IDS及其Linux下的实现(2) 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_IDS_E5_8F_8A_E5_85_B6L_c101_251034.htm 5、Linux下的实现 5.1 系统框架 HereLine是一个在Linux下运行的基于标识检测的网络IDS。从逻辑上分为数据采集、数据分析和结果显示三部分，符合CIDF的规范。 从实现结构上看，HereLine分成三个应用程序，它们分别是： 1、数据收集及分析程序(watcher) ； 2、告警信息收集程序(listener)； 3、告警信息显示程序(console) 。watcher是数据采集和数据分析的合体；listener接收watcher发出的告警信息，并将接到的信息存储为日志；console为管理员提供了更友好的观察日志的图形界面。 同大多数商业IDS一样，HereLine采用了分布式的结构。运行HereLine建议采用两台PC机，一台运行watcher,另一台运行listener和console. 与其他同类程序相比，HereLine的优点主要有： 1、提供了完整的框架，可以灵活的应用于各种环境并扩充； 2、采用数据分析与告警程序分离，便于在大规模的网络环境下集中管理； 3、已经实现了对分片的处理，解决了利用分片逃避检查的问题； 4、数据分析部分不完全依赖已有攻击程序，而是分析其核心特征以察觉其变种攻击；但同时依靠已有攻击程序的细节来评估判别的准确性。以上设计增加了告警的可靠性。 各部分的实现流程及重要问题说明 5.1.1 数据采集部分 watcher采用了Linux2.2内核中提供的PF_PACKET类型的socket（并未采用libpcap提供的API接口） ,直接从链路层获取数据帧。（直接采用操作系统提供的接口主要是考虑高效性，但为了将来的移植问题，以后仍然可能会改为使

用libpcap库提供的函数接口。）根据Linux的要求，建立这样的一个socket需要root权限,即uid=0。从packet socket读到的数据是链路层格式的数据，但经过处理（socket函数的第二个参量SOCK_DGRAM表示要去掉第二层的数据头，第三个参量ETH_P_IP表示只接收ipv4的数据包）后，缓冲区内的内容是个完整的IP包（未经任何其它处理）。分析工作交由数据分析程序去做。 数据采集部分还做了一项工作就是将网卡置于混杂模式，这样可以监听到整个网段的数据。 HereLine的这种实现，实际是通过socket将数据拷贝到应用层。这种结构比较灵活与安全（应用程序崩溃不会导致系统崩溃），但频繁的应用态与核心态的转换浪费了CPU。还有一种办法就是采用Linux中的模组（modular）的办法，将IDS作为内核的一部分。《Building Into The Linux Network Layer》提供了一个内核中的sniffer的框架。可以利用其结构实现嵌入内核的、更加高效的入侵检测系统。HereLine可以很容易的转移到这种方式，但考虑到其对操作系统稳定性的影响和调试的难度，目前未采用。 需要指出的是，watcher只是监听数据包，并不参与操作系统协议栈的处理。如果操作系统被攻击导致拒绝服务，watcher也将无法运行。因此，首先要保证起所运行的系统是安全的。有些商业入侵检测系统（如NFR）将数据采集部分放在专门的、经过改进的、高度安全的系统之上，以保证IDS这一系统中的重要程序正常工作。对于HereLine本身来说，建议采用增加了stackguard功能的gcc编译器，减少潜在的缓冲区溢出（buffer overflow）漏洞。 watcher将数据读到缓冲区之后，首先将其封装为sbuff结构，当数据在程序中传递时，均采用此结构。其定义为： struct sbuff { union{ struct tcphdr

*tcph. struct udphdr *udph. struct icmphdr *icmph. struct igmphdr *igmph. } h. union{ struct iphdr *iph. } nh. unsigned char *data. } . sbuff中定义了指向协议头的指针，子程序可根据这些指针快速的定位数据头位置。