网络协议基础及安全性分析

王轶骏， Eric Ericwyj@sjtu.edu.cn

SJTU.INFOSEC.A.D.T, 2008

前言

- 充分理解TCP/IP协议栈的意义
  - TCP/IP协议栈应用极其广泛。
    - 作为现代操作系统的一部分，许多上层应用都是基于TCP/IP协议栈的，协议本身存在的安全缺陷给操作系统及上层应用带来了极大的隐患。
    - 充分地理解TCP/IP的协议格式，才能更好地发现并分析所遭受到的攻击。
  - 许多安全设备都是基于协议分析的。
    - 只有充分理解协议，才能更好地部署并控制这些设备（例如防火墙、IDS等）。
  - TCP/IP协议栈也是进行网络管理的基础。
  - 只有充分理解TCP/IP协议栈，才能更好的开发网络系统及软件。

大纲

- 协议和参考模型
- 链路层协议及安全性分析
- 网络层协议及安全性分析
- 传输层协议及安全性分析
- 应用层协议及安全性分析

主题 1

- 协议和参考模型
  - 协议和分层结构
  - OSI参考模型
  - TCP/IP参考模型
  - TCP/IP协议栈
- 链路层协议及安全性分析
- 网络层协议及安全性分析
- 传输层协议及安全性分析
- 应用层协议及安全性分析

1.1 协议和分层结构

- 协议
- 分层结构的优点
- 分层结构的工作原理

1.1.1 协议

- 什么是协议?
  - 协议是网络中计算机或设备之间进行通信的一系列规则的集合。

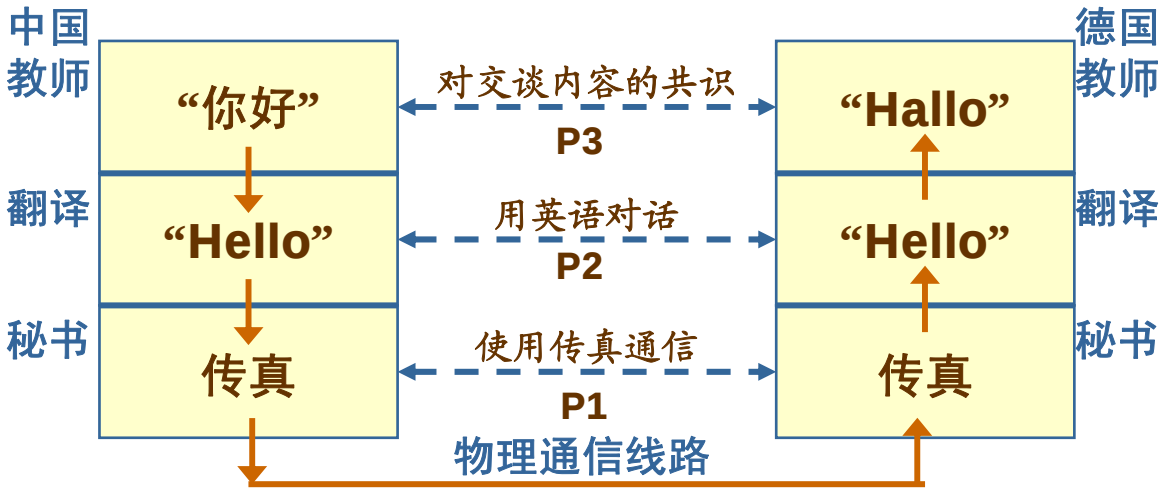
协议示例  A方向B方发送消息“HELLO ERIC”

0	1	0	H	E	L	L	O		E	R	I	C

- 协议是分层的。
  - OSI参考模型
  - TCP/IP模型

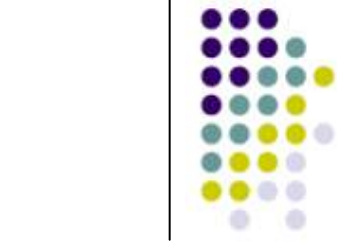
1.1.2 协议的分层结构

- 分层结构的工作原理
  - 纵向通信
    - 低层服务为高层服务提供服务（Service Provider），高层服务使用低层服务提供的服务（Service User）。
  - 横向通信
    - 对应的分层协同工作，以保证能够成功的完成通信。
- 分层结构的优点
  - 各层之间相互独立，某一层的变化不会影响其他层。
  - 促进标准化工作。
  - 使网络易于实现和维护。



对等通信实例

1.2 OSI参考模型



第 7 层	应用层	Application
第 6 层	表示层	Presentation
第 5 层	会话层	Session
第 4 层	传输层	Transport
第 3 层	网络层	Network
第 2 层	链路层	Data link
第 1 层	物理层	Physical

ISO在1974年颁布了0SI／RM模型。
该模型分为七个层次，也称为OSI七层模型。

- **物理层**
  - 通过物理传输比特（bit）流。
  - 建立、维护和取消物理连接。
- **数据链路层**
  - 将比特信息加以组织封装成数据帧（Frame）。
  - 通过使用接收系统的硬件地址或物理地址来寻址。
  - 包含两个子层：媒体访问控制（MAC）和逻辑链路控制（LLC）。
- **网络层**
  - 基于网络层地址（IP地址）进行不同网络系统间的路径选择。
  - 分割和重新组合数据包（Packet）。
  - 差错检验和可能的修复。
  - 可能的数据流量控制。

- **传输层**
  - 在不同物理节点上的应用程序间建立连接以传输数据。
    - 将数据组织成数据段（Segment）
  - 连接类型包括两种：面向连接（Connection-oriented）和无连接（Connectionless）
  - 用一个寻址机制来标识一个特定的应用程序。
    - 传输层地址（即端口号）
- **会话层**
  - 建立、管理和终止会话。
- **表示层**
  - 系统的应用层送出的信息可被另一个系统的应用层所读取。
  - 利用一种公用的信息表示格式翻译多种信息。
  - 数据表示、数据安全、数据压缩。
- **应用层**
  - 网络服务与使用者应用程序间的一个接口。

1.3 TCP/IP参考模型

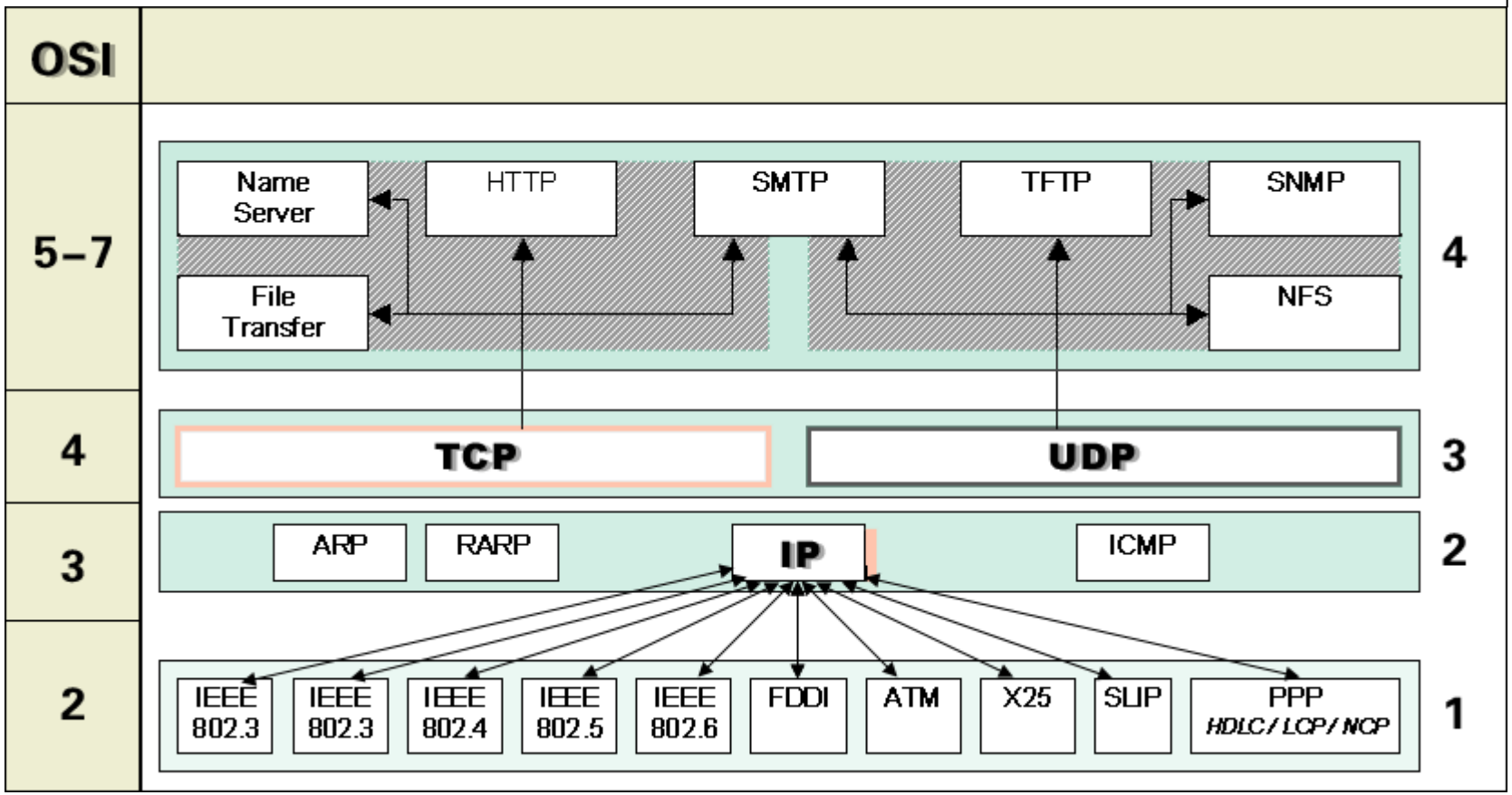
- TCP/IP参考模型和OSI参考模型的对应关系

- **第1层：网络接口层（Network Interface）**
  - 网络接口层对应OSI物理层和数据链路层并实现与它们相同的功能，其中包括LAN和WAN的技术细节。
- **第2层：互联网络层（internet）**
  - 互联网络层的目的是运送数据包，将数据从任何在相连的网络上送到目的地，而不在乎走的是哪个路径或网络。
- **第3层：传输层(Transport)**
  - 传输层负责处理有关服务质量等事项，如可靠度、流量控制和错误校正。该层可以提供不同服务质量、不同可靠性保证的传输服务，并且协议发送端和目标端的传输速度差异。
- **第4层：应用层(Application)**
  - 应用层包括会话层和表示层的功能，用来建立应用层来处理高层协议、有关表达、编码和会话控制。TCP/IP将所有应用程序相关的内容都归为一层，并保证为下层适当的将数据封装成数据包。

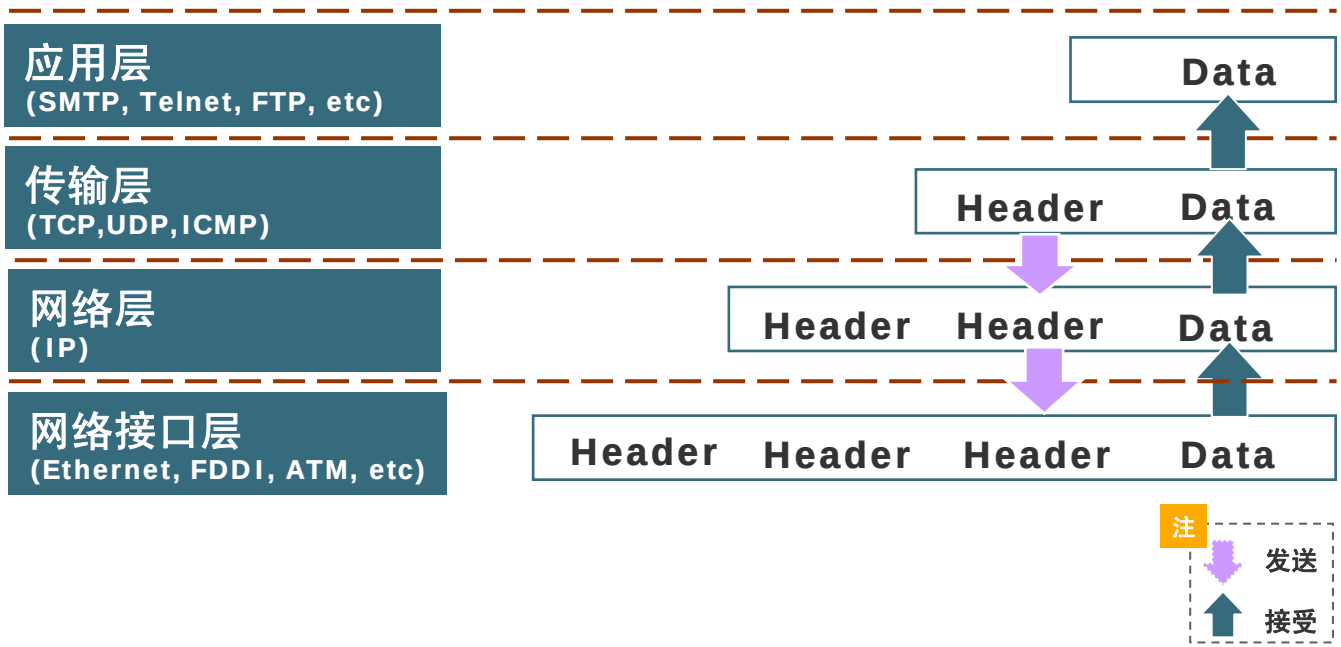
1.4 TCP/IP协议栈

- 协议栈概述
  - 在网络中，为了完成通信，必须使用多层上的多种协议。这些协议按照层次顺序组合在一起的实现就是协议栈（Protocol Stack），也称为协议族（Protocol Suite）。
- 常用的协议栈
  - TCP/IP
  - IPX/SPX
  - AppleTalk

TCP/IP协议栈


OSI
5–7
4
3
2
Name Server
HTTP
SMTP
TFTP
SNMP
File Transfer
NFS
4
TCP
UDP
3
ARP
RARP
IP
ICMP
2
IEEE 802.3
IEEE 802.3
IEEE 802.4
IEEE 802.5
IEEE 802.6
FDDI
ATM
X25
SLIP
PPP
HDLC / LCP / NCP
1

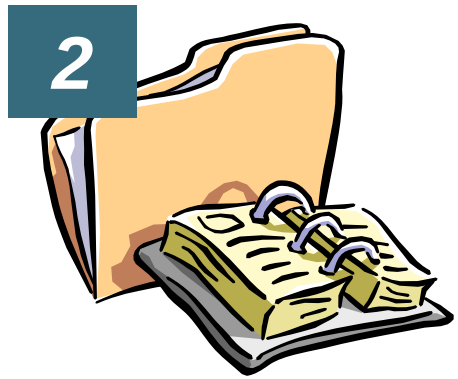
协议的封装



- 在每层，分组由两部分构成：
  - 首部（头部）：包含与本层相关的协议信息。
  - 本体（数据）：包含本层的所有数据。
- 每层分组要包含来自上层的所有信息，同时加上本层的首部，即封包。
- 最上层的应用层包含的就是要传送出去的数据。

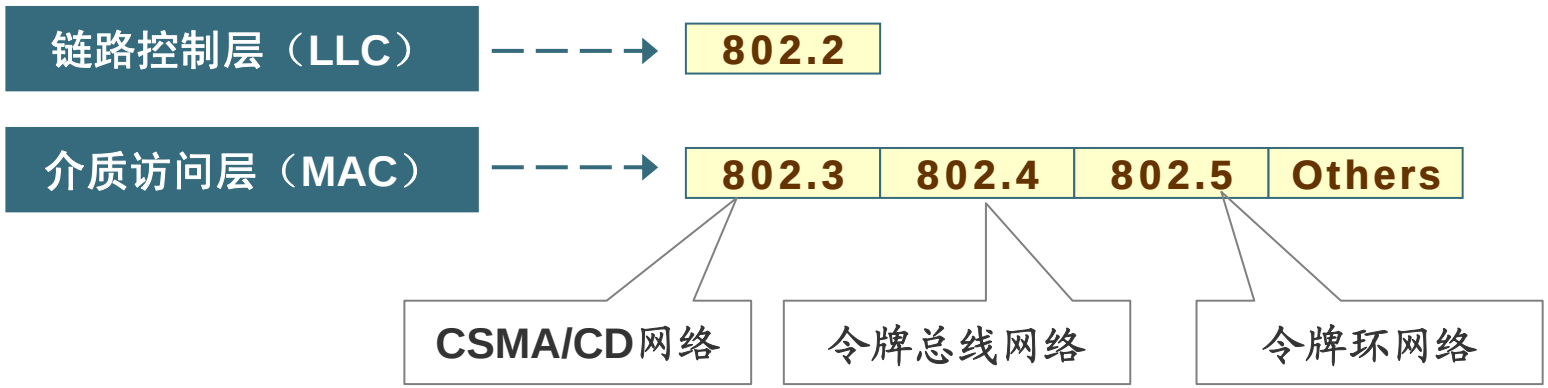
主题 2

- 协议和参考模型
- 链路层协议及安全性分析
  - 数据链路层协议规范
  - 以太网的封装格式
  - 网卡的MAC地址
  - ARP协议及其安全性
- 网络层协议及安全性分析
- 传输层协议及安全性分析
- 应用层协议及安全性分析

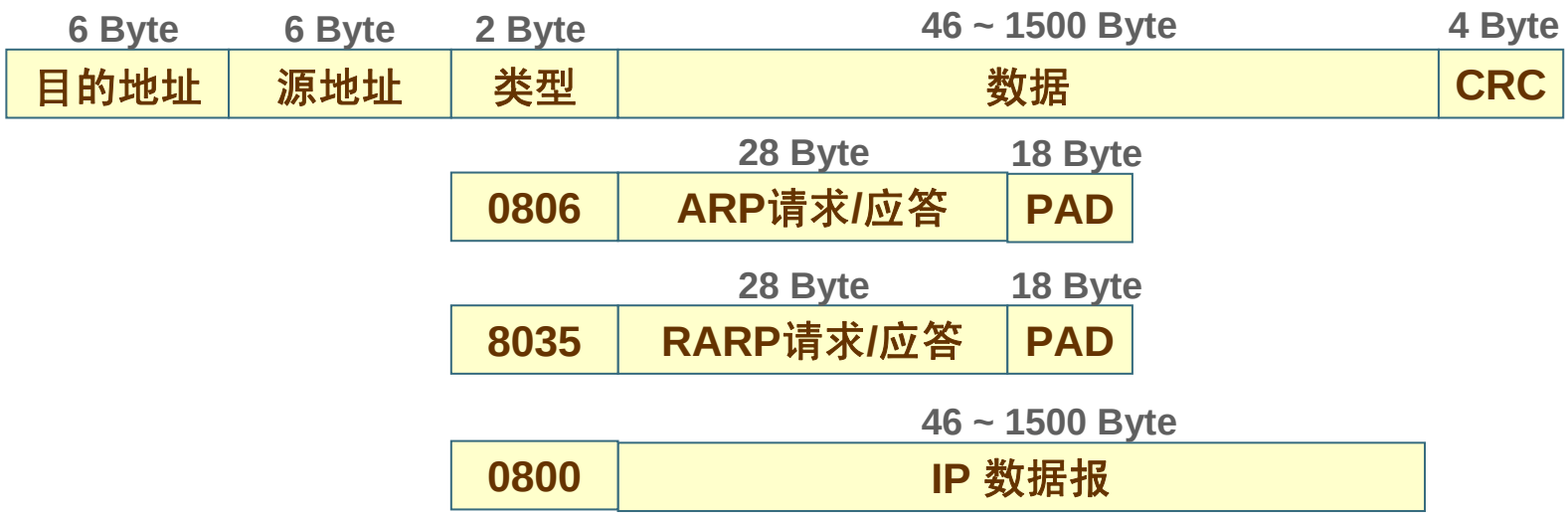


2.1 数据链路层协议规范

IEEE（电子电气工程师协会）
802局域网规范



2.2 以太网的数据封装格式



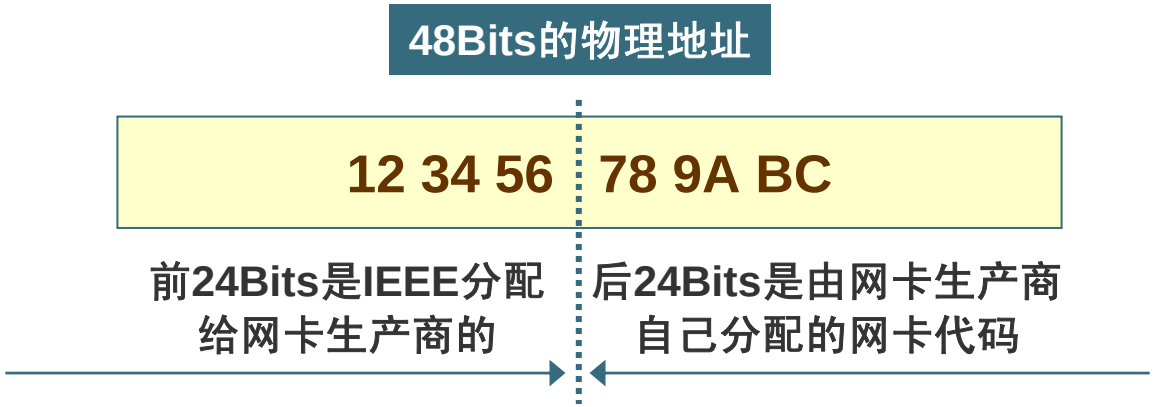
RFC 894

注

以太网最小传输单元 — 46 Byte。
以太网最大传输单元 — 1500 Byte。

2.3 网卡的MAC地址

- 网卡的MAC地址也称作物理地址、硬件地址。



注

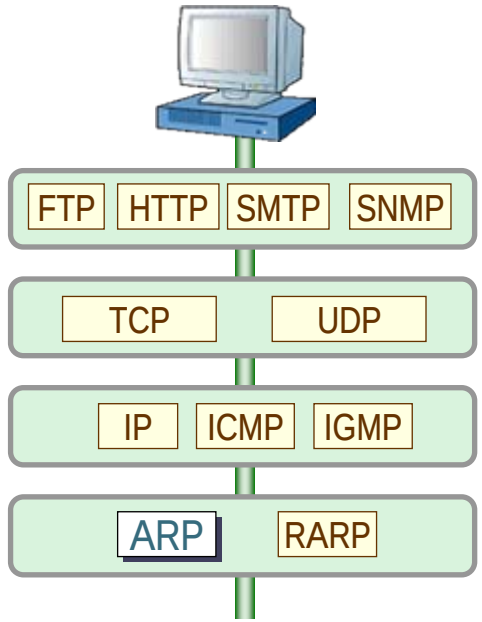
FF FF FF FF FF FF表示广播地址。

2.4 ARP协议及其安全性

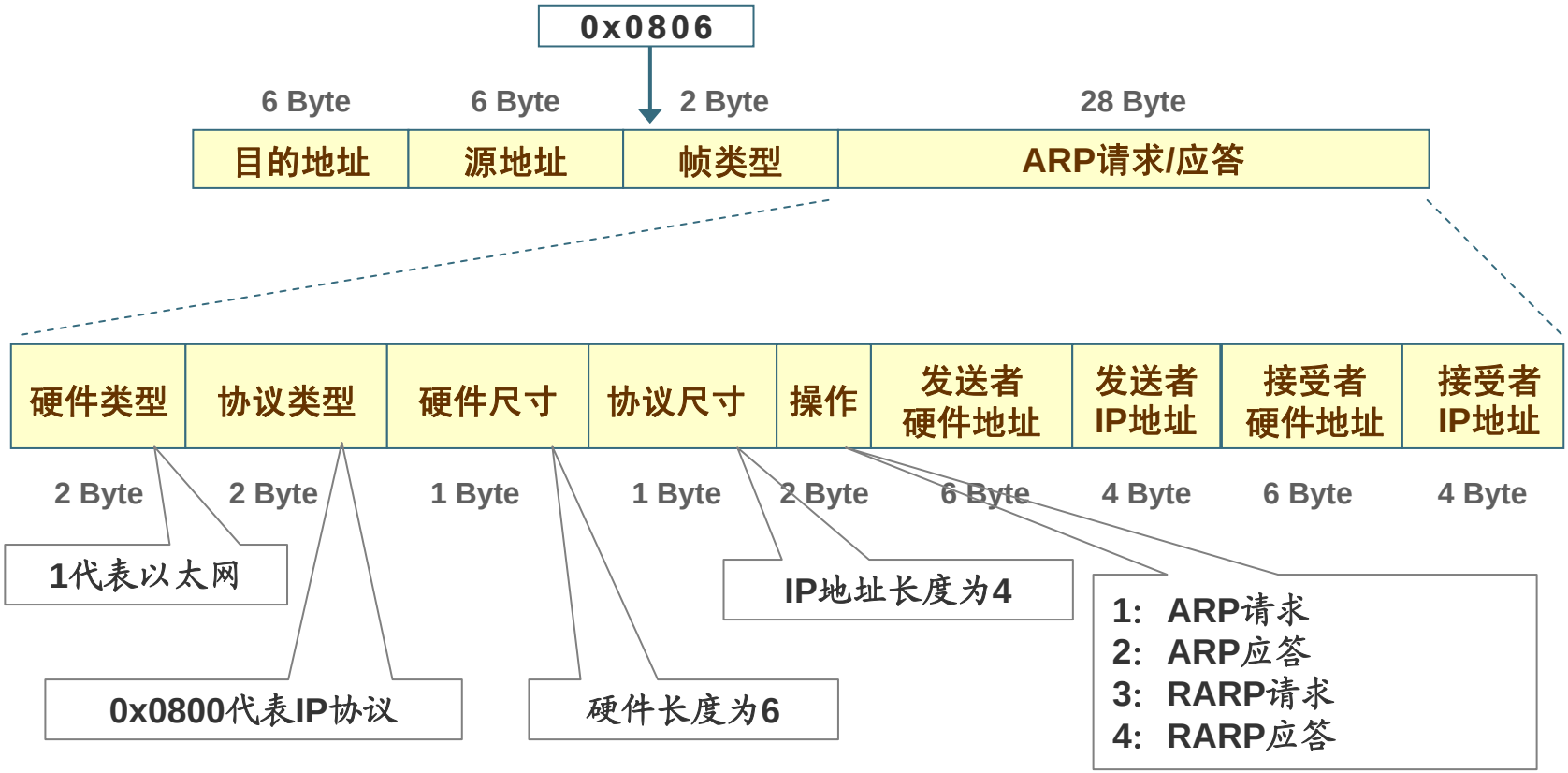
- ARP协议的基本概念
- ARP地址解析过程
- 基于ARP协议的欺骗攻击

2.4.1 ARP协议的基本概念

- **ARP（Address Resolution Protocol，地址解析协议）**
  - ARP协议的作用就是根据目标主机的IP地址来解析其MAC地址，并将解析过的IP地址和MAC地址的对应关系保存在系统ARP缓存中。
- **RARP（Reverse Address Resolution Protocol，反向地址解析协议）**



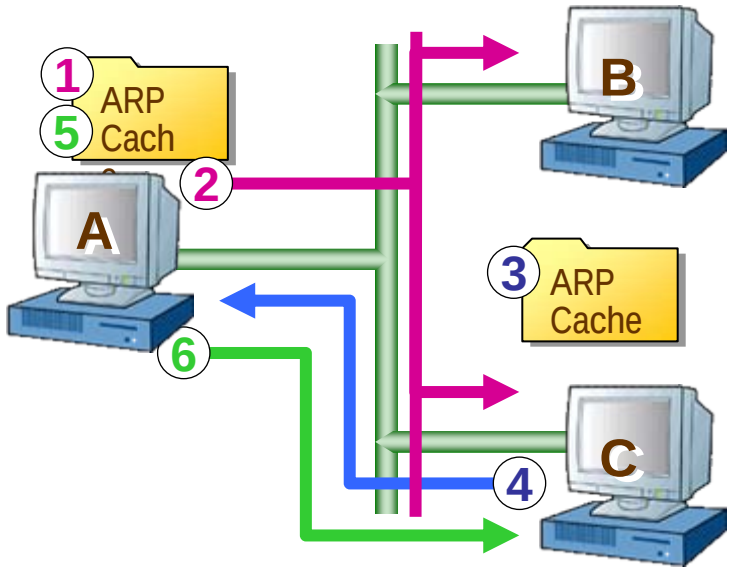
ARP协议数据包格式


0x0806
6 Byte
6 Byte
2 Byte
28 Byte
目的地址
源地址
帧类型
ARP请求/应答
硬件类型
协议类型
硬件尺寸
协议尺寸
操作
发送者硬件地址
发送者IP地址
接受者硬件地址
接受者IP地址
2 Byte
2 Byte
1 Byte
1 Byte
2 Byte
6 Byte
4 Byte
6 Byte
4 Byte
1代表以太网
0x0800代表IP协议
硬件长度为6
IP地址长度为4
1: ARP请求
2: ARP应答
3: RARP请求
4: RARP应答

2.4.2 ARP地址解析过程

示例  主机A访问服务器C

1. A检查本地的ARP缓存。
2. A在网络中发出ARP广播请求。
3. C将A的MAC加入ARP缓存中。
4. C回应ARP消息。
5. A将C的MAC加入本地的ARP缓存中。
6. A使用IP协议向C发送数据包。



管理主机的ARP缓存表

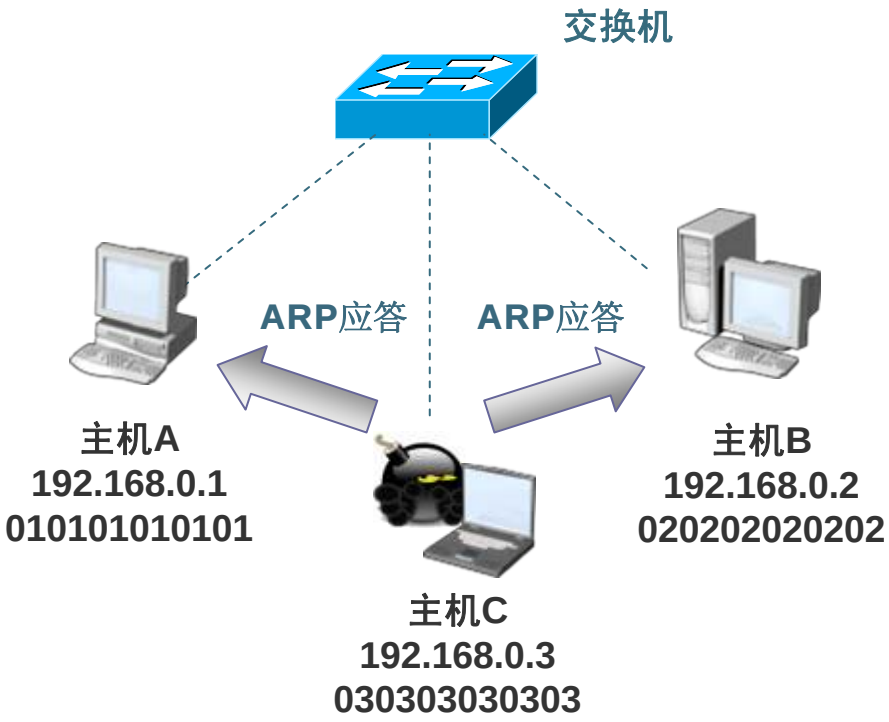
- **ARP命令**
  - 添加静态的地址映射表项
    - **arp –s 172.16.15.1 00-01-02-03-04-05**
  - 删除静态的地址映射表项
    - **arp –d <IP Address>**
  - 显示主机当前ARP缓存中的所有地址映射表项
    - **arp -a**

```
# arp -a
Interface: 172.16.15.116 on Interface 0x1000005
  Internet Address      Physical Address      Type
  172.16.15.91          00-0c-76-c7-b7-81     dynamic
  172.16.15.1           00-01-02-03-04-05     static
```

注

静态的地址映射表项是在系统运行期间一直存在的，而不会动态刷新。这即是平时所说的IP地址与MAC地址的绑定。

2.4.3 ARP欺骗攻击

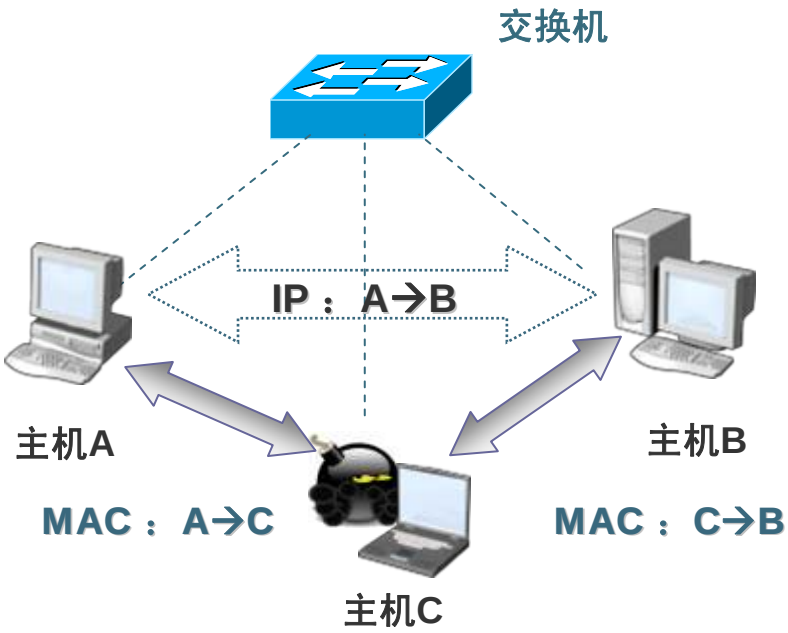


C发给A的ARP应答包

010101010101	000000000000	0806
0001	0800	06	04	0002
030303030303	192.168.0.2	
000000000000	192.168.0.1	

C发给B的ARP应答包

020202020202	000000000000	0806
0001	0800	06	04	0002
030303030303	192.168.0.1	
000000000000	192.168.0.2	



A发给C的数据包中

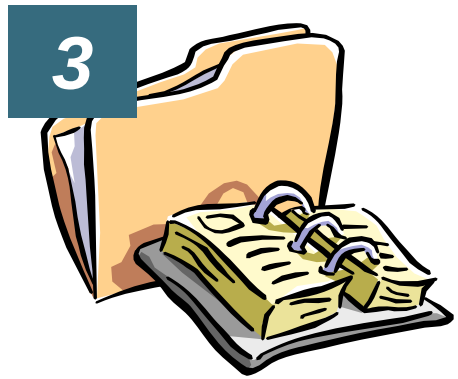
源MAC:	010101010101
源IP:	192.168.0.1
目的MAC:	030303030303
目的IP:	192.168.0.2

C发给B的数据包中

源MAC:	030303030303
源IP:	192.168.0.1
目的MAC:	020202020202
目的IP:	192.168.0.2

主题 3

- 协议和参考模型
- 链路层协议及安全性分析
- 网络层协议及安全性分析
  - IP协议及安全性
  - ICMP协议及安全性
  - IGMP协议及安全性
- 传输层协议及安全性分析
- 应用层协议及安全性分析



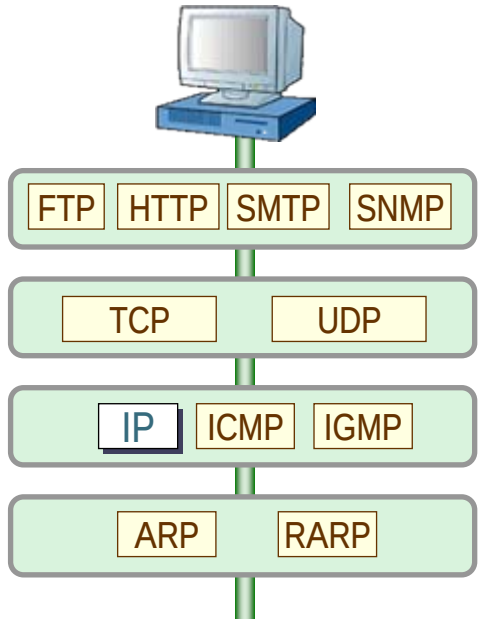
3.1 IP协议及安全性

- IP协议的基本概念
- IP协议的数据报头格式
- IP地址格式
- IP协议的安全性分析

RFC 791

3.1.1 IP协议的基本概念

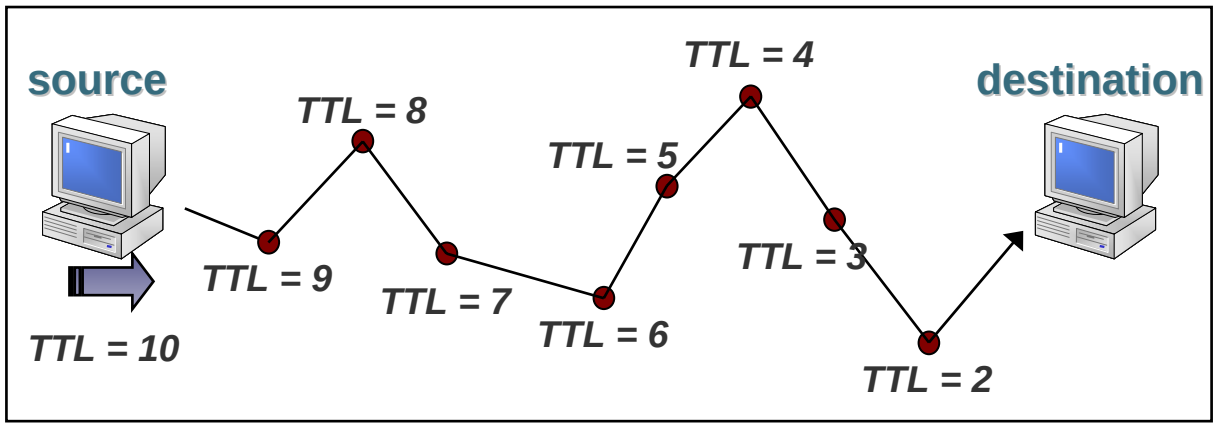
- IP（Internet Protocol）协议提供的是一种不可靠的、无连接的数据报传输服务。
  - 不可靠：IP尽最大努力进行数据报传送，但不保证能够成功到达目的地。错误发生时，通过ICMP消息来通知，任何必须的可靠性由上层协议（如TCP）来提供。
  - 无连接：IP并不维护关于连续发送的数据报的任何状态信息，每个数据报都被单独处理，每个数据报独立寻径，传送过程中可能出现错序。
- TCP、UDP、ICMP以及IGMP等协议都是通过IP数据报来传送的。



3.1.2 IP协议的数据报头格式

- 协议版本号
  - 当前的IPv4为0x04，今后的IPv6为0x06。
- 首部长度
  - 首部占32Bit（4Bytes）的个数，因此首部最长为60Bytes。
- 服务类型（TOS）
  - 3bit的优先权子字段（现已忽略）。
  - 4bit的TOS子字段，分别代表：
    - 最小时延（如Telnet和Rlogin）。
    - 最大吞吐量（如FTP文件传输）。
    - 最高可靠性（如网络管理SNMP和路由选择协议）。
    - 最小费用（如用户网络新闻NNTP）。
  - 1bit的未用位（置0）。

- 总长度
  - 整个IP数据报（包括首部和数据）的长度。
  - 占16Bit，因此IP数据报的最长可达65535Bytes。
- 标识符
  - 唯一标识主机发送的每一份数据报。
  - 通常每发送一份报文，该值就会加1。
- 3bit的标志和13bit的偏移
  - IP数据报分片和重组时使用。
- 生存时间（TTL, Time to Live）

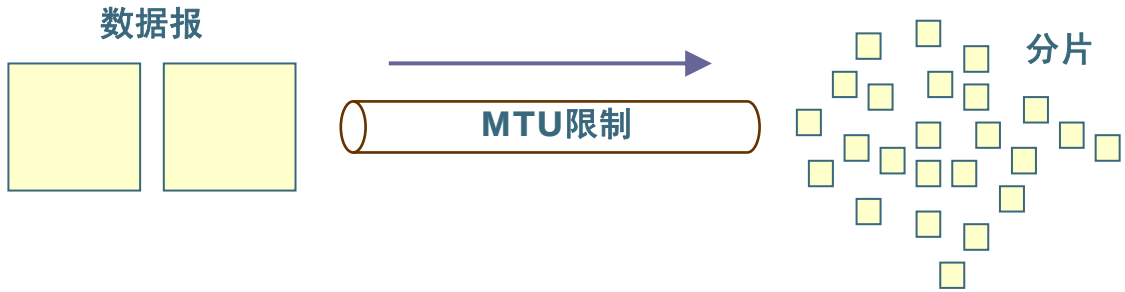


- **协议**
  - 用以识别哪种协议向IP传送数据。
- **首部检验和**
  - 不对首部后面的数据进行计算。
  - 发送端的计算方法：
    - 把检验和字段置为0。
    - 对首部中每个16bit进行二进制反码求和，存于检验和字段中。
  - 接受端的校验方法：
    - 对首部中每个16bit进行二进制反码求和。
    - 如果首部在传输过程中没有任何差错，那么计算结果应为全1；否则检验和错误，那么IP就丢弃收到的数据报，但不生成差错报文，由上层去发现丢失的数据报并进行重传。

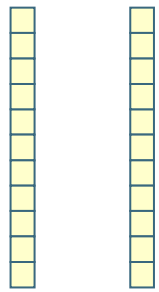
可选项

- 可变长，在必要的时候插入值为0的填充字段。
- 可作如下定义：
  - 安全和处理限制（用于军事领域）。
  - 记录路径（让每个路由器都记下它的IP地址）。
  - 时间戳（让每个路由器都记下它的IP地址和时间）。
  - 宽松的源站选路（为数据报指定一系列必须经过的IP地址）。
  - 严格的源站选路（要求数据报只能经过指定的这些地址，不能经过其他的地址）。

IP数据报分片

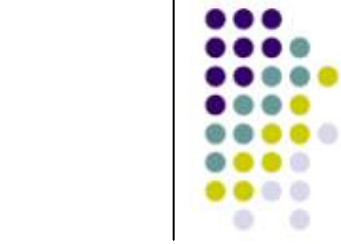


- 数据报到达目的地时才会进行组装。
- 需要组装在一起的数据分片具有同样的ID号。
- 通过分片偏移量来指示数据分片在数据报组装过程中的位置。
- 除了最后的数据分片，其他的数据分片均会置“MF”位（为1），表示后面还有其他分片。
- 如果“DF”位置为1，表示不允许传输过程中路由器对数据进行分片。
- 正常的分片包其Flag应该为“001”。



接收主机的分片重组缓冲区

3.1.3 IP地址格式




A类
1bit
7bit
24bit
0
网络号
主机号
0.0.0.0 –
127.255.255.255
B类
2bit
14bit
16bit
1
0
网络号
主机号
128.0.0.0 –
191.255.255.255
C类
3bit
21bit
8bit
1
1
0
网络号
主机号
192.0.0.0 –
223.255.255.255
D类
4bit
28bit
1
1
1
0
组播组标识符
224.0.0.0 –
239.255.255.255
E类
5bit
27bit
1
1
1
1
0
保留
240.0.0.0 –
247.255.255.255

保留IP地址

- RFC1918规定了3段保留地址，这些地址保留给内部网用户，不可以联入Internet。
- 许多组织使用这些地址作为试验环境或者内部网络地址，通过代理服务器或NAT技术与外部网络连接。

10.0.0.0 ~ 10.255.255.255

172.16.0.0 ~ 172.31.255.255

192.168.0.0 ~ 192.168.255.255

3.1.3 IP协议安全性分析

- **IP欺骗的问题**
  - IP数据报在路由过程中并不检查源地址，源地址可以被任意填写。
  - 通过IP地址欺骗，黑客可以绕过防火墙，获取信任，进行DoS攻击，进行会话劫持等等。
- **IP分片的问题**
  - 某些操作系统在实现TCP/IP协议栈时，对IP分片的处理存在问题，可能遭受Ping of Death、TearDrop等类型的攻击。
- **IP选项的问题**
  - IP数据报头部中允许有若干选项，但并不常用，某些选项域可能被黑客利用，如源路由选项。

IP源路由欺骗攻击

- 攻击方式
  - 源路由允许发送者指定数据包到达目的地前在Internet上经由的路径。
  - 攻击者可以使用源路由进行地址欺骗。

例如：想通过某个邮件服务器（内网地址192.168.1.1）发送垃圾邮件，首先将本机的IP地址配置成该服务器所在网络的某个可信IP，然后打开一个指定源路由的TCP连接到达目的地址。

```
# ifconfig eth0 192.168.1.2 netmask 255.255.255.255

# nc -g 10.0.0.254 -g 10.1.0.254 g 192.168.22.254 -g 192.168.1.1 25
```

- 防御对策：阻止源路由

```
# echo 0 > /proc/sys/net/ipv4/conf/eth0/accept_source_route
```

IP转发隐患

- 攻击方式
  - 如果双IP地址主机被配置成在两个网络（内部网络和外部网络）之间路由数据包，则攻击者就可能通过该主机访问隐藏在内部网络的主机。
- 防御对策：关闭IP转发
  - 大部分的Linux系统默认都关闭IP转发功能。
  - 如果开启了，则可作如下设置：

```
# echo 0 > /proc/sys/net/ipv4/ipv4/ip_forward

# vi /etc/sysctl.conf
…
net .ipv4.ip_forward = 0
```

3.2 ICMP协议及安全性

- ICMP协议的基本概念
- ICMP协议的数据报头格式
- ICMP协议的实用工具分析
- ICMP协议的安全性分析

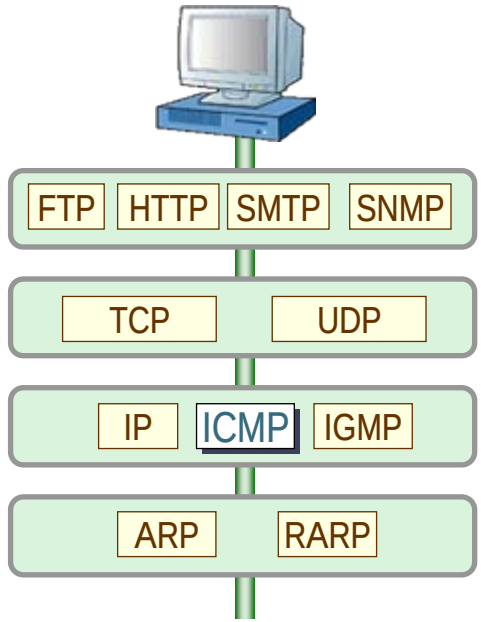
RFC 792

3.2.1 ICMP协议的基本概念

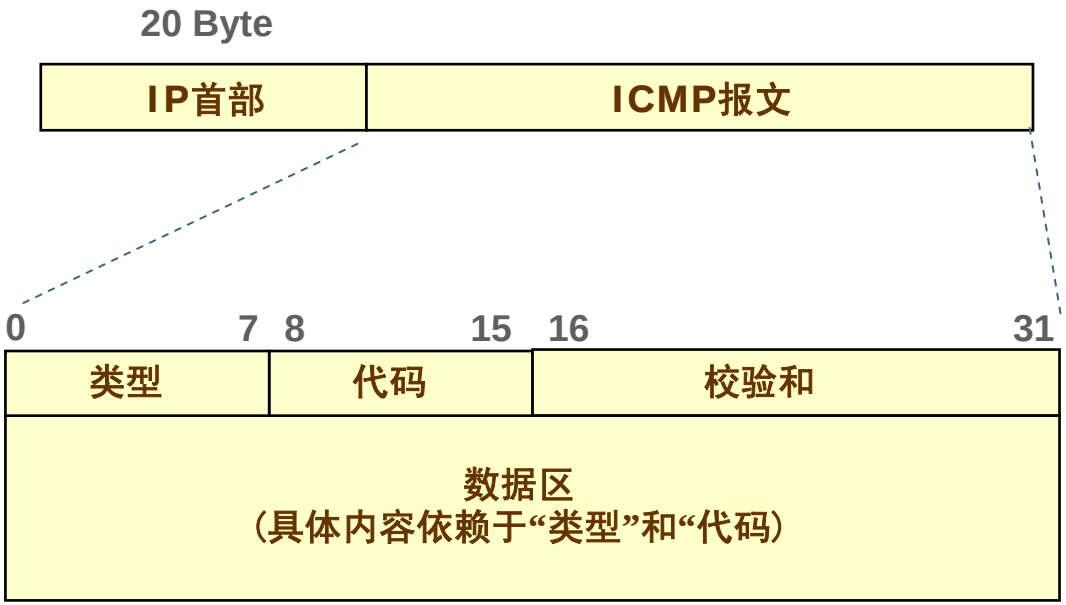
- **ICMP**

  （Internet Control Message Protocol，互联网控制消息协议）建立在IP之上，用来报告数据报传递处理过程中的相关错误，并提供一些网络管理及状态信息。

- **两种引发ICMP报文的因素：**
  - 各种ICMP请求引发的ICMP应答。
  - 其他协议或网络自身问题引发的出错信息。
- **ICMP协议的一个原则**
  - 从来不会为了响应一个ICMP出错消息而产生另一个ICMP出错报文。



3.2.2 ICMP协议的数据报头格式



几种主要的ICMP消息类型

类型	代码	描述
0	0	Echo应答。
3		目的不可达。
	0	网络不可达。
	1	主机不可达。
	2	协议不可达。
	3	端口不可达。
	4	需要分段，但DF置位。
	5	源路由失败。
	13	通信被过滤禁止。

类型	代码	描述
5		路由重定向。
	0	网络重定向。
	1	主机重定向。
	2	服务类型及网络重定向。
	3	服务类型及主机重定向。
8	0	Echo请求。
11		超时。
	0	TTL为0（Traceroute）。
	1	重组期间TTL为0。

3.2.3 ICMP协议的实用工具分析

- **Ping**
  - 技术原理
    - 向需要探测的目标主机发送“Echo Request”类型的ICMP数据包，等待“Echo Reply”类型的ICMP数据包。
  - Ping命令的使用

用法：ping [选项] <目标地址列表>

```
# ping -c 4 www.yahoo.com
PING www.yahoo.akadns.net (66.94.230.34) 56(84) bytes of data.
64 bytes from p3.www.scd.yahoo.com (66.94.230.34): icmp_seq=1 ttl=53 time=191 ms
64 bytes from p3.www.scd.yahoo.com (66.94.230.34): icmp_seq=2 ttl=53 time=191 ms
64 bytes from p3.www.scd.yahoo.com (66.94.230.34): icmp_seq=3 ttl=53 time=191 ms
64 bytes from p3.www.scd.yahoo.com (66.94.230.34): icmp_seq=4 ttl=53 time=190 ms

--- www.yahoo.akadns.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3038ms
rtt min/avg/max/mdev = 190.794/191.195/191.416/0.589 ms
```

常用的Ping命令选项

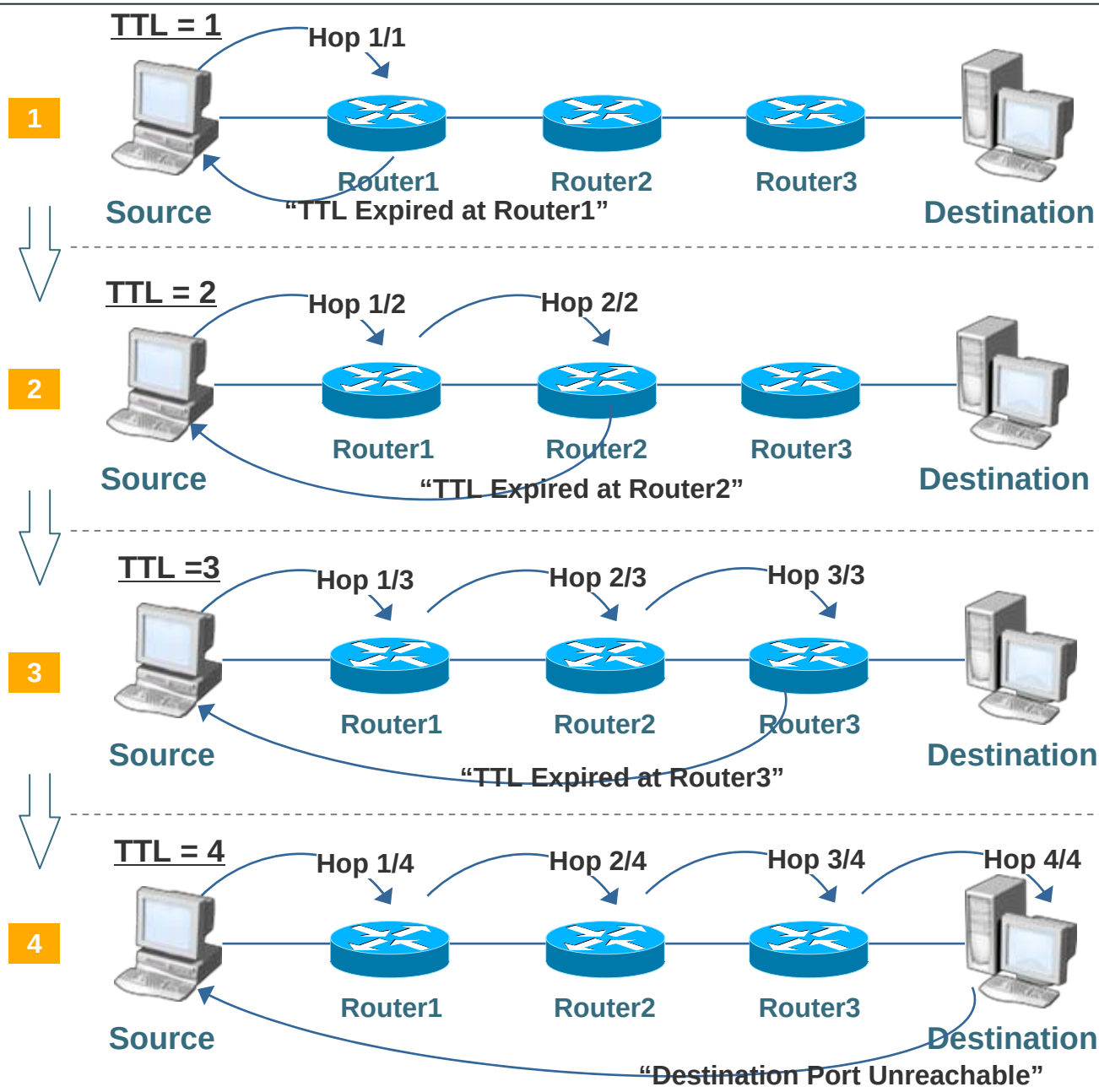
Windows系统中的选项	含义
-n <count>	设置发送Echo Request数据包的个数。
-a	解析IP地址对应的主机名称。
-i <ttl>	设置数据包的网络生存时间（Time to Live）。
-l <size>	设置ICMP数据包的报文长度，默认为24个字节。
-w <timeout>	设置等待回复的超时时间。
-t	一直不停地发送查询数据包，除非用户键入Ctrl+C中断。

Unix系统中的选项	含义
-c <count>	设置发送Echo Request数据包的个数。
-m <ttl>	设置数据包的网络生存时间（Time to Live）。
-n	不解析IP地址对应的主机名称。
-s <size>	设置ICMP数据包的报文长度，默认为56个字节。
-w <timeout>	设置等待回复的超时时间。
-f	以尽可能快的速度发送查询数据包。
-i <wait>	设置在两次发送查询数据包之间等待的时间（默认为1秒）。

Tracert / Traceroute

- traceroute的技术原理
  - 反复向目标主机某个不知名的端口发送UDP探测包，包的TTL值递增1（初始值为1）。
  - 所有探测包经过的中间路由器都将依次返回“TTL超时”的ICMP信息。
  - 探测包到达目的主机后将返回“端口不到达”的ICMP消息。
- tracert的原理
  - 发送ICMP echo request探测包，TTL值递增1（初始值为1）。
  - 中间路由器返回的是“TTL超时”的ICMP消息。
  - 目标主机最终返回的是ICMP echo reply消息。

路由探测技术原理


TTL = 1
Hop 1/1
1
Router1
Router2
Router3
Source
"TTL Expired at Router1"
Destination
TTL = 2
Hop 1/2
Hop 2/2
2
Router1
Router2
Router3
Source
"TTL Expired at Router2"
Destination
TTL =3
Hop 1/3
Hop 2/3
Hop 3/3
3
Router1
Router2
Router3
Source
"TTL Expired at Router3"
Destination
TTL = 4
Hop 1/4
Hop 2/4
Hop 3/4
Hop 4/4
4
Router1
Router2
Router3
Source
Destination
"Destination Port Unreachable"

– tracert（Windows）命令的使用

用法：tracert [选项] <目标地址列表>

>tracert www.yahoo.com.cn

Tracing route to web.search.vip.cnb.yahoo.com [202.43.217.78]
over a maximum of 30 hops:

1	<10 ms	<10 ms	<10 ms	USER-P9JO4LM57T [172.16.15.91]
2	14 ms	22 ms	17 ms	218.1.60.196
3	11 ms	11 ms	12 ms	218.1.62.97
4	12 ms	11 ms	11 ms	218.1.3.9
5	12 ms	12 ms	11 ms	218.1.0.202
6	11 ms	12 ms	12 ms	202.101.63.161
7	12 ms	14 ms	16 ms	202.101.63.234
8	36 ms	34 ms	35 ms	202.97.34.45
9	37 ms	36 ms	36 ms	202.97.57.214
10	35 ms	34 ms	34 ms	219.142.8.230
11	35 ms	35 ms	35 ms	po2.bas1.cnb.yahoo.com [202.165.96.198]
12	36 ms	38 ms	35 ms	web.search.vip.cnb.yahoo.com [202.43.217.78]

Trace complete.

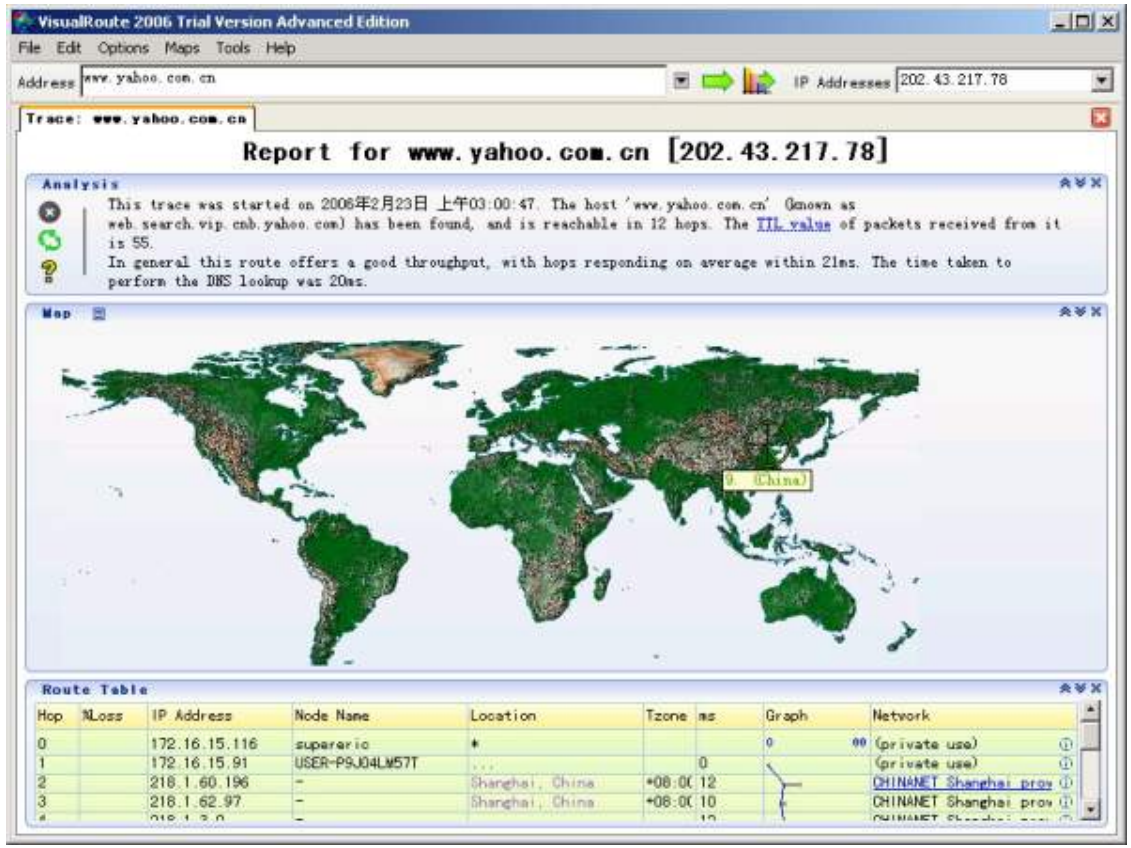
– traceroute（Unix）命令的使用

常用的路由探测选项

tracert选项	含义
-d	不解析主机名称。
-h <hops>	设置探测的最大跳（hop）数。
-w <timeout>	设置等待回复的超时时间。

traceroute选项	含义
-n	不解析主机名称。
-m <hops>	设置探测的最大跳（hop）数。
-w <timeout>	设置等待回复的超时时间。
-p	设置UDP探测端口号。
-I	使用ICMP数据包进行探测。

- 图形化的路由查询工具 — VisualRoute
  - 集成了Ping、whois与traceroute的功能。
  - 自动分析网络连接结果并呈现在世界地图上。

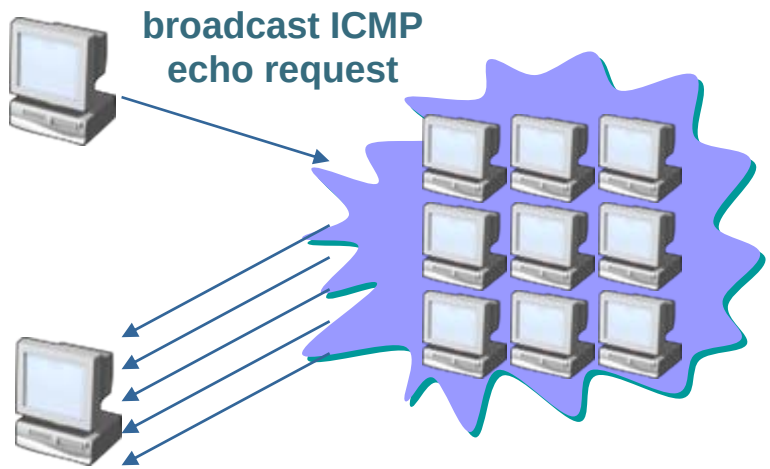


3.2.4 ICMP协议安全性分析

- 利用ICMP Echo Request及IP欺骗进行Smurf攻击。
- 利用ICMP路由重定向消息进行攻击。
- 利用ICMP目的不可达消息进行攻击。

Smurf ICMP Flood攻击

- 攻击方式
  - 向网络广播地址发送ICMP ECHO REQUEST数据包，并将其源地址伪造为目标主机地址。
  - 网络上的所有计算机都响应该请求，从而淹没目标主机。



- 防御对策
  - 忽略ICMP ECHO广播数据包

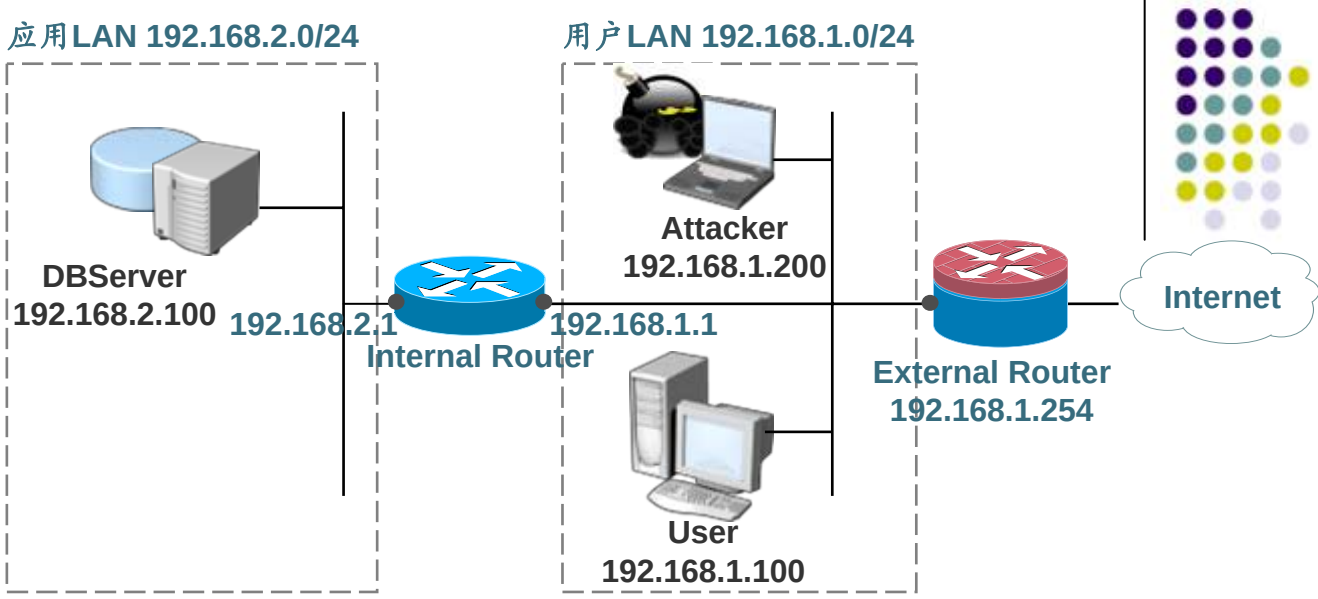
```
# echo 1 > /proc/sys/net/ipv4/icmp_ignore_broadcasts
```

  - 忽略所有ICMP ECHO请求数据包

```
# echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

ICMP重定向攻击

- 攻击方式
  - 在有多于一个路由器的网络中，经常需要通过ICMP重定向来正确的选择路由器。
  - 攻击者可以通过假冒路由器发送ICMP重定向包给目标机，使之先将数据包流向攻击者的主机。



1. 当User访问DBServer时，假设其一开始选择的默认路由器为External Router，则External Router会发送一条ICMP重定向消息告诉User应该选择Internal Router。
2. Attacker可假冒External Router向User发送ICMP重定向包，那么最终的数据包流向为：
   User → Attacker → Internal Router → DBserver

- 防御对策：拒绝ICMP重定向；根据网络拓扑手工添加路由信息。
  - 大部分的Linux系统默认都关闭ICMP重定向功能。
  - 如果开启了，则可作如下设置：

```
# echo 0 > /proc/sys/net/ipv4/ipv4/conf/all/accept_redirects
```

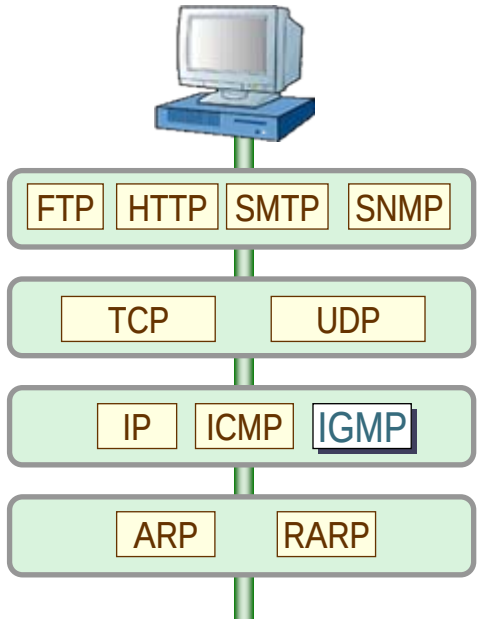
3.3 IGMP协议及安全性

- IGMP协议的基本概念
- IGMP协议的数据报头格式
- IGMP协议的安全性分析

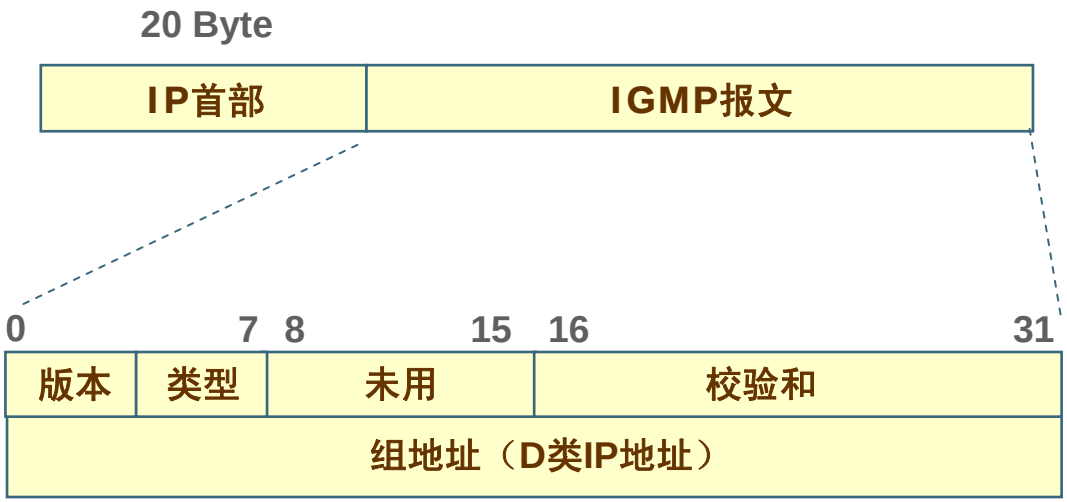
RFC 1112

3.3.1 IGMP协议的基本概念

- IGMP协议是用于进行多播成员组管理的一种注册协议。
  - 通过IGMP协议，主机与直接连接的路由器进行通信，路由器可以定时询问本网络主机是否属于某个多播组，主机可以通知路由器它希望加入或者离开哪个多播组。



3.3.2 IGMP协议的数据报头格式

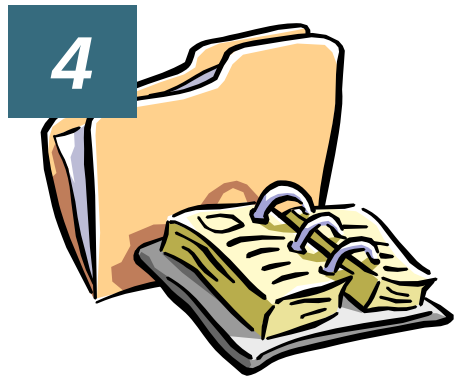


3.3.3 IGMP协议安全性分析

- **IGMP Flood**
  - 攻击方式
    - 不使用多播的D类IP地址，而是将目的IP地址设置为被攻击目标的IP地址。
    - 许多操作系统并不能很好地处理这样的IGMP报文，包括Windows 95/98/NT，一旦遭受攻击，目标主机的反应会明显变慢。
  - 防御对策
    - Windows 2000/XP/Server 2003不受此类攻击影响。

主题 4

- 协议和参考模型
- 链路层协议及安全性分析
- 网络层协议及安全性分析
- 传输层协议及安全性分析
  - UDP协议及安全性
  - TCP协议及安全性
- 应用层协议及安全性分析



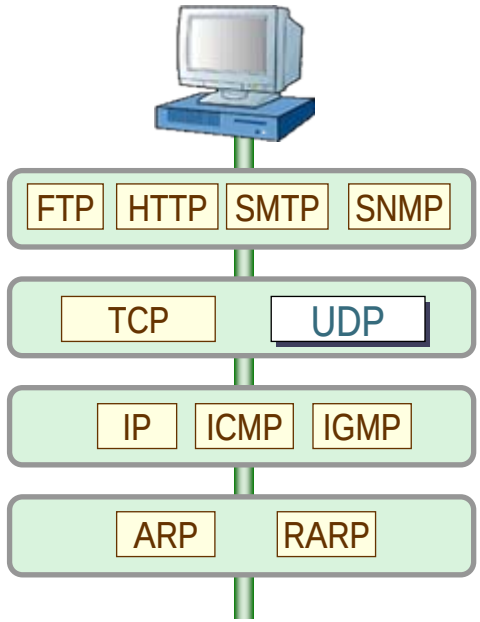
4.1 UDP协议及安全性

- UDP协议的基本概念
- UDP协议的数据报头格式
- UDP协议的安全性分析

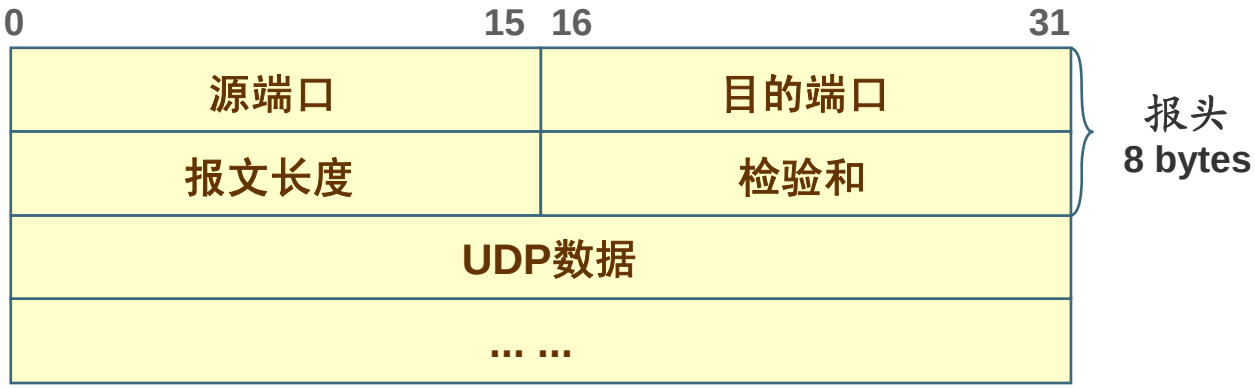
RFC 768

4.1.1 UDP协议的基本概念

- UDP是一个简单的面向数据报的传输层协议，进程的每个输入输出操作刚好产生一个UDP数据报，该数据报导致一个IP数据报的发送，这有别于面向流的协议（TCP）。
- UDP不提供可靠性，但比TCP更有效，常用于多媒体应用及基于请求/应答方式的网络服务（DNS、NIS、NFS）。



4.1.2 UDP协议的数据包报头格式



与UDP相关的一些“小型”服务

服务名称	端口号	描述
Echo	7/udp 7/tcp	服务器原样返回客户端发送来的数据。
Discard	9/udp 9/tcp	服务器丢弃客户端发送的数据。
Daytime	13/udp 13/tcp	服务器以可读的格式返回时间及日期。
Chargen	19/udp 19/tcp	服务器返回一个包含ASCII字符串的报文。 服务器持续发送字符流，直到客户端中断连接。
time	37/udp 37/tcp	服务器以32位二进制格式返回时间。

4.1.3 UDP协议安全性分析

- **UDP Flood攻击**
  - 攻击方式
    - 使用伪造的源地址向ECHO（7/udp）、CHARGEN（19/udp）这样的UDP服务发送畸形数据包（过量大小、错误分片等），从而造成系统内核崩溃或者资源耗尽。
  - 防御对策：不启动无用的UDP小型服务
    - 在inetd.conf中注释掉小型UDP服务。

```
# perl -i.bak -ne ‘print unless /\binternel\b/’ /etc/inetd.conf
# killall -HUP inetd
```

    - 多数xinetd默认不安装这些服务。

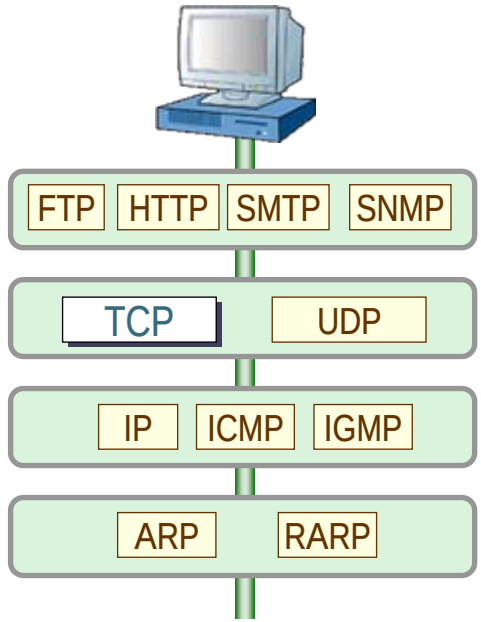
4.2 TCP协议及安全性

- TCP协议的基本概念
- TCP协议的数据报头格式
- TCP连接与断开的协议过程
- TCP协议的安全性分析

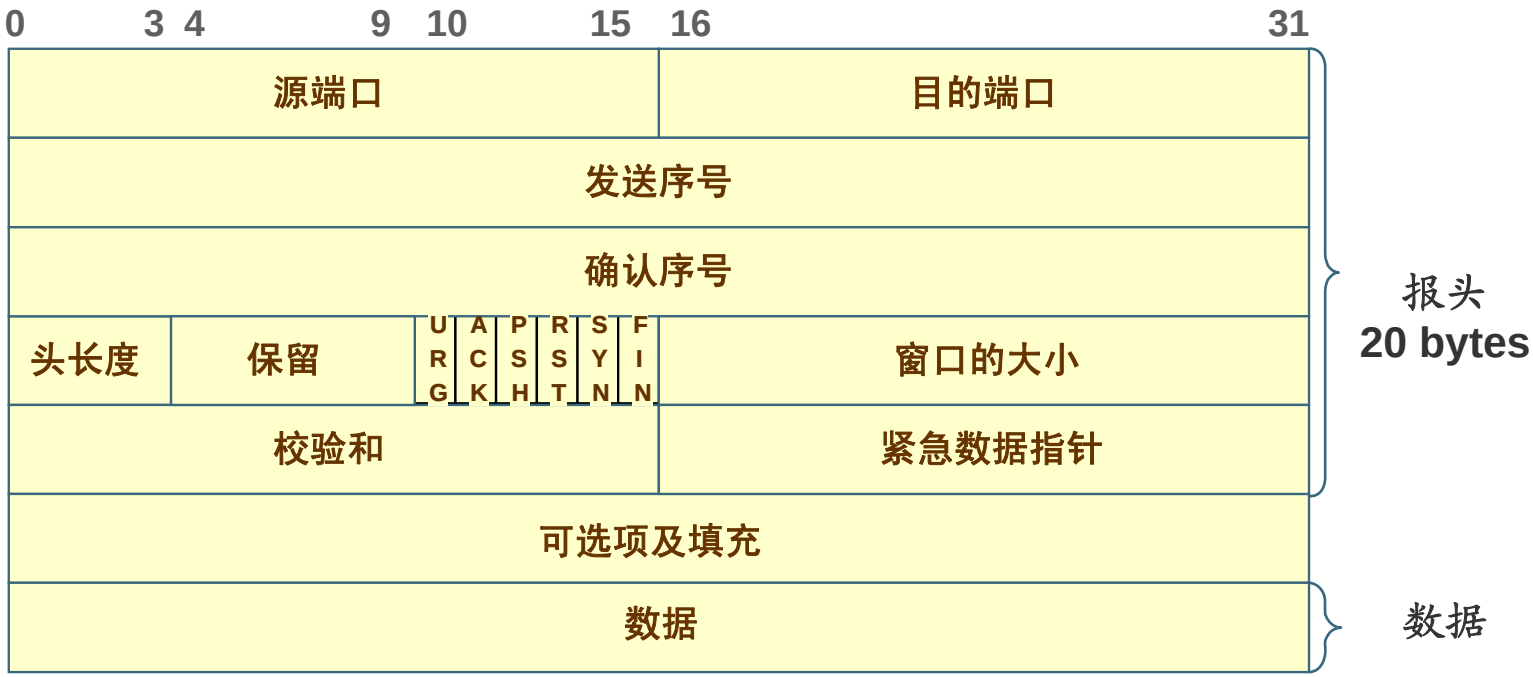
RFC 793

4.2.1 TCP协议的基本概念

- **TCP提供面向连接的、可靠的数据流服务。**
  - 数据流：数据是以连续的字节流方式传输的，而不是基本的数据块。TCP负责数据的分段，而与应用无关。
  - 可靠性：每个传输的字节都有一个序列号，并期望从接收方得到肯定的确认（ACK）。
  - 流量控制：用来指明接受缓冲区的大小。
  - 多路复用：用端口实现（同UDP协议）。
  - 逻辑连接：由发送进程与接受进程所用的套接字对进行标识，一个套接字（socket）由一个IP地址和一个端口号所组成。
  - 全双工：TCP提供双向并发数据。



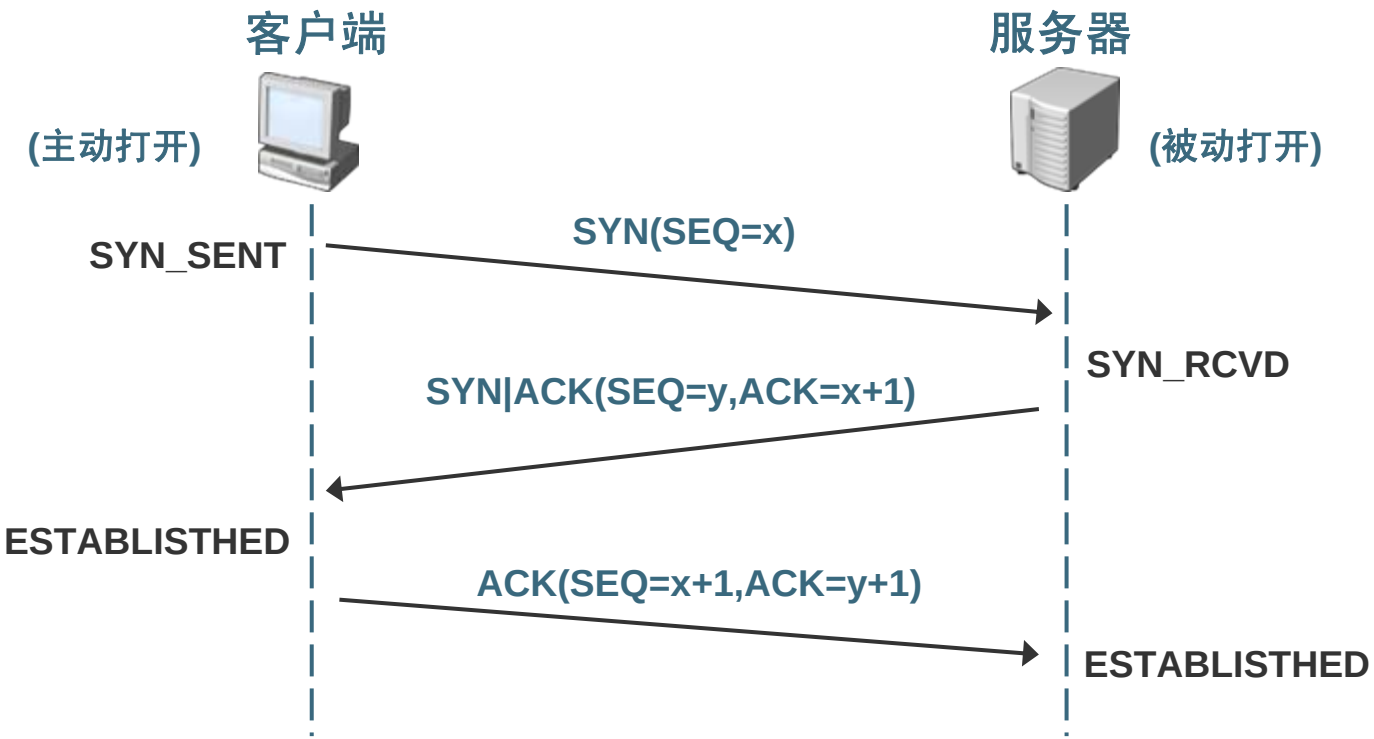
4.2.2 TCP协议的数据包报头格式



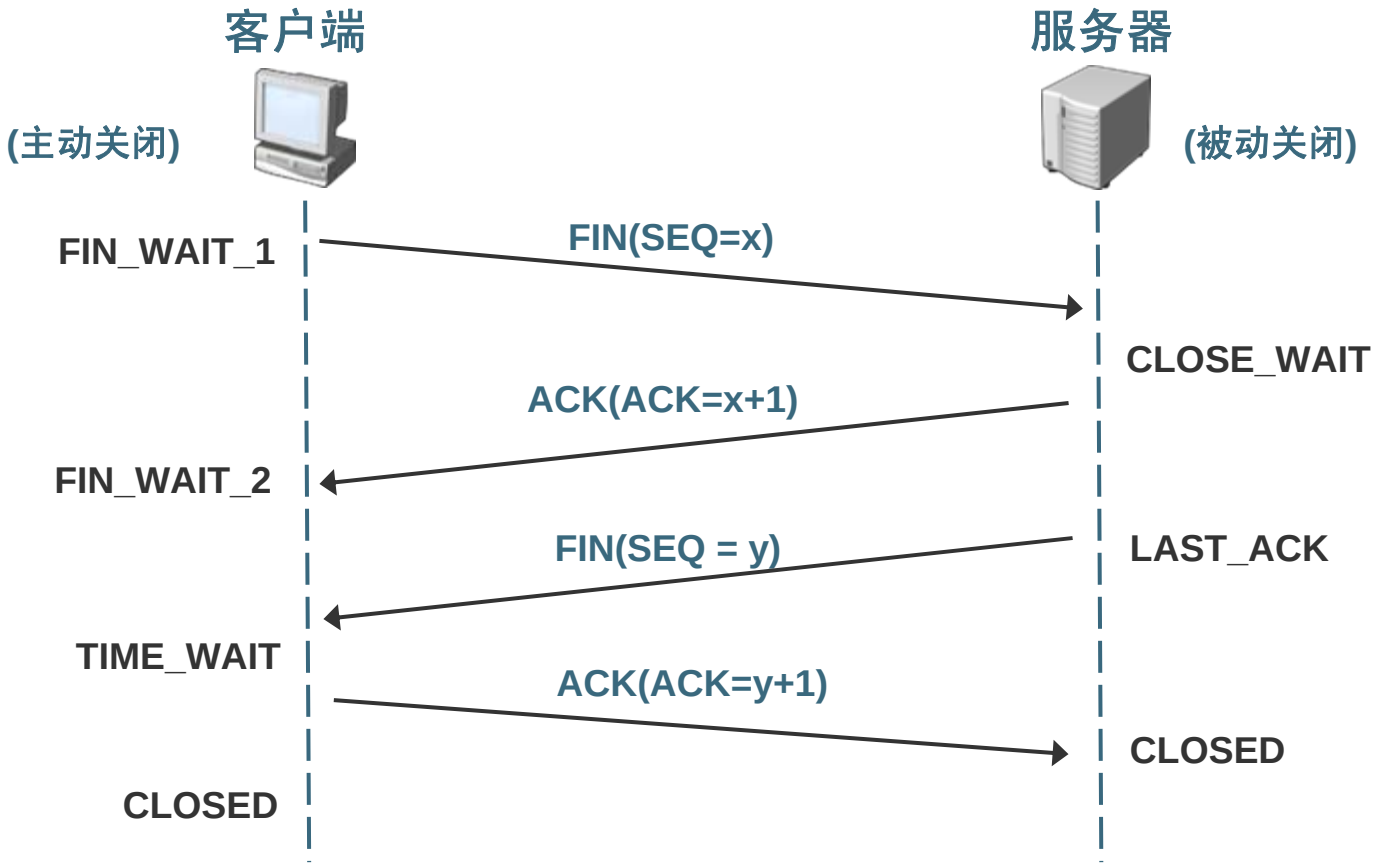
TCP协议报文头部的6个标志位

- URG
  - 紧急指针有效（发送带外数据）。
- ACK
  - 确认号有效。
- PSH
  - 要求接收方不要缓存数据，而是直接传给上层应用。
- RST
  - 本次连接被复位。
- SYN
  - 连接请求放置初始序列号，初始化一次连接。
- FIN
  - 发送方欲结束发送数据。

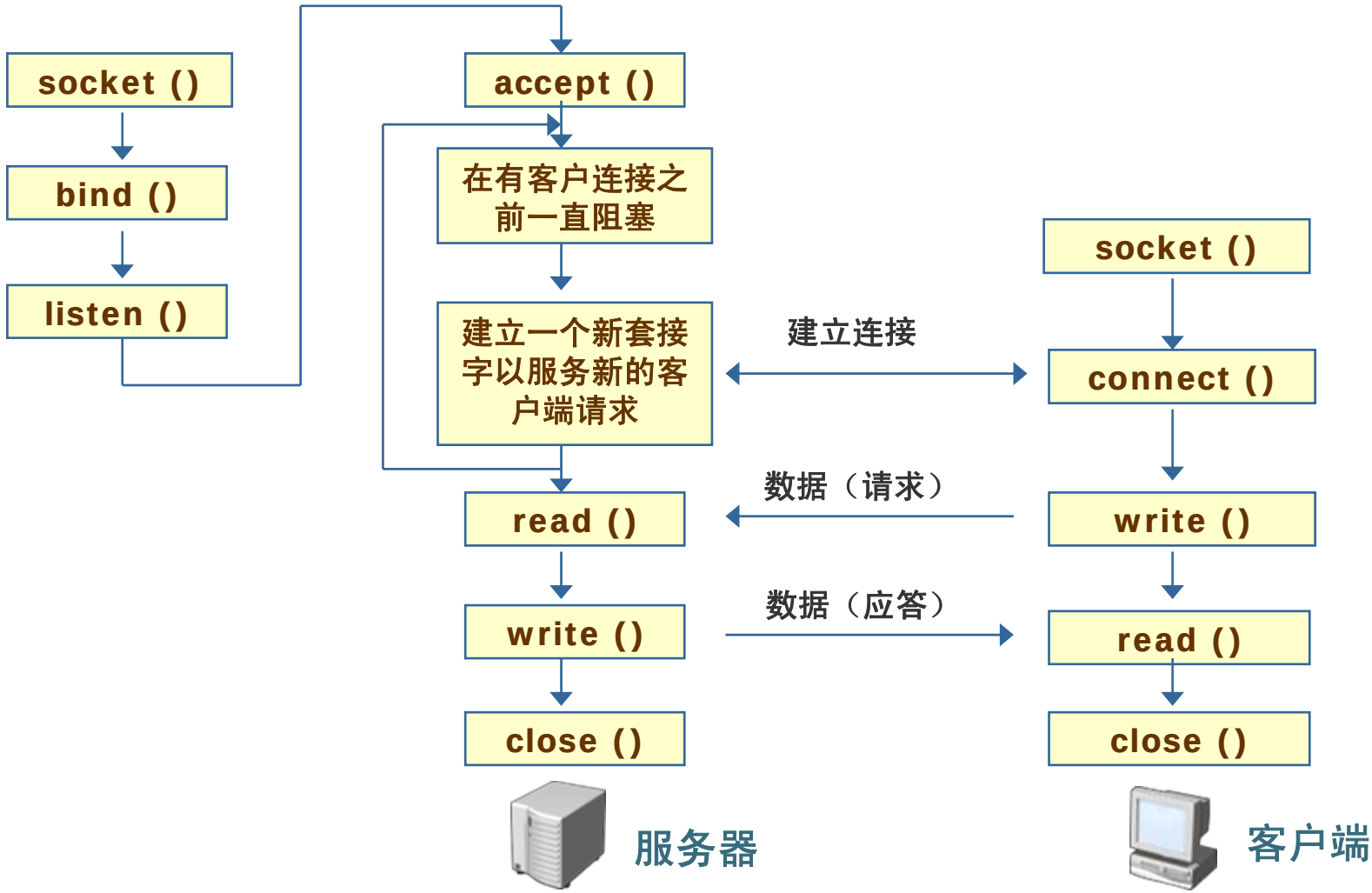
4.2.3 建立TCP连接的三次握手过程



断开TCP连接的过程


客户端
服务器
(主动关闭)
(被动关闭)
FIN_WAIT_1
FIN(SEQ=x)
CLOSE_WAIT
ACK(ACK=x+1)
FIN_WAIT_2
FIN(SEQ = y)
LAST_ACK
TIME_WAIT
ACK(ACK=y+1)
CLOSED
CLOSED

基于TCP协议的套接字系统调用

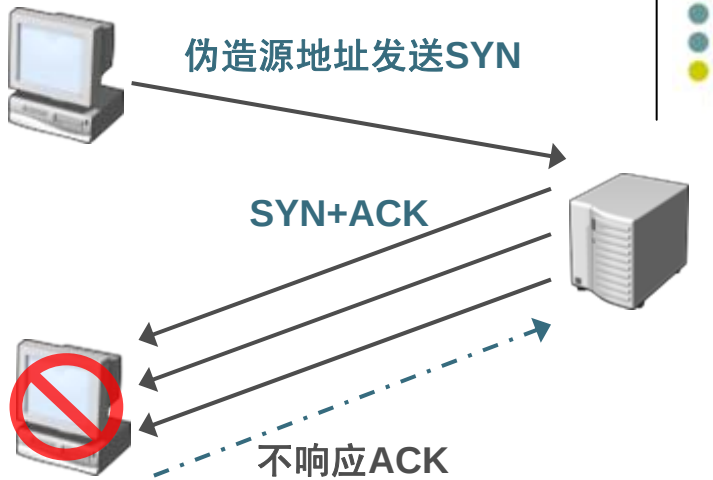


4.2.4 TCP协议的安全性分析

- 拒绝服务攻击
  - SYN Flooding：利用TCP连接机制中存在的问题和IP地址欺骗实施攻击。
  - OOB：利用TCP带外数据实施攻击。
  - Land。
- TCP欺骗攻击
  - 利用TCP序列号猜测、弱的访问控制机制、IP地址欺骗实施的攻击。
- TCP会话劫持攻击
- TCP端口扫描

TCP SYN Flood攻击

- 攻击方式
  - 使用虚假的源地址向服务器发送巨大数量的TCP SYN（请求初始连接）数据包但不响应服务器的SYN|ACK。
  - 服务器不断在等待和重试虚假客户端的确认，从而耗尽系统资源。



- 防御对策
  - 调整内核TCP SYN参数：减小等待超时值；增大队列的最大容量。

```
# echo 100 > /proc/sys/net/ipv4/vs/timeout_synack
# echo 10 > /proc/sys/net/ipv4/vs/timeout_synrecv
# echo 128 > /proc/sys/net/ipv4/tcp_max_syn_backlog
```

  - 启用SYN Cookies机制：创建与客户端IP地址、连接端口、时间戳和本地计数相关的TCP序号发回给客户端，只有当系统接受到客户端ACK包之后验证（SYN接收队列满之后才检查）其Cookie值正确，才允许建立连接。

```
# echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

主题 5

- 协议和参考模型
- 链路层协议及安全性分析
- 网络层协议及安全性分析
- 传输层协议及安全性分析
- 应用层协议及安全性分析

  - FTP协议及安全性
  - HTTP协议及安全性
  - SMTP协议及安全性
  - SNMP协议及安全性
  - DNS协议及安全性

5.1 FTP协议及安全性

- FTP协议的基本概念
- FTP协议的安全性分析

RFC 959

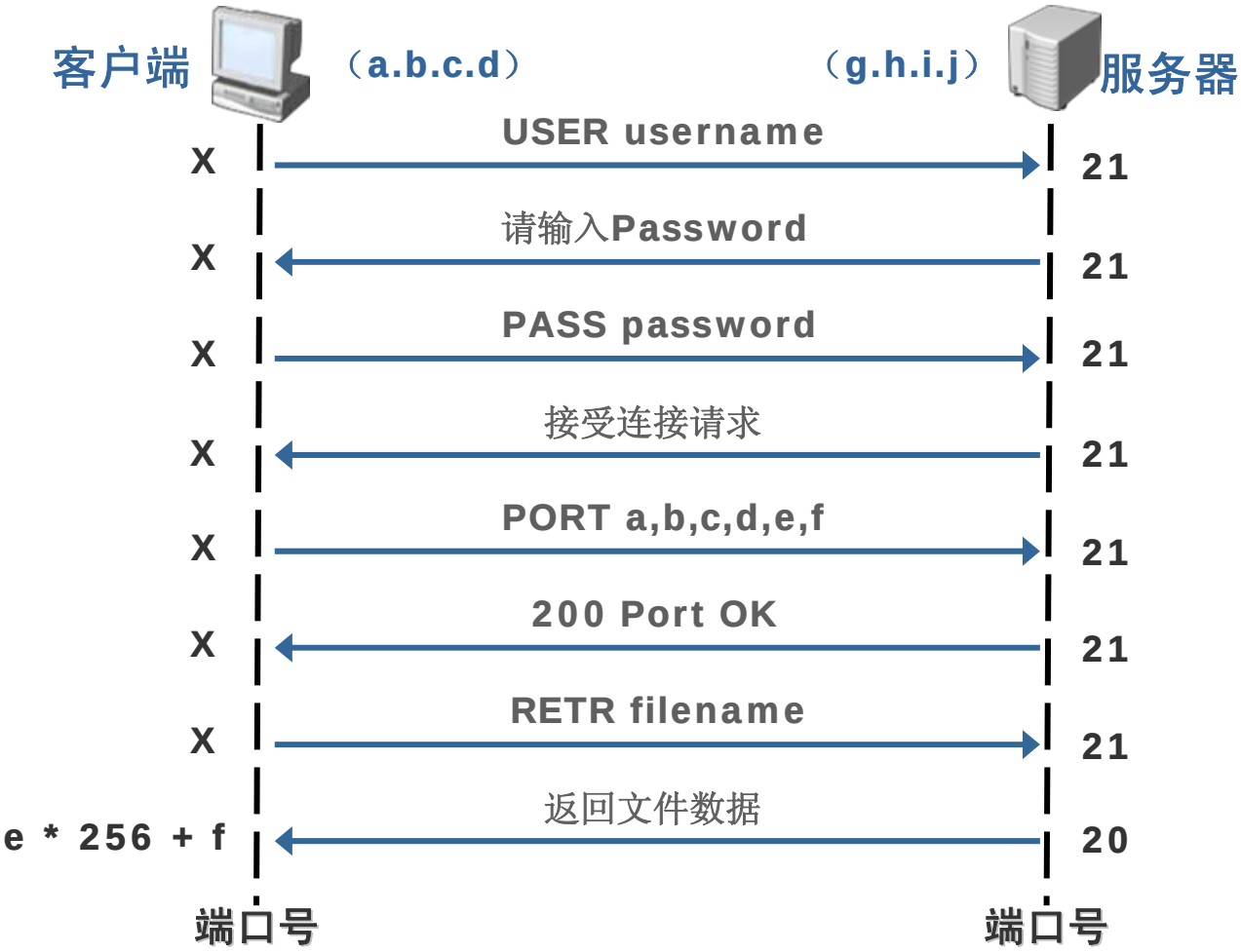
5.1.1 FTP协议的基本概念

- 客户端与服务器端建立两条独立的连接
  - 控制连接（通道）：传递控制命令（端口21/TCP）。
  - 数据连接（通道）：传递实际的数据，例如文件、目录内容。
- 控制连接在会话期间一直保持，数据连接只针对具体的数据传输任务，是暂时的，其建立方式有两种：
  - 主动模式（PORT）：客户端用PORT命令告诉服务器其监听端口，服务器与之建立连接（使用端口20/TCP）。
  - 被动模式（PASV）：服务器打开某个监听端口，客户端与之建立连接。

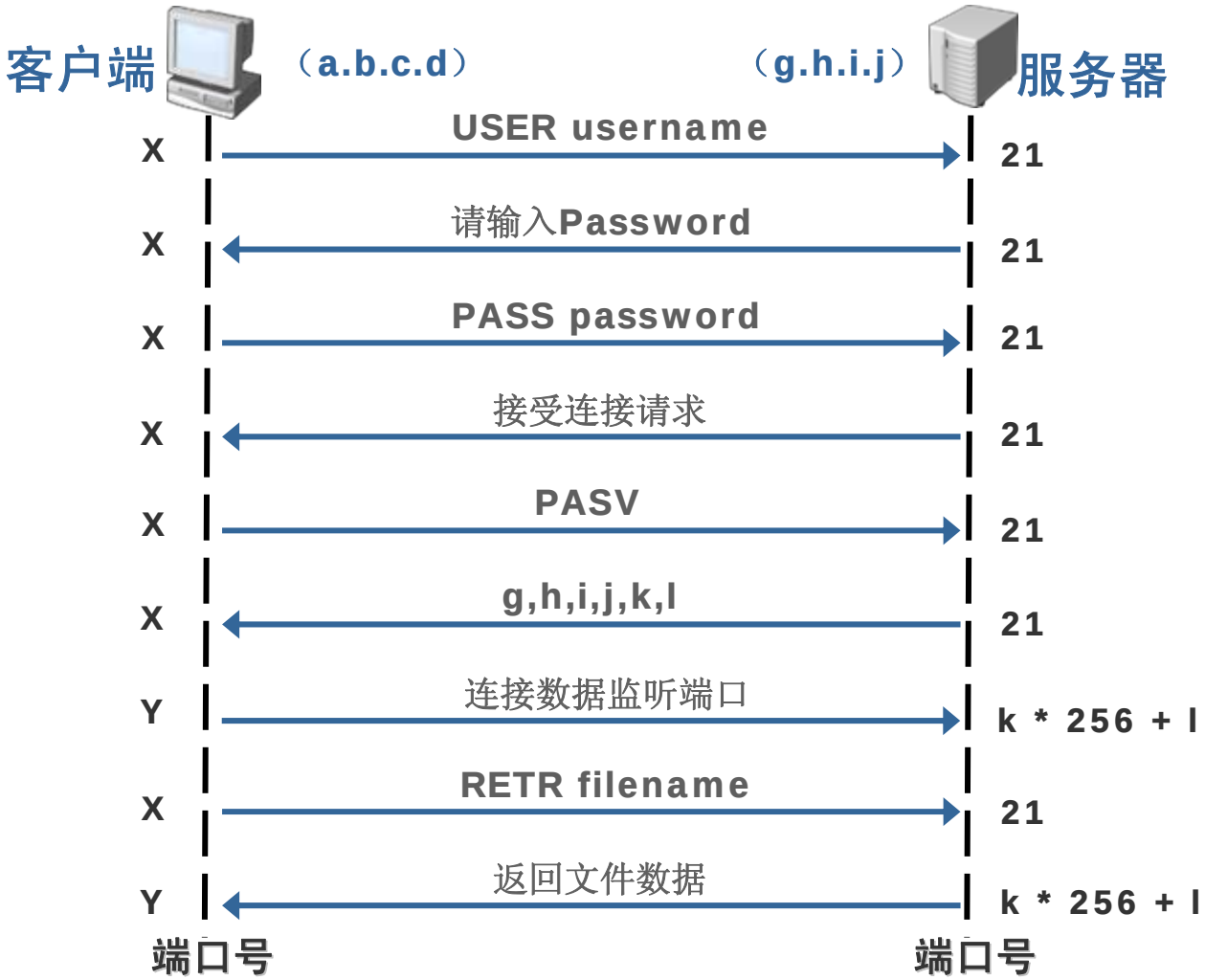
注

通常，如果客户端所在的防火墙禁止外来的连接请求时，FTP客户端可采用被动模式与服务器端交互。

FTP协议的主动模式通信机制



FTP协议的被动模式通信机制



5.1.2 FTP协议的安全性分析

- **FTP服务器的旗标（Banner）信息泄露**
  - 安全性分析
    - 当客户端建立和FTP服务器的连接之后，服务器立即向客户发送banner信息，包括FTP服务器名、软件版本号、当前时间等信息。

```
220 ftp.infosec.net FTP Server (Version wu-2.6.0(1) Sat Feb 19 23:37:43 EST 2005) ready.
```

  - 防御对策：更改FTP服务的banner
    - WU-FTPD：在/etc/ftpaccess中设置如下多个配置选项：

选项	含义
greeting full	提供完整的欢迎信息
greeting brief	只显示主机名
greeting terse	只输出“FTP server ready”
greeting text message	准确而不加修饰地输出欢迎信息

    - ProFTPD：更改/etc/proftpd.conf中的ServerName变量。

```
ServerName “Unauthorized use of this FTP server Prohibited. Go away.”
```

Nmap FTP反射式扫描

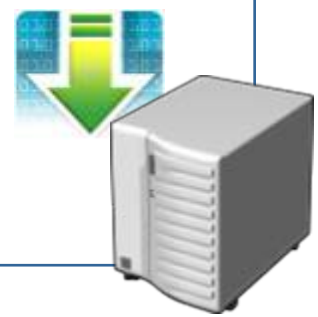
- 安全性分析
  - 当攻击者向FTP服务器建立一个主动模式的数据连接时，可通过PORT命令提供扫描目标的IP地址和端口，使得FTP服务器打开一个到扫描目标的连接来进行端口扫描。
- 防御对策
  - 现在多数FTP服务器都配置成拒绝IP地址与客户端主机不符的PORT命令。

PASV FTP数据劫持

- 防御对策
  - 不使用被动模式的FTP。

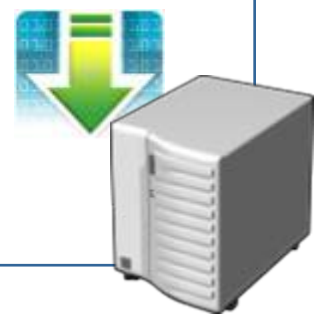
PORT FTP数据劫持

- 防御对策
  - 不使用主动模式的FTP。



FTP协议的明文传输

- 安全性分析
  - 攻击者可以使用嗅探器在网络中分析FTP数据包，从中获取登录FTP站点的用户名和密码。
- 防御对策
  - 若只需要支持匿名下载文件，则根本不使用FTP服务器，而应当运行web服务器。
  - 若需要上传文件，则应当使用scp或sftp来取代FTP，它们都是OpenSSH的组成部分。



5.2 HTTP协议及安全性

- HTTP协议的基本概念
- HTTP协议的安全性分析

[HTTP v1.0版本] ← RFC 1945

[HTTP v1.1版本] ← RFC 2616

5.2.1 HTTP协议的基本概念

- **HTTP协议是基于请求/响应模式的客户端/服务器协议。**
  - 当TCP连接建立之后，客户端（一般通过浏览器或者直接telnet）向服务器端发送一条消息，通常称之为HTTP请求，然后服务器回送所请求的信息。客户端收到信息后，根据其类型进行处理或显示。
  - 可启用Cookie机制来在客户端与服务器端之间交换信息。

请求信息
[method] [target_url] [HTTP/1.n]
Header1: value11, value12 Header2: value21 …… HeaderN: valueN
CRLF（换行符，十六进制值0d0a）
MIME Object（Optional）

响应信息
HTTP 1.n StatusCode Message
Header1: value1 Header2: value2 …… HeaderN: valueN
CRLF（换行符，十六进制值0d0a）
MIME Object（Optional）

HTTP协议客户端请求方法

选项	含义
GET	请求得到指定URL资源的实体对象
HEAD	与GET类似，但不要求服务器在响应消息中附带实体内容，只需要响应表头中包含关于资源的元信息。
POST	向服务器发送请求，要求将请求头后实体的内容当作请求行中URL资源的附加子项（用于表单内容的提交）
OPTIONS	请求可用在对指定URL请求/响应通信时的配置信息
PUT	请求将附带的实体内容存储在指定的请求URL位置上
DELETE	请求删除指定的URL资源

HTTP协议响应消息状态码

状态码分类	状态码编号	说明
信息	100 ~ 199	应用程序特定的消息
成功	200 ~ 299	已成功处理的请求
重定向	300 ~ 399	客户端需要进一步操作以处理请求
客户端错误	400 ~ 499	客户端出现错误
服务器错误	500 ~ 599	服务器端出现错误

HTTP协议中参数提交方式一：GET方法

- 在HTTP请求行中包含传送的参数字符串
  - 例如在Google中搜索字符串“test”，浏览器的URL是：

```
http://www.google.com/search?hl=zh-CN&newwindow=1&q=test
&btnG=%E6%90%9C%E7%B4%A2&lr=
```

  - 真正的HTTP消息请求行应该是：

```
GET /search?hl=zh-CN&newwindow=1&q=test
&btnG=%E6%90%9C%E7%B4%A2&lr= HTTP/1.1
```

参数字串由若干域组成，每个域有形如“名字=值”这样的构成，不同域之间用“&”分隔，“%”引起一个双字节的十六进制转义字符，字符串中间若有空格，则用“+”来代替。

请求信息中没有消息实体。

HTTP协议参数提交方式二：POST方法

- 在HTTP请求消息的实体正文中包含传送的字符串参数。

```
POST /search HTTP/1.1
Host: www.google.com
Accept: image/gif, */*
Accept-Language: zh-cn
Content-Type: application / x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Content-Length: 57
Connection: Keep-Alive
                                        ← 这里必须有一个空行
hl=zh-CN&newwindow=1&q=test&btnG=%E6%90%9C%E7%B4%A2&lr=
```

HTTP协议传递的内容

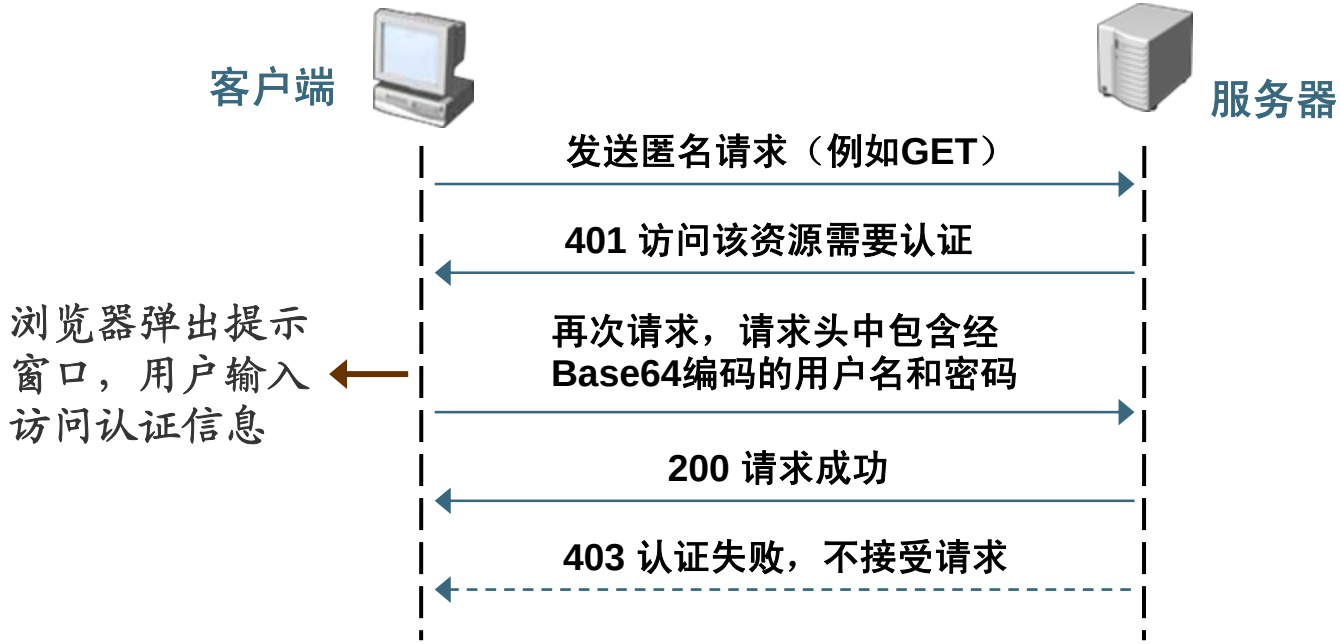
- 静态内容：即静态网页，不依赖于客户端的任何输入：
  - HTML（Hypertext Markup Language）
  - XML（Extensible Markup Language）
- 客户端动态内容
  - ActiveX
  - Java Applet
  - VBScript / JavaScript
- 服务器端动态内容
  - CGI：基于客户端的输入产生动态响应，可以使用多种编程语言。
  - 服务器专用API，例如Netscape Server API（NSAPI）和Microsoft Internet Information Server API（ISAPI）。
  - Servlet，在服务器端执行的Java Applet。
  - 服务器端包含（Server-side include，SSI），.shtml扩展名。
  - Java Server Page（JSP），.jsp扩展名。
  - Active Server Page（ASP），.asp扩展名。

HTTP的认证机制

用户名和密码是在请求消息头部的Authorization域中提交的，用户名和密码之间用冒号分开，并用Base64进行编码：

```
Authorization: Basic QWxhZGRpbjpvcGVuIHN1c2FtZQ=
```

这种认证方式极不安全，通过嗅探获得上述认证信息，然后用简单的Base64解码即可以获得敏感信息。



5.3 SMTP协议及安全性

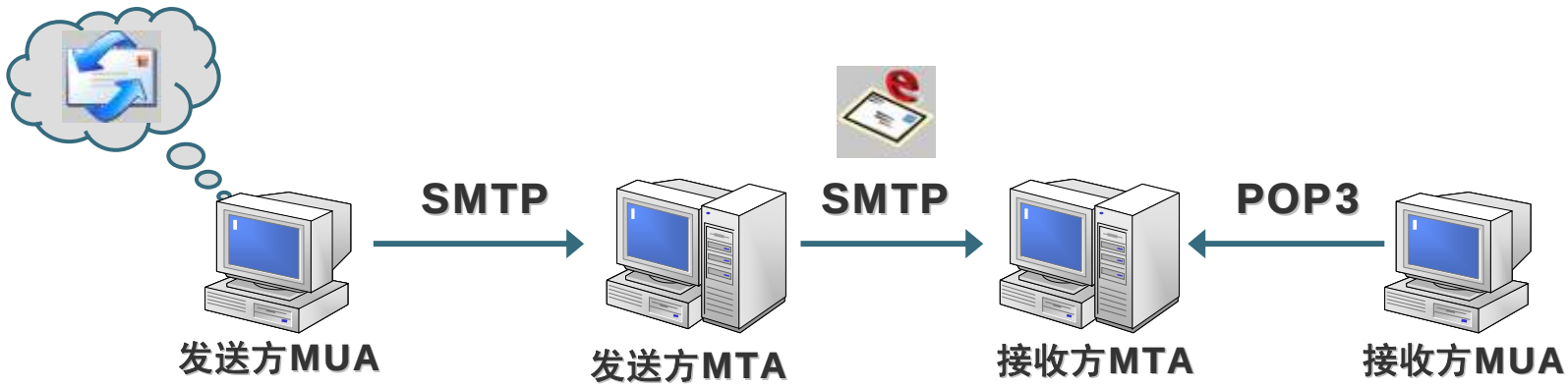
- SMTP协议的基本概念
- SMTP协议的安全性分析

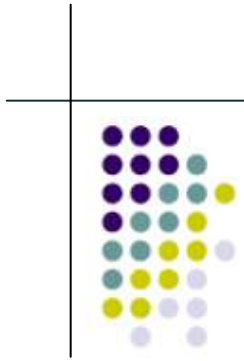
RFC 821

RFC 822

5.3.1 SMTP协议的基本概念

- 邮件传输过程中有两个角色：
  - MUA：邮件用户代理，即直接被用户使用的接口程序，通常称之为邮件客户端软件。
  - MTA：邮件传输代理；通常称之为邮件服务器。
    - 发送邮件的标准是（E）SMTP -（Extended）Simple Mail Transfer Protocol，即（扩展）简单邮件传输协议。
- 邮件传输过程中，通常涉及到4个实体：
  - 发送方MUA，发送方指定的MTA，接收方MTA以及接收方的MUA





220 mx.sjtu.edu.cn ESMTP Postfix

EHLO/HELO supereric

250-mx.sjtu.edu.cn

AUTH LOGIN

334 VXNlcm5hbWU6
(Base64解码后为 Username:)

XXXXXXXXXXXX
(经Base64编码后的用户名)

334 UGFzc3dvcmQ6
(Base64解码后为 Password:)

XXXXXXXXXXXX
(经Base64编码后的密码)

235 Authentication successful

MAIL FROM: ericwyj@sjtu.edu.cn

250 Ok

RCPT TO: <ericwyj@263.net>

250 Ok

Data

354 End data with <CR><LF>.<CR><LF>

```
Date: Mon, 20 Dec 2004 17:03:10 +0800
From: "eric" <ericwyj@sjtu.edu.cn>
To: "ericwyj" <ericwyj@263.net>
Subject: Test
X-mailer: Foxmail 5.0 [cn]
Mime-Version: 1.0
Content-Type: text/plain;
          charset="gb2312"
Content-Transfer-Encoding: 7bit
This is A Test!
```

.

250 Ok: queued as 1C898DDEA

QUIT

221 Bye

邮件投递过程

5.3.2 SMTP协议的安全性分析

- 邮件服务器的root权限获取漏洞
  - 安全性分析
    - 邮件服务器的最大问题就是它需要绑定tcp端口25，因此必须以root启动，只要在服务器中发现漏洞，攻击者就可能立即获得root权限。
  - 防御对策：以非root用户ID运行邮件服务器
    - Qmail、Postfix、Exim：默认即包括该特性。
    - Sendmail：设置sendmail.cf文件中的RunAsUser选项。

```
# vi /etc/mail/sendmail.cf
…
O RunAsUser = sendmail:mail
```

邮件服务器的旗标（Banner）信息泄露

- 安全性分析
  - 当客户端建立和邮件服务器的连接之后，MTA立即向客户发送banner信息，包括邮件服务器名、软件版本号、当前时间等信息。

```
220 mail.infosec.net ESMTP Sendmail 8.12.8/8.12.8; Fri, 18 Feb 2005 02:28:12 +0800
```

- 防御对策：更改SMTP服务的banner
  - Sendmail：更改sendmail.cf文件中的SmtpGreetingsMessage。

```
# vi /etc/mail/sendmail.cf
…
O SmtpGreetingMessage = $j  xxxxxxx; $b
```

  - Qmail：修改qmail-smtpd的smtpgreeting值。
  - Postfix：更改mail.cf中的smtpd_banner。
  - Exim：更改/etc/exim.conf中的smtpd_banner。

SMTP VRFY命令

- 安全性分析
  - VRFY命令最初用于帮助机器确定用户名或地址是否合法，现很少使用。
  - 攻击者常用VRFY命令来对用户名实施蛮力攻击，并于之后实施用户名/口令猜测攻击。
  - 垃圾邮件兜售商可用来收集邮件地址。
- 防御对策：关闭VRFY
  - Sendmail：更改sendmail.cf文件中的PrivacyOptions。

```
# vi /etc/mail/sendmail.cf
…
# privacy flags
O PrivaceOptions = authwarnings, novrfy
```

  - Qmail、Postfix、Exim：默认对于任何VRFY请求都会响应无用信息。

SMTP EXPN命令

- 安全性分析
  - EXPN命令扩展所提供的用户名或邮件地址。
  - 与VRFY类似，攻击者可用EXPN命令来猜测用户名和邮件地址并收集一些有用的信息。
  - 垃圾邮件兜售商可用来收集邮件地址。
- 防御对策：关闭EXPN
  - Sendmail：更改sendmail.cf文件中的PrivacyOptions。

```
# vi /etc/mail/sendmail.cf
…
# privacy flags
O PrivaceOptions = authwarnings, noexpn
```

  - Qmail、Postfix、Exim：默认都不支持EXPN命令。

邮件中继（Mail Relay）

- 安全性分析
  - 如果邮件服务器允许向任意接收地址进行邮件中继，则可能导致别有用心之人将该邮件服务器作为发送垃圾邮件的中转站，从而造成恶劣的影响。
- 防御对策：不中继来自非授权域的邮件
  - **Sendmail：Sendmail 8.9及之后的版本默认拒绝邮件中继。若需要中继某些主机的邮件，可设置/etc/mail/access：**

```
# cat /etc/mail/access
…
localhost                    RELAY
internal.infosec.sjtu.edu.cn RELAY
```

  - **Qmail：Qmail 0.91及以上版本默认拒绝邮件中继。**
  - **Postfix、Exim：默认拒绝邮件中继。**
  - **使用SMTP AUTH，一种SMTP的扩展（ESMTP），定义了服务器验证用户的机制，需要用户连接时提供合法的用户名和口令。**

垃圾邮件

- 安全性分析
  - 垃圾邮件会浪费磁盘空间，耗用带宽，毫无意义地占用CPU时间。
- 防御对策：服务器端阻塞垃圾邮件（DNS查询法）
  - 当客户端主机连接时，服务器对其IP地址进行DNS查询，如果已经登记在DNS黑名单（DNSBL，DNS Blackhole List）上，则将拒绝来自于该主机的任何邮件。
  - Sendmail：更改sendmail.mc文件。

版本	Sendmail.mc项目
8.9	FEATURE (rbl, ‘rbl-plus.mail-abuse.org’)
8.10	FEATURE (dnsrbl, ‘bl.spamcop.net’, ‘error message’)
8.11	HACK (‘check_dnsbl’, ‘relay.ordb.org’, ‘’, ‘general’, ‘reason’)

  - Qmail：结合使用Rblsmtpd和Qmail smtpd，重写tcpserver的配置文件。

```
tcpserver <options> smtp /usr/bin/rblsmtpd -b \
    -r “ipwhois.rfc-ignorant.org: Open relay problem” \
    /var/qmail/bin/qmail-smtpd <options>
```

邮件炸弹和其他拒绝服务攻击（DoS）

- 安全性分析
  - 攻击者可对邮件系统的SMTP端口以潮涌方式发送请求或用许多大尺寸信息来填满邮件队列。
  - 过量的连接会使合法邮件无法到达系统，迅速耗尽系统磁盘空间。
- 防御对策：实行资源限制
  - **Sendmail**：设置如下多个选项：

选项	含义
MaxDaemonChildren	限制并发运行的Sendmail进程数
ConnectionRateThrottle	限制每秒并发入连接数量
MaxRcptsPerMessage	限制单个邮件的接受者数量
MaxMessageSize	限制邮件的最大尺寸

  - **Qmail**：默认只允许同时处理20封外发邮件，可修改/var/qmail/control/concurrencyremote中的相应设置。

5.4 SNMP协议及安全性

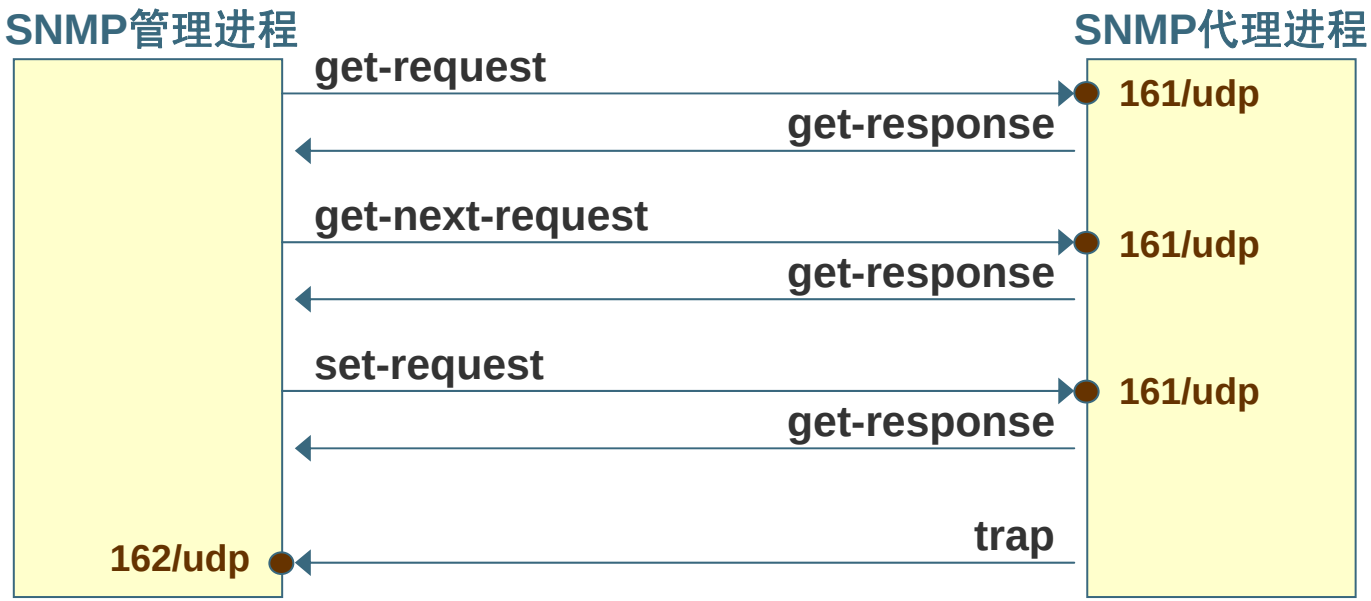
- SNMP协议的基本概念
- SNMP协议的安全性分析

[SNMP v1版本] ← RFC 1157
[SNMP v2版本] ← RFC 1441
[SNMP v3版本] ← RFC 2570

5.4.1 SNMP协议的基本概念

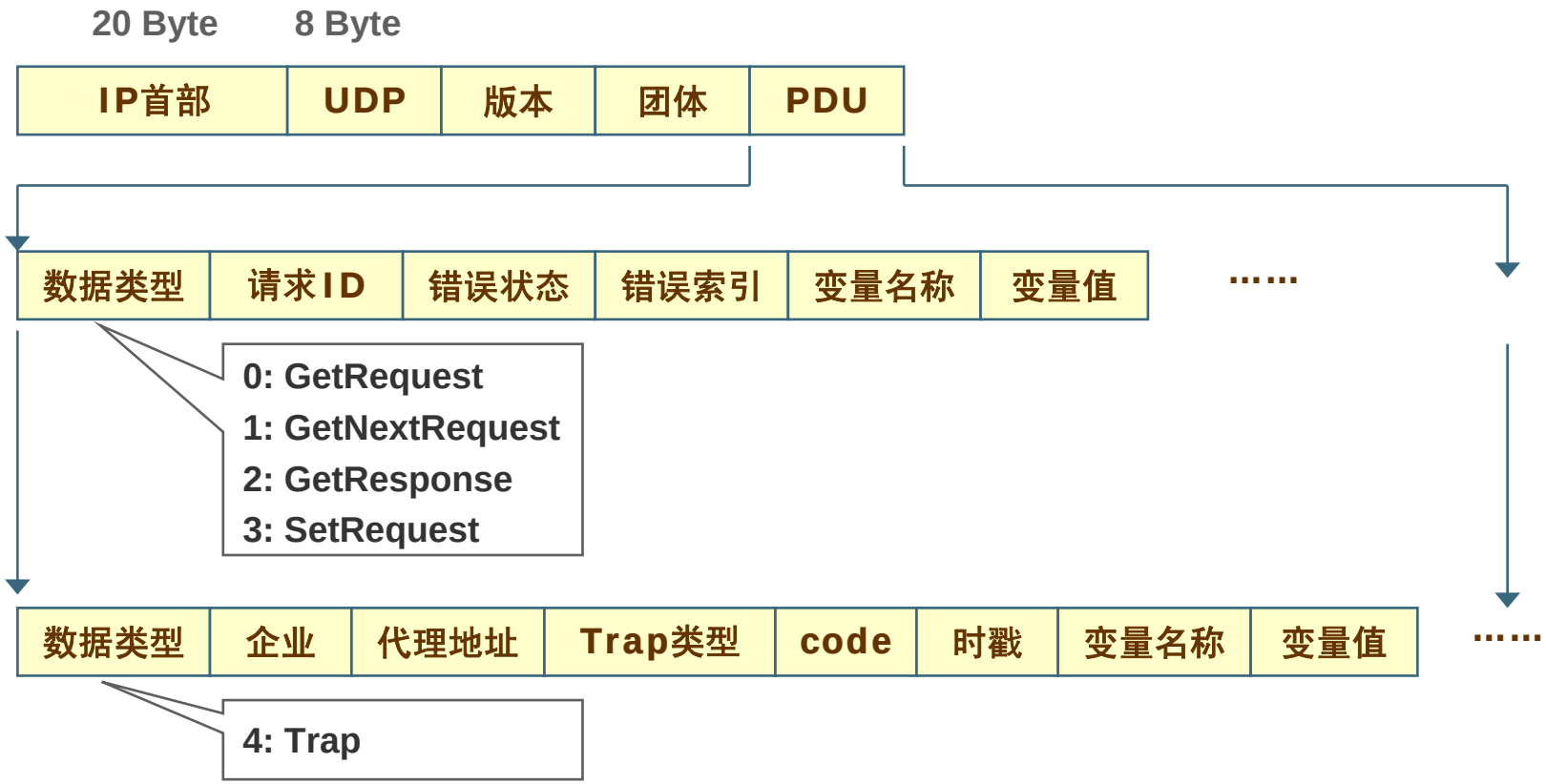
- SNMP协议是管理网络设备的协议。
- 基于SNMP协议的网络管理模型
  - 被管对象：上面运行代理，负责与管理工作站通信，并维护本地的MIB信息，包括路由器、交换机、打印机、服务器等设备。
  - 管理工作站。
  - 管理信息库MIB：包含所有代理进程的所有可被查询和修改的参数。
  - 管理信息结构SMI：关于MIB的一套公用的结构和表示符号。
  - 管理协议：管理进程和代理进程之间的协议。

SNMP协议的通信操作方式



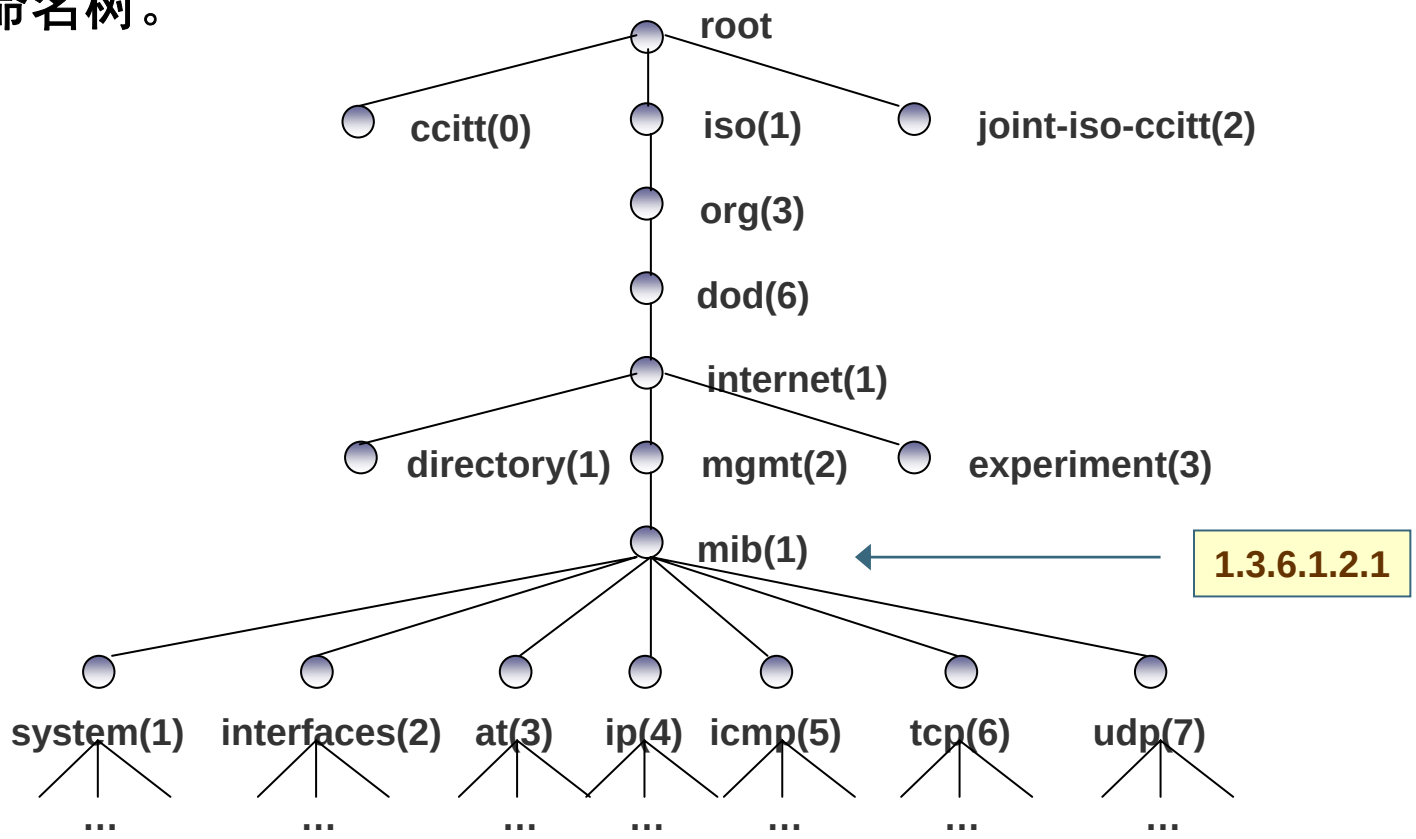
- get-request：从代理进程处提取一个或多个参数值。
- get-next-request：从代理进程处提取一个或多个参数的下一个参数值。
- set-request：设置代理进程的一个或多个参数值。
- get-response：返回的一个或多个参数值。
- trap：代理进程主动通知管理进程有某些事情发生。

SNMP协议的报文格式



MIB变量的表示方法

- MIB变量（对象）是用对象标识符OID来表示的，OID是一个由“.”号来分隔的数字序列，是一种层次结构的对象命名方法，被称作对象命名树。



5.4.2 SNMP协议的安全性分析

- SNMPv1是基于Community（团体）字串来进行访问控制的。
  - 读和写操作需要各自的团体字串。
  - 由于缺省设置分别为Public和Private，并且在网络中以明文传输，因此给基于SNMP的通信带来了极大的危险性。
- 常用的SNMP操作工具
  - Windows 2000 Resource Kit工具包中的snmputil。
  - Unix系统中的snmpwalk、snmpget以及snmpset。
  - SolarWinds工具套件等。

5.5 DNS协议及安全性

- DNS协议的基本概念
- DNS协议的安全性分析

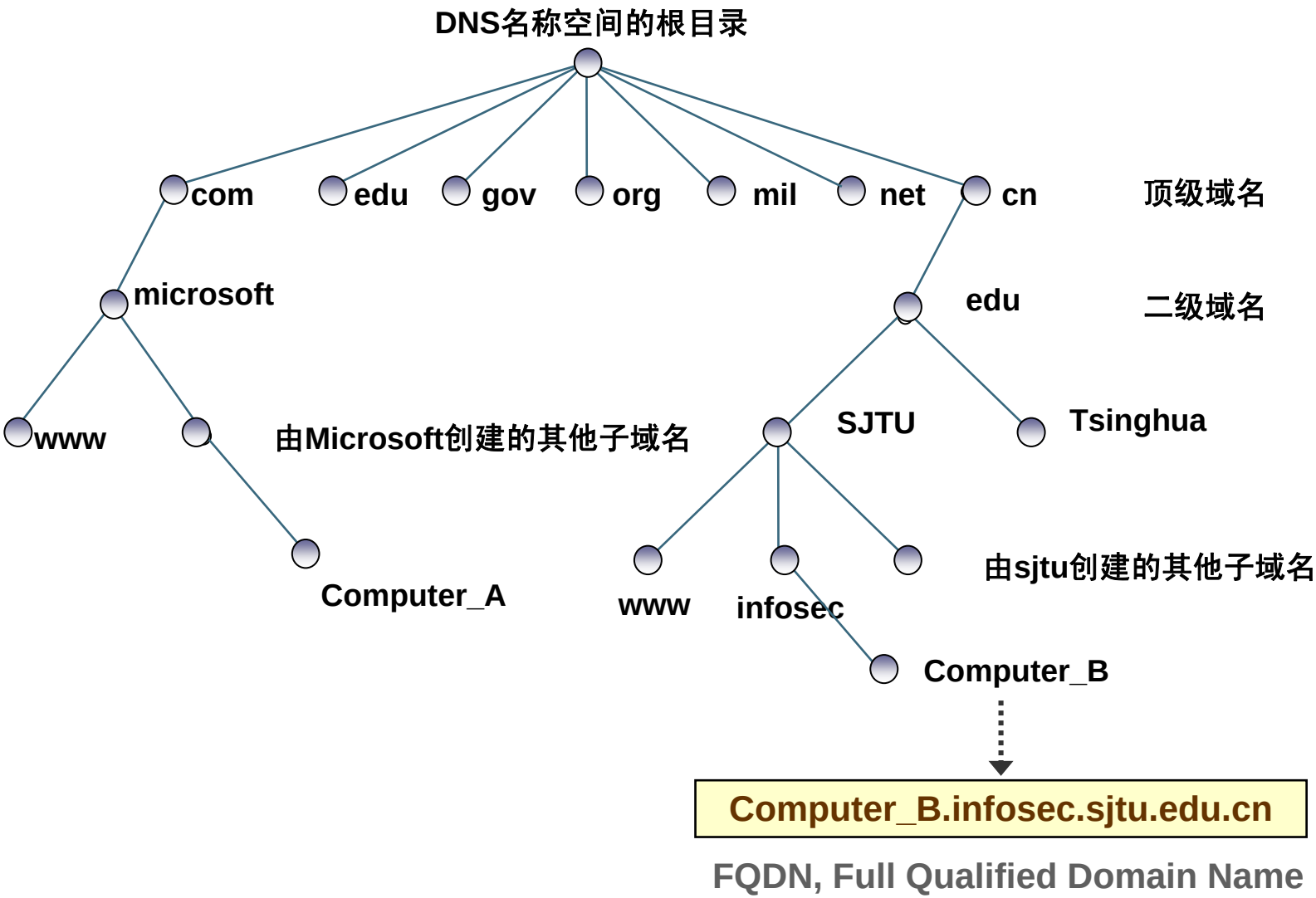
RFC 1035

RFC 1034

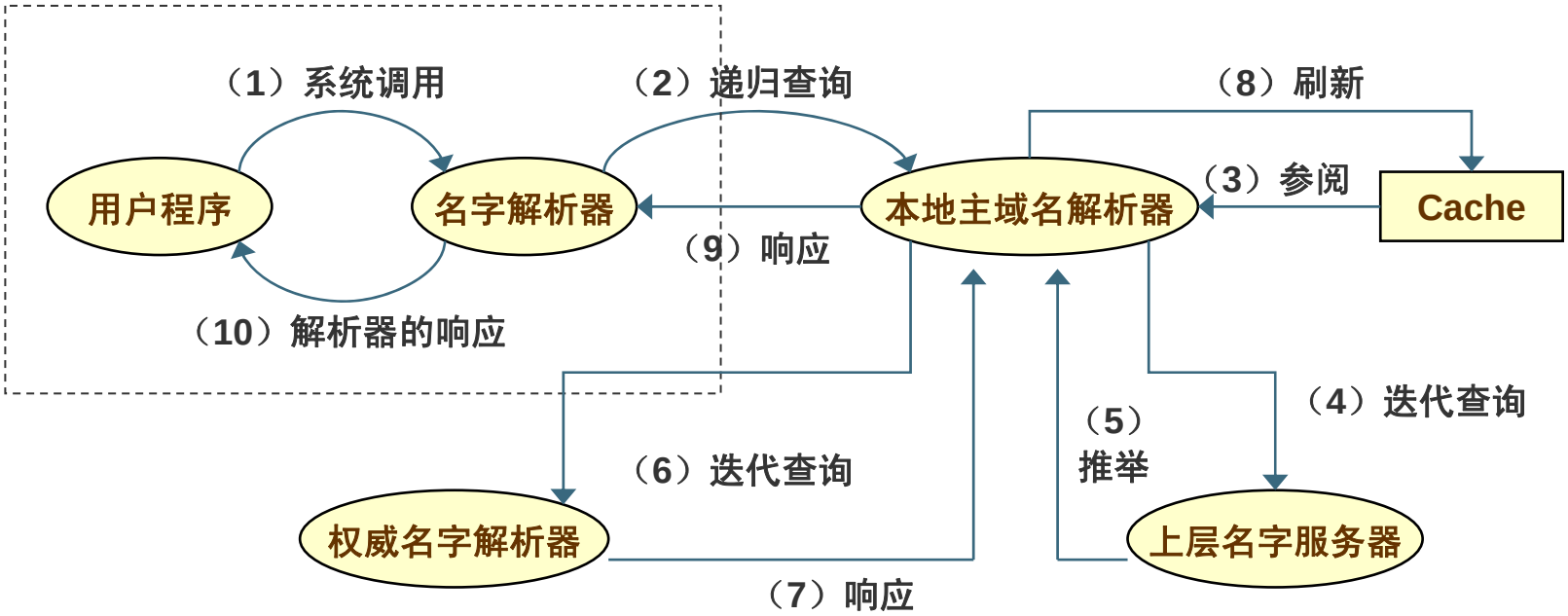
5.5.1 DNS协议基本概念

- Domain Name System是个分布式数据库系统，在基于TCP/IP的网络中，用来进行主机名和IP地址间的相互转换。
- 层次结构的名字空间被分成了许多不同的域，这些域被不同的名字服务器所管理，名字服务器是其所辖域的权威。
- Fully Qualified Domain Name（FQDN）
  - 正式域名，或者叫绝对域名，以一个点号结束，如infosec.sjtu.edu.cn。

DNS服务的层次结构


DNS名称空间的根目录
com
edu
gov
org
mil
net
cn
顶级域名
microsoft
edu
二级域名
www
由Microsoft创建的其他子域名
SJTU
Tsinghua
Computer_A
www
infosec
由sjtu创建的其他子域名
Computer_B
Computer_B.infosec.sjtu.edu.cn
FQDN, Full Qualified Domain Name

- 客户端使用解析器向DNS服务器发送域名解析请求。
- 客户端解析器进行域名请求时可采用两种方式：
  - 递归（recursive）
  - 迭代（iterative）



域名解析过程

DNS资源记录（Resource Record）

- 名字服务器维护着若干保存数据的区域文件（Zone File），一个域名往往对应两个区域文件，其一用作主机名到IP地址的转换，另一个用作反向地址解析，资源记录是区域文件内容的构成部分，是DNS的核心数据。

RR类型	说明
开始授权记录SOA	所有正向和反向区域文件中必要的第一个条目，提供每个域需要的若干关键信息，最重要的是指定了域的授权域名服务器。
域名服务器记录NS	指定DNS域的域名服务器。
反向查询指针记录PTR	提供反向地址解析的关键记录（IP到域名）。
地址记录A	主机名到IP地址的映射。这些记录在正向查找区域文件和根缓存文件中使用。
邮件交换记录MX	提供邮件交换服务器的记录信息。
规范名记录CNAME	为一个规范名（域名、主机名）建立别名。
主机信息记录HINFO	标示CPU和操作系统等主机信息。

5.5.2 DNS协议的安全性分析

- DNS服务器很容易被欺骗，可能遭受如下攻击：
  - DNS欺骗攻击。
  - DNS会话劫持攻击。
  - Cache毒害攻击。
    - 某些DNS服务器会不加甄别地把收到的DNS应答保存在Cache中，攻击者可以向该DNS服务器发送伪造的DNS应答包，以毒害其Cache
- DNS服务器中保存有大量敏感信息，如果配置不当，可能遭受域名文件传输攻击，导致内部网络拓扑结构为人所知。
- DNS软件本身存在漏洞（如BIND），可能遭受缓冲区溢出攻击。
- DNS负责整个网络的域名解析任务，其性能状况直接关系到网络访问的好坏，为此，攻击者常对DNS服务器实施DoS攻击。

SJTU Information Security Institute

Network Attack & Defence Technology Research Studio

Any Questions ?

确定 取消