# Chap6

## 硬件攻击技术

硬件攻击技术是网络安全中的重要领域，它涉及利用硬件设计和实现中的漏洞、硬件木马、硬件故障以及侧信道攻击等手段来攻击系统。以下是硬件攻击技术的相关内容：

## 漏洞攻击 P15

漏洞攻击利用硬件设计中存在的缺陷来攻击系统。硬件漏洞主要指硬件设计和生产中存在的缺陷，这些缺陷通常由于设计说明不完善、设计人员编码不规范、硬件性能优化、测试与验证覆盖率不够高等因素引起。

- **指令集漏洞**：不同系统架构对相同指令集的实现方式有所不同，指令集漏洞是由于处理器开发人员在指令集的硬件实现过程中，对一些特殊情况考虑不周导致的。各大处理器厂商会定期发布产品的勘误表，列出指令集漏洞并提供修复措施。
- **浮点除（FDIV）的实现漏洞**：在某些处理器中，浮点除法的实现存在错误，导致计算结果不准确。例如，因特尔奔腾处理器中的FDIV错误就是一个经典的实现漏洞。
- **性能优化缺陷**：Meltdown和Spectre源于乱序执行和分支预测（提高CPU性能）

## 硬件木马(P21)

硬件木马是指被第三方故意植入或设计者有意留下的特殊模块和电路，它们潜伏在正常的硬件中，并在特定条件下被触发，实施恶意功能，如窃取敏感信息或拒绝服务攻击。

### 不可信的产业链（页码：22）

硬件木马通常由不可信的合作厂商引入，特别是对于我国的高端服务器、操作系统等核心软硬件，大部分都是进口的。这导致了木马和后门问题的严重性。

### 木马分类（页码：23）

硬件木马可以根据以下特征进行分类：

- **插入阶段**：指木马的植入阶段，可以是在制造过程中的偶然插入（由制造过程的缺陷引起），或者是蓄意插入（由攻击者植入）。
- **抽象级别**：指木马在电路中的位置，可以是在较高级别的模块中（如处理器、芯片组），也可以是在较低级别的模块中（如外围接口芯片）。
- **激活机制**：指木马被触发的机制，可以是在已知的功能状态下触发，也可以是在特定状态的组合或序列下触发。
- **影响**：指木马对系统的影响程度，可以是功能参数错误，也可以是泄露敏感信息等。
- **位置**：指木马在电路中的具体位置，可以是在信号路径上，也可以是在控制路径上。

### 木马与故障的区别（页码：24）

硬件木马与电路故障是不同的概念：

- **电路故障**是指在电路的生产加工过程中由于工艺或流程不完美导致的电路缺陷。
- **硬件木马**是指人为蓄意加入的恶意电路模块，其功能超出了指定功能之外。与电路故障不同，硬件木马通常较难检测到。

| | 电路故障 | 硬件木马 |
|---|---|---|
| 激活 | 通常在已知的功能状态下 | 任意电路中间节点特定状态的组合或者序列（可以为数字或者模拟信号） |
| 植入 | 偶然（制造过程的某些缺陷） | 蓄意（由产业链的攻击者植入） |
| 作用 | 电路功能或参数错误 | 电路功能参数错误、泄漏信息等 |

### 软件木马与硬件木马对比（页码：25）

软件木马是一种带有恶意代码的软件，与硬件木马攻击目标相似，软件木马通常可以在应用领域内解决，而硬件木马一旦植入就很难移除。

| | 软件木马 | 硬件木马 |
|---|---|---|
| 激活 | 存在恶意代码的软件，在代码运行过程中激活 | 存在集成硬件中，在硬件运行过程中激活 |
| 感染 | 用户交互过程中传播 | 由集成硬件设计流程中不可信的参与方植入 |
| 补救措施 | 通过软件更新移除 | 流片一旦加工成型就不能移除 |

### 硬件故障（页码：26）

硬件故障是常见的硬件问题，攻击者可以利用合适的硬件故障来窃取敏感信息或实施硬件攻击。这利用了硬件的异常行为来达到攻击的目的。

## 故障注入攻击（P27）

故障注入技术最初被用来验证系统的可靠性和可用性，并在电子器件的整个制造过程中得到广泛使用，主要做法是试图通过受控实验改变正在运行的系统的工作环境，从而引入故障，并根据系统的工作状态评价系统的可靠性和可用性。最初，故障注入技术被用于验证系统的可靠性和可用性，并在电子器件制造过程中广泛使用。然而，这种技术也可以被恶意使用来攻击系统的安全性。

### 故障注入分类（P28）

故障注入可以根据实施方法的不同进行分类。其中，基于硬件的故障注入攻击是一种较为常见的方法。它的实现相对简单，成功率较高，并且故障注入过程也比较可控，因此被广泛研究和应用。

### 光故障注入

光故障注入利用光电效应来激发半导体硅片产生大量的自由电子和空穴，从而造成晶体管的导通，引入故障。例如，研究人员发现通过将激光调至精确频率并对准智能语音设备，可以像用户声音一样激活语音助理并进行交互，从而解锁汽车、打开车库门等。这种方法的作用距离甚至可以达到一百多米。

### 电磁故障注入

随着CMOS工艺特征尺寸的不断缩小，电子器件变得更加敏感，对电磁攻击更为脆弱。电磁故障注入利用产生的电磁场对目标设备的内部产生干扰，引入故障。例如，电磁故障注入可以通过产生瞬态感应电压毛刺来改变芯片内部晶体管的状态，从而导致芯片的不可预期行为。

### 时序电路与故障注入

时序电路是一种按照时钟信号进行同步操作的电路。故障注入可以破坏时序电路中的约束条件，导致电路的输出产生错误。例如，提高时钟频率或降低电压都可能破坏时序电路的约束条件，导致电路功能不稳定或输出保留上次的值。

### 影响因素

故障注入攻击的效果受多个因素的影响，包括攻击时刻、攻击强度、作用时间和空间位置等。不同的因素会对故障注入攻击的有效性产生影响。

有效的故障注入攻击可以通过精确控制这些因素来引入故障，并对系统的安全性产生重大影响。因此，对于系统的设计和开发人员来说，理解和防范故障注入攻击至关重要。

## 侧信道攻击与旁路侧信道（页码：39）

**侧信道攻击**是利用硬件系统的旁路信息实施攻击的一种方式。它通过从密码系统的物理实现中获取信息，如时间信息、功耗消耗、缓存使用等。侧信道攻击的设备成本低，但攻击效果显著，严重威胁了密码设备的安全性。

**硬件隔离并不能完全保证安全，因为旁路侧信道攻击可以利用硬件系统的旁路信息来获取敏感数据，绕过硬件隔离的保护机制。（页码：40）**

## 物理侧信道攻击（页码：42）

物理侧信道攻击是一种侧信道攻击方法，基于从密码系统的物理实现中获取的信息，如时间侧信道、功耗侧信道、电磁侧信道和声波侧信道。

- **时间侧信道**：通过系统在不同行为流程下所表现出的执行时间差异，推断有用的信息，以达成或辅助攻击。
- **功耗侧信道**：通过分析设备随时间的功耗曲线，推测CPU执行过程中的某些信息。

## 时间侧信道攻击（页码：43-44）

时间侧信道攻击利用系统在不同行为流程下所表现出的执行时间差异来推断有用的信息。例如，可以通过观察登录验证的时间差异来判断用户名是否存在，进而进行针对性的攻击。

攻击者可以提交多个请求，统计从发出请求到收到服务器回执所耗费的时间，并通过测量多次取平均值或中值等方法消除网络延迟引入的时间抖动。通过统计时间较长的请求，攻击者可以推断出用户名是否存在，从而发动针对性的攻击。

## 功耗侧信道攻击（页码：45-47）

功耗侧信道攻击利用设备在不同指令执行时的功耗特征来推测CPU执行过程中的某些信息。不同指令触发的半导体数量、访问内存和缓存等因素会导致不同指令执行时产生不同的功耗特征。

功耗侧信道攻击可以分为以下三种：

- **简单功耗分析（Simple Power Analysis, SPA）**：通过直接分析设备随时间的功耗曲线，推断CPU执行的顺序或指令。
- **差分功耗分析（Differential Power Analysis, DPA）**：通过比较不同输入对应的功耗曲线之间的差异，推断出密钥或敏感数据。
- **相关功耗分析（Correlation Power Analysis, CPA）**：通过建立功耗曲线与已知输入的相关性模型，推断出密钥或敏感数据。

这些方法可以从设备的功耗变化中获取信息，从而对密码系统进行攻击（密码爆破），窃取敏感数据。

## 电磁侧信道攻击（页码：50）

电磁侧信道攻击利用与数据相关的电流在处理器中的变化导致的磁场变化，通过分析磁场来获取数据。电磁分析类似于功耗分析，都是一种非接触式的攻击方法。

密码芯片在工作过程中会产生不可避免的电磁辐射，辐射的信息和芯片内部的数据存在一定的相关性。攻击者可以分析这些电磁辐射，从中获取敏感数据。

## 微架构侧信道攻击（页码：51）

微架构侧信道攻击是基于现代处理器的优化技术（如乱序执行和推测机制）的副作用而产生的漏洞。随着处理器性能的提升，与之相关的漏洞也越来越多。Meltdown和Spectre等侧信道攻击就是这种类型的攻击。

这些攻击利用了异常延迟处理和推测错误导致的微架构状态的变化，在架构层级上未显示，但在处理器的微架构状态中留下痕迹。通过隐蔽信道，攻击者可以传输微架构状态的变化到架构层级，从而恢复出秘密数据。

## Cache侧信道攻击（页码：52-59）

Cache侧信道攻击利用多核之间的缓存数据共享以及缓存命中和失效所对应的响应时间差异来推测缓存中的信息，从而获取隐私数据。

Cache是存储器层次结构中的一级缓存，包含L1、L2和L3等级，用于存放从主存调入的指令和数据块。Cache具有地址转换部件和替换部件，通过建立目录表来实现主存地址到缓存地址的转换，并在缓存已满时按一定策略进行数据块替换。

Cache侧信道攻击主要包括以下四种方法：

- **Prime-Probe**：通过填充特定的缓存组，并测量读取时间来推测缓存中的信息。
- **Flush-Reload**：通过驱逐缓存中的数据，并测量重新加载的时间来推测缓存中的信息。
- **Evict-Reload**：与Flush-Reload类似，通过驱逐共享内存中的数据块并重新加载来推测缓存中的信息。
- **Flush-Flush**：通过执行flush指令清空缓存中的原始数据，并测量刷新时间来判断原始数据是否被缓存。

这些攻击方法利用了缓存的工作原理和多核之间的共享特性，通过测量访问时间的差异来推测缓存中的信息，从而获取隐私数据。

# 硬件防护技术

## 木马检测技术

木马检测是一种防护措施，用于检测和防御恶意软件中的木马程序。木马检测方法大致可以分为破坏性和非破坏性两类。（P62）

### 检测挑战

- 选择合适的木马模型。
- 生成测试向量来激活木马或增加侧信道测量中的木马敏感度。
- 消除/校准测试中的环境噪声和测量噪声。

### 集成电路生命周期（页码：63）

在集成电路的生命周期内的各个阶段都可能存在不可信的组件/人员。一般来说，芯片设计阶段是可信的，因此可以获得标准的设计和测试向量来进行木马检测。制造阶段通常不可信，其他阶段则可能同时存在可信和不可信的情况。

## 方法分类（页码：64）

### 破坏性检测

- 化学去封
- 电路扫描
- 逆向分析
- 芯片重构
- 模板匹配

破坏性检测方法费时耗力，不能逐个检测，实用性较低。

### 非破坏性检测

- 实时监测（在线监测）
- 芯片测试
- 测试向量
- 特征检测
- 结果分析

非破坏性检测方法可以逐个测试，但测试向量不一定全面，检测分析可能不完善。

## 逻辑测试和侧信道分析（页码：65）

木马检测可以使用逻辑测试法和侧信道分析法。

**逻辑测试法**侧重于生成并使用**测试向量**来尝试激活可能的木马电路，并观察对于主输出端有效荷载的影响。

侧信道分析法基于在芯片中植入任何恶意电路都会影响某些**旁路信道**的参数值，如漏电流、静态电源电流、动态功耗轨迹、路径延迟特性和电磁辐射。

| | 逻辑测试 | 侧信道分析 |
|---|---|---|
| 优点 | 对小型木马有效 | 对大型木马有效 |
| 缺点 | 测试生成复杂，大型木马检测具有挑战性 | 对过程噪声脆弱，小型木马检测具有挑战性 |

## 隔离技术（P66）

隔离技术是一种访问控制手段，用于限制用户访问非授权的计算资源。通过硬件隔离技术，可以实现计算资源的共享、安全计算环境和不安全计算环境之间的隔离。以下是几种常见的隔离技术：

### 存储器隔离（页码：67）

存储器隔离是现代处理器通常具备的一种特性。它通过存储器保护技术、EPT硬件虚拟化技术和存储加密技术来实现。系统使用IOMMU（IO Memory Management Unit）来限制设备对存储器的访问。IOMMU通过重定位寄存器和界地址寄存器来实现，其中重定位寄存器包含物理地址，界寄存器包含逻辑地址。

### EPT硬件虚拟化技术（页码：68-69）

EPT（Extended Page Tables）是Intel针对软件虚拟化性能问题推出的硬件辅助虚拟化技术。它提供了一种地址转换机制，将客户机虚拟地址（GVA）转换为客户机物理地址（GPA），然后通过EPT将客户机物理地址转换为宿主机物理地址（HPA）。这两次转换由CPU硬件自动完成，转换效率非常高。

### [illegible]（页码：69-71）

### ARM TrustZone（页码：60-71）

ARM TrustZone将CPU内核隔离成安全和普通两个区域。单个处理器内核包含了安全处理器核和普通处理器核，可以以时间片的方式从安全区域和普通区域执行代码。安全核只能访问安全世界的系统资源，而非安全核只能访问普通世界的系统资源。切换安全核和非安全核之间通过SMC（Secure Monitor Call）指令或硬件异常机制的一个子集来实现。

### Intel SGX（页码：71-72）

Intel SGX（Software Guard Extensions）是一组CPU指令，可让应用程序创建安全区域（Enclave），这些区域是应用程序地址空间中的受保护区域。即使存在特权恶意软件，SGX可以提供机密性和完整性。SGX可以减少应用程序的攻击面，提供了机密性和完整性，具有低学习曲线和远程认证的特点。

通过隔离技术，可以限制对计算资源的访问，提高系统的安全性和隐私保护能力。这些技术在硬件和软件层面上实现了资源隔离和保护，有助于防止恶意行为和攻击对系统的影响。

## 密码技术（页码：75）

现代密码技术可以分为对称密钥和非对称密钥两种体系。对称密钥常用于批量数据的加解密，而非对称密钥更多用于密钥交换。此外，还有一种新型的密钥生成电路称为**物理不可克隆函数（Physical Unclonable Function, PUF）**电路，它可以保证密钥的唯一性和不可克隆性。

**物理不可克隆函数（PUF）** 利用在半导体生产过程中自然发生的深亚微米变化，赋予每个晶体管些许随机的电特性。PUF具有唯一性、隐匿性、稳定性和随机性等特点。根据实现方式的不同，PUF可以分为非电子PUF、模拟电路PUF和数字电路PUF。PUF的应用领域包括身份认证、密钥生成和创建信任根。（页码：75-78）

## 旁路信息隐藏与掩码技术（页码：79-81）

攻击者利用旁路信道的输出来获取足够的信息，以确定和芯片操作相关的敏感数据。为了降低攻击难度，**可以通过增大噪声或减小信号来降低信噪比**。旁路信息隐藏的一种措施是**掩码技术**，它从功能模块的中间节点入手，移除输入数据与旁路信道的相关性。掩码技术可以基于门或模块实现。

模块划分和物理防御 P81

在设计防护措施时，可以将芯片操作中的明文操作区和密文操作区分开，并对芯片不同区域进行物理隔离。除了物理隔离，还需要对片内共享的基础设施进行分离，包括供电基础设施、时钟基础设施和测试基础设施。模块划分以及物理安全是降低攻击者利用旁路信道进行攻击的关键。（页码：80-81）

另外，一些已知的旁路信道攻击包括**Meltdown**、**Spectre**和**VoltJockey**，它们利用了处理器中的旁路信息泄露敏感数据的漏洞。

## 典型漏洞分析

### [illegible]（页码：[illegible]）

# MOLES（页码：83）

MOLES是一种利用木马通过侧信道泄漏芯片内部信息（如密钥）的攻击方式，然后利用差分能量分析技术提取密钥。

# Meltdown漏洞（页码：84）

Meltdown是一种利用乱序执行漏洞的攻击，它可以允许攻击者在任意地址读取数据，包括内核内存。

## 乱序执行（页码：85）

乱序执行是一种现代处理器的执行机制，它允许处理器在指令执行的顺序上进行重排序，以提高执行效率。乱序执行过程包括获取指令、解码后存放到执行缓冲区（保留站）、乱序执行指令并将结果保存在一个结果序列中，然后进行重排和安全检查，并将结果提交到寄存器。

## 攻击过程（页码：86）

Meltdown攻击利用乱序执行漏洞来泄露内核内存。攻击者在用户级特权下执行以下指令：

1. 将目标内核地址加载到寄存器rcx和rbx。
2. 对寄存器rax进行清零。
3. 从目标内核地址中读取一个字节并加载到寄存器al中（步骤1）。
4. 将寄存器rax左移12位（相当于乘以4096）。
5. 从rbx和rax的和地址中加载一个qword（8字节）。

攻击过程中的乱序执行使敏感数据泄露到缓存中，然后通过Cache侧信道进行提取。

## 攻击场景

Meltdown攻击的目标是读取内核数据。攻击者以用户级特权运行，并利用乱序执行漏洞泄露敏感数据到缓存中。通过观察缓存集的痕迹，攻击者可以提取出敏感数据。

这种攻击可以影响几乎所有的Intel处理器和ARM Cortex-A75处理器。

# Spectre漏洞（页码：87-89）

Spectre是一种利用分支预测错误的攻击方式。分支预测是一种处理器的执行技术，它通过在分支指令执行结束之前猜测哪一条分支将会被运行，以提高指令流水线的性能。

攻击过程中，攻击者训练CPU的分支预测单元，使其在运行代码时执行特定的预测。然后，攻击者进行分支预测越权访问敏感数据，并将其映射到缓存中。最后，通过缓存侧信道，攻击者可以通过观察缓存使用情况来窃取敏感数据。

Spectre漏洞的关键步骤类似于Meltdown漏洞。当CPU发现分支预测错误时，会丢弃分支执行的结果，但不会恢复CPU Cache的状态。利用这一点，攻击者可以突破进程间的访问限制，从而获取其他进程的数据。攻击者可以通过在代码中设置故意的分支预测错误来利用Spectre漏洞。

# Meltdown漏洞（页码：90-91）

## VoltJockey漏洞（页码：90-96）

VoltJockey是一种利用动态电源管理技术（Dynamic Voltage and Frequency Scaling, DVFS）的漏洞。DVFS是一种在满足用户对性能需求的前提下，根据处理器的负载状态动态改变电压和频率，以实现节能的技术。它被广泛应用于现代处理器中的低功耗技术。

攻击者利用VoltJockey漏洞，通过控制DVFS的参数，将处理器的电压设置得较低，导致特定操作中的AES加密函数发生电压故障。然后，利用缓存侧信道监视被攻击函数的缓存使用情况，并在故障注入点引入特定时间的电压故障。最后，通过差分故障分析技术提取AES的加密密钥。

在VoltJockey攻击中，攻击者需要进行一些准备工作，如分析公开的加密函数以找到合适的注入点，分析故障注入的合适电压与时间，并使用NOP指令精确控制时间。攻击者还需要将进程绑定到特定的处理器内核，并使用缓存侧信道监控被攻击函数的缓存使用情况。通过调整故障电压分析、确定临界电压、预备延迟等参数，攻击者可以成功地利用VoltJockey漏洞来窃取AES的加密密钥。

## COVID-19家庭检测套件的漏洞（页码：96）

COVID-19家庭检测套件包括棉签拭子、试剂和一个分析仪。用户使用棉签从鼻子或喉咙中收集粘液样本，与测试溶液混合并滴在检测卡上。用户通过手机应用程序发送检测请求，并在大约20分钟内收到检测结果。

然而，该应用存在漏洞，使得一些信息可能被匿名发送到云端，从而可能导致个人隐私泄露的风险。此外，定制PCB板上的LED灯珠和透镜也可能用于照亮测试条和读取结果线，分析仪通知移动应用程序的过程中也存在安全风险。