

2023-2024学年春季学期

计算机体系结构安全
Computer Architecture Security

授课团队：史岗，陈李维

计算机体系结构安全

Computer Architecture Security

[课程内容]

计算机体系结构安全概论

计算机体系结构基础

编译和操作系统基础

计算机内存架构基础

安全体系结构原理（原理、体系结构实践、微体系结构实践）

计算机内存安全

先进计算架构安全（云、大数据系统、机器学习平台）

计算机体系结构安全

Computer Architecture Security

[考核方式]

- **平时表现 (10%)**
 - 考勤、课堂表现
- **平时作业 (50%)**
 - 每2次课布置1次作业，调研为主，提交调研报告
- **期末考试 (40%)**
 - 1次大作业/考试，调研或实验任务，提交调研或实验报告，准备PPT，课堂展示汇报（最后1节课）

计算机体系结构安全

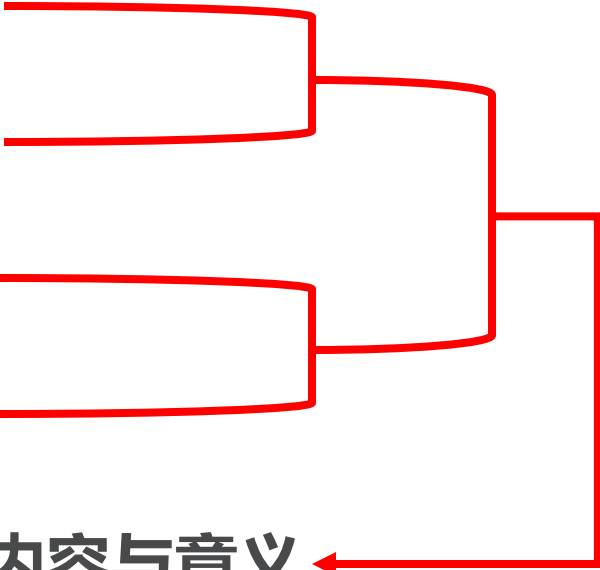
Computer Architecture Security

[第1次课] 计算机体系结构安全概论

授课教师：史岗

授课时间：2024. 2. 26

内容概要

- 安全及其属性
 - 攻击与漏洞
 - 漏洞分类及产生的根源
 - 安全机制分类
 - 体系结构与安全机制关系
 - 计算机体系结构安全课程内容与意义
- 

内容概要

- **安全及其属性**
- 攻击与漏洞
- 漏洞分类及产生的根源
- 安全机制分类
- 体系结构与安全机制关系
- 计算机体系结构安全课程内容与意义

○安全的含义

○中文的含义

- 文言文中“安”代表现代汉语中的“安全”，《辞海》对“安”字的第一个释义就是“安全”。
- 《周易·系辞下》：“是故君子安而不忘危，存而不忘亡，治而不忘乱，是以身安而国家可保也。”
- 《现代汉语词典》对“安全”的解释是：“没有危险；不受威胁；不出事故”。

典故：

“曲突徙薪”
《汉书·霍光传》

“魏文王问扁鹊”
《鹖冠子·世贤》

○安全的含义

○英文的含义

○Safety

- 自然属性的安全

- 自然灾害

地震、飓风、海啸

- 非人为攻击，不确定性

○Security

- 人为属性的安全

- 故意攻击

盗取资料、伤害某人

- 人为攻击，强确定性

*Safety*侧重被保护的状态，*Security*侧重保护的行动。

○安全的定义

○"安全"这个词在不同的上下文中有不同的含义，但通常它指的是一种状态或条件，即在这种状态或条件下，人、物品或信息不会受到伤害或损失。

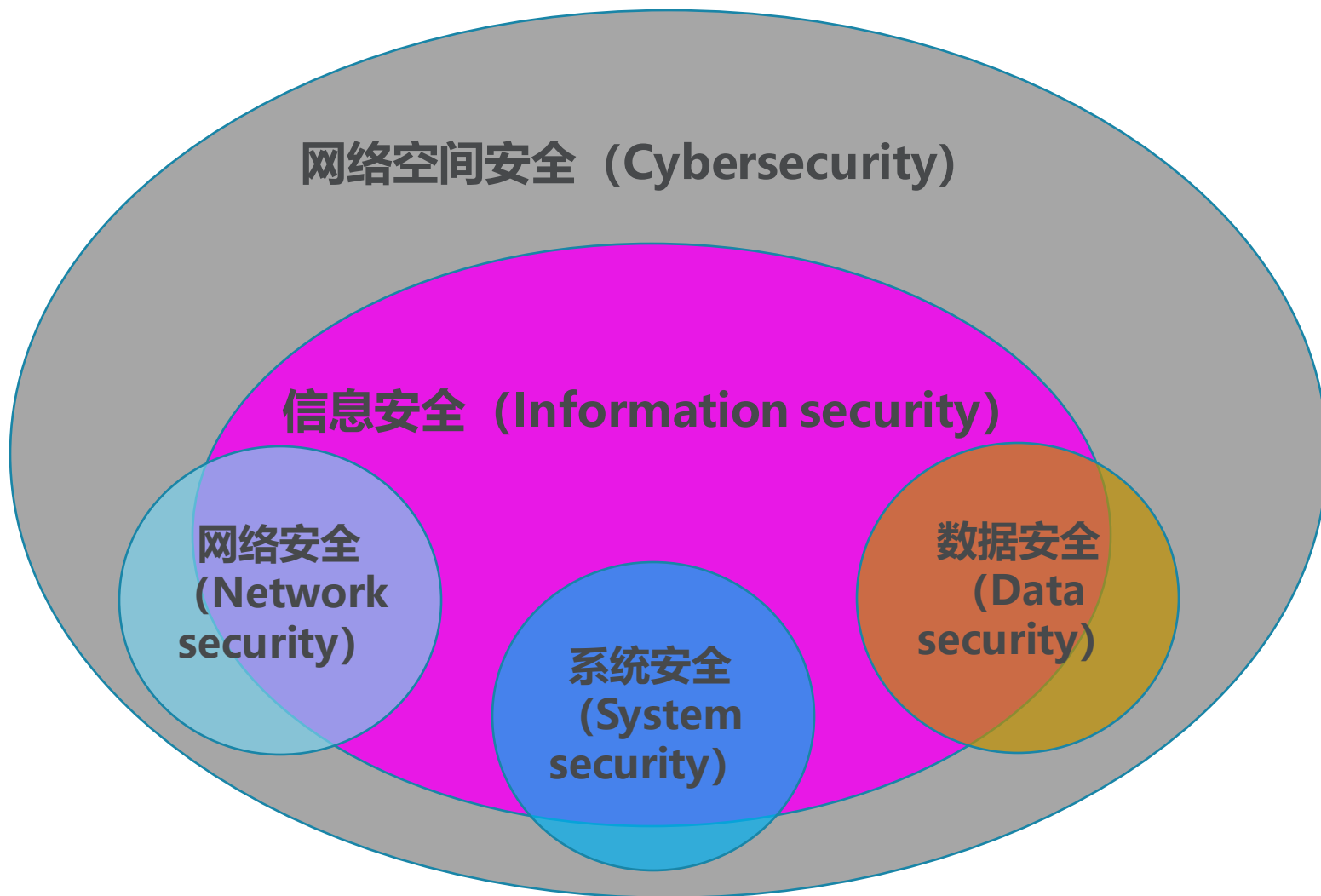
○在物理环境中，安全指的是人们不会受到身体伤害的环境或条件。

○在社会环境中，安全指的是人们的生命、财产和权利不会受到侵犯的状态。

○在计算机和网络环境中，安全通常指的是信息和系统不会受到**未经授权**的访问、使用、修改或破坏的状态，这就是我们通常说的**信息安全或网络空间安全**。

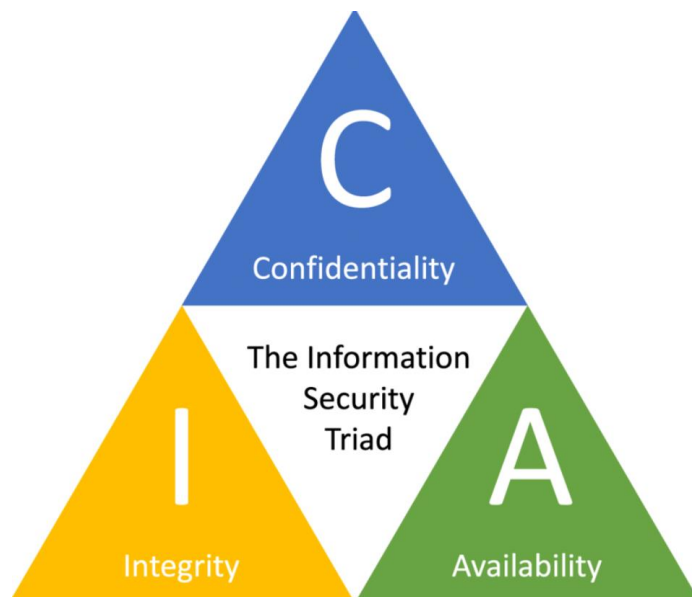
○信息安全的层次

- 物理安全**：关注于保护计算系统的物理组件，以防止未经授权的访问、破坏或盗窃。
- 系统安全**：涉及到保护整个计算系统，包括硬件、操作系统、应用程序等，以防止恶意软件、未经授权的访问和其他威胁。
- 网络安全**：涉及到保护计算机网络和网络可访问资源，包括网络设备、网络软件，以防止未经授权的访问、滥用和拒绝等
- 数据安全**：旨在保护存储在计算机系统中的数据，以防止数据泄露、损坏或未经授权的访问。
- 内容安全**：关注于防止不适当、有害或违法内容的传播。这可以包括网络过滤、版权保护和反网络欺凌措施。



○安全的最基本属性（核心三件套）：

- 机密性 (Confidentiality)**：信息仅被合法的实体访问，不泄漏给未授权的实体；系统仅被合法的实体使用
- 完整性 (Integrity)**：信息只能由授权实体修改，不被偶然或蓄意地篡改、伪造、丢失等；系统没有受到未经授权的操控进而完好无损地执行预定功能
- 可用性 (Availability)**：信息能够随时被授权实体访问并使用；系统及时工作并向授权用户提供所需的服务



○其他安全属性

- 可控性 (Controllability)**：确保某个实体（用户、进程等）身份的真实性，确保信息内容的安全合法，确保系统状态可被授权方所控制，通常通过监控、审计等手段对活动和内容进行监管和控制。
- 不可抵赖性 (Non-repudiation)**：确保一个操作或事件的发起者和接收者不能否认他们的行为，通常通过数字签名、审计日志等手段来实现。
- 可存活性 (Survivability)**：确保系统在面对各种攻击或错误的情况下继续提供核心服务，而且能够及时地恢复全部的服务。
- 可认证性 (Authenticity)**：保证信息的服务者和使用者都是真实声称者，防止冒充和重演的攻击。
- 可审计性 (Auditability)**：保证使用者的行为有证可查，并能对出现的安全问题提供调查依据和手段。

内容概要

- 安全及其属性
- **攻击与漏洞**
- 漏洞分类及产生的根源
- 安全机制分类
- 体系结构与安全机制关系
- 计算机体系结构安全课程内容与意义

○攻击的本质是对机密性、完整性和可用性的破坏。

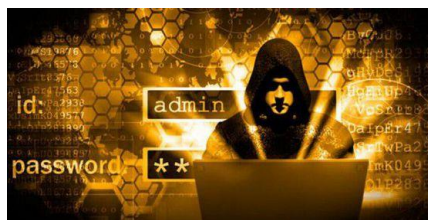
- 对机密性的破坏：**攻击者窃取敏感信息，如用户的个人信息、公司的商业秘密等，这就破坏了信息的保密性。
- 对完整性的破坏：**攻击者篡改信息，如修改数据库中的数据、更改网站的内容等，这就破坏了信息的完整性。
- 对可用性的破坏：**攻击者使系统或网络无法正常工作，如发动拒绝服务攻击、破坏系统的正常运行等，这就破坏了信息的可用性。

○攻击模型

- 攻击模型（Attack Model）是对计算机系统或网络中可能发生的攻击进行描述和建模的方式。它定义了攻击者的能力、目标、行为和假设，以帮助分析和评估系统的安全性，并采取相应的防御措施。

○最基本的攻击模型

- 探索（发现漏洞），攻击（利用漏洞，实现攻击）
- 漏洞是探索的目标，是攻击的基础与前提
- 漏洞利用是攻击最关键的步骤



攻击者

发现
利用



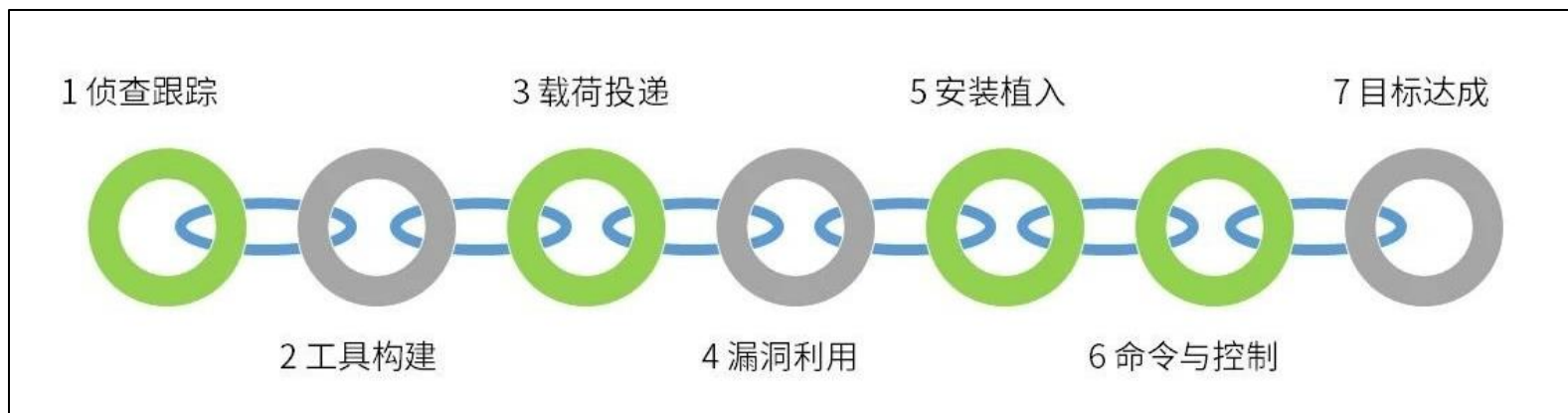
漏洞



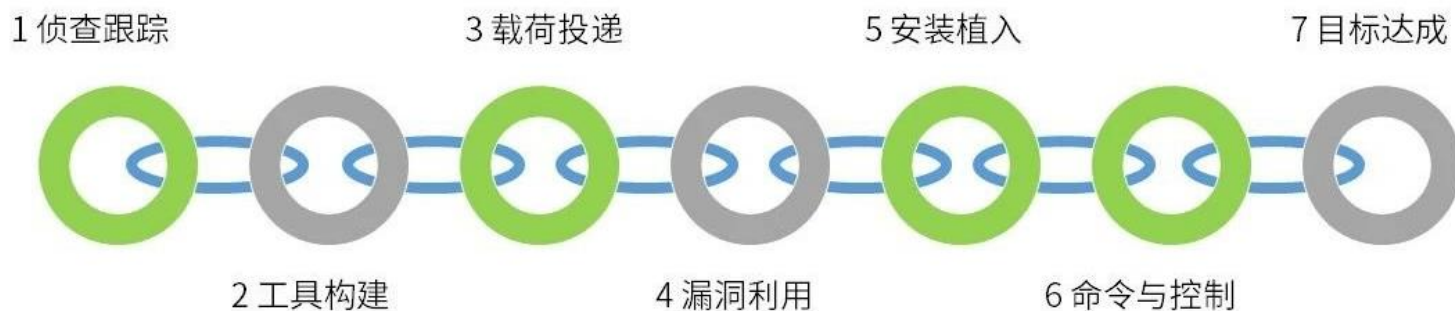
恶意
目的

○七步杀伤链模型

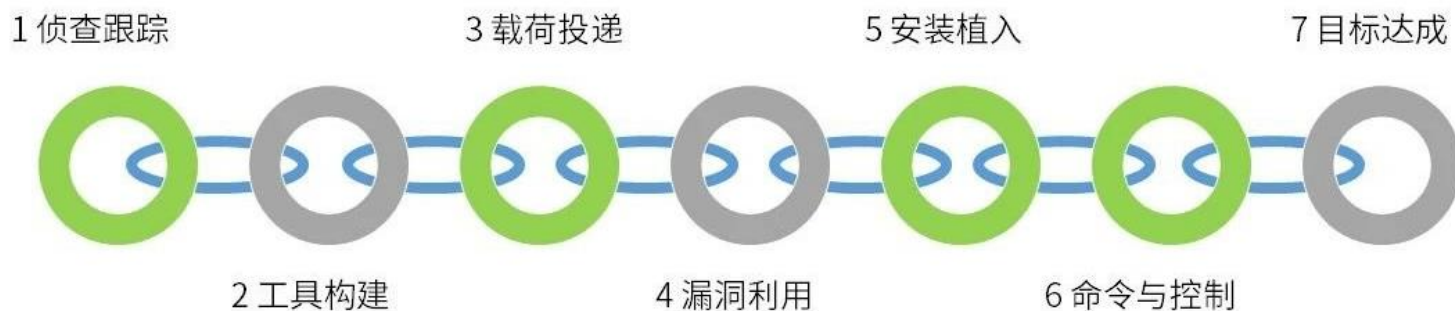
- 描述网络攻击过程的模型，由美国洛克希德·马丁公司提出。
- 内容包括成功的网络攻击所需的七个阶段：侦察跟踪、工具构建、载荷投递、漏洞利用、安装植入、命令与控制、目标达成。



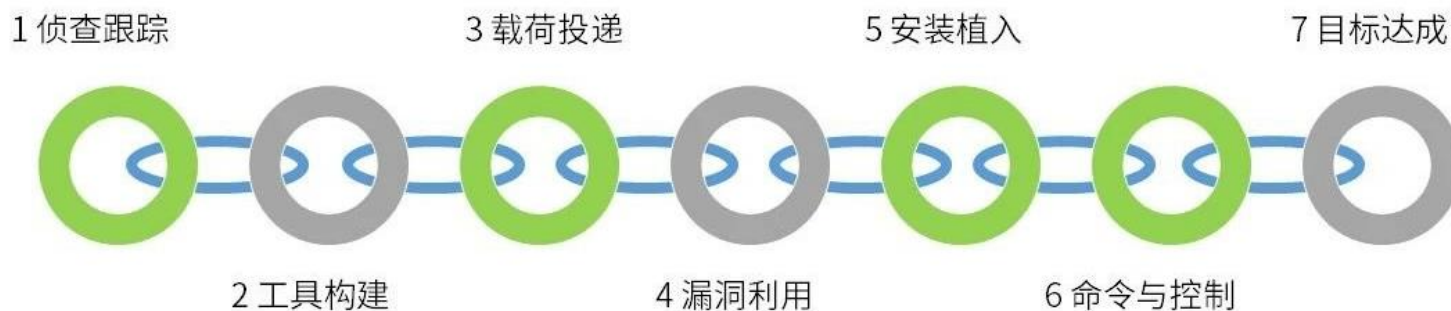
- 侦察跟踪 (Reconnaissance) : 攻击者收集目标系统的信息, 如IP地址、操作系统、开放的端口、运行的服务等 (收集可利用的漏洞信息)
- 工具构建 (Weaponization) : 攻击者创建或获取用于攻击的恶意代码, 如病毒、蠕虫、木马等 (基于可利用的漏洞, 构建针对性的恶意代码)



- **载荷投递 (Delivery)** : 攻击者将恶意代码传送到目标系统, 这可以通过各种方式实现, 如电子邮件、网页、USB设备等 (将恶意代码传送到**漏洞位置**)
- **漏洞利用 (Exploitation)** : 恶意代码利用目标系统的漏洞, 以执行攻击者的命令

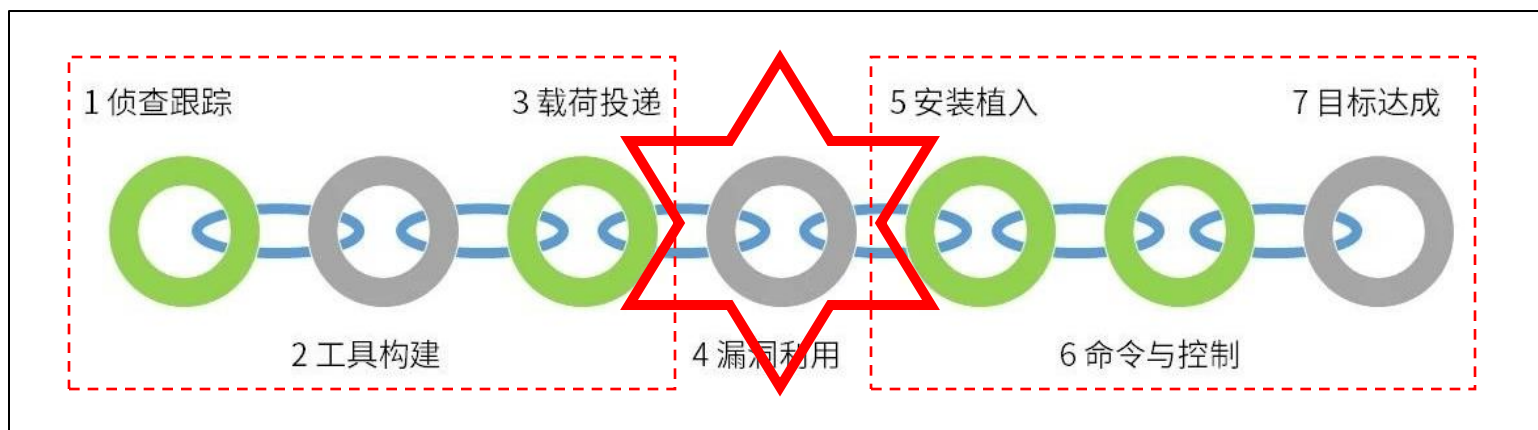


- **安装植入 (Installation)** : 恶意代码在目标系统上安装自己, 以持久化地控制目标系统
- **命令与控制 (Command & Control)** : 攻击者通过网络远程控制目标系统, 发送命令并接收结果
- **目标达成 (Actions on Objectives)** : 攻击者利用对目标系统的控制权, 实现自己的目的, 如窃取敏感信息、破坏系统、发起其他攻击等



○七步杀伤链

- 第一步，第二步，第三步，均是为了实施漏洞利用，而进行的前置步骤
- 第五步、第六步、第七步，均是在漏洞利用的基础上，进一步展开的恶意行动
- 因此，漏洞利用是整个攻击模型的核心部分，漏洞是整个攻击过程的关键所在



收集信息、网络分析、制作
恶意软件、发送出恶意软件

从攻击者角度出发，
对攻击进行的准备

桥梁

在受害者系统上安装后门或恶意软件、
恶意软件执行获取机密信息

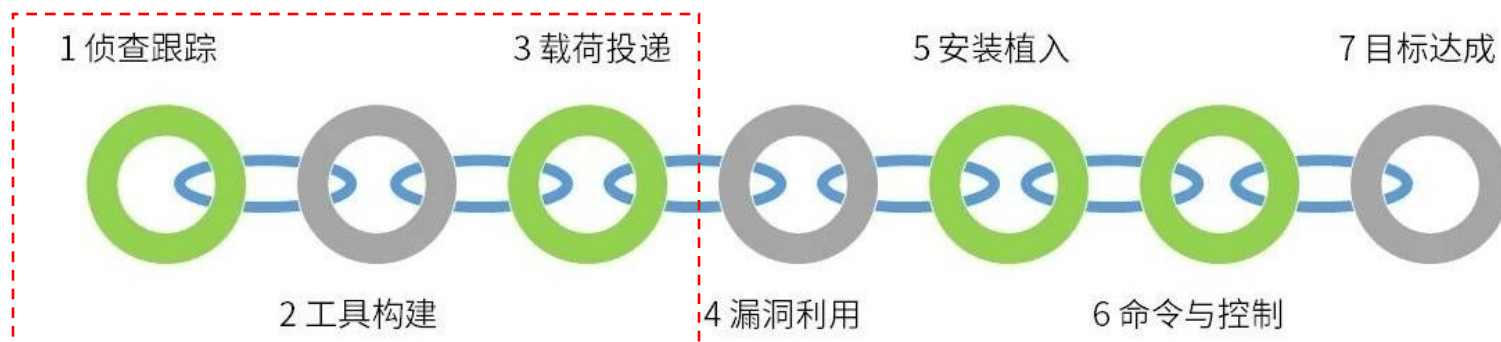
从受害者角度来看，恶意软件
已经在受害者机器上运行

○例子：震网APT攻击

- 震网（Stuxnet）是一种著名的网络蠕虫病毒，被认为是世界上首个针对工业控制系统编写的破坏性蠕虫。
- 2010年6月17日，白俄罗斯公司VirusBlockAda的安全研究人员发现一种能感染可移动存储设备的恶意软件。
- 2010年7月，震网蠕虫攻击事件浮出水面，引发了国际主流安全厂商和安全研究者的全面关注，各大安全厂商，著名安全研究者，以及多国的应急组织和研究机构，都投入到了全面的分析接力中。
- 2010年11月，伊朗总统艾哈迈迪内贾德公开承认，一种计算机病毒对其（核）离心机中为数不多的几台机制造了一些问题。

○例子：震网APT攻击

- 侦查跟踪、工具构建：美国在前期对伊朗核设施进行了长达数年的信息收集（核设施施工建设、内部工控系统SCADA架构、工控软件Wincc等），并设计蠕虫病毒。
- 载荷投递：前期是特工将U盘插入目标主机进行传播，后期是通过网络攻击5家伊朗核设施供应商技术人员的电脑或U盘，间接将病毒带入核设施工厂的。



○例子：震网APT攻击

○漏洞利用、安装植入：

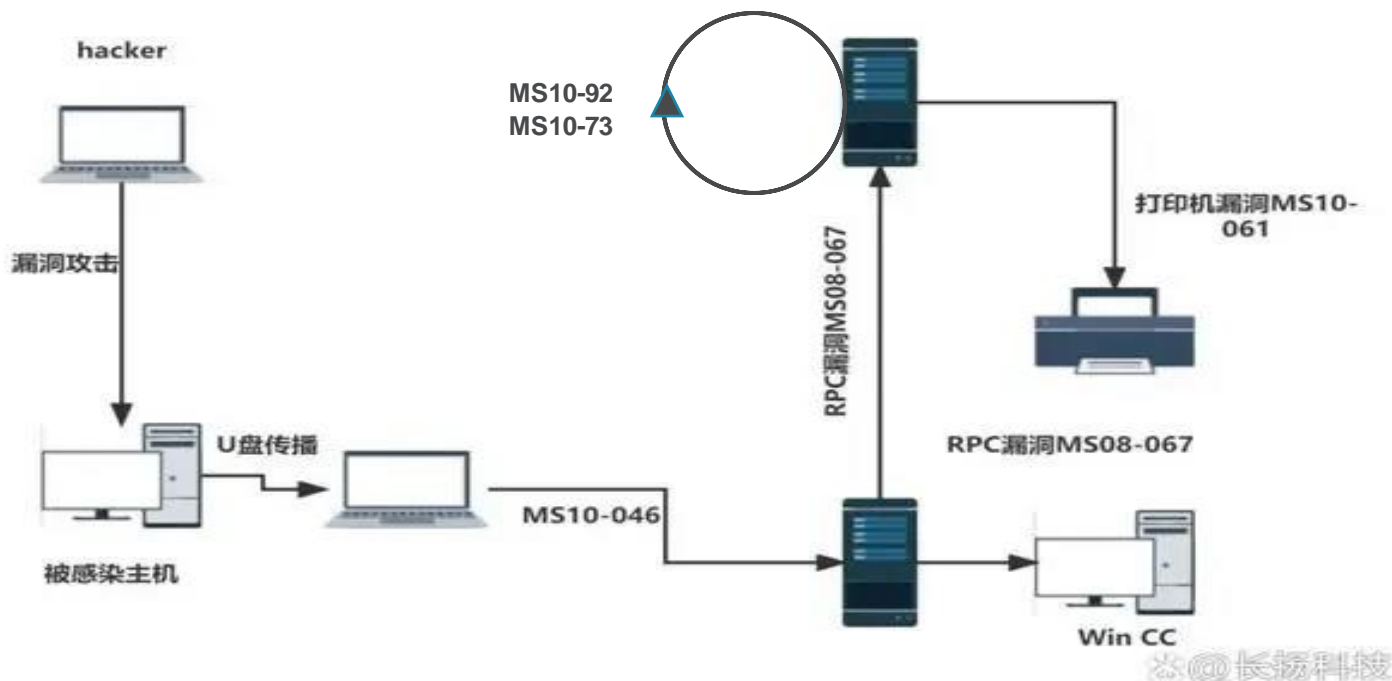
快捷方式文件解析漏洞 (MS10-046)

RPC远程执行溢出漏洞 (MS08-067)

打印机后台程序服务漏洞 (MS10-061)

任务计划程序权限提升漏洞 (MS10-092)

内核模式驱动程序权限提升漏洞 (MS10-073)



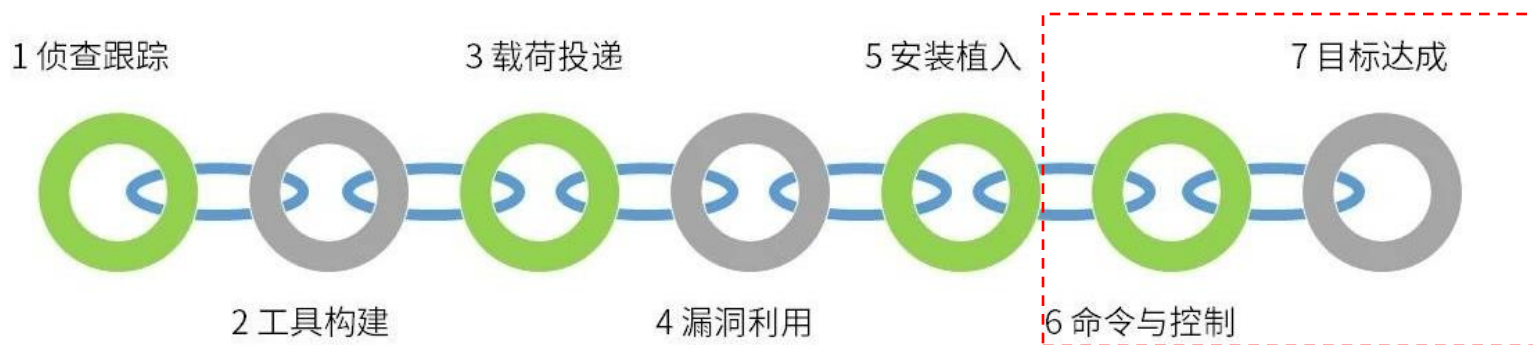
○例子：震网APT攻击

- 命令与控制、目标达成：寻找SCADA工控体系的机器，也就是安装了Wincc和Step7工控软件的机器，这种主机可以为PLC下发控制指令，从而控制泄压阀门开关、离心机转速、设备的运行停止等等，从而达成控制核设施的目的。

2个针对西门子SIMATIC 的WinCC系统的漏洞：

通过WinCC硬编码漏洞访问系统的SQL数据库

通过DLL加载策略漏洞读取函数的Hook



- 对攻击过程分析可知

- 漏洞利用是整个攻击模型的核心环节，是整个攻击过程的关键。



- **漏洞利用**：通过**利用**软件、硬件或系统中存在的**漏洞**，来得到计算机的控制权（使自己编写的代码越过具有漏洞的程序的限制，从而获得运行权限），从而获取未授权的访问权限、执行恶意代码、窃取敏感信息等。



○漏洞利用和攻击关系

○区别

- 漏洞利用是一种技术行为，强调的是攻击者对系统或软件漏洞的研究和利用，以实施攻击。而攻击是一个更广义的概念，包括多种手段和方式，不仅限于漏洞利用。
- 攻击可以通过多种途径实施，包括社会工程学、恶意软件、网络钓鱼等，不仅仅依赖于漏洞利用。攻击可以利用漏洞来实现特定的目标，也可以通过其他方式进行，例如诱骗用户泄露密码或使用恶意软件进行攻击。
- 漏洞利用是攻击的一部分，攻击还包括其他一系列的行为，如侦察目标、入侵系统、窃取数据等。

○漏洞利用和攻击关系

○联系

- 漏洞利用是攻击的一种手段或方式。攻击者通过利用计算机系统或软件中的漏洞来实施攻击，从而获取未经授权的访问权限或执行恶意操作。
- 漏洞利用是**攻击者利用安全漏洞的过程**，它涉及对系统或软件的分析、研究和实施特定的技术手段，以达到攻击的目的。

对漏洞产生机理的探索，是安全研究的基础和出发点。

内容概要

- 安全及其属性
- 攻击与漏洞
- 漏洞分类及产生的根源
- 安全机制分类
- 体系结构与安全机制关系
- 计算机体系结构安全课程内容与意义

- **漏洞**，或**脆弱性**（Vulnerability），是指计算机系统安全方面的缺陷，使得系统或其应用数据的机密性、完整性、可用性等面临威胁。
 - 在《GB/T 25069-2010 信息安全技术术语》中，将**脆弱性**定义为“资产中能被威胁所利用的弱点（缺陷）”
- **缺陷**通常指的是软件或硬件中存在的错误或问题，从而影响系统正常运行、甚至被恶意利用。
- **安全威胁**是指可能对信息系统造成损害或破坏的潜在因素

○安全威胁和漏洞

- 区别：**威胁是一种可能性，而漏洞是一种实际存在的弱点。威胁是对安全的潜在风险，而漏洞是威胁成为现实的途径。
- 联系：**威胁需要通过漏洞来实现。如果一个系统没有漏洞，那么它受到威胁的可能性极小。反过来，如果一个系统有漏洞，那么它就可能面临威胁。



○错误、漏洞及后门的区别：

○错误 (Bug)

- 是指设计和实现上的缺陷。

○漏洞 (Vulnerability)

- 是一类特殊的错误，即可被攻击利用，产生安全威胁的错误。

- 通常是设计者无意识留下的。

○后门 (Backdoor)

- 是指设计者故意留下的可被利用的错误。

- 后门的隐蔽性更高，危害性更大。

○根据漏洞的起源和形态分类

○软件漏洞

- 源于软件设计的错误，软件的形态呈现
- 主要由于软件编程人员的疏忽

○硬件漏洞

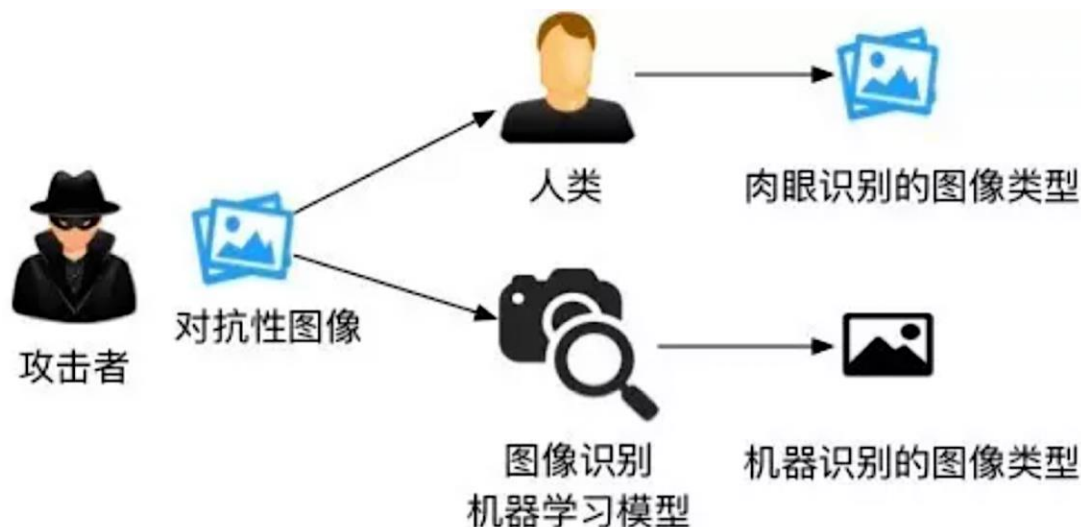
- 由于硬件设计的错误，硬件的形态呈现
- 主要由于硬件设计人员的疏忽

○结构漏洞

- 源于软硬件结构设计的错误，通常以软件的形态呈现
- 不仅仅是设计人员的疏忽，往往有更深层次的原因
 - 软件结构漏洞：软件架构设计上的问题
 - 硬件结构漏洞：硬件结构设计上的问题
 - 体系结构漏洞：体系结构设计上的问题

○软件漏洞（算法漏洞）

- 现阶段人工智能的发展很多都是基于深度神经网络。随着神经网络模型越来越复杂，存在一些难以被发现的错误逻辑，可被利用，破坏算法的功能。
- 例如对抗样本攻击，在人脸照片上添加少量干扰或将对抗补丁打印制作后由真人佩戴，就能欺骗人脸识别系统。
- 例如对自动驾驶汽车搭载的感知系统攻击，在障碍物上加装干扰因素，改变表面颜色、修改形状，就可能误导感知算法，让车辆无法感知到障碍物。



○硬件漏洞

○硬件故障

- 硬件设备可能因为制造缺陷或使用过程中的损耗而出现故障，例如内存条出现故障导致数据错误，硬盘故障可能导致数据丢失。

○电磁泄漏

- 硬件设备在工作过程中可能会产生电磁信号，这些信号可能被攻击者捕获并分析，以获取敏感信息。这种攻击通常被称为侧信道攻击（电磁侧信道）。

○物理篡改

- 如果攻击者可以直接接触到硬件设备，他们可能会尝试物理篡改设备，以改变其行为或获取敏感信息。例如在设备上安装额外的硬件。

○供电和时钟攻击

- 通过控制设备的电源或时钟信号，攻击者可能可以引导设备进入异常状态，以获取敏感信息或影响其行为。例如，电源分析攻击就是通过分析设备的电源消耗来推断其内部状态。

○结构漏洞

○软件结构漏洞

- 软件结构漏洞，由于软件结构设计不合理，对安全性考虑不足，导致的安全漏洞。
- 例如整型溢出、内存泄漏、缓冲区溢出

○体系结构漏洞

- 由于计算机体系结构设计不合理，对安全性考虑不足，导致的安全漏洞。
- 例如代码注入漏洞、提权漏洞

○硬件结构漏洞

- 由于硬件结构设计不合理，对安全性考虑不足，导致的安全漏洞。
- 例如熔断、幽灵、Rowhammer漏洞

○软件结构漏洞

○整型溢出

- 未正确计算或转换所产生数字
- 漏洞表象：由于整数变量超过其最大存储值，发生溢出
- 漏洞根源：处理器定点运算部件中寄存器长度限制，如32位寄存器的范围限制在 $0 \sim (2^{32}-1)$

○缓解措施：

- 软件层面：使用合适的数据类型，检测数值范围，或者发现错误，进行错误处理
- 系统结构层面：使用安全编程语言，程序编译时会自动检查数值的溢出
- 微结构层面：让硬件提供数值范围检查和处理，即便软件层面没检查，也不会导致错误

○软件结构漏洞

○缓冲区溢出

- 对输入的数据存放位置缺少正确的验证而产生的漏洞
- 漏洞表象：软件层面上是缺少正确的边界数据验证
- 漏洞根源：内存设计没有考虑数据越过边界问题

○缓解措施：

- 软件层面：在数据赋值时，加入越界判断
- 系统结构层面：
 - 加入Canary（金丝雀）位，用来检测是否发生溢出
 - 设计专门的数据结构，如胖指针，指定数据的上界和下界，判断是否发生溢出

○软件结构漏洞

○内存泄露

- 内存使用后没被释放，导致内存无法被其他程序使用，耗尽系统内存
- 漏洞表象：程序在使用完内存后，编程人员没有将内存释放
- 漏洞根源：内存设计时，没有考虑内存自动回收机制，内存释放的权利掌握在上层软件的设计人员手中。

○缓解措施：

- 系统结构层面：使用自动内存管理的编程语言，如Java、Python等，可以自动回收不再使用的内存，防止内存泄漏

○体系结构漏洞

○代码注入

- 攻击者把攻击代码作为数据，注入程序的函数栈中，将数据作为代码进行执行
- 漏洞表象：程序在执行过程中，把可以动态写入栈中的数据当作代码执行
- 漏洞根源：冯诺依曼结构未区分数据和代码，导致数据可以作为代码进行执行

○缓解措施：

- 微体系结构层面：加入硬件级别的安全特性进行保护，如不可执行位（DEP/NX）

○体系结构漏洞

○提权漏洞

- 攻击者利用特权指令存在的漏洞，提升攻击者的权限
- 漏洞表象：特权指令设计存在漏洞，让攻击者可以利用
- 漏洞根源：计算机系统的特权级设计未考虑周全，特权级划分和切换存在问题

○缓解措施：

- 体系结构层面：为安全设计独立的高权限等级，实施严格的权限等级切换操作。

○硬件结构漏洞

○熔断Meltdown、幽灵Spectre

- 针对乱序执行和分支预测等技术，利用侧信道攻击泄漏系统信息
- 漏洞根源：乱序执行和预测执行遇到异常或预测错误时，CPU恢复之前的状态（保证程序正常执行），但微结构状态并不会恢复，因此便可用侧信道攻击泄漏出微结构状态，从而泄露信息。

○缓解措施：

- 体系结构层面：
 - 进一步隔离，阻止进程或线程共享分支预测器等CPU部件
- 微结构层面：
 - 修改乱序执行和分支预测部件，尽可能恢复之前状态
 - 修改Cache微结构，阻止信息泄漏

○硬件结构漏洞

○Rowhammer攻击

- 反复读取或写入某一行的内存，使得相邻行的内存发生翻转，即0变为1，或者1变为0，从而引发故障
- 漏洞根源：DRAM存储单元的高密度排列，使DRAM在运行过程中产生意外电荷泄漏效应，导致存储器单元泄露电荷并可能造成比特翻转

○缓解措施：

- 微体系结构层面：设计具有错误检测和纠正(EDAC)的内存控制器，即ECC内存，利用额外的位来帮助纠正错误

内容概要

- 安全及其属性
- 攻击与漏洞
- 漏洞分类及产生的根源
- **安全机制分类**
- 体系结构与安全机制关系
- 计算机体系结构安全课程内容与意义

- 回顾安全技术的发展历程，有下列五类主要的安全机制：
 - 权限管理
 - 信任保护
 - 隔离保护
 - 密码保护
 - 主动防御

○权限管理

○权限管理是安全的重心

- 信息安全的一种定义：防止信息被非授权的泄露或更改

○攻击者非法读取或修改数据，获得非法权限

- 权限提升是攻击成功的标志

○合法用户不正常地读取或修改数据，滥用合法权限

- 合法权限干坏事

○权限管理既要防止权限过大，又要保证功能的正常运行；既要分配合理，又要管控牢固。

○信任保护

○信任是安全交互的基石

○系统需具备分辨可信程序与不可信程序的能力

○信任是可度量和可传递的

- 首先，建立一个绝对可信的信任根

- 然后，建立一条信任链。从信任根开始，到硬件平台，再到操作系统，最后到应用软件，一级认证一级，一级信任一级，最终将信任扩展到整个计算机系统。

○ 隔离保护

○ 可信程序隔离

- 假设外界不安全，隔离一个更加安全的环境，供高安全程序运行，如可信执行环境

○ 恶意程序隔离

- 假设内部不安全，隔离一个受约束的执行环境，供危险程序运行，如沙箱隔离

○密码保护

- 加解密能够有效阻止信息泄露，增加了信息理解的难度和门槛

 - 关键数据加密

- 随机化增加了系统的不确定性，增加了攻击成功的难度和门槛

 - 内存地址空间布局随机化ASLR

○主动防御

○主动监测发现，识别潜在风险，防范于未然

- 基于机器学习的威胁发现

○主动调节控制，依据当前风险，动态调整防御强度，随机应变

- 管理员权限动态调节、热补丁

○主动欺骗干扰，让攻击者攻击门槛和难度大大提升

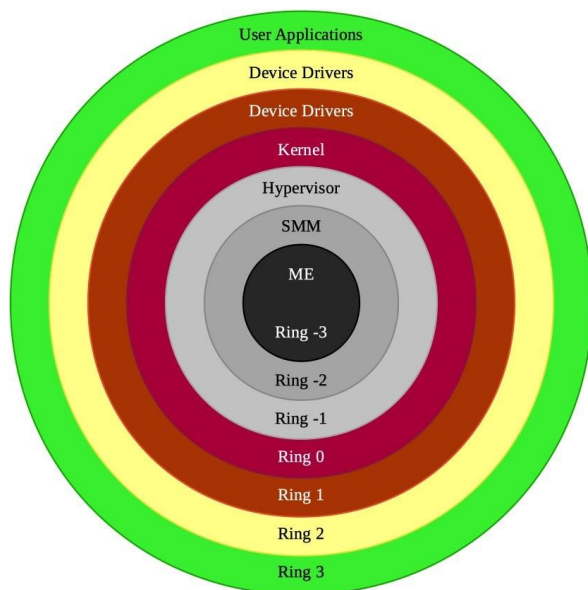
- 蜜罐、蜜洞

内容概要

- 安全及其属性
- 攻击与漏洞
- 漏洞分类及产生的根源
- 安全机制分类
- **体系结构与安全机制关系**
- 计算机体系结构安全课程内容与意义

○权限等级与权限模型

- 如果没有权限等级，所有程序执行的权限一样，一个程序可以任意访问和修改其它程序的代码段或数据段，甚至能修改系统的核心代码！
- 因此，现代计算机都设置多个特权级，让不同的程序运行在不同的特权级

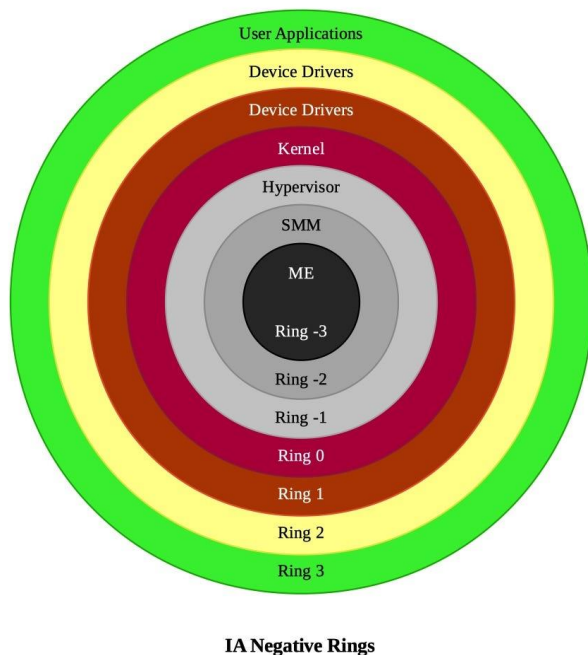


IA Negative Rings

- Ring 3: 应用程序
- Ring 0: 操作系统内核
- Ring -1: Hypervisor (虚拟化)
- Ring -2: SMM(系统管理模式)
- Ring -3: ME(管理引擎)

○Ring-1到-3的加入可能带来安全隐患

- Hypervisor (Ring -1): 虚拟化逃逸攻击, 攻击者在虚拟机中执行恶意代码, 获得对宿主系统或其他虚拟机的访问权限。
- SMM(Ring -2): UEFI rootkits, 这类rootkits的目标是感染系统的UEFI固件, 以在系统启动时插入恶意代码。
- ME(Ring -3): 窃取数据的恶意软件使用Intel AMT工具绕过防火墙。



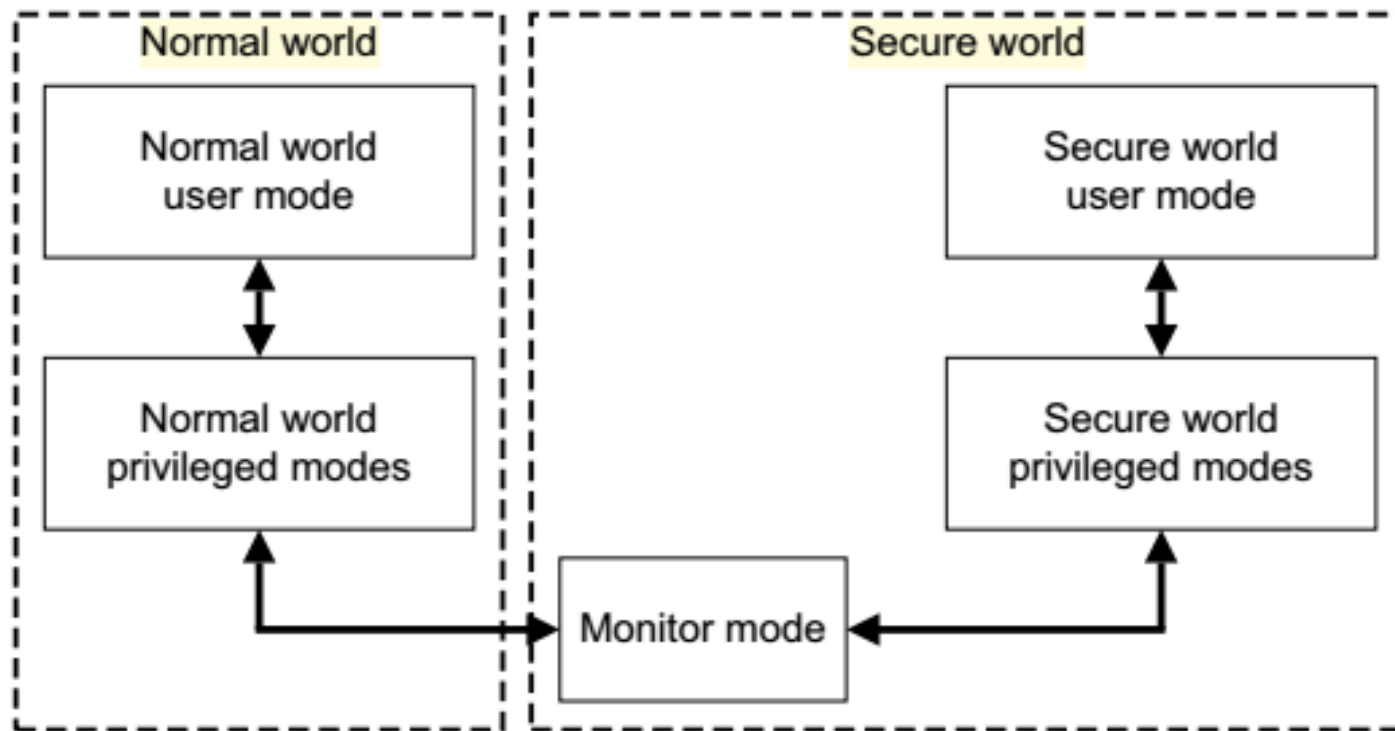
Ring 3: 应用程序
Ring 0: 操作系统内核
Ring -1: Hypervisor (虚拟化)
Ring -2: SMM(系统管理模式)
Ring -3: ME(管理引擎)

这些安全隐患的根本原因是负环设计的目的是增加新的功能, 而非为了安全。

○ 隔离执行环境

○ ARM Trustzone

- ARM提供的一种硬件安全架构，其目标是建立一个安全的程序执行环境，以保证程序和数据的机密性和完整性。
- 硬件和软件资源划分为Secure World和Normal World两个世界，通过一个名为Monitor Mode的模式进行转换。



○隔离执行环境

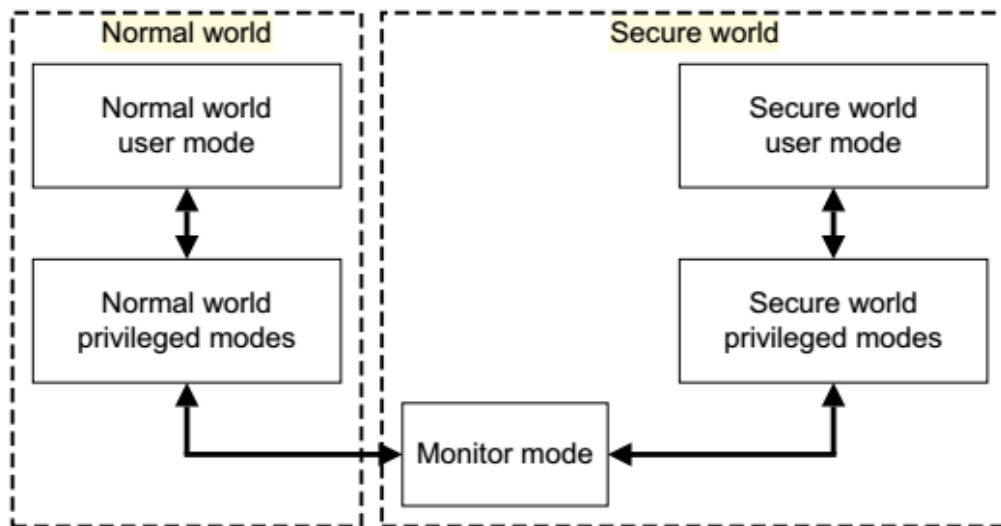
○ARM Trustzone

○TrustZone将每个物理核虚拟为两个逻辑核

- 非安全核（Non-secure Core），运行非安全世界的代码
- 安全核（Secure Core），运行安全世界的代码

○两个世界的切换

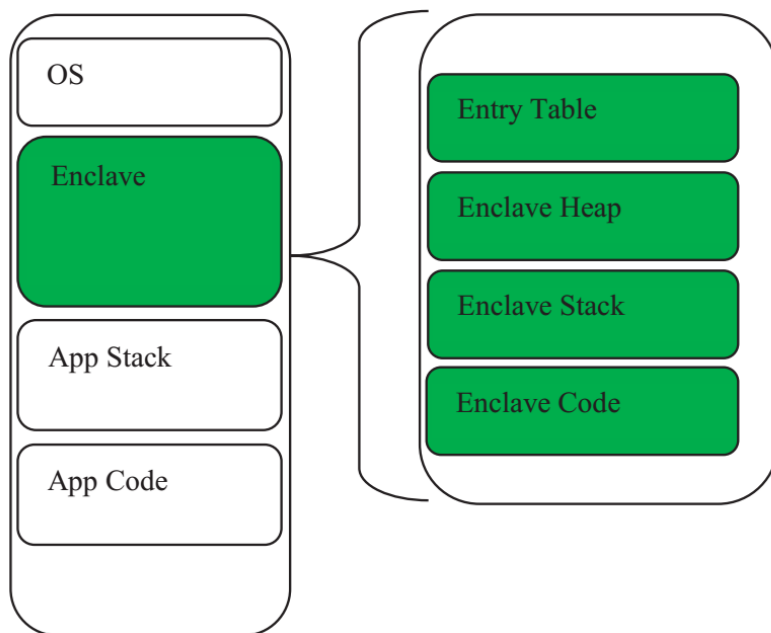
- 通过Monitor Mode进行切换
- 以基于时间片的方式运行
- 实现了在同一CPU上两个操作系统间的切换



○ 隔离执行环境

○ Intel SGX (Software Guard Extension)

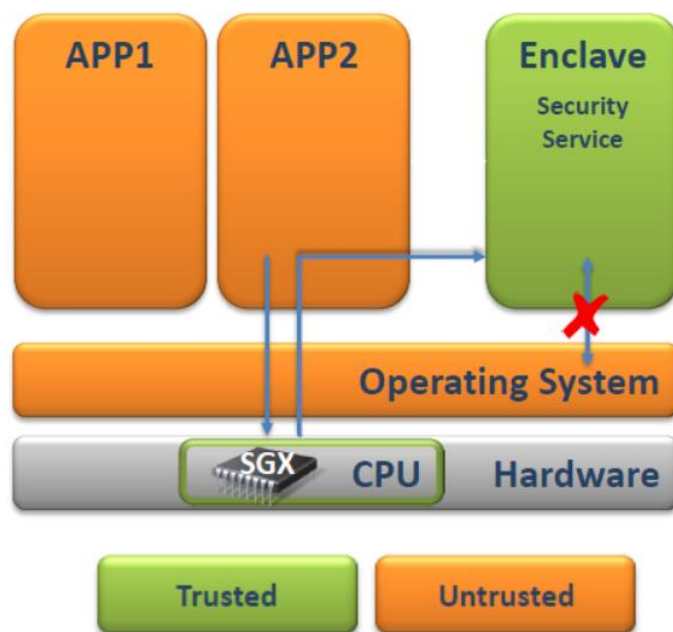
- Intel提出的一种指令集扩展，用于构造enclave（硬件容器）其目标是将合法软件封闭起来，隔离外部不可信软件。
- Enclave是从内存中划分出来的一块独立空间，一个SGX平台上可以同时运行多个Enclave，Enclave间彼此不干扰。
- Enclave运行在用户态。



○ 隔离执行环境

○ Intel SGX (Software Guard Extension)

- enclave 的访问进行限制
- 保护内存地址的映射
- 透明内存加密
- 所有数据以加密形式存在内存中



○ 隔离执行环境

○ 虚拟机隔离

- 可以在虚拟机上运行恶意程序
- 利用虚拟机管理器VMM监控恶意程序
- 问题
 - 虚拟机本身可能存在被利用的漏洞
 - 恶意程序可以通过控制虚拟机管理器VMM，控制整个系统

○ 沙箱隔离

- 位于操作系统之上，在用户态中提供隔离环境
- 为不可信/不安全的执行提供资源隔离，监控恶意程序的行为
- 问题
 - 沙箱和被隔离程序仍然共享大量资源
 - 沙箱本身也可能存在漏洞

○加解密

○加解密是非常常用的一种基础运算，常用于各种安全场景

- 计算要求高，支持多种算法
- 实时性较高，对应用无感
- 自身安全性高，防止密钥泄漏

○处理器增加专门的加解密指令、运算模块、随机数发生器等

- 提高加解密运算的速度
- 运算在处理器内部完成，避免了数据泄露
- 处理器内置的随机数发生器能够替代软件的伪随机数发生器，随机性更高

○加解密

○一种常见的加解密指令集

- AES指令集：AES-NI

- 是一个x86指令集架构的扩展

- 由Intel在2008年3月提出

- 改进应用程序使用AES执行加解密的速度，支持多种算法

○物理不可克隆函数PUF(Physical Unclonable Function)

- 利用集成电路制造过程中的工艺偏差，相同的电路结构产生不同的偏差，具有唯一性，解决了密钥随机性的问题。且数据无需保存，使用时再对芯片做密钥提取即可，解决了密钥安全性的问题。

内容概要

- 安全及其属性
- 攻击与漏洞
- 漏洞分类及产生的根源
- 安全机制分类
- 体系结构与安全机制关系
- 计算机体系结构安全课程内容与意义

○体系结构安全目的

- 保障软硬件结构本身安全
- 为安全机制提供结构支撑

○例子

- Meltdown、Spectre攻击、影子栈防御（保障计算机体系结构本身安全）
- Sgx、Trustzone、真随机数发生器、可信根、硬件加解密（为安全机制提供结构支撑）

○保障软硬件结构本身安全

- 由于计算机结构的设计存在缺陷，对安全考虑不足，信息系统不可避免的存在安全漏洞，导致攻击者利用漏洞对信息系统进行攻击
- 因此，必须要对攻击进行分析，找到漏洞及其产生的根源，从软硬件结构上消除安全隐患

○主要研究方法：

- 攻击入手
- 发现漏洞
- 分析机理
- 设计或改进结构

○为安全机制提供结构支撑

- 体系结构是信息系统的基础，在体系结构之上还有软件，而软件面临着严重的安全威胁
- 为了保障软件的安全，计算机体系结构需要提供结构支撑，从而更有效的实现安全机制，提升安全机制的效率，增强安全机制的安全性

○主要研究方法：

- 安全机制的种类（五种）
- 体系结构对权限管理的支持
- 体系结构对信任建立的支持
- 体系结构对隔离保护的支持
- 体系结构对密码计算的支持
- 体系结构对主动防御的支持

○作业（二选一）

一、对漏洞分类进行调研

对漏洞分类进行调研，分析各种漏洞分类方法的原理及优缺点，将这些漏洞分类方法与本课程提出的漏洞分类进行对比，按照本课程提出的漏洞分类对目前已有的漏洞类型进行梳理。

二、对体系结构如何支持安全机制进行调研

关于体系结构支持信任建立和主动防御，课上未展开，调研体系结构（硬件）支持上述两类安全机制的技术，并说明结构是如何更好支持这两类技术的。

Q&A