Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Neil Prendergast

Table of Contents

This document contains the following sections:

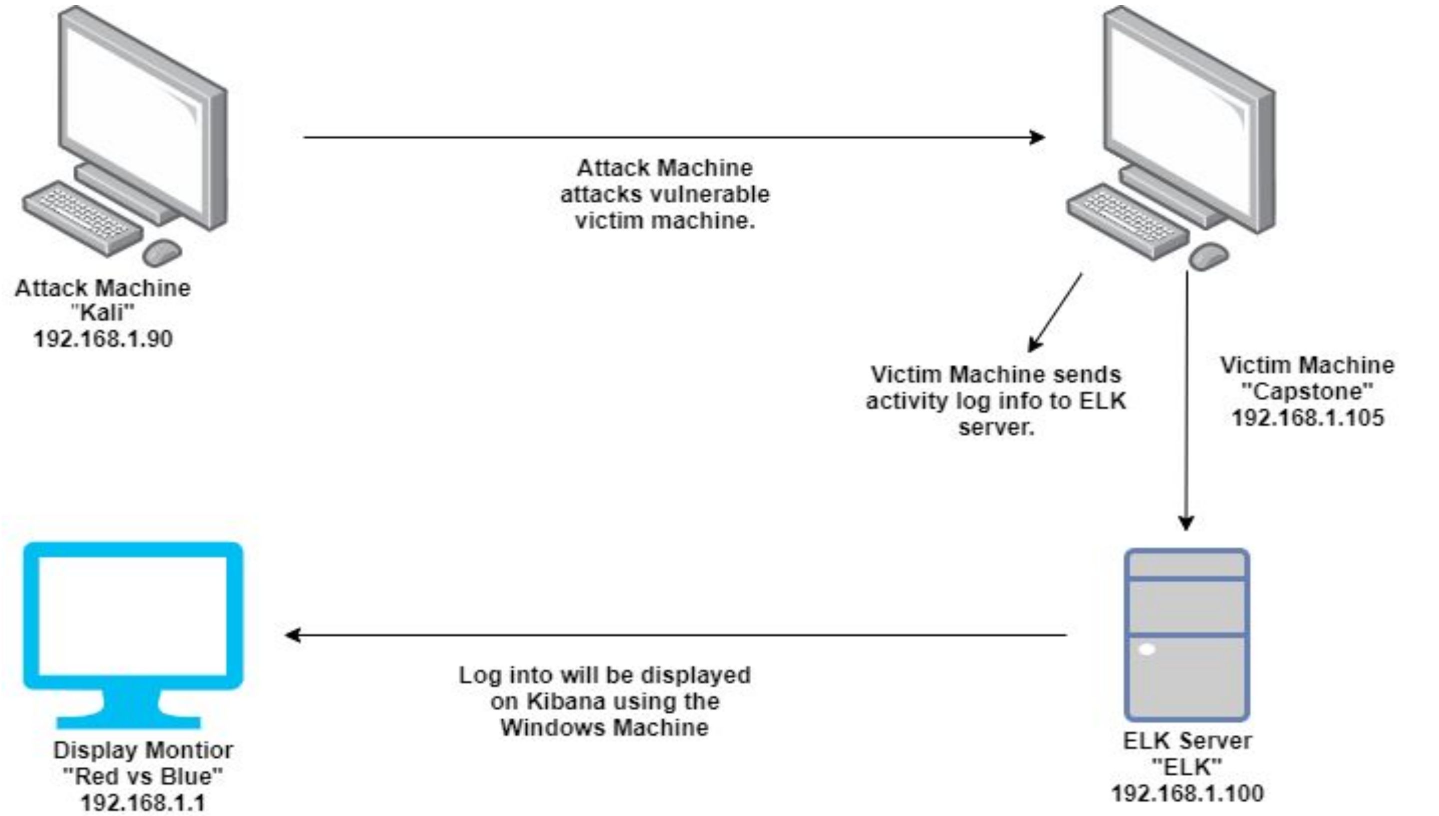
01 **Network Topology**

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

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

Network Topology

Network Topology


Attack Machine attacks vulnerable victim machine.
Attack Machine "Kali" 192.168.1.90
Victim Machine sends activity log info to ELK server.
Victim Machine "Capstone" 192.168.1.105
Log into will be displayed on Kibana using the Windows Machine
Display Montior "Red vs Blue" 192.168.1.1
ELK Server "ELK" 192.168.1.100

Network
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Linux 2.6.32
Hostname: Kali Linux

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.0.100
OS: Linux
Hostname: ELK

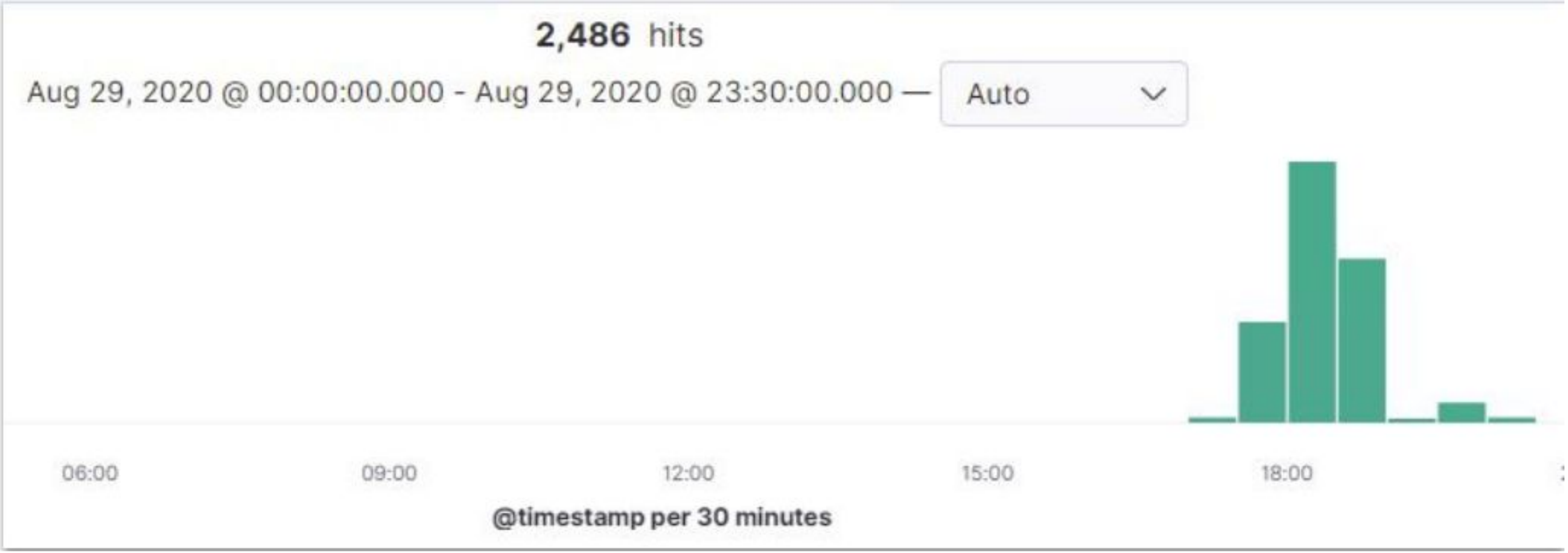
IPv4: 192.168.1.1
OS: Windows
Hostname: Red vs Blue - ML-REFVM

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The Port scans started at 17:30
- There were 2,486 hits to 192.168.1.90
- Nmap is scanning port 443, and filtering for that, we see the results below

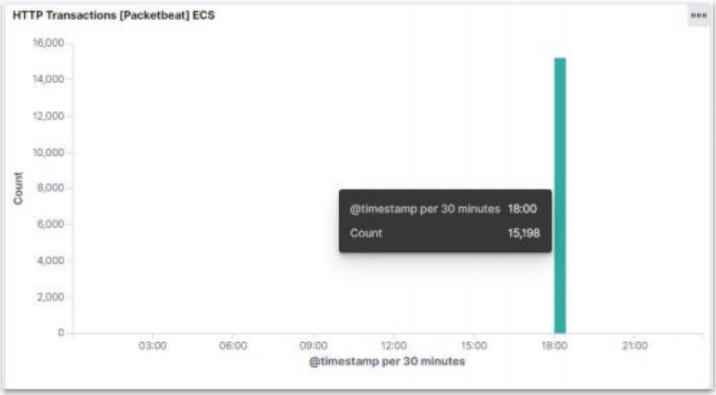


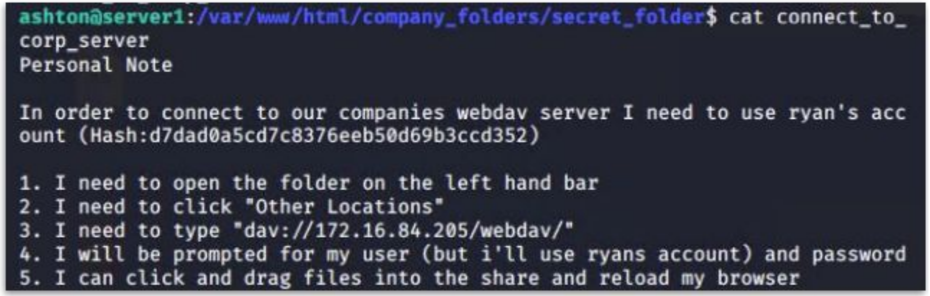
2,486 hits

Aug 29, 2020 @ 00:00:00.000 - Aug 29, 2020 @ 23:30:00.000 — Auto

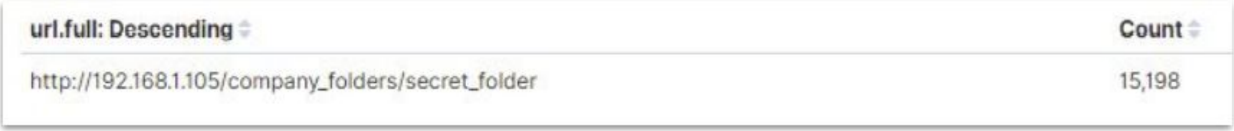
Analysis: Finding the Request for the Hidden Directory

- 15,168 requests for the hidden directory occurred at 6 PM.
- The file requested was a secret folder hidden within the company's folders.
- The hidden folder contained information for how to access the webdav server using an employee, Ryan's, account.





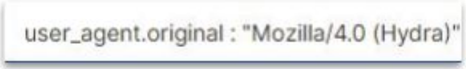
```
ashton@server1:/var/www/html/company_folders/secret_folder$ cat connect_to_
corp_server
Personal Note

In order to connect to our companies webdav server I need to use ryan's acc
ount (Hash:d7dad0a5cd7c8376eeb50d69b3ccd352)

1. I need to open the folder on the left hand bar
2. I need to click "Other Locations"
3. I need to type "dav://172.16.84.205/webdav/"
4. I will be prompted for my user (but i'll use ryans account) and password
5. I can click and drag files into the share and reload my browser
```



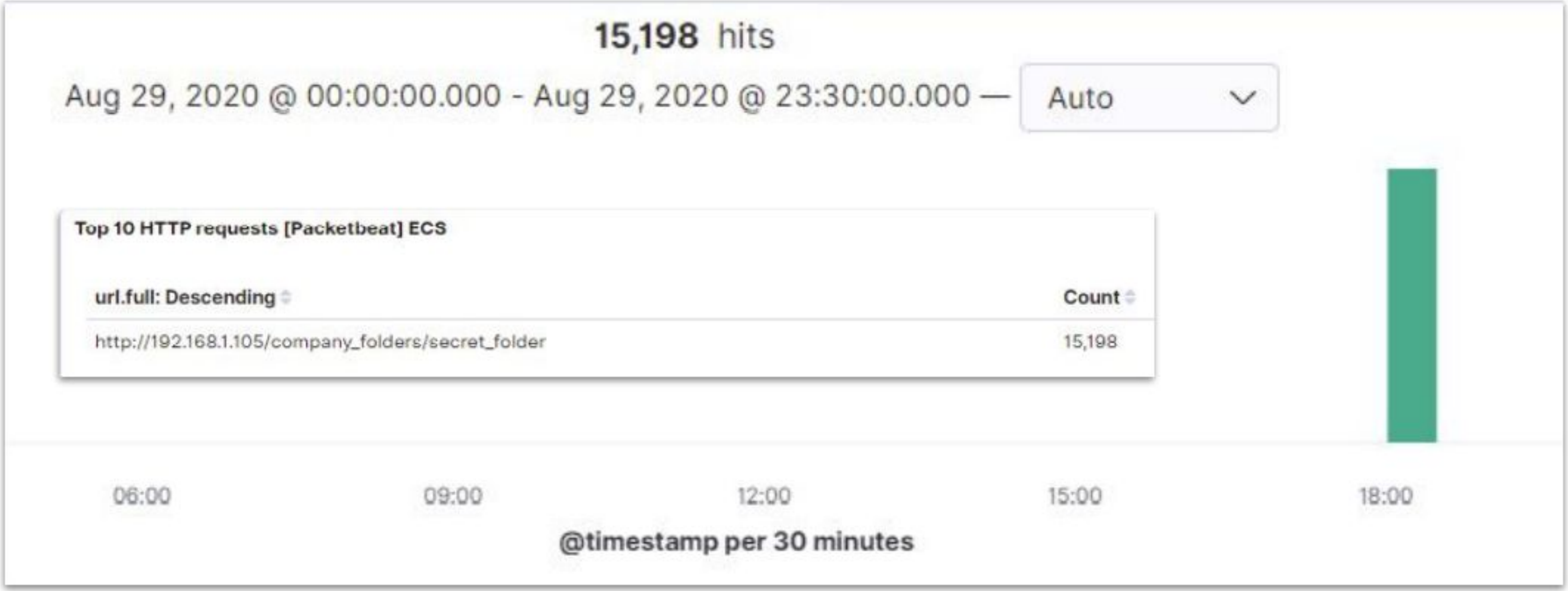
url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,198

Analysis: Uncovering the Brute Force Attack

- **15,198 requests were made in the brute force attack**



user_agent.original : "Mozilla/4.0 (Hydra)"



15,198 hits

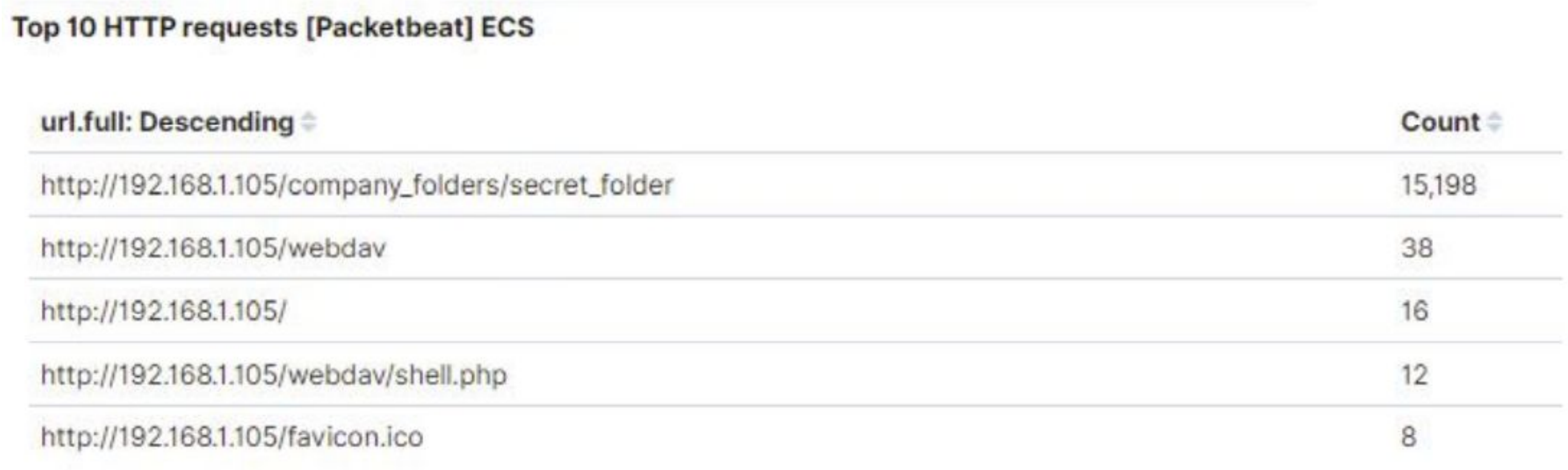
Aug 29, 2020 @ 00:00:00.000 - Aug 29, 2020 @ 23:30:00.000 — Auto

Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,198

Analysis: Finding the WebDAV Connection

- 38 requests were made to the WebDav directory.
- The shell.php file was requested. This was part of the red team's shell attack to start listening for activity on the victim machine.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,198
http://192.168.1.105/webdav	38
http://192.168.1.105/	16
http://192.168.1.105/webdav/shell.php	12
http://192.168.1.105/favicon.ico	8

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

An alarm could be tripped for when a firewall detects more than 10 port scans or 100 consecutive ICMP requests.

You could employ a firewall, or IPS, which could be tailored or tuned to detect and cut off these types of attacks in real time.

System Hardening

Explicitly allow only traffic and protocols that need to access your hosts. This would cover a standard range of ports and request types.

Mitigation: Finding the Request for the Hidden Directory

Alarm

If the directory were required, and files, were required we could set an alert off when any attempt is made to access those files. Someone would then need to follow-up and validate that access attempt.

Alerts would trigger in excess of of one attempt.

System Hardening

If the directory files aren’t needed they could be removed.

rm -r ../company_files to remove the files/

If they are needed you could move the files to a safer or more secure location, and if there were a legitimate reason for storing that information it could be encrypted.

Mitigation: Preventing Brute Force Attacks

Alarm

We could alert on event messages such as ‘401 unauthorized’. We would need to set a reasonable amount of attempts per hour, such as 10, and refine it from there as needed or based on events.

Check for values such as ‘Hydra’ being passed in user_agent.original.

System Hardening

Randomly pad authentication or access attempts with some timers to slow down authentication attempts and possibly break automated tools. Add a captive portal after successive failed logons forcing the user to answer a question, such as the color of a square on screen, or solve a simple math problem.

You could also drop the IP involved in the attack for a limited amount of time.

Mitigation: Detecting the WebDAV Connection

Alarm

An alert could be triggered any time that this directory is accessed and it is NOT a particular set of security principles accessing it. You could configure webdav to use an alert based on which OS user connects to it. WebDav is capable of using impersonation, so you could determine which users/contexts need to access it, and send alerts when it's any context not qualified on that list.

One attempt by security principles outside that set would trigger an alarm.

System Hardening

Since this WebDav directory is on a Linux system, which isn't using IIS or NTLM you would need to place an an authentication service in front of it, as WebDav has no native authentication capability.

You could also force WebDav to work over port 443 exclusively or route it through an SSH server using SSH tunneling.

Mitigation: Identifying Reverse Shell Uploads

Alarm

Set an alert that triggers based on attempts to upload a particular file type, or if repeated attempts to upload files fail from a particular host or IP address.

The threshold could exceed one attempt.

System Hardening

Restrict uploads to specific file extensions

Only allow authenticated specific users to use the feature.

Write to the file on storage to include a random header that makes it non-executable.

The End