一、绪论

信息是一切生物进化的导向资源，信息是知识的来源、是控制与决策的依据、是思维的材料、是管理的基础.

信息：一种以特殊物质形态存在的实体。

香农定义：信息是用来减少随机不定性的东西。

信息安全：数据安全+系统安全

信息安全学关注信息本身的安全（不丢、不坏、不冒、不泄）。信息安全的任务是保护信息财产。

信息安全属性：**完整性，保密性，不可否认性（不可抵赖性）**（安全三要素CIA，在消息层次看），从网络和系统层次看：可用性，可控性

采取合适的安全措施，使安全事件业务造成的影响降低最小，保障组织内业务运行的连续性

信息安全研究：基础理论，应用技术，安全管理

PDRR：保护（Protect），检测（Detect），反应（React），恢复（Restore）。

保护（Protect）指采用可能采取的手段来保障信息的保密性、完整性、可用性、可控性和不可否认性

检测（Detect）指提供工具检查系统可能存在的黑客攻击、白领犯罪和病毒泛滥等脆弱性

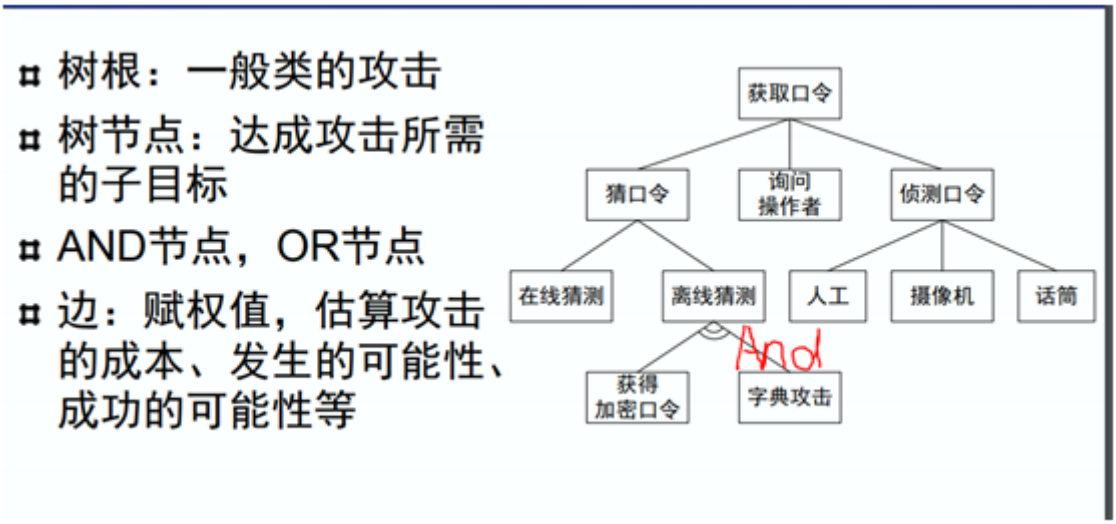
反应（React）指对危及安全的事件、行为、过程及时做出响应处理，杜绝危害的进一步蔓延扩大，力求系统还能 提供正常服务

恢复（Restore）指一旦系统遭到破坏，尽快恢复系统功 能，尽早提供正常的服务

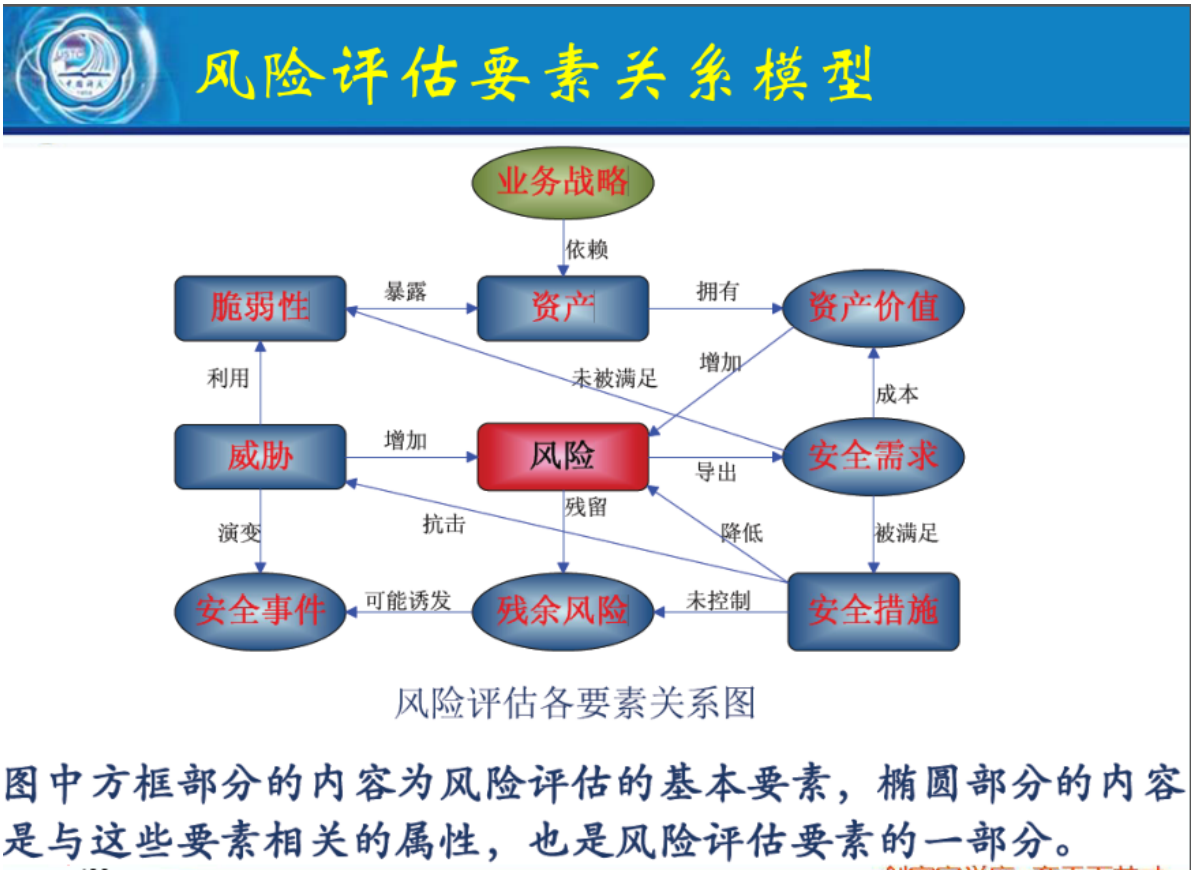
风险=资产×威胁×漏洞

攻击树：定量分析攻击

威胁可根据其可能性来评定



风险管理由两个主要的和一个基础的活动构成；**风险评估**和**风险消减**是主要活动，而**不确定性分析**是基础活动



风险评估各要素关系图

图中方框部分的内容为风险评估的基本要素，椭圆部分的内容是与这些要素相关的属性，也是风险评估要素的一部分。

二、计算机安全基础

平衡可用性和安全性

PDR：预防，检测，反应。**特别重视预防**

PDRR

PPDRR：策略，防护，检测，响应，恢复

保密性：防泄露；完整性：防篡改；可用性：防截留

其他有：可控性，不可否认性，可审计性/问责性，可靠性

保密性：阻止未授权用户阅读敏感信息。个人秘密叫隐私，组织秘密叫秘密。

只有经授权的用户/系统可以访问受保护的数据（系统资源，系统服务）

保密性不涉及可信，仅反映客体敏感级别

有以下方面：授权者、授权客体、授权的操作粒度、授予的权限和访问方式

完整性：即使授权用户也不得以破坏或者丢失的方式**修改**数据，防止未经授权的信息修改

数据完整性：当计算机处理的数据与源文件中的数据一样，且没有受到意外的或恶意的修改或破坏时所呈现的状态

完整性是**外部一致性**的同义词：存储在计算机系统中的数据应当正确地反映计算机系统外的某些事实

通信安全中的完整性：对因蓄意操作或随机的传输错误而引起的传输数据的修改、插入、删除或重放而进行的检测和纠正

系统完整性（硬件资源，软件资源）

完整性一般是其他安全属性的先决条件，分三个方面：1.经授权的资源操作、分离、保护 2.出错检测 3.校正

计算机安全主要关心防止对数据的恶意修改

可用性：防止未经授权的信息或资源截留。需要时可被一个授权实体访问和使用的属性

拒接服务攻击（DoS）：阻止对资源的授权访问或延误时间关键性的操作，比如泛洪攻击

可靠且及时访问数据或者资源

外部攻击：DoS，DDoS，DRDoS

以前的大多计算机安全的成功集中在保密性和完整性上

3A：认证，授权，审计（4A加入记账）

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。重点：正确授权和访问控制

正确授权假设存在一个安全策略，即一组声明什么行为是允许的和什么行为是禁止的规则

安全策略域即是由策略所控制的用户、数据客体、机器等实体组成的集合

国际标准化组织(ISO)将"计算机安全"定义为："为数据处理系统建立和采取的**技术和管理的安全保护**，保护计算机**硬件、软件数据**不因偶然和恶意的原因而遭到破坏、更改和泄露。"此概念偏重于**静态信息保护**

也有人将"计算机安全"定义为："计算机的**硬件、软件和数据**受到保护，不因偶然和恶意的原因而遭到**破坏、更改和泄露，系统连续正常运行。**"该定义着重于**动态意义描述**

橘皮书：TCSEC

数据代表了信息，信息是数据的(主观)解释。数据依照约定所选择的用来表现人们概念上和真实世界中某些方面的物理现象。**赋予数据的含意称为信息**

信息=数据+处理

数据流：物理；信息流，逻辑

隐蔽信道：

隐蔽通道是一个不受安全机制控制的信息流，信息通过一个隐蔽通道传输是可能的。**一个状态变量**：**一次传递一个比特**信息

存储通道：如果一个进程直接或间接写一个存储单元，另一个进程直接或间接读该存储单元

定时通道：如果一个进程通过调解它对系统资源的使用，适应到另一个进程的真实响应时间，实现一个进程向另一个进程传递信息

旁道攻击（旁路攻击，侧信道攻击）：

不攻击密码本身，而攻击实现于不安全系统上的加密系统，有声音，时间，电磁辐射，耗能等。

计算机安全的五个设计原则：

1.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数据、操作还是用户上？

2.一个安全机制应该被放在计算机系统的哪一个层次上？（从硬件到软件）

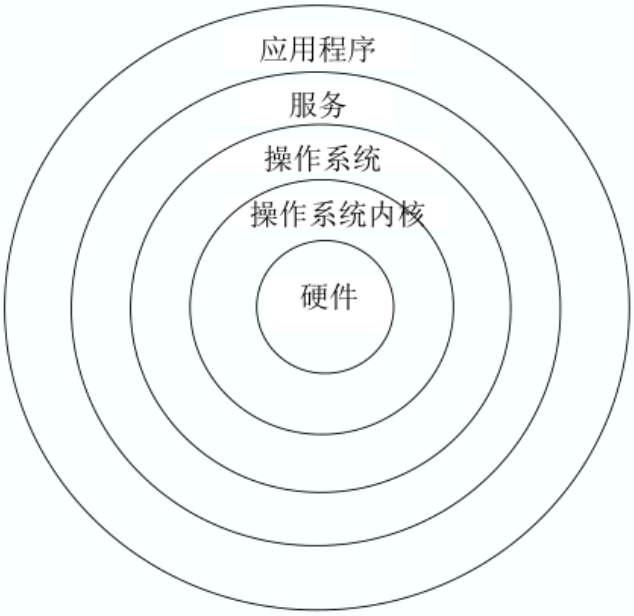
3.特色的安全环境和简单性与更高的保证

4.定义和实施安全的任务是给一个中央实体，还是系统中的成员

5.如何防止攻击者访问位于保护机制下面的层

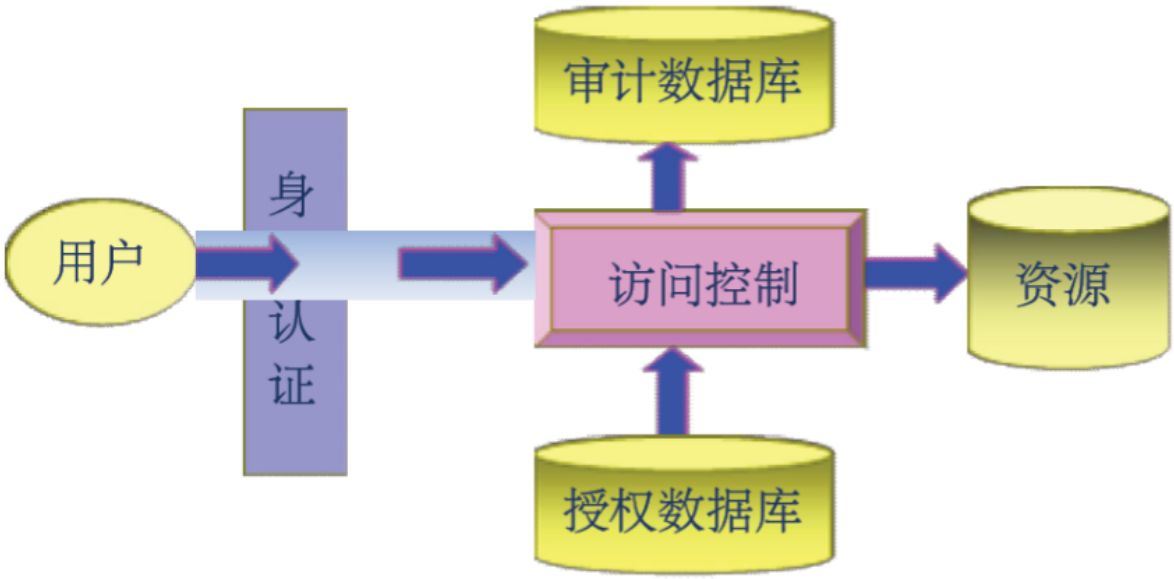
保护机制切面模型

- 新的设计策略　将一个计算机系统的安全机制想象成一些同心的保护环，其中硬件机制位于中心，应用机制位于外围
- 靠近中心的机制趋向于更通用、更面向计算机和更关心对数据的控制访问，外围的机制则更适合解决个别的用户需求



保护机制切面模型

三、身份认证与识别



身份认证基于以下信息：

1. What you know（口令，PIN，令牌等）；
2. What you have（物理令牌如钥匙等，USB Key）；
3. Who are you（生物技术）；

4. How to do（签名，打字，手势等）；
5. Where are you（地理位置等）

身份认证方法：1.基于用户名和口令的认证；2.基于USB Key的身份认证（基于对称或者非对称密码的认证）；3.基于生物特征的身份认证（指纹，视网膜，语言，人脸）

三种口令认证机制面临的威胁：

1.用户长期使用同样口令，没有妥善保管，使用弱口令等不安全的使用措施，使得攻击者通过穷举或者猜测攻击破解口令；

2.欺骗攻击与重放攻击。攻击者伪造网页，或者监听获得口令或通信内容，重放通信内容冒充用户；

3.未加密的口令。口令文件泄露等情况

四、访问控制

客体（O）：可供访问的各种软硬件资源

主体（S）：访问的发起者，通常为进程，程序，用户。

授权：资源所有者对他人使用资源的许可，引用监控器决定访问是否被准许或拒绝

角色：指一个组织或任务中的工作或位置，代表了一种资格，权利和责任。

主角（S）：能被授予访问对象的权限或做出影响访问控制决策的声明

访问操作（A）：由基本的内存操作(读/写)扩展而来，指代面向对象系统的方法调用，不同的系统会使用不同的访问操作

主体是IT系统活动的实体。如**主角是用户ID，主体是用户进程。**

在安全策略用主角，安全策略的执行系统是主体。

引用监控器是检查主体绑定的主角是否有权访问，主体引起信息在客体间流动

访问控制的核心是**授权控制**，满足：一致性（资源控制没有二义性，定义不冲突）；统一性（对所有资源集中管理，有审计功能，尽可能提供细粒度的控制）

访问控制分类：系统访问控制（OS）、网络访问控制、物理访问控制

主体客体是相对的，比如进程既可以是主体，也可以是客体

两种访问模式：**查看、修改**

访问权限：主体对客体进行访问的具体形式。例如，读、写和执行等

BLP模型的四种访问权限：读，写，执行，附加（盲写）

访问权限和访问模式的关系：

	read	write	append	execute
observe	X	X		
alter		X	X	

Multics系统用e r a w 表示Bell-LaPadula访问权限，访问属性和访问权限的关系是：

数据段		目录段	
读	r	状态	r
执行	e, r	状态和修改	w
读和写	w	附加	a
写	a	搜索	e

Unix/Linux系统：读写执行。控制对文件目录的写访问。Unix中为文件规定的访问权限是通过修改文件在目录中的表项来改变

Windows是指定特殊的创建和删除文件权限。

所有权是物权中最重要也最完全的一种权利，具有绝对性、排他性、永续性，具体包括占有、使用、收益、处置

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。也称为基于身份的访问控制（IBAC）

强制访问控制（MAC）：系统策略规定谁可以访问，也称为基于规则的访问控制

DAC和MAC都是直接赋予使用者（角色）权限

访问控制矩阵。当主客体数量较多，可以用中间控制层

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

- bill.doc可以被Bob读和写，但完全不允许Alice访问
- edit.exe可以被Bob和Alice执行，但除此以外他们不能访问
- fun.com可以被两个用户读和执行；只有Bob能够写这个文件

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

访问控制矩阵是一个抽象概念，不适用于直接实现，也不方便进行安全管理。是稀疏矩阵。

能力（访问控制矩阵的行）：访问权限和主体保存在一起。行：访问能力表

每个主体被赋予一个能力，列出了一个主体所有的访问权限或一个用户拥有的能力。

访问能力表（CL）

能力与自主访问控制联系

访问控制列表（访问控制矩阵的列）：访问权限和客体保存在一起。列：访问控制表

访问控制列表（ACL）声明谁可以访问一个给定的客体，是橙皮书C2类安全操作系统的一个典型特征

ACL VS. CL：

1. 二者鉴别实体不同
2. 保存位置不同
3. 浏览访问权限：ACL容易，CL难
4. 访问权限传递：ACL难，CL容易
5. 访问权限回收：ACL容易，CL难
6. ACL和CL的转换：ACL->CL难，CL->ACL容易

组：将用户置于组中，也可以从组中获得访问权限。可以用来简化访问控制策略定义，具有类似访问权限的用户可以集中在一个组中，组被授予访问客体的许可

否定的许可：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

主体——特权（类似组）——操作

基于角色的访问控制（RBAC）：一组特定应用的操作叫角色，主体从履行的角色上获得访问权限

保护环主要被用于完整性保护，数字越小越重要

Windows NT：每个客体有一个安全描述符（Security Descriptor），如果它未包含访问控制列表，则所有用户都能访问该客体；如果包含是空表，则拒绝所有访问

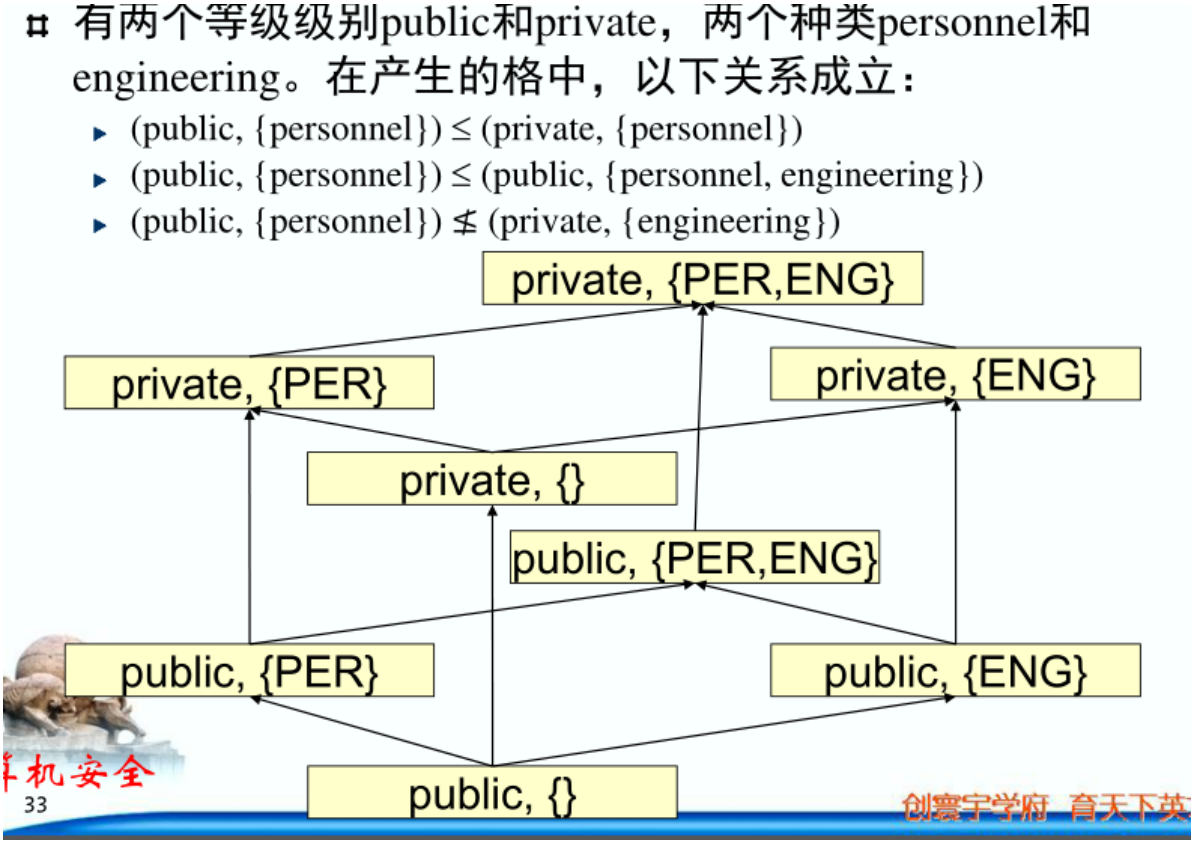
系统低，系统高：最大下界和最小上界

多级安全MLS，强制安全策略：**无向上读，无向下写**

绝密——机密——秘密——非保密，公开

需要知道（need-to-know）策略

安全级别的格：



基本模型RBAC0：包括用户，juese客体，操作，许可权

登记模型RBAC1：引入了角色的继承关系，孩子继承父亲的许可

约束模型RBAC2：引入责任分离关系

统一模型RBAC3：1+2

五、使用控制

使用控制模型：UCON，核心ABC模型

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

两个扩展元素：主体属性、客体属性

A：授权规则；B：义务；C：条件

pre（A、B、C）：使用前；on（A、B、C）：使用中

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）

16种基本模型：**决策因素pre(A/B)：属性改变只能在使用前后，条件不能改变主客体属性

决策因素	0(不可改变)	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

基本模型：比如$UCON_{preA1}$，表示以授权规则为决策因素，并在使用前改变主客体有关可变属性的基本模型

组合模型：比如$UCON_{preC0onC0}$：表示使用前和使用中都执行条件决策因素，都不改变属性

DAC->$UCON_{preA0}$

(1)S表示主体即用户或用户组，O表示客体，N表示身份标记集合，R为操作权限集

(2) id表示用户到标记集合N之间的一对一映射关系

(3) ACL表示客体与N×R之间的映射关系，其中N×R为用户身份与操作权限的组合关系

(4) ATT(S) = {id}

(5) ATT(O) = {ACL}

(6) 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

MAC->$UCON_{preA0}$

(1) L表示具有偏序关系的安全级别集合；

(2) clearance: S → L，是访问主体S和安全级别L之间的映射函数

(3) classification: O → L，表示客体和安全级别L之间的映射函数

(4) ATT(S)={clearance}

(5) ATT(O)={classification}

(6) allowed(s，o，read) => clearance (s) ≥ classification (o)，即主体可以“下读”客体

(7) allowed(s，o，write) => clearance (s) ≤ classification (o)，即主体可以“上写”客体

RBAC1->UCON$_{preA0}$

(1) P = {(o，r)}，P表示许可集合，(o，r)为客体-权限对

(2) ROLE表示角色层次的偏序关系

(3) actRole表示激活角色，实现“用户-角色”分配

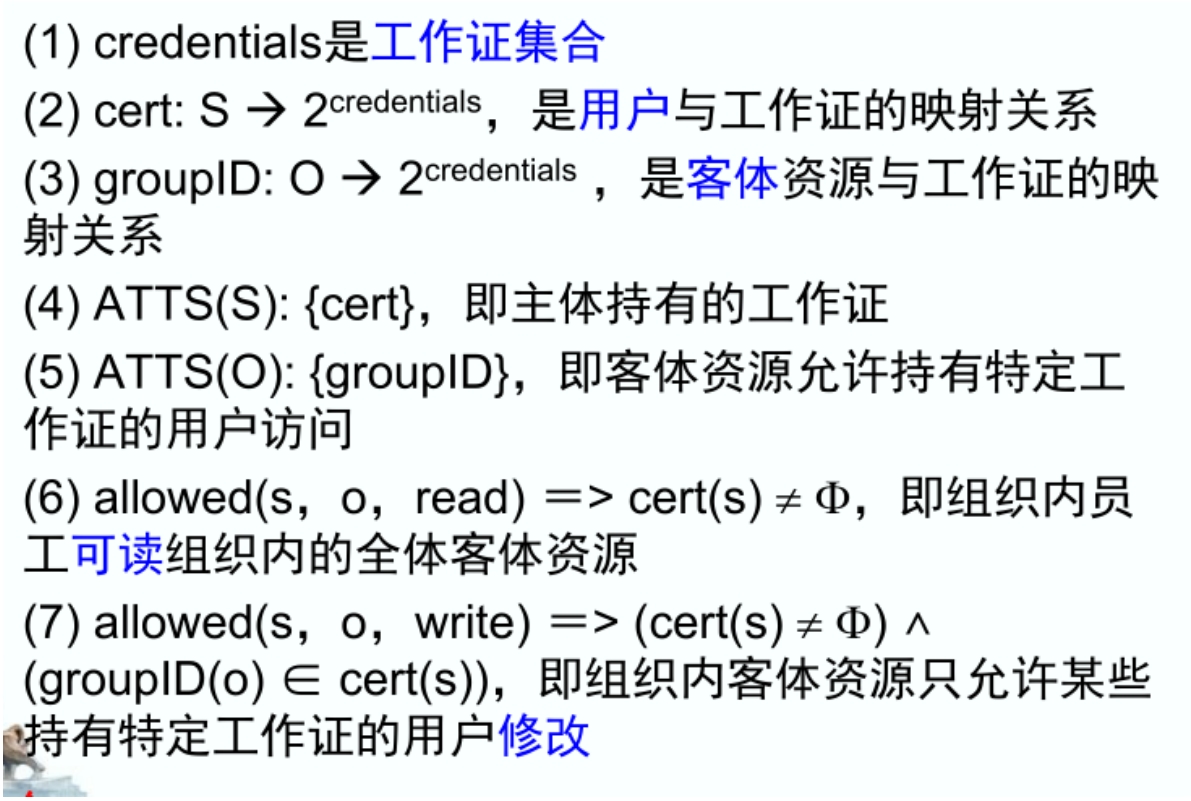
(4) pRole表示授权角色，实现“角色-许可”分配

(5) ATT(S) = {actRole}；

(6) ATT(O) = {pRole}

(7) allowed(s，o，r) = role ∈ actRole(s) ∧ role' ∈ pRole(o，r) ∧ role ≥ role'，即如果存在授权角色(pRole(o，r))，其偏序关系≤激活角色(actRole(s))，则访问请求被允许

某个实际例子：



六、访问监控器

引用监控器（RM）：一个访问控制概念，指所有主体对客体访问进行冲裁的抽象机器

安全内核（SK）：实现RM的可信计算基的硬件固件软件总和，必须处理所有访问控制。

可信计算基（TCB）：计算机系统中全部保护机制的总和，包括硬件固件软件

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、硬件和负责系统安全管理的人员一起组成了系统的可信计算基（Trusted Computing Base，TCB）

橘皮书定义：SK是RM的实现，TCB包括了SK

访问控制机制的理论基础是访问监控器

引用验证机制（RVM）的三个原则（核心要求）：

1. 必须有自我保护能力，以保证引用验证机制即使受到攻击也能保持自身完整性，从而保证系统安全
2. 必须总处于活跃状态，是所有访问请求的唯一入口（不可旁路），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
3. 必须设计的足够小，利于分析测试，保证引用验证机制的实现正确性和符合要求

安全内核包括RVM的实现、对系统自身的访问控制，以及组成管理用户和程序的安全性组件

受控调用：系统在root模式只执行一组预先定义的操作，而在控制交回给用户前返回用户模式

可信路径：交互安全敏感数据时，需要建立一条从输入或输出设备到TCB的可信路径

门：一个指向某个·程序（在某个代码段中）的系统客体，门具有与它指向的代码不同的特权级别

门允许对内环的程序进行**只执行**，但仍限制对外的调用

如果一个过程要使用门，门必须和过程位于同一个环内

调用程序不能写入内环

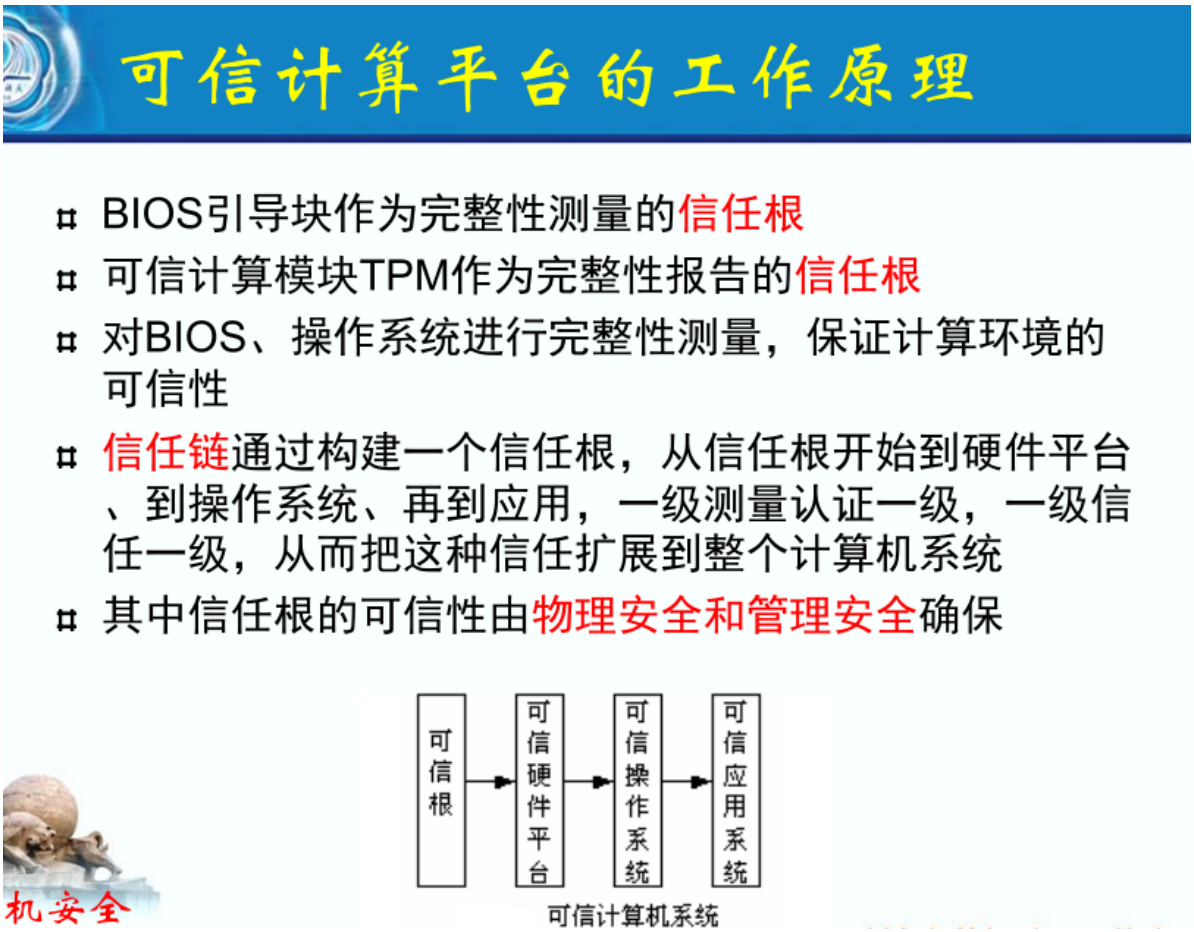
七、计算机实体安全

可信计算：计算机终端是安全源头

用途：

1. 风险管理
2. 数字版权管理
3. 电子商务
4. 安全监测和应急

可信计算"技术的核心是称为TPM(可信平台模块)的安全芯片，具有安全存储和加密功能



机房的安全考虑：场地选择、结构**防火**、机房内部装修、活动地板、供配电系统、空调系统、其它设备和辅助材料、火灾报警及消防设施、**防水**安全计算机房、**防静电**计算机房、**防雷击**、**防鼠害**、**电磁波**的防护

计算机工作环境：温度10-35度，湿度40%-60%

八、Unix/Linux安全

Unix实现了自我访问控制，粒度是owner(属主)，group(组)，other

主角：

用户标识符（UID）和组标识符（GID），均为16位数字，比如0：root，2：bin，3：sys。**root的UID是0**

有关主角的信息被存在用户账号以及主目录中

超级用户的特权：所有安全检查均被关闭，可以切换到其他任何用户，能改变系统时钟。能够绕过施加的少数限制

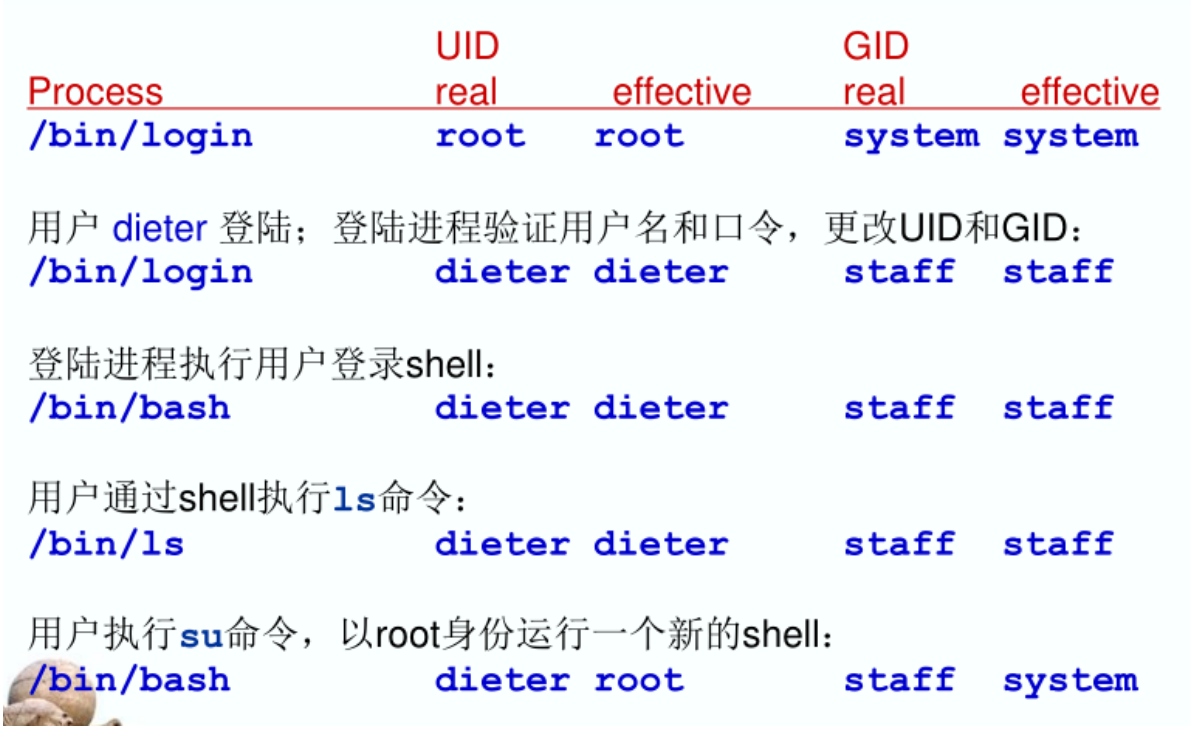
每个用户属于一个基本组，GID存在/etc/passwd

一般系统会预留一部分GID给系统群组，和一般使用者群组分开

主体：

进程。每个进程都有一个PID。

每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**

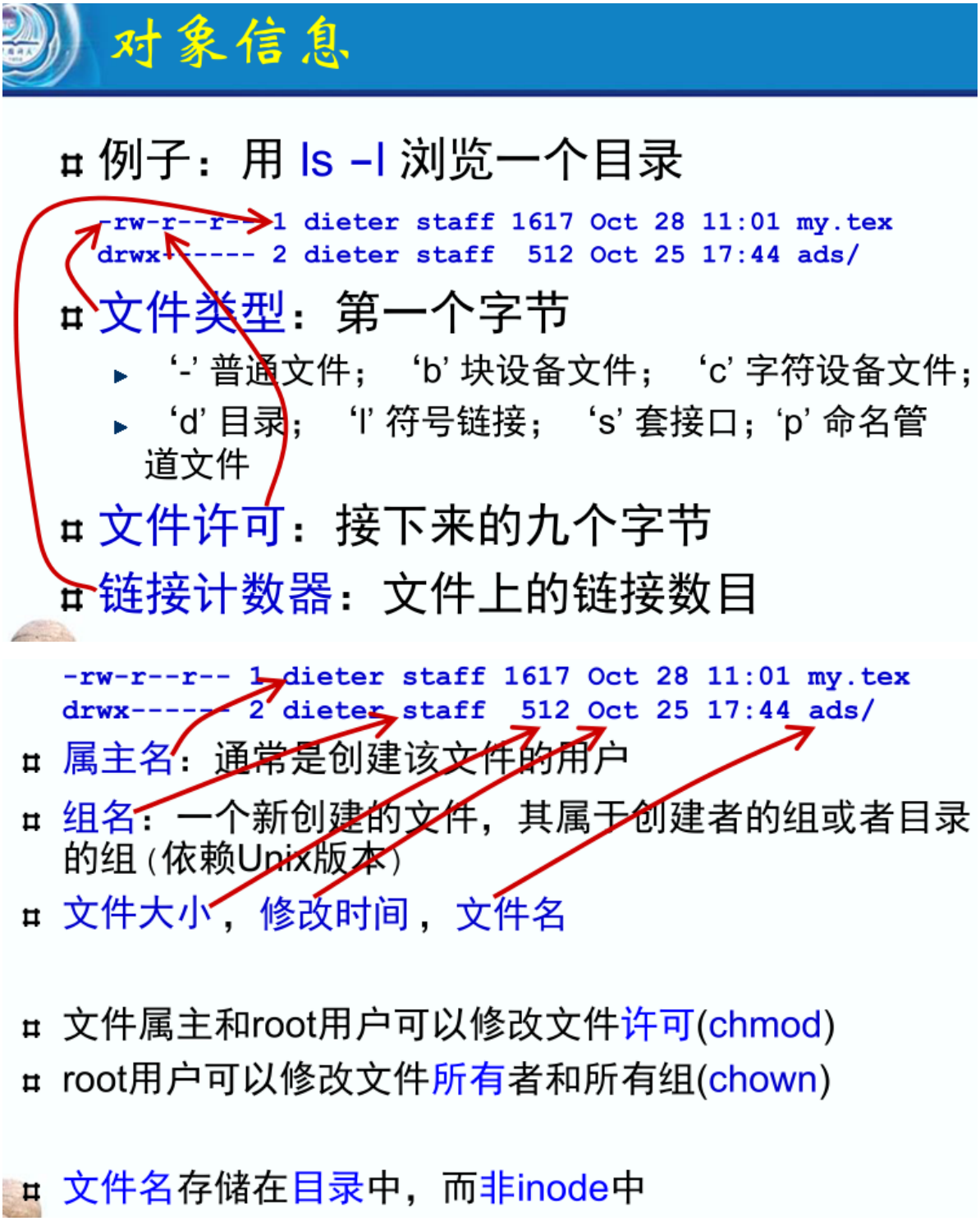


口令：口令域为空，不必提交口令，口令域为*，用户不能登陆，口令存在影子口令文件中，只有root能访问

客体：

文件、目录、存储设备以及I/O设备被统一视为资源(resource)

目录中的每一个文件表项都是一个指针，指向名为i节点(inode)的数据结构，inode中的mode属性是文件类型和访问权限



文件许可位分为属主，属组，其他的读写执行权利，-表示没权，比如rw-r--r--，是属主可以读写，属组和其他只能读。用3bit表示，就是110100100=644

三个附加位：设置UID为文件owner的SUID，设置GID为文件group的SGID，和粘滞位。在最前面的一位表示附加位。4000，设置UID；2000，设置GID；1000，设置粘滞位（UGS的顺序）

当ls –l命令显示一个SUID/SGID程序时，属主/属组的执行许可位是s而不是x

目录许可也有读写执行

粘滞位目录的文件，只有用户是文件属主，目录属主且可以写或者为root才能删除或者重命名。用ls -l查看有粘滞位的目录时，用t不是x表示执行许可

用chmod修改访问权限，仅限文件属组或者root，比如chmod 0754 filename或chmod u+wrx,g+rx,g-w,o+r,o-wx filename

用有效UID/GID进行访问控制，顺序是先判断属主，再组，再其他。文件属主总是能更改访问许可位

新建文件默认权限666，新建程序默认777，可以用Umask调整。例：默认666，umask 077，先按位取反为700，再与666按位与为600。umask 777拒绝所有访问，000不做任何限制

受控调用：即sudo

Unix安全：用户名+口令，审计日志等

九、Android安全

安卓的系统架构：分层架构

应用程序层：Java程序

应用程序框架层：Dalvik虚拟机

系统运行库层：C/C++程序

Linux核心层：内核、驱动

应用开发语言：Java，C/C++

Dalvik是安卓系统的一个JVM，基于寄存器（对提前优化提供更好支持），执行特有的DEX文件格式（dex字节码将多个文件整合成一个，提高查找速度，减少整体文件尺寸等），核心内容是实现库（libdvm.so），高性能，编译时间更短

Dalvik是**一个应用，一个虚拟机实例，一个进程**，即每个应用都运行在一个虚拟机中，每个虚拟机都是一个独立的进程空间。虚拟机关闭或中止不会影响其他虚拟机，保护应用的安全和独立运行。

ART（Android Runtime）：安卓4.4版本ART取代Dalvik。使用Ahead-Of-Time（AOT）编译（在应用安装时就预编译字节码到机器语言），改变了程序运行根本机制，**程序启动更快更省资源**，但增加了程序安装的时间，安装后占更多空间

在Dalvik下，应用每次运行的时候，字节码都需要通过**即时编译器**转换为机器码，这会拖慢应用的运行效率，在ART 环境中，应用在**第一次安装**的时候，字节码就会**预先编译**成机器码，使其成为真正的本地应用

安卓系统的安全机制：

1. 进程沙箱隔离机制
2. 用户ID机制
3. 权限机制
4. 签名机制
5. SEAndroid机制

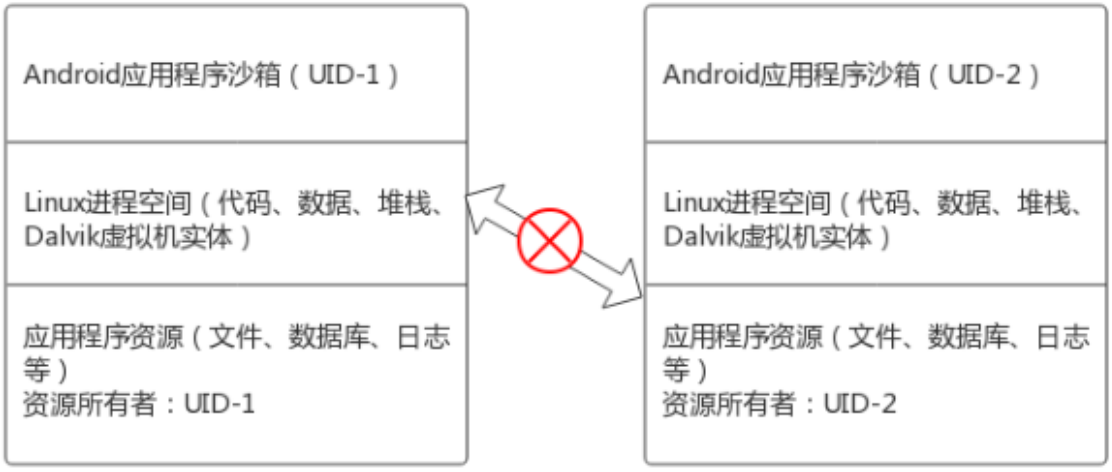
安卓最重要安全设计：**沙箱**和**权限**

最主要安全机制：用户ID、权限、签名

沙箱隔离机制，使得安卓应用在安装时被赋予独特的UID，并永久保持。应用和运行的Dalvik虚拟机运行在独立的Linux进程空间，与其他应用完全隔离。除非显示声明premissions，来获得额外能力，是静态在程序中声明，所以会在程序安装时被知晓并不会再改变。

如果其他程序拥有该应用程序的共享的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其数据

Android应用程序沙箱机制
两个应用程序的UID分别为1、2



安卓的安全限制措施是在进程级实施的。

权限：应用能做的事，在安装时确定，而不在运行的时候。权限包括：名称，属于的权限组，保护级别。

权限组：把权限按功能分成不同集合

保护级别：权限通过protectionLevel来标识保护级别，分为：普通、危险（在安装应用时会提醒用户）、签名（具有同一签名的应用才能访问）、系统/签名（不推荐使用）

普通级别和危险级别属于比较低的级别，申请即可授予，签名级别和系统/签名级别需要使用者的应用程序和系统使用同一个数字证书才可以申请成功

使用权限，自定义权限，组件权限，发送广播时支持权限，其他支持权限，URL权限等

十、Windows安全

Windows两种模式：用户模式和内核模式

内核中安全组件：安全引用监控器

用户模式安全组件：**1.登陆进程（WinLogon）**；**2.本地安全权威（LSA）**，用户登录时，检查有用户账户并创建访问令牌，负责审计；**3.安全账户管理员（SAM）**，维护用户账户数据库，在本地用户认证时，LSA使用数据库，口令的hash存在SAM中

注册表：Windows**配置数据**的中央数据库，表项称为**键**，修改查看：**注册表编辑器**，**注册表蜂房**：一组键、子键以及键值

域：实现**一次签到**和**集中式的安全管理**，是共享公用用户账户数据库和安全策略的计算机集合。要有一台服务器当**域控制器（DC）**，其他计算机加入域，域管理员在DC上创建并管理域用户和组

活动目录：对象被组织在活动目录中，可以看成一个有特定类型对象构成的树，每一种对象类型有一个特性的属性和唯一的GUID，每一种属性都有GUID。

容器：可以包含其他对象的对象，活动目录可以通过添加新的对象类型以及为已存在的对象类型添加新属性动态扩展，活动目录服务是Windows平台的核心组件

主角：本地用户，别名（本地组），域用户，组，及其，有uesr name（用户名）和SID（安全标识符），LSA创建的用户称为本地用户

域控制器（DC）为一个域提供安全服务，域控制器权威（DCA）认证时充当可信第三方，设计原则是集中式认证（口令管理）（域内），域间：分布式服务

主体：OS的活动实体，Windows的主体但是进程和线程，安全凭证存放在**访问令牌**中，令牌包括一系列主角和其他安全属性

认证：绑定一个主体到一个主角

对象是访问操作中的被动实体，**安全对象具有安全描述符**：

1. 属主SID：属主是一个主角，对象在创建时就有属主，通常有读写权利
2. 主组：为了和POSIX兼容
3. 自主访问控制列表（DACL），定义了谁将被给予或者拒绝访问
4. 系统访问控制列表（SACL），定义了审计策略

许可（访问权限）：描述可以施加于对象的操作，许可被编码为32bit掩码（访问掩码），主体所要求的访问操作以访问掩码的形式给出

改变文件的DACL不会影响已经打开的文件句柄

DACL：访问控制表项（ACE）的列表，ACE包括主角SID

访问检查：从属主得到许可；遍历DACL；当主体的令牌包含一个配对的SID时检查ACE，允许访问，当所请求的所有许可都已经获得，拒绝访问，找到否定的ACE或者到达ACE末尾

否定ACE在优先于肯定的ACE，必须被放置在DACL的顶端

Empty DACL：访问一直被拒绝，NULL DACL：一直被允许

ObjectType：定义对象类型的GUID，对于一个给定的请求，只有具有一个匹配的ObjectType或不具有ObjectType的ACE才会被评估，不具有ObjectType的ACE将应用于所有对象

受限上下文：以受限令牌运行的进程，设计原则：最小特权

安全管理：审计，升级

SACL：定义审计规则，非自主，由管理员设置。ACE种类：肯定的（审计允许），否定的（审计拒绝）

一个ACE可以同时是肯定的和否定的

审计事件由特殊审计函数产生

DEP：数据执行保护，软硬件技术，在内存上执行额外检查，以防止在系统上运行恶意代码，原理是将所有内存页标为不可执行，CPU会抛出异常而不是执行代码

ASLR：地址空间局部随机化（针对缓冲区溢出）

SAM：安全账户管理器，保存账户名和口令

练习2 Windows系统中，DACL和SACL有什么区别？

- DACL（Discretionary Access Control List），其指出了允许和拒绝某用户或用户组的存取控制列表。 当一个进程需要访问安全对象，系统就会检查DACL来决定进程的访问权。如果一个对象没有DACL，那么就是说这个对象是任何人都可以拥有完全的访问权限。其中ACE的类型是肯定(允许)或否定（拒绝） 。 DACL中的ACE格式包括： 类型：肯定(允许)或否定(拒绝) 标志；Object Type； Inherited Object Type；访问权限；主角SID： ACE应用的主角。
- SACL（System Access Control List），其指出了在该对象上的一组存取方式（如，读、写、运行等）的存取控制权限细节的列表。审计规则在SACL中定义， SACL中的ACE格式包括： Type：肯定的（审计允许的许可）或否定的(审计拒绝的许可)； Trustee：一个 SID（个人，组，别名）； Mask：许可(32-bit 掩码)。一个ACE可以同时是肯定的和否定的 。

十一、BLP模型

主体：主要是进程，能执行一系列动作，客体：被动元素

主体集合（S），客体集合（O），访问操作集合（A）：读写执行，偏序的安全级别集合（L）

模型的状态集必须包括当前所有的许可和当前所有主体访问客体的实例，状态集=B×M×F

状态集 B，M

- B = P(S×O×A)是当前访问的集合
  - 元素b∈B是一个三元组（s, o, a），表示主体s当前正在客体o上执行操作a，例如(Alice, fun.com, read)
  - B = P(S×O×A)，例如{(Alice, fun.com, read), (Bob, fun.com, write), …}
- M表示所有获得访问许可的矩阵的集合，每个访问许可矩阵$m = (m_{so})$ s∈S, o∈O
  - M = $\{\{read, write\}_{Alice, fun.com}, \{read, write, execute\}_{Bob, fun.com}\}$

状态集 F

- F⊆ $L^S \times L^S \times L^O$是安全级别分配的集合。元素$f \in F$是一个三元组$(f_S, f_C, f_o)$，其中
  - f_S: $S \rightarrow L$给出了每个主体可以拥有的最高安全级别
  - f_C: $S \rightarrow L$给出了每个主体当前的安全级别
  - f_o: $O \rightarrow L$给出了所有客体的安全级别
- 一个主体的当前级别不能高于他的最高级别，因此$f_C \leq f_S$，读作“f_S控制f_C”或“f_S支配f_C”
- 高安全级别有时称为主体的许可(clearance)，其它文献只用许可来表示用户的安全等级

进当主体安全级别大于客体才可以读（强制性安全策略），访问必须被访问控制矩阵允许，**下读上写**：主体对可以向下读，向上写

ss-property：简单安全性，s大于o，a是读或写。**无向上读安全策略**。主体能够作为通道，读一个内存对象并且将信息传送到另一个内存对象中(会有安全问题)

主体作为一个通道

- ss-property不足以防止一个低级别的主体读一个高级别客体的内容
  - 可以创建一个高级别的特洛伊木马，读高级别的客体并将它拷贝（写它的内容）到一个低级别的客体
  - 高级别的主体必须是可信的
  - 必须通过*-property来控制写访问

*-property（星特性）：无向下写安全策略，s小于o，a是添加或写

  - 此外，如果存在一个元素，有$(s,o,a)\in b$，访问操作a是添加或写，那么对于所有的客体o'，$(s,o',a')\in b$，a'是读或写，必须有$f_O(o') \le f_O(o)$

- BLP模型的第一个版本没有包括*-property
- ss-property和*-property称为强制BLP策略

避开*-property限制

- *-property表明高级别的主体不能发消息给低级别的主体。有两种方法可以避开这个限制：
  - 临时地降低一个高级别主体的等级，这是引入当前安全等级f_C的原因
  - 确定一组允许违背*-property的主体，这些主体称为可信主体(trusted subjects)
- f_S规定了用户的许可，用户允许以低于许可的级别注册，f_C表明了用户实际注册的级别
- *-property只对不可信的主体有效。如果确信一个主体不会造成伤害，则称它是值得信赖的(trustworthy)

一个可信主体可以违背安全策略，要区分**可信主体**和**值得信赖的主体**

ds-property（自主安全性）：对于每个元素，a都在m_{so}中。

如果(b,m,f)满足三个property，则称状态是安全的，如果两个状态都是安全的，则称从状态1到状态2的迁移是安全的。

- 状态迁移保持ss-property，当且仅当：
  - 每个$(s,o,a)\in b_2\backslash b_1$满足关于$f_2$的ss-property；（$b_2\backslash b_1$表示$b_2$和$b_1$的差集）及
  - 如果$(s,o,a)\in b_1$不满足关于f_2的ss-property，那么$(s,o,a)\notin b_2$。
- *-property和ds-property特性的保持可以用类似的方法来描述

基本安全定理：系统中所有状态迁移都安全，初始状态也安全，则不管输入如何，之后每个状态都是安全的。

证明建立在这样的事实上，即每个状态转换保持安全，且不涉及特定的BLP安全特性

基本安全定理是状态机模型的产物，而不是BLP模型中选用的特定安全特性的结果

练习8.1 用基本访问模式术语alter（修改）和observe（查看）描述*-property

- 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$f_C(s)\le f_O(o)$，那么状态(b, m, f)满足*-property
- 此外，如果存在一个元素，有$(s,o,a)\in b$，访问操作a是添加或写，那么对于所有的客体o'，$(s,o',a')\in b$，a'是读或写，必须有$f_O(o')\le f_O(o)$

- 如果主体可以对客体进行alter，那么主体s的当前级别受客体o的当前级别控制，也就是$f_C(s)\le f_O(o)$。
- 如果主体可以alter客体o，可以observe客体o'，那么$f_O(o')\le f_O(o)$。

---

1 PART

练习8.5 改写针对Multics操作系统的ss-property

- 如果对于每一个元素都有$(s,o,a)\in b$，访问操作a是读或写，主体s的安全级别控制客体o的安全级别，即$f_s(s)\ge f_o(o)$，那么状态(b, m, f)满足ss-property
- Multics的主体就是进程。每个主体都有一个描述符段（descriptor segment）对每一个客体，在每个主体的描述符段中都有一个段描述符字（segment descriptor word， SDW）。
- 对于一个活动进程的描述符段中的任何段描述符SDW，如果读或者写指示器处于开启状态，则进程的当前级别支配段级别。

十二、Biba模型和中国墙模型

Biba：第一个解决**完整性**问题的安全模型

Biba：禁止干净的高级别实体别脏的低级别实体污染，信息只能向下流动

静态完整性级别：

简单完整性（无向上写）：s可以修改o，则s大于o

完整性*特性：s可以读o，则当o大于o‘，s可以读o’

这两个是强制性BLP策略的对偶，也是主张完整性是保密性的对偶的基础

Biba：无向上写：完整性

BLP：无上读无下写：保密性

动态完整性级别：如果一个实体已经接触了低级别的信息，**低水印性策略**自动调整实体的完整性级别

主体低水印性：s读任何客体o，主体新的级别是s和o的最低值

客体低水印性：s修改任何客体o，客体的新级别s和o的最低值

就是印度种姓

调用策略：主体可以调用另一个主体来访问客体

调用性：一个低级别主体s1不能通过调用主体s2来间接访问一个高级别客体，主体可以调用主体仅当s1大于s2

保护环：低级别主体s1调用高级别工具s2来间接访问一个高级别客体。s1可以访问所有客体，s1大于o的时候才能修改，s1小于s2的时候可以调用主体s2

保护环和调用性不一致，操作系统通过保护环来进行完整性保护

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

冲突的产生是因为客户是同一个市场的直接竞争者或者是因为公司的所有权

分析员必须遵循以下安全策略：一定不能有引起利益冲突的信息流

组件

公司的集合用C表示

主体：分析员，主体的集合用S表示

客体：信息条目，每一客体均指向一个独立的公司，客体的集合用O表示

公司数据集(company dataset)：有关同一公司的所有客体的集合。函数$y:O \rightarrow C$给出了每个客体的公司数据集

利益冲突类(conflict of interest)：相互竞争的公司；函数$x: O \rightarrow P(C)$给出了客体o的利益冲突类，即不应该知道该客体内容的公司集合

安全标签：(利益冲突类,公司数据集)，即$(x(o),y(o))$

净化的信息：去除了敏感细节，不受访问限制。一个被净化的客体的安全标签为$(\Phi,y(o))$

简单安全性：当客体o是：1.用户已经拥有的一个公司数据集；2.一个完全不同的利益冲突类。

$N = (N_{so})_{s\in S,o\in O}$，布尔型矩阵，$N_{so}$ = true表示主体s访问过客体o

ss-property：主体s允许访问客体o，仅当对于所有$N_{so'}$ = true的对象o'，有$y(o) = y(o')$ 或 $y(o) \notin x(o')$

$*$ - Property：主体s允许对客体o进行写访问，仅当s对$y(o) \neq y(o')$ 和 $x(o') \neq \Phi$的对象o'没有读访问权

仅当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

每次访问操作，主体的访问权限都要重新赋值

中国墙：访问权限动态变化

十三、信息流模型

信息流控制策略：描述信息流动的合法路径的安全策略

保密性：信息流不流向未授权用户

完整性：信息流流向的进程可信度不高于数据可信度

信息从客体x->客体y，记为y<-x，x是来源，y是目标

强信息流（显式信息流）：x的值直接影响y的值，比如函数

弱信息流（隐式信息流）：不存在显式xy函数，x的值不直接影响y，弱信息流中x的值一般作为条件，比如：if(f1(x1)==0) then y=f2(a) else y=f3(b)，其中ab是客体

信息沿着三种类型的信道流动：

隐蔽通道：一个不受安全机制控制的信息流，信息通过一个隐蔽通道传输是可能的。**一个状态变量**：**一次传递一个比特**信息

存储通道：如果一个进程直接或间接写一个存储单元，另一个进程直接或间接读该存储单元

定时通道：如果一个进程通过调解它对系统资源的使用，适应到另一个进程的真实响应时间，实现一个进程向另一个进程传递信息

计算弱信息流的熵例子：

考虑赋值“IF *x=0* THEN *y:=1*”

设*x*和*y*是二进制变量，*y*的初始值设为0，*x*的两个值很可能相等，有

$$p(0|0)=p(1|1)=0, p(1|0)=p(0|1)=1, 因此 H_y(x)=0$$

事实上，在执行赋值后观察*y*，可知道*x*的确切值，*x*中的所有信息流入了*y*

如果*x*为0, 1, 2的概率相等，将得到

q(0)=2/3, q(1)=1/3

$$p(0|0)=p(1|1)=p(2|1)=0, p(1|0)=p(2|0)=\frac{1}{2}, p(0|1)=1, 因此 H_y(x)=\frac{2}{3}$$

十四、安全评估

橘皮书TCSEC：第一个评估安全产品（OS）的指导方针

将计算机安全从高到低**A、B、C、D四类七个级别**，共27条准则，安全级别定义是递增的

D：**最小保护级**

C：**自主保护级**（need-to-know）（分为C1自主安全保护，C2受控的存取保护），C2基本是主流计算机OS的数据库管理的版本

B：**强制保护级**（基于标签）（B1标记安全保护，B2结构化保护，B3安全域）

A：**验证保护级**（验证设计级A1）

ITSEC：应用于安全产品和安全系统，打破功能性的保证性的关系，方便兼容新的安全需要，针对**完整性**

评估对象（TOE）的安全目标进一步取决于法律和其他规则，这些形成了所需的安全功能和评估级别，安全对象定义了评估相关的TOE的所有方面，TOE的安全功能可以独立定义，也可以引用预定义的功能类，E0-E6七个评估级别表示了安全功能执行的正确性的保证级别

将安全概念分为**功能**和**功能评估**：

功能标准：F1-F10共10级，1-7级对应TCSEC的D到A，6-10还对应：

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：包括保密性的完整性的网络安全

评估准则：七级，在评估准则中首次提出CIA

E0：最低，E1：测试，E2：配置控制和可控的分配，E3：能访问详细设计和源码，E4：详细的脆弱性分析，E5：设计与源码明线对应，E6：设计与源码在形式上一致

CC（通用准则）：使用对象：最终用户、开发者、测评者

对产品和系统(评估对象TOE)进行安全评估的标准

保护框架（PP）：适合特定用户需求的安全需求（包括一个EAL）的（可重用的）集合，用户应该开发自己的PP来捕获自己的典型安全需求

安全目标（ST）：表示一个特定的TOE的安全需求，例如对一个PP的引用，是任何安全评估的基本要素

评估保证级别（EAL）：定义了一个TOE的开发者个安全评估者的责任，有七个递增的级别：**EAL1~EAL7**

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

EAL6：半形式化验证的设计和测试

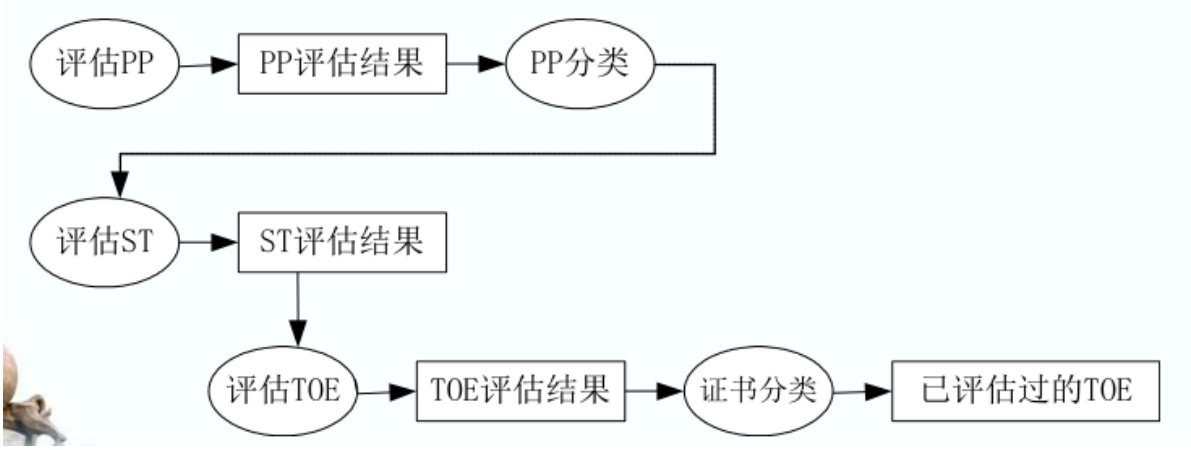
EAL7：形式化验证的设计和测试

评估类型：

PP评估：评估的目标是为了证明PP是完备的、一致的、技术合理的，而且适合于作为一个可评估TOE的安全要求的声明

ST评估：适合于用作相应TOE评估的基础，当某一ST宣称与某一PP一致时，证明ST满足该PP的要求

TOE评估：目标是为了证明TOE满足ST中的安全要求



十五、网络安全等级保护

等级保护的核心是对信息系统特别是对业务应用系统安全分等级、按标准进行建设、管理和监督

等级保护的主要内容：

1. 对国家秘密信息、法人和其他组织及公民的专有信息、公开信息分类分等级进行管理和保护
2. 对信息系统按业务安全应用域和区实行分等级保护
3. 对系统中使用的信息安全产品实行按分等级许可管理
4. 对等级系统的安全服务资质分等级许可管理
5. 对信息系统中发生的信息安全事件分等级响应、处置

- 五级：访问验证保护级
- 四级：结构化保护级
- 三级：安全标记保护级
- 二级：系统审计保护级
- 一级：用户自主保护级

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）

主要工作：1、定级；2、备案、3、建设/整改；4、定期登记测评；5、定期监督检查

测评流程：测评准备活动，方案编制活动、现场测评活动、报告编制活动

业务信息和具体服务：相对应，分别从静态和动态体现了信息系统的重要作用

受侵害客体：国家安全；社会秩序、公共利益；公民、法人和其他组织的合法权益

侵害程度：一般损害、严重损害、特别严重损害

信息系统的安全保护等级由业务信息安全保护等级和系统服务安全等级较高者决定

业务信息安全被破坏时所侵害的客体	对相应客体的侵害程度		
	一般损害	严重损害	特别严重损害
公民、法人和其他组织的合法权益	第一级	第二级	第二级
社会秩序、公共利益	第二级	第三级	第四级
国家安全	第三级	第四级	第五级

重点

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

十六、数据库安全

视图：本身不存储数据

快照：有自己的独立存储的数据

存储过程：一组为了完成特定功能的SQL语句集。有系统存储过程、本地、临时、远程、扩展

授权：grant和revoke

触发器主要是通过事件进行触发而被执行的，存储过程通关存储过程名字可以直接调用，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

攻击：

聚集：数据库中一组值计算出来的聚集的敏感度，与单个元素的敏感度不同

推断问题：从一些不敏感的数据中派生出一些敏感数据

推断问题：从一些不敏感的数据中，派生出一些敏感的数据：

- 直接攻击：聚集是从小组样例中计算出来的，这样私人的数据项就可能被泄露出去
- 间接攻击：这种攻击结合了与几个聚集关联的信息
- 跟踪攻击：这是间接攻击的一种非常有效的类型
- 线性系统的漏洞攻击：这种攻击又比跟踪攻击进了一步，它采用查询集中的算术关系，来构建等式，以产生期望的信息

Name	Sex	Program	Units	Grade Ave.
Alma	F	MBA	8	63
Bill	M	CS	15	58
Carol	F	CS	16	70
Don	M	MIS	22	75
Errol	M	CS	8	66
Flora	F	MIS	16	81
Gala	F	MBA	23	68
Homer	M	CS	7	50
Igor	M	MIS	21	70

直接攻击

```
Q1 : SELECT COUNT(*)
     FROM Students
     WHERE Sex = 'F' AND Program = 'CS'
```

Returns count 1

```
Q2 : SELECT AVG(Grade Ave.)
     FROM Students
     WHERE Sex = 'F' AND Program = 'CS'
```

Returns 70: average for a single student

跟踪攻击

```
Q3 : SELECT COUNT(*)
     FROM   Students
     WHERE  Programme = 'CS'
```

Returns count 4

```
Q4 : SELECT COUNT(*)
     FROM   Students
     WHERE  Programme = 'CS' AND Sex = 'M'
```

Returns count 3

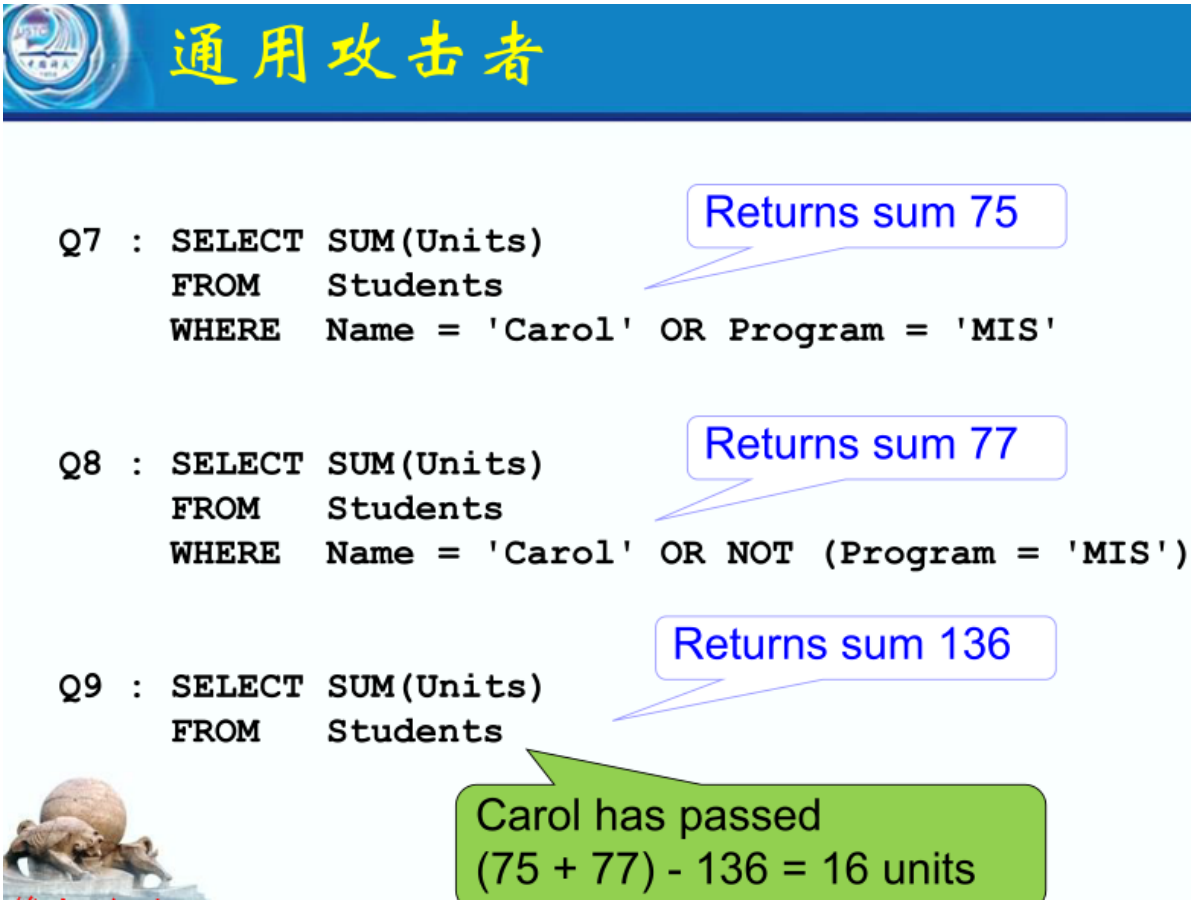
```
Q5 : SELECT AVG(Grade Ave.)
     FROM   Students
     WHERE  Program = 'CS'
```

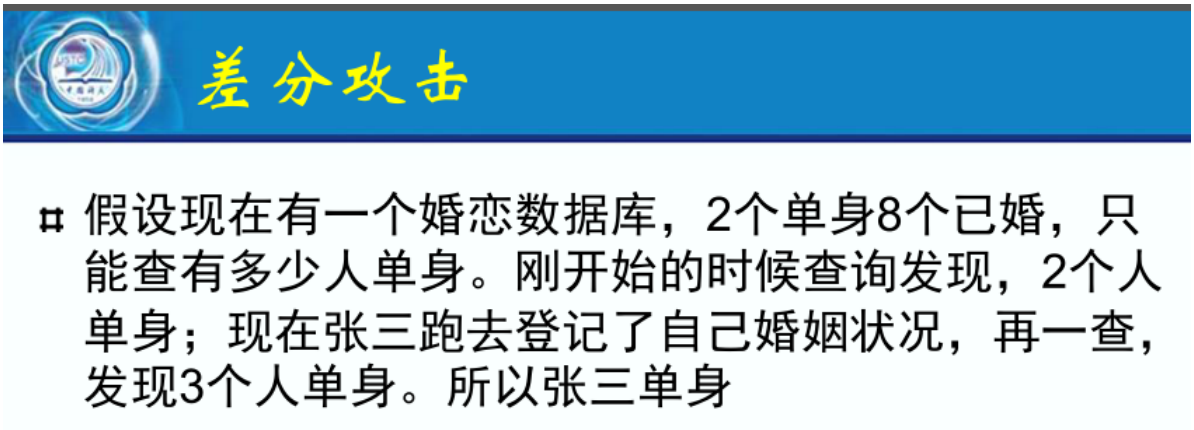
Returns average 61

```
Q6 : SELECT AVG(Grade Ave.)
     FROM   Students
     WHERE  Program = 'CS' AND Sex = 'M'
```

Returns average 58

Carol's grade average:
$4 \cdot 61 - 3 \cdot 58 = 70$





差分隐私：加入随机噪声

数据备份策略：完全备份，增量备份，差分备份

完全备份：对整个系统进行包括系统和数据的完全备份

增量备份：每次备份数据=上一次**备份**后增加和修改的数据

差分备份：每次备份数据=上一次**完全备份**后增加和修改的数据

主流存储技术：RAID（独立冗余磁盘阵列）

RAID 0没有冗余或错误修复能力

RAID 1称为磁盘镜像：把一个磁盘的数据镜像到另一个磁盘上

RAID5：1+0，但是奇偶校验信息来恢复，至少需要三块硬盘

RAID10

数据容灾：

数据备份是数据容灾的基础

容灾不是简单备份

容灾不仅是技术，是一个工程，不仅包含技术，还有一整套容灾流程、规范和具体措施

容灾等级：0（本地备份、本地保存的冷备份），1（本地备份，异地保存的冷备份），2（热备份站点备份），3（活动互援备份）

容灾技术：互连，远程镜像，快照，存储虚拟化

十七、基于代码的访问控制

基于代码的访问控制：访问控制许可被赋值给代码环境，主要例子：Java安全模型，.NET安全框架等

与代码相关的安全属性有：

代码来源

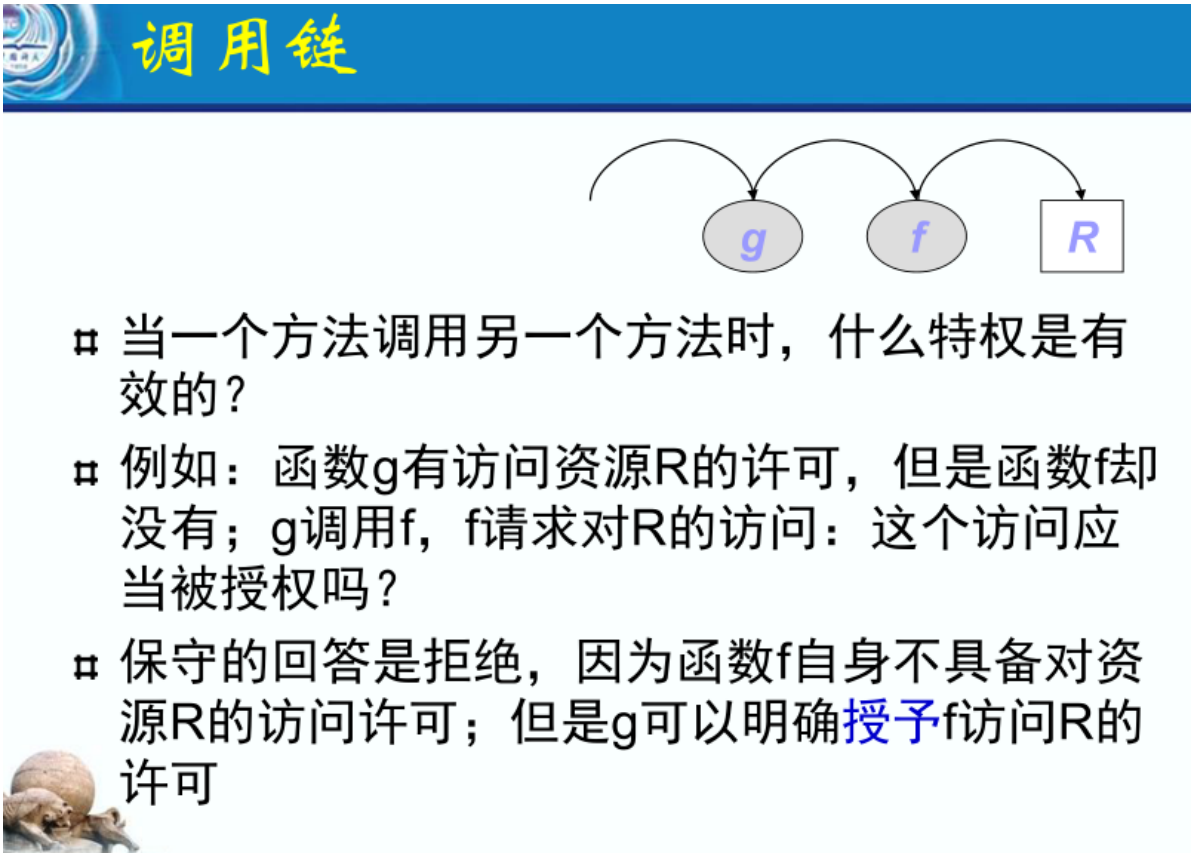
代码URL（内部网或者因特网）

代码签名

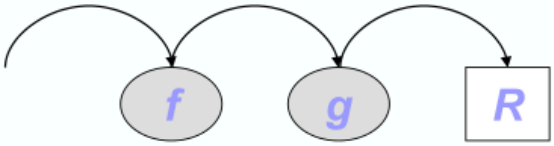
代码标识（经过检验的可信任程序）

代码校验

发送者的身份等



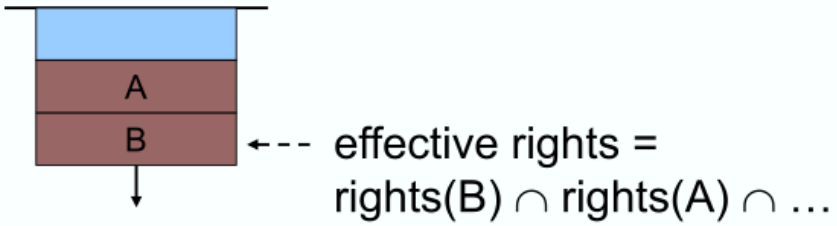
调用链



- 例子：函数g有访问资源R的许可，但是函数f却没有；f调用g，g再去访问R：这个访问应当被授权吗？
- 困惑的代理人问题：一个“不可信”的实体请求一个“可信”的实体去做一些非法的行为
- 保守的回答还是拒绝；但是g可以明确的断言这个权限

堆栈游走（stack walk）

- 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

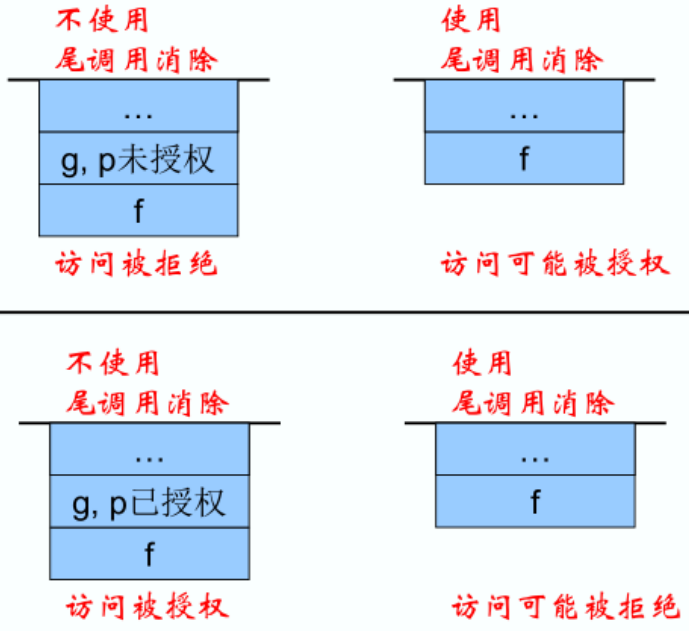


断言：允许不可信的调用者调用可信的方法，许可断言附着在当前栈帧上，当从调用部件返回时撤销，对战遍历在遇到许可断言的帧并授予这个许可时终止，至此被检查的所有帧都有这个许可

尾调用消除：

尾调用清除对堆栈审查的影响

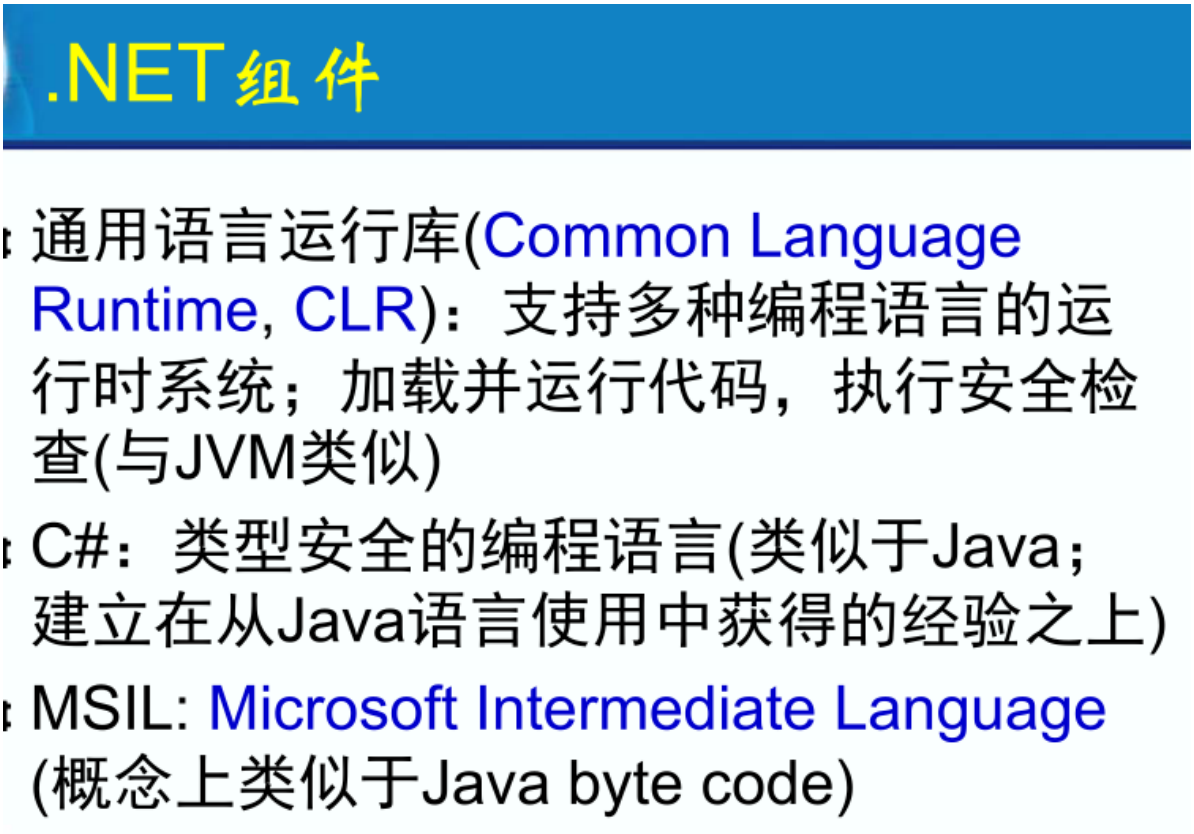
- 尾调用：函数g的最后一个指令是函数调用指令f
- 一旦f被调用，g的帧就不需要了，他被f的新帧覆盖掉

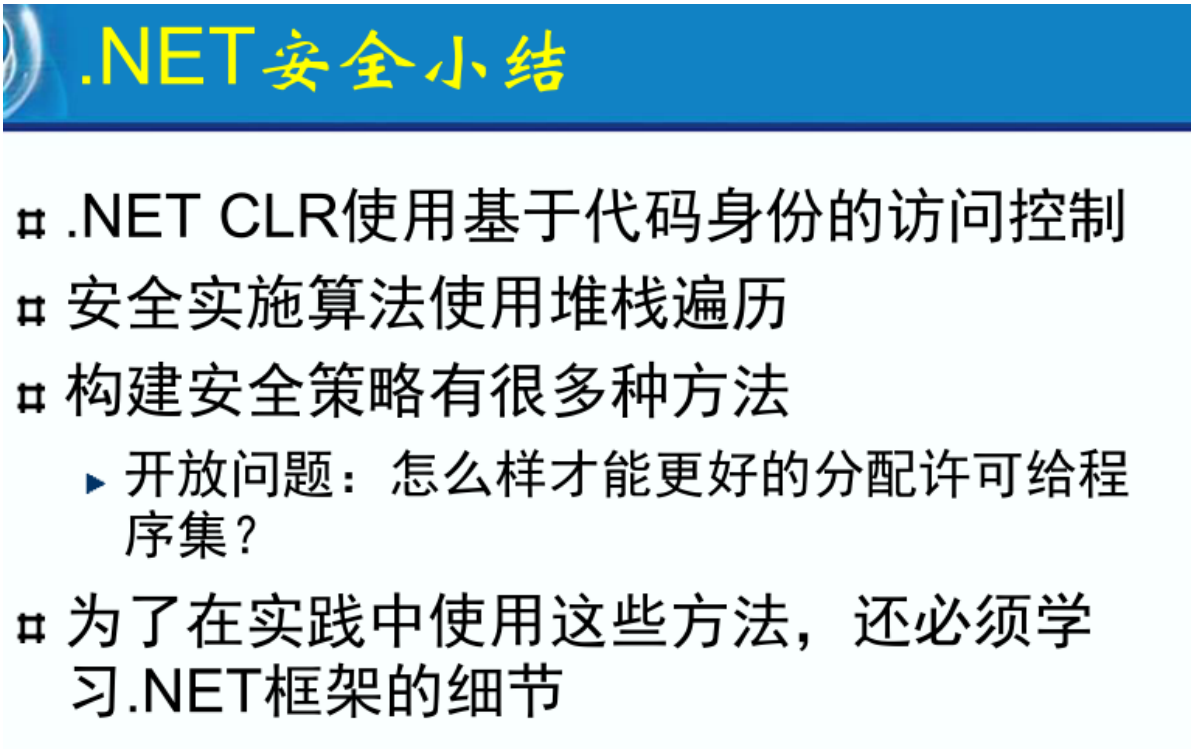


安全

Java安全

- Java：强类型的面向对象的语言；通用目的编程语言
- Java是类型安全的；Java对象的类型可以通过存储在对象上的类标签指示
- 静态(和动态)类型检查：检查操作在执行时总是得到正确类型的参数
- 安全优势：没有指针类型；指针导致的内存访问缺陷在C/C++中非常普遍





十八、云计算安全

谷歌云计算三驾马车：GFS，MapReduce，BigData

云计算的主要特性：

1. 按需自主服务：客户根据需要获得所需计算资源
2. 泛在接入：通过各种终端随时随地使用服务
3. 资源池化：将资源提供给多个客户使用，这些物理的、虚拟的资源根据客户的需求进行**动态分配**或重新分配

4. 快速伸缩性：根据需要快速、灵活、方便地获取和释放计算资源。对于客户来讲，这种资源是“无限”的，**能在任何时候获得所需资源量**
5. 服务可计量：云计算可按照多种计量方式自动控制或量化资源

云计算的服务模式：

1. SaaS（软件即服务）
2. PaaS（平台即服务）
3. IaaS（基础设施即服务）

云计算的部署模式：

1. 私有云：对某个特定客户，分场外（云服务商拥有管理运营）场内*客户自己搞的）
2. 共有云
3. 社区云：特定的客户群体（分场外场内）
4. 混合云

（私有链、公有链、联盟链）

十九、入侵检测

PDR/PPDR

策略：模型的核心，意味着网络安全要达到的目标

防护：安全的第一步，基础是检测和相应结果

检测：弥补漏洞发现或攻击手段发明与响应的防护间的时间差

响应：发现攻击后，需要及时对系统反应

入侵检测是PDR模型的关键

入侵：尝试破坏（外部攻击者）或者误用（内部用户）系统

入侵检测：系统的运行状态进行监视，发现各种攻击企图、攻击行为或者攻击结果，保证系统资源的保密性、完整性、可用性

入侵检测系统（IDS）：进行入侵检测的软硬件组合

分布式入侵检测系统（DIDS），一个里程碑产品

IPS（入侵防御系统）：IDS（发现）和防火墙（阻断）联动，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或是具有伤害性的网络资料传输行为

DPI（深度包检测）：基于应用层的流量检测和控制技术（识别IP包载荷）

DFI（深度/动态流检测）：基于流量行为，不同应用类型在会话连接或数据流上的状态不同，建立流量特征模型来鉴别应用类别

态势感知：基于环境的，动态整体洞悉安全风险的你呢管理，从全局视角提升对安全威胁的发现识别、理解分析、相应处置能力的一种方式，是安全能力的落地

IDS基本功能部件：信息收集->信息分析->结果处理

入侵检测性能的关键参数：误报（异常活动定义为入侵），漏报

入侵检测方法：

异常检测：入侵是异常活动的子集。首先总结**正常操作应该具有的特征（用户轮廓）**，当用户活动与**正常行为有重大偏离**时即被认为是入侵。指标：**漏报率低，误报率高**。不需要对每种入侵行为进行定义，因此能有效检测未知的入侵

误用检测：所有的入侵行为都有可被检测到的特征。收集**非正常操作的行为特征**（特征提取），建立相关的攻击特征库（要经常更新），当监测的用户或系统行为与库中的记录**相匹配**时，系统就认为这种行为是入侵。指标：**误报低，漏报高**。攻击特征的细微变化，会使得误用模式无能为力

入侵检测系统分类：

1. 基于主机：系统获取数据的依据是系统运行的主机，保护目标也是主机
2. 基于网络：系统获取的数据是网络传输的数据包，保护的是网络的运行
3. 混合型

Snort：入侵检测工具（IDS）

二十、网络侦查

网络入侵的前奏是网络侦查，即收集目标网络/主机的有用信息

社会工程：利用人的弱点进程诸如欺骗、伤害等危害手段，获得利益

社会工程学是未来十年最大的安全风险，比如电信诈骗，网络钓鱼

网卡设为杂凑模式，则该网卡就可接收任何流经它的数据包，不论数据包的目标地址是什么

TCPDUMP：经典的网络监控和数据捕获嗅探器

Wireshark（Ethereal）：抓包软件，用于网络监听

网络扫描：端口扫描：就是通过连接到目标系统的TCP或UDP端口来确定什么服务正在运行，也可以识别目标系统的OS类型，或识别某个应用或某个特定服务的版本号（扫描希望隐藏自己）

Nmap：端口扫描和操作系统识别扫描器

二十一、拒绝服务攻击

DoS：拒绝服务攻击。通过某些手段使得目标系统或者网络不能提供正常的服务，或明显降低系统性能。利用了TCP/IP的缺陷和网络协议栈的缺陷

分类：发送**少量**恶意包，使得系统或网络瘫痪，修正bug或者管理员干预恢复；发送**大量**垃圾数据，使得系统或网络无法响应正常请求，恢复系统不需要或只需要少量的人工干预；物理硬件的移除破坏

DDoS：分布式拒绝服务攻击。攻击者控制大量的攻击源，同时对攻击目标发起一种DoS，使目标宽带迅速耗尽

主要两个步骤：攻占代理主机和向目标发起攻击。

攻击者->（传令）**主控端**->（传令）**代理端**->真正发起攻击

特点：攻击力巨大，难以发现攻击者

DRDoS：分布式反弹DoS。利用反弹服务器（就是会回应）实现攻击。步骤和DDoS的区别是，向反弹服务器发送大量欺骗数据包，源地址为受害目标服务器（是为了攻击接受回应的服务器）

比DDoS更难抵御，反弹技术可以使攻击时的洪水流量变弱，最终才在目标机汇合为大量的洪水，其攻击规模也比传统DDoS攻击大得多，更难追查攻击来源。