中国农业大学学报 2006,11(4):109-113
Journal of China Agricultural University

事务级数据库入侵检测系统的设计

刘大勇[1,2] 张玉清[3]
(1. 中国科学院 研究生院,北京 100039; 2. 廊坊师范学院,河北 廊坊 065000;
3. 中国科学院 研究生院国家计算机网络入侵防范中心,北京 100039)

摘 要 针对传统入侵检测系统对数据库入侵检测时只能检测出非法用户,而不能检测出该用户进行的具体恶意事务操作的问题,设计了事务级数据库入侵检测系统。在现有入侵检测技术和角色访问控制理论的基础上,训练阶段采用数据挖掘技术对数据库访问角色的日志进行数据事务间的关联规则挖掘,形成知识规则库;在数据库系统正常运行阶段,利用入侵检测算法检测数据库用户异常行为和进行的恶意事务操作。实验测试结果表明,与传统数据库入侵检测系统相比,本设计根据数据依赖关系进行检测,检测粒度更细,维护相对容易;系统适用于对数据库入侵检测要求比较细化的环境。

关键词 数据库安全; 入侵检测; 数据挖掘; 数据依赖
中图分类号 TP 393.08 **文章编号** 1007-4333(2006)04-0109-05 **文献标识码** A

Design of an intrusion detection system with transaction-level database

Liu Dayong[1,2], Zhang Yuqing[3]
(1. Graduate School of Chinese Academy of Sciences, Beijing 100039, China;
2. Langfang Normal University, Langfang 065000, China; 3. National Computer Network Intrusion Protection Center, Graduate School of Chinese Academy of Sciences, Beijing 100039, China)

Abstract The purpose of the paper is to design a new type of intrusion detection system with a transaction-level database, which can be used to detect illegal users and their malicious transactional operations on the basis of the intrusion detection theory, the role-based access control mechanism and the data mining technology. In the training period, the rule database is formed by the sequential pattern discovery method to mine the role log files. When the database works, the system can find malicious transactions by using the algorithm of database intrusion detection. Based on the test, we draw the conclusion that the detection granularity is finer and its maintenance is easier.

Key words database security; intrusion detection; data mining; data dependency

数据库入侵[1]是指没有被授权的个人或组织非法使用了数据库,或者虽然被授权但却滥用自身的权利。目前,国内外对数据库入侵检测技术的研究主要集中在操作系统层,即只能针对 SQL 注入等攻击方式进行检测,对于内部攻击方式则不能识别[2];数据库入侵检测技术缺乏专用的数据库入侵检测产品。在对数据库的入侵容忍技术研究[3]和对恶意事务的恢复技术研究[4]中对数据库的入侵检测要求精确到事务级,这样才能通过事务之间的依赖关系对数据库系统进行事后恢复。

在现有数据库入侵检测技术研究中,基于数据挖掘技术的入侵检测是当前研究的一个重要方向[5]。文献[6]提出使用用户轮廓来检测数据库用户的滥用行为;但由于其工作局限在用户级范围,只能识别出用户是否非法使用了数据库,而无法判断用户操作的恶意事务。此外,文献[6]和[7]中的数据挖掘方法实质上是对用户项目频繁集的挖掘,在新增用户后系统需要重新学习,否则会产生没有相

收稿日期: 2006-04-26
基金项目: 国家自然科学基金资助项目(60573048); 北京市科技计划基金资助项目(H020120090530)
作者简介: 刘大勇,副教授,主要从事网络安全、数据库应用的研究,E-mail: liudyliudy@126.com;张玉清,研究员,博士生导师,通讯作者,主要从事网络与信息安全的研究,E-mail:zhangyq@nipc.org.cn

应用户规则的情况,因此在用户维护时系统管理也过于繁琐。本研究旨在通过对数据之间依赖关系的相对稳定性和角色对数据库访问的规律性的研究,实现对数据库恶意事务入侵的检测。

1 基本原理及系统设计

1.1 系统模型设计

对于数据库内部用户提升权限后的入侵,在操作系统层的入侵检测系统无法检测其进行的恶意事务活动,可以由数据库管理员 DBA 对事务日志进行手工分析,以检测攻击活动。但对于数据库系统来说,每天产生的事务日志数量非常庞大,检测时不仅需要花费 DBA 大量时间和精力,而且漏报率较高,检测效果不好。相比之下,数据挖掘技术非常适用于从历史行为的大量数据中提取出有用的信息,发现隐藏在数据背后的角色模式和特征。所以本研究在设计入侵检测系统时采用数据挖掘技术挖掘角色进行的事务日志操作,形成知识规则库。设计的基于数据挖掘的数据库入侵检测系统模型结构[8]主要由数据采集、数据预处理、数据挖掘、入侵检测 4 部分组成(图 1),分别工作在训练阶段和入侵检测阶段。

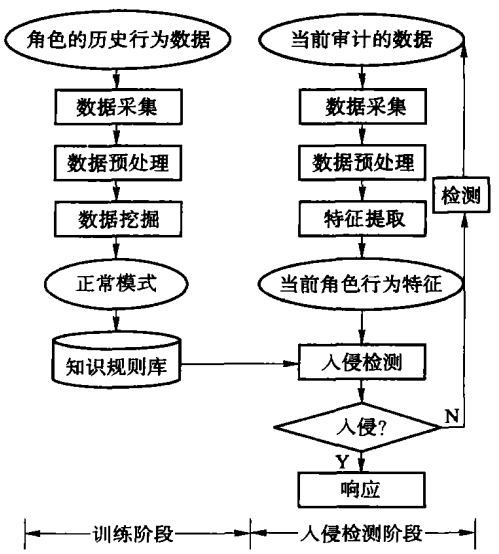

图 1 基于数据挖掘的入侵检测系统模型

Fig. 1 Intrusion detection system based on data mining

数据采集:在训练阶段,对数据库服务器主机日志文件中角色正常操作的历史行为数据进行特征采集,为构造知识规则库做准备;在入侵检测阶段,对服务器当前审计数据进行收集,为入侵检测做准备。

数据预处理:将采集到的数据进行集成和处理,为下一步的数据挖掘或特征提取准备数据。

数据挖掘:采用数据挖掘技术从系统数据中提取有关行为特征和规则,从而用于建立数据库安全正常模式。

知识规则库:知识规则库中存有系统需要的正常模式,数据库入侵检测系统将角色的行为特征与其进行比较判断,如果不符合则可以判断出角色的行为是入侵行为;否则为正常行为。

特征提取:采用类似于数据挖掘的技术从当前角色的行为数据中提取当前角色的行为特征。

入侵检测:系统根据入侵检测算法,从知识规则库中提取相关规则数据,对当前角色行为特征进行入侵检测;根据检测的结果作出相应的行动,如果属于入侵行为则系统作出报警,并采取一定措施防止入侵。

1.2 角色学习

根据访问角色进行学习,主要原因是:首先,当今主流商业数据库系统都支持基于角色的访问控制,以简化授权管理;其次,数据库系统在对用户进行权限验证时,需要对角色权限进行验证;第三,为了克服由于用户维护频繁引起系统维护频繁的问题,可以对数据库审计日志根据角色分组形式挖掘。因为相对于用户来说,角色授权后进行的权限变动较少,所以系统维护代价也较小。

1)用户、角色和权限[9]。用户由用户名/口令字构成;权限是用户对一个功能点能做的操作;角色是一个已命名的权限集合。相互之间的对应关系为:用户和权限之间没有直接对应关系,一个用户可以扮演多个角色,多个用户可以扮演相同的角色;一个角色对应了由若干个权限组成的集合,一个权限也可以分属多个不同的角色。

使用数据库系统时,所有用户都通过同一个登录页面登录,登录页面对用户的用户名/口令字进行验证,如果合法则在会话(Session)对象的集合里面记录用户的角色。在使用任何一个功能点之前验证用户的权限是否足够,从 Session 对象里面获取用户角色,从数据库里面查看用户是否有足够的权限。

2)角色学习设计。以 Oracle 数据库系统的用户权限管理方式为例,该系统采用基于角色和对用户直接授权的方式,用户自动具有自身模式所有对象的操作权限。可根据需要将其他模式中的表、视图等对象权限和系统权限对用户进行授权,本设计要求采用严格的角色授权,即要求用户的权限分配

只能通过角色间接授权。用户登录后,通过在 Oracle 数据库字典中 dba_role_prvis 视图获得该用户具有的所有角色信息,实现用户到角色的映射。

例 1:对管理员→教师→学生角色继承关系的学习。

为了记录数据库角色关系之间的信息,需要定义数据库访问角色集合 AR(AR1,AR2,…,ARn),其中 1,2,…,n 为访问角色的权值,代表其在角色继承关系中的级别。本例中"学生"角色为最低级别分配权值 1,"教师"角色继承"学生"角色分配权值 2,"管理员"角色继承"教师"角色分配权值 3。AR1 代表"学生"角色,AR2 代表"教师"角色,AR3 代表"管理员"角色。在系统训练阶段,按照角色继承关系学习,先对低级别角色进行学习,高级别角色在继承低级别角色的行为后继续学习自己的特有行为。

1.3　事务日志表设计

对于系统来说需要记录数据库系统所有角色的事务操作,而数据库系统自身所提供的审计功能不能详细记录事务的操作内容,这就要求对数据库事务操作自定义审计内容,本设计的事务日志表关键字段说明如下。

TransactionID:事务 ID;

IsUpdate:事务执行完毕后,是否有记录内容被修改;

IsDelete:事务执行完毕后,是否有记录被删除;

IsInsert:事务执行完毕后,是否有记录被新增;

DataItem:数据项被修改的内容;

IsMalicious:事务是否为恶意事务;

UserID:初始化这个事务的用户 ID;

AccessRole:该用户具有的所有访问角色。

1.4　数据事务间的关联规则挖掘

在数据库应用中,虽然基于数据库开发的应用程序经常改变,但相对于数据库表间的结构和数据间的关系来说改变很少,可以说数据之间的依赖关系具有相对稳定性。本设计采用数据事务间的关联规则挖掘方法[10]挖掘数据之间的依赖关系[11],实现对数据库中数据项事务之间的相关性分析,得到基于角色的关联规则库。

本系统针对数据项被恶意修改的入侵进行检测,因此在工作中借助于写操作的事务定义 1、2 进行研究,对单纯读模式的操作事务忽略不计。

定义 1:数据项 x 的读序列〈r(d_1),r(d_2),…,r(d_n),w(x)〉,表示事务在更新数据项 x 前需要依次读取数据项 d_1,d_2,…,d_n。

定义 2:数据项 x 的写序列〈w(x),r(d_1),r(d_2),…r(d_n)〉,表示事务在更新数据项 x 后接着需要对数据项 d_1,d_2,…,d_n 进行写操作。

下面通过对 teacher 角色进行关联规则挖掘,说明其生成角色知识规则库的关键步骤。

1)在数据库日志中利用 AprioriAll 算法[12]由用户根据经验给定一最小支持度如 30 %进行挖掘得到顺序模式(表 1)。

表 1　对支持度 ≥30 %的顺序模式挖掘生成的数据项事务关联规则

Table 1　Data dependency rules generated by mining sequential patterns

顺序模式	支持度/ %	关联规则	信任度/ %
teacher r(2)	35	teacher w(2) →r(2)	100
teacher w(3)	40	teacher w(5) →r(6)	100
teacher r(2),w(2)	40	teacher w(4) →r(6)	83
teacher r(6),w(5),w(4)	40	teacher w(5) →w(4)	80
teacher r(7),r(6),w(4)	30		

2)将只包含读或写操作的事务去掉,如 r(2)、w(3)。

3)将包含读和多个写操作的事务进行拆分,如把 r(6),w(5),w(4)拆分为读序列 r(6),w(5)和 r(6),w(4),写序列 w(5),w(4),生成读序列集〈teacher r(2),w(2);teacher r(6),w(5);teacher r(6),w(4);teacher r(7),r(6),w(4)〉和写序列集 < teacher w(5),w(4) > 。

4)生成数据项事务关联规则,如果规则的信任度大于最小信任度,则加入到数据项事务关联规则集,形成知识规则库(表 1)。注意对于读序列集中 w(4)可生成信任度为 83 %的 w(4) →r(6)和信任度为 50 %的 w(4) →(r(7),r(6))两条规则,设最小可信度为 70 %,所以只保留前一条规则。

1.5　恶意事务的检测

读写规则生成后,在数据库正常运行阶段,可以通过检测数据库日志文件检测恶意事务[13],那些不符合数据关联规则而修改了数据项的事务被标志为恶意事务。

例 2:已知事务 T_1:r(2),r(5),w(2),r(6),r(1),w(7),r(3),r(4),r(5),w(5),判定是否为恶意

事务?

在 T_1 中对数据项2、7和5进行了写操作,对于数据项2,规则 $w(2) \rightarrow r(2)$ 满足,因为数据项2在写前已经被事务所读;对于数据项7,因为规则里没有挖掘到数据关联,所以不需要对其进行检测;对于数据项5,规则 $w(5) \rightarrow r(6)$ 满足,但规则 $w(5) \rightarrow w(4)$ 不满足,因为数据项5在写后没有数据项4被 T_1 所写,由此可判定 T_1 是恶意事务。

2 算法设计

本系统算法设计分为2部分:利用数据挖掘技术的训练阶段算法和系统运行时的入侵检测阶段算法。

2.1 训练阶段算法

在系统训练阶段通过对角色的正常行为挖掘得到知识规则库,其算法设计如下。

输入:正常行为日志,最小支持度 minSupport,最小可信度 minConfidence;输出:正常行为知识规则库[14]。

1)初始数据库的浏览,对数据库所包含的表、视图、主键和外键约束进行标识。

2)对正常审计跟踪事务数据信息按照角色 ID 值进行升序排列形成 TAi,TAi 包含所有角色 ARi 所有事务的操作($1 \leq i \leq n$)。

3)for 在角色序列 ARi 中的每一个角色(首先浏览低级别的角色 AR1,循环学习到最高级别角色 ARn)。

4)挖掘数据之间的依赖关系算法[14]:

初始化读序列集 RS = {}和写序列集 WS = {}

初始化读规则集 RR = {}和写规则集 WR = {}

借助于挖掘算法建立序列方式 X = { Xi| Xi 的支持度 support(Xi) > 最小支持度 minSupport}

for Xi where | Xi| > 1

if 在 Xi 存在写操作

for 每一个写操作 Wi 属于 Xi

if 读操作$\langle r(d_{i1}), \cdots, r(d_{in}), w(d_i)\rangle$不属于 RS and $\langle r(d_{i1}), \cdots, r(d_{in}), w(d_i)\rangle$不为空

将$\langle r(d_{i1}), \cdots, r(d_{in}), w(d_{di})\rangle$加入到 RS

if 写操作$\langle w(d_i), w(d_{j1}), \cdots, w(d_{jn})\rangle$不属于 WS and $< w(d_i), w(d_{j1}), \cdots, w(d_{jn})\rangle$不为空

将$\langle w(d_i), w(d_{j1}), \cdots, w(d_{jn})\rangle$加入到 WS

for 在 RS 中的每一个序列

if 支持度 support ($\langle r(d_{i1}), \cdots, r(d_{in}), w(d_i)\rangle$) / support ($\langle w_i(d_i)\rangle$) > 最小信任度 minConfidence

将 $w(d_i) \rightarrow r(d_{i1}), \cdots, r(d_{in})$ 加入到 RR

for 在 WS 中的每一个序列

if 支持度 support ($\langle w(d_i), w(d_{j1}), \cdots, w(d_{jn})\rangle$) / support ($\langle w_i(d_i)\rangle$) > 最小信任度 minConfidence

将 $w(d_i) \rightarrow w(d_{j1}), \cdots, w(d_{jn})$ 加入到 WR

2.2 入侵检测阶段算法

在数据库正常运行期间对用户的异常活动进行检测,并对其进行操作的恶意事务进行记录,借助于滑动窗口[15]的思想,算法设计如下。

输入:数据库日志;输出:恶意事务。

1)获得当前用户的角色信息,将当前窗口中审计跟踪事务数据信息按照角色 ID 值升序排列。

2)将相同角色的事务数据信息与知识规则库中该角色的规则进行比较。

3)如果2)中,比较的规则符合说明事务是正常事务,如不符合说明是恶意事务,则记录该事务为恶意事务并报警输出。

4)将窗口向前推动,为空说明全部角色检测完毕退出,不为空则继续重复2)~4)。

3 结束语

与传统数据库入侵检测系统通过对挖掘用户项目频繁集来检测用户的异常操作相比,本研究提出的对访问角色事务关联规则进行挖掘的方法,其检测粒度更细,能够检测出数据库系统用户的恶意事务入侵,维护也相对容易。为了提高系统性能,本设计的数据挖掘算法 AprioriAll 有待改进。另外,由于数据依赖关系算法对数据关联的依赖程度较高,如何在较低的数据关联度下,提高检测正确率和降低误报率,也需要进一步研究。

参 考 文 献

[1] Bertino E, Sandhu R. Database security-concepts, approaches, and challenges[J]. IEEE Transactions on Dependable and Secure Computing, 2005, 2(1): 2-19

[2] Low W L, Joseph L. DIDAFIT: Detecting intrusions in databases through fingerprinting transactions[C]. Spain: International Conference on Enterprise Information Systems, 2002: 264-269

[3] Luenam P,Liu P. The design of an adaptive intrusion tolerant database system. Proc[J]. IEEE Workshop on Intrusion Tolerant Systems (ITS '02),2002,7:14-21

[4] Ammann P,Jajodia S. Recovery from malicious transactions[J]. IEEE Transactions on Knowledge and Data Engineering,2002,15(5):1167-1185

[5] Lee W,Stolfo S J. Data mining approaches for intrusion detection[C]. San Antonio: Proceedings of the 7th USENIX Securtiy Symposium,1998:533-567

[6] Chung C Y,Gertz M. DEMIDS: A misuse detection system for database systems[C]. Amsterdam: In Third Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems,1999,11:159-178

[7] 王丽娜,董晓梅. 基于数据挖掘的网络数据库入侵检测系统[J]. 东北大学学报(自然科学版),2003,24(3):225-228

[8] Lee W,Stolfo S J. A data mining framework for building intrusion detection models[J]. IEEE Security and Privacy,1999,5:120-132

[9] David F,Ferraiolo,Sandhu R,et al. A proposed standard for role-based access controls[J]. ACM Transactions on Information and Systems Security,2001,4(3):224-274

[10] Srikant R,Agrawal R. Mining sequential patterns: generalizations and performance improvements[C]. Avignon: Proc of the 5th International Conf on Extending Database Technology,1996:3-17

[11] Yang J,Wang W. Mining long sequential patterns in a noisy environment[C]. Wisconsin: Proceedings of the 2002 AC SIGMOD International Conference on Management of Data,2002:406-409

[12] 毛国君,段立娟,王实,等. 数据挖掘原理与算法[M]. 北京:清华大学出版社,2005:199-202

[13] Liu P,Ammann P,Jajodia S. Rewriting histories: recovering from malicious transactions[J]. Distributed and Parallel Databases,2000,18 (1):7-40

[14] Hu Y,Panda B. A data mining approach for database intrusion detection[C]. Madrid: Proceedings of the 2004 ACM Symposium on Applied Computing,2004:713-716

[15] 凌军,曹阳,尹建华,等. 基于时态知识模型的网络入侵检测方法研究[J]. 计算机学报,2003,26(11):1591-1597