4 安全审计

2019年4月10日 23:36

1 概述

1.1 相关概念

- 安全审计
  - 对系统中与安全相关的活动进行记录、检查及审核
- 安全审计的目的
  - 检测、阻止非授权用户对系统的入侵；
  - 检测、显示授权用户的误操作。
- 审计事件
  - 信息系统审计用户操作的最基本单位。
  - 实例（操作系统中）：
    - 注册事件
    - 使用系统的事件
    - 利用隐蔽通道的事件
- 审计踪迹
  - 定义：关于操作系统、应用程序或用户活动的一组记录。
  - 在信息系统中，可按照所要监测的事件类型，设置若干个审计踪迹。

1.2 安全审计的作用

系统安全的最后防线，访问控制的必要补充。

- 追踪用户在系统中的活动
- 重建事件

- 监测潜在的入侵，提供入侵检测所需的原始数据
- 故障监测
- 定位安全问题
    - 帮助故障分析
    - 发现系统不足
- 与其它安全机制联动

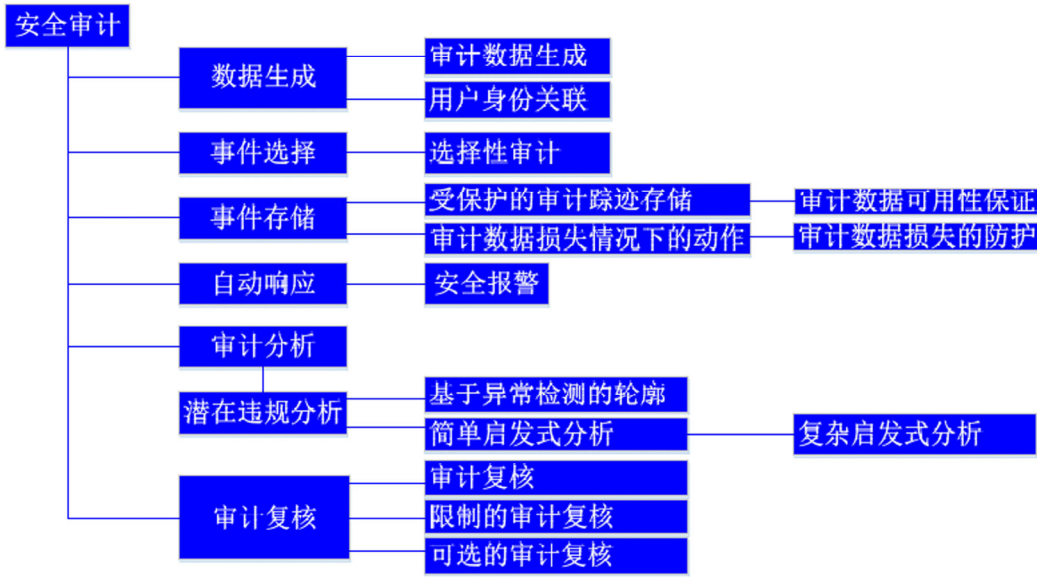
1.3 安全审计的实现要求

- 记录相关信息
- 重现安全事件
- 违规事件检测和分析
- 尽可能减少对生产系统的影响
- 保障自身安全
- 审计过程相对独立，与系统其它功能隔离。

2 安全审计系统模型

- 2.1 功能需求
- 2.2 X.816标准定义的审计系统模型
- 2.3 基于审计数据应用层次的系统模型

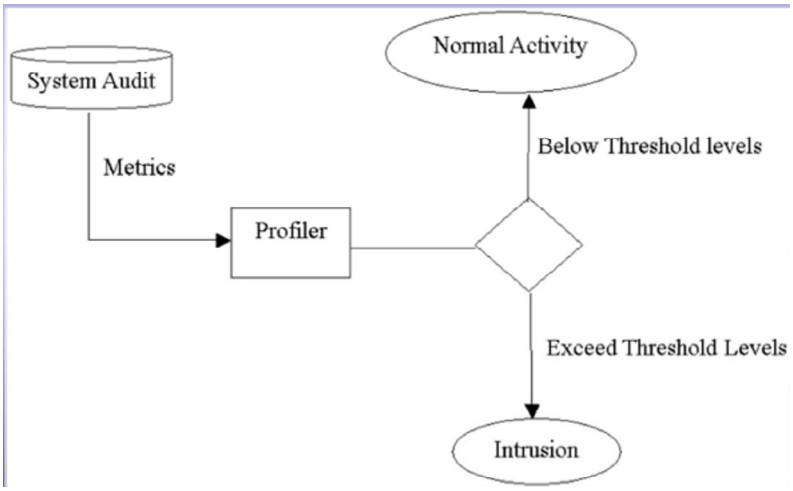
(不考)2.1 功能需求

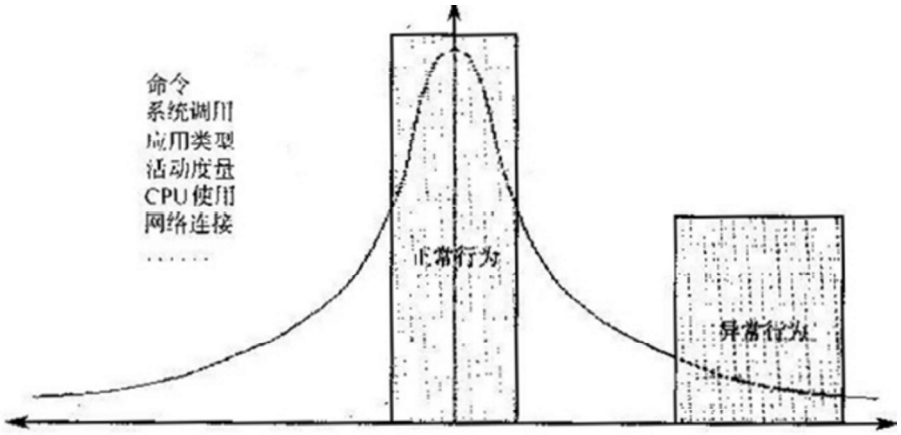
- 

入侵检测

- 入侵（Intrusion）：试图破坏计算机系统中资源的完整性、机密性及可用性的行为，以及违反安全策略的行为。
- 入侵检测（IntrusionDetection）：监视发生在计算机系统中的事件，分析隐藏的安全问题，监视、检测计算机系统中的目标活动并作出响应的功能。

异常检测

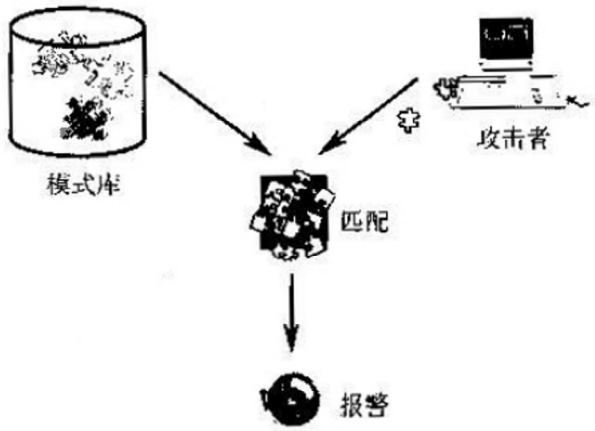
- 定义
  - 异常检测（AnomalyDetection）是根据非正常行为和使用计算机资源的非正常情况检测入侵行为。
- 异常检测原理
  - 异常检测试图用定量方式描述常规的或可接受的行为，以标记非常规的、潜在的入侵行为；
  - 异常检测需要建立目标系统及其用户的正常活动模型，然后基于这个模型对系统和用户的实际活动进行审计，以判定用户的行为是否对系统构成威胁。
- 轮廓（profile）
  - 轮廓是IDES（IntrusionDetectionExpertSystem）模型用来刻划主体对客体的行为，并使用随机变量(metrics)和统计模型来定量描述观测到的主体对客体的行为活动特征。
- 异常检测模型
  - 

- 异常入侵检测的主要前提是:将入侵活动作为异常活动的子集
- 异常检测的理想情况
  - 

- 异常检测的局限性
  - 理想的情形是，异常活动集同入侵性活动集是一样的。这样，识别所有的异常活动即识别了所有的入侵性活动，就不会造成错误的判断结果。
  - 然而，入侵性活动并不总是与异常活动相符合。
  - 入侵性活动常常是由单个活动组合起来执行，单个活动与异常性独立无关。

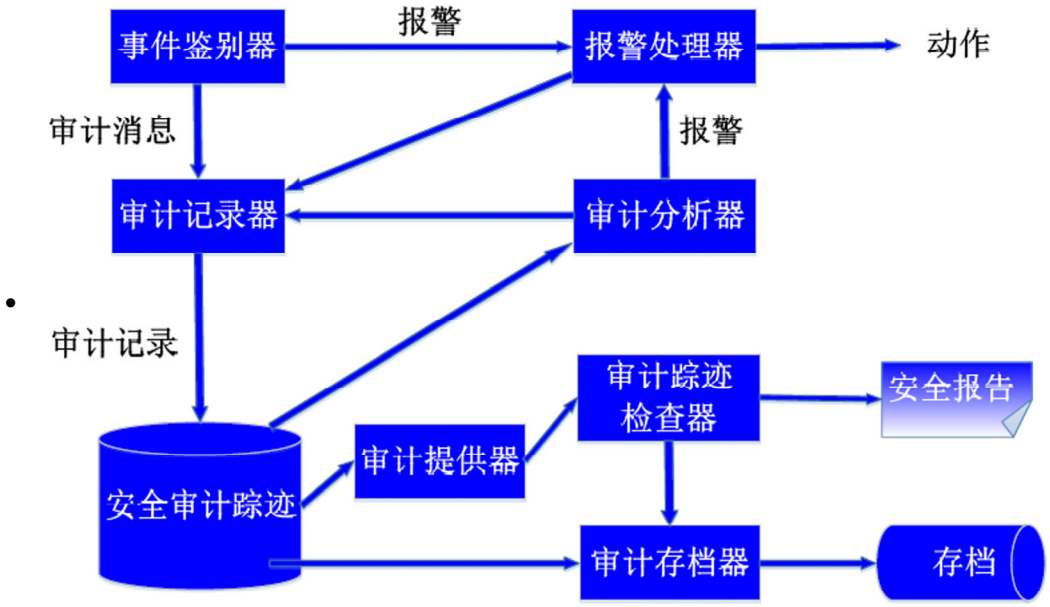
- 异常检测的结果：四种可能,每种情况的概率都不为零：
  - 入侵而非异常。活动具有入侵性却因为不是异常而导致不能检测到，这会造成漏检；
  - 非入侵而异常。活动不具有入侵性，而因为它是异常的，IDS报告入侵，这会造成虚报；
  - 非入侵非异常。活动不具有入侵性，IDS没有将活动报告为入侵，这属于正确的判断。
  - 入侵且异常。活动具有入侵性并因为活动异常，IDS将其报告为入侵，这属于正确的判断。

误用入侵检测

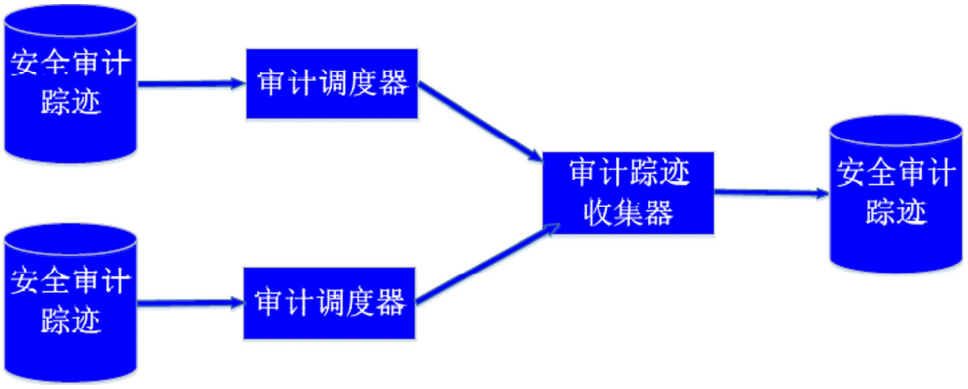
- 原理
  - 误用检测（MisuseDetection）基于已知的系统缺陷和已知的入侵模式进行入侵活动的检测。
  - 入侵者常常利用系统和应用软件中的弱点实施攻击，而这些弱点易编成某种模式，如果入侵者的攻击方式恰好匹配上检测系统中的模式库，则入侵者即被检测到。显然，误用入侵检测依赖于模式库，如果没有构造好模式库，则IDS就不能检测到入侵者。
- 与异常入侵检测的差异
  - 与异常入侵检测相反，误用入侵检测能够直接检测不利的或不可接受的行为，而异常入侵检测是发现同正常行为相违背的行为。
- 误用入侵检测模型
  - 

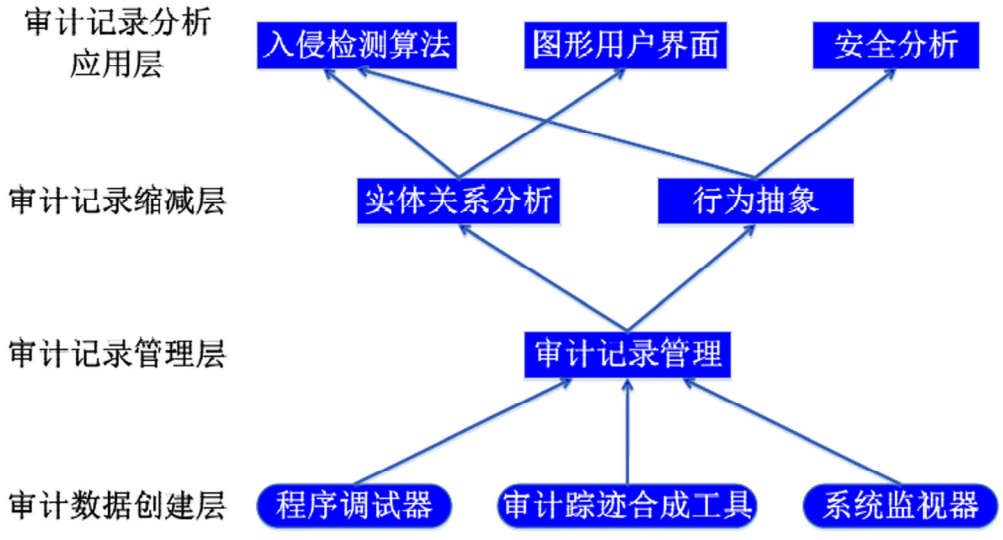
- 误用入侵检测的优点
  - 可以有针对性地建立高效的入侵检测系统，其精确性较高。
- 误用入侵检测的局限性
  - 误用入侵检测的主要假设是具有能够被精确地按某种方式编码的攻击。通过捕获攻击及重新整理，可确认入侵活动是基于同一弱点进行攻击的入侵方法的变种。从理论上讲，以某种编码能够有效地捕获独特的入侵不是都有可能。
  - 某些模式的估算具有固有的不准确性，这样造成IDS误报警和漏检。误用入侵检测主要的局限性是可以检测已知的弱点，对检测已知入侵的变种及未知的入侵可能用处不大。

2.2 X.816标准定义的审计系统模型



X.816标准定义的分布式审计踪迹模型





2.3 基于审计数据应用层次的系统模型

- 审计数据创建层
- 审计记录管理层
- 审计记录缩减层
- 审计记录分析应用层

(不考)3 安全审计系统的实现

1. 确定审计策略-明确需要审计的内容
    - 分析要审计的事件类型
        - 对安全信息系统，主要有注册事件、使用系统的事件和利用隐蔽通道的事件等。
    - 确定审计事件集
        - 明确对哪些事件进行审计，并非所有事件都要审计！
    - 审计踪迹
        - 审计踪迹维护系统活动的记录。
        - 系统级审计踪迹
        - 应用级审计踪迹
        - 用户级审计踪迹
        - 物理访问审计踪迹
2. 定义审计记录
    - When（时间）
    - Where（地点）
    - Who（主体）
    - Which（客体）
    - What（操作）
    - Result（结果）
3. 确定审计点
    - 研究在信息系统中的哪些点可以捕获到所有需要的审计事件。
        - 完整性
        - 效率
        - 安全性
4. 审计记录的存储
    - 存储形式
        - 以日志文件形式保存
        - 以数据库形式保存
    - 存储位置
        - 本地存储
        - 异地存储
5. 审计记录的安全性要求
    - 完整性要求
    - 机密性要求

    - 可用性要求
6. 审计信息的使用与管理
    - 审计记录的查阅
    - 审计记录的分析
    - 审计信息的管理
    - 审计信息的维护
7. 审计系统的构成
    - 审计发生组件（捕获审计事件）
    - 日志记录组件（记录审计事件）
    - 日志分析组件（分析审计事件）
    - 事件报告组件（报告审计事件）
    - 审计管理组件（系统管理）
8. 降低审计开销
    - 有选择地审计
    - 开辟审计缓冲区
    - 设法节约磁盘空间