

20200620信息安全实训

笔记本： 我的第一个笔记本

创建时间： 2020/6/20 8:56

更新时间： 2020/6/20 17:32

作者： 820410740@qq.com

网络协议基础

截取网络数据 socket编程



ASCII码

bytes 字节 bits 比特
一个字节包括8个比特

二进制数据

b' ' bytes类型
\xff 二进制

帧

传输需求1

1. 区分终端
2. 避免“泛洪”

MAC地址

RFC-source
0x0800 IP
0x0806 ARP
0x0835 DRARP RARP
0x814C SNMP
0x86DD IPv6

删除用apt下载的包

删除软件及其配置文件

apt-get --purge remove <package>

删除没用的依赖包

apt-get autoremove <package>

此时dpkg的列表中有“rc”状态的软件包，可以执行如下命令做最后清理：

```
dpkg -l |grep ^rc|awk '{print $2}' |sudo xargs dpkg -P
```

定义函数解析ethernet头

```
def mac_bytes_to_str(h):  
    return '%02x:%02x:%02x:%02x:%02x:%02x'%(  
        b[0], b[1], b[2], b[3], b[4], b[5]  
    )
```

```
eth_h = frame[0:14]
```

```
src_mac = mac_bytes_to_str(eth_h[6:12])  
dst_mac = mac_bytes_to_str(eth_h[0:6])
```

```
print(src_mac, dst_mac)
```

```
08:00:27:77:95:32 90:86:9b:86:b2:a0  
90:86:9b:86:b2:a0 08:00:27:77:95:32  
e0:69:95:ad:7b:3e ff:ff:ff:ff:ff:ff  
e0:69:95:ad:7b:3e ff:ff:ff:ff:ff:ff  
f8:a9:63:bb:dd:5b ff:ff:ff:ff:ff:ff  
e0:69:95:ad:7b:3e ff:ff:ff:ff:ff:ff  
e0:69:95:ad:7b:3e ff:ff:ff:ff:ff:ff
```

ethernet层定义了下一层使用的协议 (ether type)

0x0800 IP

0x0806 ARP

0x0835 RARP

0x814C SNMP

.....

网络协议基础

传输需求1

格式化输出函数

```
import curses
```

```
def frame_bytes_to_hex_str(b):  
    return ['%02x'%x for x in b]
```

```
def display(frame_list):  
  
    def main(stdscr):  
  
        stdscr.clear()  
  
        max_y, max_x = stdscr.getmaxyx()  
  
        x = 0  
        y = 0  
        for i, v in enumerate(frame_list):  
  
            if y > max_y - 4:  
                stdscr.addstr(y, x, '..')  
                stdscr.refresh()  
                break  
  
            stdscr.addstr(y, x, v)  
  
            if (i+1) % 8 == 0:  
                x += 6  
            else:  
                x += 3  
  
            if x > 80:  
                x = 0  
                y += 1  
  
            stdscr.refresh()
```

```
curses.wrapper(main)
```

传输需求2

传输需求2

1. 网段跳转

2. 传输路径

3. 内网空间

报

定义函数解析ip头

OSPF协议 最短路径协议

L2TP协议 二级隧道协议

定义函数解析ip头

```
def ip_bytes_to_str(b):  
    return '%d.%d.%d.%d'%(b[0], b[1], b[2], b[3])  
  
if ether_type == 'IP':  
    ip_h = frame[14:34]  
  
    ip_src = ip_bytes_to_str(ip_h[12:16])  
    dst_src = ip_bytes_to_str(ip_h[16:20])  
  
    print(ip_src, dst_src)
```

传输需求2

Ip层定义了下一层使用的协议 (ip protocol)

1 icmp

6 tcp

17 udp

.....

C段扫描 C段旁注

传输需求3

1. 建立“连接” 流媒体
2. 切分、组合
3. 缺块重发
4. 分块大小
5. 断开“连接”
6. 数据校验

控制比特 (Control Bits)

紧急 URG —— 当 $URG = 1$ 时，表明紧急指针字段有效。它告诉系统此报文段中有紧急数据，应尽快传送(相当于高优先级的数据)。

确认 ACK —— 只有当 $ACK = 1$ 时确认号字段才有效。当 $ACK = 0$ 时，确认号无效。

推送 PSH (PuSH) —— 接收 TCP 收到 $PSH = 1$ 的报文段，就尽快地交付接收应用进程，而不再等到整个缓存都填满了后再向上交付。

复位 RST (ReSeT) —— 当 $RST = 1$ 时，表明 TCP 连接中出现严重差错(如由于主机崩溃或其他原因)，必须释放连接，然后再重新建立运输连接。

同步 SYN —— 同步 $SYN = 1$ 表示这是一个连接请求或连接接受报文。

终止 FIN (FINish) —— 用来释放一个连接。 $FIN = 1$ 表明此报文段的发送端的数据已发送完毕，并要求释放运输连接。

提取TCP头内容

切片排序，接收回应



SYN洪水攻击

护网禁止DDOS攻击

TCP没有规定下一层的应用层协议

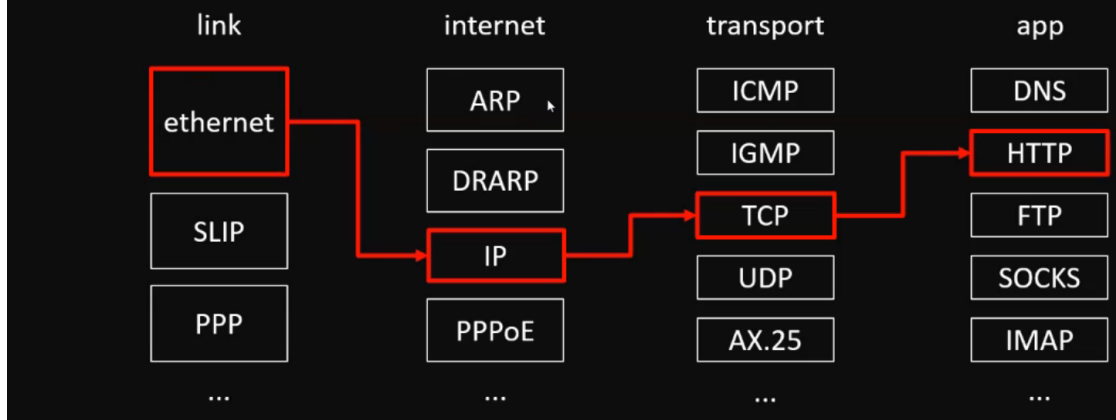
SMTP协议 邮件协议

DNS协议 域名解析

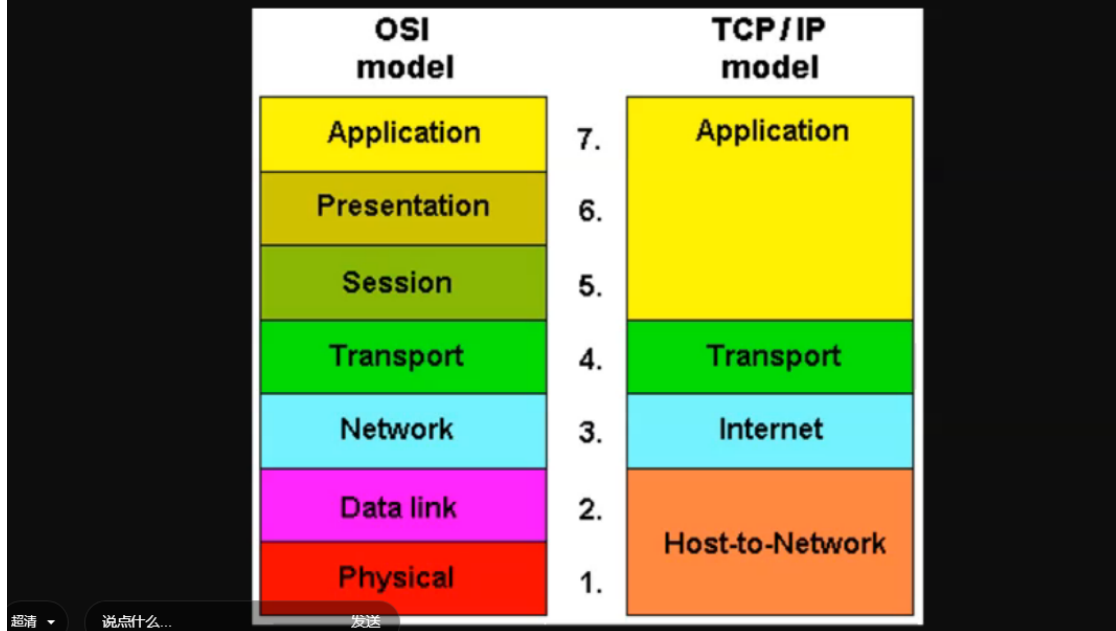
FTP协议 文件传输

、
TCP/IP协议栈（四层）

TCP/IP协议栈（四层）



OSI模型（七层） vs TCP/IP模型



抓包与网络层攻击

网络层攻击

Dos (Denial Of Service) 攻击

发送大量http请求，占用连接资源

SYN洪水攻击

hping3工具

hping3工具

```
sudo apt-get install hping3
```

```
sudo hping3 -S -p 80 -flood -V --rand-source 192.168.1.11
```

Ddos攻击 (分布式Dos攻击)
CC攻击 (Challenge Collapsar)

1. 模拟真实用户的行为
2. 针对不同的资源