

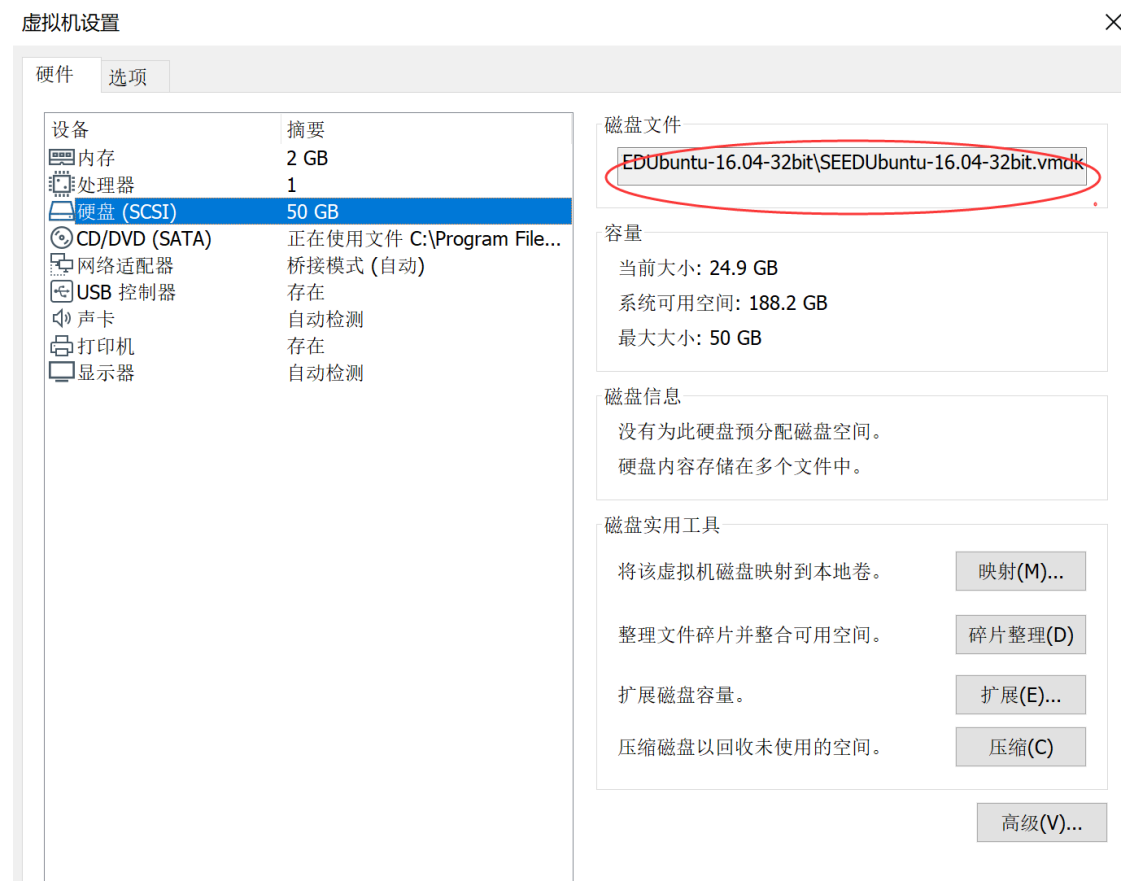
网络安全实验常见错误及解答

目录

-----TCP 实验-----	
1. 虚拟机找不到操作系统.....	2
2. bind 错误: 提示 address in use.....	2
3. Sniff 监听失败, 提示找不到设备.....	2
4. 发送报文, 对方未接收到.....	3
5. C 语言实现监听代码运行以后出现 “segmentation fault”	3
6. 运行 py 脚本, 提示 “operation not permitted”	3
7. 明明已经安装了 scapy, 但是运行 py 脚本, 提示找不到 scapy 模块.....	3
8. 在容器里面执行 tcpdump 出错.....	4
9. telnet 到 server 以后, 出现错误.....	4
10. 用户机和服务器建立的 TCP 连接, 为什么在攻击机器上截获不到该连接的报文?	5
11. TCP SYN-Flood 攻击没效果, 客户端仍然可以正常建立连接.....	5
-----DNS 实验-----	
12. service bind9 start 启动失败.....	5
13. Service bind9 start 启动后没反应, 不知道是否成功	5
14 在做 DNS 攻击实验时, 为什么客户机上无法解析域名?	5
15. 在做本地 DNS 攻击时, 用 netwox 伪造了响应报文, wireshark 也监听到了伪造报文, 为什么客户机解析出来的还是原来的 IP?	6
16. 远程 DNS 攻击时, 在 system 调用 dig 命令之后, 为什么伪造的第一个报文都比真实响应还慢?	6
-----VPN 实验-----	
17. VPN 实验中, 第一个 vpn 例程运行以后, 客户端连接服务器, 服务器为什么没有反应?	7
18. VPN 实验中, 第一个 vpn 例程运行以后, 已经按指导书添加了虚拟网络的路由, 但是从客户端主机 ping 内网主机, 网络不通	7
19. vpn 隧道建立以后, wireshark 抓包只能看到 4433 的 tcp 报文, 看不到 tls 报文.....	7
20. tlsclient 和 tlsserver 建立 ssl 不成功, 客户端验证服务器证书失败, 服务器端提示 unknown ca 错误.....	8
21. tlsclient 和 tlsserver 建立 ssl 不成功, 客户端验证服务器证书失败, 服务器端提示:	8
22. tlsclient 和 tlsserver 建立 ssl 不成功, 客户端验证服务器证书失败, 服务器端提示 error:14094415:SSL routines: ssl3_read_bytes: sslv3 alert certificate expired:s3_pkt.c:1487:SSL alert number 45 的错误	8
24. tlsclient 和 tlsserver 建立不成功, 客户端验证服务器证书失败:	10
25 .crt 文件和.pem 文件有什么差别	10
26. VPN 实验中, 最后支持多客户端那个部分, 内网主机返回的报文到达 VPN 服务器以后, tun0 中的数据应该向哪个隧道转发? 怎么区分?	10

1. 虚拟机找不到操作系统

解答： 请将虚拟机设置里面的硬盘文件路径改成 seedubuntu-16.04-32bit 目录下的磁盘文件



2. bind 错误： 提示 address in use

解答： 说明绑定的端口被使用了，需要停掉原来的进程，或者用一个新的端口号
可以通过 `sudo netstat -naup` 查看 udp 端口使用的进程
`sudo netstat -natp` 查看 tcp 端口使用的进程

3. Sniff 监听失败，提示找不到设备

解答： sniff 的 iface 参数指定错误，因为例子代码跟你的主机接口名字不一样。可以 ifconfig 查看本机的接口，
跟容器通信的接口为 docker0，跟外网通信的接口为 enxxxxx

4. 发送报文，对方未接收到

解答：可能是发送报文的地址填写错误，“127.0.0.1”是指的本机地址，即发送报文的地址是到自己

5. C 语言实现监听代码运行以后出现“segmentation fault”

解答：可能是过滤器的语法错误，可以用“tcpdump -i 网卡名 过滤规则“ 验证过滤器的语法

6. 运行 py 脚本，提示“operation not permitted”

```
[03/25/22]seed@VM:~$ python sniff_spoof.py
Traceback (most recent call last):
  File "sniff_spoof.py", line 26, in <module>
    sniff(filter='icmp and src host 192.168.60.3',prn=spoof_pkt)
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/sendrecv.py", line 7
31, in sniff
    *arg, **karg)] = iface
  File "/home/seed/.local/lib/python2.7/site-packages/scapy/arch/linux.py", line
567, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(typ
e))
  File "/usr/lib/python2.7/socket.py", line 191, in __init__
    _sock = _realsocket(family, type, proto)
socket.error: [Errno 1] Operation not permitted
[03/25/22]seed@VM:~$
```

解答： 权限不够，需要 sudo 执行

```
[03/25/22]seed@VM:~$ sudo python sniff_spoof.py
```

7. 明明已经安装了 scapy，但是运行 py 脚本，提示找不到 scapy 模块

(1)情况 1：超级用户执行

```
root@VM:/home/seed# python3 sniff_spoof.py
Traceback (most recent call last):
  File "sniff_spoof.py", line 2, in <module>
    from scapy.all import *
ImportError: No module named 'scapy'
root@VM:/home/seed#
```

(2) 情况 2：python3 执行

```

^[[^A^C[03/25/22]seed@VM:~$ sudo python3 sniff_spoof.py
Traceback (most recent call last):
  File "sniff_spoof.py", line 2, in <module>
    from scapy.all import *
ImportError: No module named 'scapy'
[03/25/22]seed@VM:~$

```

解答：请用普通用户执行（不要用 root 用户，提示为\$，而不是#），sudo python 执行脚本，而不是 python3

```

[03/25/22]seed@VM:~$ sudo python sniff_spoof.py

```

8. 在容器里面执行 tcpdump 出错

```

root@3f796f9d8299:/# tcpdump
ERROR: ld.so: object '/home/seed/lib/boost/libboost_program_options.so.1.64.0' from LD PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_filesystem.so.1.64.0' from LD PRELOAD cannot be preloaded (cannot open shared object file): ignored.
ERROR: ld.so: object '/home/seed/lib/boost/libboost_system.so.1.64.0' from LD PRELOAD cannot be preloaded (cannot open shared object file): ignored.
tcpdump: error while loading shared libraries: libcrypto.so.1.0.0: cannot open shared object file: Permission denied
root@3f796f9d8299:/#

```

解决办法：（在容器里执行以下命令）：

```

mv /usr/sbin/tcpdump /usr/bin/tcpdump
ln -s /usr/bin/tcpdump /usr/sbin/tcpdump

```

9. telnet 到 server 以后，出现错误

```

Connection closed by foreign host.
root@user:/# telnet 172.17.0.2
Trying 172.17.0.2...
Connected to 172.17.0.2.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
2091fb242a3d login: seed
Password:
Last login: Thu Apr  8 20:31:32 CST 2021 from 172.17.0.3 on pts/1
sh: 1: cannot create /run/motd.dynamic.new: Directory nonexistent
[04/08/22]seed@2091fb242a3d:~$

```

解决：这种情况，在容器里面创建一下/var/run 到/run 的链接。

```

ln -s /var/run /run

```

10. 用户机和服务器建立的 TCP 连接，为什么在攻击机器上截获不到该连接的报文？

解答： 请确认攻击机上 wireshark 监听的网卡是跟其它两台机器配置同一网络 IP 的那个网卡，而不是 any。

11. TCP SYN-Flood 攻击没效果，客户端仍然可以正常建立连接

解答： 1) scapy 脚本由于发包速度不够快，确实造成不了 DoS，但是还是可以看出有很多连接的；

2) netwox 攻击的时候，可以不带-s 参数，仍然伪造了源 IP，是可以攻击成功的；

3) C 程序发包速度也很快，也可以攻击成功，不成功的原因可能是没有 sudo 执行。

12. service bind9 start 启动失败

提示加载 liblwres.so.141 动态库失败，权限不够

```
root@HostM:/home/seed# service bind9 start
* Starting domain name service... bind9
/usr/sbin/named: error while loading shared libraries: liblwres.so.141: cannot open shared object file: Permission denied
[fail]
root@HostM:/home/seed#
```

解决: 创建容器的时候，docker run 后面不带--privileged 参数

13. Service bind9 start 启动后没反应，不知道是否成功

启动失败，无错误原因提示

查看错误信息: named -d 3 -f -g

一般是配置文件语法错误引起的

14 在做 DNS 攻击实验时，为什么客户机上无法解析域名？

解答： 首先确认服务器上的域名服务是否正确启动

1) 可以用 sudo netstat -na 查看监听端口，是否在 udp 53 号端口上监听（在任意地址监听，不是 127.0.0.1）

2) 也可以通过 sudo ps aux|grep named 查看 named 进程是否在运行；

3) 以上过程发现域名服务没有启动的话，请启动域名服务

4) 启动过程中如果出错，一般都是配置文件有问题；目前发现较多的问题是指导书上写的要在 **/etc/bind/named.conf** 中增加的两个 zone 信息，需要添加到

/etc/bind/named.conf.default-zones 文件中

```
zone "example.com" {  
  
    type master;  
  
    file "/etc/bind/example.com.db";  
  
};  
  
zone "0.168.192.in-addr.arpa" {  
  
    type master;  
  
    file "/etc/bind/192.168.0.db";  
  
};
```

15. 在做本地 DNS 攻击时，用 netwox 伪造了响应报文，wireshark 也监听到了伪造报文，为什么客户机解析出来的还是原来的 IP？

解答：伪造的 DNS 响应需要比真实服务器的响应**先**到达客户机才能攻击成功。指导手册中真实服务器也在局域网，所以伪造的响应报文很难比真实服务器响应更快，建议选择外网的域名（比如 baidu.com, 163.com 等）进行测试，这样外网服务器返回响应的时间就要长一些，我们伪造的响应报文可以先于真实响应到达。

16. 远程 DNS 攻击时，在 system 调用 dig 命令之后，为什么伪造的第一个报文都比真实响应还慢？

解答：system("dig www.example.com");
sendto();

system 语句是在执行完 dig 命令之后才会执行 system 之后的下一条 sendto 语句（system 函数会阻塞在此），此时真实的响应报文已经到达 apollo 甚至是攻击主机了，后面再发送的伪造响应报文都肯定在真实响应之后。需要在 dig 命令之后加上 & 符号，让 dig 命令放在后台运行，system 立即返回，即改成 system("dig www.example.com &");

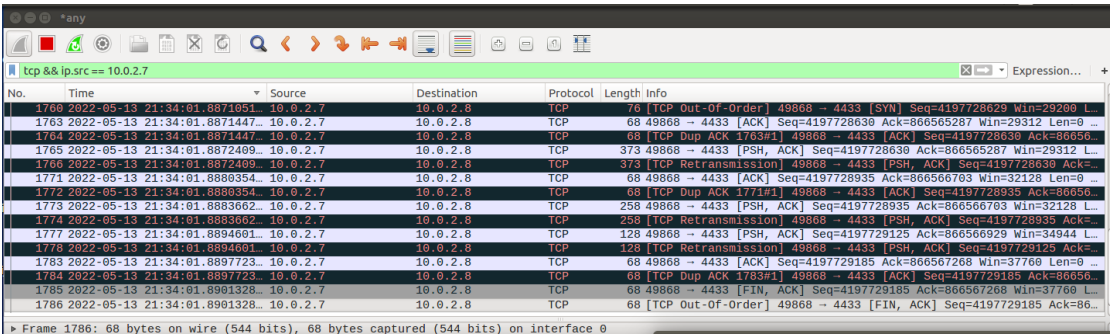
17. VPN 实验中，第一个 vpn 例程运行以后，客户端连接服务器，服务器为什么没有反应？

解答：服务器端运行 `ps aux|grep vpn` 查看是否运行了多个 `vpnserv`，若是，杀掉进程。
原因：vpn 例程中缺少错误处理机制，尤其是 `bind` 函数的返回值没有判断，如果已经运行了一个程序，再运行的时候 `bind` 会出错，正确的处理应该是提示错误，程序退出，不然会造成干扰。

18. VPN 实验中，第一个 vpn 例程运行以后，已经按指导书添加了虚拟网络的路由，但是从客户端主机 ping 内网主机，网络不通

解答：可能的原因：
(1) 同样可能是由于 `vpnserv`，`vpndclient` 运行了多个程序
(2) 虚拟路由添加不完全，请确认客户端主机上到 192.168.60.0/24 和 192.168.53.0/24 的路由转发接口为 `tun0` 接口，VPN 服务器上到 192.168.53.0/24 的路由转发接口为 `tun0`
(3) VM 上的防火墙没清空，`ip_forward` 转发开关未打开都可能造成通信不通的问题

19. vpn 隧道建立以后，wireshark 抓包只能看到 4433 的 tcp 报文，看不到 tls 报文



解答：TLS 的标准端口为 443，需要在 wireshark 中设置 4433 端口解析为 SSL。
具体操作：选择一个到 4433 端口的报文，右键选择 `decode as`，将 4433 端口解析为 SSL

20. tlsclient 和 tlsserver 建立 ssl 不成功，客户端验证服务器证书失败，服务器端提示 unknown ca 错误

```
error:14094418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown  
ca:s3_pkt.c:1487: SSL alert number 48 的错误
```

解答：这是因为没有找到信任的证书链。

在客户机运行 “c_rehash 证书目录” ， c_rehash 为证书目录下的证书文件创建一个符号连接，并将此符号连接的名称设为文件的 hash 值，作用是让 openssl 在证书目录中能够找到证书。

注意：如果程序里面指定 CA 证书路径时指定的是目录，需要做上面的工作，如果指定的是 CA 证书文件路径，不需要以上操作。

21. tlsclient 和 tlsserver 建立 ssl 不成功，客户端验证服务器证书失败，服务器端提示：

```
error:14094416:SSL routines:ssl3_read_bytes:ssl3 alert certificate  
unknown:s3_pkt.c:1487:SSL alert number 46 的错误
```

解答：不知道

22. tlsclient 和 tlsserver 建立 ssl 不成功，客户端验证服务器证书失败，服务器端提示 error:14094415:SSL routines:ssl3_read_bytes: sslv3 alert certificate expired:s3_pkt.c:1487:SSL alert number 45 的错误

解答：这个问题是因为**证书过期**了。如何查看证书的有效期：

windows 下是可以双击证书，在详细信息里面看得到证书有效期
linux 下可以用 openssl 命令查看：`openssl x509 -in 证书文件名 -text` 可以查看证书有效期。

解决办法 1) 懒人办法: 将机器时间改到有效期之内

2) 重新生成证书, 新生成的证书注意有可能因为时区的原因导致 8 小时候后才生效, 也有可能及时生效。

如果程序运行不再报 45 号错误, 那说明这个问题解决了。

23. tlsclient 和 tlsserver 建立不成功, 客户端验证服务器证书失败:

```
root@HostU:/home/seed/tls# ./tlsclient 10.0.2.8 4433
before SSL_connect
after SSL_connect
3070428864:error:14090086:SSL routines:ssl3_get_server_certificate:certificate v
erify failed:s3 clnt.c:1264:
root@HostU:/home/seed/tls#
```

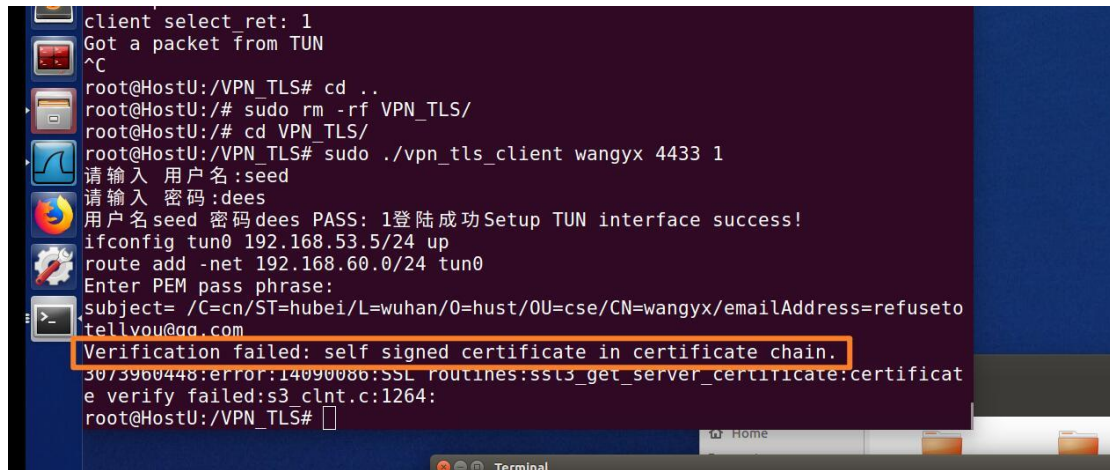
解决办法: 不能用服务器地址来进行访问, 需要在/etc/hosts 文件中增加服务器的域名, 域名名字跟 server 证书主题 (Subject) 的 CN (Common Name) 字段名字一致

```
root@HostU:/home/seed/tls# cat /etc/hosts
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
10.0.2.8 wmzserver
root@HostU:/home/seed/tls#
```

```
root@HostU:/home/seed/tls# ./tlsclient wmzserver 4433
before SSL_connect
after SSL_connect
SSL connection is successful
SSL connection using AES256-GCM-SHA384
HTTP/1.1 200 OK
Content-Type: text/html

<!DOCTYPE html><html><head><title>Hello World</title></head><style>body {backg
und-color: black}h1 {font-size:3cm; text-align: center; color: white;text-shad
: 0 0 3mm yellow}</style></head><body><h1>Hello, world!</h1></body></html>
```

24. tlsclient 和 tlsserver 建立不成功，客户端验证服务器证书失败：



```
client select ret: 1
Got a packet from TUN
^C
root@HostU:/VPN_TLS# cd ..
root@HostU:/# sudo rm -rf VPN_TLS/
root@HostU:/# cd VPN_TLS/
root@HostU:/VPN_TLS# sudo ./vpn_tls_client wangyx 4433 1
请输入 用户名:seed
请输入 密码:dees
用户名seed 密码dees PASS: 1登陆成功Setup TUN interface success!
ifconfig tun0 192.168.53.5/24 up
route add -net 192.168.60.0/24 tun0
Enter PEM pass phrase:
subject= /C=cn/ST=hubei/L=wuhan/O=hust/OU=cse/CN=wangyx/emailAddress=refuseto
tellyou@qq.com
Verification failed: self signed certificate in certificate chain.
3073960448:error:14090086:SSL routines:ssl3_get_server_certificate:certificat
e verify failed:s3_clnt.c:1264:
root@HostU:/VPN_TLS#
```

解答：客户端程序也需要指定 CA 证书的路径（参见服务器端程序），因为这个 ca 是自己生成的自签名证书，不是可信任的根证书，所以验证的时候报错，程序加载 ca 证书，就等于信任了该 CA 证书

25 .crt 文件和.pem 文件有什么差别

解答：.pem 和.crt 都是证书文件，只是文件名的不同而已，他们的格式都是 PEM 格式，

在程序代码中都符合 SSL_FILETYPE_PEM 格式。在 windows 下一般用.crt 后缀，可以直接双击就可以解析证书文件内容。直接.pem 文件的后缀改成.crt 就能在 windows 下解析了。linux 下证书文件可以用以下命令查看：

`openssl x509 -in 证书文件 -text`

26. VPN 实验中，最后支持多客户端那个部分，内网主机返回的报文到达 VPN 服务器以后，tun0 中的数据应该向哪个隧道转发？怎么区分？

解答：解决方式有多种：

1. 可以针对每个隧道做一个会话记录，记录客户端隧道 ip（真实地址）、分配的虚 ip 地址、服务器端连接的套接字、跟客户端 ssl 会话的 SSL 指针等。那么就可以根据返回报文的目的地址（虚 ip 地址）来查会话表，从而确定跟客户端的 SSL 信息。也就是说最好在建立

立隧道的时候，服务器能有分配虚 ip 的机制，告诉客户端该虚 ip 地址信息，并且服务器在内存维护这样的一个会话表。

2. 还有另一种实现方式，openvpn 是一个开源的典型的 sslvpn 软件，它既有上面的这种模式，也有子网模式，每个客户端跟服务器建立隧道以后，服务器给每个客户端分配一个 /30 的虚拟子网（一个虚地址给客户端 tun 接口，一个地址给服务器端的虚接口），服务器为每个隧道启动 1 个 tun 接口，这样服务器就会有多个 tun 接口，tun0、tun1....，每个 tun 接口上会有对应的虚拟子网路由，内网主机返回的报文，直接根据目的地址（虚 IP 地址）查找路由就可以找到对应的 tun 接口。（此方案服务器可以采用多进程/线程，如下图所示，看蓝色手绘标注线，只适合小规模网络，不是最佳方案，但是至少可实现）

