# 基于scapy的流量构造程序

## 程序运行

程序2.3.4先运行后 再运行1

即

Src(1.py)

Dst-proxy（4.py）

）

## tcp方面的计算问题

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 包总类（前半格表示发送端发出，后半~） | | 十六进制FLAGS值 | seq | ack | len |  |  |  |
| SYN |  | 0X002 | 0 | null | 0 |  |  |  |
|  | SYN+ACK | 0X012 | 0 | 1 | 0 |  |  |  |
| ACK |  | 0X010 | 1 | 1 | 0 |  |  |  |
| PSH+ACK |  | 0X018 | 1 | 1 | x |  |  |  |
|  | ACK |  | 1 | x+1=y | 0 |  |  |  |
| PSH+ACK |  |  | y | 1 | m |  | 分片计算时 |  |
| PSH+ACK |  |  | y+m | 1 | n |  |  |  |
|  | ACK |  | 1 | y+m+n | 0 |  |  |  |
| 若干PA与A后 | | | | | | | | |
|
|
| FIN+ACK+PSN |  | 0X019 | 前项的seq+len | 1 | z |  |  |  |
|  | ACK |  | 1 | 前前项的seq+len+z +1=k | 0 |  | 换句话说这俩个包完全相同 |  |
|  | FIN+ACK | 0X011 | 1 | k | 0 |  |  |
| ACK |  |  | k | 2 | 0 |  | 此时k数值上等于发送端所有发包的总负载长度+2（握手时的一次加1 和挥手时的一次加1） |  |

## tshrak读取pcap文件（例）

tshark -r test.pcap -T fields -e frame.number -e frame.time\_relative -e frame.len -E header=y > test.csv

## 防火墙配置

centos7

1. firewall-cmd --direct --add-rule ipv4 filter OUTPUT 1 -p tcp --tcp-flags RST RST -j DROP #对应rst包问题
2. firewall-cmd --add-port=80/tcp --permanent #对应icmp 管理员拒绝包
3. firewall-cmd –reload

window下路由可能有icmp重定向问题