Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

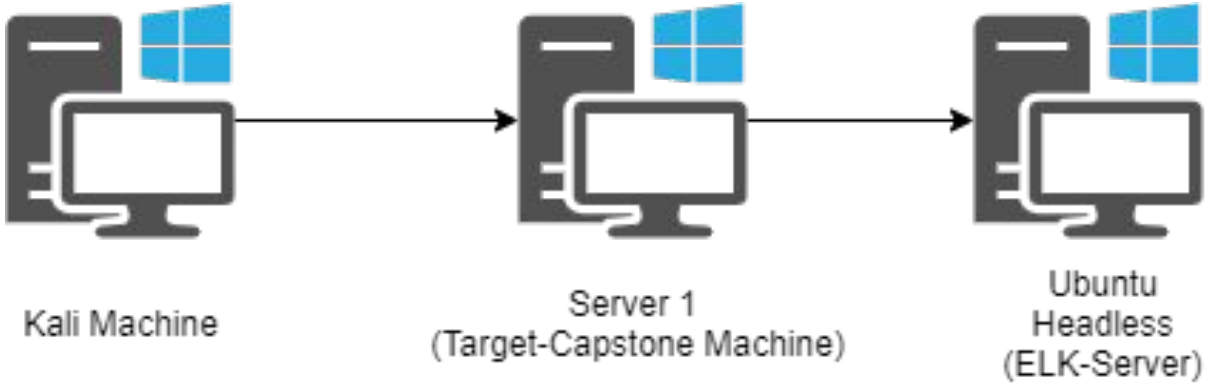
Bryan Mussato

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address
Range:192.168.0.1/24
Netmask:
255.255.255.240
Gateway:192.168.1.1

Machines
IPv4:192.168.1.8
OS:Kali
Hostname:Kali

IPv4:192.168.1.100
OS:Ubuntu 18.04.3
Hostname:
Ubuntu Headless (ELK)

IPv4:192.168.1.105
OS:Linux
Hostname:Server 1
(Target-Capstone)

Red Team
Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Kali	192.168.1.8	Attacker Machine
Ubuntu Headless	192.168.1.100	Network Log Monitoring
Server 1	192.168.1.105	Target Machine

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Directory List	The Directory List is viewable via using command line injection into a browser.	This can reveal sensitive data unless it requires extra authentication.
Weak Password or Weak Hash	Using a weak password such as: password, 123456, or anything that can easily be guessed.	An attacker can easily brute force the password using a common wordlist in a short amount of time.
Reverse Shell Attack	This attack uses a listener to lure the target machine into opening a file that will give the attacker access to the machine and allow control despite firewalls.	If shell.php was opened it allows the attacker to use a shell on the target machine and have access to any file on that machine.

Exploitation: Directory List

01

Tools & Processes

After a port scan using NMAP, I could access some of the directories using a web browser and command injection. The text files located in the directory showed a secret_folder that could be accessed using a password.

02

Achievements

This exploit allowed me to see information that led me to a secret folder. I also was able to find a path to upload the shell.php file to the directory.

03

Exploitation: Weak Passwords and/or Weak Hashes

01

Tools & Processes

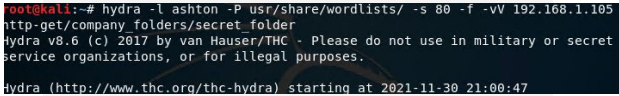
Using Hydra and a custom wordlist I strung these two together on a command line and ran it against the hash that I found.

02

Achievements

This exploit cracked the password using a brute force technique using hydra. This gave me access to a password protected directory.

03



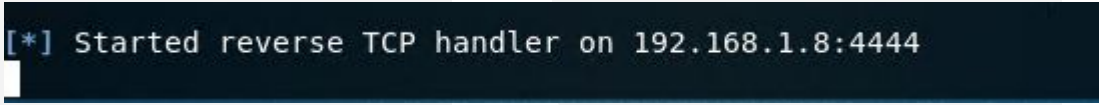
Exploitation: Reverse TCP Handler Exploit

01

Tools & Processes
Using a msfconsole instance on the Kali machine I was able to run a process to monitor the

02

Achievements
What did the exploit achieve? For example: Did it grant you a user shell, root access, etc.?



03

Blue Team
Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The Request began at 01:20:20 on November 24th, 2021
- 4 packets were sent to the target machine
- This port scan is indicated by the use of port 80 and that it is qualified as network flow

Analysis: Finding the Request for the Hidden Directory

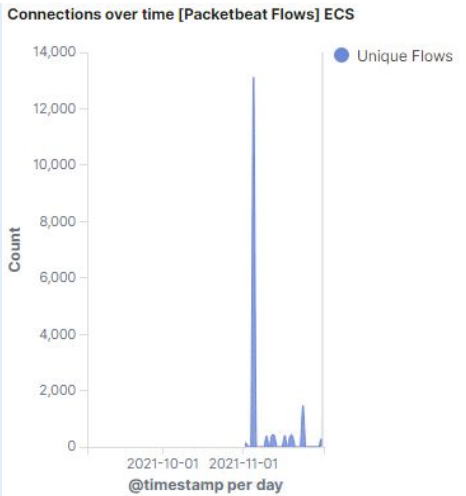
- The request for the hidden directory was made November 5th, 2021 at 13:03:02. These files contained the important data used for logins on the WebDav directory.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	10,015
http://127.0.0.1/server-status?auto=	5,473
http://192.168.1.105/webdav/shell.php	33
http://192.168.1.105/webdav	23
http://169.254.169.254/2014-02-25/dynamic/instance-identity/document	19

Analysis: Uncovering the Brute Force Attack

- There were 10,537 requests total.
- 10,536 requests before the password was discovered.



Analysis: Finding the WebDAV Connection

- There were 33 total requests made to this directory.
- In this directory there was a .,php file called shell.php that was used for this attack.

> Nov 24, 2021 @ 01:27:41.233

@timestamp: Nov 24, 2021 @ 01:27:41.233 http.request.method: get
http.request.referrer: http://192.168.1.105/webdav/ http.request.bytes: 415B
http.request.headers.content-length: 0 http.response.status_phrase: ok
http.response.status_code: 200 http.response.bytes: 204B
http.response.body.bytes: 2B http.response.headers.content-length: 2

Blue Team
Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

Create an alert that will trigger when a scan is used against the network. This alert should trigger after 1 event to make sure it is contained and assessed.

System Hardening

Close ports and use a firewall. For example using firewalld in a command line argument: firewalld-cmd --permanent --remove-port=80/tcp

Mitigation: Finding the Request for the Hidden Directory

Alarm

An alarm or tripwire could be set for the directory to trap the attacker and report the IP address.

System Hardening

Removing the directory from the web would be the first thing to do. If that isn't possible than setting a new password to the whole directory and giving access to only privileged users. Finally setting a tripwire in the directory would help to block the attacker before they can do damage.

Mitigation: Preventing Brute Force Attacks

Alarm

Setting an alarm in Splunk to report when someone is brute forcing passwords and setting the threshold to under 5.

System Hardening

Setting up an alert through splunk that will send an email to the security team would help to harden against this vulnerability. Another way is setting requirements on the complexity of passwords and changing those passwords often.

Mitigation: Detecting the WebDAV Connection

Alarm

Setting an alarm that triggers anytime the directory is being accessed from outside the IP range of the company devices.

System Hardening

Similarly to the hidden directory, setting an alarm for any outside IP addresses and only allowing certain IP addresses to the WebDav connection would harden it. Also implementing another tripwire would be an alternative to the first solution.

Mitigation: Identifying Reverse Shell Uploads

Alarm

Set an alert for any traffic coming through port 4444 and set alerts for any unknown file coming into the directory.

System Hardening

-Uploads should require authorization from an admin.
-The server admin should restrict and ban any executable files from being uploaded.
-Disable ssh by commenting out port 22 in the sshd_config file.


The End