**

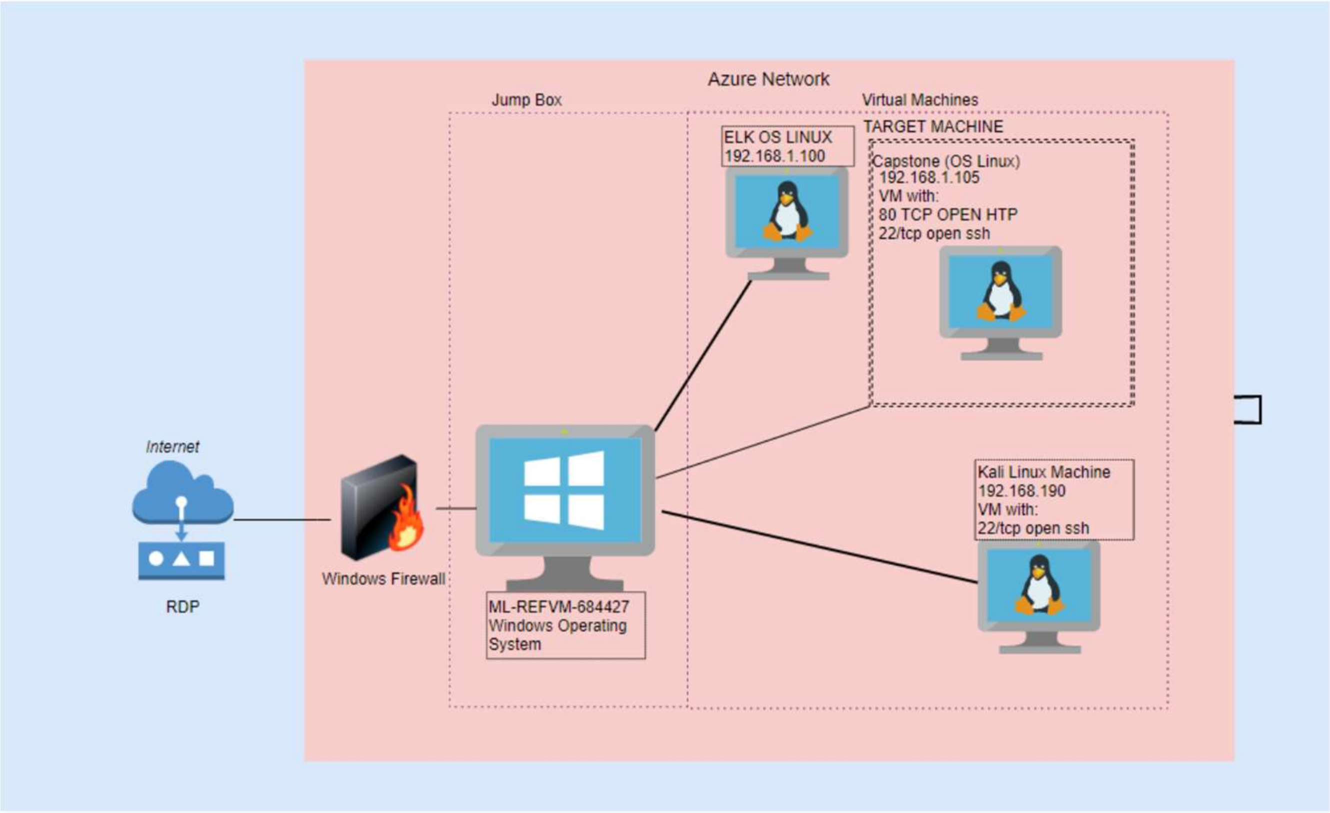
02 **Red Team**: Security Assessment

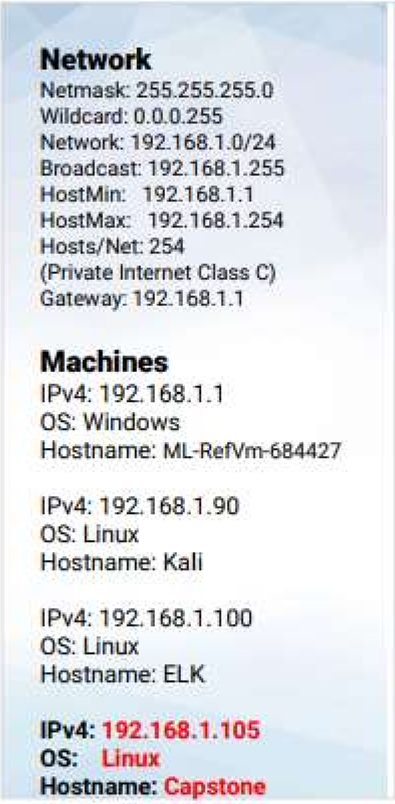
03 **Blue Team**: Log Analysis and Attack Characterization

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

Network Topology

Network Topology





Network
Netmask: 255.255.255.0
Wildcard: 0.0.0.255
Network: 192.168.1.0/24
Broadcast: 192.168.1.255
HostMin: 192.168.1.1
HostMax: 192.168.1.254
Hosts/Net: 254
(Private Internet Class C)
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Red Team
Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
CAPSTONE	192.168.1.105	WEB Server
ELK	192.168.1.100	SIEM System
ML-RefVm-684427	192.168.1.1	Default Gateway
Kali	192.168.1.90	Attacking Virtual Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
The directory listing is widely available on the Apache Web Server	*Anyone can use the browser to navigate contents of directories on Capstone Apache web server*	*Attackers immediately know what folders are available, which have employees names and other useful information in beginning an attack.*
There is no failed password lockout, which allows attackers to enter passwords until they access the system. There is not password criteria, allowing weak password names	We quickly found the password using ‘rockyou”. We were never locked out for failed login attempts allowing brute force attack.	An attacker has no problem getting credentials to penetrate the system.
There is Persistent Reverse Shell Backdoor, which means the root directory can be accessed by the attacker.	Reverse shell payloads can be exploited on the web server as IPS/IDS/Firewall(s) allow outbound ports and undetected reverse shell.	Capstone Apache web Server has reverse backdoor shell access.

Exploitation: : Directory Listing Enabled on Apache

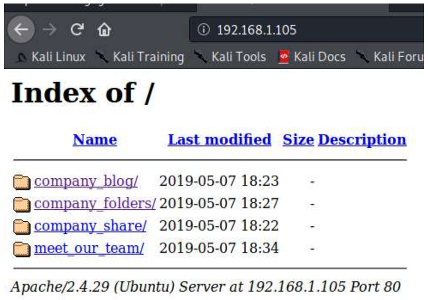
01

First we navigate with Firefox to 192.168.1.105/ and find the Apache Server index of the company folders. Inside the folders are the employee names and roles, exposing which employees many have excessive privileges.

02

After searching the company folders, we read the company blog which introduces us to Ashton and his position indicated excessive privilege. When then navigate to Ashton's folder and find out he is responsible for the folder /company_folders/secret_folder/

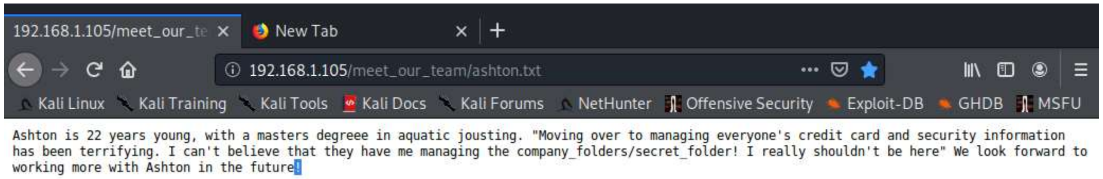
03



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



```
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!
```

Exploitation: Weak Password & No Failed Password Lockout

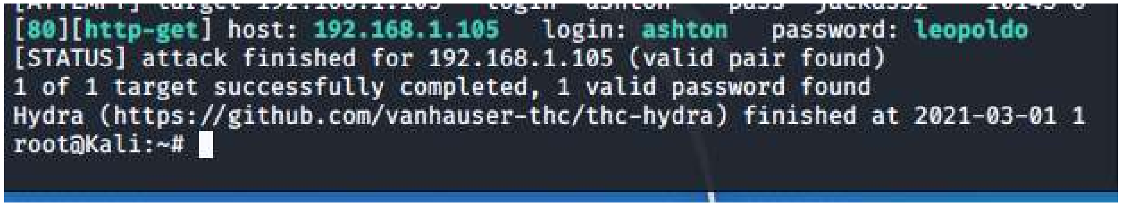
We use Hydra brute force attack to get the password for Ashton's account.

Password for Ashton was "leopoldo" found in 'rockyou' dictionary.

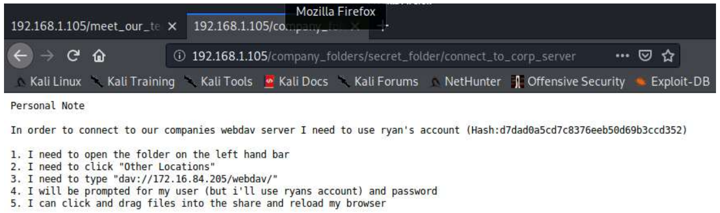
We log into 192.168.1.105/company_folders/secret_folder Ashton's credentials.

We are then able to access /secret_folder/.

Secret folder gives us access info for /webdav/ system.



```
[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 2021-03-01 1
root@Kali:~#
```

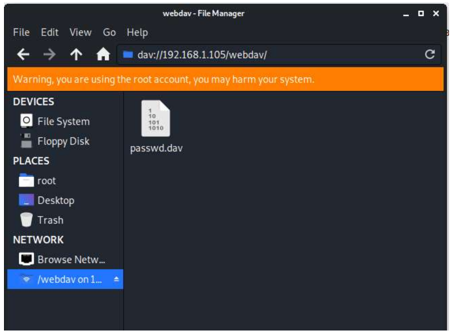


```
Personal Note

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
```

Secret folder has the hash for Ryan's password, we cracked the hash and reveal the password "linux4u" allowing us access to webdav.



Exploitation: Persistent Reverse Shell Backdoor

01

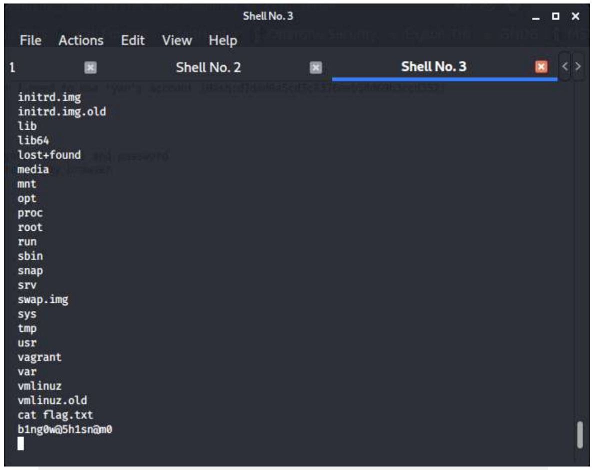
Using msfvenom payload we establish a remote listener and execute a reverse shell backdoor on the Apache server of the victim machine

02

Now that we have opened a remote backdoor, we gain access to the root directory on the victim machine.

In doing so we are able to search the director and capture the flag file.

03



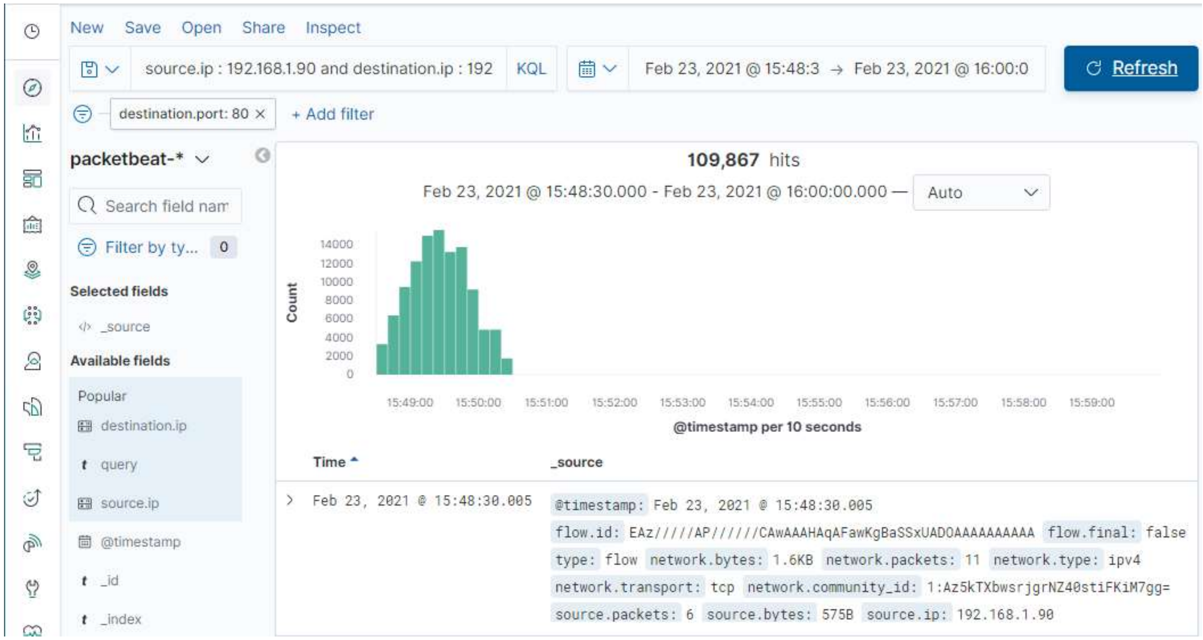
```
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
snap
srv
swap.img
sys
tmp
usr
vagrant
var
vmlinuz
vmlinuz.old
cat flag.txt
b1ng0w@5h1sn@m0
```

Blue Team

Log Analysis and
Attack Characterization

Analysis: Identifying the Port Scan

- What time did the port scan occur? Port scan occurred at 15:48:26.00 on February 23, 2021.
- How many packets were sent, and from which IP? There were 109, 867 packets sent.
- What indicates that this was a port scan? Since multiple ports were requested simultaneously from the same ip address, this indicates it was a port scan.



source.ip : 192.168.1.90 and destination.ip : 192

Feb 23, 2021 @ 15:48:3 → Feb 23, 2021 @ 16:00:0

destination.port: 80

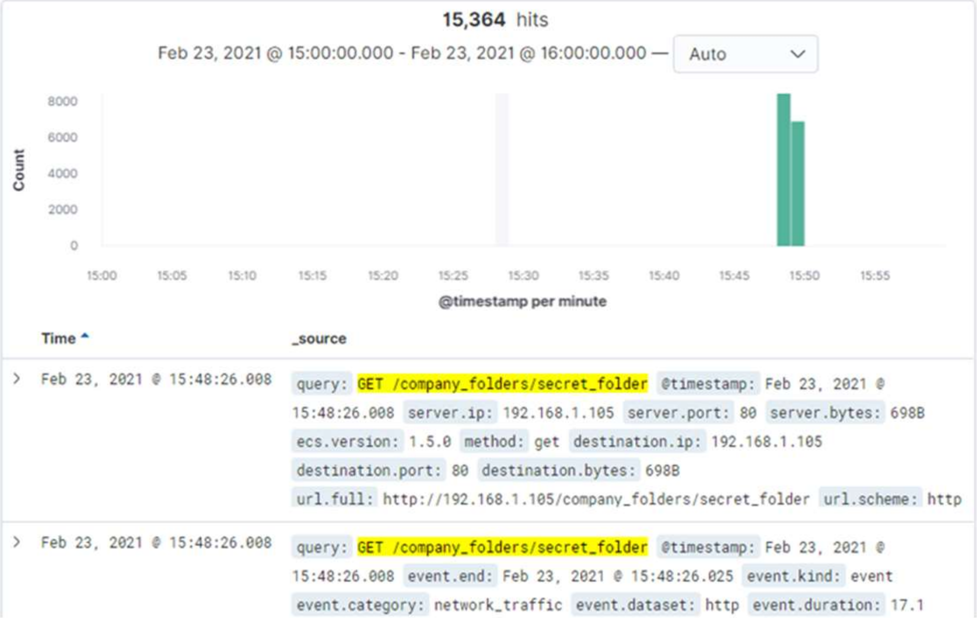
109,867 hits

Feb 23, 2021 @ 15:48:30.000 - Feb 23, 2021 @ 16:00:00.000 —

```
Time      _source
Feb 23, 2021 @ 15:48:30.005   @timestamp: Feb 23, 2021 @ 15:48:30.005
flow.id: EAz/////AP//////CAwAAAHAqAFawKgBaSSxUADOAAAAAAAAAA flow.final: false
type: flow network.bytes: 1.6KB network.packets: 11 network.type: ipv4
network.transport: tcp network.community_id: 1:Az5kTXbwsrjgrNZ40stiFKiM7gg=
source.packets: 6 source.bytes: 575B source.ip: 192.168.1.90
```

Analysis: Finding the Request for the Hidden Directory

- What time did the request occur? 15:48:26 February 23, 2021
- How many requests were made? 15, 364
- Which files were requested? company_folders/secret_folder
- What did they contain? Hash code password for user Ryan



15,364 hits

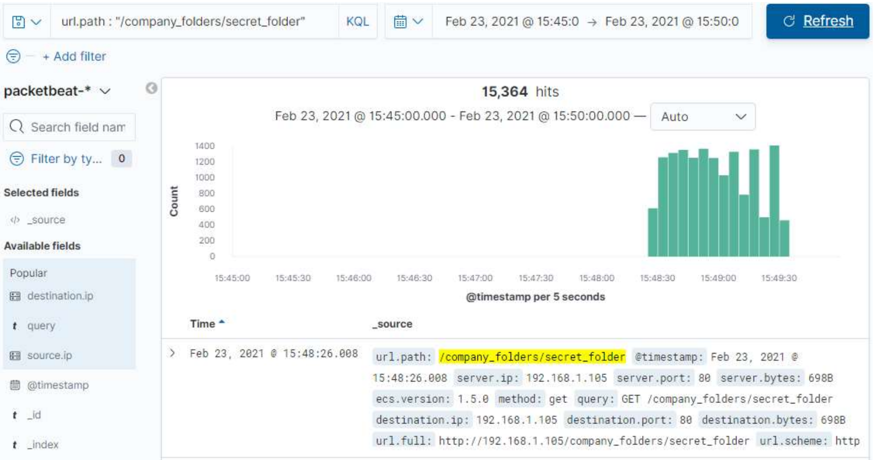
Feb 23, 2021 @ 15:00:00.000 - Feb 23, 2021 @ 16:00:00.000 — Auto

```
Time ▴                      _source
> Feb 23, 2021 @ 15:48:26.008  query: GET /company_folders/secret_folder @timestamp: Feb 23, 2021 @
                               15:48:26.008 server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B
                               ecs.version: 1.5.0 method: get destination.ip: 192.168.1.105
                               destination.port: 80 destination.bytes: 698B
                               url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
> Feb 23, 2021 @ 15:48:26.008  query: GET /company_folders/secret_folder @timestamp: Feb 23, 2021 @
                               15:48:26.008 event.end: Feb 23, 2021 @ 15:48:26.025 event.kind: event
                               event.category: network_traffic event.dataset: http event.duration: 17.1
```

Analysis: Uncovering the Brute Force Attack

Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- How many requests were made in the attack? 15, 364
- How many requests had been made before the attacker discovered the password? 15,362



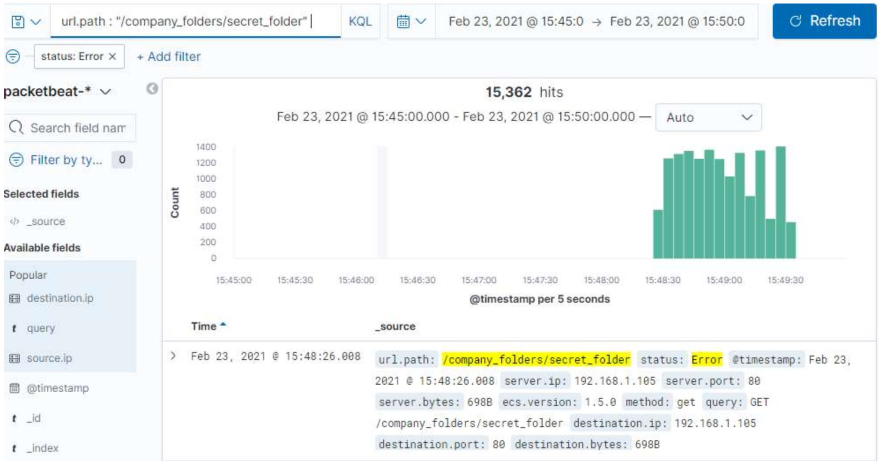
url.path : "/company_folders/secret_folder"

Feb 23, 2021 @ 15:45:0 → Feb 23, 2021 @ 15:50:0

15,364 hits

Feb 23, 2021 @ 15:45:00.000 - Feb 23, 2021 @ 15:50:00.000 —

```
> Feb 23, 2021 @ 15:48:26.008   url.path: /company_folders/secret_folder @timestamp: Feb 23, 2021 @
                                15:48:26.008 server.ip: 192.168.1.105 server.port: 80 server.bytes: 698B
                                ecs.version: 1.5.0 method: get query: GET /company_folders/secret_folder
                                destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 698B
                                url.full: http://192.168.1.105/company_folders/secret_folder url.scheme: http
```



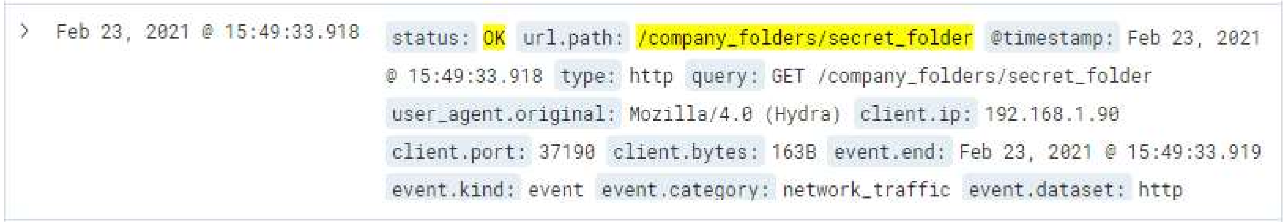
url.path : "/company_folders/secret_folder"

Feb 23, 2021 @ 15:45:0 → Feb 23, 2021 @ 15:50:0

15,362 hits

Feb 23, 2021 @ 15:45:00.000 - Feb 23, 2021 @ 15:50:00.000 —

```
> Feb 23, 2021 @ 15:48:26.008   url.path: /company_folders/secret_folder status: Error @timestamp: Feb 23,
                                2021 @ 15:48:26.008 server.ip: 192.168.1.105 server.port: 80
                                server.bytes: 698B ecs.version: 1.5.0 method: get query: GET
                                /company_folders/secret_folder destination.ip: 192.168.1.105
                                destination.port: 80 destination.bytes: 698B
```



```
> Feb 23, 2021 @ 15:49:33.918   status: OK url.path: /company_folders/secret_folder @timestamp: Feb 23, 2021
                                @ 15:49:33.918 type: http query: GET /company_folders/secret_folder
                                user_agent.original: Mozilla/4.0 (Hydra) client.ip: 192.168.1.90
                                client.port: 37190 client.bytes: 163B event.end: Feb 23, 2021 @ 15:49:33.919
                                event.kind: event event.category: network_traffic event.dataset: http
```

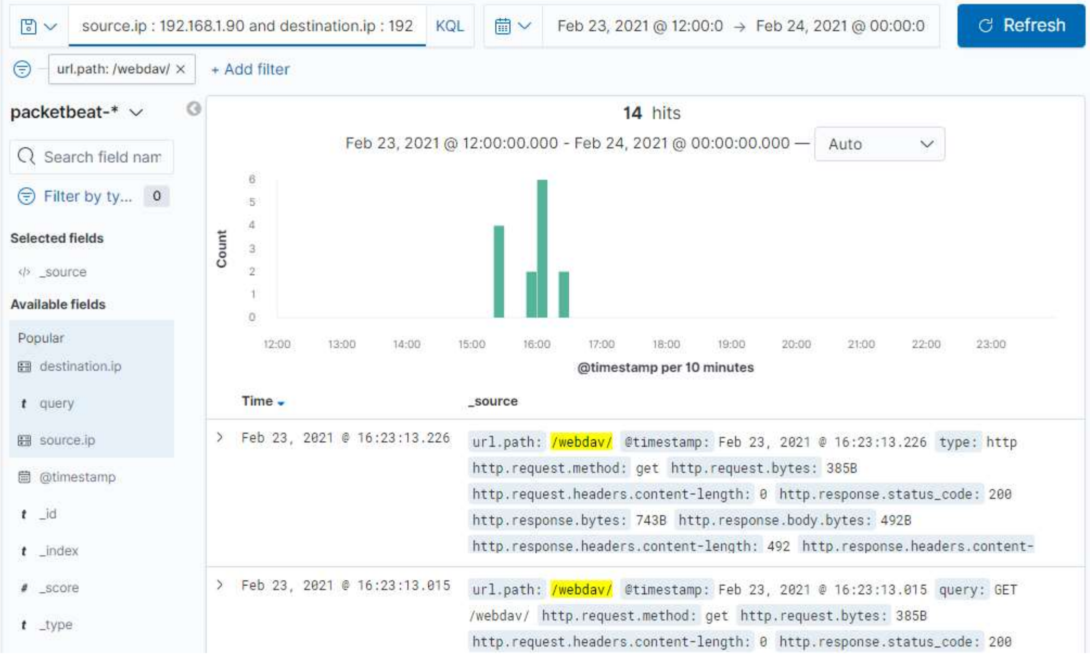
Analysis: Finding the WebDAV Connection

- How many requests were made to this directory?

There were 14 requests made to WebDav directory

- Which files were requested?

/webdav/ /webdav/shell.php/ and /webdav/passwd.dav



source.ip : 192.168.1.90 and destination.ip : 192

Feb 23, 2021 @ 12:00:0 → Feb 24, 2021 @ 00:00:0

url.path: /webdav/

14 hits

Feb 23, 2021 @ 12:00:00.000 - Feb 24, 2021 @ 00:00:00.000 —

Time	_source
Feb 23, 2021 @ 16:23:13.226	url.path: /webdav/ @timestamp: Feb 23, 2021 @ 16:23:13.226 type: http http.request.method: get http.request.bytes: 385B http.request.headers.content-length: 0 http.response.status_code: 200 http.response.bytes: 743B http.response.body.bytes: 492B http.response.headers.content-length: 492 http.response.headers.content-
Feb 23, 2021 @ 16:23:13.015	url.path: /webdav/ @timestamp: Feb 23, 2021 @ 16:23:13.015 query: GET /webdav/ http.request.method: get http.request.bytes: 385B http.request.headers.content-length: 0 http.response.status_code: 200

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

We suggest an alarm to detect future port scans:

Search criteria:
destination.ip: 192.168.1.105 and source.ip: (not 192.168.1.105) and destination.port: (not 443 or 80)
Report criteria:
Number of ports accessed per source IP per second.
Alarm criteria/threshold:
Alert email and log when > 3 none port 403 or port 80 scans detected at the same timestamp from the same IP occur.

System Hardening

In this case were going to recommend IPtables/Firewall port blocking and scan delays to prevent future portscans.
Kibana or Splunk can be used to send alerts of multiple port access coming simultaneously from a single source ip and prepare a response to the potential threat.

Mitigation: Finding the Request for the Hidden Directory

Alarm

An alarm set to expose future unauthorized access:

Search criteria:

source.ip: (not 192.168.1.105 or 192.168.1.1) and url.path : *secret_folder*

Report criteria:

Number of times “secret_folder” accessed from external IP

Alarm criteria/threshold:

Alert email and log when > 0 access is detected on “secret_folder” from IPs other than 192.168.1.105 or 192.168.1.1.

System Hardening

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

The configuration file will be modified to prevent unauthorized access to the “secret_folder”.

Open your httpd.conf file:

- nano /etc/httpd/conf/httpd.conf * Locate directory section (/var/www/) and set it as follows:

```
➢ Order allow,deny
Allow from 192.168.1.1
 Allow from 192.168.1.105
Allow from 127
Deny from 192.168.1.90
</Directory>
```

Mitigation: Preventing Brute Force Attacks

Alarm

The following alarm can be set to detect future brute force
attacks:
Search criteria:
http.request.method : "get" and user_agent.original :"Mozilla/4.0 (Hydra)" and url.path :"/company_folders/secret_folder/" and status :(Error or OK)
Report criteria:
Number of times Error (401) response detected in 10 second interval.
Alarm criteria/threshold:
Alert email and log when, on protected files and folders, > 5
Error (401) responses occur at any time OR any OK (200) responses occur from non-trusted IPs

System Hardening

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

A strong password policy is an essential mitigation strategy, one that can't be penetrated by a brute force attack.

Locking out the user after multiple failed logins will also prove to be an effective mitigation strategy.
Multi layered login, send a success (200) response for a failed password and forward the user to try again on a second login page.

Security questions and CAPTCHA will ensure that the user is human and not a system attack.

Mitigation: Detecting the WebDAV Connection

Alarm

The following alarm can be set to detect future unauthorized access to this directory::

Search criteria:

http.request.method : * and url.path: *webdav*

and

source.ip: (not 192.168.1.150 or 192.168.1.1)

Report criteria:

Number of times the directory is requested from non-trusted IPs.

Alarm criteria/threshold:

Alert email and log when requests are made, on protected

files and folders, from non-trusted IPs

System Hardening

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

```
> nano /etc/httpd/conf/httpd.conf
```

Navigate to the following directory:

```
<Directory /var/www/webdav/>
Order allow,deny
Allow from 192.168.1.1
Allow from 192.168.1.105
Allow from 127
Deny from all
</Directory>
```

Mitigation: Identifying Reverse Shell Uploads

Alarm

The following alarm can be set to detect future unauthorized file uploads:
Search criteria:
http.request.method : “put” and url.path: *webdav* and source.ip: (not 192.168.1.1 or 192.168.1.105)
NOTE: Also Reverse Shell Signature for Consideration of Reverse Shell Detection:
source.ip: 192.168.1.105 and destination.ip: (not 192.168.1.1
or 192.168.1.105) and destination.port > 0 and network.protocol: (not *) and http.response.body.bytes: (not *) and source.port: (not 80 or 22)
Report criteria:
Count directory “put” method from non-trusted IPs.
Alarm criteria/threshold:
Alert email and log when “put” request methods are made, on protected folders, from non-trusted IPs

System Hardening

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

```
<Directory /var/www/webdav/>
Order allow,deny
Allow from 192.168.1.1
Allow from 192.168.1.105
Allow from 127
Deny from all
<LimitExcept GET POST HEAD>deny from all
</LimitExcept>
</Directory>
```

The End