Document Name	Testing Report – Brute-Force Protection on Ubuntu (Playbook 2)	Version	1.0
Author	Anusha Ramu Chakravarthi	Date Created	24/04/2025
Test Type	Brute-Force Protection Testing (SSH, HTTP, WordPress)	Last Modified	12/05/2025

Purpose

This report summarizes the results of testing the brute-force protection mechanisms on the Ubuntu 22.04 server. It evaluates the effectiveness of **Fail2Ban** in protecting SSH, HTTP (Nginx), and WordPress against brute-force attacks from tools such as **Hydra** and **WPScan**.

Tools Used

- **Hydra**: Used for SSH brute-force testing.
- **WPScan**: Used for WordPress brute-force testing.
- **Fail2Ban**: Used for automating the blocking of malicious IPs based on failed login attempts.
- **Nginx**: Web server used for HTTP (login) protection.
- **WordPress**: Target for WordPress login brute-force protection.

Test Procedures

1. SSH Brute-Force Testing

- **Tool**: Hydra
- **Target**: SSH (port 22) on Ubuntu 22.04
- **Protection**: Fail2Ban, SSH Hardening (Key-based authentication, root login disabled)
- **Test Setup**: Attack initiated from Kali VM using Hydra to simulate multiple SSH login attempts.

2. WordPress Brute-Force Testing

- **Tool**: WPScan
- **Target**: WordPress login page on Ubuntu Web Server
- **Protection**: WordPress login protection plugins (Wordfence), Nginx rate limiting, Fail2Ban
- **Test Setup**: WPScan used to enumerate valid WordPress usernames and attempt login with common passwords.

3. HTTP Brute-Force Testing (Nginx Protection)

- **Tool**: Hydra
- **Target**: HTTP login page for WordPress
- **Protection**: Nginx rate limiting and Fail2Ban
- **Test Setup**: Hydra used to attempt multiple HTTP login attempts for WordPress.

Observations

1. SSH Brute-Force Testing with Hydra

- **Testing Outcome**:
  - Hydra successfully attempted SSH brute-force.
  - Fail2Ban detected repeated failed login attempts and blocked the Kali VM's IP after exceeding the configured threshold (e.g., 3 attempts).
  - Once blocked, further Hydra attempts failed, with the message "Connection refused."
  - Event logs showed multiple entries for **sshd** authentication failures (e.g., sshd[5678]: Failed password for invalid user root from [IP]).
  - No successful login occurred due to key-based authentication enforcement.

- **Conclusion**:
  - **Fail2Ban** successfully blocked repeated brute-force attempts.
  - **SSH hardening** (e.g., key-based authentication, root login disabled) proved effective in preventing unauthorized access.
  - **SSH brute-force protection** is effective as long as hardening and Fail2Ban configurations are intact.

2. WordPress Brute-Force Testing with WPScan

- **Testing Outcome:**
  - WPScan attempted to enumerate usernames and login with common passwords.
  - The WordPress login page was protected by multiple layers:
    - **Fail2Ban** (triggered after 3 failed login attempts via Nginx logs)
    - **Nginx rate-limiting** (configured to throttle excessive requests)
    - **Wordfence plugin** (enabled firewall, login lockout, and 2FA for admin)
  - After several failed attempts:
    - **Fail2Ban** blocked the Kali VM's IP, based on Nginx logs and regex filters.
    - **Wordfence** also logged the brute-force attempts, enforced temporary login lockouts, and sent email alerts.
    - The attacker (Kali) was unable to proceed after the IP was blocked and the account was temporarily locked.
  - **Nginx logs** showed multiple HTTP 403 errors.
  - **Wordfence logs** confirmed detection and mitigation of login brute-force.
  - **No successful login occurred.**

- **Conclusion:**
  - **Fail2Ban + Nginx rate-limiting** provided effective external brute-force mitigation.
  - **Wordfence plugin** added application-layer protection via:
    - Login attempt limits
    - IP blocking
    - Two-Factor Authentication (2FA)
    - Real-time alerts and visibility into attacks
  - Combined use of Fail2Ban, Nginx, and Wordfence created a **multi-layered defense** that successfully mitigated WPScan brute-force attempts.
  - The system remained secure with no compromise observed.

3. **HTTP Brute-Force Testing (Nginx)**
   - **Testing Outcome**:
     - Hydra successfully attempted brute-force against the WordPress login page over HTTP.
     - Nginx **rate-limiting** kicked in and blocked the Kali VM's IP after repeated login attempts.
     - Fail2Ban's additional blocking of IPs further reduced Hydra's ability to continue its attack.
     - The Nginx logs showed 401 Unauthorized errors corresponding to failed login attempts.
   - **Conclusion**:
     - **Nginx rate-limiting** and **Fail2Ban** worked together to effectively protect the WordPress login page against brute-force attacks.
     - **HTTP brute-force protection** was successful, showing the effectiveness of multiple layers of defense.

Overall Conclusions

- **SSH Protection**:
  The combination of **SSH hardening** (key-based authentication and disabling root login) and **Fail2Ban** effectively blocked brute-force attempts against SSH.
- **WordPress Protection**:
  - **Fail2Ban** and **Nginx rate limiting** successfully mitigated brute-force attacks on the WordPress login page.
  - **Wordfence** provided additional protection through login attempt throttling, IP blocking, 2FA, and alerting. No successful login attempts were recorded.
- **HTTP Brute-Force Protection**:
  Nginx's rate-limiting combined with **Fail2Ban** successfully mitigated HTTP brute-force attacks.

Recommendations

1. **SSH**:
   - Continue enforcing **key-based authentication** and **disabling root login**.
   - Review and adjust **Fail2Ban** configurations to ensure optimal blocking of malicious IPs.
   - **Consider implementing Multi-Factor Authentication (MFA)** for SSH for additional security.

2. **WordPress**:
   - **Strengthen WordPress login protection** with plugins like **Wordfence** and **Limit Login Attempts** to further limit brute-force attack success.
   - Review and adjust **Nginx rate-limiting** configurations for more aggressive limits on login attempts.
   - **Review and maintain Fail2Ban** rules for WordPress login to ensure continued effectiveness.

3. **General Recommendations**:
   - Consider implementing **intrusion detection systems (IDS/IPS)** like **Suricata** or **Snort** to catch more advanced brute-force or other attack patterns.
   - **Regularly update WordPress plugins** to patch known vulnerabilities.
   - **Review logging configurations** to ensure that logs are capturing critical events, including failed login attempts, for both SSH and WordPress.