计算机安全期末复习

绪论

- **信息安全的属性**（CIA三要素）
  - 保密性 Confidentiality：防止保密信息泄露给非授权的实体
    - 访问控制：防止未授权的实体获取受保护的信息
    - 加密保护：防止未授权的实体理解受保护的信息
  - 完整性 Integrity
    - 数据完整性：确保数据只能按照授权的指定方式进行修改的属性
    - 系统完整性：系统没有收到未经授权的操控进而能完好无损地执行预定功能的属

性
        - 数据的非法修改
            - 在没有获得授权的情况下修改数据
            - 没有按照授权中指定的方式修改数据
        - 完整性的支撑机制
            - 预防机制：组织非法修改数据或非法操控系统
            - 检测机制：判断完整性是否收到破坏
    - 可用性 Availability：确保系统及时工作并向授权的实体提供所需的服务
- **系统研究的方法**
    - 还原论：把大系统分解为小系统，通过对小系统的研究推知大系统的行为
    - 整体论：把系统堪称一个完整的统一体，而非简单的微观组成元素的集合
- 系统的整体特性：综合特性、涌现性
    - 综合特性：可以分解成系统组成部分的特性
    - 涌现性：不可分解为系统组成部分的特性
- **系统安全工程**：把安全性相关活动和任务融合到系统工程的过程之中，形成的一个系统工程专业分支。力求从系统生命周期的全过程保障系统的安全性
- **系统安全思维**
    - 运营整体论思想分析安全问题
    - 在系统的全生命周期中衡量系统的安全性
    - 通过系统安全工程做事建立和维护系统的安全性
- 安全系统建设过程
    - 从分析现实安全问题开始
    - 结合现实环境和现实目标，指定现实安全策略
    - 加上计算环境因素，形成安全策略
    - 把安全策略表示成精确的安全模型
    - 根据安全模型设计出便于实现的安全机制
    - 实现安全机制，开发出安全系统
- **风险与威胁分析**
    - 风险=资产×漏洞×威胁
    - 资产：IT系统中包括硬件、软件、数据、信息、名誉等
    - 漏洞：系统的脆弱性，可被用来破坏资产
    - 威胁：利用漏洞损坏资产的行为
- **攻击树**
    - 树根：一般类的攻击，攻击目标
    - 树结点：达成攻击所需的子目标

    - 叶子节点：发起一个攻击的不同方式
    - 除了叶子结点外的每一个结点：与节点(AND-node)或或结点(OR-node)
    - 边：赋权值，估算攻击的成本、发生的可能性、成功的可能性
- 攻击面
    - 网络攻击面
    - 软件攻击面
    - 人为攻击面

计算机安全基础

- 计算机安全的各个方面
    - 计算机安全：研究如何预防和检测计算机系统用户的非授权行为
    - CIA三要素
    - 可生存性 Survivability：在系统遭受攻击、故障或意外事故的情况下能够及时完成任务，并及时修复被损坏的服务的能力
    - 可审计性（问责性） Accountability
    - 不可否认性 Non-repudiation：可审计性的更强形式，能对发生的具体行为提供不可欺骗的证据
    - 可控性 Controllability：对于信息安全风险的控制能力，即通过一系列措施，对信息系统安全风险进行事前识别、预测，并通过一定的手段来防范、化解风险，减少遭受的损失的可能性
        - 对信息传播及内容具有控制能力的特性
        - 控制授权范围内信息流向及行为方式
        - 使用授权机制，控制信息传播范围、内容，必要时能够恢复密钥，实现对网络资源及信息的可控性
- 4A：认证、授权、审计、记账
- PDR/PDRR/PPDRR模型
    - 预防 Prevention(PDR)
    - 策略 Policy(PPDRR)
    - 保护 Protect(PDRR/PPDRR)：采取可能的手段保障CIA…
    - 检测 Detect：检擦系统存在的漏洞、脆弱性，检测攻击、恶意代码等
    - 反应 React：对危机安全的事件、行为、过程及时做出响应处理，防止危害进一步扩大，力求系统还能提供正常服务
    - 恢复 Restore：指一旦系统遭到破坏，能尽快恢复系统功能，提供正常的服务
- 计算机安全设计原则

    - 在一个给定的应用中，一个计算机系统中的保护机制集中在数据或操作或用户上
    - 安全机制防止在计算机系统的层次
    - 简单通用机制或功能丰富的安全环境
    - 集中控制或分布控制
    - 防止攻击者访问位于保护机制下面的层
- **侧信道攻击**
- **计算机安全防护原则**
    - 整体性原则
    - 分层性原则
    - 最小授权原则
    - 简单性原则
    - 等级性原则

身份识别与认证

- 认证：验证用户身份的过程
    - eg：张三-*外部实体*；用户id-*身份*；用户进程-*主体*
    - 外部实体必须提供信息，允许系统证实它的身份
    - `实体身份`控制与`其关联的主体`可以进行的`操作`
    - 认证系统的组成
        - 鉴别信息的集合A：实体用来证明其身份的特定信息的集合
        - 辅助信息的集合C：系统存储并且用来证实鉴别信息的信息集合
        - 辅助函数的集合F：由鉴别信息产生辅助信息的函数的集合
          > 如加密算法或hash
        - 鉴别函数的集合L：验证身份的函数的集合
          > $l \in L, l : A \times C \rightarrow \{true, false\}$
        - 选择函数的集合S：允许实体创建或改变鉴别信息和辅助信息
          > 如创建、修改口令，选择加密算法等
    - **可供选择的身份认证方法**
        - 你知道的事情
        - 你拥有的东西
        - 你是谁
        - 你做什么

        - 你在哪里
        - …
- **基于口令的身份认证**
    - **威胁**：弱口令猜测、破解，欺骗，文件泄露，遗忘
    - **缺点**：易于被窃取或攻击
        - 窃听分析：网络数据流窃听、认证信息截取/重放
        - 暴力破解：字典攻击、穷举尝试
        - 社工：窥探口令、骗取口令、垃圾搜索
    - **口令空间的计算**
    - **暴力攻击的时间**
- 基于USB Key的身份认证
    - 特点
        - 带有安全存储空间。可以存储数字证书、用户密钥等秘密数据
        - 硬件实现加解密算法。加解密运算都在USB Key内部进行，保证了用户密钥不会出现在计算机内存中，杜绝了用户密钥被黑客截取的可能
        - 便于携带，安全可靠
    - 工作方式
        - “挑战-应答”认证模式
        - 数字证书热症
- 基于生物特性的身份认证

访问控制

- 访问控制：是在身份认证基础上，依据授权对提出的资源访问请求加以控制

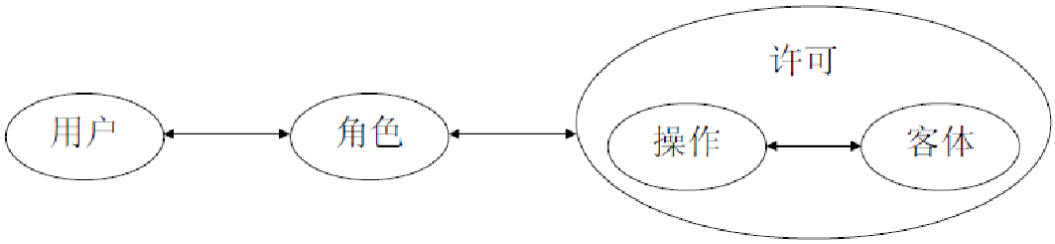
  > 限制已授权的实体访问本系统中资源的过程

    - **主体**：访问的发起者。通常为进程、程序或用户
    - **客体**：可供访问的各种软硬件资源。访问一种资源，就是从这个资源中获取信息、修改资源或利用它完成某种功能
    - **访问**：读写、执行、删除、创建、搜索等
    - 授权：资源所有者对他人使用资源的许可
    - 角色：一个组织或任务中的工作或位置
    - 敏感标签：表示课题安全级别并描述客体数据敏感性的一组信息
- 访问控制的内容
    - 保密性控制：保证数据资源不被非法读出

    - 完整性控制：保证数据资源不被非法改写或删除
    - 可用性控制：保证网络/系统中所有课题不被非法主体破坏
- 访问控制的分类
    - 系统访问控制
    - 网络访问控制
    - 物理访问控制
- 访问控制的基本原则
    - 最小权限原则
    - 多人负责原则
    - 职责分离原则
- **主角和主体**：都是用来指代一个访问操作的活动实体
    - eg: 用户ID-*主角*；用户进程-*主体*
    - 主角：能被授予访问对象的权限或做出能像访问控制决策的声明
    - 主体：IT系统的活动实体
        - 在谈到安全策略时使用主角；在谈论执行安全策略的系统时使用主体

- BLP模型
    - 读 Read
    - 写 Write
    - 添加 Append
    - 执行 Execute
- 创建和删除文件的安全策略
    - Unix：控制对文件目录的写访问
    - Windows：指定特殊的创建和删除权限
- **所有权**
    - 自主访问控制DAC：为每个资源定义一个所有者，让所有者规定谁可以访问
    - 强制访问权限MAC：系统策略规定谁可以访问
- **访问控制矩阵**
    - $M = (M_{so})_{s\in S, o\in O}$，当$M_so \subseteq A$
    - S-*主体集合*；O-*客体集合*；A-*访问操作集合*
    - M_{so}表项规定了主体s可以施加于客体o上的访问操作的集合
    - 列-主体；行-客体
    - 访问能力表CL——访问控制矩阵中的一行
    - 访问控制列表ACL——访问控制矩阵中的一列
    - ACL vs CL
        - 保存位置不同

        - 二者需要鉴别的实体不同：ACL鉴别客体，CL鉴别主体
        - 浏览访问权限：ACL容易，CL困难
        - 访问权限传递：ACL困难，CL容易
        - 访问权限回收：ACL容易，CL困难
        - ACL->CL困难，CL->ACL容易
        - 原因：主体少，客体多
- **基于角色的访问控制RBAC**：将访问权限分配给角色，用户通过被指派为角色从而获得角色所拥有的访问权限
    - RBAC0模型
        - 基本数据元素
            - 用户：一个可以独立访问计算机系统中的数据或用数据表示的其他资源的主体
            - 角色：一个组织或任务中的工作或者位置，它代表一种权限、资格
            - 客体
            - 操作
            - 许可：允许对一个或多个客体执行的操作

    - RBAC1模型：引入角色间的**继承**关系
    - RBAC2模型：引入责任分离关系
        - **角色限制**
            - 角色互斥
            - 角色基数限制
            - 先决条件：如果用户已被分配给另一个指定角色时，用户只能被分配一个特定角色
    - RBAC3模型：既提供了角色间的继承关系，又提供了责任分离关系
    - 特点
        - 支持最小权限原则、责任分离原则
        - 支持数据抽象原则和继承概念
        - 模型中概念与实际系统紧密对应
        - 没有提供信息流控制机制
        - 没有提供操作顺序控制机制

- 基于任务的访问控制TBAC
  - 特点
    - **上下文相关的访问控制模型**
    - **动态授权**
    - 基于实例的访问控制模型
  - 组成
    - 工作流
    - 授权结构体
    - 授权步
    - 受托人集
    - 许可集
    - 任务
    - 依赖
- 保护环
- 访问控制策略：可用ability标记主体和客体，当且仅当客体的ability是主体的ability的一个前缀时允许访问
- **安全级别**
  - 格
    - 格(L,≤)由集合L和偏序≤组成，对于任意两个元素 $a,b\in L$，存在一个最小上界 $u\in L$ 和一个最大下届 $l\in L$ 即：
      - $a\leq u,b\leq u,\forall v\in L:(a\leq v\wedge b\leq b)\Rightarrow(u\leq v)$
      - $l\leq a,l\leq b,\forall k\in L:(k\leq a\wedge k\leq b)\Rightarrow(k\leq l)$
      - 在安全中，如果 $a\leq b$，则说“a受b支配”或“b支配a”
  - 系统高、系统低
    - 受所有其他级别支配的安全等级称为系统低
    - 支配其他所有级别的安全等级称为系统高
    - 系统低和系统高不一定都存在；但如果存在，则一定是唯一的
    - 如果集合L有限，那么系统高和系统低一定存在
  - 安全级别上的格

使用控制

- ABC模型（授权、义务、条件）
  - 三个基本元素：主体、客体、权限

    - 两个扩展元素：主体属性、客体属性
    - 三个授权相关元素：授权规则、义务、条件
- UCON(Usage CONtrol)的16种基本模型
    - 授权规则：preA、onA
    - 义务：preB、onB
    - 条件：preC、onC
    - 属性可变性：不改变0、使用前改变1、使用中改变2、使用后改变3
- **用控制模型描述DAC/MAC/RBAC**

访问监视器RM

- **访问监视器**：所有主体对客体访问进行仲裁的抽象机器
- **安全内核**：实现RM的TCB的硬件、固件、软件的总和
- **可信计算基**：操作系统的安全依赖于一些具体实施安全策略的软件和硬件。这些软件、硬件和负责系统安全管理的人员一起组成系统的可信计算基TCB
    - 操作系统的安全内核
    - 具有特权的程序和命令
    - 处理敏感信息的程序，如系统管理命令等
    - 与TCB实施安全策略有关的文件
    - 其他有关的固件、硬件和设备
    - 负责系统管理的人员
    - 保障固件和硬件正确的程序和诊断软件
- 访问验证机制RVM
    - **三个核心要求**
        - 具有自我保护能力，即使受到攻击也保持自身完整性，从而保证系统安全
        - 总是处于活跃状态，是所有访问请求的唯一入口，以保证程序对资源的所有访问都可以得到访问验证机制的仲裁
        - 设计得足够小，以利于分析和测试，保证访问验证机制实现的正确性和符合要求
- 操作系统完整性
    - 安全策略：用户一定不能修改操作系统（安全内核）
    - 处理两个相互冲突的要求
        - 用户能使用操作系统
        - 用户不得滥用操作系统
    - 实现方式
        - 操作模式（状态信息）

        - 引入状态标志`status flag`

          > 例如Intel 80x86有两个状态位，支持四种模式
          > 例如Unix有root模式和用户模式

      - **受控调用**（受限特权）：系统在root模式只执行一组预先定义的操作，在控制交回给用户前返回用户模式的过程称为受控调用
- 把安全性置于系统核心的原因
  - 在更高的保证级别上评估安全
    - 置于底层，防止受到下层安全机制的连累
    - 为了评估系统的安全性，必须证明安全机制不会被绕过。系统越复杂，越难证明。而在系统的核心，有可能找到相当简单的能够经得起全面分析的结构
  - 减轻因安全性引起的性能损失
  - 将安全策略推向“人机标尺”的机器端
- 陷阱：是CPU的一种特殊输入，包括中断向量表中一个称为中断向量的地址
  - 中断处理器：中断向量表给出了一些程序的位置，这些程序用于处理陷阱指定的时间，称为中断处理器
  - 当一个陷阱发生时，系统在堆栈中保存当前状态，然后执行中断处理器，用这种方式将控制从用户程序那里接管过来。中断处理器必须保证在将控制返回给用户程序之前，将系统恢复到一个正确的状态，如清除root状态位
- 存储器保护
  - 需要的操作
    - 隔离用户空间和操作系统空间：保证自身完整性
    - 用户间逻辑隔离：防止用户访问其他用户的数据
      - 文件管理：处理逻辑存储对象
      - 存储管理：处理物理存储对象
    - 约束一个进程可以访问的存储对象
  - 缺页能够创建一个隐藏信道
  - 安全编址的三种方法
    - 地址沙盒法：OS修改从用户进程接收到的地址
    - 相对编址：操作系统利用用户进程受到的相对地址构造有效地址
      - 栅栏寄存器：使用户只能访问系统空间之外的空间，其志昂用户空间的最高端
      - 基址寄存器：与栅栏寄存器相关的相对编址
      - 边界寄存器：定义用户空间的最低端

      > 基址和边界寄存器允许隔绝各个程序的数据空间

        - 操作系统检查从用户进程受到的地址是否在给定的界限内

计算机实体安全

- 实体安全：物理安全
    - 设备安全：盗窃、损坏、侧信道攻击等
    - 环境安全：机房场地的温度、湿度运行环境
    - 介质安全：存储介质被盗、删除、泄露等
    - 容灾技术：地震、水灾等自然灾害
- TPM安全芯片（可信平台模块）：含有密码运算部件和存储部件的小型片上系统，由CPU、存储器、I/O、密码运算器、随机数产生器和嵌入式操作系统等部件
    - 核心功能：对CPU处理的数据流进行加密，同时检测系统底层的状态
- 可信计算
    - 系统的启动从一个可信任源（通常为BIOS的部分或全部开始），依次验证BIOS、操作系统装载模块、操作系统等，从而保证可信计算平台启动链中的软件未被篡改

Unix/Linux安全

- 主角：用户标识符UID和组标识符GID
    - 一个UID/GID是一个16位数字
    - 某些UID有特殊的意义，root的UID总是0
- 用户账号
- PAM安全认证机制：提供一个框架和编程接口，将认证工作由程序员交给管理员
- 主体
    - 主体是进程。每个进程都有一个进程标识符PID
    - 每个进程都有一个RUID/RGID（真实）和一个EUID/EGID（有效）
        - RUID/RGID：继承于父进程，通常是登录用户的UID/GID
        - EUID/EGID：继承于父进程或正在被执行的文件
- 客体
    - 文件许可分为三组分别定义属主、属组、其他人的rwx权限
    - 目录的许可
        - 有对目录的写许可可以删除一个文件，而不需要对文件的任何许可
        - 设置了粘滞位的目录中的文件，只有在用户是*文件的属主*、*目录的属主*且拥有写权限的情况下或位root情况下才能删除或重命名

    - 四字符的八进制数：第一位表示SUID、SGID、粘滞位
    - 文件许可位的表示
        - 二进制表示
        - 八进制表示
- SUID/SGID
    - SUID/SGID程序与属主或数组的EUID/EGID仪器运行，拥有暂时的或受限的访问权限，这些访问权限一般是不赋予普通用户的
- SELinux
    - 强制访问权限MAC：对所有的文件、目录、端口这类的资源的访问，都可以是基于策略设定的，这些策略是由管理员制定的，一般用户没有权限更改
    - 类型强制TE
        - 所有文件都赋予一个交`type`的文件类型标签
        - 所有进程赋予各自的一个叫`domain`的标签
        - `domain`能执行的操作是由`access vector`在策略里定好的
    - DET模型（Domain, Type, Entity）：把主体分到不同的域，客体设定不同的类型，主客体都称为实体，根据域和类型综合判断进行访问控制
    - RBAC
    - SETE模型切换工作域的条件
        - 进程的新工作域必须拥有对可执行文件的类型的`entrypoint`权限
        - 进程的旧工作域必须拥有对入口点程序的类型的`execute`权限
        - 进程的旧工作域必须拥有对进程新工作域的`transition`权限

```
allow user_d passwd_exec_t: file{getattr execute}
allow passwd_d passwd_exec_t: file entrypoint
allow user_d passwd_d: process transition
```

Android安全

- 分层架构
    - Linux核心层
    - 系统运行库层
        - Dalvik VM：是Google为Android系统设计的JVM
            - 基于寄存器
            - 执行特有的格式——dex字节码
            - 一个应用，一个虚拟机，一个进程

        - ART(Android Runtime)
            - Ahead-Of-Time(AOT)：在应用安装时就预编译字节码到机器语言
        - AOT+解释执行+JIT：
            - 在应用安装时，dex问价不会被预先编译成机器码
            - App运行时，dex文件先通过解释器直接执行，热点函数会被识别并被JIT编译后存储在`JIT code cache`中并生成profile文件记录热点函数信息
            - 手机进入`idle`状态或充电时，系统扫描app目录下的profile文件进行AOT编译
    - 应用程序框架层
    - 应用程序层
- 微内核&宏内核
    - 微内核：Windows NT, Minix, QNX, HarmonyOS
        - 微内核本身完成很少功能，主要传递一个模块对另一个模块的功能请求
        - 降低设计难度、简化维护和修改、可靠性高；通信有效率损失
    - 宏内核：Unix, Linux
        - 模块间通讯通过直接调用其他模块中的函数实现，内核接管内存、文件管理等任务
        - 功能块之间的耦合度太高，修改维护难度大；效率高
- 沙箱
    - Android应用程序在安装时被赋予独特的UID并永久保持

    > 一个应用程序就是一个用户

    - 应用程序与其运行的Dalvik虚拟机运行在独立的Linux进程空间并域其他应用程序完全隔离
    - 如果其他程序拥有该应用程序的共享UID，即设置自己的`sharedUserID`为该应用程序的UID，其他程序就可以突破沙盒的限制访问它的数据
- 权限
    - 系统和应用之间的安全性通过Linux的facilities在进程级别强制实现：给应用程序分配UID&GID
    - 通过Permission机制对特定进程的特定操作进行限制
    - 应用程序之间一般不可以互相访问
    - 一个应用的权限是在其安装的时候确定的，而非运行的时候

Winodws安全

- Windows登录进程**Winlogon**
  - 使用LogonUI收集登录信息
  - 调用**本地安全授权LSA**(Local Security Authority)的`LsaLogonUser`命令
- 本地安全授权子系统LSASS(Local Security Authority Subsystem)
  - 用户模式进程
  - 负责本地系统安全策略（如允许纳西用户登录到本地机器、口令策略、授权、系统安全设计设置、用户认证、创建访问令牌、发送安全审计消息到事件日志等功能）
  - LSASS策略数据库是包含本地系统安全策略设置的数据库，保存在注册表中
    - 哪些域是可信任的
    - 谁允许访问系统
    - 如何访问
    - 分配给谁哪些权限
    - 执行哪一种安全审计
    - ……
- **安全账户管理器SAM**(Security Account Manager)：负责管理一个数据库，包含本地机器上已定义的用户名和组（包括口令和属性）。保存在注册表中
- **注册表**：Windows配置数据的中央数据库
  - 注册表蜂房(hive)是一组键、子键和键值
- **域**：共享用户账户数据库和安全策略的计算机集合
  - 域用户、组、别名、机器的名字形式为:principal@domain=DOMAIN\principal

    > eg: **diego@europe.Microsoft.com = EUROPE\diego**

- **活动目录**：由特定类型对象所构成的树
- **主角**：安全策略中的活动实体，可以被允许或拒绝访问的实体
  - 主角有一个人们可读的名字user name，以及机器可读的标识符SID
  - SID的格式：S-R-I-SA-SA-SA-N
    - S:字符’S’
    - R:更正号
    - I:48bit身份权威
    - SA:32bit子权威
    - N:相对描述符，在权威名空间中是唯一的

    > 主角：Guest S-1-5-21--501
    > ：96bit唯一的机器域标识符，在Windows或控制器安装时创建

- **主体**：操作系统中的活动实体，Windows中进程和线程是主体
- **令牌**：主体的属性
  - 用户安全标识符
  - 组和别名标识符
  - 特权
  - 新对象的默认值
  - 杂项
- **安全描述符**：客体的属性
  - 属主SID
  - 主组
  - **自主访问控制列表DACL**
    - ACE头：访问/拒绝
    - 访问掩码：访问类型（读、写等）
    - SID
  - **系统访问控制列表SACL**
- 许可（访问权限）
  - 许可被编码为32-bit掩码
  - 标准访问权限
    - DELETE
    - READ_CONTROL
    - WRITE_DAC
    - WRITE_OWNER
    - SYNCHRONIZE
  - 通用访问权限
    - GENERIC_READ
    - GENERIC_WRITE
    - GENERIC_EXECUTE
    - GENERIC_ALL
- **受限上下文**
- **数据执行保护DEP**(Data Execution Prevention)
  - 基本原理：将数据所在内存页标识为不可执行

数据库安全

- 数据库安全：保证数据库信息的保密性、完整性、可用性、可控性和隐私性

- 数据库的访问控制
  - 数据库系统：用户认证、访问控制
  - 数据存储：加密存储
- 数据库的完整性控制
  - 物理完整性
  - 逻辑完整性
  - 元素完整性：保持数据字段内容的正确性与准确性。通过定义完整性约束条件、提供完整性检查方法和违约处理实现
    - **触发器Triger**：用户对表进行操作时自动激活相应的触发器进行完整性控制。可以完成如下功能：
      - 检查取值类型与范围
      - 依据状态限制
      - 依据业务限制
    - 纠错与恢复：采用冗余，增加附加信息来检测数据中的不一致性
      - 检验纠错码
      - 镜像技术
      - 恢复
    - 并发控制
      - 主键的子集不能为空
      - 不能有不匹配的外键值
      - 字段值必须与字段数据类型、格式、有效范围吻合
      - 自定义完整性
- **自主访问控制**
  - 自主访问授权
    - (S,O,A,P):谓词P为真时，主体S有权对客体O进行操作A
    - 授权发放与回收
      - GRANT(S,O,A)->GRANT A ON O TO S

        ```
        GRANT SELECT ON emp TO bob
        ```

        - `WITH GRANT OPTION`可以进行授权委托
        - 缺省或`CASCADE`为递归回收
        - `RESTRICT`为非递归回收
      - REVOKE(S,O,A)->REVOKE A ON O FROM

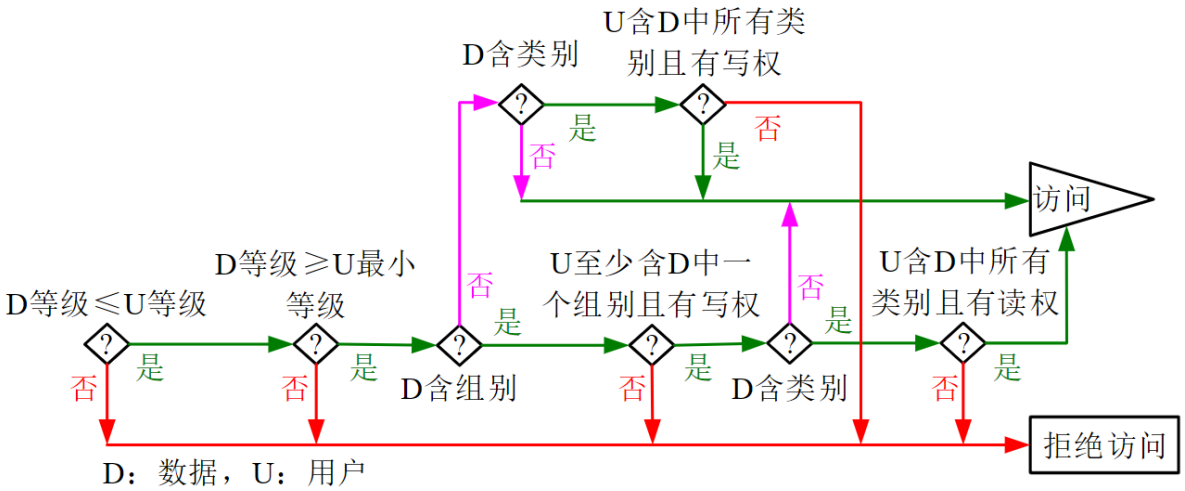
        ```
        REVOKE SELECT ON emp FROM bob
        ```

      - 否定优先原则：同时拥有否定式授权和肯定式授权，不论授权发放先后，否

定式授权发挥作用

- 个体优先原则：用户拥有的授权优先级高于小组拥有的授权

- **基于视图的访问控制**
  - 优势
    - 支持基于内容的访问控制
    - 将复杂度查询编写为视图，减少用户查询的复杂度
    - 限制某个试图只能访问基表中的部分数据，提高安全性
    - 增强上下文依赖和数据依赖的安全策略
    - 视图能够实施受控调用
    - 安全的视图可以代替安全标签（类似MAC）
    - 数据可以很容易地重新分类
  - 劣势
    - 视图多，实现和维护复杂
    - 完整性和一致性不易实现
    - 对于单个数据项难以决定哪个个体有访问权
- **基于角色的访问控制**
  - 由用户执行授权操作时，回收用户的授权引起递归回收；由角色执行授权操作时，回收用户的授权不会引起递归回收
- 强制访问控制机制——**甲骨文标签安全机制**OLS(Oracle Label Security)
  - 三元组<等级，类别，组别>
  - “读”访问判定
    - 用户标签中的等级必须≥记录标签中的等级
    - 用户标签中的组别至少包含记录标签中的一个组别并拥有对改组别的写权限
    - 用户标签的类别必须包含记录标签中的所有类别
  - “写”访问判定



        - 用户标签≥记录标签≥用户最小等级
        - 用户标签的组别至少包含记录标签中的一个组别并有对该组别的写权限
        - 用户标签的类别必须包含记录标签中的所有类别
        - 若记录标签不含组别，则用户必须对记录标签的所有类别有写权限
        - 若记录标签含有组别，则用户必须对记录标签的所有类别有读权限
- **数据库数据推理**
    - 统计数据库的推理
    - 跟踪攻击
    - 差分隐私
    - 防御
        - 只有统计型查询覆盖一个足够大的子集时才被允许执行
        - 把已生成的统计结果按一定方式合并后再提供给查询者
        - 从数据库中抽取一个随机样本数据库，再样本数据集上查询生成结果
        - 给数据库的数据项添加随机噪声，在加噪数据集上生成结果
- **数据库备份**
    - 完全备份
    - 差异备份：只备份上次完全备份后发生更改的部分数据库
    - 增量备份：每次备份的数据知识相当于上次备份后增加和修改过的数据

系统完整性检查

- **系统可信引导与系统安全引导**
    - 系统可信引导：确保引导过程中获得控制权的所有组件的完整性都没有受到过破坏
        - ASGIS ROM：存储组件的原始指纹
        - 系统BIOS分为两部分
            - 主BIOS：包含执行完整性验证任务的代码
            - 辅BIOS：包含BIOS的其他成分以及CMOS
        - UP-ASGIS机制假设AEGIS ROM和主BIOS时可信的
        - 基于签名的组件验证：在系统中保存原始指纹的数字签名而非原始指纹
    - 系统安全引导：系统可信引导不能抵御拒绝服务攻击，在可信引导基础上增加系统恢复功能
- **进程完整性**
    - 进程模式
        - 普通模式
        - 篡改响应模式：需要进行完整性验证

    - 初始状态中的完整性
        - 计算并检查程序hash
        - 检查运行环境
    - 中断过程中的完整性
        - 保护寄存器信息
    - 存储介质中的完整性
        - 读取时验证完整性
        - 受验证的内存位置：最高地址为=1，只允许一个进程修改
    - 输出结果完整性
        - CPU私钥对{H(porg),M}数字签名
    - 基于默克尔树实施完整性验证

BLP模型

- 安全策略：划分系统状态为一组经授权的/安全状态和一组未经授权/不安全状态的陈述
- 安全模型：
    - 精确的，无歧义的
    - 简单的和抽象的，容易理解
    - 一般只涉及安全性质，不过多设计系统的功能或实现
    - 安全策略的显示表示
- 状态机模型
    - 流程
        - 定义安全的状态集合
        - 检查确保所有的状态迁移都是安全的
        - 检查系统的起始状态是安全的
- **状态集**B×M×F
    - B=P(S×O×A)是当前访问的集合
        - $b \in B$ 是一个三元组 (s, o, a)
    - M表示所有获得访问许可的矩阵的集合
        - $m = (m_{SO}),\ s \in S, O \in O$
    - F是安全级别分配的集合
        - $f \in F$ 是一个三元组 (f_S, f_C, f_O)
        - f_S：主体可以拥有的最高安全级别
        - f_C：主体当前的安全级别
        - f_O：所有客体的安全级别

    - 一个独立的状态由(b,m,f)决定
- MLS策略
    - 简单安全性ss-property:无向上读
    - 星特性*-property:无向下写
    - 自主安全性ds-property
- 基本安全定理：如果系统中所有的状态迁移都是安全的，并且系统的初始状态也是安全的，那么不管输入情况如何，其后的每一个状态也都是安全的
- 隐藏信道

安全模型

- Biba模型
    - 简单完整性：无向上写
    - 完整性*-property：主体s可以读客体o，o’级别≤o级别时，s可以写o’
    - 动态完整性级别
    - 信息传递路径
    - 调用性
    - 环属性
- Chinese Wall模型：用于域权限隔离
    - 公司数据集：有关特定公司的客体的集合
    - 利益冲突类：不应该知道该客体内容的公司的集合
    - 安全标签
    - 净化信息：去除了敏感细节，不受访问限制的信息
    - ss-property
        - 访问用户已经拥有的一个公司数据集
        - 访问一个完全不同的利益冲突类
    - *-property
        - 满足ss-property
        - 没有其他的位于不同公司数据集，且包含非净化信息的客体能被阅读时，对一个客体的写访问才是允许的
- 熵
    - 设 $\{x_1,x_2,...,x_n\}$ 为变量x的可能取值

$$H(x)=-\sum_{i=1}^{n}p(x_i)log_2p(x_i)$$

- x到y的信息流用给定y的值后的x的信息量平均值(条件熵)的变化来度量

$$H_y(x) = -\sum_{j=1}^{m} q(y_j) \sum_{i=1}^{n} p(x_i|y_j) log_2 p(x_i|y_j)$$

- 信息流模型
  - 强/弱信息流
    - 强信息流：直接赋值
    - 弱信息流：信息从x流向y，但不存在形如y:=f(x)的显式赋值

安全评估

- 安全评估框架
  - 产品：将被用于多种应用且需要满足一般的安全要求的部件
  - 系统：满足给定应用特定需求的各种产品的集合体
  - 评估：评定一个产品是否具有它所声明的安全属性
  - 证明：评定一个已评估的产品是否适合给定的应用场合
  - 鉴定：确定一个已证明的产品是否可以在给定场合里应用
- TCSEC
  - 四类七个级别，27条评估标准
  - 四个安全等级
    - D - 最小保护级
    - C - 自主保护级：自主访问控制和审计跟踪
      - C1：隔离用户与数据，用户具备自主安全保护能力
      - C2：比C1有更细粒度的自主访问控制，通过注册过程控制、审计安全相关事件和资源隔离，使单个用户为其行为负责
        - C2被认为对商业应用是最合理的安全级别
    - B - 强制保护级（基于标签）：TCB应维护完整的安全标记，并在此基础上执行一系列强制访问控制规则；主要数据结构必须携带敏感标记
      - B1：标记安全保护级
      - B2：结构化保护级
      - B3：安全区域保护级
    - A - 验证保护级
- ITSEC
  - 功能准则：F1~F10

        - 1~7级对应于TCSEC的D~A
        - 6~10级对应
            - F6：数据和程序的完整性
            - F7：系统可用性
            - F8：数据通信完整性
            - F9：数据通信保密性
            - F10：包括保密性和完整性的网络安全
    - 评估准则：E0~E6
- **通用准则CC**(Common Criteria)：EAL1~EAL7

等级保护

- 等级保护：根据信息系统应用业务重要程度及其实际安全需求，实行分等级、分类、分阶段实时保护
- 等级保护制度的主要内容
    - 对各种秘密/专有/公开信息分类分等级管理和保护
    - 对信息系统按业务安全应用域和区实行分等级保护
    - 对系统中使用的信息安全产品实行按分等级许可管理
    - 对信息系统的安全服务资质分等级许可管理
    - 对信息系统中发生的信息安全事件分等级响应、处置
- **等级保护2.0的十大安全类**
    - 安全物理环境
    - 安全通信网络
    - 安全区域边界
    - 安全计算环境
    - 安全管理中心
    - 安全管理制度
    - 安全管理机构
    - 安全人员管理
    - 安全建设管理
    - 安全运维管理
- **等级保护的主要工作**
    - 定级
    - 备案
    - 建设/整改

    - 定期等级测评
    - 定期监督检查
- 等级保护对象的定级
    - 等级划分（简单记法）
        - 公民、法人、组织的合法权益 —— 0级
        - 社会秩序、公共利益 —— 1级
        - 国家安全 —— 2级
        - 一般损害 +1级
        - 严重损害 +2级
        - 特别严重损害 +3级（对公民、法人、组织的合法权益+2级）

        > eg：等级保护对象受到破坏后，会对社会秩序和公共利益造成特别严重损害，或者对国家安全造成严重损害->第四级

    - 定级流程
        - 确定定级对象
        - 初步确定等级
        - 专家评审
        - 主管部门核准
        - 备案审核
    - 定级建议
        - 一般信息系统
        - 重要信息系统、非涉密敏感信息系统
        - 涉及秘密的信息系统
        - 国家重要领域、部门中特别重要系统，如铁路、电力等调度系统，银行、证券、海关等重要行业的核心系统
        - 国家重要领域、部门中极端重要系统

密码应用安全性评估

- 含义：合规性、正确性、有效性
- 密码应用基本要求的八大安全类
    - 物理和环境安全
    - 网络和通信安全
    - 设备和计算安全

    - 应用和数据安全
    - 管理制度
    - 人员管理
    - 建设运行
    - 应急处置
- **技术标准描述的区别**（要求从高到低）
    - 应：应该、要求
    - 宜：推荐、建议
    - 可：可以、允许

云计算安全

- **云计算主要特性**
    - 按需自助服务
    - 泛在接入
    - 资源池化
    - 快速伸缩性
    - 服务可计量
- **云计算的服务模式**
    - 软件即服务SaaS(Software as a Service)
    - 平台即服务PaaS(Platform as a Service)
    - 基础设施服务IaaS(Infrastructure as a Service)
- **云计算的部署模式**
    - 公有云
    - 私有云
    - 混合云
    - 社区云
- **云服务商和客户之间的安全责任划分**
    - 物理环境、物理设备、资源抽象控制层都位于云服务商的完全控制下，安全责任由云服务商承担。
    - 应用软件层、软件平台层、虚拟化计算找资源层的安全责任由双方共同承担。
    - 越靠近底层的云服务，客户的管理和安全责任越大；反之服务商的责任越大

基于代码的访问控制

- **堆栈遍历**：当函数请求访问以一个受保护的资源，堆栈遍历用来确定调用者是否有需求的许可。调用者有效许可的计算，取决于调用栈上所有函数的许可的交集
- **惰性计算**：只有访问一个资源需要许可时采取评估授权的许可
- **热情计算**
- **许可断言**
    - 许可断言附着在当前栈帧上，当从调用部件返回时撤销
    - 对战便利在遇到许可断言的帧并授予这个许可时终止
    - 如果还有许可没有被断言，堆栈遍历将继续执行

入侵检测

- **入侵检测系统IDS**：进行入侵检测的软件和硬件的组合
- **入侵防御系统IPS**：监视网络或网络设备的网络资料传输行文，能够及时中断、调整或隔离一些不正常或是具有伤害性的网络资料传输行为
- **深度包检测DPI**：在分析包头的基础上，增加了对应用层的分析，是一种基于应用层的流量监测和控制技术
- **深度流检测DFI**：一种基于流量行为的应用识别技术
- **态势感知**：一种基于环境的、动态、整体地洞悉安全风险的能力
    - 以大数据为基础
    - 从全局视角提升对安全威胁的发现识别、理解分析、响应处置能力的一种方式
    - 最终为了决策与行动，是安全能力的落地
- **入侵检测方法**
    - 信息收集
    - 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）
    - 结果处理
- **入侵检测系统分类**
    - 按照分析方法的分类
        - 异常检测模型：偏离正常活动的特征时认为入侵
            - 漏报率低，误报率高
        - 误用检测模型：匹配非正常活动的特征时认为入侵

        - 误报率低，漏报率高
    - 按照数据来源分类
        - 基于主机
        - 基于网络
        - 混合型
    - 按照系统各模块的运行方式
        - 集中式
        - 分布式
    - 根据时效性分类
        - 脱机分析：行为发生后，对产生的数据进行分析
        - 联机分析：在数据产生的同时或发生改变时进行分析

应急响应与灾备恢复

- **信息报障技术框架IATF**(Information Assurance Technical Framework)：为保护政府、企业信息及信息基础设施提供技术指南
    - 核心思想：纵深防御（深度防护）

    > 纵深防御也被称为深度防护战略，是指网络安全需要采用一个多层次、纵深的安全措施来保障信息安全

    - 核心要素
        - 人（管理）
        - 技术
        - 操作（运维）
    - 焦点领域
        - 网络和基础设施
        - 区域边界
        - 计算环境
        - 支撑性基础设施
    - 基本安全准则
        - 保护多个区域的原则
        - 分层防御原则
        - 适度安全原则
- **应急响应**
    - 过程

        - 应急准备
        - 监测与预警
        - 应急处置
        - 总结与改进
- 灾难恢复
    - 容灾备份系统的种类
        - 数据容灾
        - 应用容灾
    - 容灾备份系统的组成
        - 数据备份系统
        - 备份数据处理系统
        - 备份通信网络系统
        - 完善的灾难恢复预案
- RAID
    - RAID0：并联硬盘。提高磁盘性能和吞吐量。没有冗余或错误修复能力
    - RAID1：磁盘镜像。不影响性能情况下保证系统的可靠性和可修复性，具有很高的数据冗余能力。磁盘利用率为50%，成本高
    - RAID10：镜像阵列条带，RAID0+RAID1
    - RAID5：一个盘存其他盘的奇偶校验信息。一个盘发生损坏后，利用剩下的数据和相应的奇偶校验信息恢复被损坏的数据。是一个存储性能、数据安全和成本兼顾的存储方案。只允许一块硬盘出现故障。