



Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

01

Network Topology

02

Red Team: Security Assessment

03

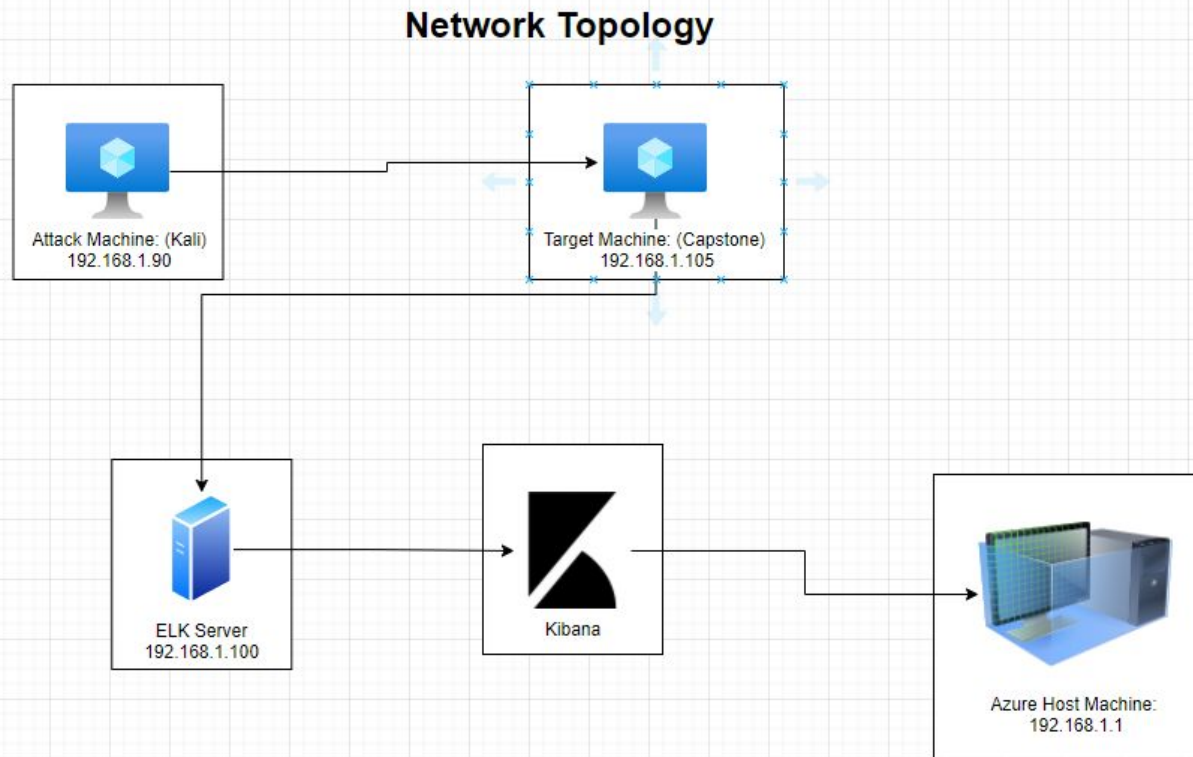
Blue Team: Log Analysis and Attack Characterization

04

Hardening: Proposed Alarms and Mitigation Strategies

Network Topology

Network Topology



Network

Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 10.0.0.76

Machines

IPv4: 192.168.1.1
OS: Windows 10
Hostname: Azure Hyper-V
ML-RefVm-684427

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

The background of the slide is a dark red, almost black, field filled with a complex, repeating geometric pattern of triangles and polygons in various shades of red and maroon, creating a textured, crystalline effect.

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Capstone	192.168.1.105	Target Machine
ELK	192.168.1.100	Security Monitoring
Kali	192.168.1.90	Attack Machine
ML-REFVM-684427	192.168.1.1	Azure Host/Cloud Based Host

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Firewall Rules: Open Ports	<ul style="list-style-type: none">-Port 22 and 80 were found, via an Nmap scan, to be open.-Several other ports were open including ports 2222 and 4444	<ul style="list-style-type: none">-Open Ports are a potential security problem because it increases the attack surface of machine/network
Weak Password Policy (CVE-2019-3746)	<ul style="list-style-type: none">- Passwords were short, simple, and changed infrequently.-Password hashes were not salted.	<ul style="list-style-type: none">-Hydra was used to brute force access to accounts.-Password hashes were able to be decrypted.
Weak Authentication Controls	<ul style="list-style-type: none">-Only usernames and passwords were required to access accounts. There was no multi-factor authentication.	<ul style="list-style-type: none">-Used the passwd.dav file and msfvenom to brute force an account and access the web server.

Exploitation: Firewall Rules: Open Ports

01

Tools & Processes

I used **nmap** to scan for open ports on the target machine.

Netdiscovery could also be used.

02

Achievements

This showed me the available/open ports to be exploited. Ports 80 and 22 being open were of interest.

03

```
root@Kali:~# nmap -sV 192.168.1.105
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-18 16:10 PDT
Nmap scan report for 192.168.1.105
Host is up (0.00055s 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 6.96 seconds
root@Kali:~#
```


Exploitation: Weak Password Policy

01

Tools & Processes

The command-line tool **Hydra** was used to bruteforce a user's account on the specified machine. The password hashes were able to be uploaded to a password cracking website (**CrackStation**) with a built-in wordlist. Alternatively, one could use **John the Ripper** with any popular wordlist such as "rockyou.txt".

```
[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 2021-10-13 16:52:12
root@Kali:/usr/share/wordlists#
```

Achievements

First access to the **/secret_folder** was gained through cracking ashton's password 'leopoldo' with **hydra**. Through the use **CrackStation** and the hash, the password 'linux4u' with username 'ryan' was used to access the **/webdav** folder.

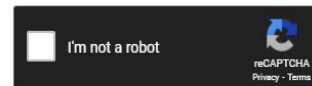
02

03

d7dad0a5cd7c8376eeb50d69b3ccd352

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), ZubesV3.1BackupDefaults

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u



Crack Hashes

Exploitation: Weak Authentication Controls

01

Tools & Processes

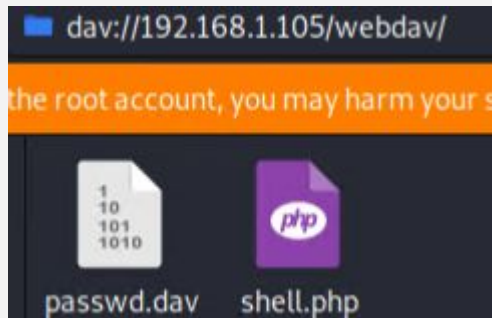
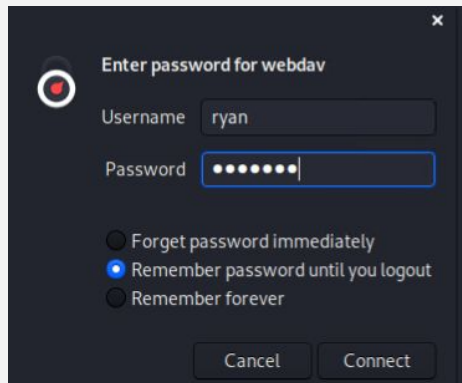
Brute forced access to Ryan's account by decrypting the MD5 hash then used **msfvenom** to run a php reverse shell meterpreter session.

02

Achievements

Accessed company's web server through *Ryan's* account. Uploaded a malicious script to the WebDAV.

03





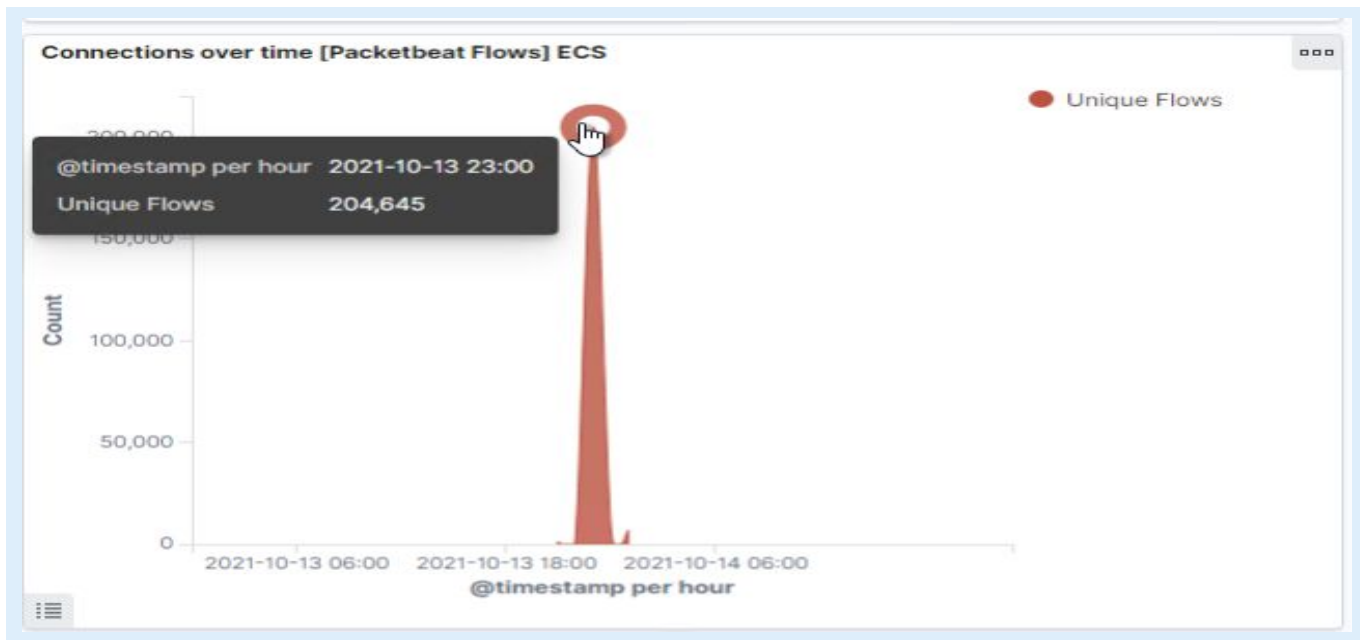
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



- The scan occurred on October 13, 2021 at 2300.
- 204,645 connections occurred from host 192.168.1.90.
- The intense and sudden peaks of traffic signify a port scan.



Analysis: Finding the Request for the Hidden Directory

- There were 15,546 requests made to the /secret_folder directory on 10-13-2021 @23:00 from IP 192.168.1.90.
- The connect_to_corp.txt, which contained detailed instructions of how to connect to the company webdav WITH Ryan's credentials (a hash is also provided).

Top 10 HTTP requests [Packetbeat] ECS

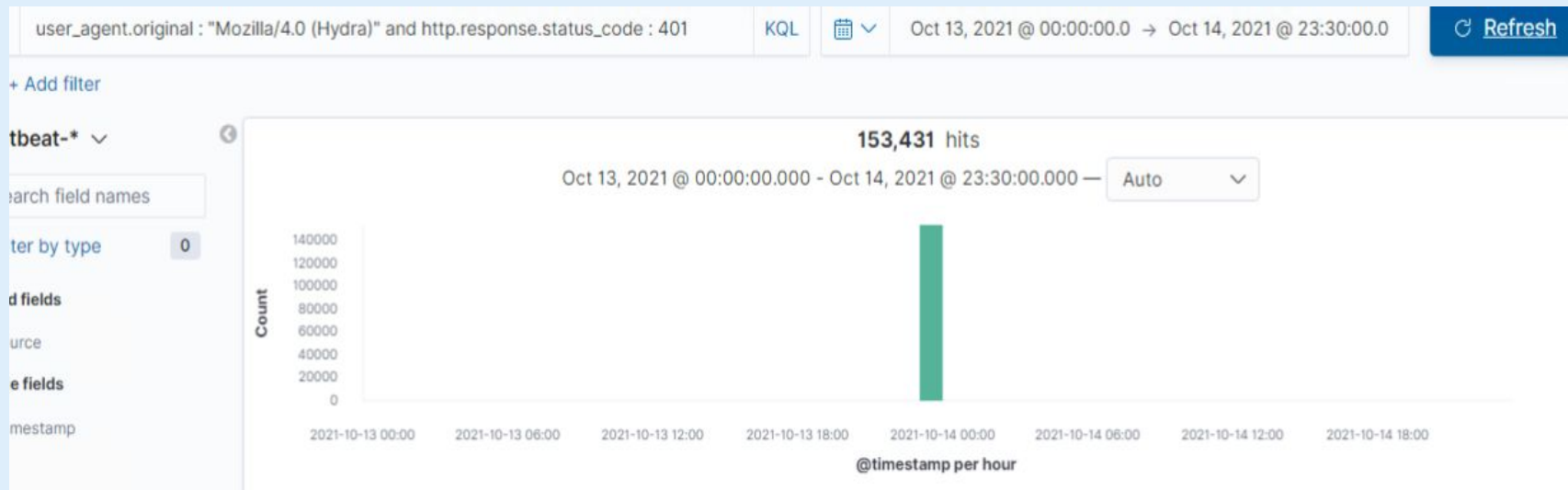
url.full: Descending ▾	Count ▾
http://192.168.1.105/webdav	138,326
http://192.168.1.105/company_folders/secret_folder	15,546
http://127.0.0.1/server-status?auto=	1,211
http://snnmnkxdhflwgthqismb.com/post.php	162
http://www.gstatic.com/generate_204	84

Export: [Raw](#) 📄 [Formatted](#) 📄

Analysis: Uncovering the Brute Force Attack



- There were 15,541 total requests.
- There were 15,531 requests made before the attacker discovered a password.





Analysis: Finding the WebDAV Connection



- There were 138,326 requests made to the /webdav directory.
- The requests were mainly for the passwd.dav and shell.php files. These contained the password for access to the WEBDAV and a malicious script to start a shell.

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	138,326

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?

- Use an IDS product to monitor, alert and log potential port scans and suspicious traffic.
- Set alert to alarm when a high amount of port communications from a single IP address occur in a small window.

What threshold would you set to activate this alarm?

- More than 500 port communications from single IP address in a 60 second time frame.

System Hardening

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

- Use iptables to configure connection policies.
- IDS product to alert, log and potentially automatically block IP address attempting to listen on several ports.

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

- Configure iptables policy on the local machine for suspicious IP addresses: example:

```
$ iptables -A INPUT -p tcp --dport ssh -s 192.168.1.90 -j DROP
```

Mitigation: Finding the Request for the Hidden Directory

Alarm

An alarm should be set anytime unauthorized access is requested for hidden/confidential resources.

Anytime there are 5 or more attempts in an hour.

System Hardening

Rename resources to something less obvious (for example, secret_folder tells on itself), encrypt critical data so only those with a key can access it. Keep tabs on each IP that sends a request for these directories/files to either safelist or denylist.

Commands for iptables:

Safelist - `$ iptables -s <IP_Address_to_whitelist> -p tcp -m multiport --dports <chosen/accessible_destination_ports> -j ACCEPT`

Denylist - `$ iptables -A INPUT -s IP-ADDRESS -j DROP`

Mitigation: Preventing Brute Force Attacks

Alarm

An alarm can be set to recognize when a 401 error is returned. This often signifies a bruteforce attack since it refers to a lack of valid credentials for a target resource.

While a 401 error in general is suspicious, I would set a threshold of 10 or more errors are made within an hour.

System Hardening

What configuration can be set on the host to block brute force attacks? Better password policy can help mitigate brute force. A policy that locks out for 30 minutes after 5 unsuccessful logins would work.

Also be sure to increase password complexity requirements for all employees, especially those with access to critical resources (like the WebDAV).

Mitigation: Detecting the WebDAV Connection

Alarm

Anytime an unknown/trusted IP makes a GET request to the **/webdav** folder, send an alert out.

Theoretically, since all IPs that are trusted should be on the whitelist, the threshold would be when any number of GET requests are made by a non-whitelisted IP for the **/webdav** folder.

System Hardening

Establish/Create and maintain a whitelist of trusted IPs and set a firewall policy to block ALL other traffic. Also ensure users with access to the WebDAV folder have complex usernames and passwords.

Command to create iptable: *\$ iptables -s <IP_Address_to_whitelist> -p tcp -m multiport --dports 80,443 -j ACCEPT*

Mitigation: Identifying Reverse Shell Uploads

Alarm

An alarm should be set to alert anytime a file is uploaded to the webdav server from a non-whitelisted IP.

This alarm should be activated any time one (1) or more files are uploaded from a non-whitelisted IP address.

Another alarm should be set to notify when the webdav server reaches out for a connection. Web servers should have incoming traffic, not outgoing traffic.

System Hardening

Configurations to set on the host to block file uploads:

- Set permissions to block access to the file upload system from any IP which is not whitelisted.
sudo chmod 770 <file_upload_directory>
- Restrict types of files which can be uploaded ex .php, .xss, .rar
- Set firewall rules to block incoming traffic from all ports other than those specified.
- Block inbound/outbound traffic to suspicious ports such as 4444.

*The
End*