Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

PRESENTATION BY

01 Erkan Eksen

02 Joshua Landes

03 Steven Bar

04 Justin Pence

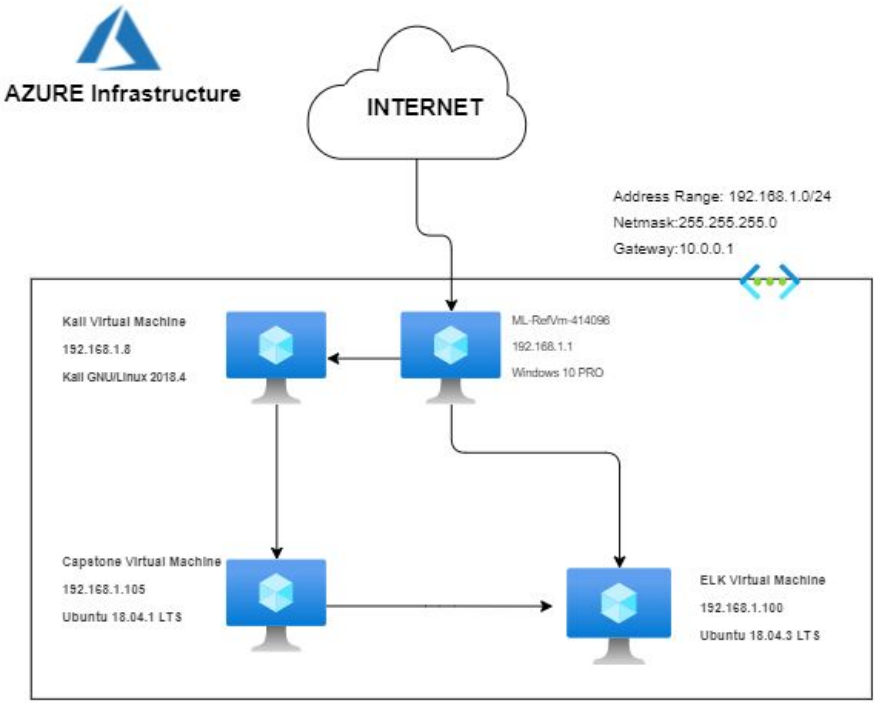
05 John Shaffer

06 Sally lamin

Network Topology

Network Topology

Project-2 Network Diagram



05/06/2021
Erkan Eksen

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 10 Pro
Hostname: ML-RefVm-414096

IPv4:192.168.1.8
OS:**Kali GNU/Linux 2018.4**
Hostname:Kali

IPv4:192.168.1.105
OS:**Ubuntu 18.04.1 LTS**
Hostname:Capstone

IPv4:192.168.1.100
OS:**Ubuntu 18.04.3 LTS**
Hostname:ELK

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RefVm-414096	192.168.1.1	Cloud based virtual desktop hosting the other 3 virtual machines
ELK	192.168.1.100	ELK SIEM Linux Virtual Machine
Capstone	192.168.1.105	Victim (vulnerable) Linux Virtual Machine
Kali	192.168.1.8	Attacking Kali Virtual Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Poorly designed web server, application and pages.	They all give the attacker whatever info they are asked.	Vulnerability lets the attacker know that he can compromise this system so he can plan and execute an attack.
Remote File Inclusion (RFI) and Local File Inclusion (LFI)	These vulnerabilities occur when a web application allows the user to submit input into files or upload files to the server.	Allows attackers to gain access to web servers.
Poor or not-configured naming conventions.	Using just names for usernames	Usernames can be guessed and vulnerability makes brute force attacks successful.
Poor or not-configured password policy (complexity, length and lockout)	Using easy to crack passwords and hashing them without salting, no lock out.	Allows attacker get the credentials very fast and easy.

Poorly Designed Web Server

01

Tools & Processes
The web server allowed all directories to be viewed via a web browser. Using DIRB we easily found hidden pages, and via reconnaissance of the visible pages, we were able to find references to hidden files and sources of sendible data, including login data.

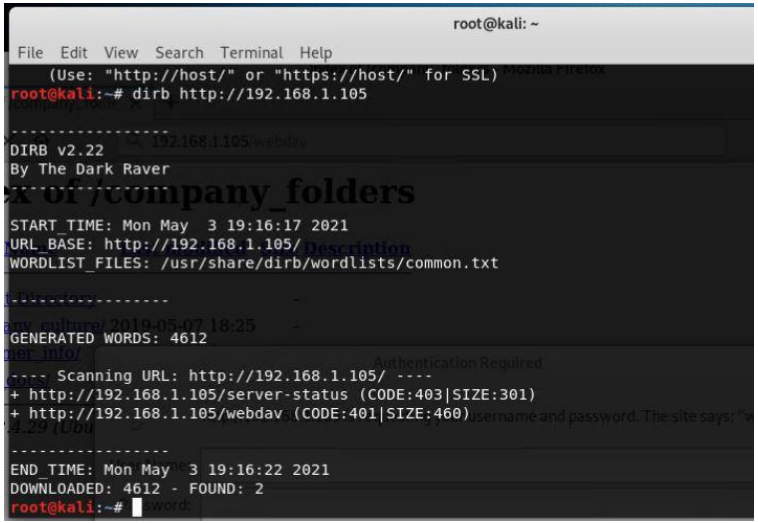
02

Achievements
By simply running some reconnaissance, we discovered two critical “hidden” directories, which helped give us full access to the system.

03

Screenshots on the following page.

Poorly Designed Web Server Screenshots



Poor Naming Conventions and Password Policy

01

Tools & Processes
On the web server, user names were configured simply as the user's first name, making them easily guessable.
The password policy was also very weak, which allowed Hydra to crack them easily and quickly.

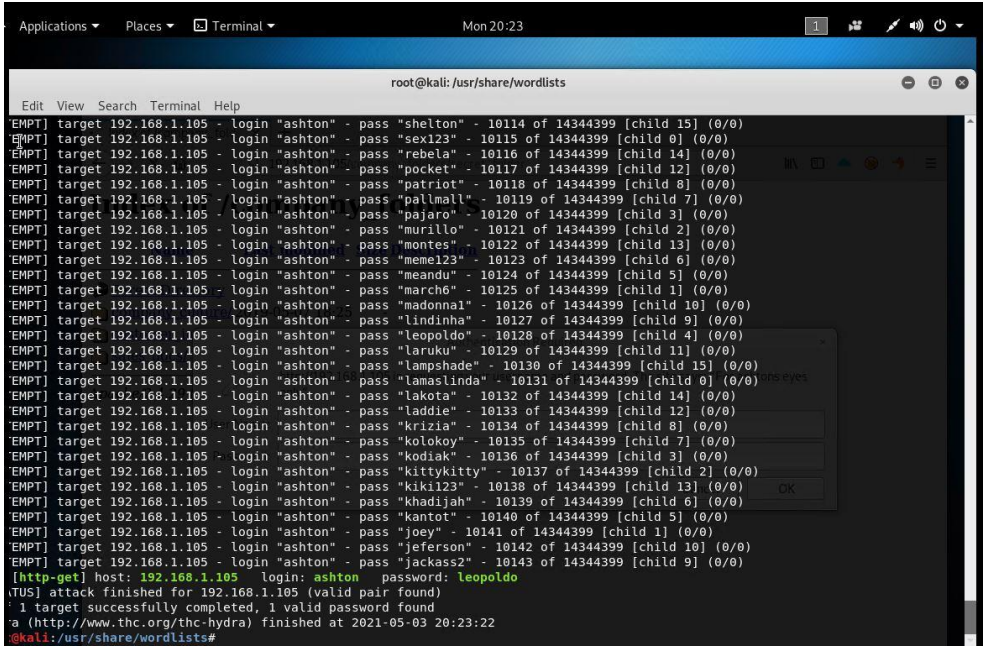
02

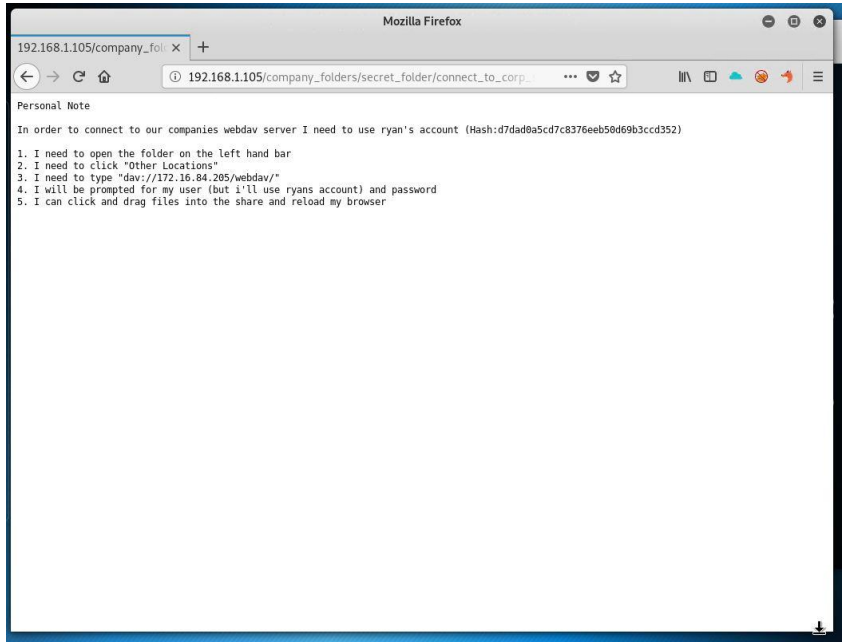
Achievements
Through reconnaissance, we had discovered usernames for several folders. Once the secret_folder was discovered, we used hydra to brute force attack the web server, and we were able to login with Ashton's login information.
On the page, we also found Ryan's login information, as well as instructions for accessing the Webdav page.

03

Screenshots on the following page.

Poor Naming Conventions and Password Policy Screenshots





Poor Naming Conventions and Password Policy Screenshots

Enter up to 20 non-salted hashes, one per line:

```
d7dad0a5cd7c8376eeb50d69b3ccd352
```

I'm not a robot

reCAPTCHA
Privacy - Terms

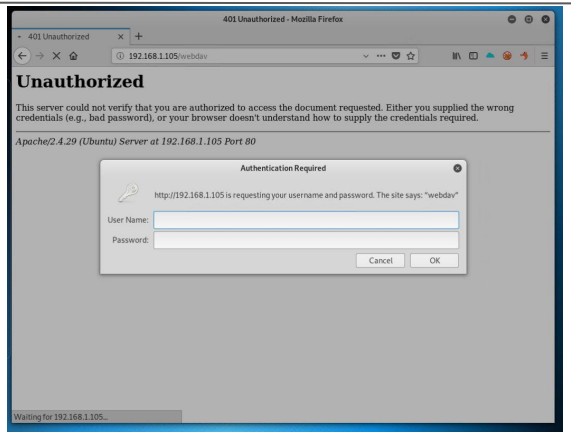
Crack Hashes

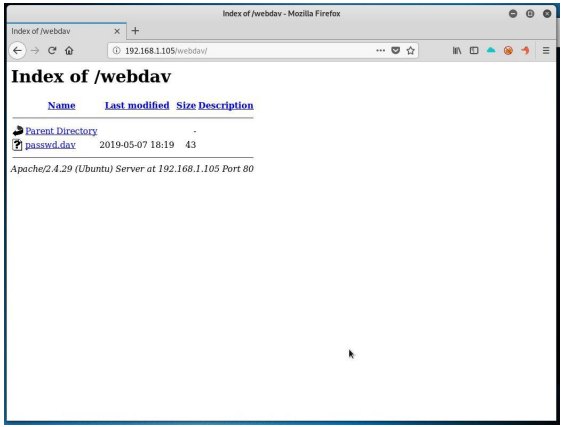
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

Download CrackStation's Wordlist





RFI and LFI Inclusion

01

Tools & Processes
The web server allowed for Remote File inclusion on the page, which allows file uploads to the server remotely with no restrictions in place.
This coupled with Port 80 and Port 22 being open (discovered through NMAP), allows for the use of msfvenom and metasploit to gain access the the server.

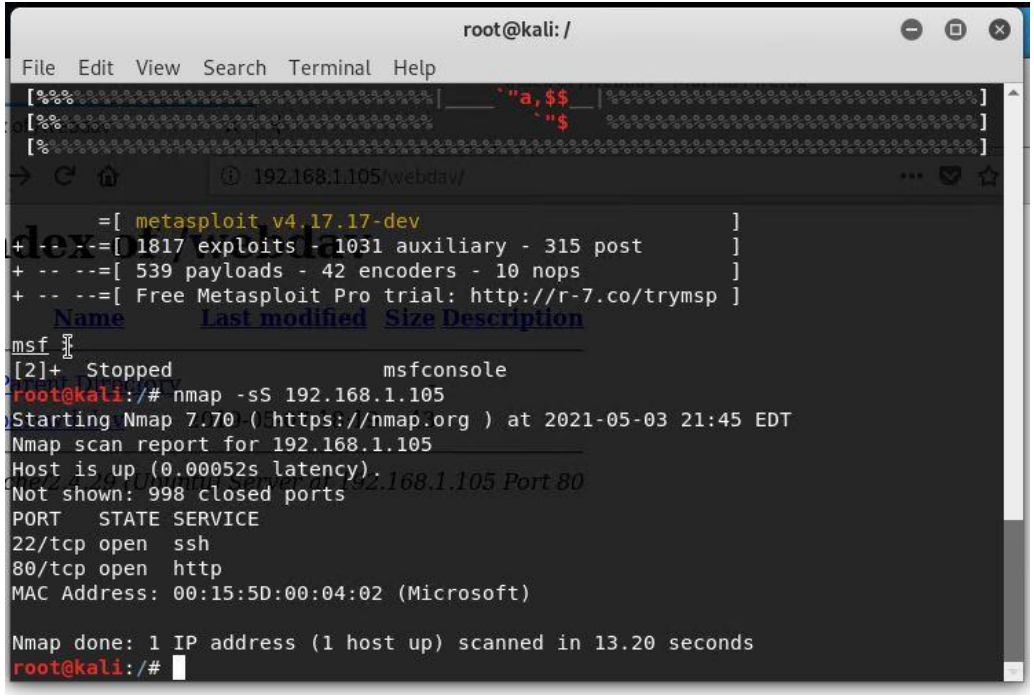
02

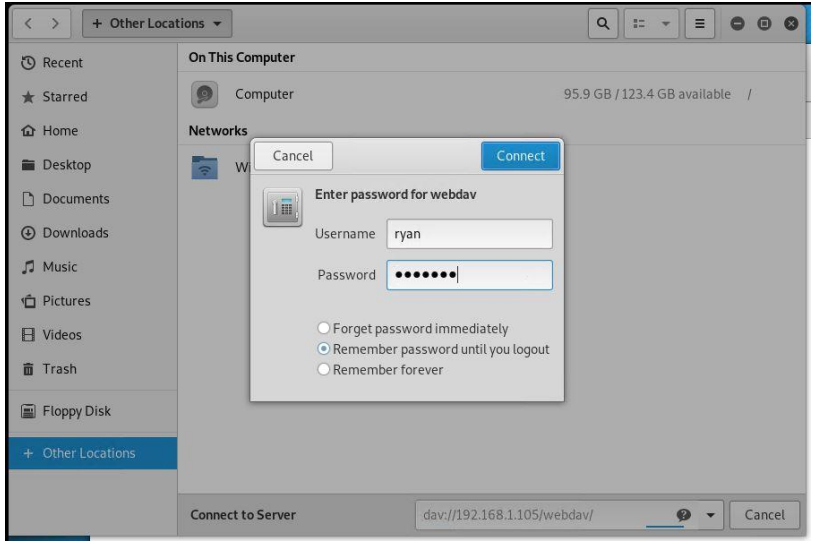
Achievements
We were able to write a malicious script and upload it to the web server, which, when ran, allowed a remote meterpreter session to gain access.

03

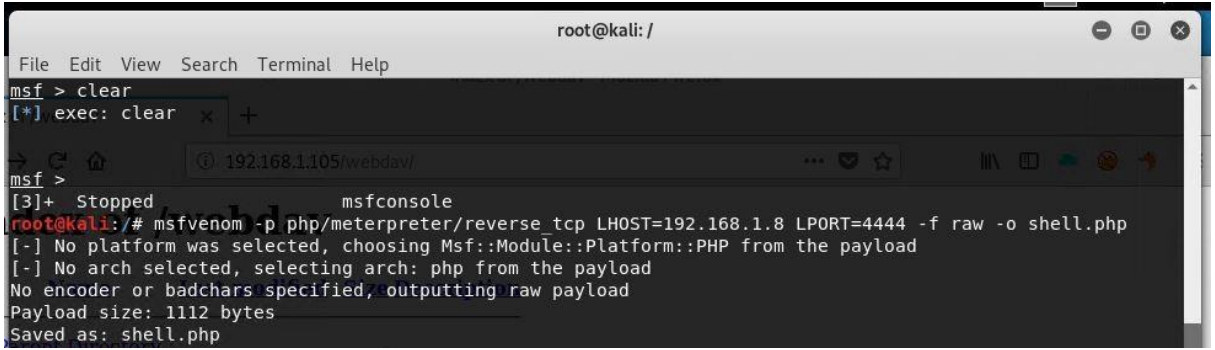
Screenshots on the following pages.

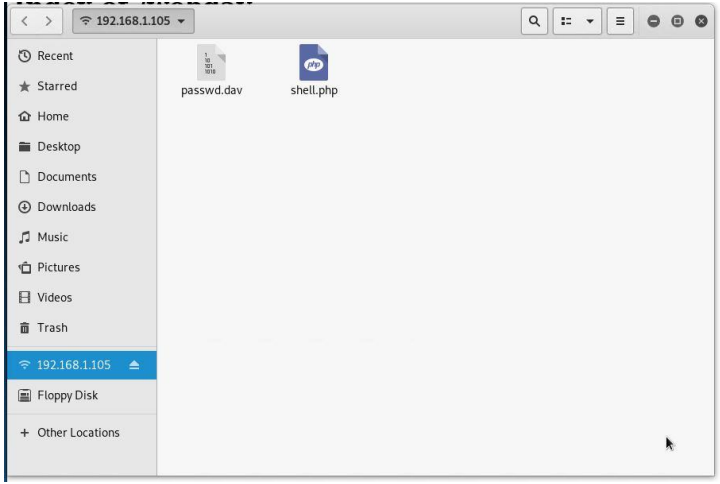
RFI and LFI Inclusion Screenshots

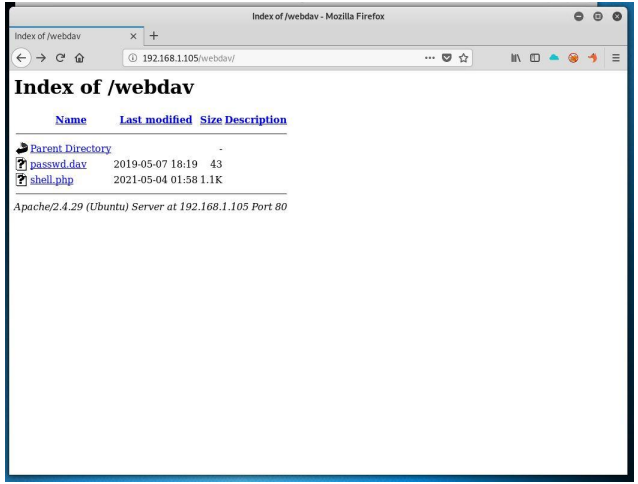




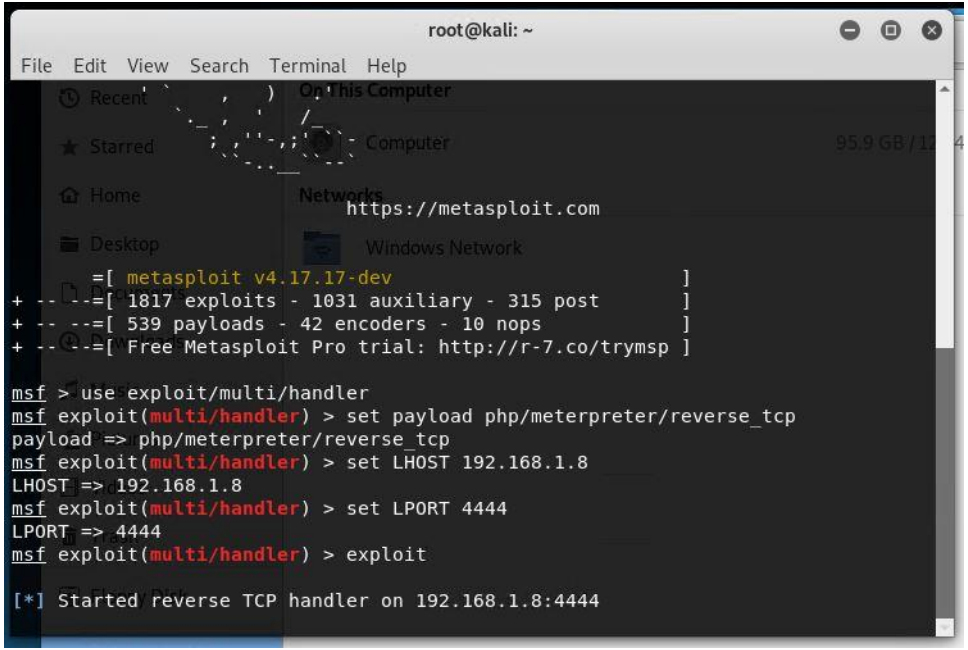
RFI and LFI Inclusion Screenshots

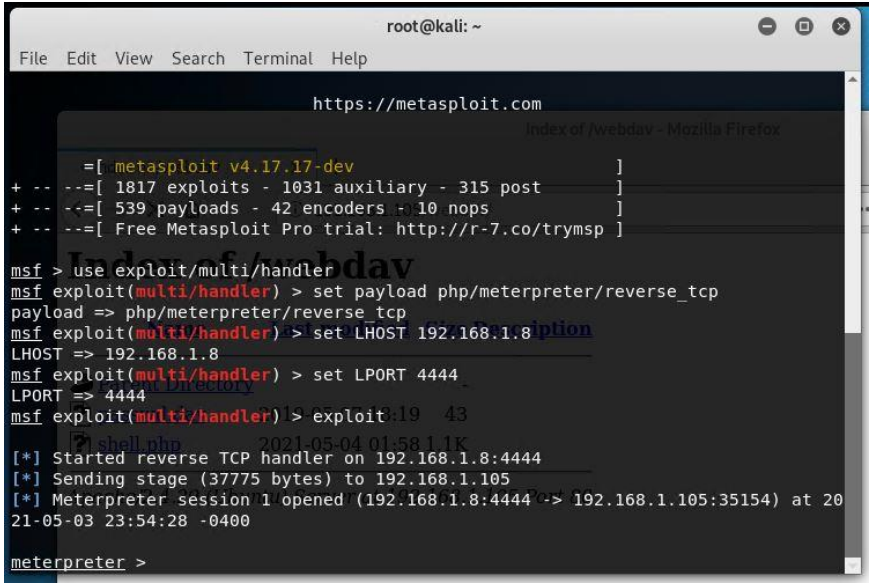




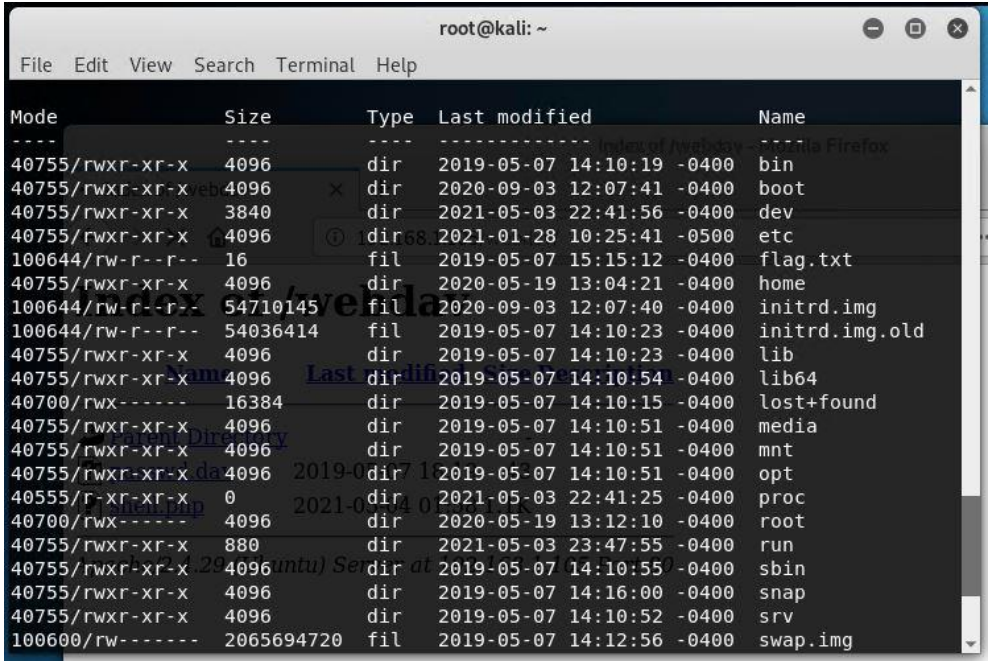


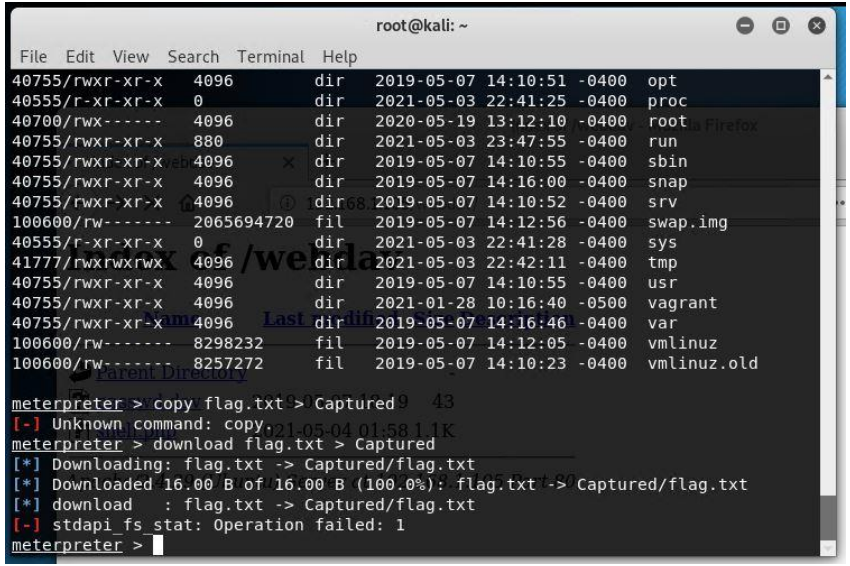
RFI and LFI Inclusion Screenshots



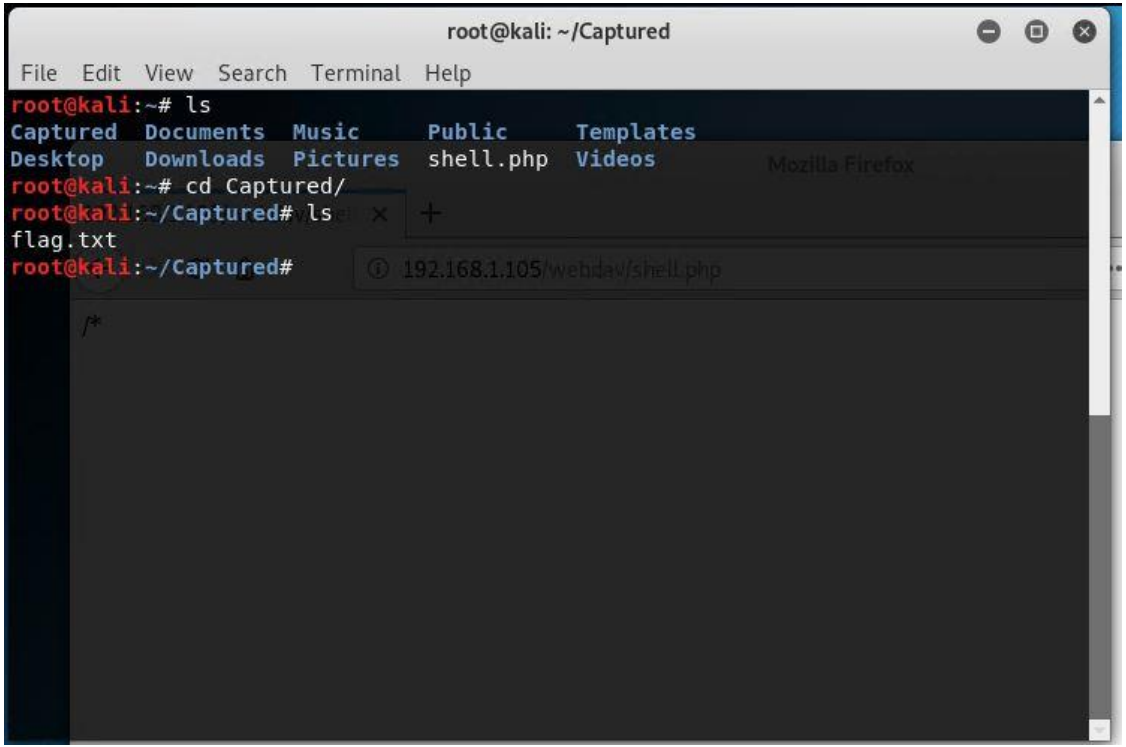


RFI and LFI Inclusion Screenshots





RFI and LFI Inclusion Screenshots



Blue Team

Log Analysis and Attack Characterization

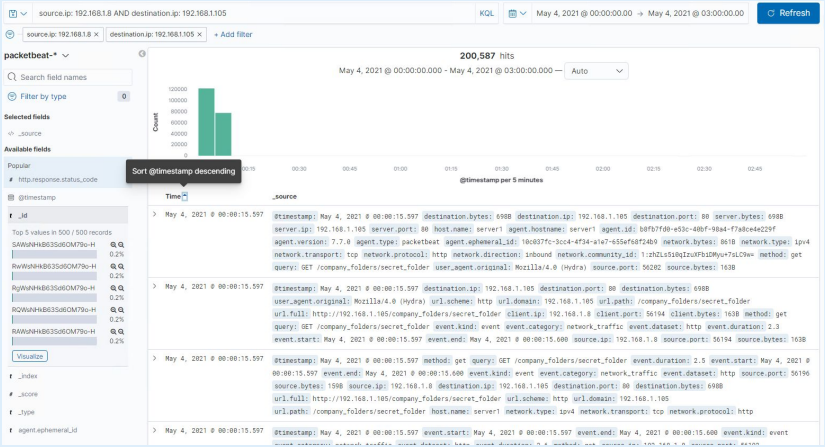
Analysis: Identifying the Port Scan

- What time did the port scan occur?
  - May 3, 2021 21:14
- How many packets were sent, and from which IP?
  - 4 packets were sent and they came from the Source.ip 192.168.1.8
- What indicates that this was a port scan?
  - If you use this query: user_agent.original: *nmap*
  - It will highlight that an Nmap Scripting Engine was being used

> May 3, 2021 @ 21:55:33.763 user_agent.original: Mozilla/5.0 (compatible; Nmap Scripting Engine; https://nmap.org/book/nse.html) @timestamp: May 3, 2021 @ 21:55:33.763 type: http ecs.version: 1.5.0 agent.type: packetbeat agent.ephemeral_id: 89c45770-918b-4bea-b8e3-cb5d95477b04 agent.hostname: server1 agent.id: b8fb7fd0-e53c-40bf-98a4-f7a8ce4e229f agent.version: 7.7.0 url.path: /evox/about url.full: http://192.168.1.105/evox/about url.scheme: http url.domain: 192.168.1.105 status: Error host.name: server1 server.ip: 192.168.1.105 server.port: 80 server.bytes: 467B event.start: May 3, 2021 @ 21:55:33.763 event.end: May 3, 2021 @ 21:55:33.763 event.kind: event event.category: network_traffic event.dataset: http event.duration: 0.2 destination.bytes: 467B destination.ip: 192.168.1.105 destination.port: 80 query: GET /evox/about source.ip: 192.168.1.8

Analysis: Finding the Request for the Hidden Directory

- What time did the request occur? How many requests were made?
  - May 4, 2021 @ 00:00:15.597 is when the interaction first took place
  - Our Dashboard showed the Top 10 HTTP Packet Requests, all 32,579 came from 192.168.1.8
- Which files were requested? What did they contain?
  - The filetype was a _doc file. The file resided in the /company_folders/secret_folder/connect_to_corp_server directory, which held information about how to access one of the servers.



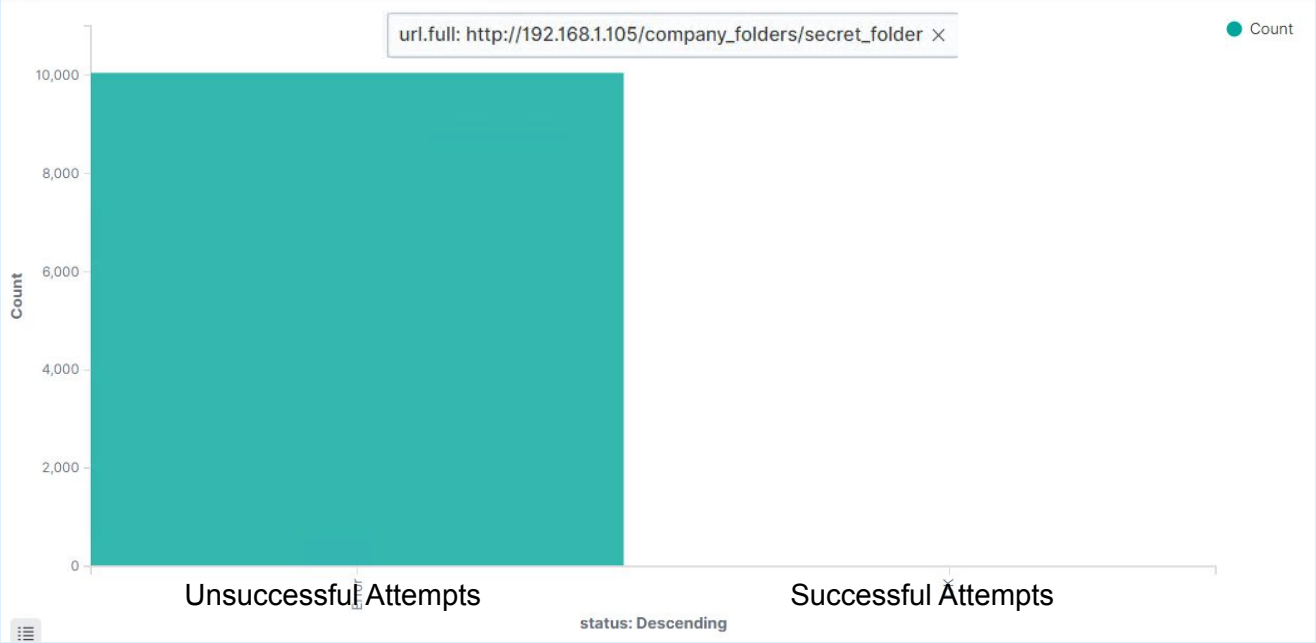
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	32,579
http://192.168.1.105/webdav/shell.php	55
http://192.168.1.105/webdav	46
http://192.168.1.105/webdav/passwd.dav	35
http://192.168.1.105/	14

Export: Raw Formatted

Analysis: Uncovering the Brute Force Attack

- How many requests were made in the attack? 10,047
- How many requests had been made before the attacker discovered the password? 10,044



Analysis: Finding the WebDAV Connection

- How many requests were made to this directory? 126,843
- Which files were requested? webdav/passwd.dav

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	126,843
http://192.168.1.105/company_folders/secret_folder	10,043
http://192.168.1.105/webdav/passwd.dav	71
http://192.168.1.105/	13
http://192.168.1.105/webdav/	11

Export: Raw Formatted

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?

Create a alarm to detect TCP and SYN requests.

What threshold would you set to activate this alarm?

20, Port scanning sends multiple request and this should be enough to alert early on.

System Hardening

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

Set up or use a firewall to block ping requests.

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

Set firewall to only allow access from authorized IPs

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

Create an alarm/alert if anyone access the directory from unauthorized IPs

What threshold would you set to activate this alarm?

1-2 failed attempts that would immediately send a alert to multiple persons

System Hardening

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

Block access or ensure proper access controls are in place

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

Remove from webserver and place on internal server without outside access. “Chmod +” to change r/w/x for user, owner, and group

Mitigation: Preventing Brute Force Attacks

Alarm

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

Create an alarm that detects a abnormal amount of 401 status codes on the server

What threshold would you set to activate this alarm?

Would set threshold to 7

System Hardening

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

Create a user lockout threshold for too many failed logins

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

Set number of failed attempts to 3 and require account to be unlocked before allowing access.

Mitigation: Detecting the WebDAV Connection

Alarm

What kind of alarm can be set to detect future access to this directory?
Create an alarm for unauthorized connections via WebDAV

What threshold would you set to activate this alarm?
Would set threshold to 1

System Hardening

What configuration can be set on the host to control access?
Only allow authorized connections and/or block all outside connections with a firewall or disable WebDAV if not being used

Describe the solution. If possible, provide the required command line(s).
Disabling WebDAV if unused or setting up a firewall for controlled access would limit potential attacks. Remove directory.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

Create an alert that notifies when a file or folder changes or is altered

What threshold would you set to activate this alarm?

Any activity that involves uploading files would activate this alarm.

System Hardening

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

Require authentication to upload files.
Restrict specific file types

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

Requiring authentication to upload files would mitigate potential of attacks.

the End