网络与信息系统安全

课程复习

国家计算机网络入侵防范中心

张玉清

课程内容

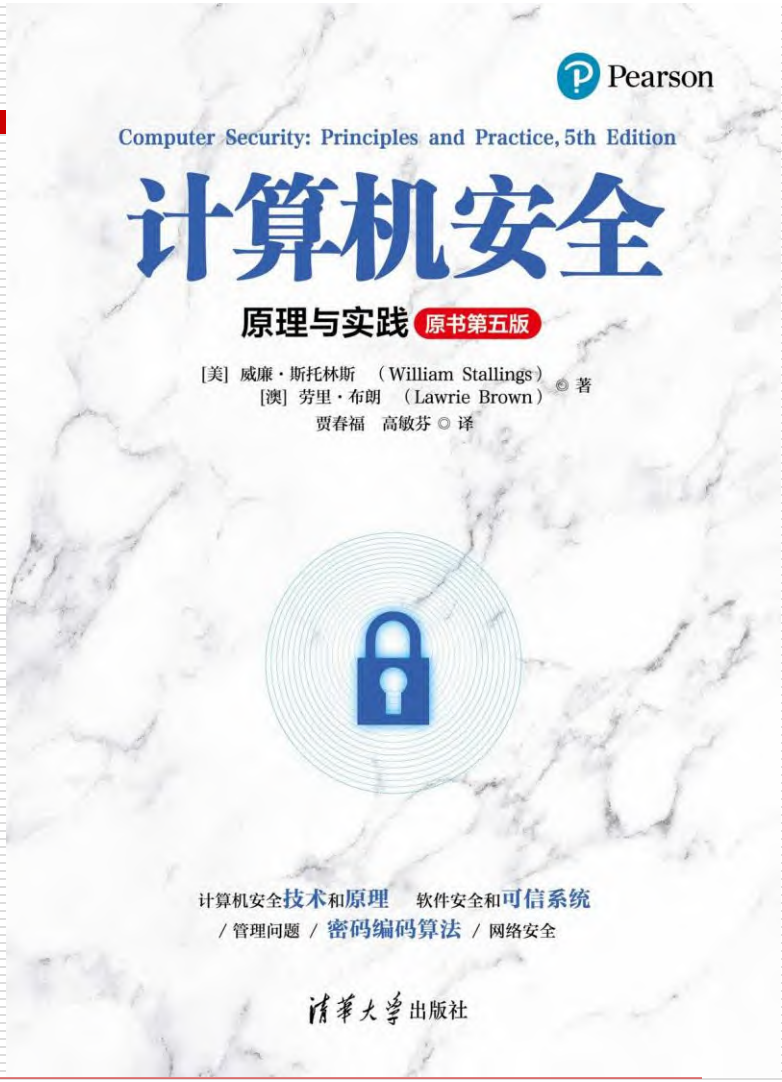
- 掌握网络信息安全的含义，**OSI**、**TCP/IP** 网络安全体系结构。
- 掌握现代加密技术的基本原理，对称加密算法、不对称加密算法。
- 掌握操作系统基本安全问题，理解自主访问控制与强制访问控制。
- 掌握网络防火墙的基本概念，了解防火墙的类型，掌握防火墙安全体系结构。
- 掌握数字签名原理、种类与方法，鉴别技术、数字凭证的方法。
- 掌握计算机病毒与网络安全，掌握蠕虫病毒的识别与防范，计算机病毒的防范与检测。
- 了解网络黑客及其攻击技术，掌握黑客的基本防范技术。

课程安排

- 课时：**40**
- 学分：**2**
- 内容：课堂讲授 + 实验（大报告）
- 考核方式：
  - 笔试（60分，开卷）
  - 实验（大报告）（40分）

教学参考资料

- 计算机安全：原理与实践（原书 第五版）
- 作者：[美]William Stallings
  [澳]Lawrie Brown
- 译者：贾春福 高敏芬
- 清华大学出版社，2024



第一章 概述

- 1.1 计算机安全的概念
- 1.2 威胁、攻击和资产
- 1.3 安全功能要求
- 1.4 基本安全设计原则
- 1.5 攻击面和攻击树
- 1.6 计算机安全策略
- 1.7 标准

三个基本问题

Q1 我们需要保护什么样的资产？

Q2 这些资产是如何受到威胁的？

Q3 我们可以做些什么来应对这些威胁？

举个例子

- 假定我们的目标是个网站，那么网页中的数据、服务器资源、后台用户数据等等一系列资源都是我们想要保护的资产（Q1）。
- 从web安全的角度上，敌手（adversary）会从中间件到搭建平台，从CMS （Web内容管理）到操作系统详尽的对站点进行信息收集寻找漏洞加以利用进行入侵，敌手的攻击以及潜在的漏洞都是站点中资产所受的威胁（Q2）。
- 从管理人员角度上，他们则会思考如何预防入侵，如何在入侵发生后使得资源损失最小化（Q3）

1.1.3 计算机安全面临的挑战

- 对于攻击者，主要优势在于他只需要找到一个安全弱点或漏洞；而管理者必须找到且消除所有的安全弱点才能得到真正的安全。
- 对于部分用户和系统管理者，有种自然倾向：在安全保障失效之前，很少能看到安全投入所带来的好处。
- 安全要求定期甚至持续地对系统进行监视，但是在目前注重时效、超负荷运转的系统环境中很难做到这一点。
- 安全性通常还是事后考虑的问题——在系统设计完成后才加入系统，而没有作为设计过程中的一个有机组成部分来看待。

补充：安全的概念

“如果把一封信锁在保险柜中，把保险柜藏起来，然后告诉你去看这封信，这并不是安全，而是隐藏；相反，如果把一封信锁在保险柜中，然后把保险柜及其设计规范和许多同样的保险柜给你，以便你和世界上最好的开保险柜的专家能够研究锁的装置，而你还是无法打开保险柜去读这封信，这才是安全…”

-Bruce Schneier

1.1.4 一个计算机安全模型

资产（asset）是用户或者管理员希望保护的对象，其大致可分为硬件、软件、数据、通信设施及网络四类。

- **硬件：**计算机系统以及其他用于数据存储、处理和通信的设备
- **软件：**操作系统、系统实用程序、应用程序
- **数据：**文件、数据库以及口令等
- **通信设施和网络：**网络中的通信设备，如链路交换机，集线器，路由器等

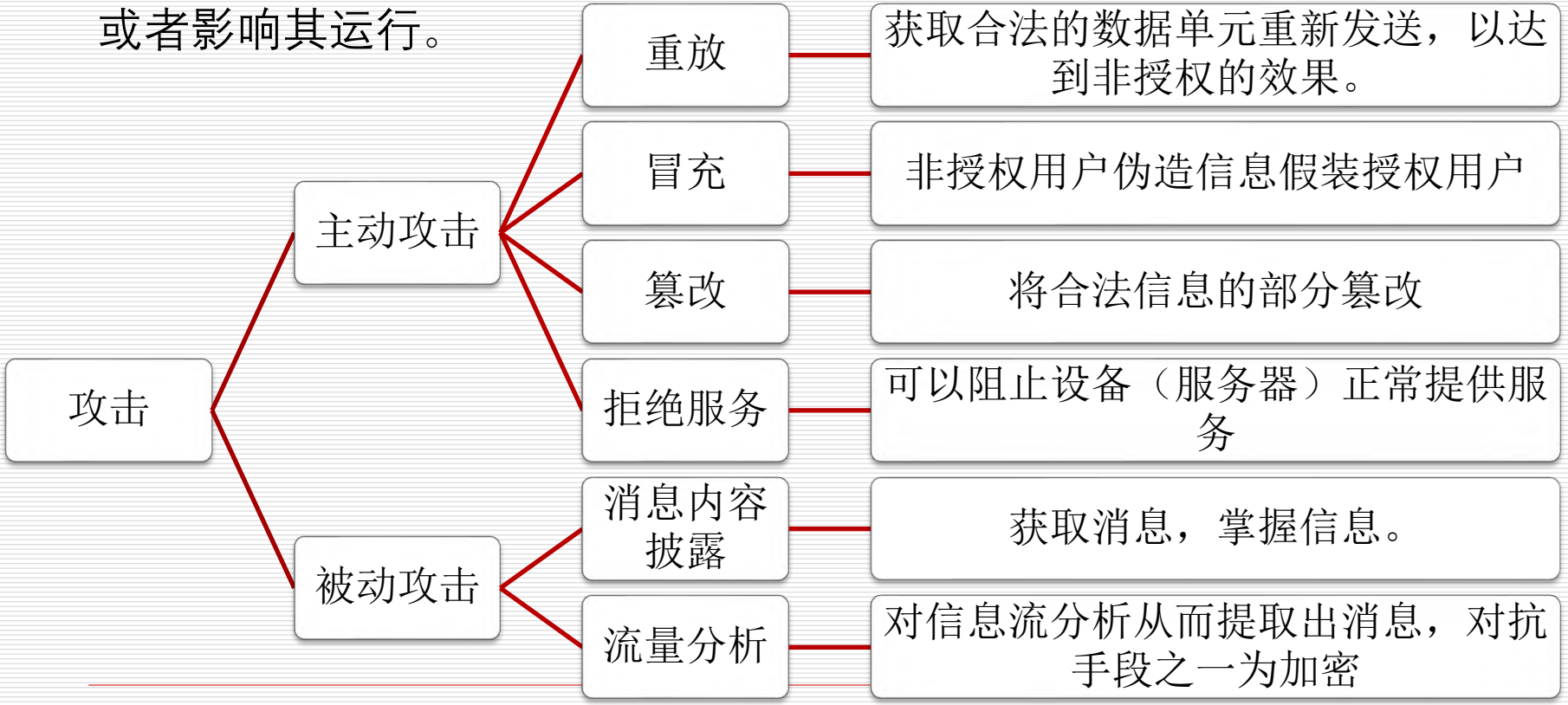
漏洞的概念

- 安全漏洞：是指信息系统在设计、实现或者运行管理过程中存在的缺陷或不足，从而使攻击者能够在未授权的情况下利用这些缺陷破坏系统的安全策略。
- 存在于信息系统的需求、设计、实现、配置、运行等环境
- 能够被恶意主体所利用，影响信息系统及其服务的正常运行
- 网络攻防的核心：
  - 攻击----漏洞利用
  - 防御----漏洞修复

Note：
*这里对于攻防的理解是比较简单和直接，当然攻防还包括除漏洞以外的技术和手段。

网络安全攻击类型

网络安全攻击可以划分为**被动攻击和主动攻击**。 被动攻击企图了解或利用系统信息，但不影响系统资源。主动攻击则试图改变系统的资源或者影响其运行。



第二章 密码编码工具

- 2.1 对称加密
- 2.2 消息认证和散列函数
- 2.3 公钥加密
- 2.4 数字签名和密钥管理
- 2.5 随机数和伪随机数
- 2.6 实际应用：存储数据的加密

2.1 对称加密

- 对称加密也称“传统加密”或“单密钥加密”，是 20 世纪 70 年代后期公钥密码产生之前唯一的加密技术，至今仍是使用最广泛的加密算法之一。

- 其基本成分包括
  - 明文（原始可理解的消息和数据）
  - 加密算法（对明文进行代换和变换）
  - 秘密密钥（加密算法的输入，特定代换和变换依赖于它）
  - 密文（算法输出的随机杂乱数据，依赖于明文和密钥）
  - 解密算法（加密算法的逆运算，输入密文和密钥可恢复明文）。

2.1 对称加密

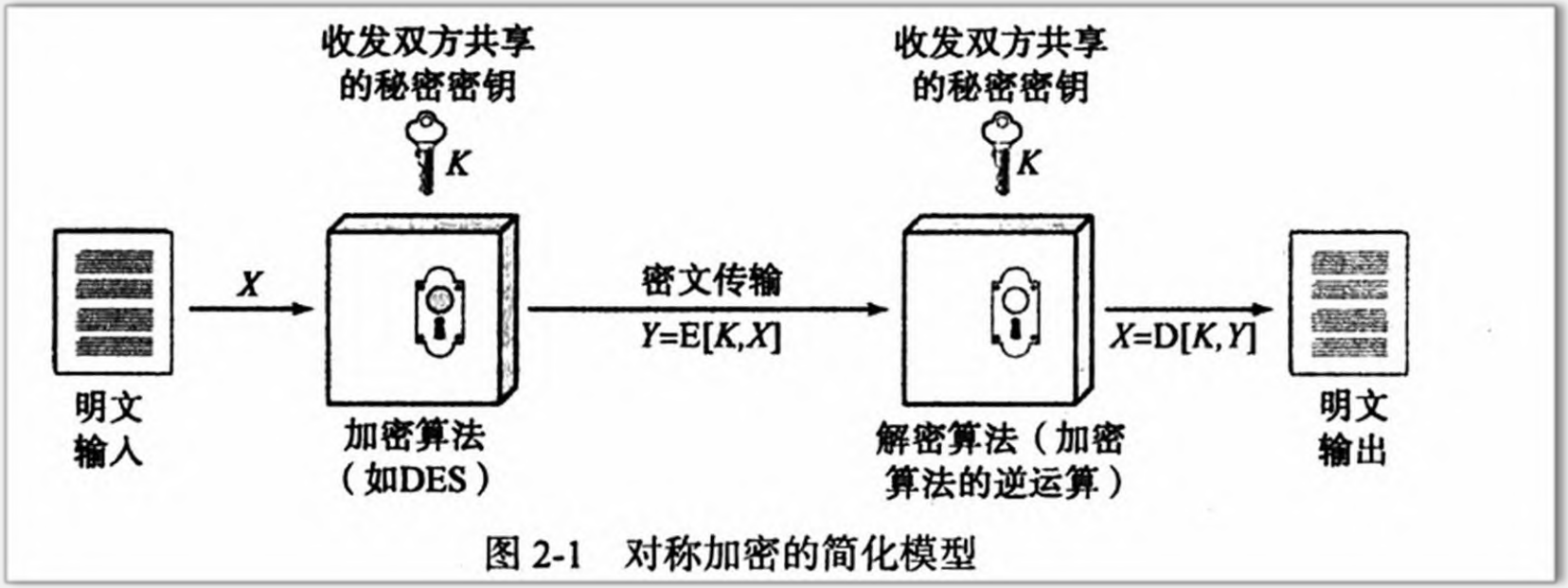


图 2-1　对称加密的简化模型

在对称加密中，加密和解密使用相同的密钥。就好像是一把特殊的“锁”和对应的“钥匙”，发送方用这把“钥匙”（密钥）将明文信息加密成密文，就像是把信息锁进了一个加密的箱子。

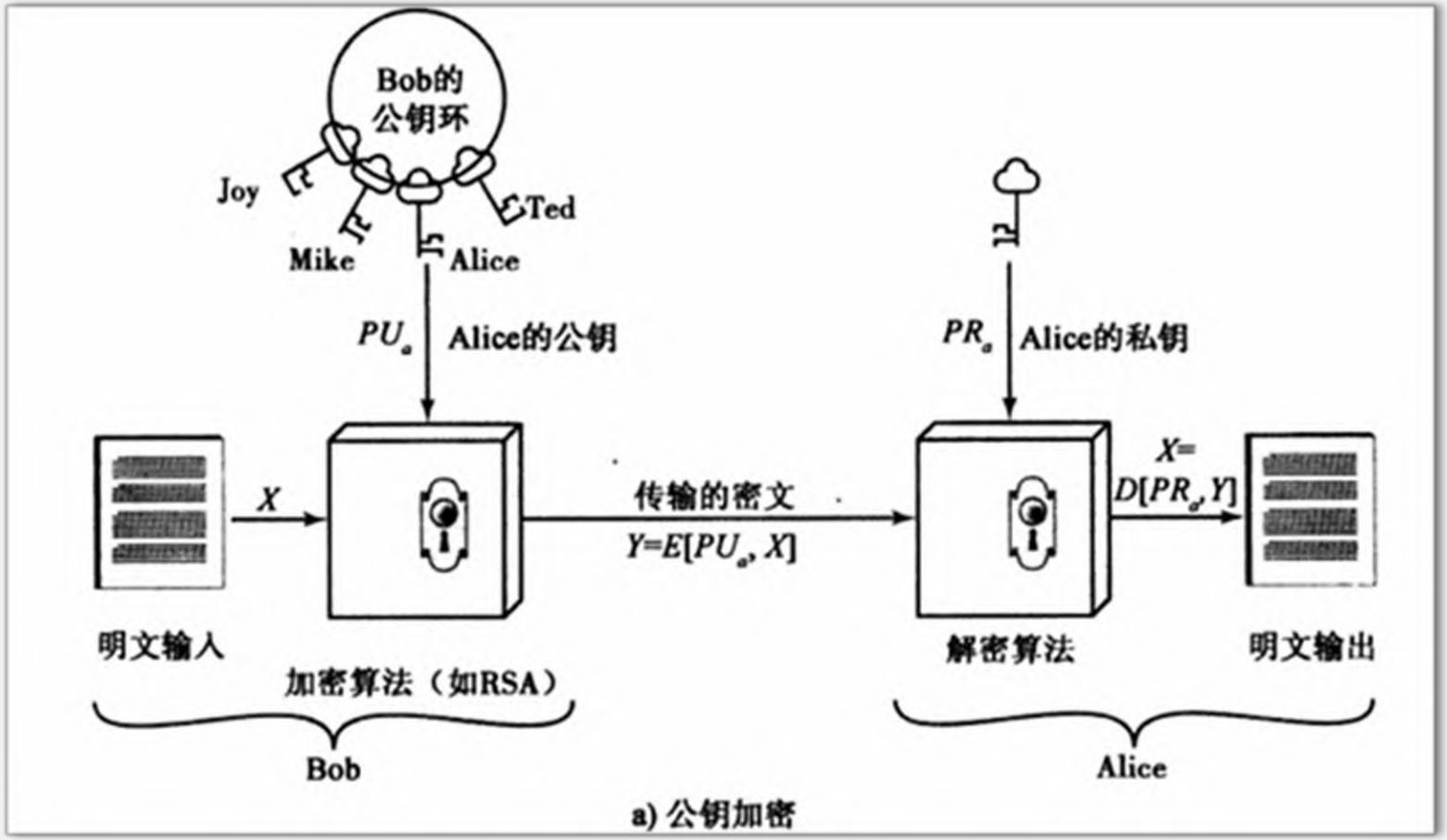
2.2 消息认证和散列函数

- 当消息、文件、文档或其他数据集合是真实的且来自于合法信源，则被称为是可信的。
- 消息认证是一种允许通信者验证所接收或存储的数据是否可信的措施。
- 认证包括两个重要方面：验证消息的内容有没有被篡改和验证信源是否可信。
- 还可以验证消息的时效性（即消息没有被人为地延迟和重放）以及两个实体之间传输的消息流的相对顺序。

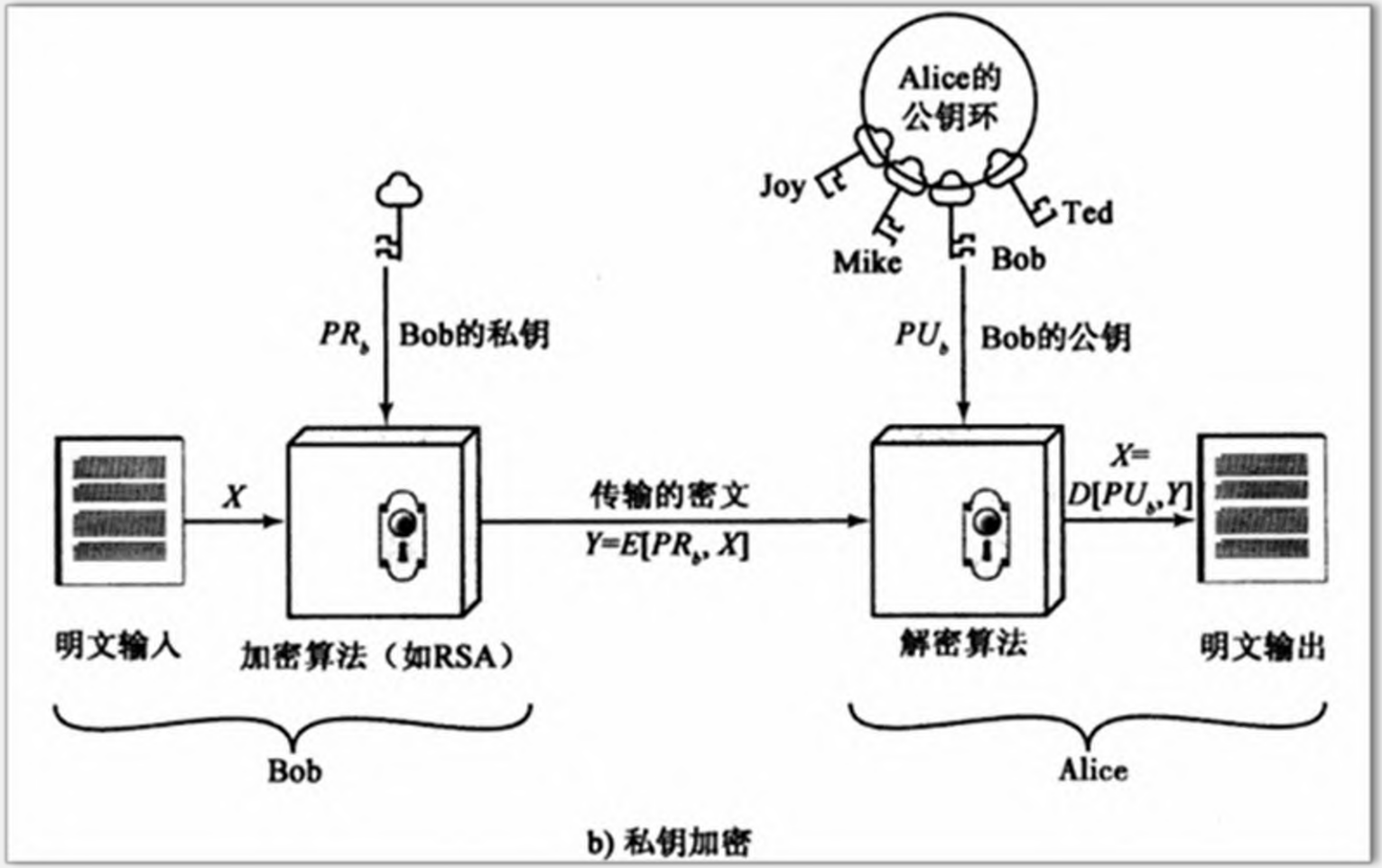
2.3 公钥加密

- 1976年，Diffie和Hellman【DIFF76】首次提出了公钥加密的思想，这是有文字记载的几千年来秘密领域第一次真正革命性的进步。
- 公钥算法基于数学函数，而不像对称加密算法那样是基于位模式的简单操作，更重要的是，公钥密码是非对称的(asymmetric）它使用两个单独的密钥。
- 而对称加密只是用一个密钥。使用两个密钥对于机密性、密钥分发和认证产生了意义深远的影响。
- 公钥对其他使用者来说是公共的。然而私钥只有它的拥有者知道。
- 一般的公钥加密算法依赖于一个加密密钥和一个与之不同但又相关的解密密钥。

2.3 公钥加密


Bob的
公钥环
Joy
Mike
Alice
Ted
PU_a
Alice的公钥
PR_a
Alice的私钥
X
传输的密文
$Y=E[PU_a, X]$
$X=D[PR_a, Y]$
明文输入
加密算法（如RSA）
Bob
解密算法
明文输出
Alice
a) 公钥加密

2.3 公钥加密


Alice的
公钥环
Joy
Mike
Bob
Ted
PR_b Bob的私钥
PU_b Bob的公钥
X
传输的密文
$Y=E[PR_b, X]$
$X=D[PU_b, Y]$
明文输入
加密算法（如RSA）
解密算法
明文输出
Bob
Alice
b) 私钥加密

2.3 公钥密码体制的应用

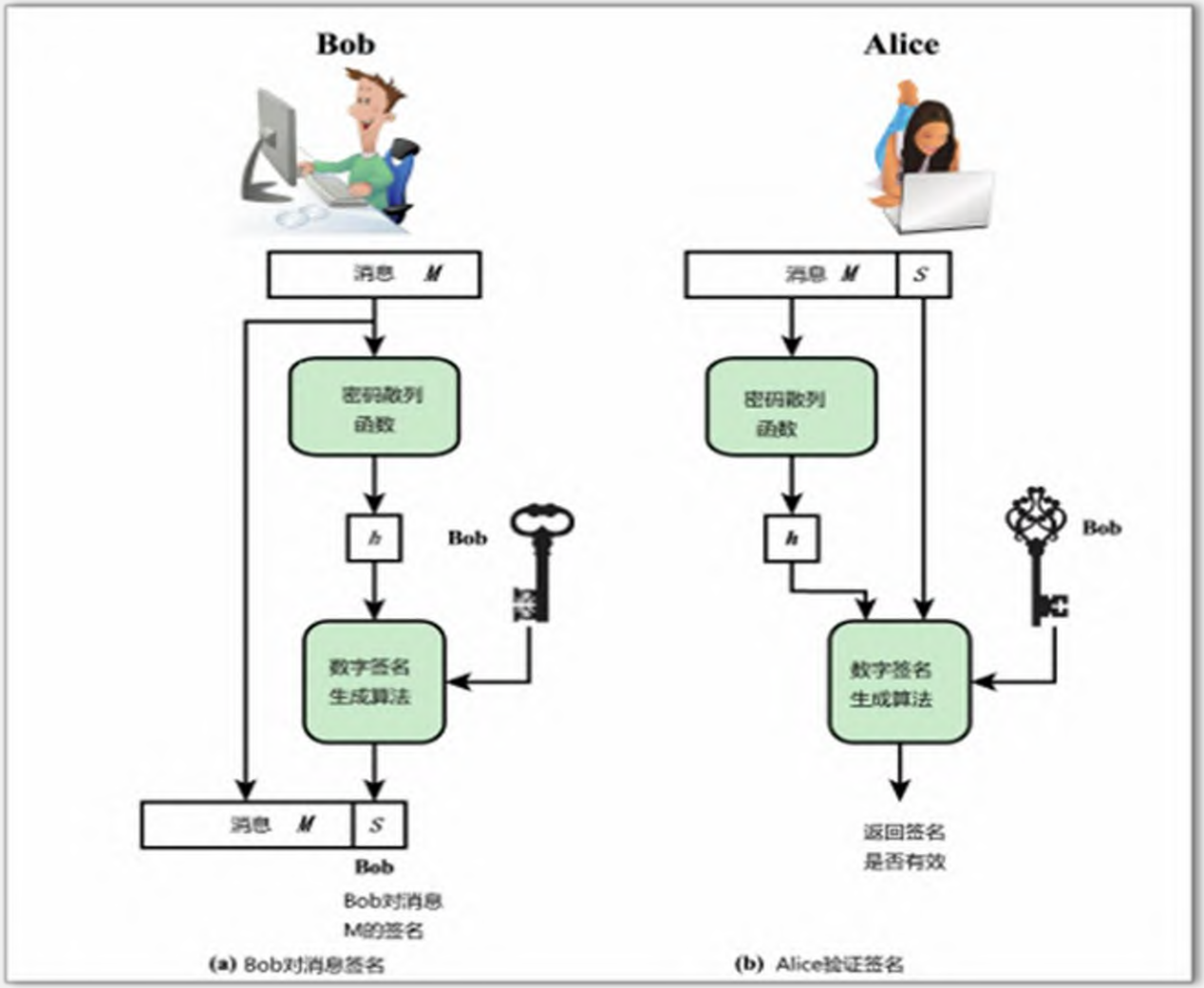
- 公钥密码体制的特点是使用具有两个密钥的密码算法，其中一个密钥是私有的，另一个是公有的。
- 根据不同的应用，发送方可使用其私钥或者接收方的公钥或同时使用两者来实现密码功能。
- 一般来讲，可以把公钥密码体制的应用划分为三类：数字签名、对称密钥分发和秘密密钥加密。

算　法	数字签名	对称密钥的分发	秘密密钥加密
RSA	是	是	是
Diffie-Hellman	否	是	否
DSS	是	否	否
椭圆曲线	是	是	是

2.4 数字签名

- NIST FIPS PUB 186-4 定义了数字签名:
  - “数据的加密转换的结果（如果得到适当实施）能够提供一个机制来保证原始认证、数据完整性以及签名的不可依赖性。”
- 数字签名是依赖于数据的位组合格式，由代理根据文件、消息或其他形式的数据块生成
- FIPS 186-4 指定了以下三种数字签名算法:
  1. 数字签名算法(DSA)
  2. RSA 数字签名算法
  3. 椭圆曲线数字签名算法(ECDSA)

2.4 数字签名


Bob
消息 M
密码散列函数
h
Bob
数字签名生成算法
消息 M
S
Bob
Bob对消息M的签名
(a) Bob对消息签名
Alice
消息 M
S
密码散列函数
h
Bob
数字签名生成算法
返回签名是否有效
(b) Alice验证签名

2.5 随机与伪随机

- 密码应用大多使用算法来生成随机数。这些算法是确定的，所以产生的序列并非是统计随机的。不过要是算法好的话，产生的序列可以经受住随机性检测。这样的数一般被称为伪随机数。
- 在大多数情况下，伪随机数在实际应用中会表现得想真随机数一样。尽管“像真随机数一样”这种说法是非常主观的，然而伪随机数已被普遍接受。
- 一个真随机数发生器是利用不确定的源来生成真随机数。大部分是通过测量不可预测的自然过程（如电离辐射效应的脉冲检测器、气体排放管和漏电电容器）来实现的。

第三章 用户认证

- 3.1 数字用户认证方法
- 3.2 基于口令的认证
- 3.3 基于令牌的认证
- 3.4 生物特征认证
- 3.5 远程用户认证
- 3.6 用户认证中的安全问题
- 3.7 实际应用和案例学习

用户认证功能

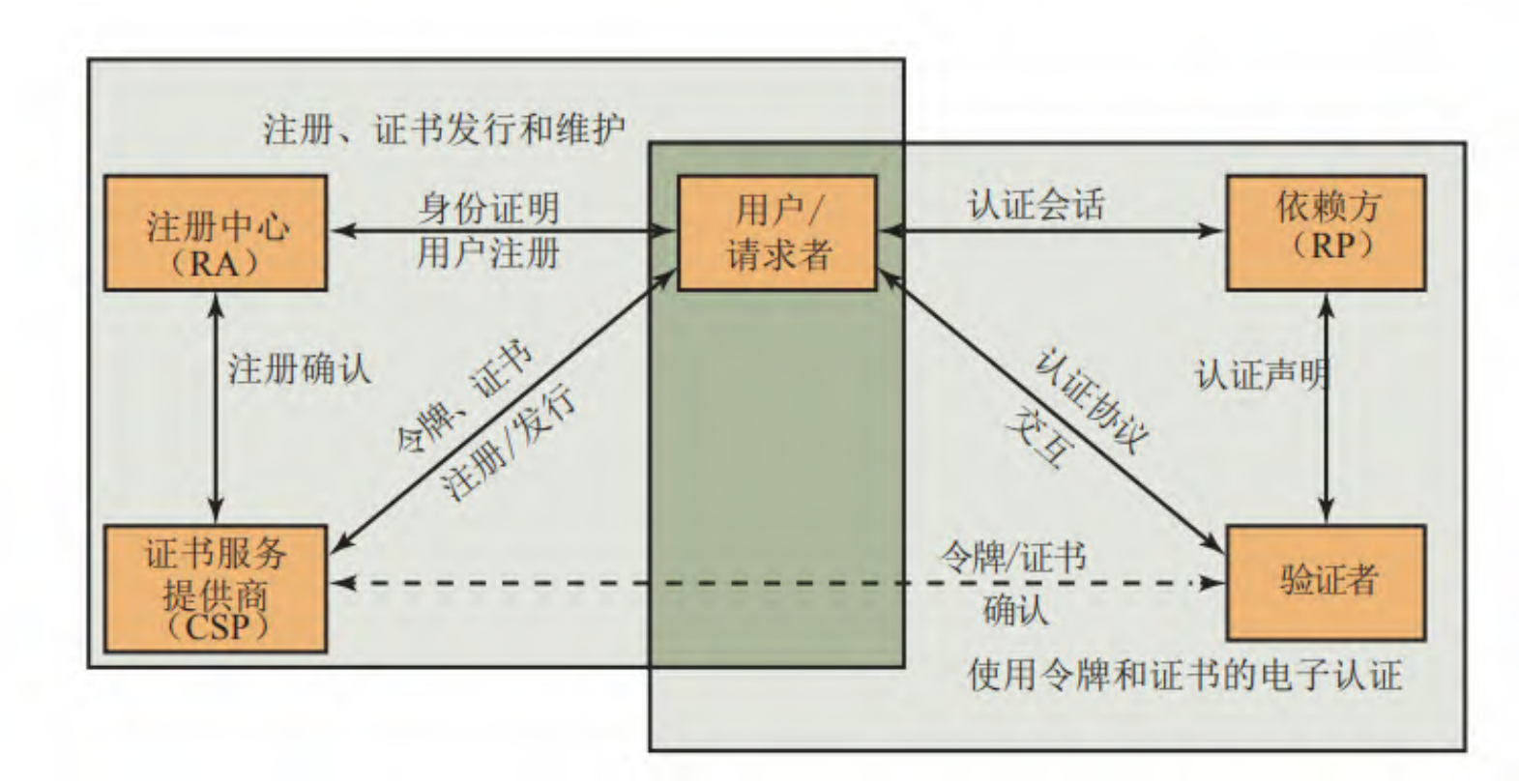
用户认证

用户通过呈现一个凭证（例如用户ID）来识别自身以登录系统

系统通过交换认证信息来核实用户身份

3.1.1 电子用户认证模型

我们参考下图讨论用户身份认证通用模型：



3.1.2 认证方法

验证用户身份的一般方法有四种，它们可以单独使用也可以组合起来使用：

个人知道的信息	• 口令、个人识别码 (PIN) 或预先安排的一组问题的答案
个人拥有的物品	• 电子钥匙卡、智能卡和物理钥匙。 这种类型的身份验证器称为令牌。
个人生理特征（静态生物特征）	• 指纹识别、虹膜识别和人脸识别
个人行为特征（动态生物特征）	• 通过语音模式、笔迹特征和打字节奏进行识别

3.1.3 多因素认证

- 多因素认证（Multifactor Authentication，MFA）：用户需提供两个或多个独立身份验证因素（如所知、所有、所属、所做）的组合，以验证身份。
- 例子：网上银行系统要求用户首先输入口令（用 户所知），然后输入通过短信发送到手机的验证码（用户所有）。
- NIST SP 800-63B 要 求使用多种身份验证方法来获得更高的身份验证保证级别。

基于口令的认证

基于口令的认证对应的是“**个人所知道的信息**”，可以通俗的理解成账号密码。

保证用户id安全性

- 决定用户是否被授权访问系统
- 决定了该用户所拥有的访问权限
- 应用在自主访问控制机制中

第四章 访问控制

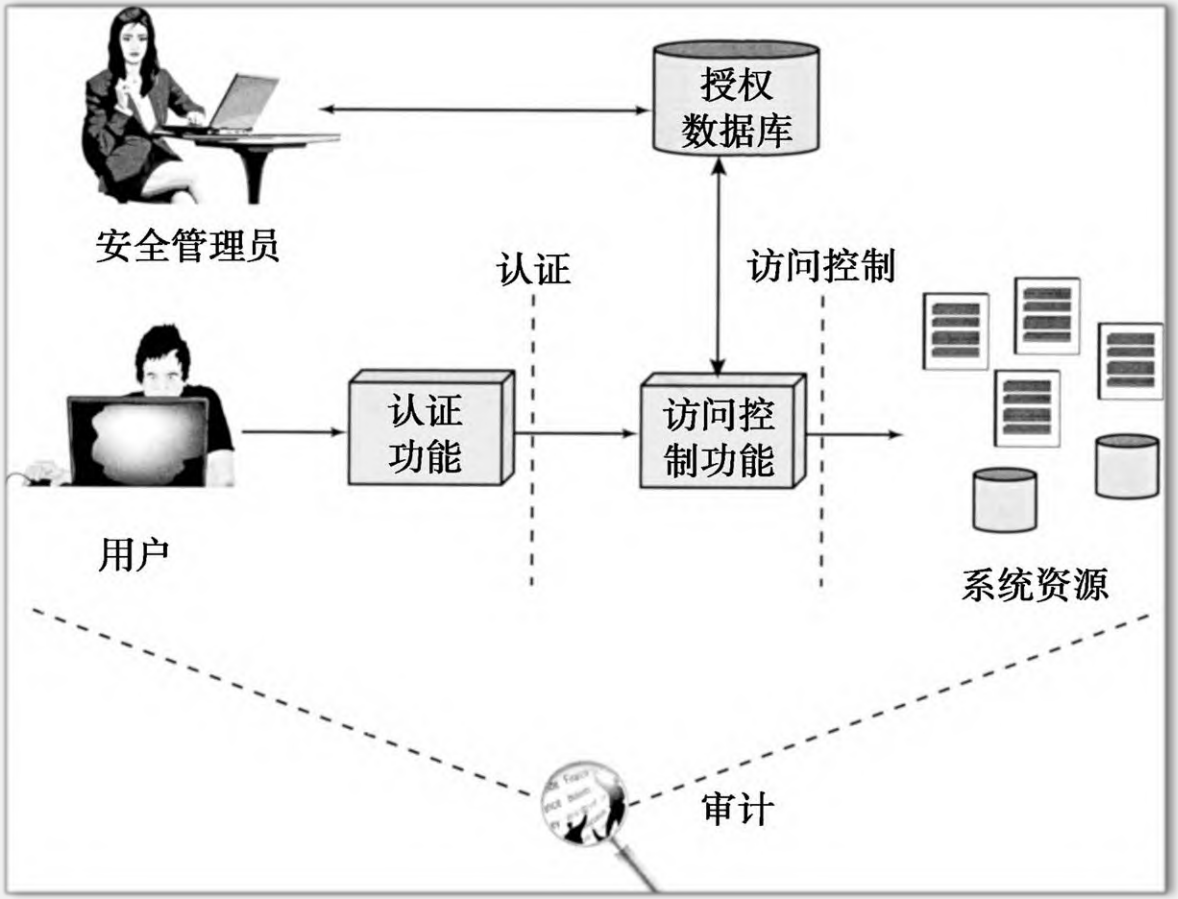
- 4.1 访问控制原理
- 4.2 主体、客体和访问权
- 4.3 自主访问控制
- 4.4 实例：UNIX文件访问控制
- 4.5 基于角色的访问控制
- 4.6 基于属性的访问控制
- 4.7 身份、凭证和访问管理
- 4.8 信任框架
- 4.9 案例学习：银行的RBAC系统

4.1.1 访问控制语境

广义来讲，所有的计算机安全都与访问控制有关。

RFC 4949定义计算机安全如下：

“用来实现和保证计算机系统的安全服务的措施，特别是保证访问控制服务的措施。”



访问控制与其它安全功能的关系

4.1.2 访问控制策略

- 自主访问控制（discretionary access control，DAC）：基于请求者的身份和访问规则（授权）控制访问，规定请求者可以（或不可以）做什么。这种策略被称为“自主的”是因为允许一个实体按其自己的意志授予另一个实体访问某些资源的权限。
- 强制访问控制（mandatory access control，MAC）：通过比较具有安全许可（表明系统实体有资格访问某种资源）的安全标记（表明系统资源的敏感或关键程度）来控制访问。这种策略被称为“强制的”是因为一个具有访问某种资源的许可的实体不能按其自己的意志授予另一个实体访问那种资源的权限。
- 基于角色的访问控制（role-based access control，RBAC）：基于用户在系统中所具有的角色和说明各种角色用户享有哪些访问权的规则来控制访问。
- 基于属性的访问控制（attribute-based access control，ABAC）：基于用户、被访问资源及当前环境条件来控制访问。

4.2 主体、客体和访问权

主体

能够访问客体的实体

三类主体：

所有者；组；世界

客体

外界对其访问受到控制的资源

客体是一个用来包含和/或接收信息的实体

访问权

描述了主体可以访问客体的方式

可包括:

读；写；执行；删除；创建；搜索

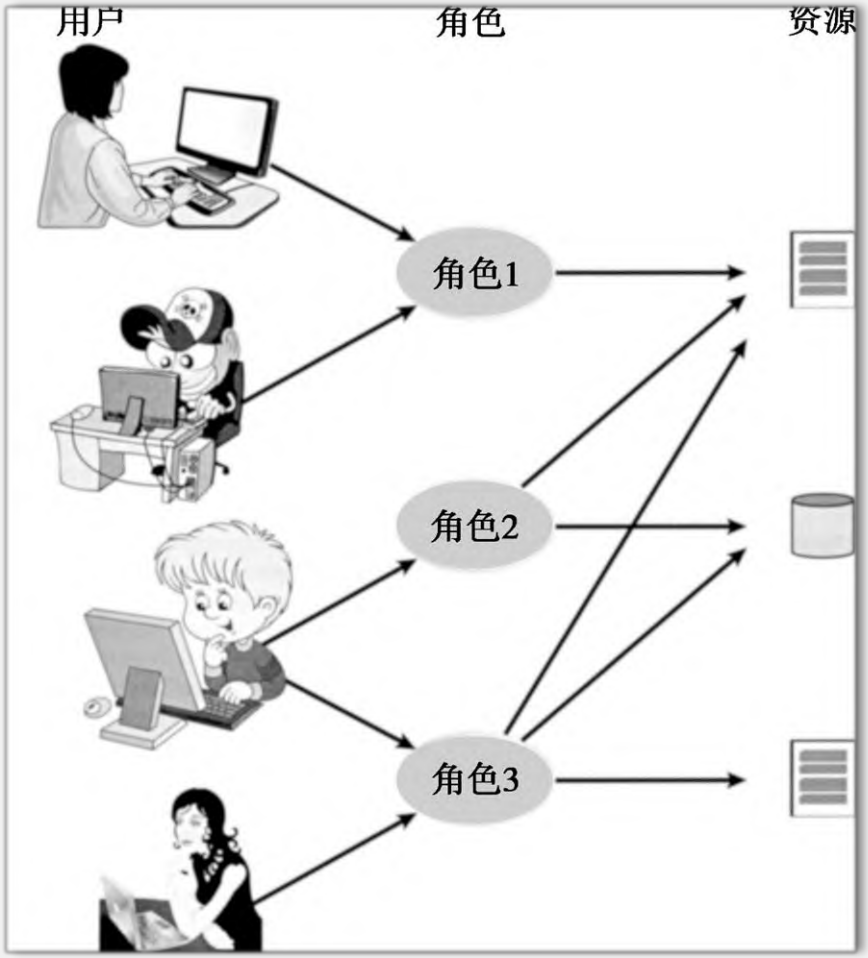
4.3 访问矩阵的一个简单例子

	文件1	文件2	文件3	文件4
用户A	Own Read Write		Own Read Write	
用户B	Read	Own Read Write	Write	Read
用户C	Read Write	Read		Own Read Write

由图可知，用户A拥有文件1和3并具有对这些文件的读、写权限，用户B具有对文件1的读权限，依此类推。

4.5 基于角色的访问控制

- 传统的DAC系统定义了单独的用户和用户组的访问权。与之相反，RBAC基于用户在系统中设定的角色而不是用户的身份。
- 一般地，RBAC模型定义角色为组织中的一项工作职责。RBAC系统给角色而不是给单独的用户分配访问权。
- 用户与角色的关系是多对多的，角色与资源或系统对象的关系也是多对多的



第五章 数据库与数据中心安全

- 5.1 数据库安全需求
- 5.2 数据库管理系统
- 5.3 关系数据库
- 5.4 SQL注入攻击
- 5.5 数据库访问控制
- 5.6 推理
- 5.7 数据库加密
- 5.8 数据中心安全

5.1 数据库安全需求

- 针对数据库的安全问题已经成为整个组织或机构安全策略的重要组成部分，但是数据库安全却始终没有跟上数据库应用发展步伐，有如下原因：
  - DBMS的复杂程度不断增加，配套的安全措施**落后于**新模块开发速度。
  - 数据库系统的**交互协议复杂程度高**，如果想要匹配相应的措施，必须对相关内容非常熟悉。
  - 大多组织机构中，没有技能与岗位相匹配的数据库**管理人员**。
  - 许多企业对于数据的管理可能采用了不同的数据平台，这些不同的数据平台构成了一个**异构环境**，对于这样的数据存储环境的管理难度非常高。
  - 云技术的发展让不少组织选择将数据上云，对相关安全技术和**管理人员**的要求提高。

5.4 SQL注入攻击

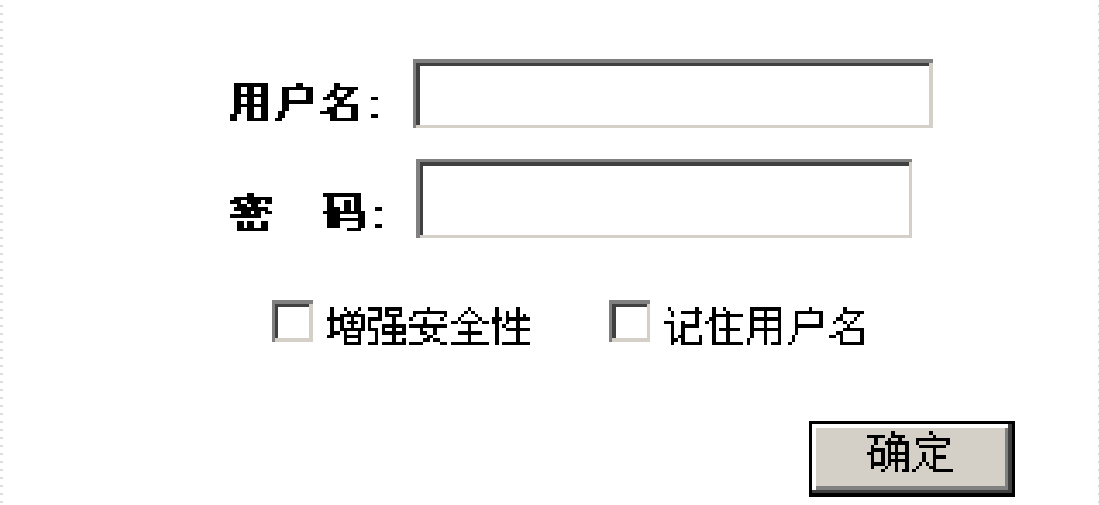
SQL注入（简记为SQLi）攻击

- 最普遍和最危险的基于网络的安全威胁之一
- 一般而言，旨在利用Web应用程序页面的特性
- 向数据库服务器发送恶意SQL命令
- 最常见的攻击目标是从数据库中批量提取数据
- 根据环境的不同，还可以利用SQL注入来：
  - 修改或删除数据
  - 执行任意操作系统命令
  - 发起拒绝服务（DoS）攻击

补充：最简单的SQL注入实例

- 假设这么一个情景，一个网页的后台入口处需要验证用户名和密码，验证程序的SQL语句是这样写：

  Select * from admin where user='TxtBox1.txt' and pass='TxtBox2.txt'



补充：最简单的SQL注入实例(2)

- 如果用户填写的用户名和密码都是: **'abc' or '1'='1'**
- 那么将导致SQL语句是:

  Select * from admin where user='abc' or '1'='1' and pass= ='abc' or '1'='1'
- 这条语句是永真式，那么攻击者就成功登陆了后台。这就是最简单的SQL注入方式。

第六章 恶意软件

- 6.1 恶意软件的类型
- 6.2 高级持续性威胁
- 6.3 传染-感染内容-病毒
- 6.4 传播-漏洞利用-蠕虫
- 6.5 传播-社会工程学-垃圾电子邮件、木马
- 6.6 载荷-系统损坏
- 6.7 载荷-攻击代理-僵尸程序（zombie、bot）
- 6.8 载荷-信息窃取-键盘记录器、网络钓鱼、间谍软件
- 6.9 载荷-隐蔽-后门、rootkit
- 6.10 对抗手段

6.1.1 恶意软件的粗略分类

两大类：

- 基于其向目标传播和感染的方式进行分类
- 基于其工作方式或有效负载进行分类

分类依据：

- 依附于宿主程序（寄生代码，如病毒）
- 独立的、自成一体的程序（蠕虫、特洛伊木马和僵尸程序）
- 不自我复制的恶意软件（特洛伊木马和垃圾邮件）
- 能够自我复制的恶意软件（病毒和蠕虫）

APT的特征

高级	持续性	威胁
• 使用多种的入侵技术和恶意软件，如果有需要的话还会开发定制的恶意软件 • 其中单一的组件在技术上也许不先进，但是每个组件都是针对目标精心选择的	• 攻击者在很长时间内确定针对攻击目标的攻击应用可以最大化攻击成功的几率 • 攻击手段的种类是逐渐递增的，通常是非常隐秘的，直到目标被攻陷	• 针对选定目标的威胁来自于有组织的、有能力的和良好经济支持的攻击者，他们试图攻陷这些目标 • 攻击者的积极参与极大程度提升了自动攻击工具的威胁等级，也增加了成功攻击的可能性

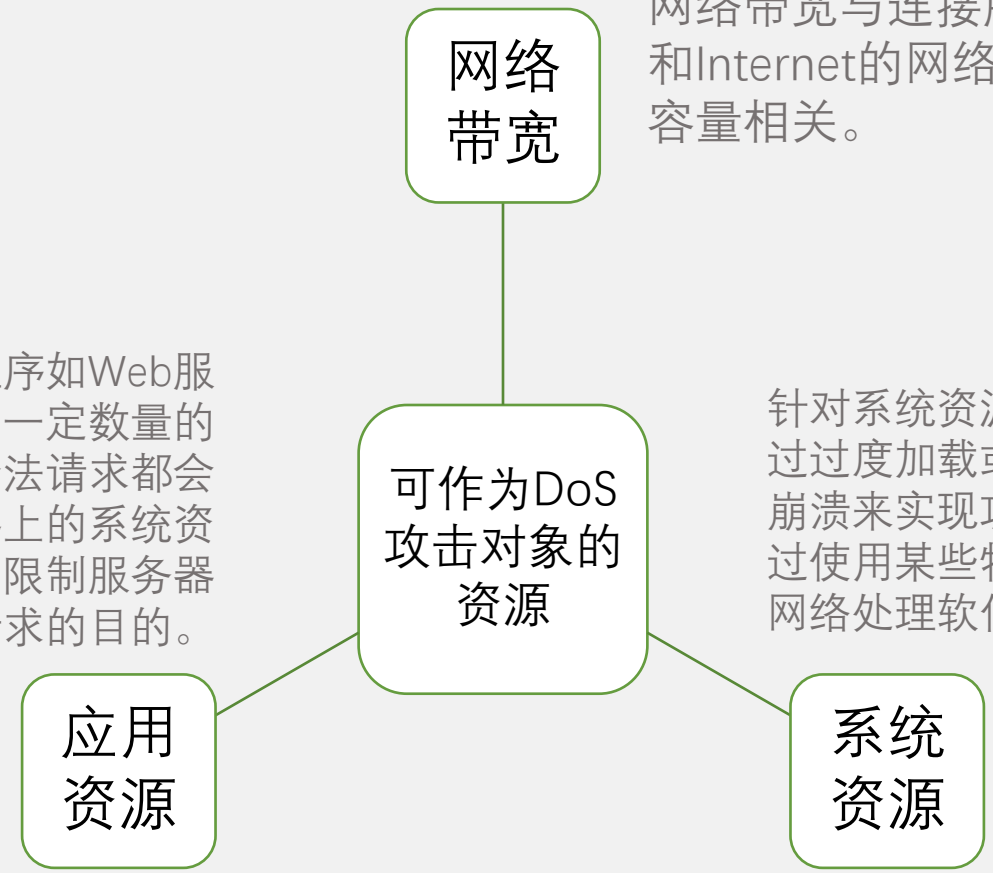
第七章 拒绝服务攻击

- 7.1 拒绝服务攻击
- 7.2 洪泛攻击
- 7.3 分布式拒绝服务攻击
- 7.4 基于应用的带宽攻击
- 7.5 反射攻击与放大攻击
- 7.6 拒绝服务攻击防范
- 7.7 对拒绝服务攻击的响应

7.1.1 拒绝服务攻击的本质

- 拒绝服务（Denial-of-Service，DoS）攻击是一种针对某些服务可用性的攻击。在计算机和通信安全的背景下，DoS攻击一般攻击目标系统的网络服务，通过攻击其网络连接来实现。这种针对服务可用性的攻击不同于其他传统意义上的不可抗力产生的攻击，它是通过造成IT基础设施的损害或毁坏而导致服务能力的丧失。
- NISTSP 800-61计算机安全事故处理指南（NIST Computer Security Incident Handling Guide）[CICH12]中对DoS攻击给出的定义如下：拒绝服务（DoS）是一种通过耗尽CPU、内存、带宽以及磁盘空间等系统资源，来阻止或削弱对网络、系统或应用程序的授权使用的行为。

7.1.1 拒绝服务攻击的本质



7.1.2 经典的拒绝服务攻击

- 对于一个组织，最简单的经典DoS攻击就是洪泛攻击（flooding attack）。
- 洪泛攻击的目标就是占据所有到目标组织的网络连接的容量。如果攻击者能够访问具有大容量网络连接的系统，那么这个系统可能会产生比目标连接容量大得多的通信流量。
- 在经典的ping洪泛攻击中，ICMP回送请求数据包的源地址使用的是攻击者的真实IP地址，攻击的源很容易被识别。
  ① 由于攻击源很容易被明确地识别，那么被发现和受到法律追究的可能性大大增加。
  ② 目标系统会尽可能地响应请求。每当服务器接受到一个ICMP回送请求数据包，就会发送一个ICMP回送响应数据包直接给攻击者，这会将攻击反射给攻击源。

7.2 分布式拒绝服务攻击

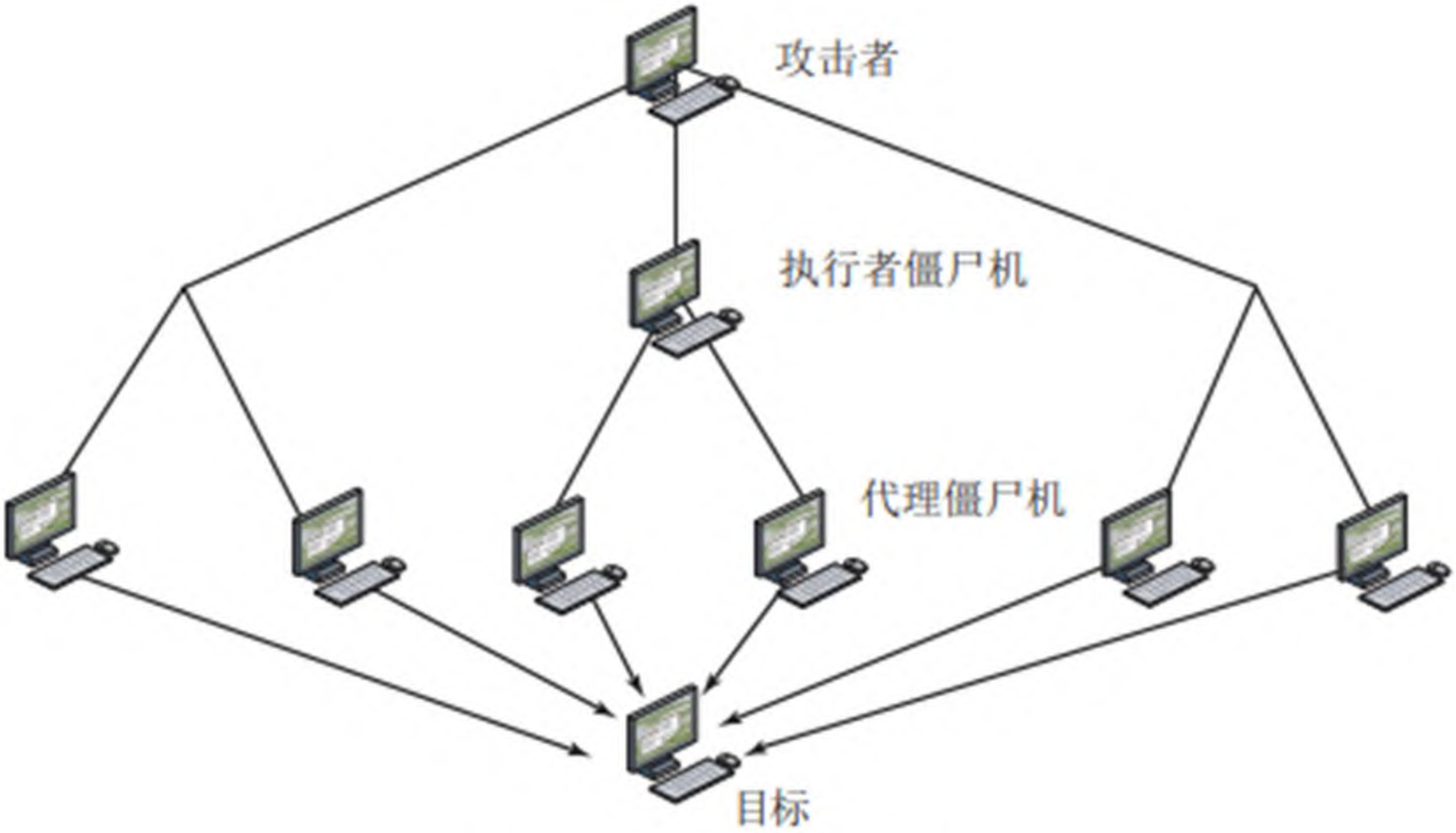


图 7-4　DDoS 攻击体系结构

第八章 入侵检测

- 8.1 入侵者
- 8.2 入侵检测
- 8.3 分析方法
- 8.4 基于主机的入侵检测
- 8.5 基于网络的入侵检测
- 8.6 分布式或混合式入侵检测
- 8.7 入侵检测交换格式
- 8.8 蜜罐
- 8.9 实例系统：Snort

8.2 入侵检测系统IDS

- IDS包括三个逻辑组件:
  - 传感器（sensors）：传感器负责收集数据。传感器的输入可以是包含入侵证据的系统的任何一部分。传感器输入的类型包括网络数据包、日志文件和系统调用痕迹。传感器收集并向分析器转发这些信息。
  - 分析器（analyzers）：分析器从一个或多个传感器或其他分析器接收输入。分析器负责确定是否发生了入侵；此组件的输出表明是否发生了入侵，可以包含支持入侵发生结论的证据。分析器可以提供指导，用于判断什么活动是入侵导致的。传感器的输入也可以被存储起来用于将来的分析，这些输入可以在存储器或者数据库组件中进行检查。
  - 用户接口（user interface）：IDS的用户接口使用户能够查看系统输出或控制系统的行为。在某些系统，用户接口可以看作是经理、主管或者控制台组件。

8.2 入侵检测系统IDS

- IDS通常根据分析数据的来源和类型进行分类，如：
  - 基于主机的IDS（Host-based IDS, HIDS）：监测一台主机的特征和该主机发生的与可疑活动相关的事件，例如进程识别器、进程产生的系统调用等，用作可疑活动的证据。
  - 基于网络的IDS（Network-based IDS, NIDS）：监测特定的网段或设备的流量并分析网络、传输和应用协议，用以识别可疑的活动。
  - 分布式或混合式IDS（Distributed or hybrid IDS）：将来自大量传感器（通常是主机和基于网络的）的信息组合在一个中央分析器中，以便更好地识别和响应入侵活动。

8.3 分析方法

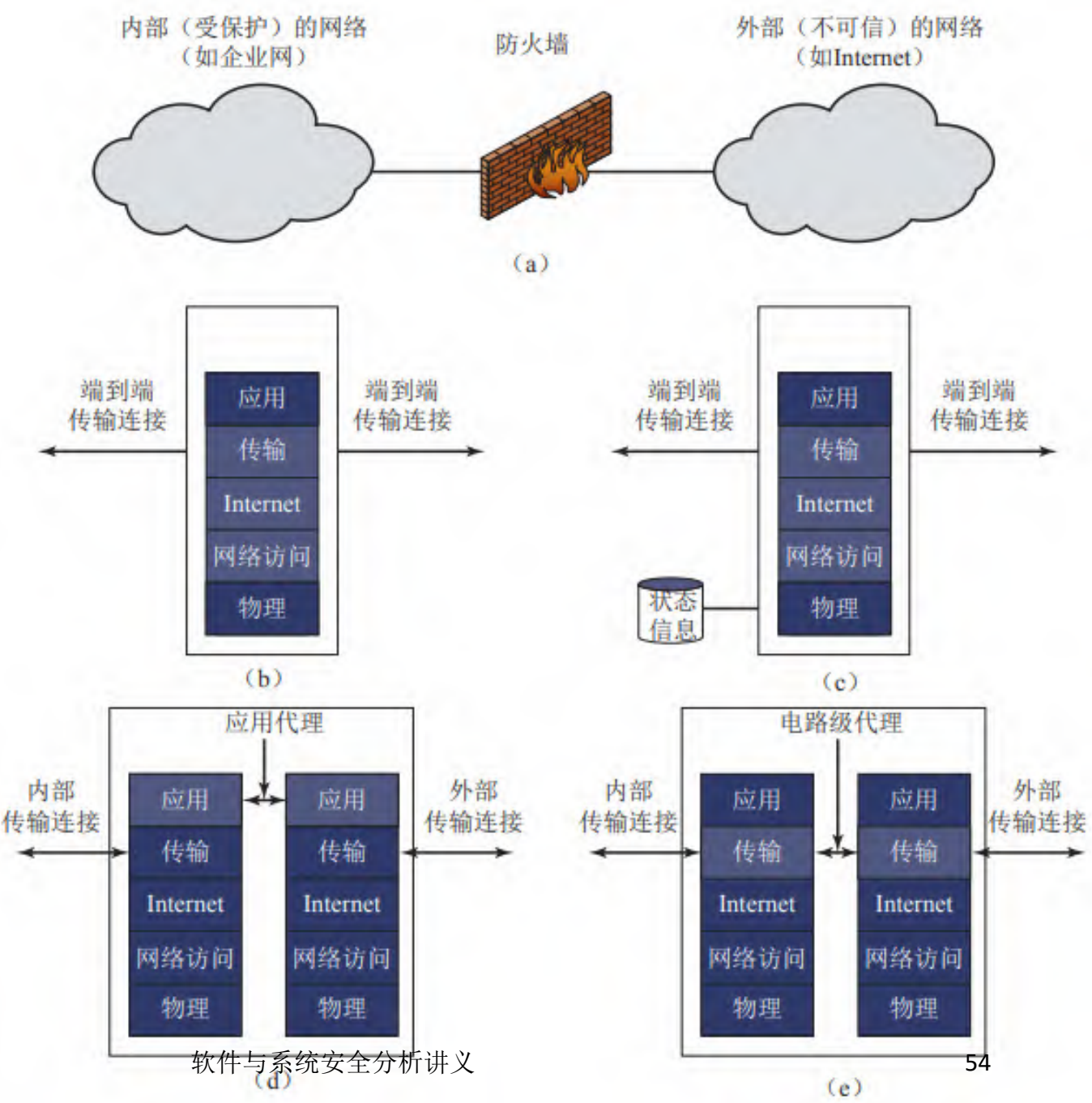
- IDS通常使用以下几种方法之一来分析传感器得到的数据进而检测到入侵：
  ① 异常检测（anomaly detection）：包括采集有关的合法用户在某段时间内的行为数据，然后分析当前观察到的行为，以较高的置信度确定该行为是合法用户还是入侵者的行为。
  ② 特征或启发式检测（Signature or Heuristic detection）：使用一组已知恶意数据模式（特征）或者攻击规则（启发式）组成的集合来与当前的行为进行比较，最终确定这是否是一个入侵者。这种方法也被称为误用检测，仅仅可以被用来识别有模式或者规则的已知攻击。
- 实质上，为了识别恶意或未经授权的行为，异常方法都旨在定义正常或预期之中的行为。特征或基于启发式的方直接定义恶意或未经授权的行为，并可以快速且有效地识别已知的攻击。然而，只有异常检测才能够检测出未知的0-day 攻击，这是因为它是用已知的正常行为去识别异常行为。由于存在这种优势，如果不是我们下面讨论的收集和分析数据的困难性，以及较高的误报率，很明显异常检测将是首选的方法。

第九章 防火墙与入侵防御系统

- 9.1 防火墙的必要性
- 9.2 防火墙的特征和访问策略
- 9.3 防火墙的类型
- 9.4 防火墙的布置
- 9.5 防火墙的部署和配置
- 9.6 入侵防御系统
- 9.7 实例：一体化威胁管理产品

9.3 防火墙的类型

（a）通用模型；
（b）包过滤防火墙；
（c）状态检测防火墙；
（d）应用代理防火墙；
（e）电路级代理防火墙



静态包过滤原理

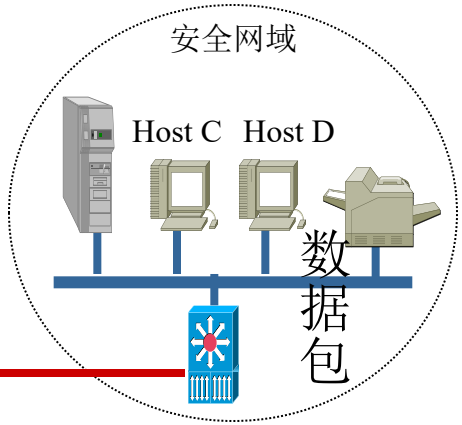
Source	Destination	Permit	Protocol
Host A	Host C	Pass	TCP
Host B	Host C	Block	UDP

控制策略

查找对应的控制策略

根据策略决定如何处理该数据包

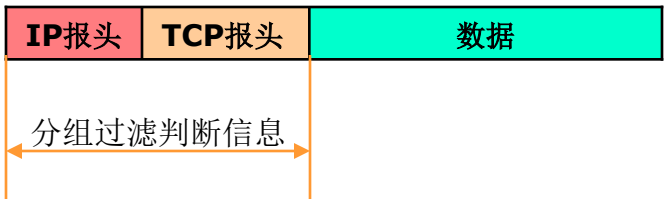
拆开数据包



数据包

数据包

数据包



过滤依据主要是TCP/IP报头里面的信息，不能对应用层数据进行处理

9.6 入侵防御系统

- 也称为入侵检测防御系统（IDPS）
- 它是IDS的扩展，能够尝试阻止或预防检测到的恶意活动
- 基于主机、基于网络、基于分布式或混合式这几种类别
- 异常检测来识别非法用户的行为，或者用特征和启发式检测来识别已知的恶意行为
- 像防火墙一样，可以阻断网络流量，但却需要根据预设的算法来决定后面该干些什么

第10章 缓冲区溢出攻击

- 10.1 缓冲区溢出概述
- 10.2 缓冲区溢出原理
- 10.3 缓冲区溢出的过程
- 10.4 代码植入技术
- 10.5 实例：ida溢出漏洞攻击
- 10.6 缓冲区溢出的防御
- 10.7 小结

10.1 缓冲区溢出概述

- 什么是缓冲区？它是包含相同数据类型实例的一个连续的计算机内存块。是程序运行期间在内存中分配的一个连续的区域，用于保存包括字符数组在内的各种数据类型。
- 所谓溢出，其实就是所填充的数据超出了原有的缓冲区边界。
- 两者结合进来，所谓缓冲区溢出，就是向固定长度的缓冲区中写入超出其预告分配长度的内容，造成缓冲区中数据的溢出，从而覆盖了缓冲区周围的内存空间。黑客借此精心构造填充数据，导致原有流程的改变，让程序转而执行特殊的代码，最终获取控制权。

程序所使用的栈

- 函数被调用的时候，栈中的压入情况如下：

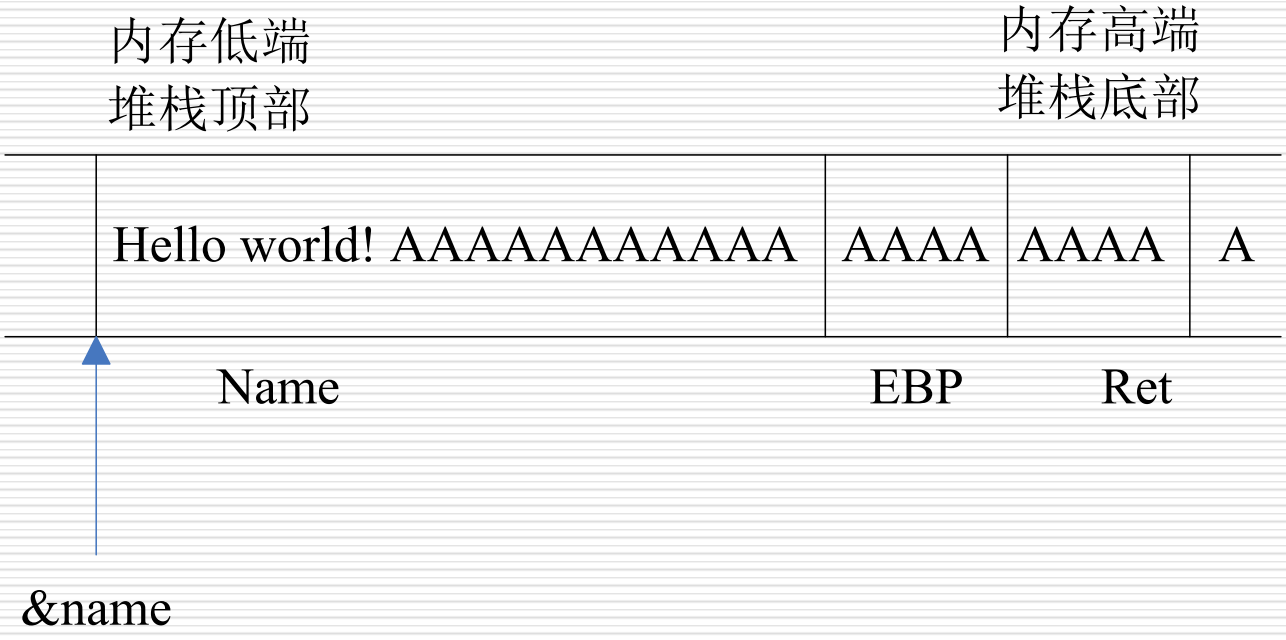
10.2.1 栈溢出实例

- 我们来看一段简单程序的执行过程中对栈的操作和溢出的产生过程。

```
#include <stdio.h>
int main(){
    char name[16];
    gets(name);
    for(int i=0;i<16&&name[i];i++)
    printf("%c",name[i]);
}
```

10.2.1 栈溢出实例

- 如果输入的字符串长度超过**16**个字节，例如输入：**hello world!AAAAAAAA……**，则当执行完**gets(name)**之后，栈的情况如图所示。



10.3 缓冲区溢出攻击的过程

- 缓冲区溢出攻击的目的在于扰乱某些工作在特殊权限状态下的程序，使攻击者取得程序的控制权，借机提高自己的权限，控制整个主机。
- 一般来说，攻击者要实现缓冲区溢出攻击，必须完成两个任务，一是在程序的地址空间里安排适当的代码；二是通过适当的初始化寄存器和存储器，让程序跳转到安排好的地址空间执行。

第十一章 软件安全

- 11.1 软件安全问题
- 11.2 处理程序输入
- 11.3 编写安全程序代码
- 11.4 与操作系统和其他程序进行交互
- 11.5 处理程序输出

11.1 软件安全问题

右表是CWE 评出的前 25 个最严重的软件错误（2022）

这些错误可以归纳为三类：

- 不安全的组件间交互
- 高风险的资源管理
- 脆弱的防御

软件错误类型：组件间的不安全交互
2. 对 Web 页生成期间输入的处理不当（跨站点脚本）
3. 对 SQL 命令中使用的特定元素处理不当（SQL 注入）
4. 对输入的验证不当
6. 对操作系统命令中使用的特定元素处理不当（操作系统命令注入）
9. 跨站点伪造请求（CSRF）
10. 对危险类型的文件不受限制地上载
12. 不信任数据的反序列化
17. 对命令中使用的特殊元素处理不当（Command 注入）
21. 服务器端伪造请求（SSRF）
24. 对 XML 外部实体引用的不当限制
25. 对代码生成控制不当（Code 注入）
软件错误类型：高风险的资源管理
1. 越界写入
5. 越界读取
7. 内存释放后使用
8. 对指向受限目录的路径名限定不当（路径穿透）
11. 空指针解引用
13. 整型溢出或环绕
19. 对内存缓冲区范围内的操作限制不当
22. 对使用共享资源的并发执行同步不当（竞态条件）
23. 不受控制的资源消耗
软件错误类型：脆弱的防御
14. 授权不当
15. 使用硬编码凭证
16. 授权缺失
18. 对关键功能的授权缺失
20. 默认权限设置不当

试题类型

- 本试卷卷面分共60分，本课程其余40分来自实验部分。
- 一 选择题
- 二 判断题
- 三 简答题
- 四 论述题

考试事宜

- 开卷笔试
- 2025年5月29日上午08：30-10：10
- 地点：教1-114

100X100感谢！

zhangyq@ucas.ac.cn