Capstone Engagement

Red Team vs. Blue Team

Assessment, Analysis,
and Hardening of a Vulnerable System

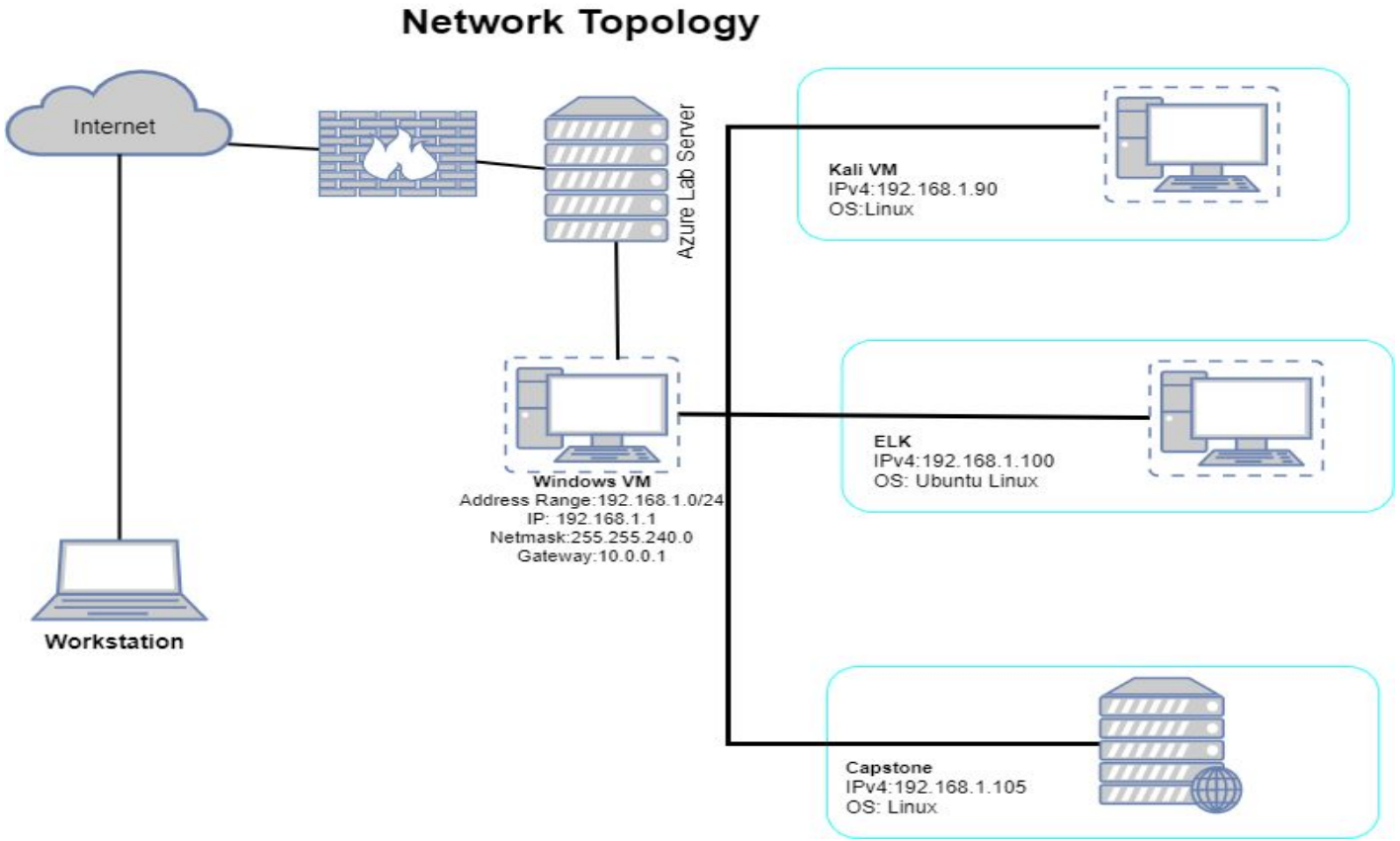
By: Modou Jarju

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range:192.168.1.0/24
Netmask:255.255.240.0
Gateway:10.0.0.1

Machines
IPv4:192.168.1.1
OS:Windows
Hostname:ELK

IPv4:192.168.1.90
OS:Linux
Hostname: Kali

IPv4:192.168.1.100
OS: Ubuntu Linux
Hostname: Elk/ Cluster: Elasticsearch

IPv4:192.168.1.105
OS: Linux
Hostname:Capstone

Red Team

Security Assessment

Recon: The Captstone, ELK and the host windows machine were all vulnerable to attack with open ports

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Windows	192.168.1.1	Windows Virtual machine & Server
Kali	192.168.1.90	Attacker's Machine
ELK	192.168.1.100	Target/victim Machine
Capstone	192.168.1.105	Target machine

Vulnerability Assessment: nmap

The assessment uncovered the following critical vulnerabilities in the target: ELK

Vulnerability	Description	Impact
SSH	*22/tcp*	*openSSH*
HTTP	80/tcp	Apache httpd 2.4.29
netbios-ssn	139/tcp	Samba smbd 3.X - 4.X
Microsoft-ds	445	Trojan/worm W32.HLLW.Deloder [Symantec-2003-030812-5056-99] IraqiWorm (aka Iraq_oil.exe)

Exploitation: [Nmap Port scanned > Port 80/tcp]

01

Tools & Processes

Used nmap and port scanned victim's network and SSH to log in to target machine via open port

02

Achievements

Discovered all open ports and vulnerabilities across the network. Amongst them, port 80/tcp on the capstone machine

03

[Ran: '**nmap -sV 192.168.1.0/24**] to scan for machines on the network, their OS versions and Vulnerabilities. Screenshots on Day 1 README file.

Exploitation: [Located hidden dir] [HTTP Port 80/tcp - Apache httpd 2.4.29]

01

Tools & Processes

Navigated to the machine's IP address 192.168.1.105 via a web browser, found a hidden directory called "secret_folder"

02

Achievements

- Brute forced and cracked hashes using 'hydra' and Crackstaion and **gained the passwords.**

  - Gained access to secret folder, connected to WebDAV server and opened path for further exploitations

03

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

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [child 6] (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 2021-03-13 09:48:19
root@Kali:~# hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder/
```

Exploitation: [Brute force, and cracked weak passwd]

01

Tools & Processes

I brute forced and cracked the password using ‘hydra’

- I used ‘CrackStation’ to break user ‘Ryan’s password hash.

02

Achievements

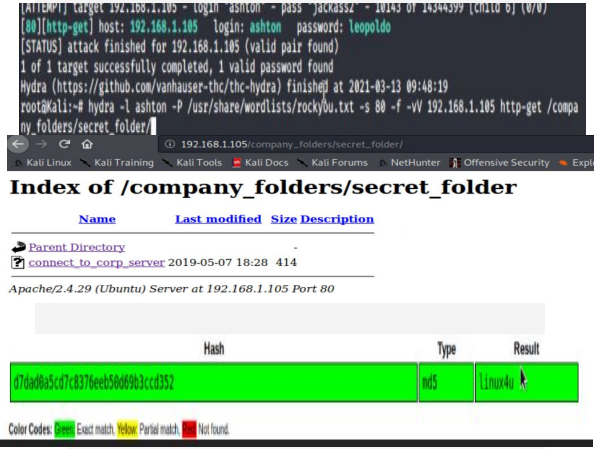
Cracked user ‘ashton’s password in 68 seconds

Broke user ‘Ryan’s hash and revealed password ‘linux4u’

Gained access to the server via Webdav

03

[Ran: **hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder]**



Exploitation: [Meterpreter Reverse Shell - TCP]

01

Tools & Processes
Created and uploaded a reverse shell via ‘msfvenom’

- Connected to server via WebDAV and uploaded my exploit ‘shell.php’
  - Lauched ‘msfconsole’ and started listener

-Used cracked password, connected to WebDAV folder and executed exploit ‘shell.php.

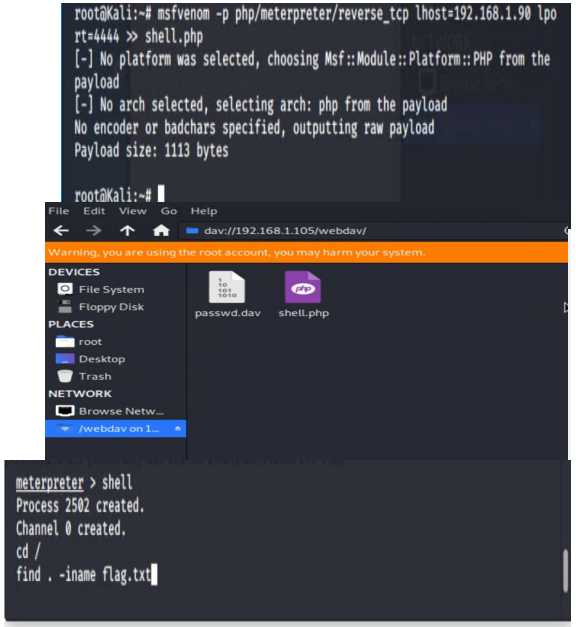
02

Achievements

Gained user shell and printed a file ‘flag.txt’;. This could be any file on target machine.

-With such exploit, I could access any file on victim’s machine

03

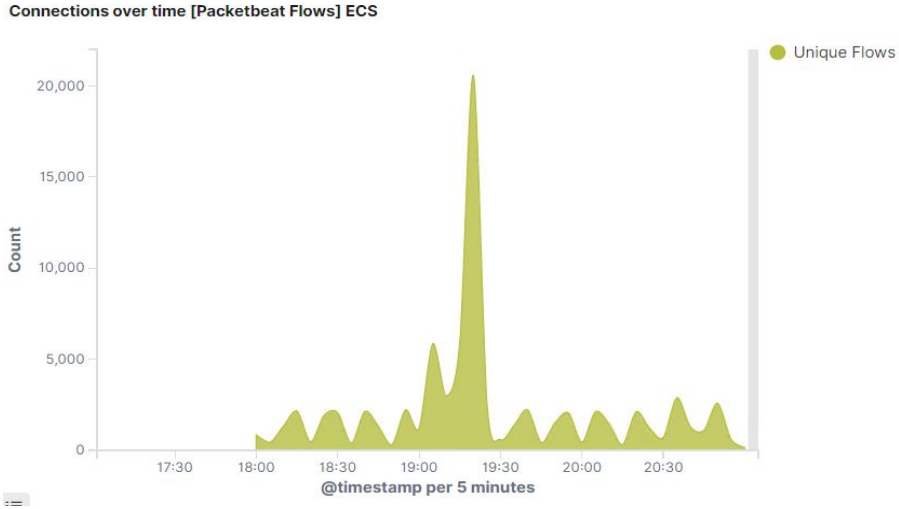


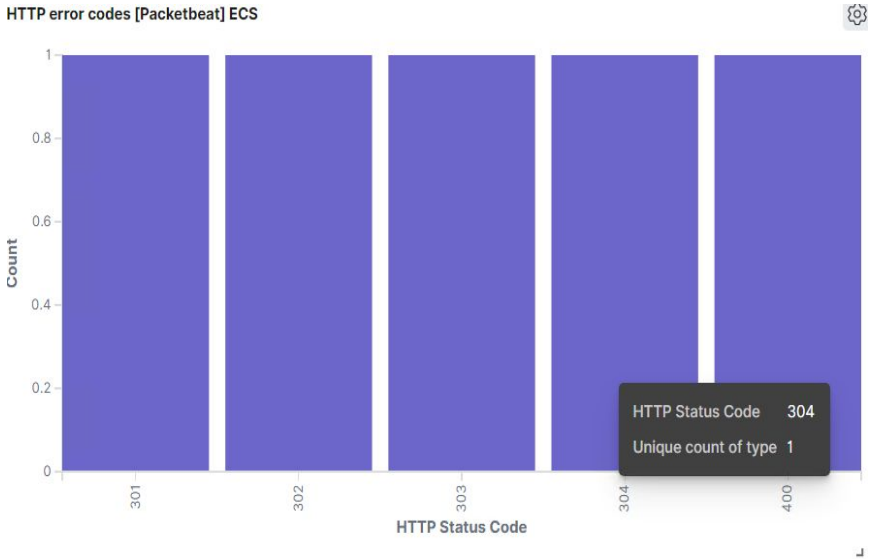
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

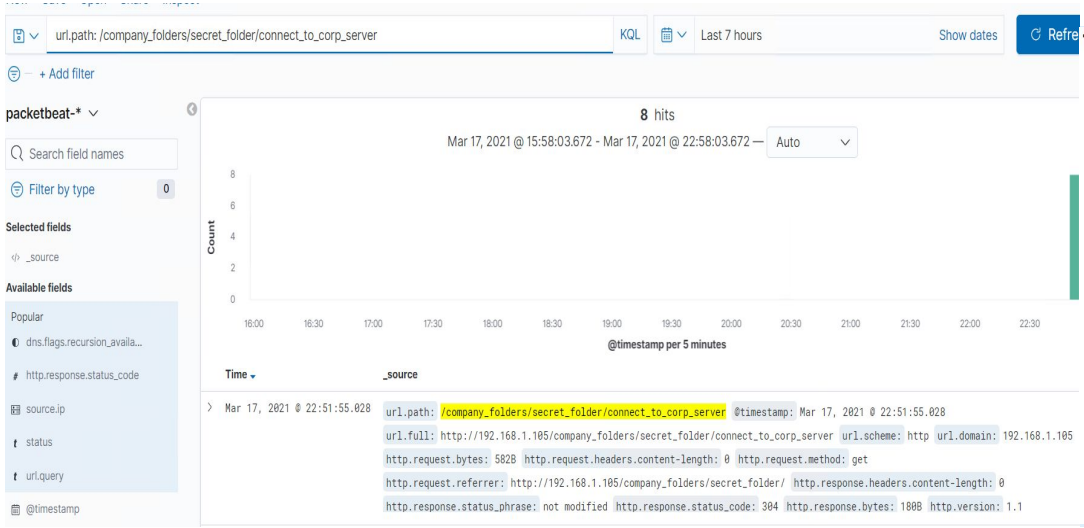
- Port scan time = `Mar 13, 2021 @ 17:31:48.888`
- Number of packets = VM MAX OUT, SORRY, TO ANSWER ON THIS
- Indications of a port scan = Multiple ports requested at the same time

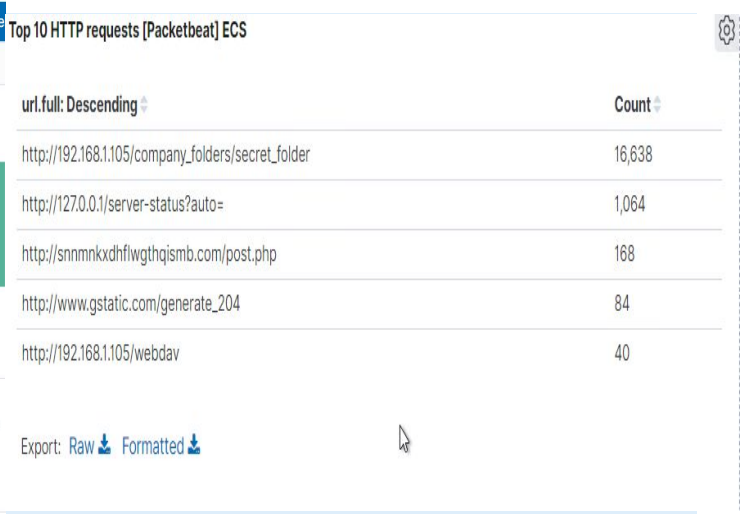




Analysis: Finding the Request for the Hidden Directory

- Time of request = `@ 17:31:48.888`
- Number of requests made = `16,638 Requests at 19:27 and 8 were successful`
- File requested = `connect_to_corp_server`
- Requested file content = `passwords 'passwd.dav'.`





url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,638
http://127.0.0.1/server-status?auto=	1,064
http://snnmnkxdhflwgthqismb.com/post.php	168
http://www.gstatic.com/generate_204	84
http://192.168.1.105/webdav	40

Analysis: Uncovering the Brute Force Attack

- Requests made in the attack? = `16,638 Requests at 19:27`
- Requests made before the attacker discovered the password? = `16,630 requests were made and 8 were successful == Total of 16,638 requests on the "secret_folder" directory`

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,638

Analysis: Finding the WebDAV Connection

- Request made to ‘WebDAV directory = `40 made`
- Files requested = `connect_to_corp_server`

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	40

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

The following alarm can be set 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.

- Detecting a TCP connect scan can also be set

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

Possible configurations on the host to mitigate port scans:

Host-Based IDSs

Firewall block all incoming and outgoing ports except for those needed (80 and 443)> iptables -A INPUT -p tcp -m tcp -m multiport ! --dports 80,443 -j DR

Block/forward(honeypot)/delay port scans (web server)

Mitigation: Finding the Request for the Hidden Directory

Alarm

The following alarm can be set to detect 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 192.168.1.1, 192.168.1.100 or 192.168.1.105.

System Hardening

On host:
Modify configuration file on the host to block unwanted access to the “secret_folder” from any IP other than those listed and disable dir listings:

Open your httpd.conf file:
> nano /etc/httpd/conf/httpd.conf
* Locate directory section (/var/www/) and set it as follows: <Directory /var/www/company_folders/secret_folder/
>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 IP

System Hardening

Host Configuration:

Set password attempts to 3 maximum before a lock out

--above followed by security questions

Set an Unauthorized 401 or any other type unauthorized access

Implement a Strong Password policy to prevent weak passwords like 'linux4u'

Use a CAPTCHA to ensure human user

Mitigation: Detecting the WebDAV Connection

Alarm

Alarm to detect future access to this directory:

Search criteria:
http.request.method : * and url.path: *webdav* and source.ip: (not 192.168.1.1 or 192.168.1.105)

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

Alarm criteria/threshold:
Alert email log when requests are made, on protected files and folders, from non-trusted IPs

System Hardening

On host:
Modify configuration file on the host to block unwanted access to the “WebDAV” from any IP other than those listed and disable dir listings:

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

```
<Directory /var/www/webdav/>
>Order allow, deny
Allow from 192.168.1.1
Allow from 192.168.1.105
Allow from 127
Deny from 192.168.1.90  (or do “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)

Also Reverse Shell Signature for Consideration of Reverse Shell Detection:
source.ip: 192.168.1.90 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 log when “put” request methods are made, on protected folders, from non-trusted IPs

System Hardening

Set firewall and block all uploads from any IPs other than the allow IPs

The following could also be set:

Open httpd.conf file:
> nano /etc/httpd/conf/(example - httpd.conf or ,location may vary)
* Locate directory section (/var/www/) and set it as follows:
<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