Document Name	Brute-Force Protection (Ubuntu Web Server)	Version	1.3
Author	Anusha Ramu Chakravarthi	Date Created	24/04/2025
Protection Type	Brute-Force Defense (RDP and SMB Services)	Last Modified	27/042025

Document Description

This playbook outlines the implementation of brute-force protection for the Ubuntu 22.04 web server using **Fail2Ban**. It focuses on protecting SSH, Nginx, and WordPress (through PHP and web login attempts). This aligns with the **Protect** function of the NIST Cybersecurity Framework and ensures the security of publicly exposed services.

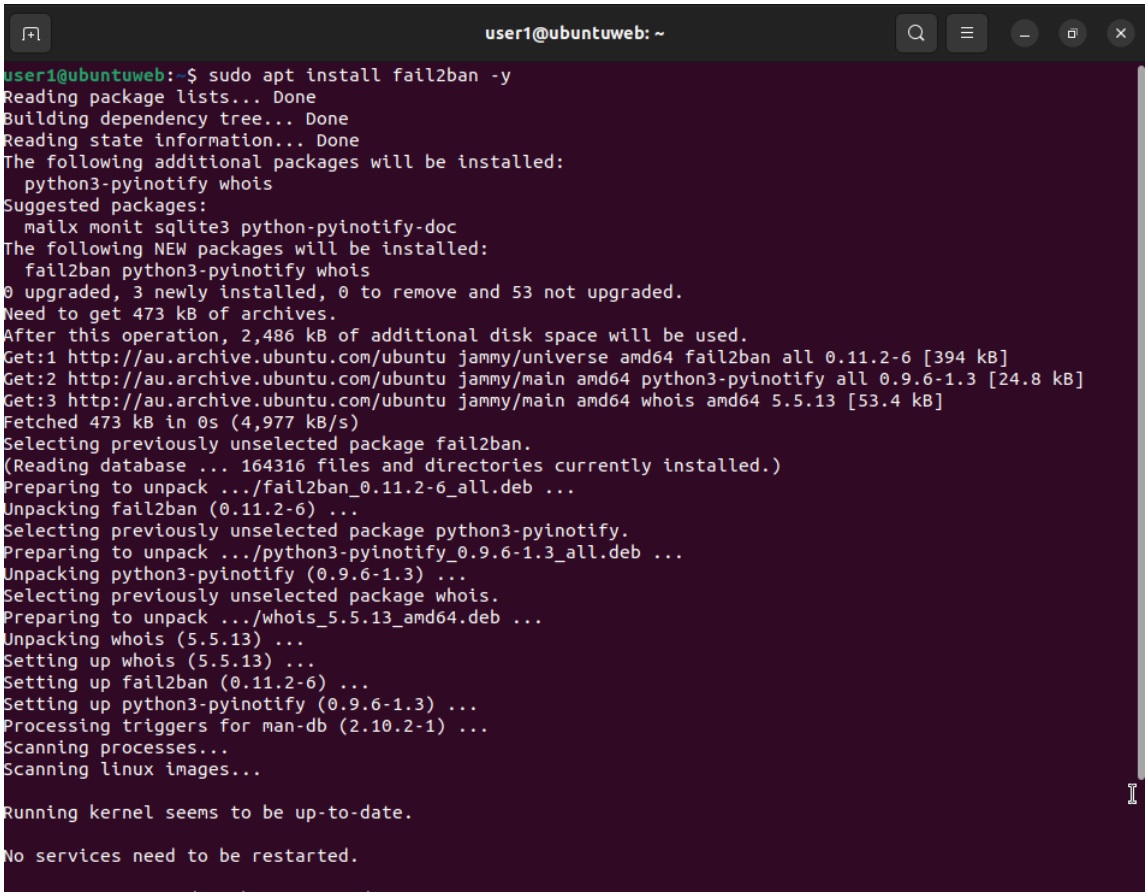
Step 1

Install Fail2Ban on Ubuntu 22.04

Install the Fail2Ban package from the official repositories.

- **Run**:

```
sudo apt update && sudo apt install fail2ban -y
```



- Ensure the service is enabled and running:
```
sudo systemctl enable fail2ban && sudo systemctl start fail2ban
```

Step 2

Protect SSH Service

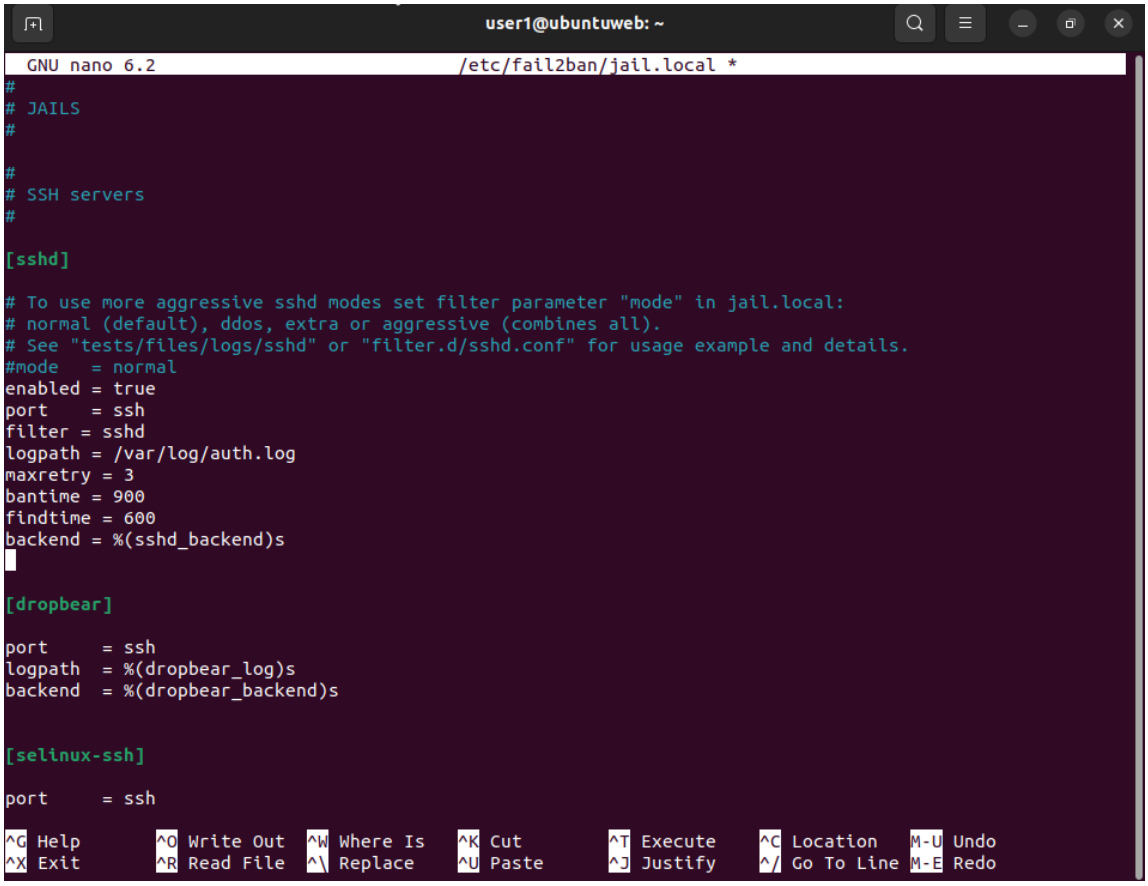
Configure Fail2Ban to protect SSH against brute-force attacks.

- **Copy default config:**
```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

- **Edit the jail file:**
```
sudo nano /etc/fail2ban/jail.local
```

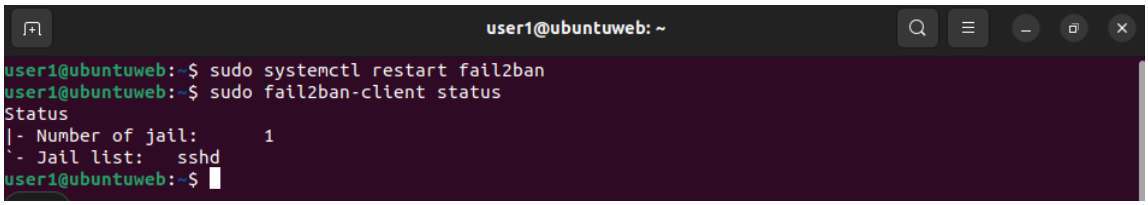
- Enable and configure the [**sshd**] jail:

```
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 3
bantime = 900
findtime = 600
```



- Save and restart:

```
sudo systemctl restart fail2ban
```



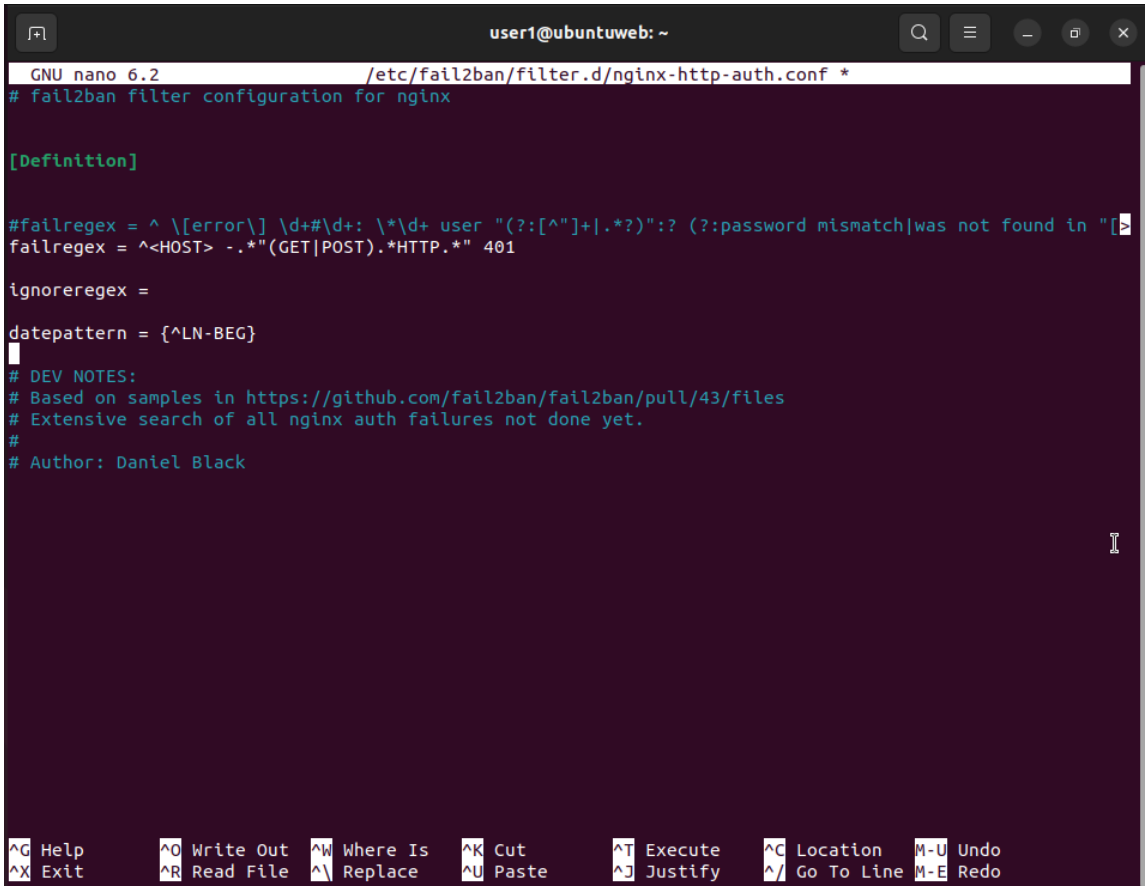
Step 3
Enable Nginx Brute-Force Protection

- Configure protection for **Nginx HTTP/HTTPS services**.
  Create **custom filter**: /etc/fail2ban/filter.d/nginx-http-auth.conf
- Add rules for detecting failed login attempts via Nginx logs.

```
sudo nano /etc/fail2ban/filter.d/nginx-http-auth.conf
```

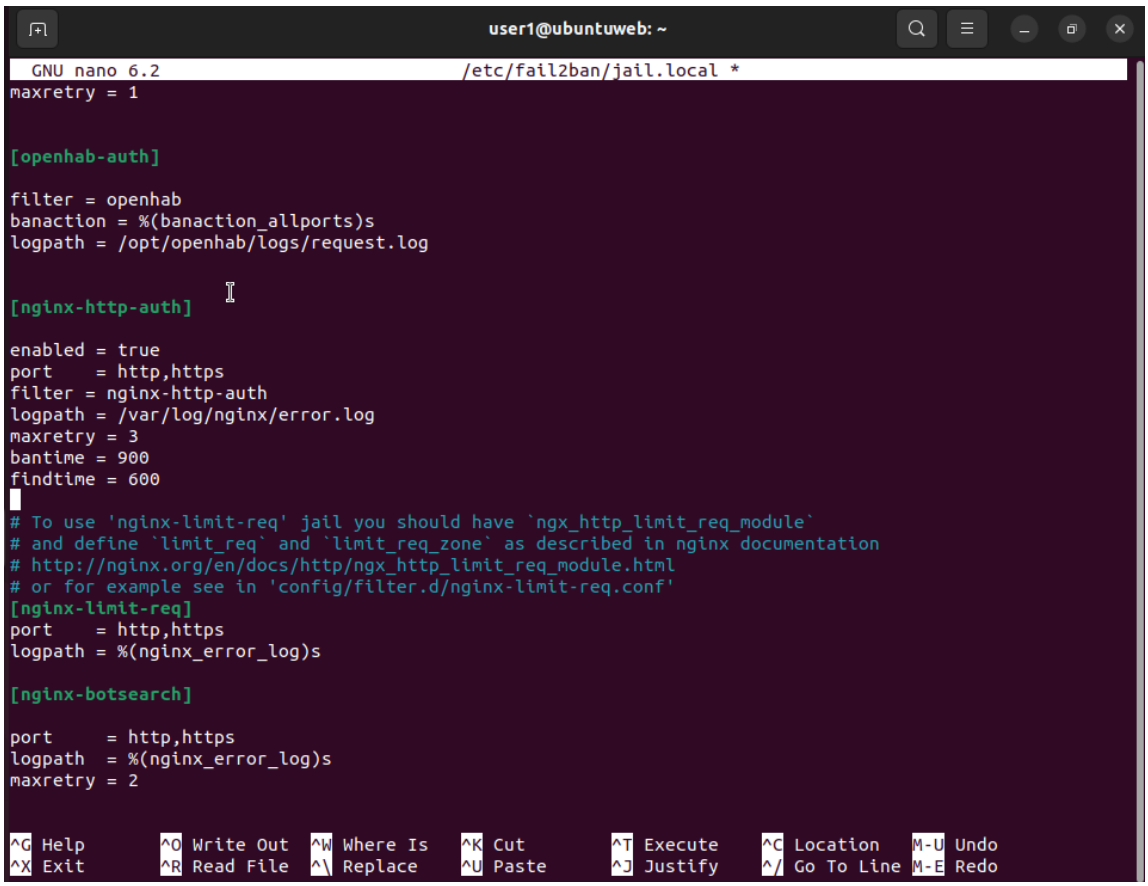
Write:

```
failregex = ^<HOST> -.*"(GET|POST).*HTTP.*" 401

ignoreregex =
```



- In **jail.local**, add:

```
[nginx-http-auth]
enabled = true
port = http,https
filter = nginx-http-auth
logpath = /var/log/nginx/error.log
maxretry = 3
bantime = 900
findtime = 600
```



- Restart Fail2Ban.

```
sudo systemctl restart fail2ban
```

Step 4

Protect WordPress via wp-login.php & XMLRPC

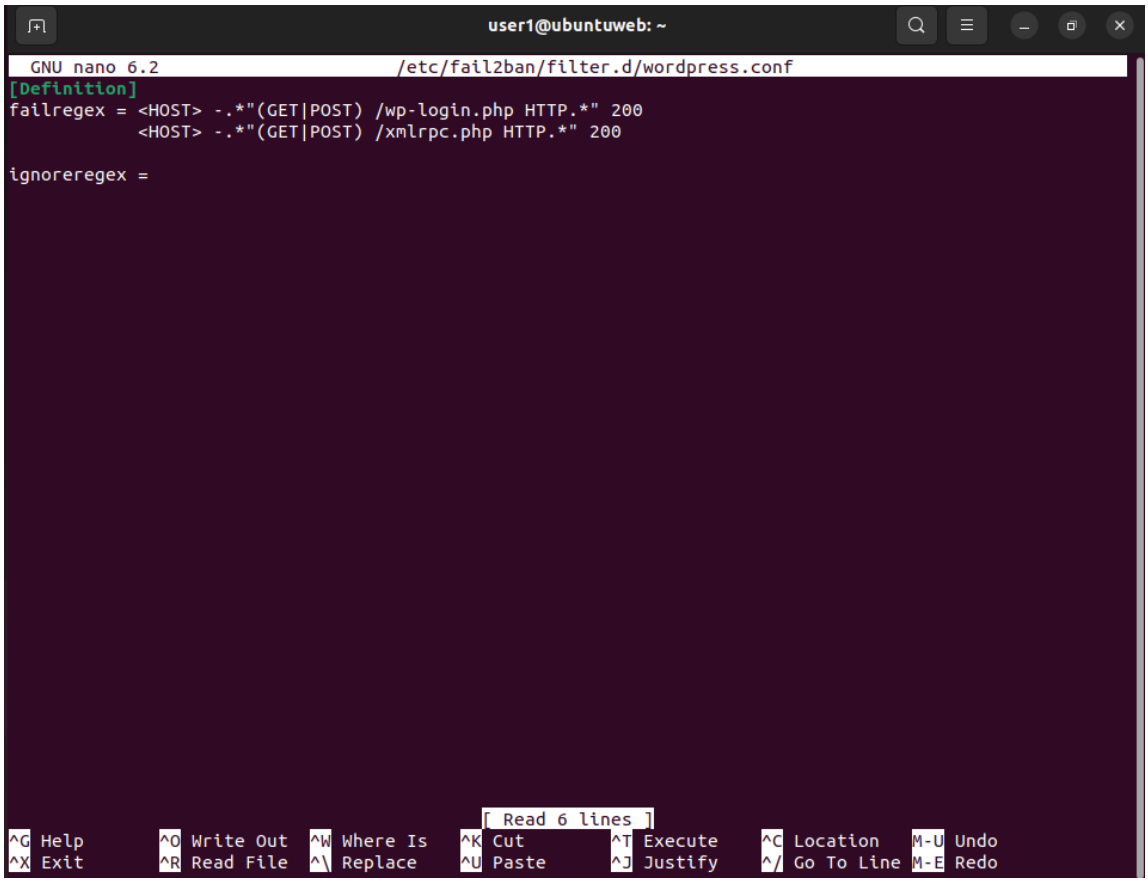
Use custom Fail2Ban filters for WordPress login abuse detection.

- Install and configure the fail2ban-wordpress filter (or create custom regex).

```
sudo nano /etc/fail2ban/filter.d/wordpress.conf
```

Write:

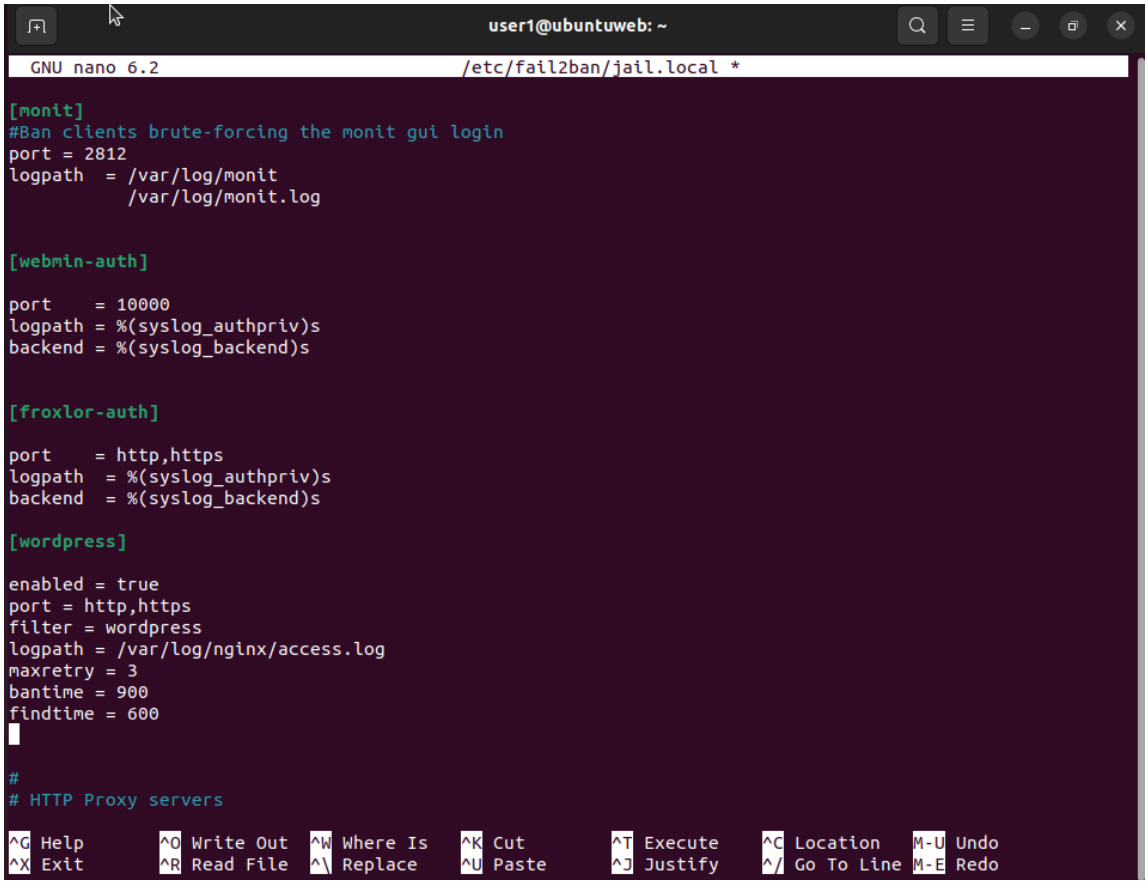
```
failregex = <HOST> -.*"(GET|POST) /wp-login.php HTTP.*" 200
            <HOST> -.*"(GET|POST) /xmlrpc.php HTTP.*" 200

ignoreregex =
```



- Monitor logs: **/var/log/nginx/access.log** or **/var/log/auth.log**

- In **jail.local**, add:

```
[wordpress]
enabled = true
port = http,https
filter = wordpress
logpath = /var/log/nginx/access.log
maxretry = 3
bantime = 900
findtime = 600
```



- Save and restart Fail2Ban.

```
sudo systemctl restart fail2ban
```

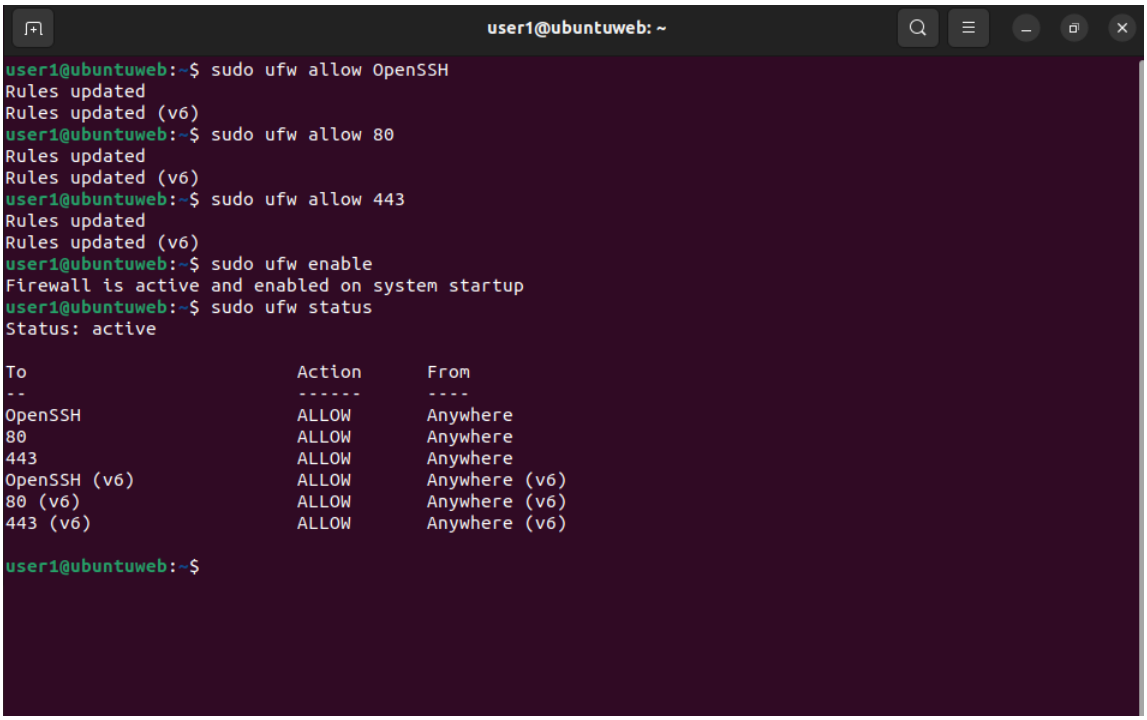
Step 5
Enable UFW and Allow Only Essential Ports

1. Allow **SSH** (default port 22):
   ```
   sudo ufw allow OpenSSH
   ```

2. Allow **HTTP** and **HTTPS**:
   ```
   sudo ufw allow 80
   sudo ufw allow 443
   ```

3. Enable UFW:
   ```
   sudo ufw enable
   ```

4. Check status:
   ```
   sudo ufw status
   ```



Step 6

Block Known Malicious User Agents in Nginx

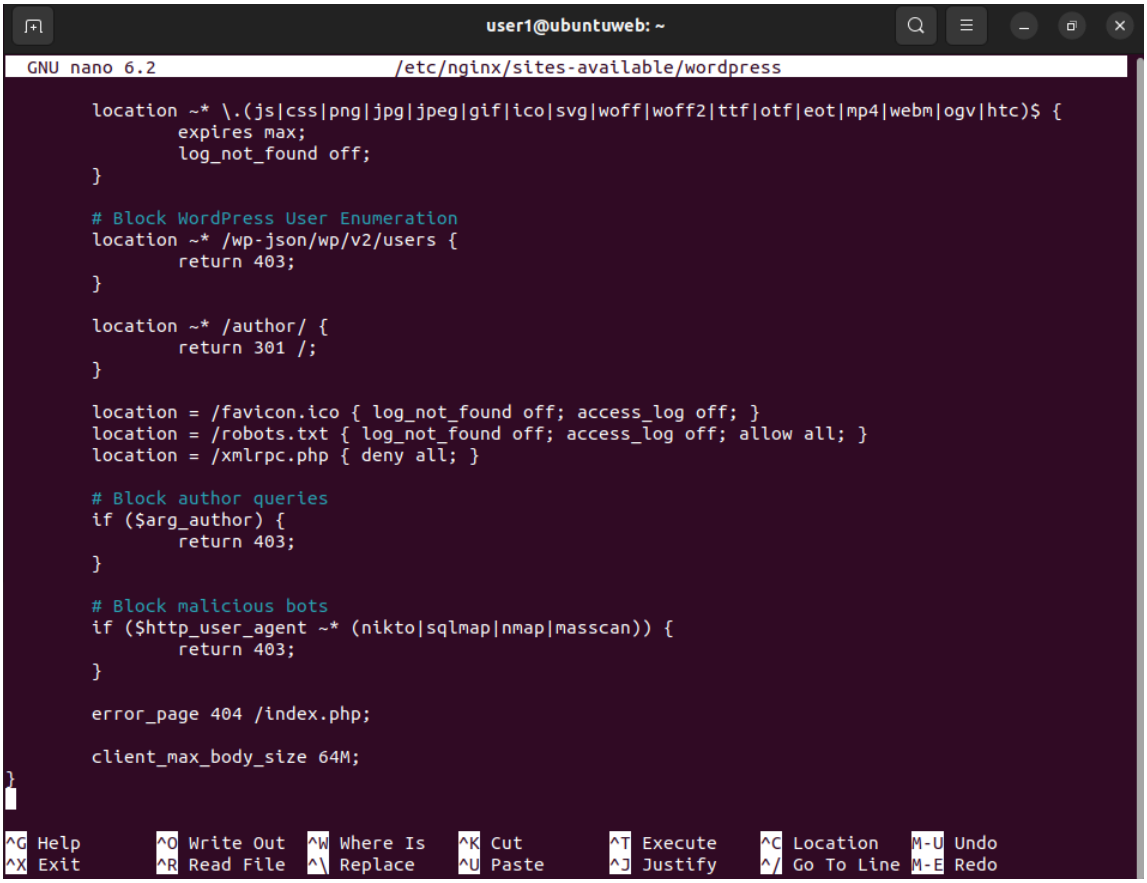
1. Open your Nginx site config:
   ```
   sudo nano /etc/nginx/sites-available/your-site.conf
   ```

2. Inside the server {} block, add:
   ```
   if ($http_user_agent ~* (nikto|sqlmap|nmap|masscan)) {
       return 403;
   }
   ```

3. Save and exit CTRL+O, ENTER, CTRL+X
4. Test config:
   ```
   sudo nginx -t
   ```

5. Reload Nginx:
   ```
   sudo systemctl reload nginx
   ```

This blocks common vulnerability scanners from reaching your site.

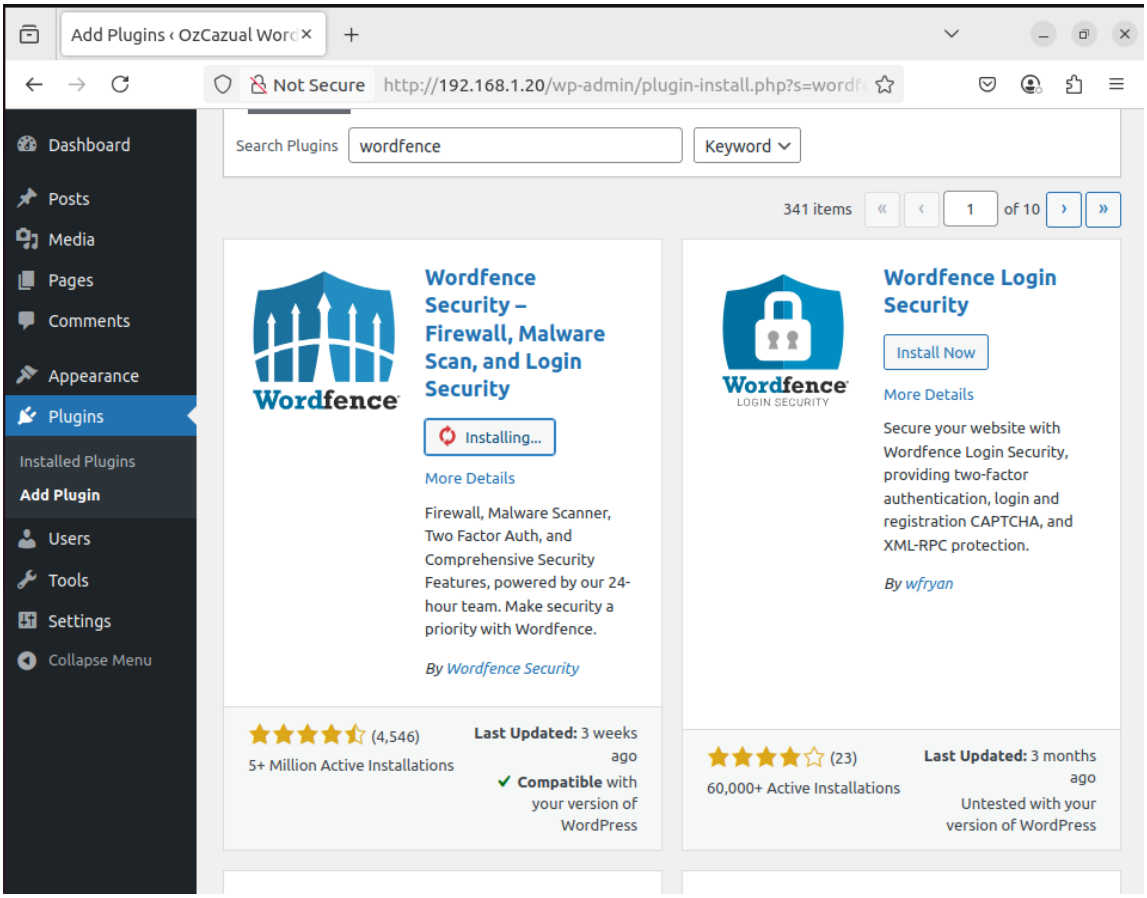


Step 7

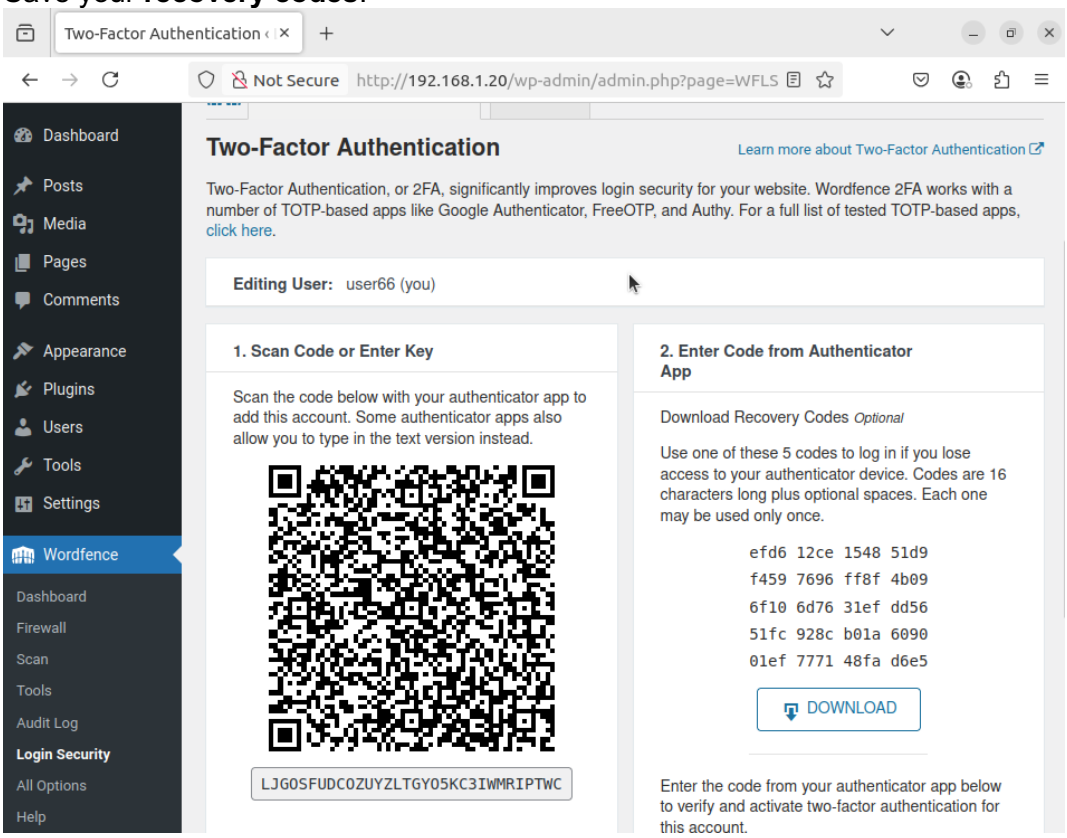
Setup WordPress Plugins (like Wordfence)

Install and configure Wordfence to enable MFA, firewall and for monitoring logs.

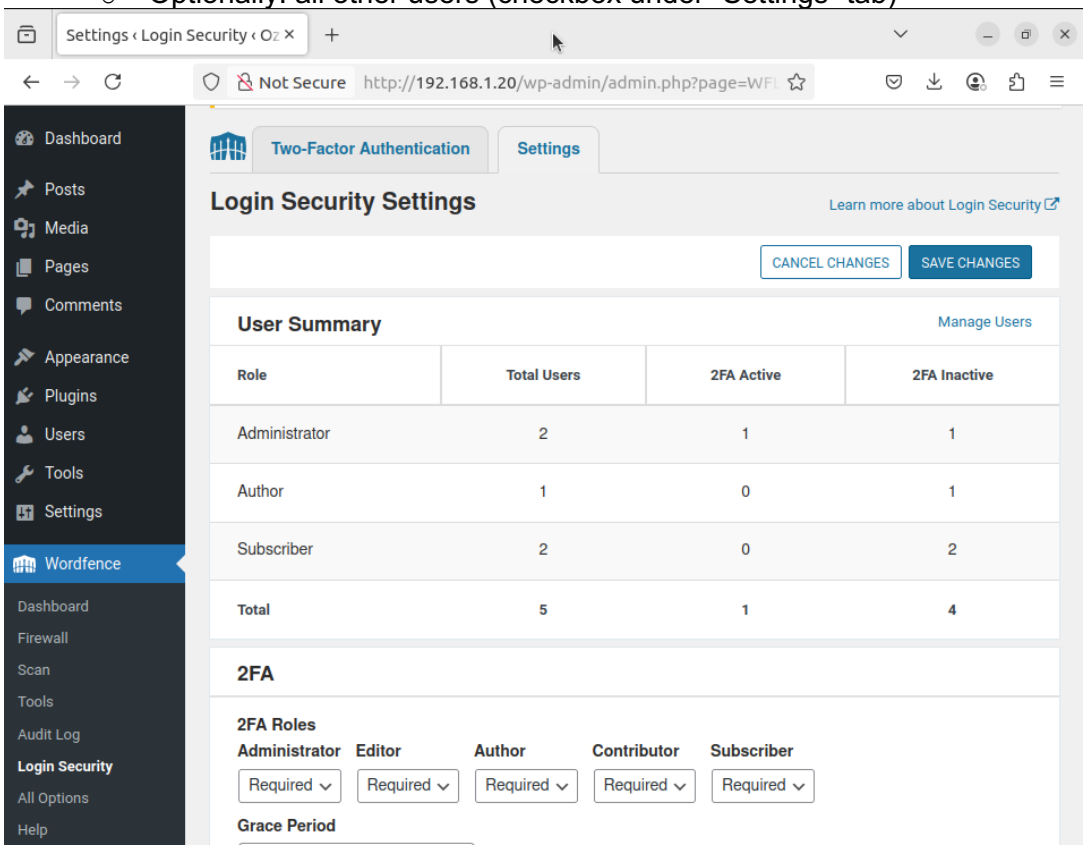
- **Install and Configure Wordfence Login Security**
  1. **Log in to WordPress Admin** (https://your-ip/wp-admin)
  2. Go to:
     Plugins > Add New
  3. Search for:
     Wordfence Security – Firewall & Malware Scan
  4. Click **Install** > **Activate**



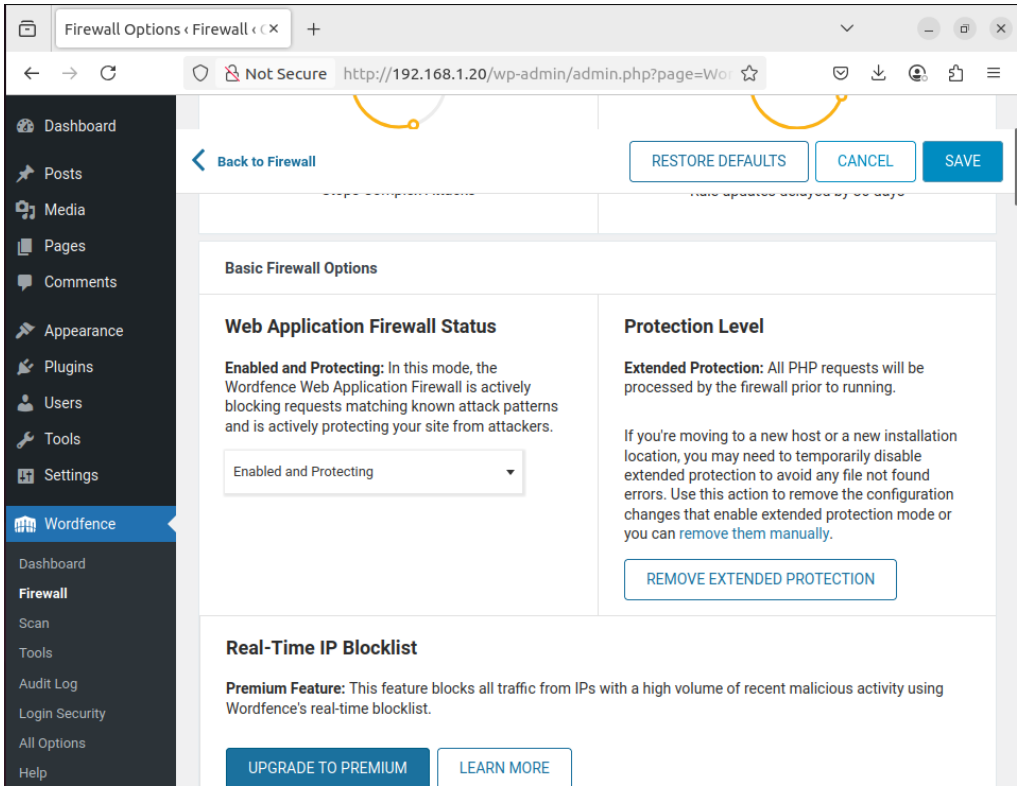
- **Configure 2FA (Wordfence)**
  1. Go to: **Wordfence > Login Security**
  2. Scan the QR code using **Google Authenticator**, **Authy**, or similar.
  3. Save your **recovery codes**.



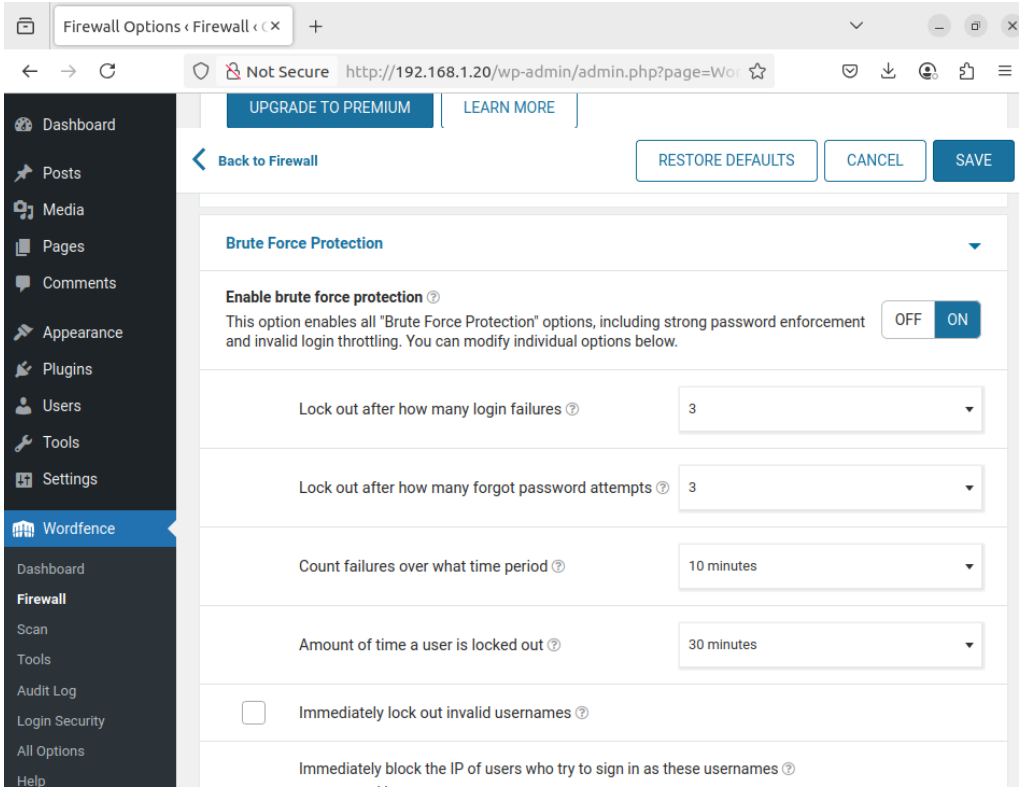
  4. Enable for:
     - **Your admin account**
     - Optionally: all other users (checkbox under “Settings” tab)



- **Configure Firewall and Lockout Settings**
  1. Go to: **Wordfence > Firewall**
  2. Click **Manage Firewall**
     - Set it to **"Enabled and Protecting"**



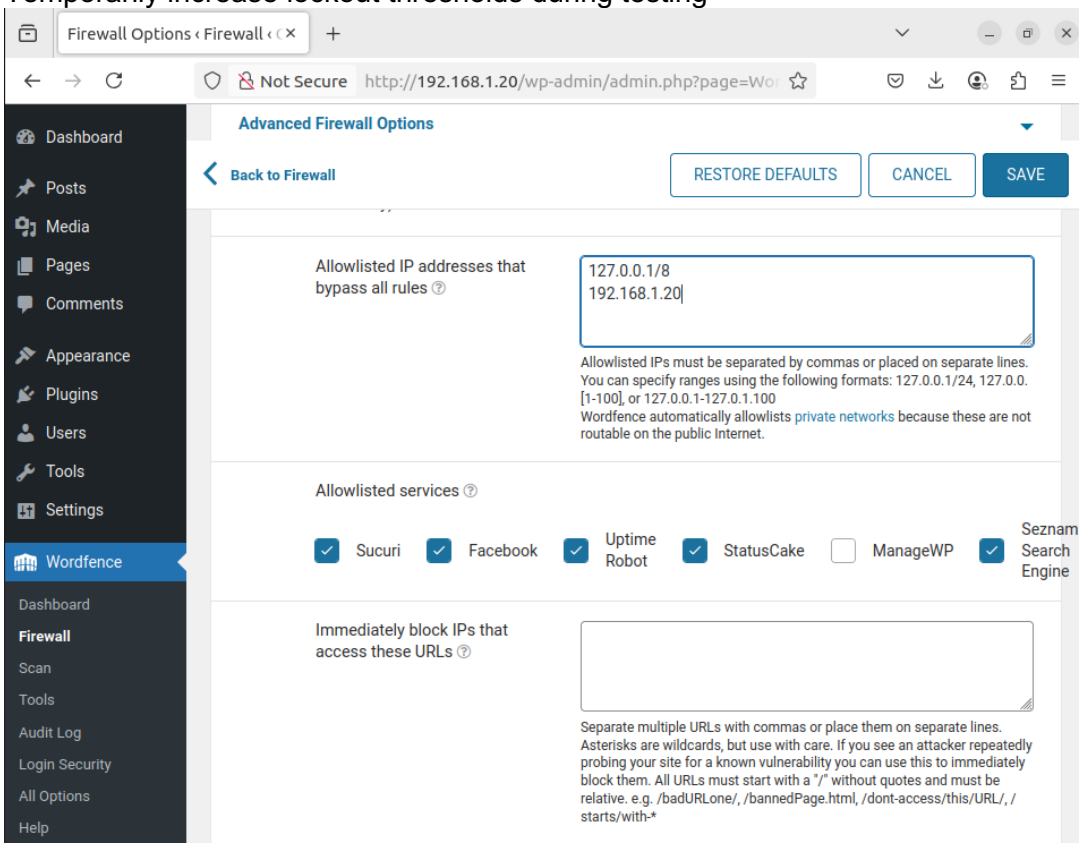
  3. Go to **Rate Limiting** tab:
     - Brute-force protection settings:
       - Lock out after 3–5 login failures
       - Lock out for 5–15 minutes
       - Choose to email on lockout



4. Go to: **Wordfence > Tools > Live Traffic**
    - Monitor brute-force attempts (IP, username, action)

- **Avoid Lockout During Hydra or WPScan**
    - Go to: **Wordfence > All Options > Brute Force Protection**
    - Add **your IP address** to **"Allowlisted IP addresses that bypass all rules"**

Temporarily increase lockout thresholds during testing



Step 8

Enable Logging and Monitoring

Ensure logs are retained for analysis and alerts are configured.

- View Fail2Ban status:
  `sudo fail2ban-client status and sudo fail2ban-client status [jailname]`

- Enable **email alerts** (optional) in **/etc/fail2ban/jail.local**
- Integrate with **log aggregation tools** if available (e.g., Graylog, ELK)

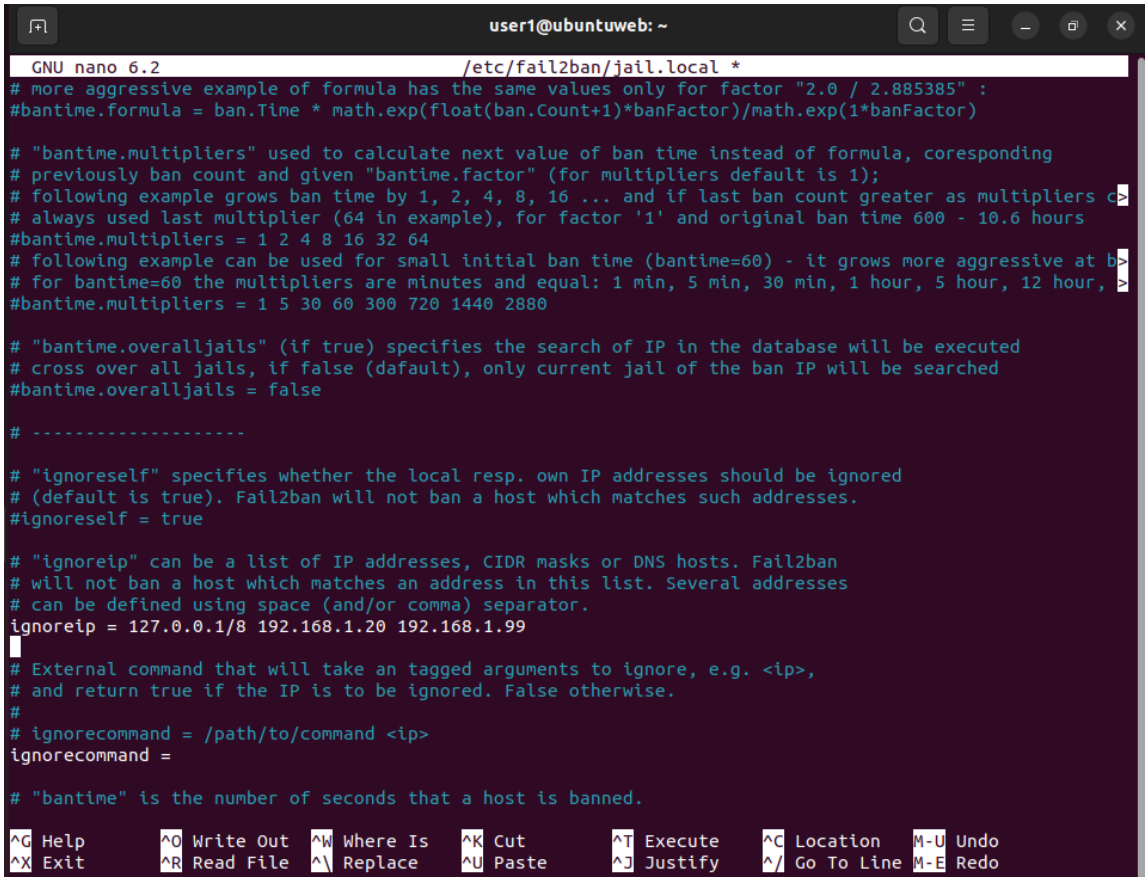
Step 9

Whitelist Internal IPs

Prevent banning of trusted internal sources.

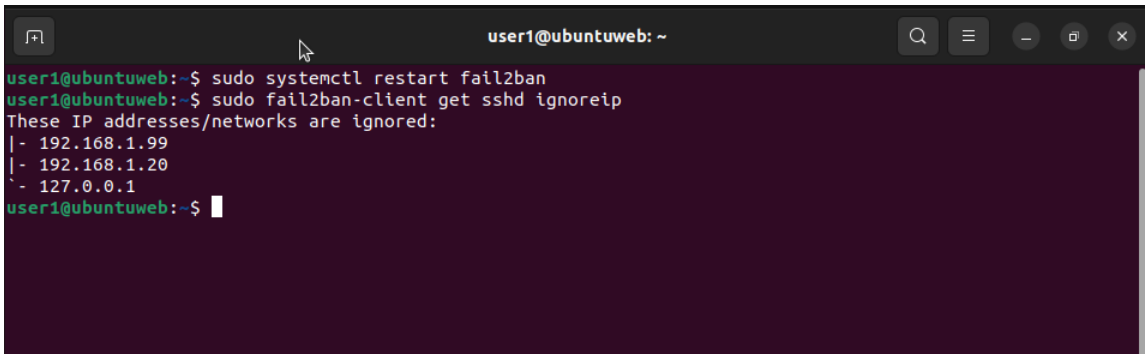
- Edit **ignoreip** in **/etc/fail2ban/jail.local**:

```
ignoreip = 127.0.0.1/8 192.168.1.20/24
```



- Save and restart Fail2Ban
- Verify the setting is active:

```
sudo fail2ban-client get sshd ignoreip
```



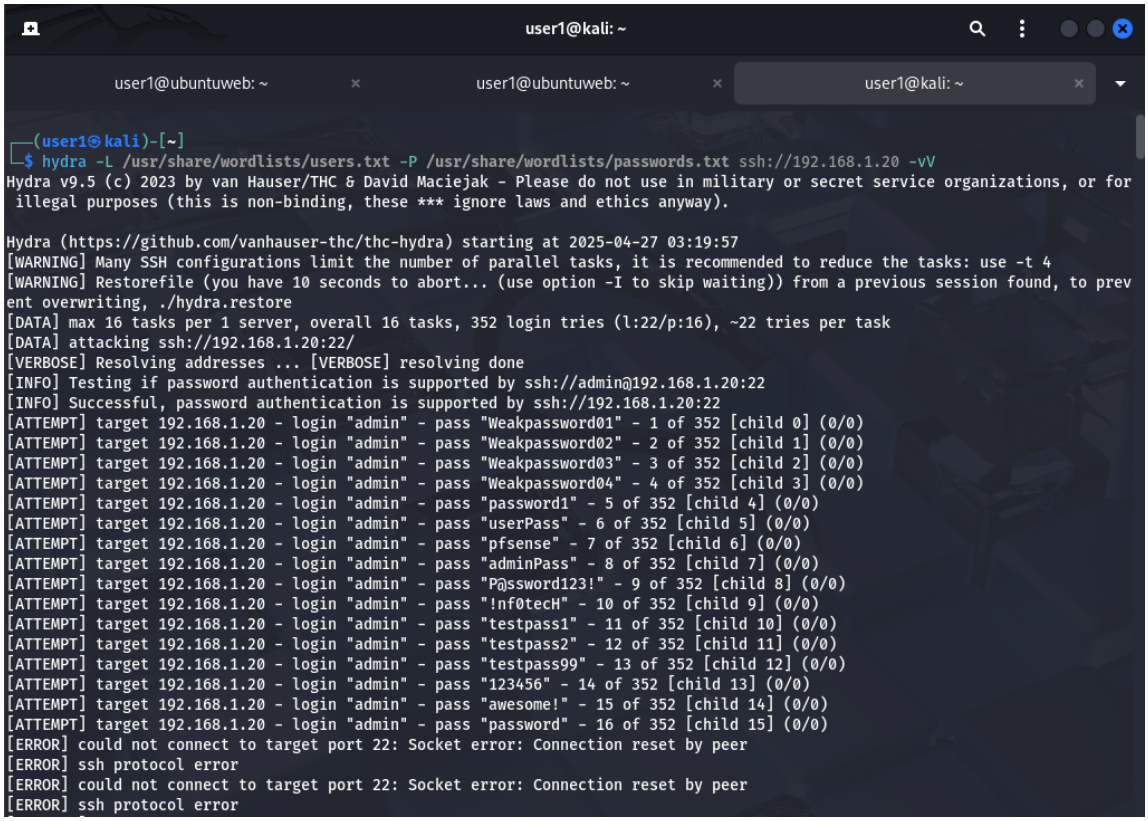
Step 10

Test Brute-Force Scenarios and Validate Protection

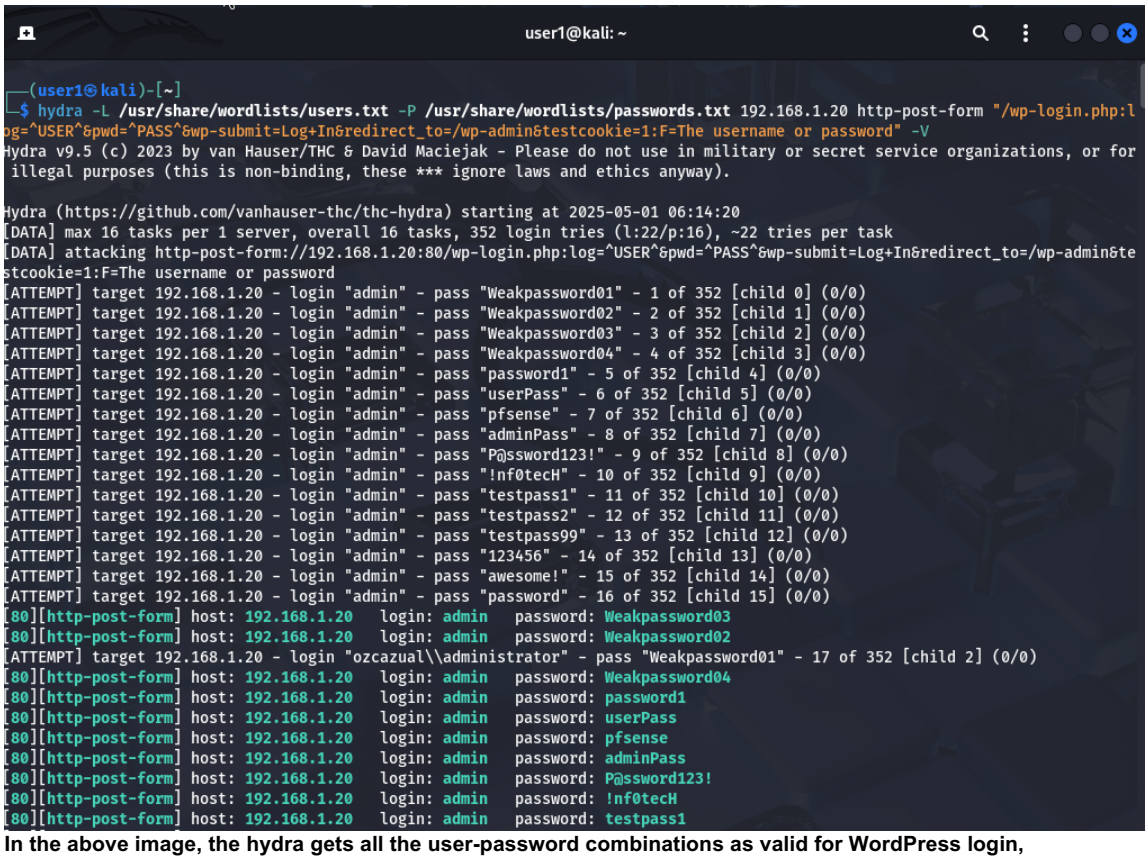
Simulate attacks using tools to confirm Fail2Ban is working.

- Use **Hydra** to simulate SSH and WordPress brute-force attacks

```
hydra -L users.txt -P passwords.txt ssh://<target_IP> -vV
```



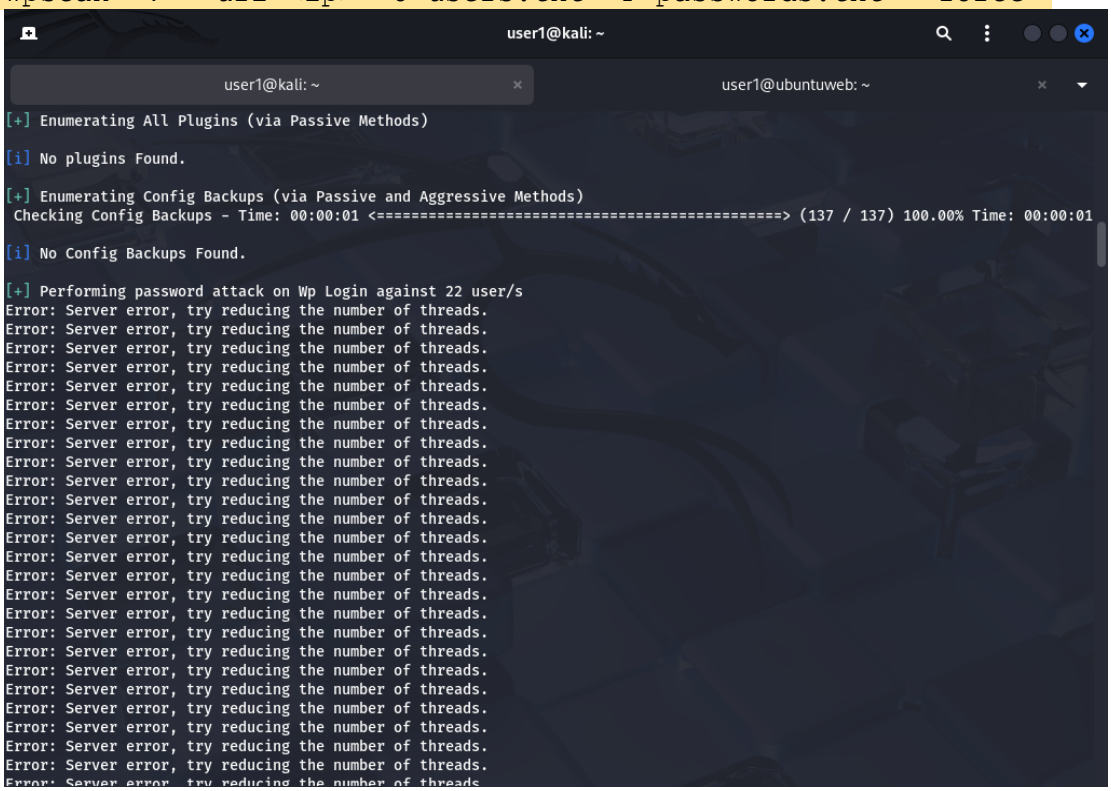
```
hydra -L users.txt -P passwords.txt <ubuntu_ip> http-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&redirect_to=/wp-admin&testcookie=1:S=wp-admin" -V
```



In the above image, the hydra gets all the user-password combinations as valid for WordPress login, as it cannot detect the valid user credentials hence confusing the attacker.

- Use **WPScan** to simulate WordPress brute-force attacks

```
wpscan -v --url <ip> -U users.txt -P passwords.txt --force
```

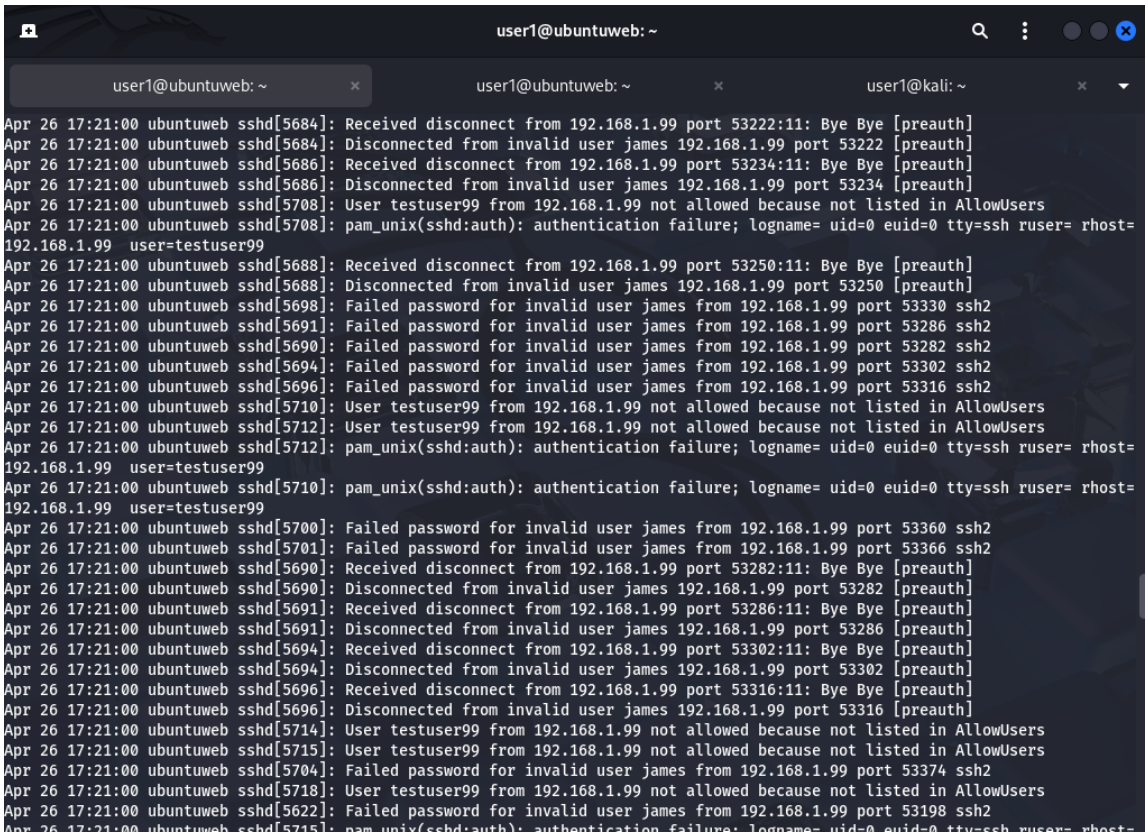


- Monitor ban behavior:

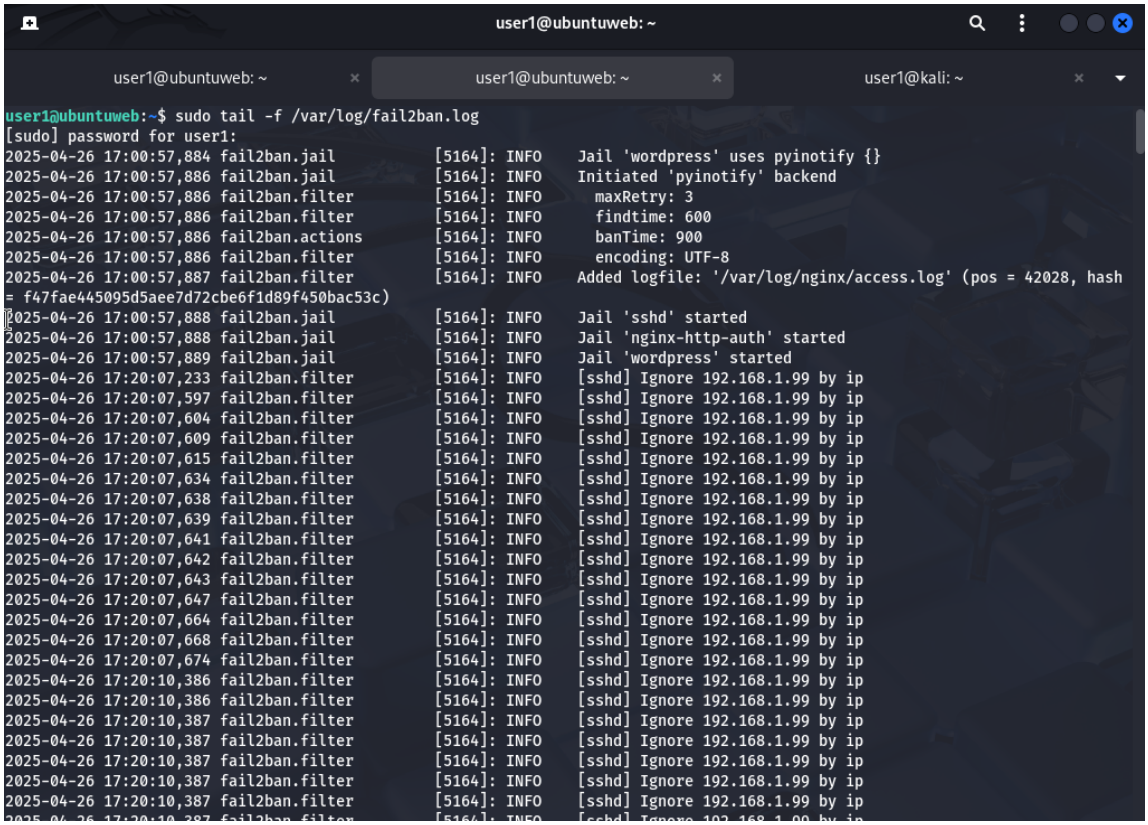
```
fail2ban-client status
fail2ban-client status sshd
fail2ban-client status wordpress
```

- Review **Nginx** and **auth logs** for matching entries

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

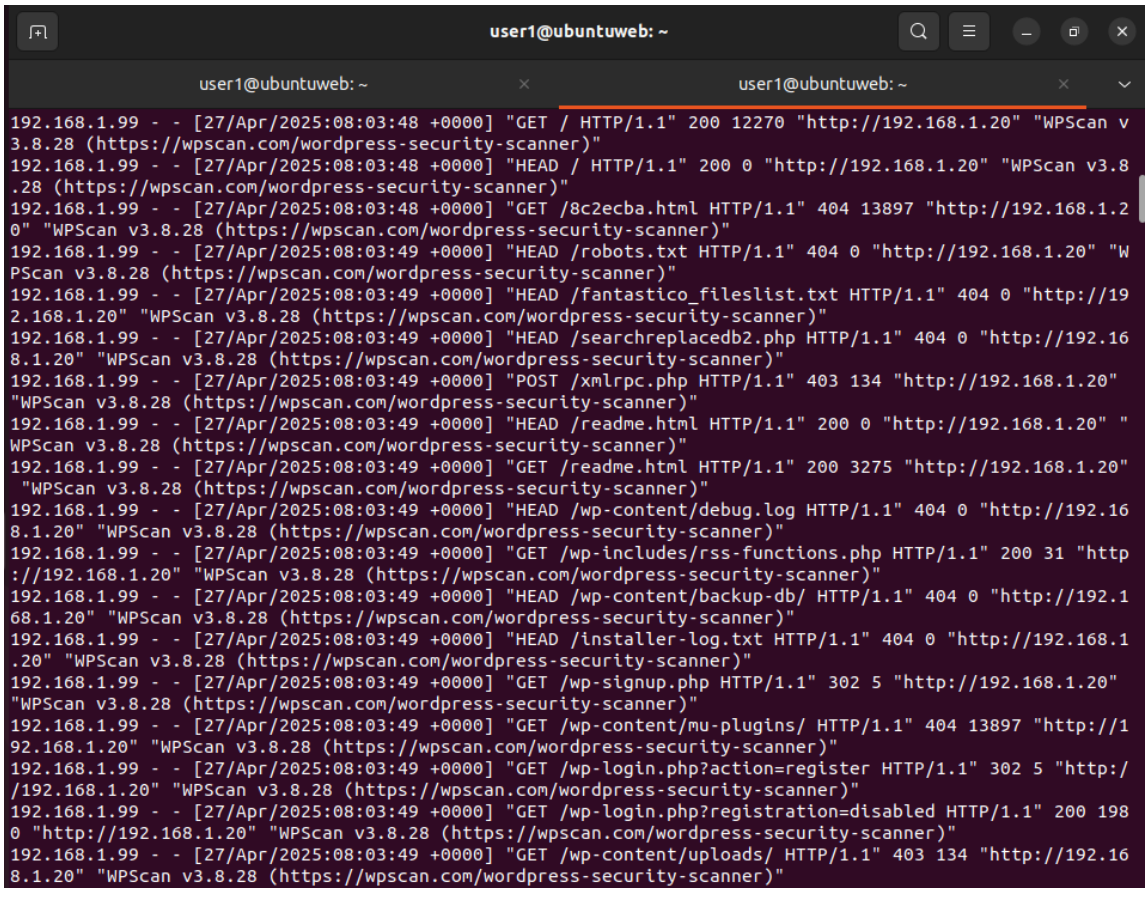


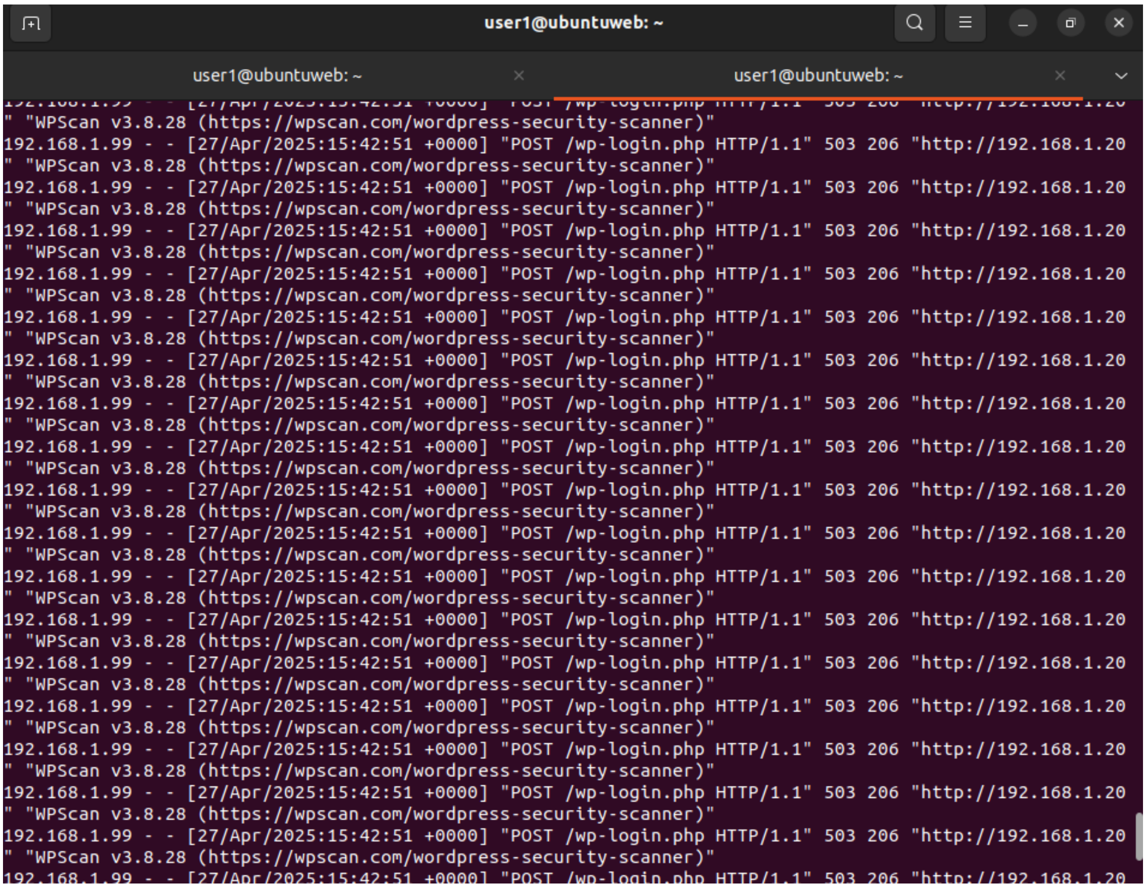
```
sudo tail -f /var/log/fail2ban.log
```



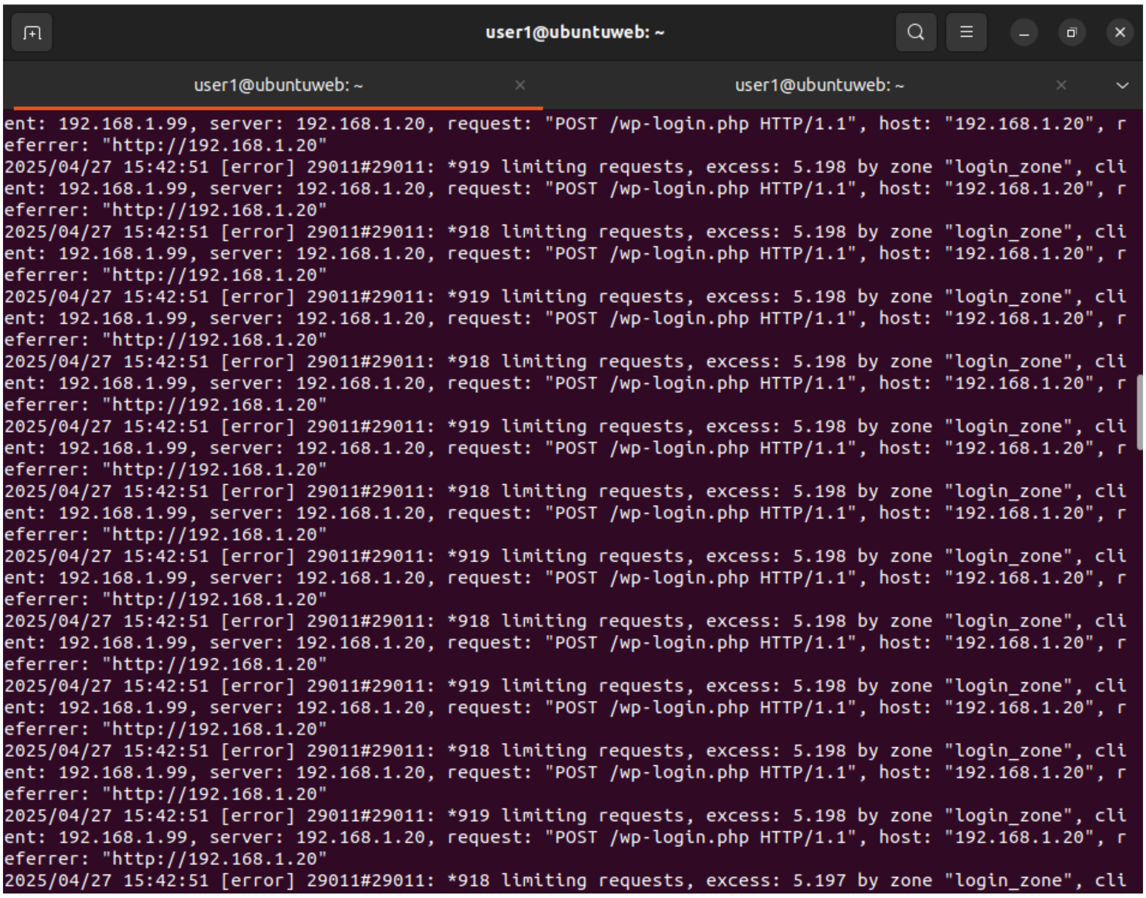
For Brute-Force with **WPScan**:

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

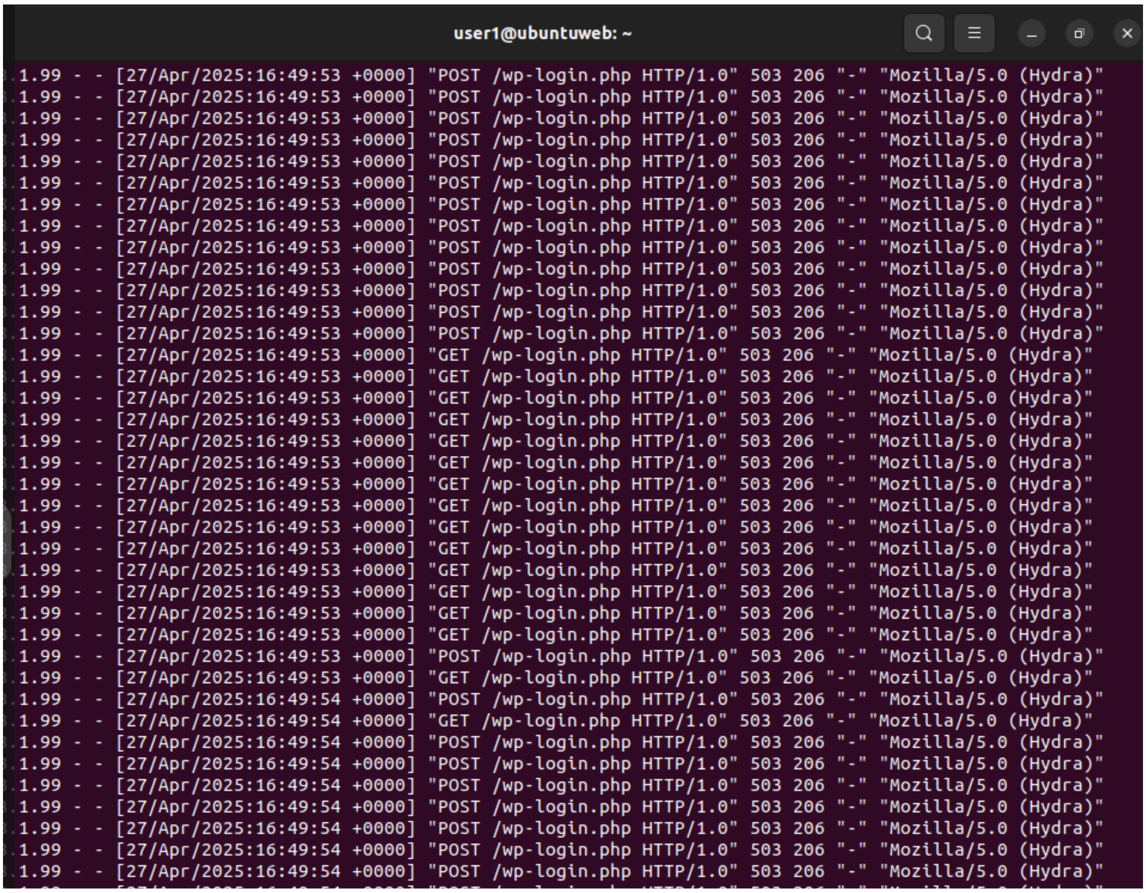





```
sudo tail -f /var/log/nginx/error.log
```



For brute-force **wordpress** with **hydra**:
`sudo tail -f /var/log/nginx/access.log`



```
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:53 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "GET /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
1.99 - - [27/Apr/2025:16:49:54 +0000] "POST /wp-login.php HTTP/1.0" 503 206 "-" "Mozilla/5.0 (Hydra)"
```

Step 11
Maintain Configuration and Update Regularly
Keep Fail2Ban up to date and validate configurations monthly.

- Regularly update filters, jail configurations
- Test protections every month
- Document logs and include in audit reports
- Apply updates with `sudo apt update && sudo apt upgrade fail2ban`