基于网络侦听的数据库审计方法①

钱正麟[1], 高 航[1], 李曙强[2]

[1](南京航空航天大学 计算机科学与技术学院, 南京 210016)
[2](南京德讯科技股份公司, 南京 210012)

摘 要: 目前, 在我国广泛使用的数据库管理系统中存在着监管失效, 内部操作不透明, 安全保护手段滞后等问题. 而系统自带的数据库审计服务也存在着审计日志格式繁多难于维护, 长期使用审计服务后导致系统性能下降等缺陷. 因此, 本文提出一种基于网络侦听技术的数据库审计方法, 该方法通过监听分析数据来获取数据库操作指令, 不改变现有网络结构, 也不影响网络的性能, 能够有效的审计监管外部入侵和内部的违规操作, 为事后责任认定提供证据, 可以有效的克服传统审计技术的不足.

关键词: 数据库; 审计; 网络侦听;协议分析;日志策略

DBMS Audit Based on Network Listener

QIAN Zheng-Lin[1], GAO Hang[1], LI Shu-Qiang[2]

[1](College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China)
[2](Datcent Technologies Inc, Nanjing, 210016, China)

Abstract: At present, widely used DBMS in our country, which has generally exist some problems. Because of regulatory failure, opaque internal operations, the lag of security measures, We can not guarantee that our database is secure. Although DBMS has provided audit services that also have many problems, such as most DBMS has different audit log format and difficult to maintain, low capability when audit server used long-term. So this paper presents a database audit method based on network monitoring technology. We can use it to collect database operations command by analyzing the monitoring data .It works without changing the existing network structure and effecting network performance, and it can effectively record external invasion or internal irregularities to provide evidence to legal responsibility. This approach overcomes the shortcomings of traditional audit techniques.

Key words: database; audit; network intercept; protocol analyse; log policy

1 引言

数据库是任何企业和政府中最具战略性的资产, 近年来, 随着网络的蓬勃发展, 数据库的可访问性得到提升, 同时也导致了数据库信息的安全面临严重挑战. CISSPS 曾经在报告中指出数据库安全在所有安全问题里居于第一位. Forrester 公司更是报告防火墙/IDS 等对数据安全的保护有效率不到 10%, 90%以上的安全威胁需要数据库本身以及专业产品的的防护, 而数据库安全威胁 80%来自于内部用户的误操作和恶意操作. 而 Version 公司 2008 年的调查结果显示数据库破坏事件占所有破坏事件的 30%, 其中破坏的记录占被破坏记录总数的 75%. 09 年则报告高危险攻击占所有攻击的 17%[1].

造成以上安全问题的原因归结起来有以下几点. 首先, 在管理层面上, 数据库的管理过程中人员职责分配不明, 管理监控失效, 后门问题, 维护过程中的错误操作等等. 其次, 在技术层面上是因为数据库管理系统是一个庞大而复杂的系统, 安全漏洞层出不穷, 黑客攻击的手段越发丰富. 最后是网内风险, 内部人员受到贪欲或者报复欲的驱使, 在不受防火墙等网络

① 收稿时间:2013-09-07;收到修改稿时间:2013-09-25

安全设备影响的情况下, 从内部给数据库的安全带来风险.

因此, 目前流行的数据库管理系统为了满足用户对于数据库的安全需求, 都提供了监视和记录数据活动的功能. 不过, 这些功能也同样都存在着不少缺陷. 依赖于数据库日志文件的审计方法, 在开启数据库日志功能会影响数据库本身的性能, 数据库日志文件会也占据硬盘的有效空间. 日志文件记录了用户对数据库的修改、插入和删除等操作, 但没有记录对数据库数据的查询或浏览, 这就无法防止不法人员查看和偷窃数据库中重要的, 保密的数据. 更严重的是日志文件有被篡改的风险, 无法体现审计工作的有效性与公正性. 而利用数据库安全监视功能,当审计和监视的事件数量太多时, 会增加服务器和监视进程的开销, 并能导致跟踪文件或跟踪表变得很大, 尤其是在进行长时间监视的情况下, 会影响业务系统的执行性能和工作效率[2-4].

本文提出的这种审计方式可以很好地解决这些问题, 它从数据链路层中采集数据库访问信息并进行分析, 恢复和还原用户的数据库访问过程, 系统的审计分析结果能够有效帮助管理员对数据库安全策略进行分析, 提升数据库安全性, 为事后责任认定提供证据, 克服了传统审计技术的不足.

2　网络侦听技术的简介与优化

网络侦听是一种监视网络状态, 数据流量以及网络上信息传输的管理工具, 它可以将网络设定成监听模式, 截获网络上所传输的信息. 网络数据中大量关键信息和操作都处于数据链路层, 但是由于 Windows 系统对网络底层进行内核级封装, 不能够直接捕获数据链路层的数据包, 所以采用 Winpcap 提供的驱动接口, 可以在 Windows 平台下, 在数据链路层实现对网络数据的采集, 分析和侦听[5].

Winpcap 是基于 Win32 平台的数据包截获和网络分析的体系结构, 为 win32 应用程序提供访问网络底层的能力, 它主要由三部分组成: 内核级的网络数据包过滤器 NPF, 低级动态链接库 Packet．dll、高级动态链接库 Wpcap.dll.

在数据采集过程中, Wpcap.dll 中提供的采集函数实质是调用底层 Packet.dll 中的抓包函数, 从内核缓存区拷贝一组数据包到用户缓存区, 再把用户缓存区的多个数据包逐个提取出来, 依次送入相应函数进行处理. 在高速大流量网络环境下, 用户缓存区处理数据包速度跟不上内核缓存区从网卡复制数据的速度, 新的数据包会因为内核缓存区满而丢弃, 这严重影响了网络监听中关键数据包的收集和准确分析.

因此, 我们可以通过优化对缓存区数据块的处理方式, 改进抓包性能. 首先, 通过适当增加内核缓存区大小, 可以提高性能. 但是, 当内核缓存区超过 4MB 时, 对丢包率的影响就很小. 第二, 上文中的两次内存拷贝过程, 已经成为抓包性能的瓶颈, 我们可以把每次从内核缓存区中拷贝用户缓存区中的数据设置为 500KB(缺省为 16KB). 当这个值较大时, 内核就会等多个数据包到达时再开始把数据拷贝到用户缓存区, 因此减少了系统调用次数, 提高了侦听性能[6].

3　数据库审计系统的设计

3.1　模型及功能简介

数据库审计系统的基本思想是在 Winpcap 的基础上, 通过侦听数据库客户端与数据库服务器之间的信息, 然后对信息进行解析, 从而获得用户对数据库所有的操作细节, 进而形成审计日志, 为数据库管理者提供及时准确的数据库操作信息.

数据库审计系统采用三层架构, 由审计客户端、审计中心和显示管理终端构成(如图 1 所示). 审计中心包括存储中心和处理器. 审计客户端包括数据侦听器、协议解析器, 检测过滤器和输出器. 而最终由用户通过操作显示管理终端来对整个系统进行操作. 一个审计中心对应多个用于事件收集的客户端, 多个显示管理终端对应一个审计中心这样的架构便于扩展, 也

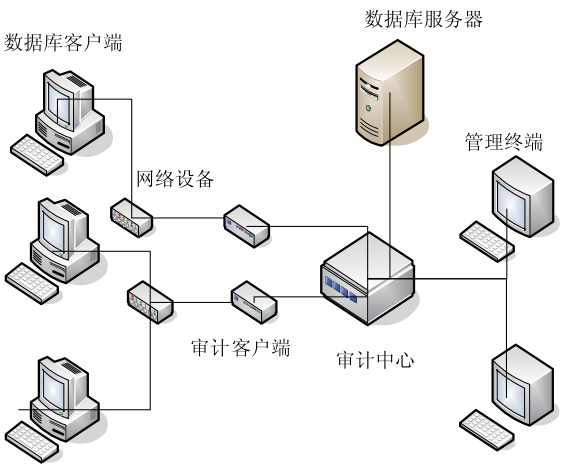

图 1　审计系统模型

方便多个审计人员同时对数据库进行审计.

显示/管理终端将完成对整个系统的统一管理, 包括各模块的统一管理和数据的收集、分析、数据存储以及提供友好的用户界面. 所有的审计结果发送到终端并显示到用户的图形界面上, 为用户提供直观、准确、可靠的信息.

审计中心的功能类似于代理, 它处理与各个审计客户端间之间的数据流: ①对客户端向管理终端的信息流进行中转, 它接收所有客户端发送的审计信息, 根据信息类别分别进行处理, 如果判断是报警信息, 则转发到管理 / 显示终端进行实时报警显示, 否则就存储在本地数据库或转发到远程数据库中; ②对管理终端向审计客户端下发的控制命令进行控制, 控制客户端的启动、停止、重启和向客户端下发过滤规则等, 同时也支持对客户端按组下发指令.

审计客户端工作在 Windows 工作机上, 它实时监听网络中流动的数据报文, 将符合规则要求的数据库操作上传到审计中心的数据库里, 作为分析数据存储起来, 以便数据库审计人员在事后进行查询与分析.

系统对客户端透明, 通过与数据库服务器相联的网络设备(比如 WindowsServer 服务器), 以服务器和客户端的通信数据, 作为审计的数据来源.

3.2 审计工作基本控制流程

根据数据库通信的基本过程, 审计系统相当于在数据库服务器与客户端之间, 对来往数据进行分析. 在数据库审计系统中要进行以下四个过程:

(1) 数据捕获, 负责收集网络中流经计算机机卡上的数据帧.

(2) 数据过滤, 按照不同的需求把不符合条件的数据忽略掉.

(3) 数据分析, 根据各种协议的数据结构进行解析, 分析处理数据, 让数据变得可读.

(4) 日志形成, 把符合条件的记录储存在日志文件中, 并对操作记录按照策略分类进行统计.

流程图见图 2

其中获取数据包的流程:

使用 pcap_findalldevs(*,*)获取本机所有物理网卡列表.

通过本机 IP 定位当前使用中的网卡, 并得到它的信息.

使用 pcap_open_live(*,*,*,*,*)获取使用中的网卡的句柄.

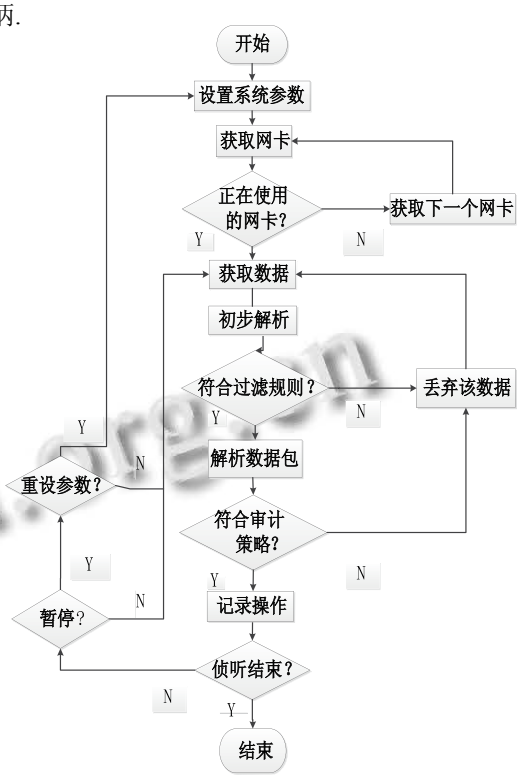

图 2 审计系统流程图

得到该句柄后, 可通过 pcap_dump_open(*,*.cap)来创建一个捕获的 CAP 包, 同时通过使用 pcap_compile 和 pcap_setfilter 来设置过滤条件. 以上完成后再使用 pcap_next_ex(*,*,*)得到原始数据包.

数据库通信在传输层基本都是使用 TCP 协议, 所以系统的数据分析在传输层层面与 TCP 协议一致, 不同的是更高层的应用层. 因此, 可以从以下三个方面考虑进行数据过滤:

协议的过滤, 只接受 TCP 协议的数据包.

IP 地址的过滤, 只接受访问目的地址为数据库服务器 IP 地址的数据包, 其他的忽略.

目的端口的过滤, 如果是 Oracle, 其连接数据库的端口为 1521, Sybase 则是 5000.

过滤的具体方法是使用一个包含高级布尔表达式的字符串并且产生一个能被过滤引擎集成到数据包驱动中的低级字节码, 再把过滤器与核心驱动抓包会话联系起来. 过滤条件一旦被使用, 相关的过滤器将被应用到所有的来自网络的数据包上, 接着把通过过滤的数据包移交给应用程序, 从而可以极大的减少数据分析工作.

3.3 数据协议分析

由于捕获到的数据是一串很长的字节流. 数据中的信息不仅仅包含单纯的数据库信息, 还包含传输层, 网络层的信息. 这些信息在网络传输过程中保持不变, 通过对这些信息的分析, 可以得到有关网络的信息. 而数据库通信在传输层基本都是使用 TCP 协议, 所以系统的数据分析在传输层层面与 TCP 协议一致, 可以根据 TCP/IP 协议的数据结构对数据进行解析以获得网络层和传输层的有关信息. 应用层的信息经过了传输层和网络层的封装, 需要针对每一种协议的具体数据结构对其附加的报头进行解析[7].

以下以 TDS5.0 即 Sybase 数据库的数据结构作分析. 首先是八个字节的 TDS 框架头的格式. 见表 1.

表 1 TDS 报文首部

1	2	3	4	5	6	7	8
类型	状态	信息长度		信道数量		包数量	状态

在然后的 TDS 数据中还蕴含这一个六字节的 Sybase 的数据包信息字段. 见表 2.

表 2 Sybase 协议信息

1	2	3	4	5	6
命令字段	当前数据记录的整个长度				当前状态

第一个字节是 TDS 命令控制字段, 如: 0x21 表示这是一个 TDS _LANGUNGE, 后面四个字节是一条操作记录的长度, 最大长度是 FFFFFFFF(4G)因此 Sybase 的一条记录最大的数据长度能达到 4G 之多. 第六个字节表示状态, 只有当值为 01 时有意义. 这些之后就是我们需要关注的部分[8].

3.4 审计日志保存

审计日志分包含了数据库操作日志和系统日志两个部分.

数据库操作日志的策略归类为权限审计, 语句审计, 用户审计三种[9].

权限审计主要是包括授予/回收权限, 创建/删除/修改用户角色等.

语句审计则是针对 SQL 语句的类型进行审计, 把所有操作信息不加筛选的记录进日志. 这种审计方法颗粒度较粗, 但是可以真实正确的反映所有的操作情况.

用户审计则是需要制定一个授权访问的 IP 列表, 然后根据用户的数据包的 IP 地址进行判断然后归类统计, 以获得给定用户的操作信息.

利用这些策略,对分析过滤后的数据包再做进一步的过滤, 一方面可以提高系统的审计效率, 更为重要的是. 当系统发现了对服务器数据库的高危操作时, 比如没有授权的 IP 访问, 通过用户界面给予报警提示.

系统日志主要是记录系统启动和退出的时间, 记录管理员 ID, 策略配置等一些系统本身属性.

为了方便使用者对审计的数据进行查询分析, 可以让所有需要记载的信息通过文件型数据库(.db)的文件格式进行保存. 而把访问的数据库类型, 访问时间, 源端口和地址, 目的端口和地址等简要信息, 保存进一个文本文件方便浏览. 由于网络中访问数据库很频繁, 存放数据的策略采用按记录数量存放, 比如 5000 条记录为一个数据库文件. 所有文件存放在审计中心里.

为了体现审计结果的公正性和全文性, 需要保存最原始的审计依据, 通过把流经某网卡上的所有数据写进一个.cap 文件中, 这个文件可以通过 wiresherk 或者 NetWorkMonitor 来进行查看, 用以体现审计工作的正确性.

由于网络中数据量很大, 不可能把每个数据包都保存在内存中, 这样给系统带来极大的负担, 因此我们把当前的数据保存在一个长度 6000 队列中, 每当队列满时就把队列释放.

4 测试分析

为了测试长时间运行两种审计服务对系统 CPU 的影响, 启动 Windows2003 服务器中安装的 Sybase 数据库, 再启动另一台同样的设备, 服务器中关闭所有与之无关的进程与服务. 第一台设备我们开启 Sybase 自带的审计服务, 第二台启用本文提出的审计服务, 部署在公司的网络中, 且设定所有涉及数据库操作的数据包都向两台服务器发送, 以保证数据一致性. 进行 48 小时的日常使用, 而后只保留六个客户端连接且切断其他连接, 每个客户端再像两个服务器发送一组同样数据库操作命令, 对比 CPU 使用情况.

明显看出图 3 切断连接以后 CPU 使用减少而后启用服务 CPU 使用情况也比较稳定. 图 4 可以看出 CPU 使用情况有几次几乎达到 95%[10]. 所以不难发现通过网络侦听方式审计数据库确实可以减少 CPU 开销, CPU 使用率也基本稳定.

存储空间上, 由于采用了灵活的存储方法并不需要预先分配日志数据库而占据一个固定空间. 本方案

中审计日志文件格式统一, 文件大小灵活变化, 且生成了用于记录的文本文件便于日志的快速定位与查找.

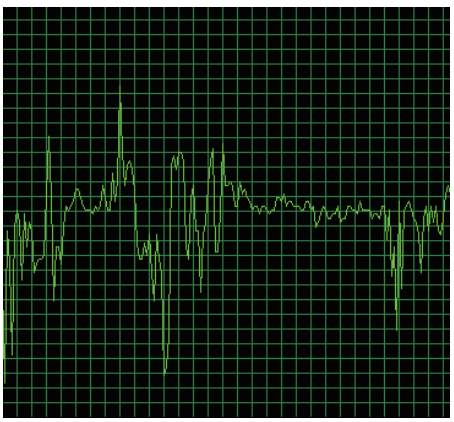

图 3 开启审计系统的 CPU 使用图

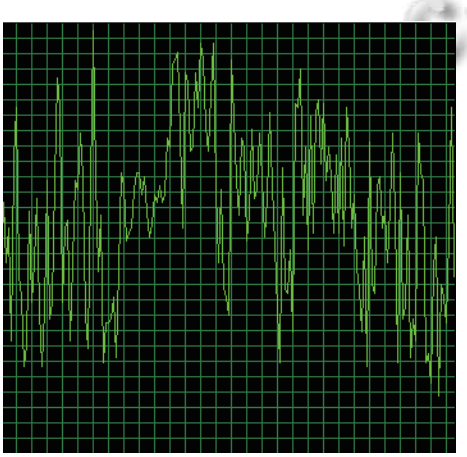

图 4 开启 Sybase 自带的审计服务 CPU 使用图

由于可以对数据库进行全语句审计, 就能对浏览数据库产生的信息进行记录. 而且, 如果日志文件被篡改, 也不会影响审计结果, 反而可以根据篡改者的操作记录而追查到篡改者的源 IP.

5 结语

本文在对当前数据库安全问题进行分析的同时, 提出一种基于网络侦听的数据库审计系统. 系统可以对用户所有的数据库操作进行监管, 实现了对数据库进行准确、高效的审计, 有效地解决了数据库系统使用中的安全问题, 为今后的进一步扩展应用提供了坚实的基础.

参考文献

1 Khanuja HK, Adane DS. Database security threats and challenges in database forensic: A survey. Proc. 2011 International Conference on Advancements in Information Technology. Singapore. IACSIT Press. 2011. 170–175.

2 Cong QS, Huang ZM, Hu JB. A collaborative computer auditing system under SOA-based conceptual model. Proc. 2011 4th IEEE International Conference on Computer Science and Information Technology. 2011, 10. 439–443.

3 Zhu YQ, Yu H, Li H, Zeng LM. Design of a new web database security model. IEEE Computer Societyeds. Second International Symposium on Electronic Commerce and Securit. ISECS 2009. Washington. Electronic Commerce and Security. 2010,1. 292–295.

4 Deng L, Wang LY. Simulatable auditing in micro- databases. Proc. 2011 4th IEEE International Conference on Computer Science and Information Technology. China. Academic Journal Electronic Publishing House. 2011, 6. 364–368.

5 沈辉,张龙.基于 WinPcap 的网络数据监测及分析.计算机科学:2012,39(10):15–18.

6 马俊,高建瓴,孙斌,赵振民.WinPcap 网络监听技术的研究与改进. 2007 通信理论与技术新发展.第十二届全国青年通信学术会议论文集.北京.中国学术期刊网络出版总库. 2007,下册.1877–1882.

7 张运明.协议行为审计关键技术研究与实现[硕士学位论文].长沙:国防科学技术大学,2010.

8 郭丽红,吴海涛.TDS 协议分析与漏洞检测.计算机工程.2009,35(18):127–129.

9 殷泰晖,李帅.基于 TNS 协议的 Oracle 数据库安全性改进方法.合肥工业大学学报(自然科学版).2012,35(2):6-9.

10 Forest JJ. Objective analysis of process safety audit data with Microsoft access. Process Safety Progress, 2011, 3(30): 221–231.