Information Security 15

防火墙

Chapter 18、20 & supplements

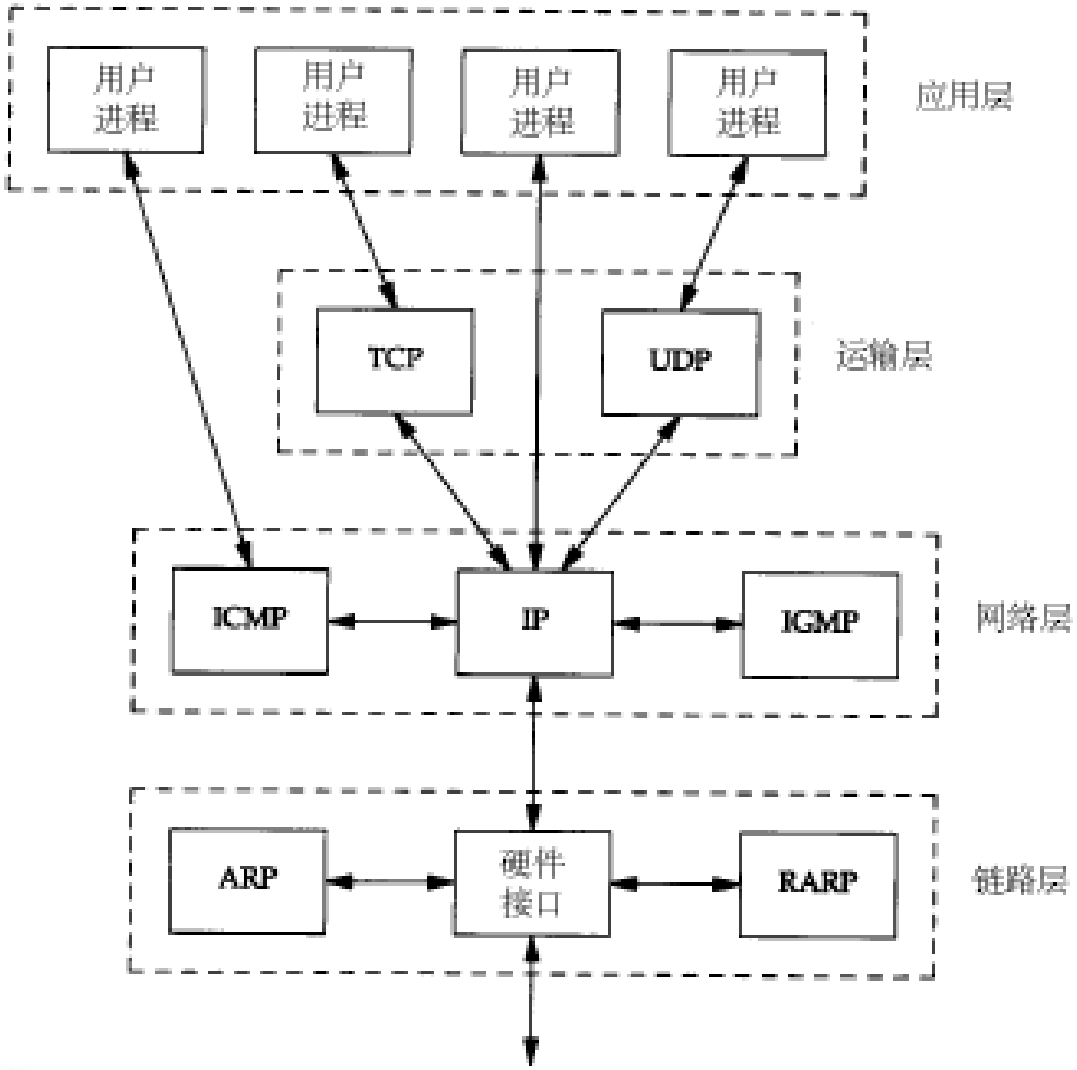
内容

- TCP/IP基础
- 防火墙
  - 防火墙的基本介绍
  - 几种防火墙的类型
  - 防火墙的配置
  - 防火墙技术的发展

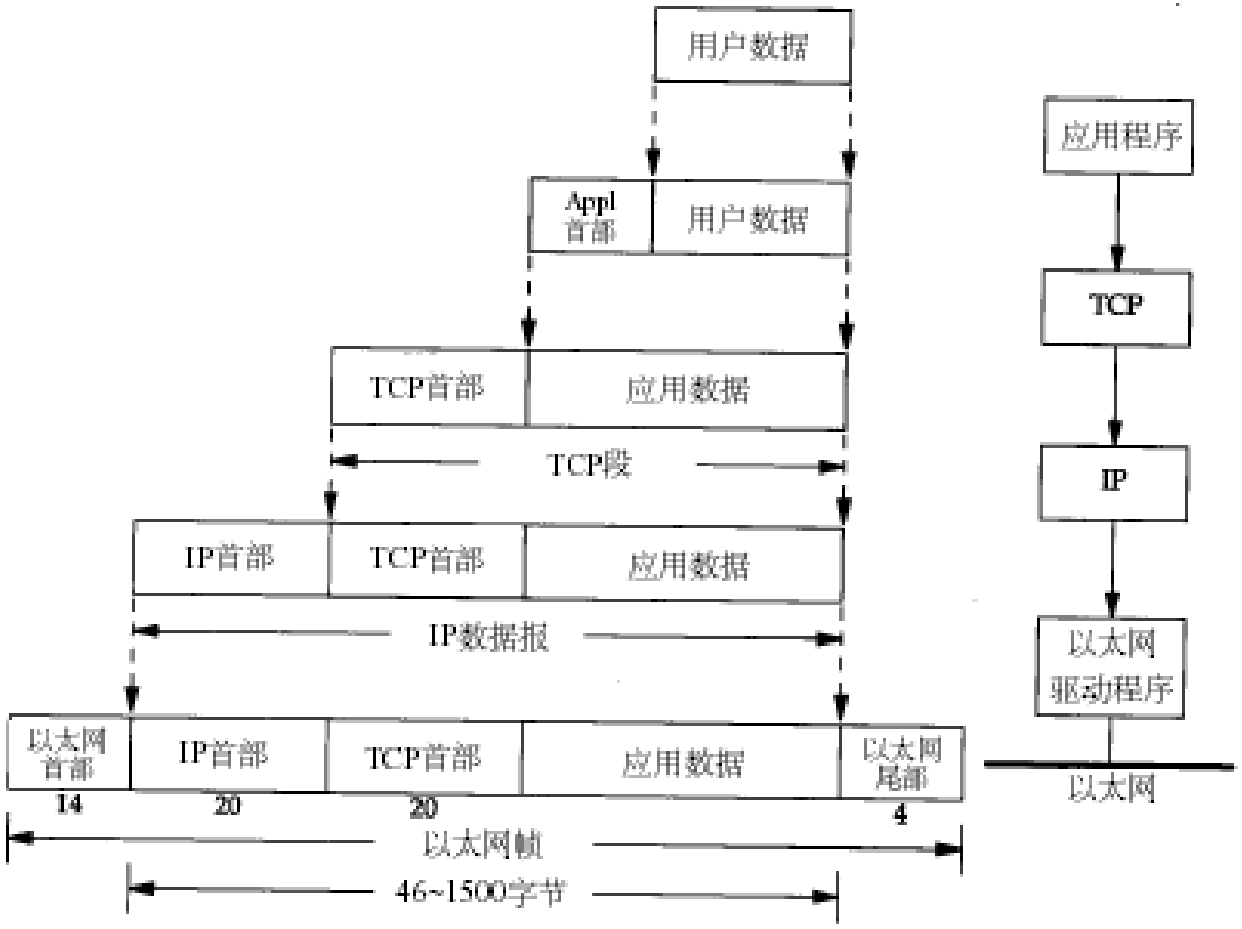
TCP/IP overview

- 协议栈
- 一些数据包的格式
  - IP数据包
  - TCP/UDP数据包
- 常用的上层协议
- 几个常用工具

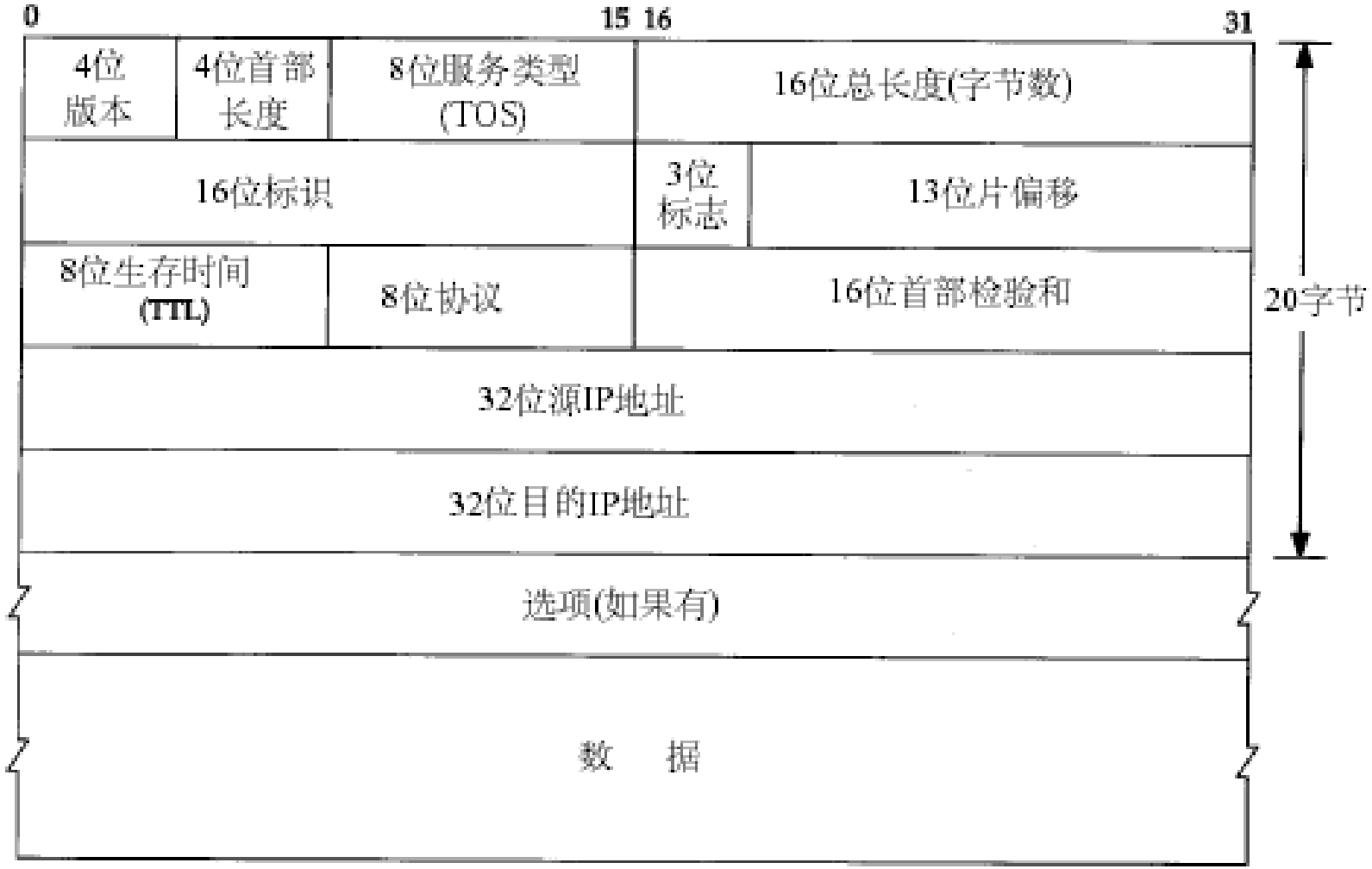
TCP/IP协议栈


用户进程
用户进程
用户进程
用户进程
应用层
TCP
UDP
运输层
ICMP
IP
IGMP
网络层
ARP
硬件接口
RARP
链路层

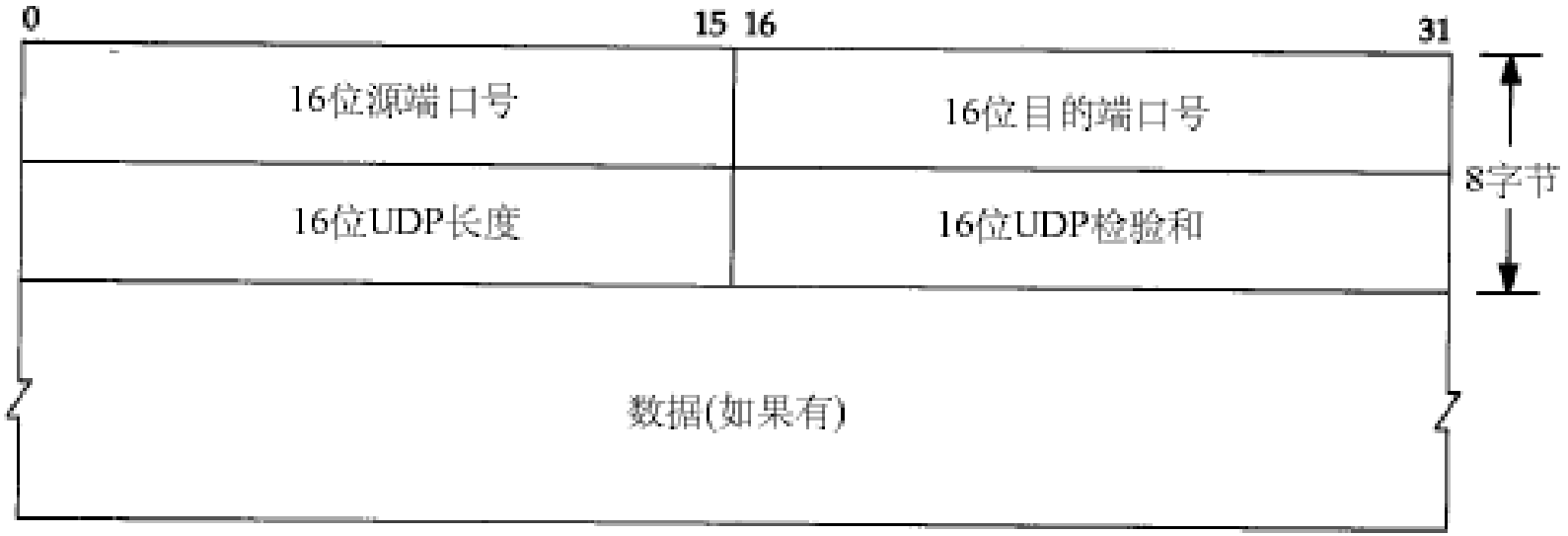
协议栈各层数据包的结构



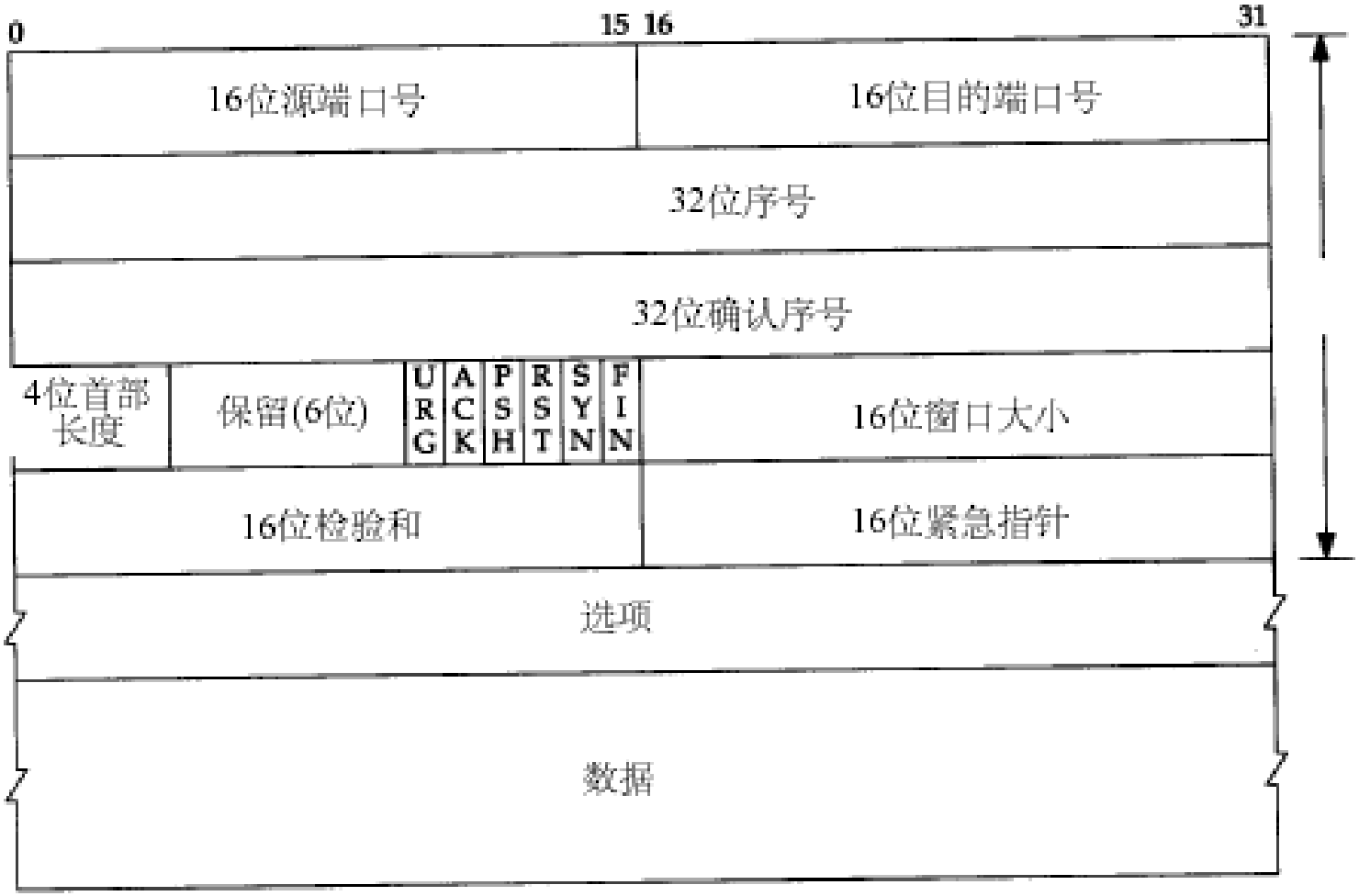
IP数据包格式


0
15 16
31
4位
版本
4位首部
长度
8位服务类型
(TOS)
16位总长度(字节数)
16位标识
3位
标志
13位片偏移
8位生存时间
(TTL)
8位协议
16位首部检验和
20字节
32位源IP地址
32位目的IP地址
选项(如果有)
数　据

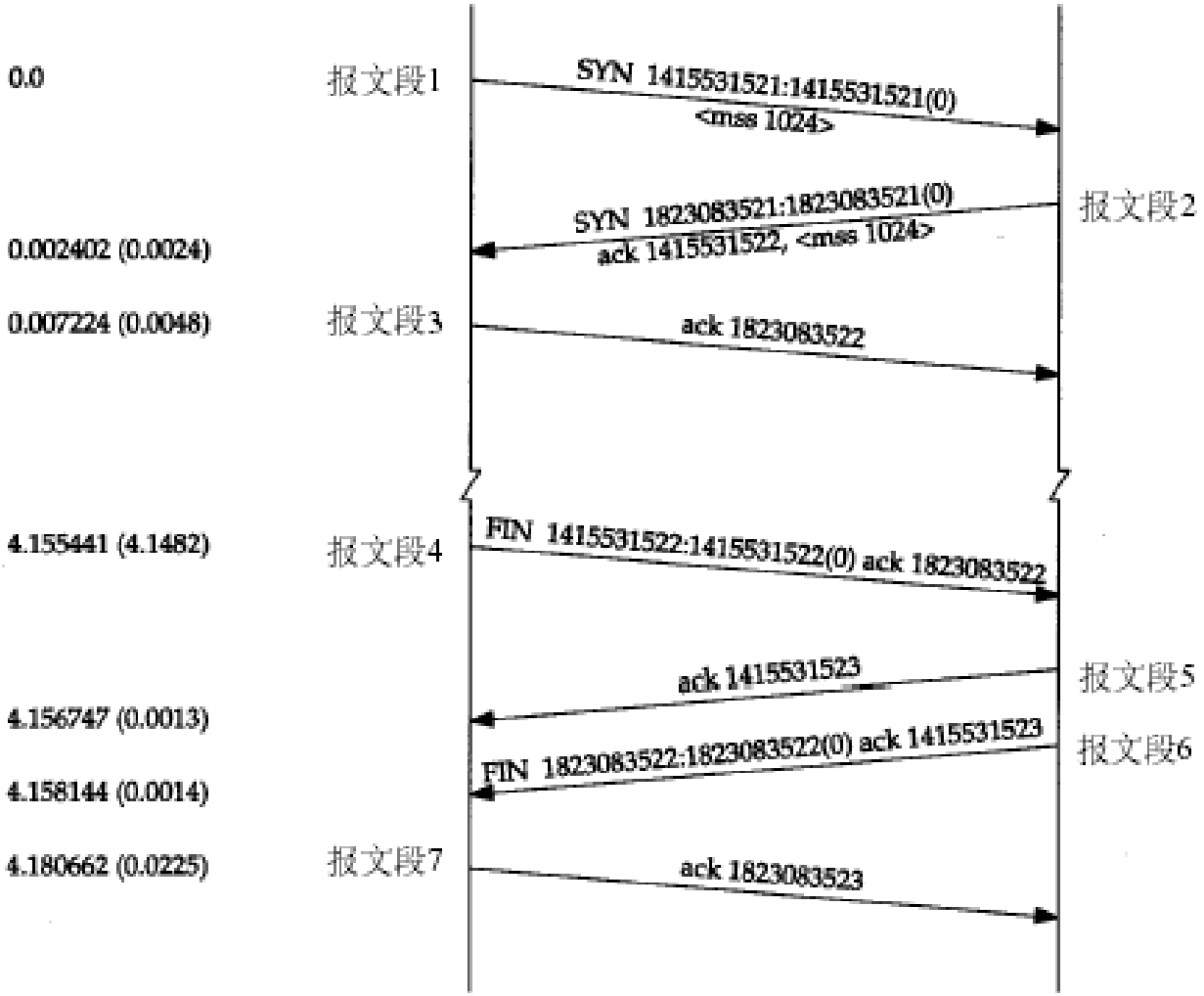
UDP数据包格式



TCP数据包格式


0
15 16
31
16位源端口号
16位目的端口号
32位序号
32位确认序号
4位首部长度
保留(6位)
URG
ACK
PSH
RST
SYN
FIN
16位窗口大小
16位检验和
16位紧急指针
选项
数据

TCP连接的建立和终止时序图



常用的上层协议

- DNS: 53/tcp,udp
- FTP: 20,21/tcp,udp
- telnet: 23/tcp,udp
- HTTP: 80/tcp,udp
- NNTP: 119/tcp,udp
- SMTP: 25/tcp,udp
- POP3: 110/tcp,udp
- 参考：IANA提供的port-numbers.txt

常用的网络工具

- Netstat
- Ipconfig/ifconfig
- Ping
- Tracert
- ……

内容

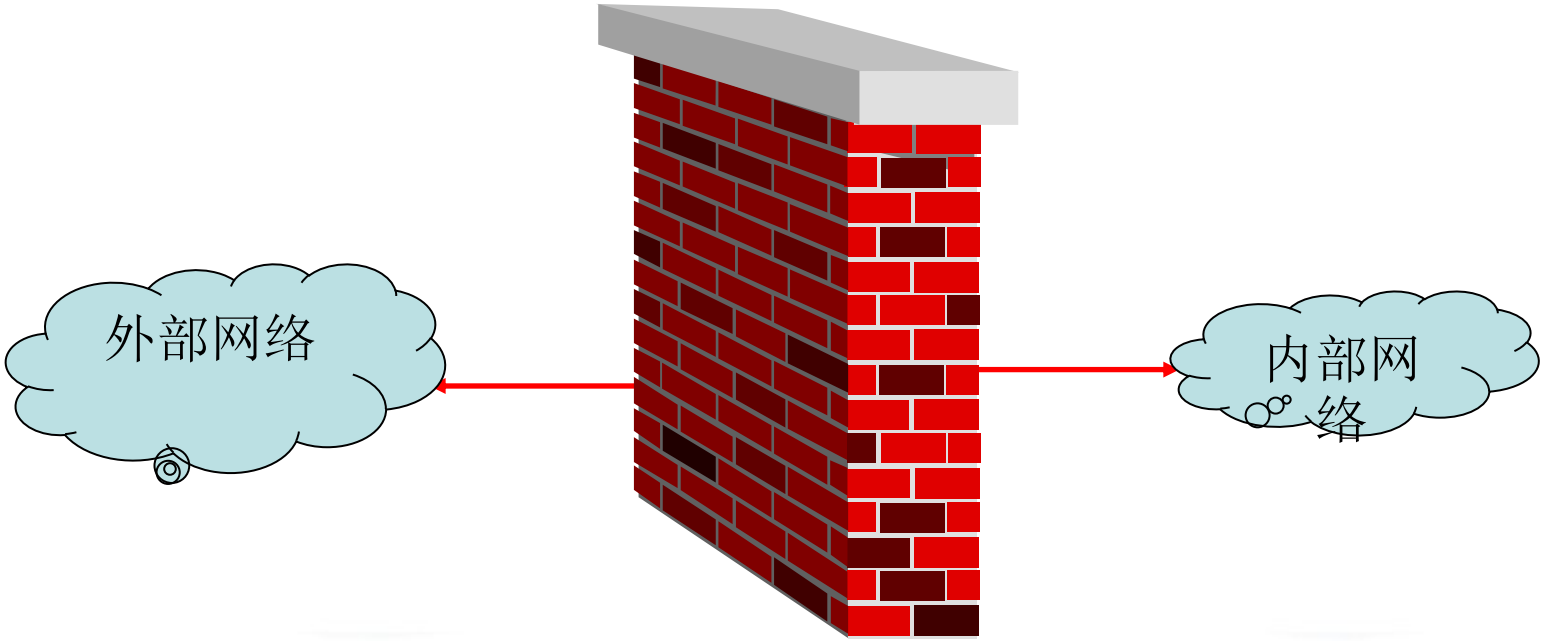
- TCP/IP基础
- **防火墙**
  - 防火墙的基本介绍
  - 几种防火墙的类型
  - 防火墙的配置
  - 防火墙技术的发展

防火墙

- 截断可燃或难燃屋顶结构

防火墙

- 互联网、内部网和外部网
- Internet ，intranet ，Extranet



互联网、内部网和外部网

- 内部网是应用于组织内部，使用Internet技术的专用网络。
  - 实现信息在组织内部及分支机构间的传播。
- 外部网是内部网的扩展，它将组织的内部网连入其业务伙伴、顾客或供应商的网络
  - 外部网是组织间主要的沟通方式
  - 类型
    - 公共网络
    - 安全（专用）网络
    - 虚拟专用网络（VPN）

防火墙(Firewall)

- 防火墙的基本设计目标
  - 对于一个网络来说，所有通过“内部”和“外部”的网络流量都要经过防火墙
  - 通过一些安全策略，来保证只有经过授权的流量才可以通过防火墙
  - 防火墙本身必须建立在安全操作系统的基础上
- 防火墙的控制能力
  - 服务控制，确定哪些服务可以被访问
  - 方向控制，对于特定的服务，可以确定允许哪个方向能够通过防火墙
  - 用户控制，根据用户来控制对服务的访问
  - 行为控制，控制一个特定的服务的行为

防火墙能为我们做什么

- 定义一个必经之点
  - 挡住未经授权的访问流量
  - 禁止具有脆弱性的服务带来危害
  - 实施保护，以避免各种IP欺骗和路由攻击
- 防火墙提供了一个监视各种安全事件的位置，所以，可以在防火墙上实现审计和报警
- 对于有些Internet功能来说，防火墙也可以是一个理想的平台，比如地址转换，Internet日志、审计，甚至计费功能
- 防火墙可以作为IPSec的实现平台

防火墙本身的一些局限性

- 对于绕过防火墙的攻击，它无能为力，例如，在防火墙内部通过拨号出去
- 防火墙不能防止内部的攻击，以及内部人员与外部人员的联合攻击(比如，通过tunnel进入)
- 防火墙不能防止被病毒感染的程序或者文件、邮件等
- 防火墙的性能要求

内容

- TCP/IP基础
- **防火墙**
  - 防火墙的基本介绍
  - 几种防火墙的类型
  - 防火墙的配置
  - 防火墙技术的发展

防火墙的类型

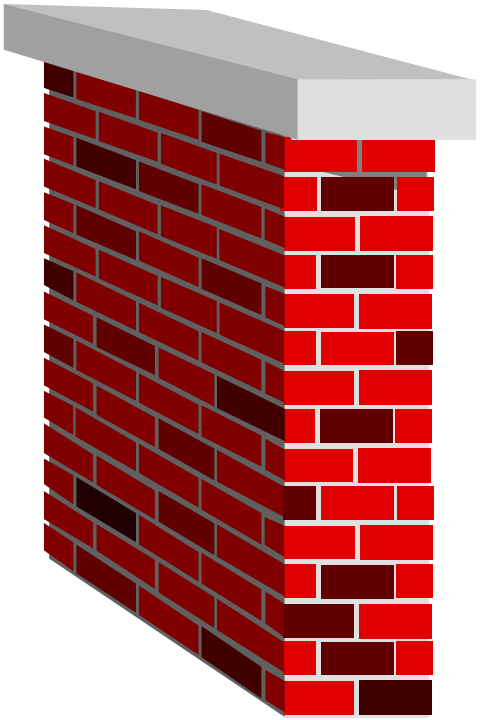
- 包过滤路由器
- 应用层网关

包过滤路由器

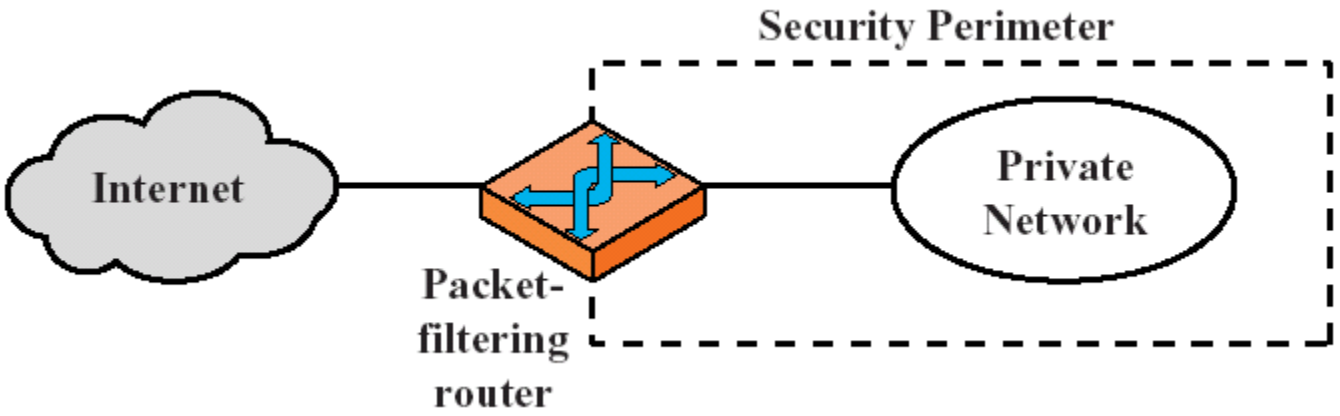
- 基本的思想很简单
  - 对于每个进来的包，适用一组规则，然后决定转发或者丢弃该包
  - 往往配置成双向的
- 如何过滤
  - 过滤的规则以IP和传输层的头中的域(字段)为基础，
    - 包括源和目标IP地址、IP协议域、源和目标端口、标志位
  - 过滤器往往建立一组规则，根据IP包是否匹配规则中指定的条件来作出决定。
    - 如果匹配到一条规则，则根据此规则决定转发或者丢弃
    - 如果所有规则都不匹配，则根据缺省策略

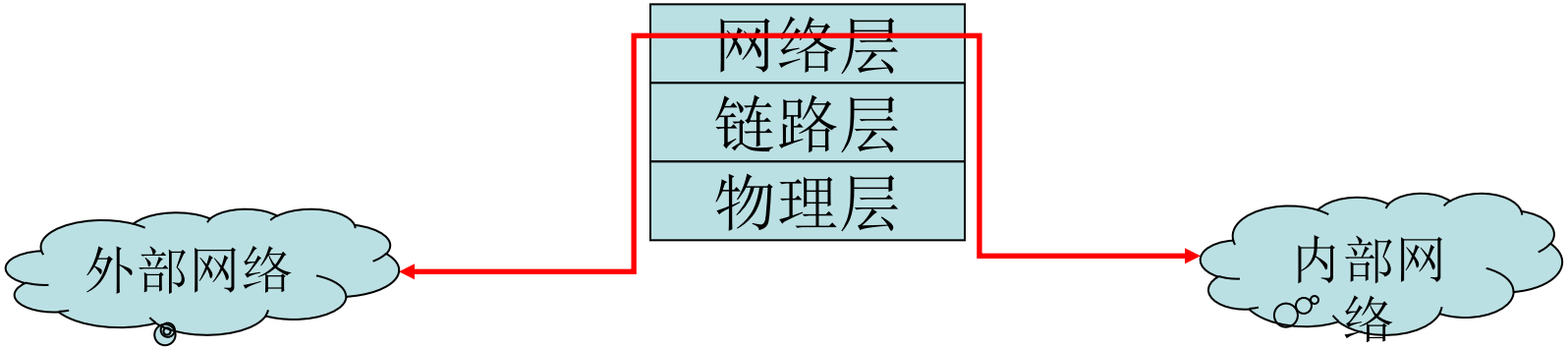
安全缺省策略

- 两种基本策略，或缺省策略
  - 没有被拒绝的流量都可以通过
    - 管理员必须针对每一种新出现的攻击，制定新的规则
  - 没有被允许的流量都要拒绝
    - 比较保守
    - 根据需要，逐渐开放



包过滤路由器示意图


Security Perimeter
Internet
Private Network
Packet-filtering router

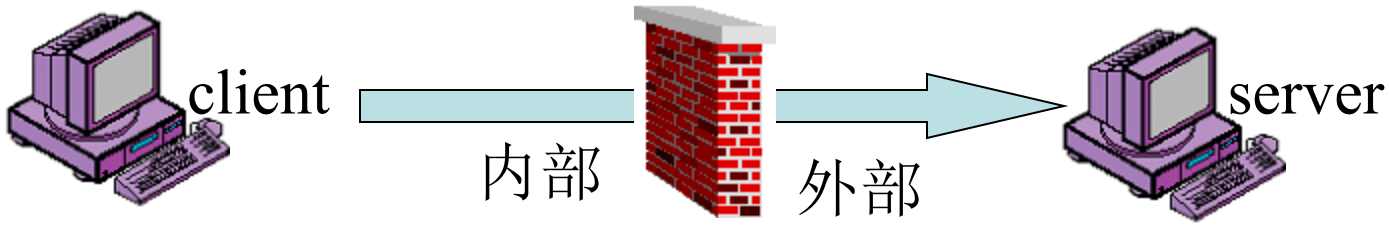

网络层
链路层
物理层
外部网络
内部网络

包过滤防火墙

- 在网络层上进行监测
  - 并没有考虑连接状态信息
- 通常在路由器上实现
  - 实际上是一种网络的访问控制机制
- 优点：
  - 实现简单
  - 对用户透明
  - 效率高
- 缺点：
  - 正确制定规则并不容易
  - 不可能引入认证机制

包过滤防火墙的设置(1)

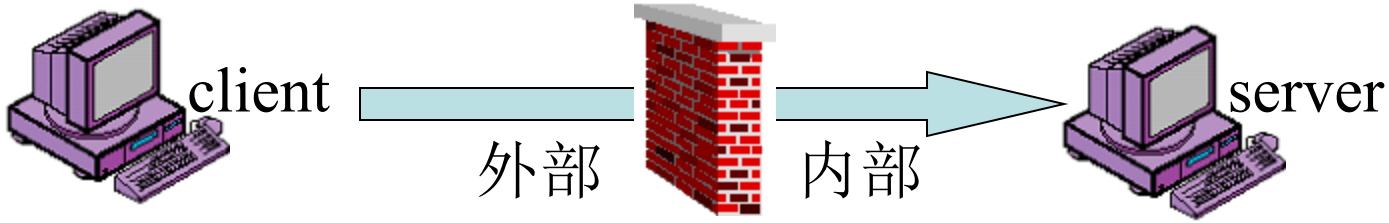
- 从内往外的telnet服务



- 往外包的特性(用户操作信息)
  - IP源是内部地址
  - 目标地址为server
  - TCP协议，目标端口23
  - 源端口>1023
  - 连接的第一个包ACK=0，其他包ACK=1
- 往内包的特性(显示信息)
  - IP源是server
  - 目标地址为内部地址
  - TCP协议，源端口23
  - 目标端口>1023
  - 所有往内的包都是ACK=1

包过滤防火墙的设置(2)

- 从外往内的telnet服务



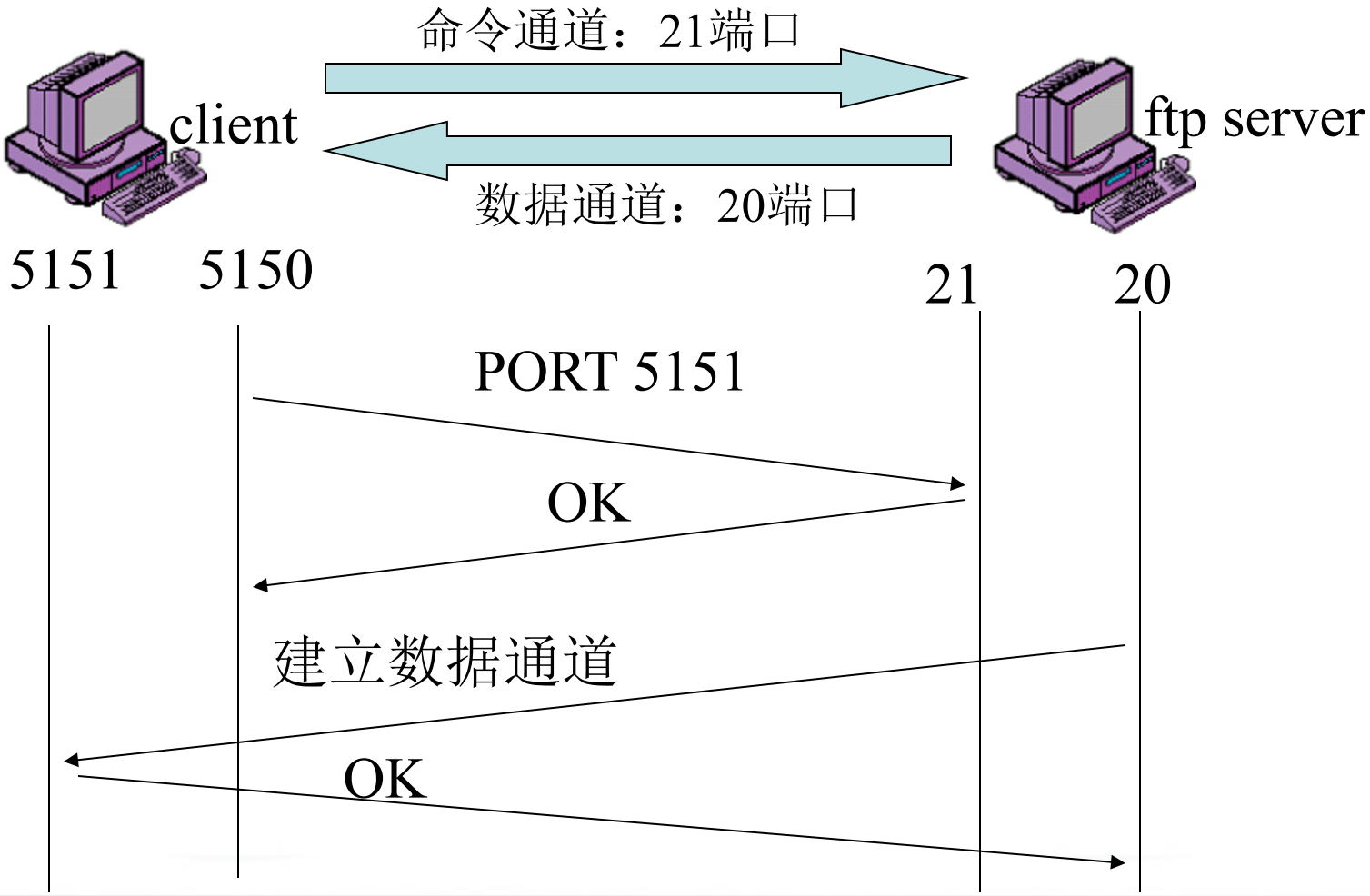
- 往内包的特性(用户操作信息)
  - IP源是外部地址
  - 目标地址为本地server
  - TCP协议，目标端口23
  - 源端口>1023
  - 连接的第一个包ACK=0，其他包ACK=1
- 往外包的特性(显示信息)
  - IP源是本地server
  - 目标地址为外部地址
  - TCP协议，源端口23
  - 目标端口>1023
  - 所有往内的包都是ACK=1

针对telnet服务的防火墙规则

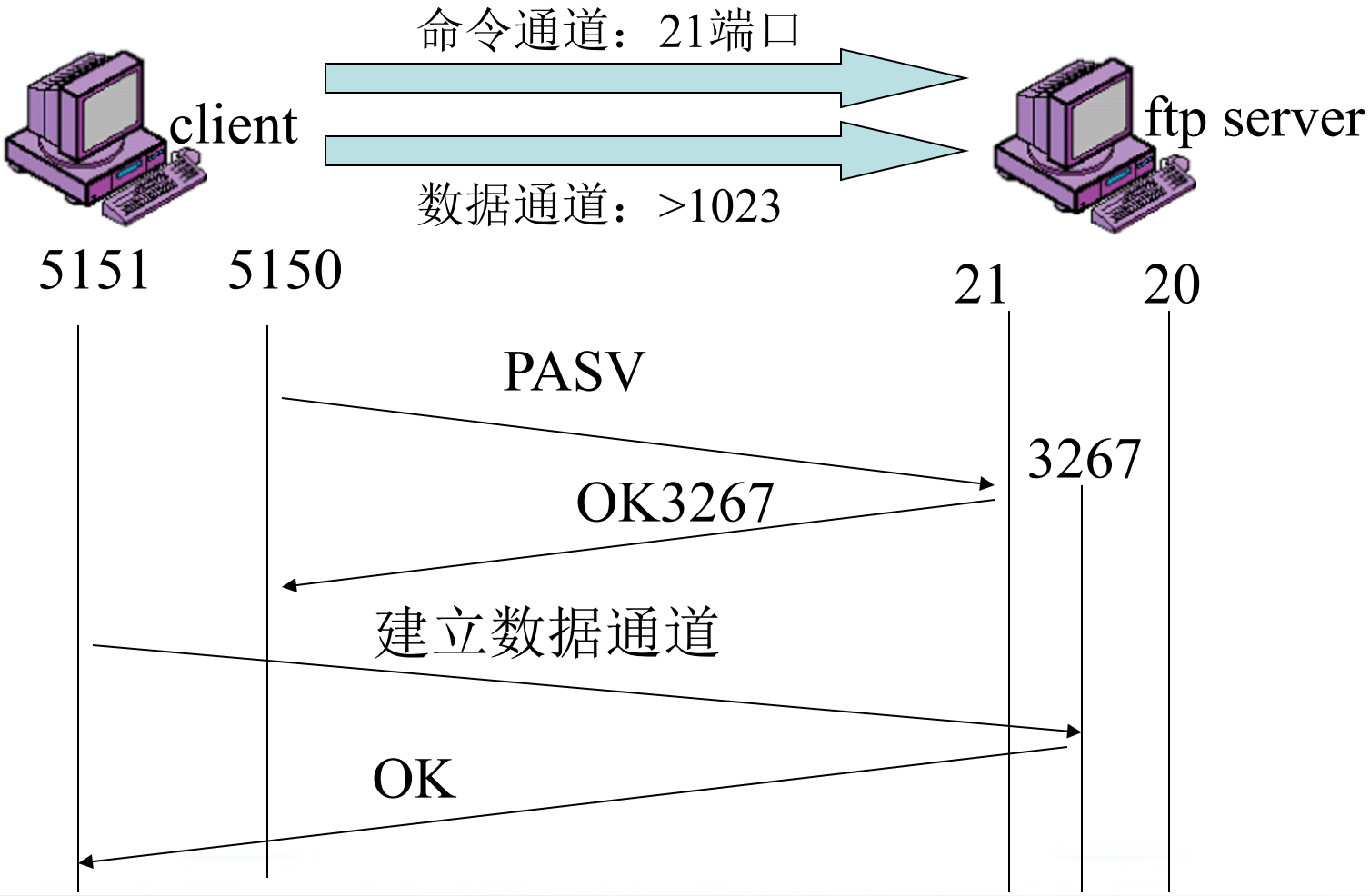
服务方向	包方向	源地址	目标地址	包类型	源端口	目标端口	ACK
往外	外	内部	外部	TCP	>1023	23	*
往外	内	外部	内部	TCP	23	>1023	1
往内	外	外部	内部	TCP	>1023	23	*
往内	内	内部	外部	TCP	23	>1023	1

*: 第一个ACK=0, 其他=1

Ftp文件传输协议


命令通道：21端口
client
ftp server
数据通道：20端口
5151
5150
21
20
PORT 5151
OK
建立数据通道
OK

Ftp文件传输协议(续)


命令通道：21端口
client
ftp server
数据通道：>1023
5151
5150
21
20
PASV
3267
OK3267
建立数据通道
OK

- 建立一组复杂的规则集
  - 是否允许正常模式的ftp数据通道？
  - 有些ftp client不支持pasv模式
- 动态监视ftp通道发出的port命令
  - 有一些动态包过滤防火墙可以做到
- 启示
  - 包过滤防火墙比较适合于单连接的服务(比如smtp, pop3)，不适合于多连接的服务(比如ftp)

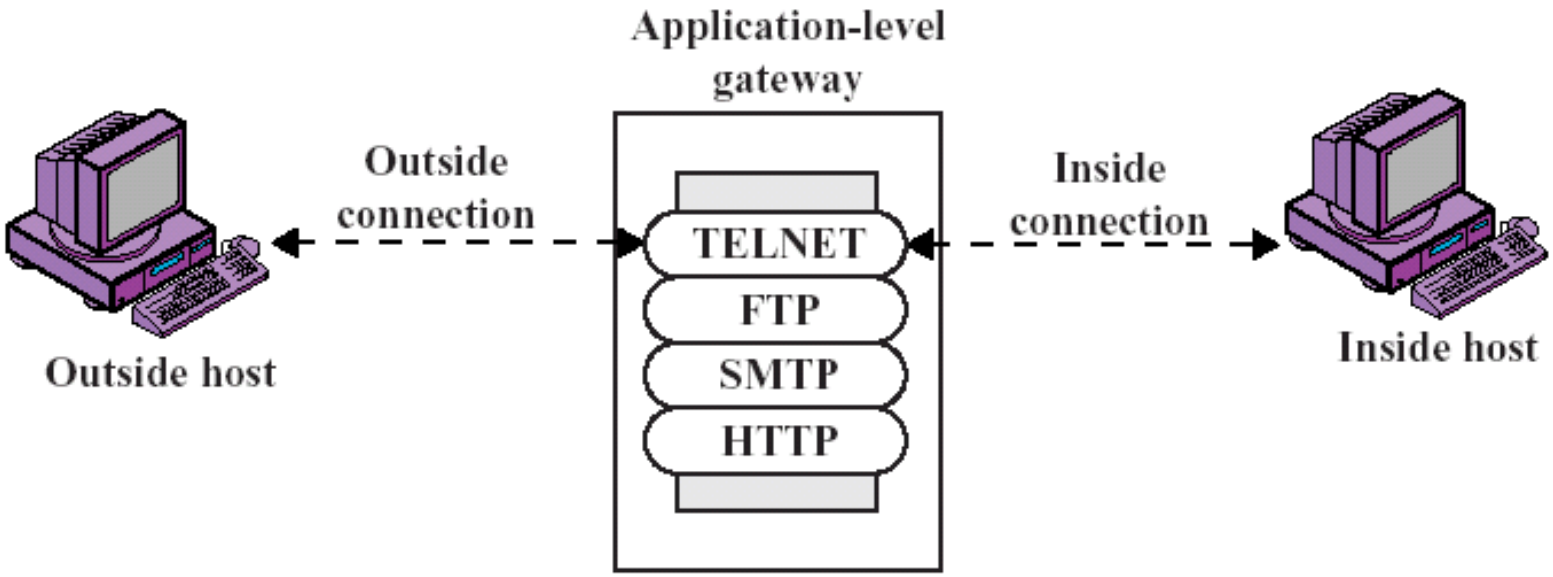
针对包过滤防火墙的攻击

- IP地址欺骗，例如，假冒内部的IP地址
  - 对策：在外部接口上禁止内部地址
- 源路由攻击，即由源指定路由
  - 对策：禁止这样的选项
- 小碎片攻击，利用IP分片功能把TCP头部切分到不同的分片中
  - 对策：丢弃分片太小的分片
- 利用复杂协议和管理员的配置失误进入防火墙
  - 例如，利用ftp协议对内部进行探查

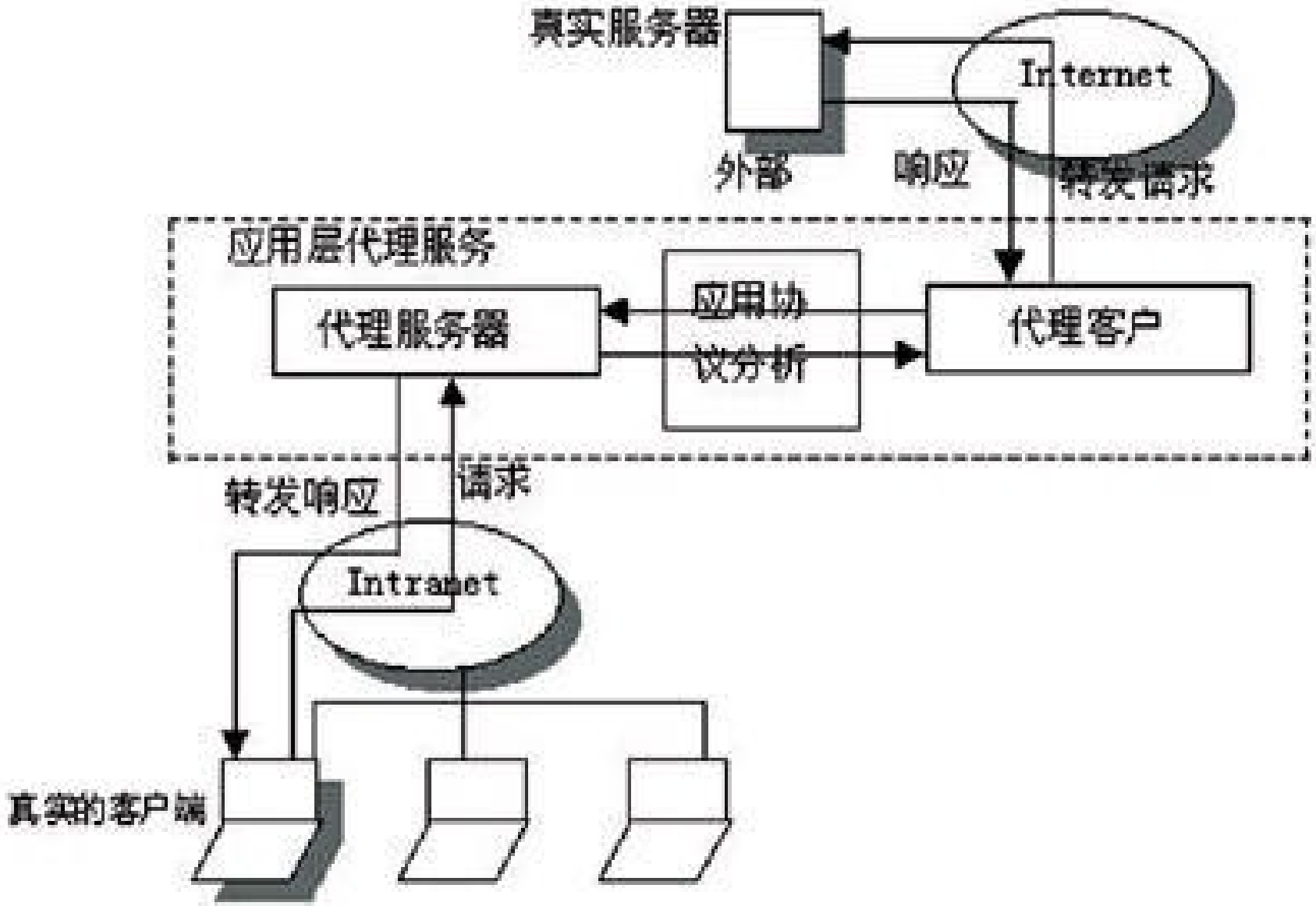
应用层网关

- 也称为代理服务器
- 特点
  - 所有的连接都通过防火墙，防火墙作为网关
  - 在应用层上实现
  - 可以监视包的内容
  - 可以实现基于用户的认证
  - 所有的应用需要单独实现
  - 可以提供理想的日志功能
  - 非常安全，但是开销比较大

应用层网关的结构示意图



应用层网关结构示意图


真实服务器
Internet
外部
响应
转发请求
应用层代理服务
代理服务器
应用协
议分析
代理客户
转发响应
请求
Intranet
真实的客户端

应用层网关的优缺点

- 优点
  - 允许用户“直接”访问Internet
  - 易于记录日志
- 缺点
  - 新的服务不能及时地被代理
  - 每个被代理的服务都要求专门的代理软件
  - 客户软件需要修改，重新编译或者配置
  - 有些服务要求建立直接连接，无法使用代理
    - 比如聊天服务、或者即时消息服务
  - 代理服务不能避免协议本身的缺陷或者限制

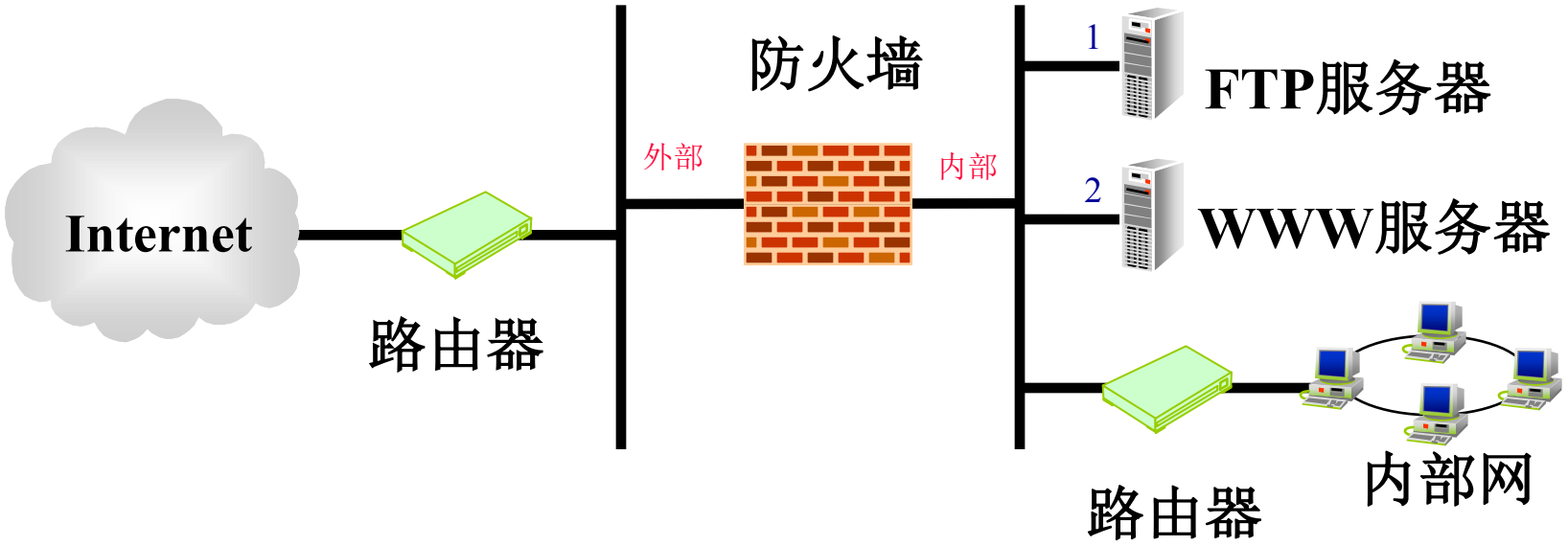
应用层网关实现

- 编写代理软件
  - 代理软件一方面是服务器软件
    - 但是它所提供的服务可以是简单的转发功能
  - 另一方面也是客户软件
    - 对于外面真正的服务器来说，是客户软件
  - 针对每一个服务都需要编写模块或者单独的程序
  - 实现一个标准的框架，以容纳各种不同类型的服务
    - 软件实现的可扩展性和可重用性
- 客户软件
  - 软件需要定制或者改写
  - 对于最终用户的透明性？
- 协议对于应用层网关的处理
  - 协议设计时考虑到中间代理的存在，特别是在考虑安全性，比如数据完整性的时候

内容

- TCP/IP基础
- **防火墙**
  - 防火墙的基本介绍
  - 几种防火墙的类型
  - 防火墙的配置
  - 防火墙技术的发展

问题：现实环境下防火墙配置？



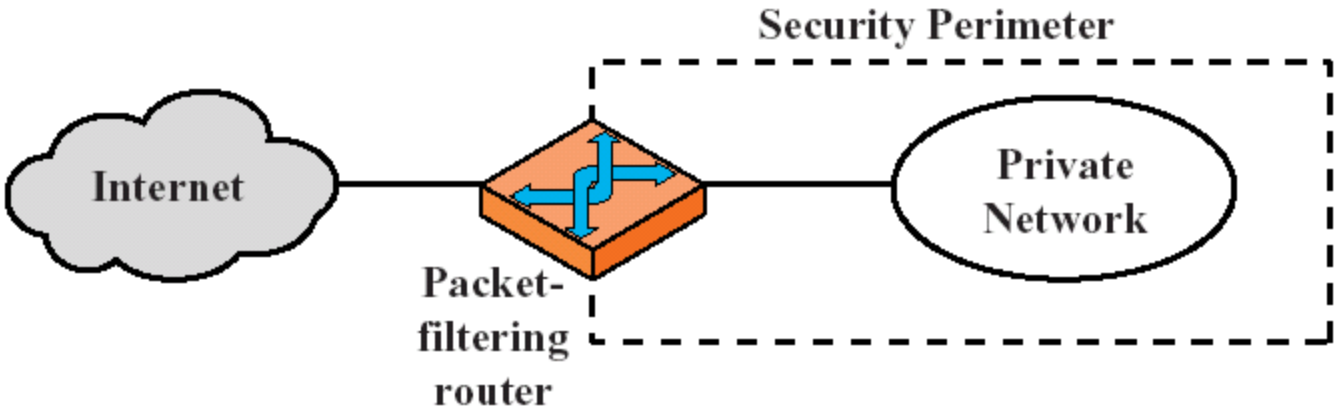
防火墙的配置

- 几个概念
  - 堡垒主机(Bastion Host)：对外部网络暴露，同时也是内部网络用户的主要连接点
  - 双宿主主机(dual-homed host)：至少有两个网络接口的通用计算机系统
  - DMZ(Demilitarized Zone，非军事区或者停火区)：在内部网络和外部网络之间增加的一个子网

防火墙几种典型配置方案

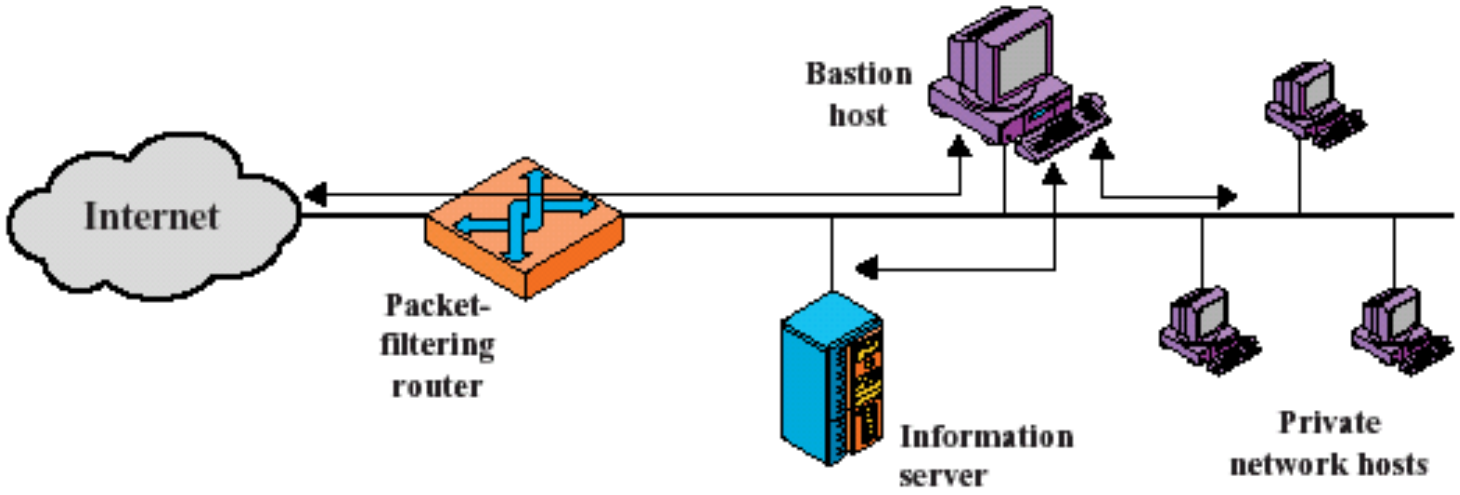
- 包过滤
- 屏蔽主机方案
  - 单宿主堡垒主机
  - 双宿主堡垒主机
- 屏蔽子网方案

配置方案一



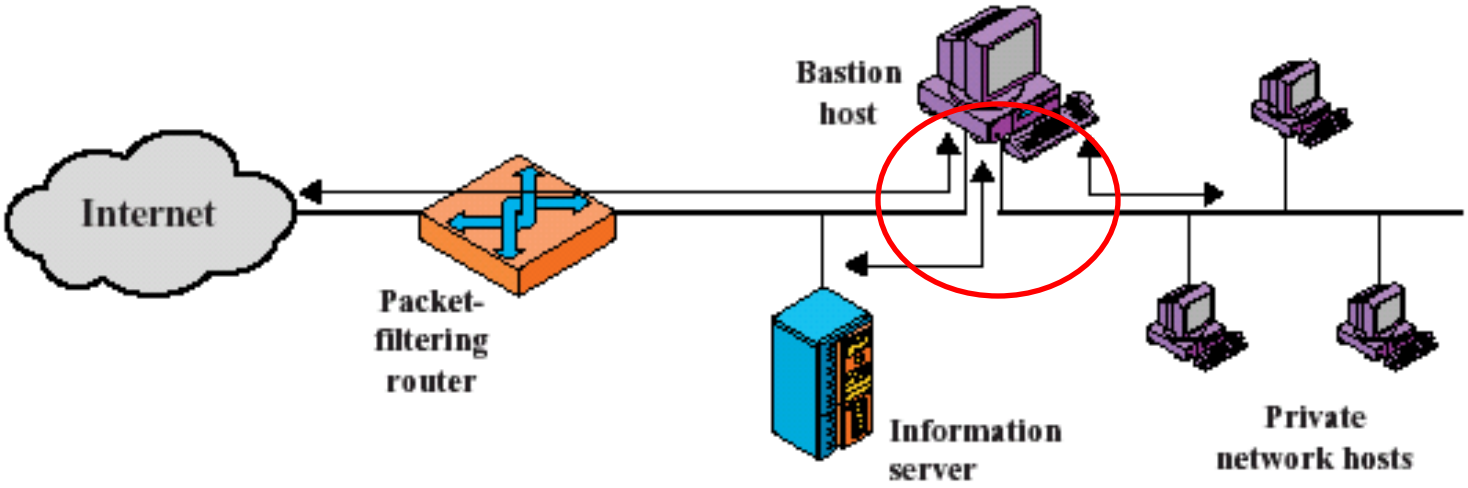
- 包过滤方案
- 所有的流量都通过堡垒（包过滤路由器）
- 优点：简单

配置方案二



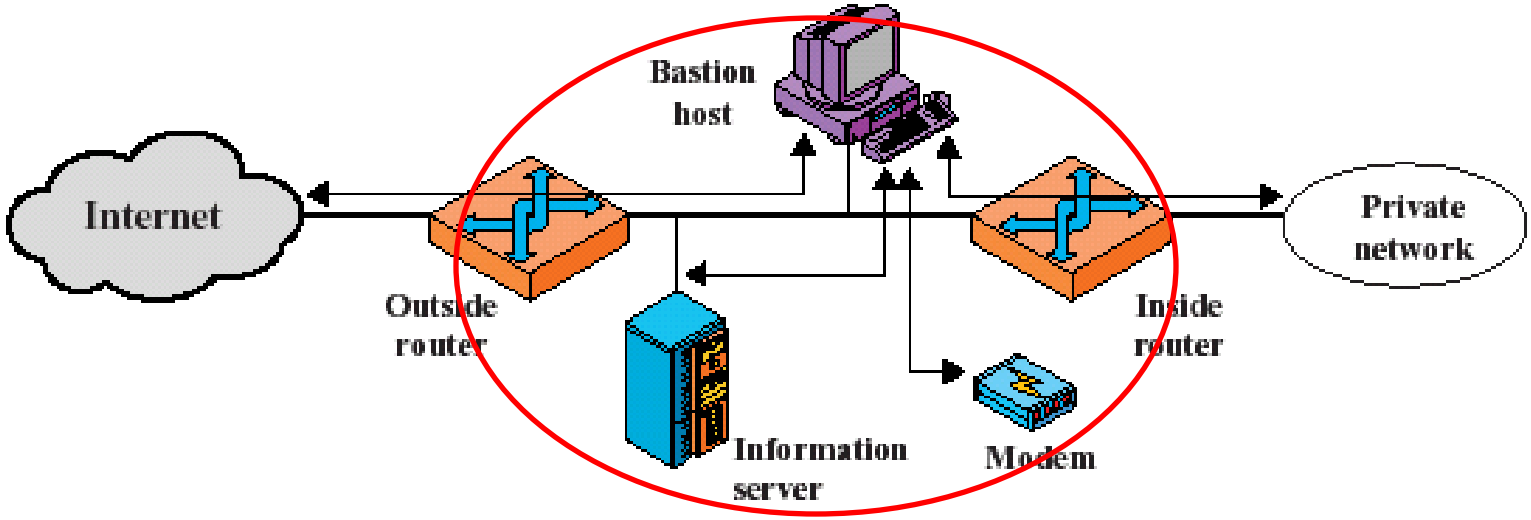
- 屏蔽主机方案：单宿主堡垒主机
- 只允许堡垒主机可以与外界直接通讯
- 优点：两层保护：包过滤+应用层网关；灵活配置
- 缺点：一旦包过滤路由器被攻破，则内部网络被暴露

配置方案三



- 屏蔽主机方案：双宿主堡垒主机
- 从物理上把内部网络和Internet隔开，必须通过两层屏障
- 优点：两层保护：包过滤+应用层网关；配置灵活

配置方案四



- 屏蔽子网防火墙
- 优点：
  - 三层防护，用来阻止入侵者
  - 外面的router只向Internet暴露屏蔽子网中的主机
  - 内部的router只向内部私有网暴露屏蔽子网中的主机

防火墙的发展

- 分布式防火墙
- 应用层网关的进一步发展
  - 认证机制
  - 智能代理
- 与其他技术的集成
  - 比如NAT、VPN(IPSec)、IDS，以及一些认证和访问控制技术
  - 防火墙自身的安全性和稳定性

Information Security 16

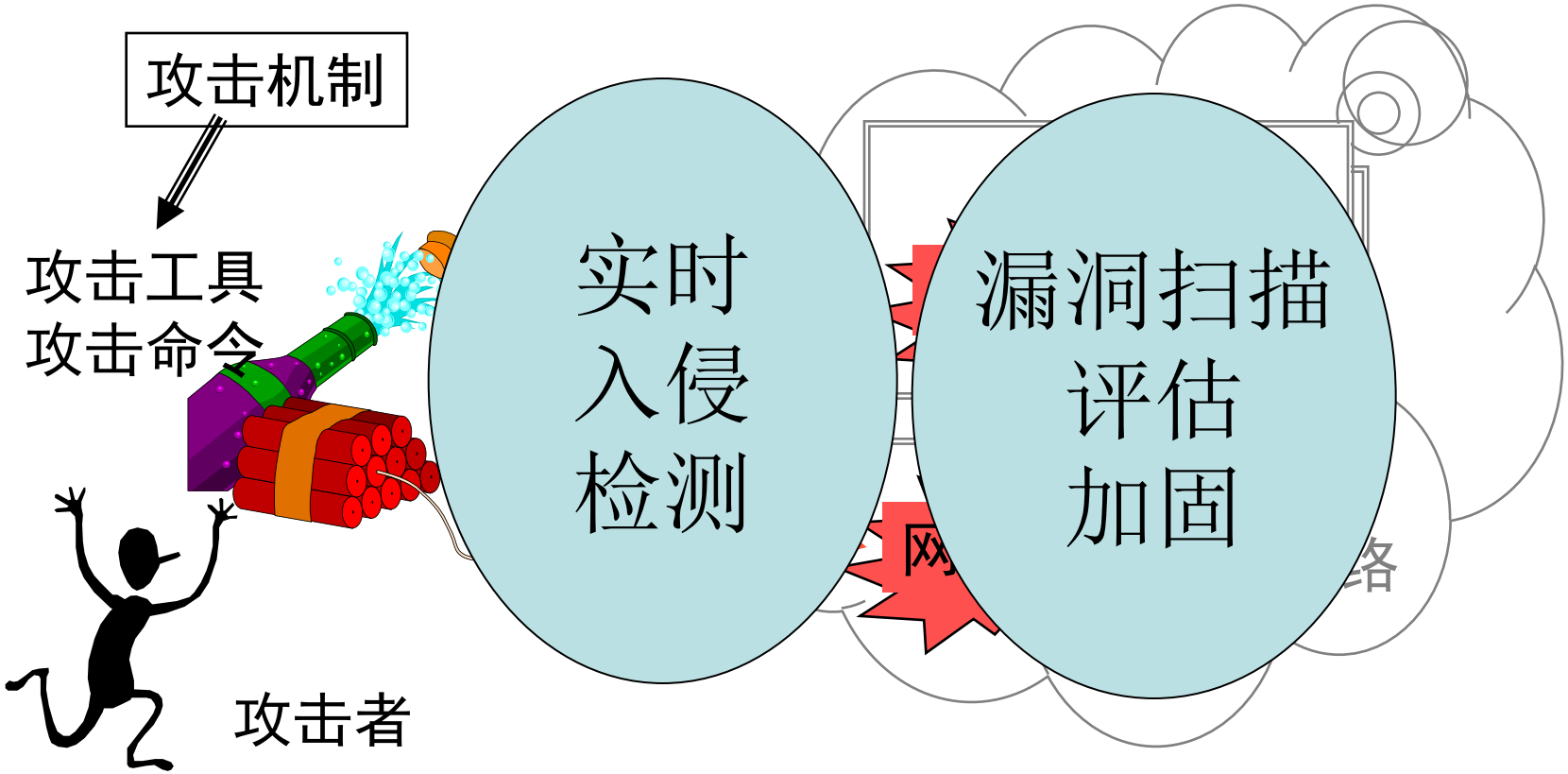
IDS (入侵检测系统)

Chapter 18、20 & supplements

IDS: Intrusion Detection System

- 入侵检测系统介绍
- 入侵检测系统分类
- 入侵检测系统用到的一些技术
- 入侵检测系统的研究和发展

IDS的用途



入侵检测系统的实现过程

- 信息收集，来源：
  - 网络流量
  - 系统日志文件
  - 系统目录和文件的异常变化
  - 程序执行中的异常行为
- 信息分析
  - 模式匹配
  - 统计分析
  - 完整性分析，往往用于事后分析

入侵检测系统的种类

- 基于主机
  - 安全操作系统必须具备一定的审计功能，并记录相应的安全性日志
- 基于网络
  - IDS可以放在防火墙或者网关的后面，以网络嗅探器的形式捕获所有的对内对外的数据包
- 基于内核
  - 从操作系统的内核接收数据，比如LIDS
- 基于应用
  - 从正在运行的应用程序中收集数据

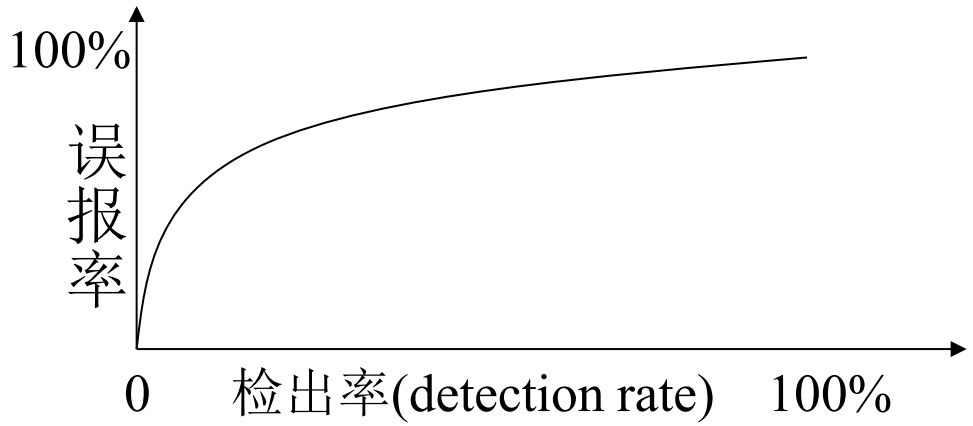
IDS的技术

- 异常检测(anomaly detection)
  - 也称为基于行为的检测
  - 首先建立起用户的正常使用模式，即知识库
  - 标识出不符合正常模式的行为活动

- 误用检测(misuse detection)
  - 也称为基于特征的检测
  - 建立起已知攻击的知识库
  - 判别当前行为活动是否符合已知的攻击模式

IDS的两个指标

- 漏报率
  - 指攻击事件没有被IDS检测到
- 误报率(false alarm rate)
  - 把正常事件识别为攻击并报警
  - 误报率与检出率成正比例关系



基于网络的IDS系统

- 收集网络流量数据
  - 利用sniff技术
  - 把IDS配置在合理的流量集中点上，比如与防火墙或者网关配置在一个子网中
- 利用某些识别技术
  - 基于模式匹配的专家系统
  - 基于异常行为分析的检测手段

一个轻量的网络IDS: snort

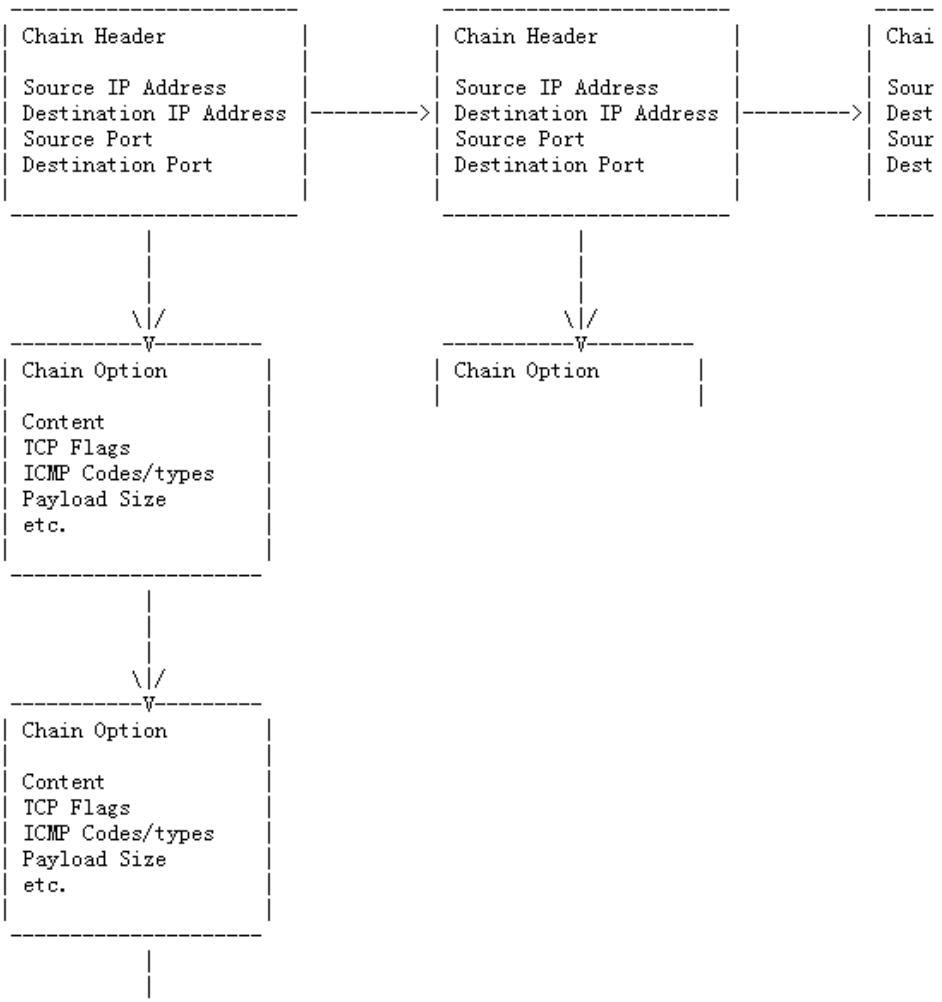
- 是一个基于简单模式匹配的IDS
- 源码开放，跨平台(C语言编写，可移植性好)
- 利用libpcap作为捕获数据包的工具
- 特点
  - 设计原则：性能、简单、灵活
  - 包含三个子系统：网络包的解析器、检测引擎、日志和报警子系统
  - 内置了一套插件子系统，作为系统扩展的手段
  - 模式特征链——规则链
  - 命令行方式运行，也可以用作一个sniffer工具

网络数据包解析

- 结合网络协议栈的结构来设计
- Snort支持链路层和TCP/IP的协议定义
  - 每一层上的数据包都对应一个函数
  - 按照协议层次的顺序依次调用就可以得到各个层上的数据包头
    - 从链路层，到传输层，直到应用层
  - 在解析的过程中，性能非常关键，在每一层传递过程中，只传递指针，不传实际的数据
  - 支持链路层：以太网、令牌网、FDDI

Snort规则链处理过程

- 二维链表结构
- 匹配过程
  - 首先匹配到适当的Chain Header
  - 然后，匹配到适当的Chain Option
  - 最后，满足条件的第一个规则指示相应的动作



Snort: 日志和报警子系统

- 当匹配到特定的规则之后，检测引擎会触发相应的动作
- 日志记录动作，三种格式：
  - 解码之后的二进制数据包
  - 文本形式的IP结构
  - Tcpdump格式
- 如果考虑性能的话，应选择tcpdump格式，或者关闭logging功能
- 报警动作，包括
  - Syslog
  - 记录到alert文本文件中
  - 发送WinPopup消息

关于snort的规则

- Snort的规则比较简单
  - 规则结构：
    - 规则头： alert tcp !10.1.1.0/24 any -> 10.1.1.0/24 any
    - 规则选项： (flags: SF; msg: “SYN-FIN Scan”;)
  - 针对已经发现的攻击类型，都可以编写出适当的规则来
- 规则与性能的关系
  - 先后的顺序
  - Content option的讲究
    - 许多cgi攻击和缓冲区溢出攻击都需要content option
- 现有大量的规则可供利用

Snort规则示例

- 规则示例

```
log tcp any any -> 10.1.1.0/24 79
```

```
alert tcp any any -> 10.1.1.0/24 80 (content: "/cgi-bin/phf"; msg: "PHF probe!";)
```

```
alert tcp any any -> 10.1.1.0/24 6000:6010 (msg: "X traffic";)
```

```
alert tcp !10.1.1.0/24 any -> 10.1.1.0/24 6000:6010 (msg: "X traffic";)
```

Option类型

1. content: Search the packet payl
2. flags: Test the TCP flags for s
3. ttl: Check the IP header's time
4. itype: Match on the ICMP type f
5. icode: Match on the ICMP code f
6. minfrag: Set the threshold valu
7. id: Test the IP header for the
8. ack: Look for a specific TCP he
9. seq: Log for a specific TCP hea
10. logto: Log packets matching th
11. dsize: Match on the size of th
12. offset: Modifier for the conte
    packet payload to begi
13. depth: Modifier for the conter
    from the start positior
14. msg: Sets the message to be se

关于snort

- 开放性
  - 源码开放，最新规则库的开放
- 作为商业IDS的有机补充
  - 特别是对于最新攻击模式的知识共享
- Snort的部署
  - 作为分布式IDS的节点
  - 为高级的IDS提供基本的事件报告
- 发展
  - 数据库的支持
  - 互操作性，规则库的标准化
  - 二进制插件的支持
  - 预处理器模块：TCP流重组、统计分析，等
  - ……

异常检测的网络IDS

- 基于规则和特征匹配的NIDS的缺点
  - 对于新的攻击不能正确识别
  - 人工提取特征，把攻击转换成规则，加入到规则库中
- 异常检测的NIDS可以有一定的自适应能力
  - 利用网络系统的已知流量模式进行学习，把正常流量模式的知识学习到IDS中
  - 当出现新的攻击时，根据异常行为来识别
  - 并且，对于新的攻击以及异常的模式可以反馈到IDS系统中

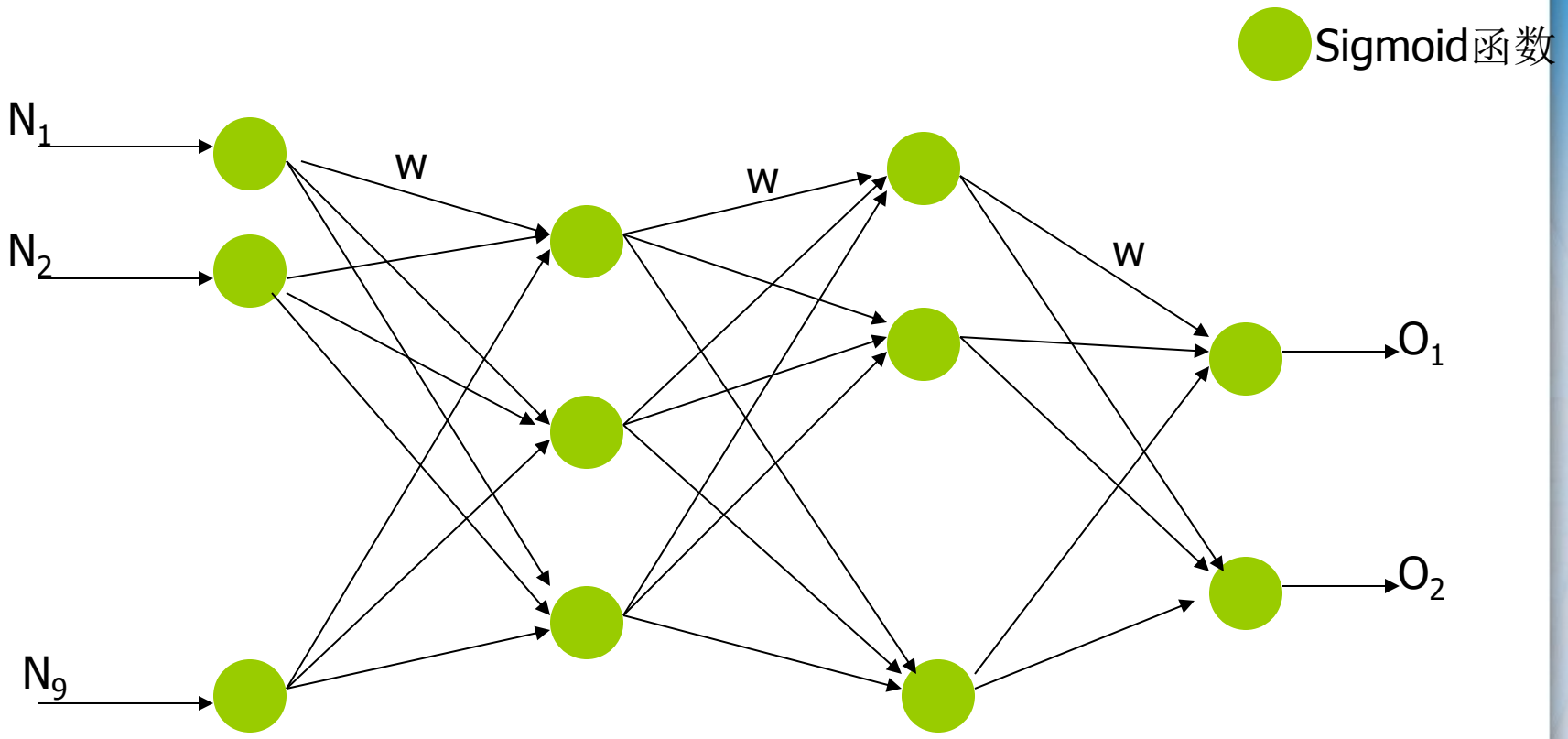
人工神经网络(ANN)用于异常检测

- ANN有比较好的非线性分析能力
- 一定程度上可以代替统计检测技术
- 通过历史数据学习用户的典型特征
- 难点：
  - 采集好的学习样本并量化表达
  - 如何获得自适应？
- 两个简单例子
  - MLP网络用于入侵检测
  - CMAC用于拒绝服务检测

MLP网络用于入侵检测

- 可以认为是误用检测，但是它对未知攻击有一定的适应能力
- 网络描述
  - 多层感知器结构，四层全连接
  - 9个输入
  - 2个输出
- 实验情况
  - 利用网络工具采集数据，包含正常网络行为和模拟攻击行为
  - 需要足够的样本用于学习，并且其中包含模拟攻击样本

MLP网络结构



MLP网络

- 9个输入
  - Protocol ID
  - Source Port
  - Destination Port
  - Source Address
  - Destination Address
  - ICMP Type
  - ICMP Code
  - Raw Data
  - Raw Data Length

- 2个输入
  - (0.0，1.0)表示没有攻击
  - (1.0，0.0)表示有攻击
- 神经元函数
  - sigmoid：1/(1+exp(-x))

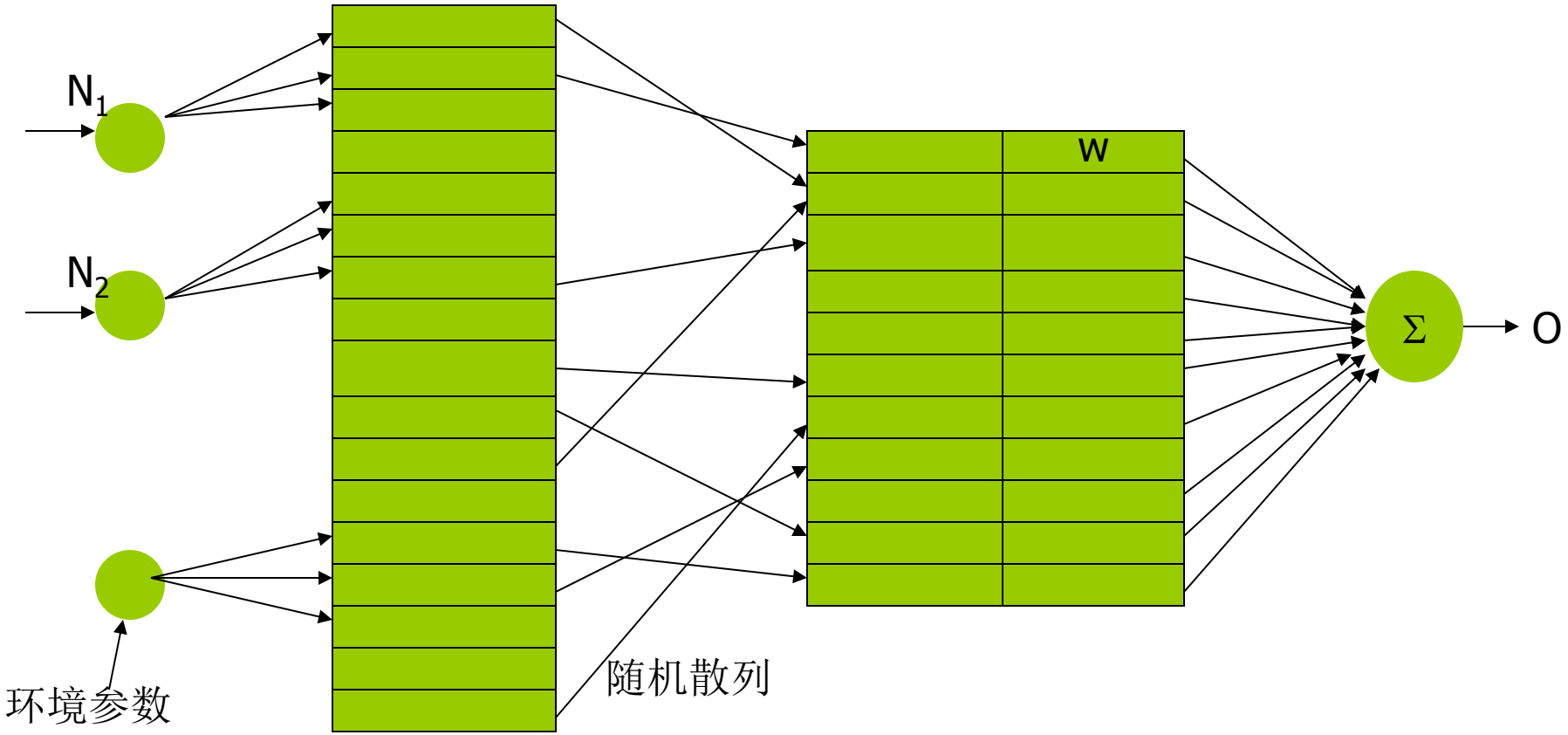
MLP网络的结果分析

- 实验室学习的结果比较好
  - 误差的均方非常小
  - 能够识别出样本中所有的攻击行为
- 只是说明了ANN用于ID的潜在的能力
  - 考虑其他的网络结构
  - 考虑动态的自适应能力

CMAC网络用于自动对抗拒绝服务

- CMAC(Cerebellar Model Articulation Controller)
  - 1975年Albus提出
  - 主要用于控制领域
- 网络模型
  - 三层前向网络
- 优势
  - 具有聚类的作用
  - 容易获得自适应能力

CMAC模型



CMAC用于检测DOS攻击

- CMAC的输入和输出
  - 环境反馈s，范围[0.0, 1.0]
    - 0.0表示系统已经停止，拒绝服务
    - 1.0表示系统正常
  - 输出O，范围[0.0, 1.0]
    - 0.0表示没有攻击
    - 1.0表示攻击
  - 接受正常数据时，输出非常小
- CMAC的学习和自适应调整
  - 学习
    - $w_{i+1} = w_i + b(O_d - O_a)$
    - $w_{i+1} = w_i + b((1-s) - O_a)$
    - $w_{i+1} = w_i + (1-s)((1-s) - O_a)$

CMAC对于拒绝服务的模拟结果分析

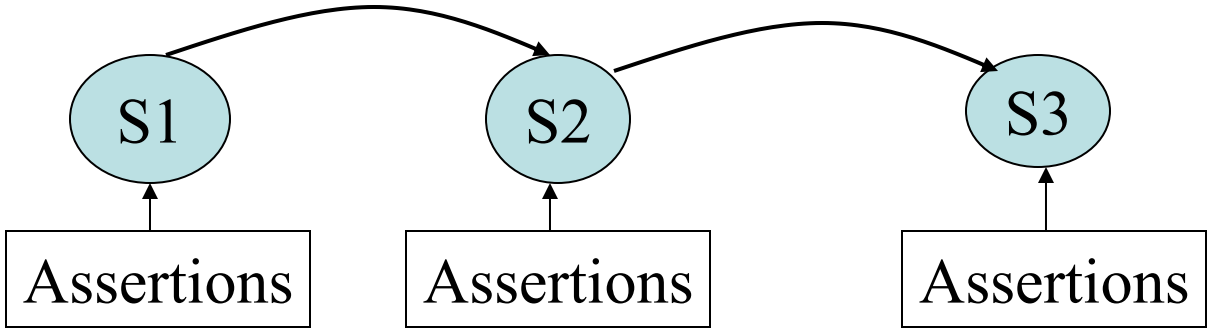
- 学习速度快
- On-line学习攻击模式
- 能够精确地识别以前碰到过的攻击模式
- 对于同类型的攻击，具有较好的适应性
  - 在训练了ping flood之后，识别UDP packet storm攻击的错误为2.2%
- 对状态的分析能力
  - 如果一种网络行为模式在初始时候被识别的概率为75%，随着系统状态的进一步退化，同样的行为模式被识别的概率应该增加
  - 系统的反馈用来增强识别攻击的可能性

基于主机的IDS系统

- 信息收集
  - 系统日志
  - 系统状态信息
  - 特点：OS相关
- 常用的分析技术
  - 统计分析
  - 状态转移分析
  - 关联分析
  - ……

STAT

- STAT: A state transition analysis tool for intrusion detection
  - 由美国加州大学Santa Barbaba分校开发
  - 从初始状态到入侵状态的转移过程
  - 用有限状态机来表示入侵过程
    - 初始状态指入侵发生之前的状态
    - 入侵状态指入侵发生之后系统所处的状态
    - 系统状态通常用系统属性或者用户权限来描述
    - 用户的行为和动作导致系统状态的转变



STAT的优缺点

- 优点：
  - 状态转移图提供了一种针对入侵渗透模式的直观的、高层次的、与审计记录无关的表示方法
  - 用状态转移法，可以描述出构成特定攻击模式的特征行为序列
  - 状态转移图给出了保证攻击成功的特征行为的最小子集，从而使得检测系统可以适应相同入侵模式的不同表现形式
  - 可使攻击行为在到达入侵状态之前就被检测到，从而采取措施阻止攻击行为
  - 可以检测协同攻击和慢速攻击
- 缺点：
  - 状态(assertions)和特征行为需要手工编码
  - Assertions和特征用于表达复杂细致的入侵模式时可能无法表达
  - STAT只是一个框架，需要具体的实现或者与其他系统协同工作
  - STAT的速度相对比较慢

STAT——USTAT

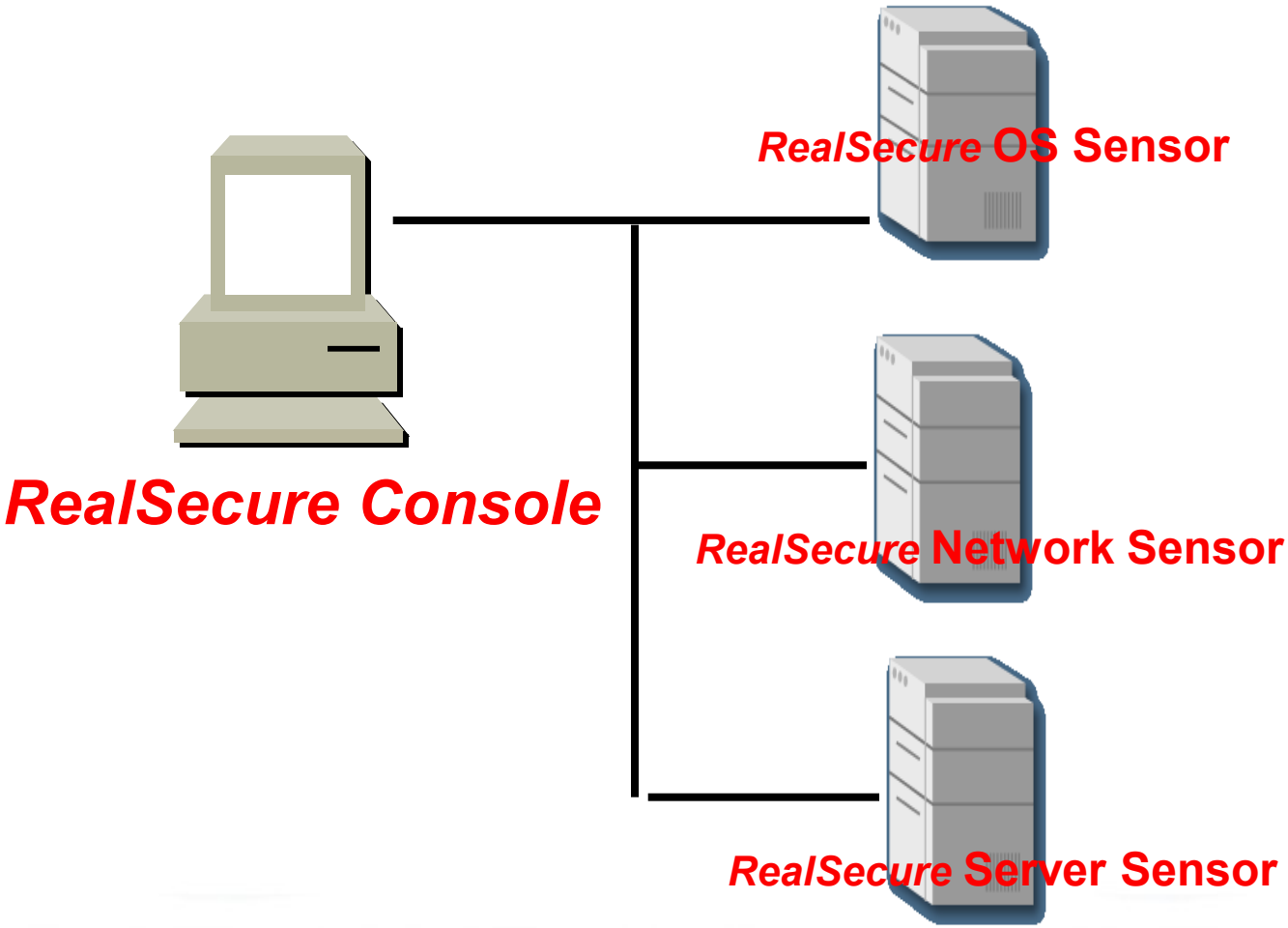
- USTAT：用于UNIX系统的STAT实现
- 包括四部分
  - 预处理器：对数据进行过滤，并转换为与系统日志文件无关的格式
  - 知识库：包括状态描述库和规则库。规则库用于存放已知攻击类型所对应的状态转移规则；状态描述库存放系统在遭受不同类型攻击下所出现的状态
  - 推理引擎：根据预处理器给出的信息和状态描述库中定义的系统状态，判断状态的变化并更新状态信息。一旦发现可疑的安全威胁，通知决策引擎
  - 决策引擎：将安全事件通知管理员，或者采取预先定义的自动响应措施

基于STAT的IDS

- USTAT
- NSTAT
  - 把USTAT扩展到多台主机，从而可以检测到针对多台共享同一个网络文件系统的主机的攻击
- NetSTAT
  - 是一个基于网络的IDS，分析网络中的流量，找到代表恶意行为的数据包
- WinSTAT
  - 基于主机的IDS，分析Windows NT的事件日志
- WebSTAT
  - 基于应用的IDS，用Apache Web Server的日志作为输入
- AlertSTAT
  - 是一个高层的入侵关联器，以其他检测器的报警作为输入，以便检测出高层次、多步骤的攻击

IDS的部署和结构

- 入侵检测系统的管理和部署
  - 多种IDS的协作
  - 管理平台和sensor
- 基于Agent的IDS系统
  - Purdue大学研制了一种被称为AAFID(Autonomous agents for intrusion detection)的IDS模型
  - SRI的EMERALD(Event Monitoring Enabling Response to Anomalous Live Disturbances)


RealSecure OS Sensor
RealSecure Console
RealSecure Network Sensor
RealSecure Server Sensor

IDS的困难

- 异常检测技术仍然需要研究和发展
- NIDS的部署
  - 交换式网络的普及
    - 有些交换机提供了“把多个端口或者VLAN镜像到单个端口”的能力，用于捕获数据包
- 应用系统的多样性
  - 需要统一的规范交换信息
- IPSec等一些加密或者隧道协议
- ……

IDS与响应和恢复技术

- IDS属于检测的环节，一旦检测到入侵或者攻击，必须尽快地做出响应，以保证信息系统的安全
- IDS的响应机制，可以从基本的管理角度来考虑，也可以从技术角度来实施，包括与其他防护系统的互操作，比如防火墙
- 对于一个企业而言，IDS与DRP(Disaster Recovery Planning)需要一起来制订和实施
- DRP包括
  - 业务影响分析(BIA, Business Impact Analysis)
  - 数据必须定期备份
  - 信息系统的根本目的是提供便利的服务
  - 灾难评估
  - 明确责任人

IDS的研究与发展

- IDS自身的发展
  - 基于异常检测的技术
  - 分布式IDS系统
  - 引入生物学免疫系统的概念
  - 与其他防护系统的集成
- IDS与其他学科
  - IDS与模式识别、人工智能
  - IDS与数据挖掘
  - IDS与神经网络
  - IDS与……
- IDS之间的互操作
  - CIDF: 由美国加州大学Davis分校提出的框架，试图规范一种通用的语言格式和编码方式来表示IDS组件边界传递的数据
  - IDEF: IETF IDWG提出的草案，规范了部分术语的使用，用XML来描述消息格式。

参考资料

- 书
  - 戴英侠等，系统安全与入侵检测，清华大学出版社，2002
- 文章
  - Fulvio Risso and Loris Degioanni, An Architecture for High Performance Network Analysis
- **Web站点**
  - **UNIX/Linux Programmer's Manual**
  - **WinPcap, http://winpcap.polito.it/default.htm**
  - **Libnet, http://www.packetfactory.net/Projects/Libnet/**
  - **STAT, http://www.cs.ucsb.edu/~rsg/STAT**
  - **Snort, http://www.snort.org/**