计算机安全知识点

绪论

信息安全的概念

信息安全学关注信息本身的安全，可以分为数据安全和系统安全。信息安全的任务是保护信息财产防止遭到恶意泄露、修改和破坏

CIA三要素

保密性：保护未经授权的信息泄露、完整性：防止未授权的信息被破坏和修改、可用性：防止未授权的信息或资源被截留

系统研究的方法

还原论：把一个大的系统分成小的系统，然后通过研究小的系统来推测大的系统的功能。

整体论：把系统视作一个整体去进行分析，一个完整的被观察单位，而不是简单的微观组成元素的集合

系统安全工程

把安全性相关的活动和任务融合到系统工程的过程之中，形成一个系统工程专业分支。力求从系统生命周期的全过程去保障系统的安全性

系统安全思维

- 运用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
- 通过系统安全工程措施建立和维护系统的安全性

风险分析的基本方法

风险=资产X威胁X漏洞

评定资产价值、漏洞危险程度、威胁可能性之后进行风险计算

攻击面

由系统中可到达的和可被利用的脆弱点构成

分类：网络攻击面、软件攻击面、人为攻击面

攻击面分析：评估系统威胁的规模和严重性有用的技术

攻击树

树根：攻击目标；树节点：达成攻击所需要的子目标；叶子结点：发起一个攻击的不同方式；除了叶子结点外的每一个节点：与节点/或节点；边：赋权值。

社会工程学

利用人的弱点，进行注入欺骗、伤害等危害手段，获取自身利益。

计算机安全基础

计算机安全的内涵

计算机安全研究在计算机系统中维护安全所使用的技术，具体来讲就是研究如何预防和检测计算机系统用户的非授权行为、保护计算机系统中的资产CIA

PDRR/PDR/PPDRR

PDRR：保护、检测、反应、恢复

保护：保障信息的CIA；检测：检测可能存在的安全威胁；反应：针对安全威胁作出响应；恢复：系统受到破坏后，恢复到正常功能。

PDR：预防、检测、反应

预防：采取措施防止资产收到损害；检测：采取措施检测出什么资产受损、怎么受损、谁造成的伤害；反应：采取措施以便能够重新获得资产，或者使资产从损害中恢复

PPDRR：策略、保护、检测、反应、恢复

计算机安全的五个设计原则

- 在一个给定的应用中，一个计算机系统中的保护机制应该集中在保护数据、操作还是用户上
- 一个安全机制应该被放置在计算机系统的哪一个层次上面（分层模型，从硬件到应用）
- 与富有特色的安全环境相比，你是否偏爱简单性和更高的保证
- 定义和实施安全的任务是应该交给一个中央实体，还是应该托付给系统中的各个成员
- 如何防止攻击者访问位于保护机制下面的层

隐蔽信道

是一个不受安全机制控制的信息流。采用特殊编码，是公开信道中的一种能够进行隐蔽通信的信道。

一个状态变量：一次传递一个比特位的信息

- 存储信道：如果一个进程直接或间接地写一个存储单元，另一个进程直接或间接地读该存储单元（属性，而不是内容）
- 定时信道：通过计数器的值，是否是隐定时信道：是否有基准时间

旁道攻击

避开了复杂的密码算法，利用密码算法在软硬件实现中泄漏处的各种信息进行攻击。

不攻击密码本身，而是攻击那些实现于不安全系统上的加密系统。

身份识别与认证

口令空间的计算

假定你只允许使用 26 个字符来构造长为 n 的口令. 进一步假设你在区分大小写和不区分大小写的两个系统中使用了此相同的口令，试给出猜测出大小写区分的口令所需要的最大尝试次数.

解：

这道题的易错点在于题意的理解. 要搞清楚题目中描述的这种密码系统的口令特点，进而确定口令空间大小.

首先从不区分大小写的系统入手尝试. 其口令空间大小为 26^n. 根据尝试得到的口令，再到区分大小写的系统上进行尝试，这时只需要确定每一个位置上到底是大写还是小写，所以只需要 2^n 次尝试.

由上述分析可知，所需要的最大尝试次数为

$$N = 26^n + 2^n$$

蛮力攻击的时间

假定口令长度为 6 个字符，可以使用字母和数字，区分大小写. 在以下条件下，蛮力攻击平均所需要的时间分别为多少?

- 检查一个口令需要 1/10 秒.
- 检查一个口令需要 1 微秒.

解：

每个字符的可能取值个数为：26+26+10=62. 所以口令空间为 62^6. 注意到蛮力平均检测口令数为口令空间大小的一半，于是可得

- 检查一个口令需要 1/10 秒

$$t = \frac{1}{2} \times \frac{1}{10} \times 62^6 = 2.84 \times 10^9 s$$

- 检查一个口令需要 1 微秒

$$t = \frac{1}{2} \times 10^{-6} \times 62^6 = 2.84 \times 10^4 s$$

用户身份认证可以基于什么信息

- 你知道的事情--秘密信息
- 你拥有的东西--物理令牌
- 你是谁--生物特征
- 你做什么--行为特征
- 你在哪里--位置信息（物理位置、IP地址）

口令认证机制面临的威胁

- 口令猜测
- 口令欺骗
- 口令文件泄漏
- 口令遗忘

口令认证机制的缺点

- 网络数据流窃听
- 认证信息截取/重放
- 字典攻击
- 穷举尝试
- 窥探口令
- 骗取口令
- 垃圾搜索

访问控制

主角

访问控制中被授予访问客体权限的对象。

主体

拥有主角身份的实体。

客体

访问动作的对象，信息系统中的某种资源，如文件

在谈到安全策略的时候使用主角，在谈论执行安全策略的系统时使用主体，当主体请求访问首保护的对象时，引用监控器会检查与主体绑定的主角是否有权限访问对象

自主访问控制DAC

也称为基于身份的是访问控制，在确认主体身份以及他们所属组的基础上对访问进行限制的一种方法

DAC对应着访问控制列表ACL。

对用户的身份进行鉴别，然后就根据ACL所赋予用户的权限允许或限制用户使用客体资源

强制访问控制MAC

也称为基于规则的访问控制。强制访问控制MAC是一种强制主体服从访问控制策略的访问方式。

MAC对应着有偏序关系的安全级别

在强制访问控制中，主体和客体都被赋予了一定的安全级别，只有满足相应的偏序关系时，访问控制决策为允许访问操作。

访问控制矩阵

是一种访问控制结构，访问权限可为每个主体和客体的组合单独定义，即访问控制矩阵。

能力

访问权限和主体保存在一起——访问控制矩阵的行

访问控制列表ACL

访问权限和客体保存在一起——访问控制矩阵的列

- bill.doc的ACL
  - Bob：读，写
- edit.exe的ACL
  - Alice：执行
  - Bob：执行
- fun.com的ACL
  - Alice：执行，读
  - Bob：执行，读，写

如果想查询主体“Alice”能够访问的客体，应当如何操作？

需要遍历查询所有客体的ACL

	bill.doc	edit.exe	fun.com
Alice	-	{执行}	{执行,读}
Bob	{读,写}	{执行}	{执行,读,写}

中间控制、组、否定许可、角色、特权

本质是通过引入中间层形成分层结构。

组：具有类似访问权限的用户

否定许可：就是访问控制结构中的一个表项，它规定了一个主体不允许执行的访问操作。和组结合，用于否认一个主体从它的组成员关系中获得的许可

特权：可以收集权限以执行某些特权操作

角色：一组特定应用的操作称为角色，主体从他们履行的角色上获得访问权限

- 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控制权限

安全级别的偏序关系

安全标签的格

- 简答——证明：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 是有限的，那么其系统高和系统低一定是存在的.(课件上有提到，课程中也有讲到)
- 大题——和作业题目 (练习 4.8) 比较像 (不过更复杂一些，还有一个难点在于安全标签的确定) 考察基于格的访问控制策略. 给了一个情景 (好像有学院、老师)，让确定有多少种安全标签，具有何种安全标签的主体能够访问给定安全标签的客体，具有给定安全标签的主体能够访问哪些客体. 这道题目我当时做错了，试卷上主要就是错了这个大题 (应该是).

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

角色继承：分为一般继承、受限继承

- 一般继承：仅要求角色继承关系是一个绝对偏序关系，允许角色间的多继承
- 受限继承：进一步要求角色继承关系是一个树结构，要求角色间单继承

角色限制：某一个用户不可能同时独立地完成这些操作

角色互斥、角色基数限制

基于任务的访问控制TBAC

动态授权：授权步的生命期、许可的次数限制和授权步的自我动态管理

上下文环境：数据在工作流中流动的时候，执行操作的用户在改变，用户的权限也在改变。

使用控制 *

ABC**模型的三个基本元素和三个授权相关元素**

三个基本元素：主体、客体、权限

三个授权相关元素：授权规则、义务、条件

- 主体属性：ATT(S)，针对主体做访问控制决策时，用于判断主体是否满足授权规则的主体参数
- 客体属性：ATT(O)，针对客体做访问控制决策时，用于判断客体是否满足授权规则的客体参数
- 授权规则：允许主体使用客体特定权限必须满足的规则集
- 义务：主体获得或形式客体的访问权利前或过程中必须完成的操作
- 条件：主体获得或行使对客体的访问权利前必须满足的系统或执行环境的强制约束条件

四个授权因素：

- 授权规则：使用前授权（preA）、使用中授权（onA）
- 义务：使用前（preB）、使用中（onB）
- 条件：使用前（preC）、使用中（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）

UCON的16种基本模型

决策因素	0(不改变)	1(使用前改变)	2(使用中改变)	3(使用后改变)
PreA	Y	Y	N	Y
OnA	Y	Y	Y	Y
PreB	Y	Y	N	Y
OnB	Y	Y	Y	Y
PreC	Y	N	N	N
OnC	Y	N	N	N

用使用控制模型描述DAC/MAC/RBAC

- UCON 描述 DAC：
  1. 主体 S 为用户或用户组；客体 O 为访问控制的对象
  2. 主体属性为用户的身份标记 id，即 ATT(S)={id}；客体属性为访问控制列表 ACL，即 ATT(O) = {ACL}
  3. allowed(s，o，r)⇒(id(s)，r) ∈ ACL(o). 其中 allowed(s，o，r) 表示主体 s 对客体 o 具有执行 r 操作的权限
- UCON 描述 MAC：
  1. 主体 S 为用户或用户组；客体 O 为访问控制的对象
  2. 主体属性为用户的安全级别，即 ATT(S)={clearance}；客体属性为客体的安全级别，即 ATT(O) = {classification}.
     注：L 表示具有偏序关系的安全级别集合.clearance：$S \to L$，是访问主体 S 和安全级别 L 之间的映射函数；classification：$O \to L$，是客体 O 和安全级别 L 之间的映射函数.
  3. allowed(s，o，read)⇒clearance(s)≥classification (o)；
     allowed(s，o，write)⇒clearance(s)≤classification (o)

用 $UCON_{preA0}$ 模型描述 RBAC0 模型：

1. 主体 S 为用户或用户组；客体 O 为访问控制的对象
2. 主体属性为激活的角色 (主体绑定的角色)，即 ATT(S)={actRole}，实现“用户-角色”分配；客体属性为授权角色 (能够访问该客体的角色)，即 ATT(O) = {pRole}，实现“角色-权限”分配.
3. allowed(s，o，r)= role ∈ actRole(s) ∧ role′∈pRole(o，r)∧role ⩾role′

用 $UCON_{preA0}$ 模型描述 RBAC1 模型：

1. 主体 S 为用户或用户组；客体 O 为访问控制的对象
2. 主体属性为激活的角色 (主体绑定的角色)，即 ATT(S)={actRole}，实现“用户-角色”分配；客体属性为授权角色 (能够访问该客体的角色)，即 ATT(O) = {pRole}，实现“角色-权限”分配.
3. allowed(s，o，r)= role ∈ actRole(s) ∧ role∈pRole(o，r)

访问监控器

访问验证机制的三个核心要求

- 具有自我保护能力，以保证引用验证机制即使受到攻击也能保持自身的完整性
- 总是处于活跃状态，以保证程序对资源的所有引用都得到引用验证机制的仲裁
- 设计的足够小，以利于分析和测试，保证引用验证机制实现的正确性和符合要求

访问监控器

是一个访问控制概念，指所有主体对客体进行仲裁的抽象机器

安全内核

实现RM的可信计算基的硬件、固件和软件的总和，必须处理所有的访问控制，必须保护安全内核，使之不被修改，并能够证明是正确的

可信计算基TCB

是计算机系统中全部的保护机制的总和。操作系统的安全依赖于一些具体实施安全策略的可信的软件和硬件。这些软件、硬件、如泽系统安全管理的人员一起组成了系统的可信计算基

受控调用

保护环，从外到内安全级别逐渐升高

必须对那些请求较高特权的操作进行访问管理

当外层级别低的程序需要调用内层级别高的程序时，需要通过门，其访问时受限制的

可信路径

输入安全敏感数据时，需要建立一条从输入/输出设备到TCB的可信路径

计算机实体安全

可信计算

可信：如果它的行为总是以预期的方式，朝着预期的目标，则一个实体是可信的

可信计算：主要是通过增钱现有的PC终端体系结构的安全性来保证整个系统的安全

TPM

可信平台模块，具有安全存储和加密功能，是一种含有密码运算部件和存储部件的SoC

软件狗

电子设备，USB-Key

Unix/Linux安全

Unix**的自主访问控制、粒度**

粒度：owner、group、other

访问控制：Unix中使用有效UID/GID进行访问控制，访问控制时文件许可位检查顺序为：

1. 检查UID：如果是文件属主，则用户的权限就是文件许可位中表示的文件属主的权限，如果不是，进行第二步
2. 检查GID：如果是文件属组，则用户的权限就是文件许可位中表示文件属组的权限，如果不是，第三步
3. 其他人访问许可位决定用户是否可以获取访问权限

主角

主角是用户标识符UID和组标识符GID，每一个进程都有一个真实UID/GID和一个有效UID/GID

- 真实 UID/GID：继承于父进程，通常是登陆用户的 UID/GID。
- 有效 UID/GID：继承于父进程或正在被执行的文件

主体

主体是进程。每一个进程都有一个进程标识符PID

客体

文件，在Unix系统中一切资源都以文件形式组织。

文件许可位：其ACL即为文件许可位，分为三组，分别定义了owner、group、other的读、写、执行权限

- r--r--r--：属主、属组和其他人都只有读取权限。

二进制：

- 100: 只读
- 110: 读写

八进制

- 411 owner读，group和other有执行权限
- 777 三个分类都有读写执行权限

三个附加位：

- SUID：设置UID为文件owner（让用户的EUID为属主UID）
- SGID：设置GID为文件group（让用户的EGID为属组GID）
- 粘滞位：只有在用户是文件的属主、目录的属主并且拥有写权限的情况下或者用户为超级用户的情况下，才能被删除或重命名

文件许可位中，SUID、SGID、粘滞位在最前面依次排列。对于 SUID 程序，文件许可位的八进制表示最高位就是 4。

受控调用

在执行某些系统调用时，Unix需要超级用户权限。

SUID（SGID）程序与属主或属组的有效UID或有效 GID一起运行，拥有暂时的或者受限制的访问权限。

这些访问权限通常情况下是不赋予普通用户的。

```
progf1:
printf("China");
exec("progf2");
printf("America");
return;
```

```
progf2:
printf("England");
exec("progf3");
printf("Canada");
return;
```

```
progf3:
printf("Australia");
return;
```

```
grp2:x:681:usr5,usr6,usr7,usr8,usr9

--x--x--x   usr1  grp1  ..... progf1
--x--s--x   usr6  grp2  ..... Progf2
--s--x--x   usr5  grp2  ..... Progf3
rw-r-----   usr5  grp2  ..... filex
```

- 假设：用户usr1 执行程序progf1启动进程P。
- 问题：进程P显示China时，对文件filex拥有什么访问权限？
- 结果：没有任何权限。

```
progf1:
printf("China");
exec("progf2");
printf("America");
return;
```

```
progf2:
printf("England");
exec("progf3");
printf("Canada");
return;
```

```
progf3:
printf("Australia");
return;
```

```
grp2:x:681:usr5,usr6,usr7,usr8,usr9

--x--x--x   usr1  grp1  ..... progf1
--x--s--x   usr6  grp2  ..... Progf2
--s--x--x   usr5  grp2  ..... Progf3
rw-r-----   usr5  grp2  ..... filex
```

- 假设：用户usr1 执行程序progf1启动进程P。
- 问题：进程P显示England时，对文件filex拥有什么权限？
- 结果：读。

```
progf1:
printf("China");
exec("progf2");
printf("America");
return;
```

```
progf2:
printf("England");
exec("progf3");
printf("Canada");
return;
```

```
progf3:
printf("Australia");
return;
```

```
grp2:x:681:usr5,usr6,usr7,usr8,usr9

--x--x--x   usr1  grp1  ..... progf1
--x--s--x   usr6  grp2  ..... Progf2
--s--x--x   usr5  grp2  ..... Progf3
rw-r-----   usr5  grp2  ..... filex
```

- 假设：用户usr1 执行程序progf1启动进程P。
- 问题：进程P显示Australia时，对文件filex拥有什么权限？
- 结果：读+写。

强制访问控制（SELinux优点、类型、域、域切换）

优点：

- MAC（强制访问控制） 对访问的控制彻底化
- TE（类型强制） 对于进程只赋予最小的权限
- domain迁移 防止权限升级

- RBAC（基于角色的访问控制） 对于用户只赋予最小的权限

类型

普通文件、目录、进程、套接字、文件系统

域切换条件

- 进程的新的工作域必须拥有对可执行文件的类型的entrypoint访问权限
- 进程的旧的工作域必须拥有对入口点程序的类型的execute访问权限
- 进程的旧的工作域必须拥有对进程的新的工作域的transition访问权限

Android安全

Android**系统架构**

需要知道，Android 系统在应用程序和内核之间，有一个虚拟机构成的运行时库，一个应用，一个虚拟机实例，一个进程。

Android**安全机制**

- 沙箱：一个应用程序的进程就是一个安全的沙箱（隔离）
- 全新啊：指定保护级别

安全机制：

- 用户ID：一个应用程序就是一个用户，每一个用户拥有一个独立的UID
- 权限：又来描述应用程序是否有做某件事的权利，在安装时就确定了
- 签名：每一个Android应用程序都必须有一个数字签名，目的是为了在应用程序的开发者和应用程序之间建立一种信任关系

Windows安全

WinLogon

总是运行着一个以SYSTEM为主角的登录进程

LSA

本地安全权威

- 用户登录时，检查有用户账户并创建访问令牌
- 负责审计功能

SAM

安全账户管理员

- 维护用户账户数据库
- 在本地用户认证期间，LSA将使用该数据库
- 存储口令（hash值）

注册表

注册表是Windows配置数据的中央数据库，其中表项称为键Key，注册表的键指向操作系统搜索特定可执行文件的位置。

域

是共享公用用户账户数据库和安全策略的计算机集合，类似访问控制中间结构的组，使用域来实现一次签到以及集中式的安全管理。

活动目录

windows的数据组织方式，活动目录可以被看作一棵由特定类型对象所构成的树。

功能：

- 基础网络服务
- 计算机管理
- 用户服务
- 资源管理
- 桌面配置
- 应用系统支撑

主角

是安全策略中的实体。在windows中，主角可以是：本地用户、别名、域用户、组、机器

主角有一个username和安全标识符SID。

每个Windows机器拥有其内置的权威LSA，LSA所创建的用户成为本地用户，域控制器DC为一个域提供安全服务，每个域有其对应的域控制器权威DCA。

主角的作用域：

- 本地主角：本地管理，只对本地计算机可见
- 域主角：域控制器上的管理员管理，对域中的所有计算机都是可见的。如：域用户、组、别名

主体

在Windows中，进程和线程是主体。安全凭证存放在访问令牌中

令牌

令牌包括一系列主角和其他安全属性

令牌内容：

安全描述符

用于访问控制。内容包括：

- 属主SID
- 主组
- 自主访问控制列表DACL：定义了谁将被给予或拒绝访问，表项是肯定或者否定，表示允许或拒绝
- 系统访问控制列表SACL：定义了审计策略，表项可以同时是肯定的和否定的，因为这里表示的是审计策略，同时肯定和否定表示——被允许的访问和被拒绝的访问都记录下来

受限上下文

以受限令牌运行的进程就是受限上下文，其设计原则就是访问控制的设计原则中的最小特权原则

向令牌添加受限SID后，只有在SID和受限SID都被允许访问的时候，具有受限令牌的进程才能获得访问权限

受限令牌

访问令牌，去掉特权，禁止组，添加一个代表程序的受限SID，被添加到程序所要访问的所有对象的DACL中

访问控制

在 Windows 系统中：

- **主体**：主体的凭证，包括其主角，存储于令牌中。
- **对象**：对象的安全参数存储于安全描述符中。
- **请求的访问权限**：主体要求的访问操作以访问掩码形式给出。

在访问控制过程中，需要检查对象安全描述符中的 DACL。DACL 是包含 ACE（访问控制表项）的列表，每个 ACE 的格式如下：

- 类型：允许或拒绝
- 标志
- ObjectType：定义对象类型的 GUID。只有具有匹配 ObjectType 或不含 ObjectType 的 ACE 会被评估。（不含 ObjectType 的 ACE 适用于所有对象）
- InheritedObjectType：只有那些具有匹配继承对象类型或不含继承对象类型的 ACE 会被复制到 ACL 中。
- 访问权限
- 主角 SID：ACE 应用的主角

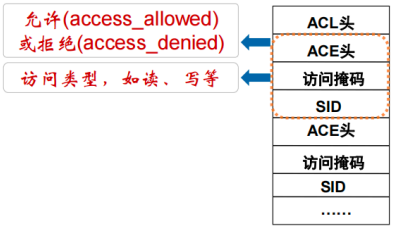
访问控制过程：获取属主许可；遍历 DACL；当主体令牌包含匹配的 SID 时检查 ACE。

- 允许访问：所有请求权限都已获得。
- 拒绝访问：找到相应的否定 ACE 或到达 ACE 末尾。
- 为了使否定ACE优先于肯定的ACE，他们必须被放置在DACL的顶端

DACL

规定了谁可以用什么方式访问该对象

安全描述符中的DACL：访问控制表项（access control entries， ACE）的列表



访问控制列表示意图

SACL

规定了哪些用户的哪些操作应该被记录在审计日志中

SACL由管理员设置。

SACL中的ACE格式

- 类型：肯定（审计允许）或否定（审计拒绝）
- 受托者：一个SID（个人、组、别名）
- 掩码：许可（32位掩码）

一个ACE可以同时包含肯定和否定权限。

DEP

数据执行保护

基本原理：将数据所在的内存页标记为不可执行。当程序溢出并成功转入 shellcode 时，程序会尝试在数据页面上执行指令，此时 CPU 会抛出异常，而不会执行恶意指令。

数据库安全

关系数据库

表的集合

视图

命名的导出关系，由其他已命名的关系定义，本身不存储数据（与快照区别）

快照

命名的导出关系，由其他已经命名好的其他关系定义，拥有自己独立的存储的数据

存储过程

存储过程（Stored Procedure）是在大型数据库系统中，为完成特定功能而编译并存储在数据库中的一组SQL语句。用户通过指定存储过程的名称和参数（如果有）来执行它。

函数

数据库函数是指当需要分析数据清单中的数值是否符合特定条件时，使用数据库工作表函数

触发器

触发器是一种特殊类型的存储过程，与之前介绍的存储过程不同。

触发器主要通过事件触发执行，而存储过程可以直接通过名字调用。

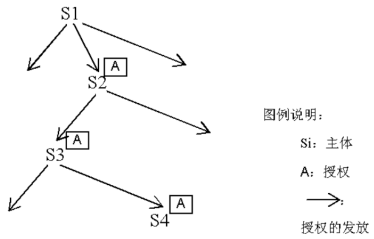
主要功能：实现由主键和外键所不能保证的复杂的参照完整性和数据的一致性

委托授权

一个主体把对客体的授权的发放和回收传递给另一个主体，使得另一个主体能够发放和回收相应客体的授权

递归回收

S1发出的回收A的操作不但回收S2的A授权而且回收S3和S4的A授权



否定式授权

- 问题：S1不向S2发放授权A，但S3可能向S2发放，这样S2依然会拥有授权A
  - S1向S2发放某种授权，屏蔽其他人发放的授权？
- 常规授权： GRANT (S, O, A)
  - 授权主体S对客体O进行A类型的访问
  - GRANT DELETE ON emp TO bob
- 否定式授权： GRANT (S, O, NO-A)
  - 禁止主体S对客体O进行A类型的访问
  - GRANT NO-DELETE ON emp TO bob

解决授权冲突的原则

否定优先原则：如果一个主体既拥有在某个客体上进行某种操作的肯定式授权，也拥有在该客体上进行同样操作的否定式授权，那么，否定式授权发挥作用，肯定式授权不起作用

个体优先原则：如果某个小组及其组中的某个用户都拥有同一个客体上的同一种访问类型的授权，并且，两者所拥有的授权是冲突的，那么，用户拥有的授权发挥作用，小组拥有的授权不起作用

视图的优点

作用：支持基于内容的访问控制

- 减少用户查询的复杂度、
- 限制某个视图只能访问基表中的部分数据、
- 增强上下文依赖和数据依赖的安全策略、
- 实施受控调用、
- 安全的视图可以代替安全标签、
- 数据可以轻松的重新分类

用户发放授权和角色发放授权的区别

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

三元组安全标签

<等级、类别、组别>

基于标签的访问判定

读：用户标签的等级必须大于等于 记录标签中的等级 用户标签中的组别至少必须包含记录标签中的一个组别，并且用户拥有对该组别的读权限 用户标签的类别，必须包含记录标签中的所有类别

写： 记录标签中的等级必须小于或等于用户会话标签中的等级

记录标签中的等级必须大于或等于用户的最小等级

用户会话标签中的区别至少必须包含记录标签中的一个组别，并且用户拥有对该组别的写权限

用户会话标签中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中的所有类别拥有写权限。若记录标签含有组别，则用户必须对记录标签中的所有类别拥有读权限。

统计数据库的推理

统计数据库，用于统计分析目的的数据库

推理方法：

- 借助求和结果进行推理
- 借助记录个数进行推理
- 借助平均值进行推理
- 借助中位数进行推理
- 借助智能填充进行推理
- 借助线性特性进行推理

跟踪攻击

允许对单个元组追踪其信息的查询谓词T，称之为该元组的单个追踪者

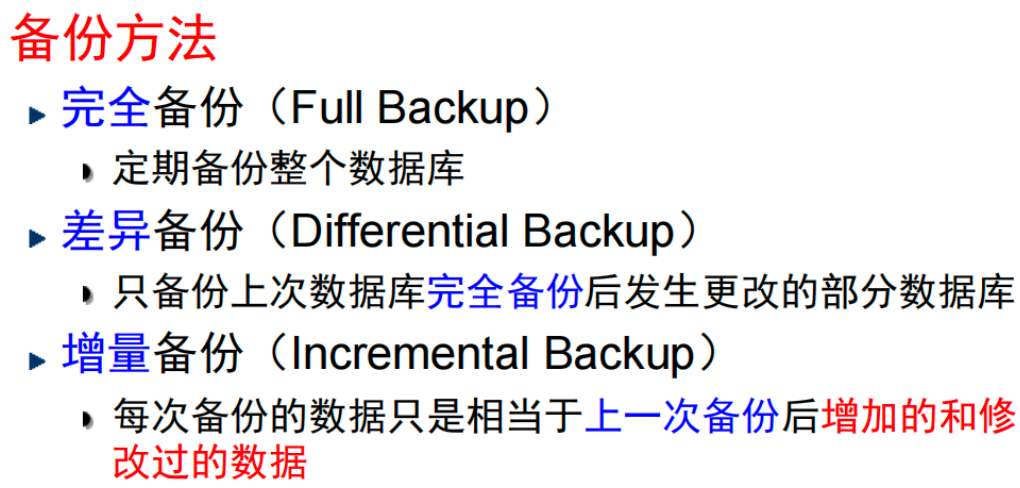
一个通用追踪者是一个谓词，能够用来找到任何不允许查询的答案

差分隐私

差分隐私需要做到的，就是使得攻击者的知识，不会因为这些新样本的出现而发生变化

差分隐私就是保留统计学特征的前提下，去除个体特征以保护用户隐私

完全备份、差异备份、增量备份



系统可信检查机制

系统可信引导和系统安全引导的异同

系统可信引导：

- **目标**：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破坏，从而确保引导起来的操作系统的完整性是有保障的。
- **方法**：需要对组件的完整性进行验证，通常通过计算组件的哈希值并与预存的原始指纹进行对比来完成验证。如果指纹匹配，则组件被认为是完整的，控制权可以交给下一个组件 。
- **特性**：注重在引导过程中每个组件的完整性验证，保证每个组件在接管控制权时都是可信的 。

系统安全引导：

- **目标**：不仅要确保引导起来的操作系统内核是完整的，还要确保操作系统内核能够顺利引导起来。
- **方法**：在系统可信引导的基础上，增加了系统恢复功能。即使某些组件出现问题，通过预存的副本和恢复机制确保系统能够继续引导。
- **特性**：除了完整性验证外，还包含恢复功能，确保系统在引导过程中即使遇到问题也能通过恢复机制完成引导。

相同点：

- 两者都致力于保证系统引导过程的安全性和完整性。
- 都涉及对系统组件的完整性验证，通过哈希值或数字签名等方式验证组件是否被篡改。

进程完整性的体现

进程完整性主要体现在以下几个方面：

1. **在初始状态中的完整性**：进程在启动时，系统会计算和检查程序的哈希值，以确保程序未被篡改。
2. **在中断过程中的完整性**：进程在运行过程中可能会被中断，系统需要确保在中断和恢复时进程的完整性没有被破坏。
3. **呈现于存储介质时的完整性**：进程的数据和代码存储在不同的介质中，包括片上缓存、片外内存和磁盘，系统需要验证这些存储介质中数据和代码的完整性。
4. **输出结果的完整性**：进程的输出结果在传输过程中需要确保其真实性和完整性，通常通过数字签名来实现。

完整性验证的方法主要包括指纹法（哈希值法），在不同模式下（普通模式和篡改响应模式）进行相应的验证。

BLP模型

12.1.1 安全策略、安全机制、安全模型之间的关系

12.1.2 状态集 (掌握)

BLP 模型的状态集为 $B \times M \times F$.

1. B：
   - $B \subset (S \times O \times A)$，是当前访问的集合.
   - $b = (s, o, a) \in B$，表示主体 s 正在客体 o 上执行操作 a.
2. M：
   - M 是所有访问许可矩阵的集合.
   - $m = (m_{so}) \in M$，其中 $s \in S, o \in O$.
3. F：
   - $F \subset (L_S \times L_C \times L_O)$，是是安全级别分配的集合.
   - $f = (f_s, f_c, f_o) \in F$
   - f_s：给出了每个主体可以拥有的最高安全级别

- f_c：给出了每个主体当前的安全级别
- f_o：给出了每个客体的安全级别
- 一个主体的当前级别不能高于他的最高级别，所以有 $f_c \leqslant f_s$(其实更严谨的说法应该是偏序，不过这门课的偏序记号也就用我们常用的大小等于符号了). 称作“f_s 支配 f_c”.

12.1.3 BLP 模型的安全策略

BLP 模型针对保密性，在 BLP 模型中，安全级别就是保密性级别(而 Biba 模型中，安全级别是完整性级别). 简单概括 BLP 保密性的安全策略就是：

- 无向上读：只能读安全级别/密级不超过自己的信息 (客体).
- 无向下写：避免将安全级别高的信息流向安全级别低的客体，从而使得低安全级别的主体读到了不应该读到的安全级别高的信息.

关于 BLP 模型的安全策略，需要掌握三种：

- 简单安全性 (ss-property)
- 星特性 (-property)
- 自主安全性 (ds-property)

其中简单安全性和星特性称为强制 BLP(安全) 策略.
这三种安全策略，都需要熟练掌握. 在考试题目中，可能作为简答题，直接问，比如“BLP 的星特性的内容 (很容易漏掉第二条，这个也是老师反复强调的)”；也可能在综合题目中考察.
在记忆这些安全策略时，要注意理解其意义，并且注意其中的不同，比如描述中的操作可能是不同的 (星特性第二条，前面的操作 a 是添加/写，后面的操作 a' 是读/写).

12.1.4 简单安全性 (掌握)

如果对于每一个元素，均有 $(s, o, a) \in b$，访问操作 a 是读/写，主体 s 的安全级别控制客体 o 的安全级别，即 $f_s(s) \geqslant f_o(o)$，那么那么状态 (b, m, f) 满足简单安全性.
这里课件上写的是 $(s, o, a) \in b$，我觉得是有问题的，和前面状态集部分的定义是矛盾的，我认为应该是 $b = (s, o, a) \in B$. 但是教材也是这样写的，我当时没有再继续深究下去. 这个供参考，如果有必要可以和老师确认一下.

12.1.5 星特性 (掌握)

简单安全性只实现了无向上读，星特性在此基础上增加了无向下写的策略.

- 如果对于每一个元素，均有 $(s,o,a) \in b$，访问操作 a 是添加或写，主体 s 的当前级别受客体 o 的当前级别控制，即 $f_c(s) \leqslant f_o(o)$，那么状态 (b,m,f) 满足星特性；
- 如果存在一个元素，有 $(s,o,a) \in b$，访问操作 a 是添加或写，那么对于所有客体 o'，$b=(s,o',a') \in B$，a' 是读或写，则必须有 $f_o(o') \leqslant f_o(o)$.

星特性的内容由两条陈述组成，注意的是二者缺一不可，如果遇到简答题，两条都需要答上. 这一点老师在课上强调过，是一个易错点.

12.1.6 自主安全性 (掌握)

自主安全性对应 DAC，也就是对应访问控制矩阵 (或者说 ACL). 所以不难记忆其内容：
如果对于每一个元素，均有 $(s,o,a) \in b$，$a \in m_{so}$，那么状态 (b,m,f) 满足自主安全性.

12.1.7 基本安全定理 (理解、掌握)

BLP 这一章课件的前面介绍了状态机的概念，及其在安全理论中的应用. 利用状态机模型可以提出安全系统的设计方法：

- 定义安全的状态集合
- 初始状态安全
- 所有状态转移都是安全的

基本安全定理是由状态机模型理论产生的. 首先定义状态安全和状态转移安全的定义：

- 如果状态 (b,m,f) 满足简单安全性、星特性、自主安全性，那么称状态 (b, m, f) 是安全的.
- 如果状态 $v_1=(b_1,m_1.f_1)$ 和状态 $v_2=(b_2,m_2.f_2)$ 都是安全的，那么称从状态 v_1 到状态 v_2 的迁移是安全的.

所以，可以得到状态迁移安全的形式化描述：

- 每个 $(s,o,a) \in b_2 \backslash b_1$ 满足 f_2 的简单安全性 (星特性、自主安全性)，($b_2 \backslash b_1$ 表示集合 b_2 和 b_1 的差集) 以及
- 如果 $(s,o,a) \in b_1$ 不满足 f_2 的简单安全性 (星特性、自主安全性)，那么 $(s,o,a) \notin b_2$.

有了上面的准备工作，我们可以给出基本安全定理：
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.

安全模型

Biba模型

13.1.1 Biba 模型中的安全级别

Biba 模型中的安全级别 (也就是完整性等级) 的含义：

- 安全等级就是可靠性的等级. 安全级别越高，可靠性越高.
- 用于衡量完整性等级的术语是“可信度”. 高级别的客体比低级别的客体更“可信”.

因此，对应的安全策略应该是：

- 无向上写：禁止“干净的”高级别实体被“脏的”低级别实体污染.

注意这里和 BLP 模型的区别. 其中这里只讨论写，因为这里关心的不是保密性而是完整性，只有写操作才能影响完整性，而读操作并不影响完整性，所以其安全策略中并不关心读的问题.

13.1.2 简单完整性 (掌握)

无向上写：如果主体 s 可以修改客体 o，则 $f_s(s) \geqslant f_o(o)$.

13.1.3 星特性 (掌握)

如果主体 s 可以读客体 o，那么仅当 $f_o \geqslant f_o(o')$ 时，s 可以写另一客体 o'.

13.1.4 Biba 模型和 BLP 模型的对比 (总结)

- 针对不同的安全属性：BLP 模型针对保密性；Biba 模型针对完整性.
- 策略不同：BLP 模型为保密性安全策略——无向上读、无向下写；Biba 模型为完整性安全策略——无向上写.
- Biba 模型有动态完整性级别 (下一节介绍).

13.1.5 动态完整性级别 (掌握)

1. 主体低水印性：主体 s 可以读 (查看)任何完整性级别上的客体 o，主体的新的完整性级别是 $\inf\{f_s(s), f_o(o)\}$.
2. 客体低水印性：主体 s 可以修改任何完整性级别上的客体 o，客体的新的完整性级别是 $\inf\{f_s(s), f_o(o)\}$.
3. 调用性：一个“脏的”主体 s_1 不能通过调用主体 s_2 来间接访问一个“干净的”的客体——主体 s_1 可以调用主体 s_2，仅当 $f_s(s_1) \geqslant f_s(s_2)$.
4. 环属性：主体 s_1 可以读任何完整性级别上的客体 o，它只能在 $f_s(s_1) \geqslant f_o(o)$ 时修改 o，仅当 $f_s(s_1) \leqslant f_s(s_2)$ 时可以调用主体 s_2.

Chinese Wall

为了理解中国墙模型，我们可以先了解一下其背景. 它模拟了咨询公司的访问规则：咨询公司的分析员必须保证他们与不同客户的交易不会引起利益冲突，即一定不能有引起利益冲突的信息流.
主体：分析员 (集合用 S 表示)；客体：信息条目 (集合用 O 表示)；公司集合：C.

14.1 主要内容

14.1.1 公司数据集，利益冲突类，安全标签

- 公司数据集：有关同一公司的所有客体的集合. 函数 $y : O \to C$ 给出了每个客体的公司数据集.
- 利益冲突类：相互竞争的公司. 函数 $x : O \to P(C)$ 给出了客体 o 的利益冲突类.
- 安全标签：(利益冲突类, 公司数据集)，即 $(x(o), y(o))$.

14.1.2 简单安全性 (掌握)

仅当客体请求属于以下情形时访问才被允许：

- 用户已经拥有的一个公司数据集.
- 一个完全不同的利益冲突类.

用矩阵 $N = (N_{so})_{s\in S, o\in O}$ 记录访问情况，$(N_{so}) = true$ 表示主体 s 访问过客体 o.
我们给出简单安全性的形式化描述：
主体 s 允许访问客体 o，仅当对于所有 $(N_{so'}) = true$ 的客体 o' 有 $y(o) = y(o')$ 或 $y(o) \notin x(o')$.

14.1.3 星特性 (掌握)

主体 s 被允许对客体 o 进行写访问，仅当 s 对 $y(o) \neq y(o')$ 和 $x(o') \neq \phi$ 的客体 o' 没有读访问权.

信息流模型

15.1 主要内容

15.1.1 强信息流、弱信息流 (掌握)

1. 强信息流 (显式信息流)：
   - x 的值直接影响 y 的值，例如赋值.
   - 形式化描述：$y = f(x)$.
2. 弱信息流 (隐式信息流)：
   - 信息从 x 流向 y，但 x 的值不直接影响 y 的值
   - 弱信息流不是因为使用 x 值进行了赋值，而是语句中基于 x 值的控制流产生了弱信息流

15.1.2 隐式信息流的信息量计算 (掌握)

x 到 y 的信息流用给定 y 的值后的 x 的信息量平均值的变化 (条件熵) 来度量.

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

15.1.3 隐信道 (理解、掌握)

见 2.1.6 小节.

15.2 作业题目解析

15.2.1 练习 9.6——条件熵计算

令 X 为一个 4 比特变量，可在 0 到 15 中间等概率地取值. 给定条件“$IF \quad X > 7 \quad THEN \quad Y := 1$”和初值“$Y = 0$”，计算条件熵 $H_Y(X)$.

解：
条件熵计算是信息论中的基本问题. 计算方法比较清晰，就是按照定义，将概率分布代入公式计算即可.
由题意知，有条件概率：

- 对于 $0 \leqslant i \leqslant 7$：$p(X = i \mid Y = 0) = \frac{1}{8}$，$p(X = i \mid Y = 1) = 0$.
- 对于 $8 \leqslant i \leqslant 15$：$p(X = i \mid Y = 0) = 0$，$p(X = i \mid Y = 1) = \frac{1}{8}$.

代入公式有：

$$H_Y(X) = -8 \times \frac{1}{2} \times \frac{1}{8}\log\frac{1}{8} \times 2 = 3\text{比特}$$

安全评估

安全评估框架

1. 评估目标对象（Target of Evaluation, TOE）：

- 产品：已出售的部件
- 系统：满足特定应用需求的产品集合

2. 评估目标：

- 评估（Evaluation）：判断产品是否具备声明的安全属性
- 认证（Certification）：确认已评估产品是否适合特定应用场景
- 鉴定（Accreditation）：确定已认证产品能否在指定场合使用

3. 评估方法：

- 面向产品：检查和测试产品，适用于发现问题
- 面向过程：检查文档和开发过程，成本低且易于获得可重复结果
- 可重复性（Repeatability）和可再现性（Reproducibility）是评估方法必须满足的要求。

TCSEC

针对保密性

安全级别	定　义
A1	验证设计（Verified Design）
B3	安全域（Security Domains）
B2	结构化保护（Structural Protection）
B1	标记安全保护（Labeled Security Protection）
C2	受控的存取保护（Controlled Access Protection）
C1	自主安全保护（Discretionary Security Protection）
D	最小保护（Minimal Protection）

1. D 类：最低保护级（不符合要求的系统）
2. C 类：自主保护级（采用自主访问控制 DAC）
   - C1 级：TCB 通过隔离用户与数据，使用户具备自主安全保护能力。
     注：TCB 是可信计算基，已在访问监控器部分介绍。考试可能会直接问 TCB 的定义。
   - C2 级：比 C1 级具有更细粒度的自主访问控制，被认为是商业应用最合理的安全级别。目前大多数商业操作系统都是 C2 级别，基本实现了完整的 DAC。
3. B 类：强制保护级（采用强制访问控制）
   - B1 级：TCB 通过隔离用户与数据，使用户具备自主安全保护能力。
     注：TCB 是可信计算基，已在访问监控器部分介绍。考试可能会直接问 TCB 的定义。
   - B2 级：将 B1 系统中的自主和强制访问控制扩展到所有主体和客体，并要求对隐蔽通道进行分析。
   - B3 级：在 B3 系统中，TCB 必须满足访问监控器需求。
4. A 类：验证保护级（形式化的安全验证）
   - A1 级: 要求用形式化设计规范和验证方法来分析系统，以确保 TCB 按设计要求实现。

TCSEC特点：只考虑保密性，不考虑完整性、可用性

ITSEC

- EO：最低
- E1：测试
- E2：配置控制和分配管理
- E3：访问详细设计和源码
- E4：脆弱性详细分析
- E5：设计与源码对应明确
- E6：设计与源码形式一致

CC

- EAL1：功能测试
- EAL2：结构测试
- EAL3：系统测试和检查
- EAL4：系统设计、测试和复查
- EAL5：半形式化设计和测试（隐蔽通道分析，和TCSEC 的B2正好是对应的）
- EAL6：半形式化验证的设计和测试（访问监控器，和TCSEC 的B3正好是对应的）
- EAL7：形式化验证的设计和测试

TCSEC、ITSEC、CC 的安全级别对应关系

CC	TCSEC	ITSEC
–	D	E0
EAL1	–	–
EAL2	C1	E1
EAL3	C2	E2
EAL4	B1	E3
EAL5	B2	E4
EAL6	B3	E5
EAL7	A1	E6

网络安全等级保护

等级保护的内涵

- 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、分阶段实施保护
- 保障信息安全和系统安全正常运行， 维护国家利益、公共利益和社会稳定以及公民、法人和其他组织的合法权益

等保2.0

技术要求：

1. 安全物理环境
2. 安全通信网络
3. 安全区域边界
4. 安全计算环境
5. 安全管理中心

管理要求

1. 安全管理制度
2. 安全管理机构
3. 安全人员管理
4. 安全建设管理
5. 安全运维管理

等级保护制度的主要内容

- 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）信息安全产品实行分等级管理
- 对信息系统中发生的信息安全事件实行分等级响应、处置

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方法

- 业务信息
- 系统服务
- 受侵害客体
- 侵害程度

等级测评流程

- 测评准备
- 方案编制
- 现场测评
- 报告编制

密码应用安全性评估

密码应用安全性评估的含义

- 合规性：使用的密码算法、协议、密钥管理符合国家法律法规和标准规定，密码产品或服务经过国家密码管理局核准和认证机构认证合格
- 正确性：密码算法、密码协议、密钥管理、密码产品和服务使用正确，即按密码相关的国家和行业标准进行正确设计和实现，密码产品和服务的部署、使用正确
- 有效性：密码保障系统在系统运行中能狗发挥实际效用，满足了信息系统的安全需求，解决信息系统面临的安全问题

密码应用基本要求的八大安全类

技术要求：

- 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

管理要求

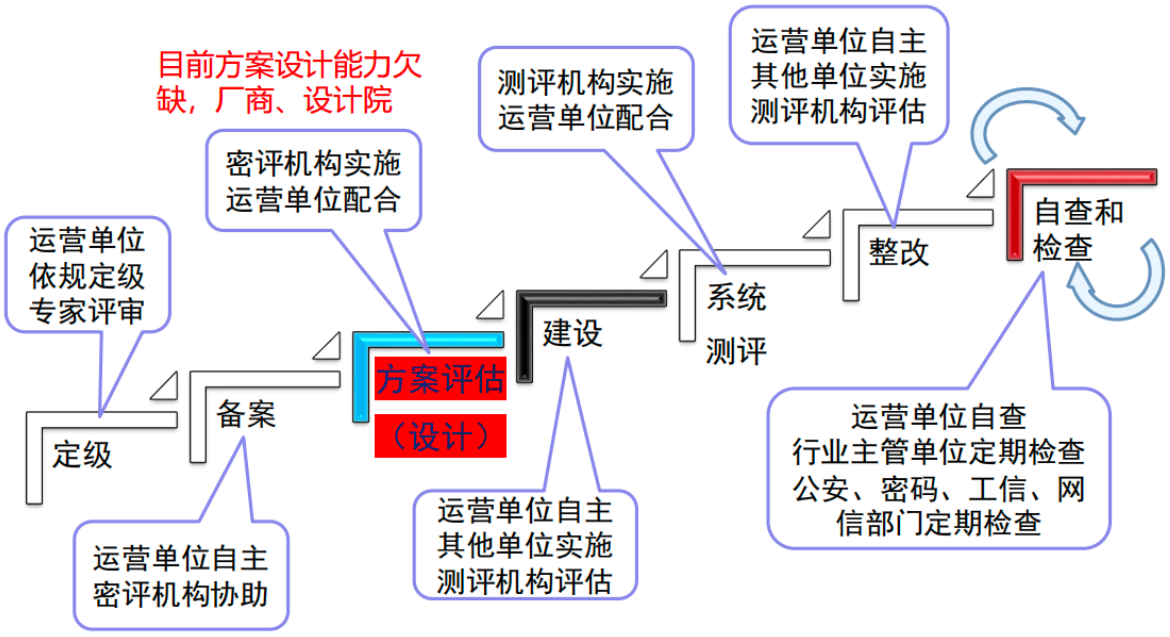
- 管理制度

- 人员管理
- 建设运行
- 应急处理

技术标准中应、宜、可的区别

- “应”表示应该或要求，是一种要求型描述，表明符合标准需满足的条件。
- “宜”表示推荐或建议，是一种推荐型描述，意味着该条款是首选但不是必须的。
- “可”表示可以或允许，是一种陈述型描述，指在标准范围内所允许的条款。

等保和密评的区别



云计算安全

云计算的主要特性

- 按需自助服务
- 泛在接入
- 资源池化
- 快速伸缩性
- 服务可计量

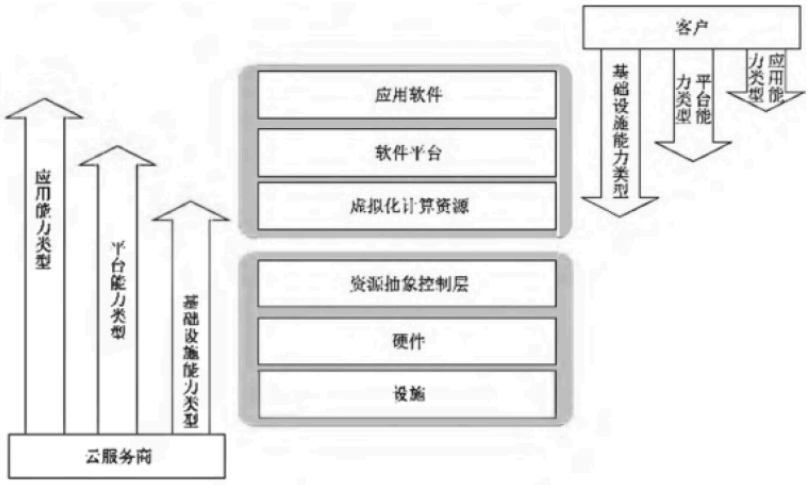
云计算的服务模式

- SaaS（软件即服务）：云服务商提供运行在云基础设施上的应用软件，客户无法管理或控制底层资源，如网络、服务器、操作系统和存储等，但可以对应用软件进行有限的配置管理。
- PaaS（平台即服务）：云服务商提供运行在云基础设施上的开发和运行平台，客户不能管理或控制底层资源，但可以配置和控制自己部署的应用及其运行环境。
- IaaS（基础设施即服务）：云服务商提供虚拟计算机、存储和网络等计算资源，并提供访问这些资源的接口。客户不能管理或控制底层基础设施，但能控制自己部署的操作系统、存储和应用，以及部分网络组件，如主机防火墙。

云计算的部署模式

- 私有云：仅供特定客户使用的云计算平台。
- 公有云：对所有客户开放的云计算平台。
- 社区云：限定特定客户群体使用，群体内客户具有共同属性（如职能、安全需求、策略等）
- 混合云：两种或多种部署模式的组合。

云服务商和客户之间的安全责任划分



在图中，云计算的设施层（物理环境）、硬件层（物理设备）和资源抽象控制层完全由云服务商掌控，所有安全责任也由其承担。应用软件层、软件平台层和虚拟化计算资源层的安全责任则由双方共同承担。越靠近底层的云服务（如基础设施即服务），客户需要承担更多管理和安全责任；反之，云服务商需承担更多。

此外，考虑到云服务商可能依赖其他组织提供的服务，如SaaS或PaaS供应商可能依赖IaaS供应商的基础资源，在这种情况下，一些安全措施将由这些其他组织提供。

基于代码的访问控制

堆栈遍历

当函数请求访问一个受保护的资源，堆栈遍历用来确定调用者是否有需求的许可

惰性计算

只有当访问一个资源需要一个许可时，才去评估授予的许可

热情计算

前摄性地估算调用者的许可

许可断言

如果这些策略被严格执行，受控调用的选项将受到严重约束。

因此，通常提供者会提供用于代码许可的断言（assert）。

- 许可断言附着在当前栈帧上，并在从调用部件返回时撤销。
- 堆栈遍历在遇到包含许可断言的帧并授予该权限时终止（此时所有被检查的帧都具有该权限）
- 如果仍有要求的权限未被断言，堆栈遍历将继续进行。

断言允许不可信任的调用者调用可信的方法。

入侵检测

IDS

入侵检测系统：进行入侵检测的软件与硬件的组合

IPS

入侵防御系统：能够监视网络或网络设备的网络资料传输行为，能够即时的中断、调整或隔离一些不正常的或是具有伤害性的网络资料传输行为

DPI

深度包检测：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检测和控制技术

DFI

深度流检测：采用一种基于流量行为的应用识别技术，即不同的应用类型体现在会话连接或数据流上的状态各有不同

DFI技术正是基于这一系列流量的行为特征，建立流量特征模型，通过分析会话连接流的包长、连接速率、传输字节量、包与包之间的间隔等信息来与流量模型对比， 从而实现鉴别应用类型

态势感知

态势感知是一种基于环境的、动态、整体地洞悉安全风险的能力

- 基于安全大数据
- 从全局视角提升对安全威胁的识别、分析和响应能力
- 目的是为了决策与行动，实现安全能力

入侵检测方法

- 异常检测：首先总结正常操作的特征（用户轮廓），当用户活动显著偏离正常行为时，即被认为是入侵。此方法漏报率低，但误报率高。
- 误用检测模型：收集异常操作的行为特征，建立相关特征库。当监测到的用户或系统行为与库中记录匹配时，系统认为这是入侵。此方法误报率低，但漏报率高。

入侵检测系统分类

- 基于主机：系统根据运行所在的主机获取数据，并保护该主机。
- 基于网络：系统通过网络传输的数据包获取数据，保护网络运行。
- 混合型：主机+网络

应急响应与灾备恢复

IATF

信息保障技术框架，

一个核心思想：

- 纵深防御（深度防护）

三个核心要素：

- 人（管理）核心
- 技术 重要手段
- 操作（运维）主动防御体系

四个焦点领域（信息安全保障区域）：

- 网络和基础设施
- 区域边界
- 计算环境
- 支撑性基础设施

三个安全基本原则：分区分域、分层防御、适度安全

这四个区域的划分同样适用于信息系统的安全评估，提供了一种实现系统安全要素和安全服务的层次结构（例如等保2.0）。

应急响应的概念

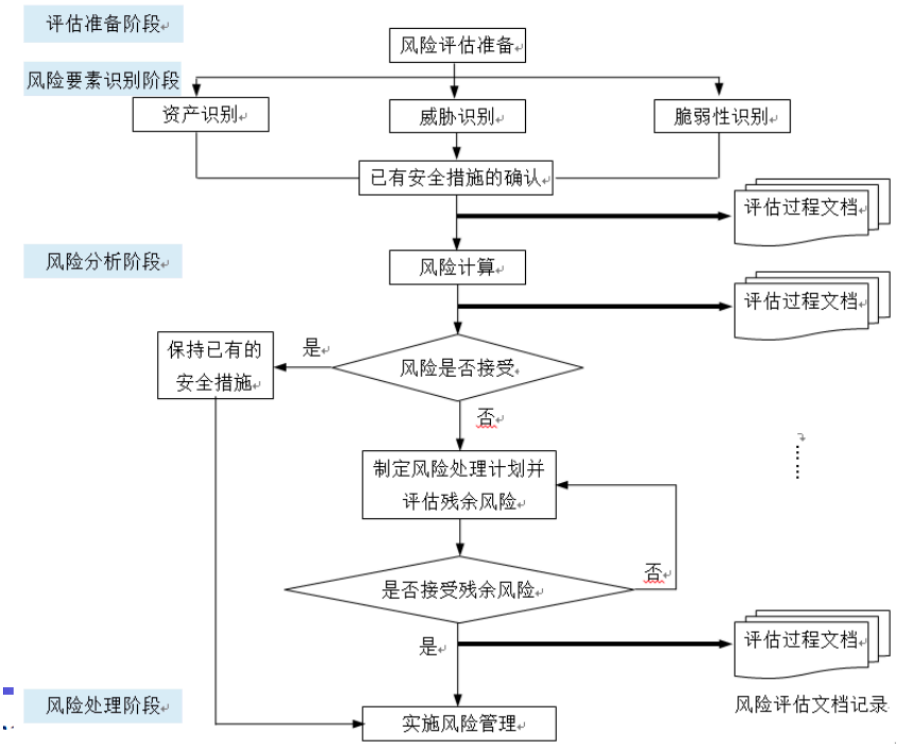
- 指一个组织为应对各种网络安全事件所做的准备，以及在事件发生后采取的措施。
- 其目的是尽量减少和控制安全事件造成的损失，提供有效的响应和恢复指导，并努力防止安全事件发生。

应急管理的重点不在于应急，而在于预防。

应急响应的过程

- 1. 应急准备
  - （1）应急响应组织架构
  - （2）应急响应制度
  - （3）风险评估与改进
  - （4）划分应急事件级别
  - （5）应急响应预案制定
  - （6）培训和演练
- 2. 监测与预警
  - （1）日常监测与预警
  - （2）核实与评估
  - （3）应急响应预案启动
- 3. 应急处置
  - （1）应急调度
  - （2）排查与诊断
  - （3）处理与恢复
  - （4）事件升级
  - （5）持续服务
  - （6）事件关闭
- 4. 总结与改进
  - （1）应急工作总结
  - （2）应急工作审核
  - （3）应急工作改进

风险评估过程



灾难恢复的概念

为了将信息系统从灾难造成的故障或瘫痪状态恢复到正常运行状态，并将其支持的业务功能从灾难造成的不正常状态恢复到可接受状态，而设计的活动和流程

容灾备份的概念

利用技术、管理手段以及相关资源确保既定的关键数据、 关键数据处理信息系统和关键业务在灾难发生后可以恢复和重续运营的过程

RAID 0、RAID 1、RAID 10、RAID 5

RAID：独立冗余磁盘阵列

- RAID是将多块独立的物理磁盘按一定方式组合成一个磁盘阵列（逻辑磁盘），通过冗余信息进行数据存储。当某个磁盘发生数据损坏时，可以利用冗余信息恢复数据，从而提供比单个磁盘更大的存储容量、更高的可靠性和更快的访问速度。
- RAID 0: 没有冗余或错误修复能力
- RAID 1:磁盘镜像，利用率50%
- RAID 10: 称为镜像阵列条带，冗余率100%
  - 像RAID 0一样，数据跨磁盘分布；像RAID 1一样，每个磁盘都有一个镜像磁盘。因此，RAID 10也叫RAID 0+1。
- RAID 5：通过奇偶校验信息来实现数据恢复。折衷方案
  - RAID 5 是一种兼顾存储性能、数据安全和成本的解决方案。
  - 至少3块硬盘
  - 只允许一块硬盘故障

类型	读写性能	安全性	磁盘利用率	成本	应用方面
RAID0	最好（因并行性而提高）	最差（完全无安全保障）	最高（100%）	最低	个人用户
RAID1	读和单个磁盘无分别，写则要写两边	最高（提供数据的百分之百备份）	差（50%）	最高	适用于存放重要数据，如服务器和数据库存储等领域。
RAID5	读：RAID 5 = RAID 0（相近似的数据读取速度） 写：RAID 5<对单个磁盘进行写入操作（多了一个奇偶校验信息写入）	RAID 5	RAID 5>RAID 1	RAID 5	是一种存储性能、数据安全和存储成本兼顾的存储解决方案。
RAID10	读：RAID10 = RAID0 写：RAID10 = RAID1	RAID10 = RAID1	RAID10 = RAID1（50%）	RAID10 = RAID1	集合了RAID0，RAID1的优点，但是空间上由于使用镜像，而不是类似RAID5的"奇偶校验信息"，磁盘利用率一样是50%