如何构建一个入门级入侵检测系统 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/251/2021_2022__E5_A6_82_E4_BD_95_E6_9E_84_E5_c101_251019.htm 通常来说，一个企业或机构准备进军此领域时，往往选择从基于网络的IDS入手，因为网上有很多这方面的开放源代码和资料，实现起来比较容易，并且，基于网络的IDS适应能力强。有了简单网络IDS的开发经验，再向基于主机的IDS、分布式IDS、智能IDS等方面迈进的难度就小了很多。在此，笔者将以基于网络的IDS为例，介绍典型的IDS开发思路。 根据CIDF规范，我们从功能上将入侵检测系统划分为四个基本部分：数据采集子系统、数据分析子系统、控制台子系统、数据库管理子系统，如附图所示。 具体实现起来，一般都将数据采集子系统（又称探测器）和数据分析子系统在Linux或Unix平台上实现，我们称之为数据采集分析中心.将控制台子系统在Windows NT或2000上实现，数据库管理子系统基于Access或其他功能更强大的数据库，多跟控制台子系统结合在一起，我们称之为控制管理中心。本文以Linux和Windows NT平台为例介绍数据采集分析中心和控制管理中心的实现。 可以按照如下步骤构建一个基本的入侵检测系统。 第一步 获取libpcap和tcpdump 审计踪迹是IDS的数据来源，而数据采集机制是实现IDS的基础，否则，巧妇难为无米之炊，入侵检测就无从谈起。数据采集子系统位于IDS的最底层，其主要目的是从网络环境中获取事件，并向其他部分提供事件。目前比较流行的做法是：使用libpcap和tcpdump，将网卡置于“ 混杂 ”模式，捕获某个网段上所有的数据流。 libpcap是Unix或Linux从内核

捕获网络数据包的必备工具，它是独立于系统的API接口，为底层网络监控提供了一个可移植的框架，可用于网络统计收集、安全监控、网络调试等应用。 tcpdump是用于网络监控的工具，可能是Unix上最著名的sniffer了，它的实现基于libpcap接口，通过应用布尔表达式打印数据包首部，具体执行过滤转换、包获取和包显示等功能。tcpdump可以帮助我们描述系统的正常行为，并最终识别出那些不正常的行为，当然，它只是有益于收集关于某网段上的数据流（网络流类型、连接等）信息，至于分析网络活动是否正常，那是程序员和管理员所要做的工作。 libpcap和tcpdump在网上广为流传，开发者可以到相关网站下载。 第二步 构建并配置探测器，实现数据采集功能 1. 应根据自己网络的具体情况，选用合适的软件及硬件设备，如果你的网络数据流量很小，用一般的PC机安装Linux即可，如果所监控的网络流量非常大，则需要用一台性能较高的机器。 2. 在Linux服务器上开出一个日志分区，用于采集数据的存储。 3. 创建libpcap库。从网上下载的通常都是libpcap.tar.z的压缩包，所以，应先将其解压缩、解包，然后执行配置脚本，创建适合于自己系统环境的Makefile，再用make命令创建libpcap库。libpcap安装完毕之后，将生成一个libpcap库、三个include文件和一个man页面（即用户手册）。 4. 创建tcpdump。与创建libpcap的过程一样，先将压缩包解压缩、解包到与libpcap相同的父目录下，然后配置、安装tcpdump。 如果配置、创建、安装等操作一切正常的话，到这里，系统已经能够收集到网络数据流了。至于如何使用libpcap和tcpdump，还需要参考相关的用户手册。 第三步 建立数据分析模块 网上有一些开放源代码的数据分析

软件包，这给我们构建数据分析模块提供了一定的便利条件，但这些“免费的午餐”一般都有很大的局限性，要开发一个真正功能强大、实用的IDS，通常都需要开发者自己动手动脑设计数据分析模块，而这往往也是整个IDS的工作重点。数据分析模块相当于IDS的大脑，它必须具备高度的“智慧”和“判断能力”。所以，在设计此模块之前，开发者需要对各种网络协议、系统漏洞、攻击手法、可疑行为等有一个很清晰、深入的研究，然后制订相应的安全规则库和安全策略，再分别建立滥用检测模型和异常检测模型，让机器模拟自己的分析过程，识别确知特征的攻击和异常行为，最后将分析结果形成报警消息，发送给控制管理中心。