第 32 卷　第 6 期　　计　算　机　工　程　　2006 年 3 月
Vol.32　№ 6　　**Computer Engineering**　　**March 2006**

·安全技术·　　文章编号：1000—3428(2006)06—0148—03　　文献标识码：A　　中图分类号：TP39

一种有效支持计算机取证的审计机制研究

曾剑平，郭东辉

(厦门大学物理与机电工程学院物理系 EDA 实验室，厦门 361005)

摘　要：审计机制是获取原始证据的一种主要途径，针对目前访问控制模型在审计机制设计中的不足，采用 Markov 链对主体访问客体的行为进行建模及预测，确定某次访问时主体的可疑程度，并根据可疑程度决定将原始证据写入不同等级的日志文件。按照这种方法生成的日志文件，能有效减小证据存储所需要的空间，缩短取证时间。

关键词：计算机取证；日志文件；Markov model

Research on Effectively Supported Computer Forensic Audit Mechanism

ZENG Jianping, GUO Donghui

(EDA Lab, Dept. of Physics, School of Physics and Mechanical & Elrctrical Engineering, Xiamen University, Xiamen 361005)

【Abstract】Audit mechanism provides a main way to get the users' operation record, but there still exists some deficiency in audit mechanism. So a new audit mechanism is proposed based on Markov model. The mechanism can predict the access mode and log files are grouped into three grades according the prediction result, and this can lead to smaller storage and shorter time to get witness.

【Key words】Computer forensic; Log file; Markov model

几乎所有的计算机系统都存在不同程度的安全隐患或安全漏洞[1]，入侵检测模型改变了访问控制中的审计机制，在线或离线地分析判断非法入侵行为。为了用法律手段来解决入侵行为而造成的危害，计算机取证技术[2] 可以从大量的原始数据中进行证据的提取及回放，取证的方法有很多，但目的是一致的，即发现证据以证明：发生了什么，发生的地点，发生的时间，谁做的以及如何做的等 5 个问题[3]。

计算机取证过程的 5 个步骤[4]包含了证据的识别、传输、保存、分析和提交，目前这方面的研究并不多见，相关的研究有：文献[5]研究了日志文件的安全存储问题，增加一台可信计算机作为日志的备份存储，同时改进了日志系统使得日志的联机存储和备份存储之间实现透明。文献[6] 侧重从网络结构方面构造一个安全审计日志服务器。文献[7]研究了使用数据挖掘算法从日志文件中发现证据。这些研究工作的核心都是日志文件，而日志文件的组织方式、文件结构等对计算机取证过程有很大的影响。目前的访问控制模型在日志生成时，只是一种简单的行为记录，这种日志生成方式容易产生很多的缺点，主要表现在以下两个方面：

(1)日志文件规模很大，管理困难

在一个存在大量的用户访问的计算机系统中，如果都将这些行为记录到日志文件中，必将导致日志文件急剧增大，从而使得日志文件的保存、传输需要花很长时间，日志的管理是一件很困难的工作。

(2)从日志文件中寻找证据的时间长，取证难度大

由于证据的提取往往需要涉及较长时间范围内的日志记录，因此，证据提取算法必须面对大量的日志记录。日志文件中的许多记录实际上对证据的提取是没有用的，反而容易导致证据提取算法效率的低下，造成取证时间长，取证难度大。

针对以上这两个问题，本文在访问控制模型中使用 Markov 链对主体访问客体的行为进行建模，根据该模型可以预测主体在某个时刻访问客体的可能性，进而可确定主体访问行为的可疑程度，根据可疑程度决定将访问记录写入到相应等级的日志文件以及写入的内容。经过这样处理之后，日志记录将大大减小，从而可以有效缩短证据的分析提取时间。

1 支持计算机取证的审计机制

审计机制是主体在访问客体的过程中起作用的，其基本功能是将这种访问行为记录到日志文件中。本文提出的这种审计机制对此进行了改进，基本原理如图 1 所示。这个改进了的审计机制由 Markov 模型的建立、行为预测与判断、日志写入 3 个主要功能以及 Markov 模型、日志文件两个主要的存储文件组成。

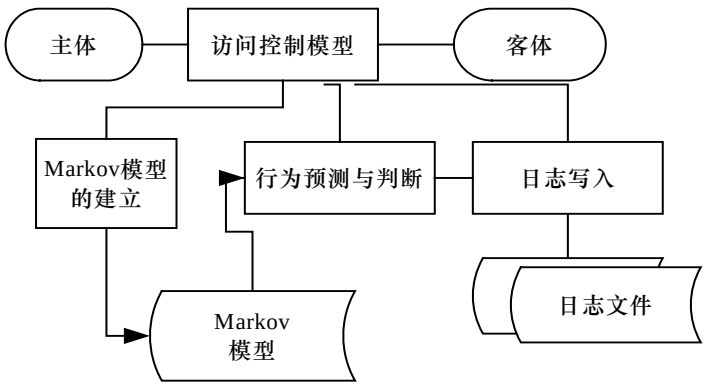

图 1 支持计算机取证的审计机制

它的工作原理描述如下：

(1)Markov 模型的建立

根据主体对客体的每一次访问，计算客体之间跳转的概率，从而为每个主体建立一个 Markov 模型，它表达了主体的在正常情况下

基金项目：国家自然科学基金资助项目(60076015)；国家人事部留学人员创业基金资助项目

作者简介：曾剑平(1973—)，男，博士生，主研方向：信息安全；郭东辉，教授、博导

收稿日期：2005-03-14　**E-mail**：zeng-jian-ping@hotmail.com

的访问行为。模型建立的目标是尽量反映这种访问行为，以便提高行为预测的准确度。

(2)行为预测与判断

行为预测过程是将主体当前所处的状态(即所处的客体)，与 Markov 模型的转移概率计算主体在下一个时刻最可能访问的客体，并得到访问该客体的概率，从而可以确定本次访问的可疑度。

(3)日志写入

与事先规定的可疑度等级相对应，日志文件也按照一定的级别分类并设置。可以根据可疑度决定访问客体的行为写入哪个类别的日志文件，而且需要记录的日志信息的详细程度也是各不相同的。因此，在这个改进了的审计机制中，日志是分级存储的。

采用日志分级存储带来的几个明显的优点是：1)可以针对不同等级的文件采用不同的管理方式，如对于可疑度较高的日志文件，需要制定并使用严格的备份、保密措施，以及延长保存时限等;而对于一般等级的文件则可以不必很严格，因为其中的日志信息的安全性要求不高。2)一般情况下，证据提取算法只需要在可疑度等级较高的日志文件中就可以分析提取证据，这样所需要处理的日志记录就大大减小，从而可以有效地缩短取证时间。3)在同一时刻，允许往不同等级的日志文件中写入日志记录，而不会造成因写入操作的冲突而引起的写入等待，因此，可以很好地解决大量数据的写入问题。

这个审计机制中需要解决的几个关键问题是：1)要求模型预测的准确度高；2)尽量缩短模型预测的时间复杂性；3)如何保证在不同等级的日志文件之间寻找日志信息。

为了方便后续的描述，本文约定如下的表示符号：s 表示主体、o 表示客体。并做如下的定义：

定义 1 访问会话(Access Session)：主体 s 通过访问控制模型许可之后，进入客体 o 所在的应用模块到 s 退出这个模块为止，这个过程在计算机中的逻辑映象称为一次访问会话。

2 模型的建立及预测

2.1 Markov 模型的适用性

Markov 过程是一类重要的随机过程，可以用来描述许多动态系统问题。参数集和状态集都为离散集的 Markov 过程，称为 Markov 链。

用 $\{X(t), t = 0,1,2,\ldots,n\}$ 表示一个随机过程，相应的状态集为 $E = \{i_0, i_1, \cdots, i_n\}$。

则该随机过程满足下面的条件，它具有 Markov 性：

$$P\{X(t+1) = j \mid X(t) = i_t, X(t-1) = i_{t-1}, \ldots, X(0) = i_0\} = P\{X(t+1) = j \mid X(t) = i_t\} \quad (1)$$

该条件表明 t+1 时刻的状态只与 t 时刻的状态有关，而与 t 之前的状态无关。在常见的随机过程中，独立随机过程与独立增量随机过程都满足 Markov 性[5]。而满足下面条件的 Markov 链称为齐次 Markov 链。

$$P\{X(t+1) = j \mid x(t) = i\} = p_{ij} \quad (2)$$

该条件说明从状态 i 到状态 j 的转移概率与现在的时刻 t 无关。

齐次 Markov 链不包含具体的时间信息，因此，根据概率转移矩阵可计算不同状态之间跳转的概率，常用于作预测。

在一个访问控制系统中，建立齐次 Markov 链，首先需要确定一个用于表示状态的随机变量。分析如下：

客体可以分成功能层客体和数据层客体两类。数据层的客体是通过操作系统的文件管理原语来调用的，而功能层客体是为用户直接提供软件功能的集合，易于在应用软件中进行控制。因此，本文以功能层客体为例说明，把功能层客体(下面简称客体)看作是一系列的菜单或功能页面，并受访问控制模型的监控。

如果把客体 $o_i, i = 1\ldots n$ 按照某种顺序排列起来构成一个线性表 L：

$$L = \{o_1 \quad o_2 \quad \ldots \quad o_n\}$$

用随机变量 X(t)表示 t 时刻访问的某个客体 o_i 在 L 中所处的位置 Pos(Pos>=1)，对应于 Markov 链的状态。则可以构造随机序列 $\{X(n), n > 0\}$ 作为 Markov 链，但它不满足式(2)，因此，预测精度是有限的。为此本文对 L 做预处理，即根据一定的时间范围对该序列 L 进行划分，使得 $L = L_1 \cup L_2 \cup \ldots \cup L_n$，对 $L_i, i = 1\ldots n$，中的每个客体的访问满足式(2)。因此主体访问通过这种方式构造出来的客体集合满足齐次 Markov 链的条件，这有助于预测精度的提高。

这样的划分是有实际含义的，例如在许多的应用系统中，用户的使用习惯一般是随着时间变化的，如用户在上午或下午的操作习惯是不同的。

2.2 Markov 模型的建立

为了提高精度，在利用 Markov 模型进行预测时，通常可以使用多步概率转移矩阵。根据 Chapman-Kolmogorov 方程[8]，多步 Markov 链的转移概率矩阵具有如下的性质。$A(i+j) = A(i) \times A(j)$，$A(n)$ 表示 n 步概率转移矩阵。因此，多步转移概率矩阵可以通过一步转移概率矩阵来计算得到。

设一步概率转移矩阵 $A(1) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & & & \\ \vdots & & & \\ p_{m1} & & & p_{mm} \end{bmatrix}$，

其中 m 是客体数。

因此，n 步概率转移矩阵的计算复杂性是 $O(m^n)$，可见，当 m 或 n 增加时，概率转移矩阵的计算复杂性将大大增加。

为解决这个问题，本文对客体的排列及分布划分成合适的树型结构，在实际的应用系统中，功能层客体划分成树型结构也是一种常见的做法。如图 2 所示的树型分布中，可以将树中的所有客体按照父-子节点进行划分。

对于由客体 $\{o_1 \quad o_2 \quad \ldots \quad o_n\}$ 构成的客体树 Tree，可以划分成以下形式的若干个子树：

$$sub-tree(j) = \{o_m, \ldots o_k, o_j\},$$

其中，

$$Parent(o_m) = o_j,$$

…

$$Parent(o_k) = o_j$$

并对子树中的客体访问行为建立一个概率转移矩阵。经过这样处理之后 n 步概率转移矩阵中所需要表达的客体数就可大大减少，从而有效减小计算复杂性。

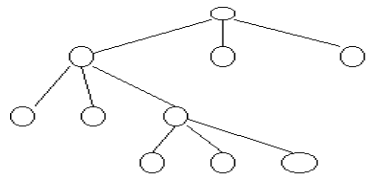

图 2 客体树

通常，Markov 链模型可以表示为一个 3 元组，

$$MC = (X, A, \Psi) \tag{3}$$

其中，X 是一个离散随机变量；A 称为概率转移矩阵，矩阵中的每一项是转移概率 tp_{ij}；Ψ 为初始状态分布。

经过上述的时间范围划分及客体分布划分之后，建立的随机序列是不能构造成式(3)表示的 Markov 链，为此，需要对 Markov 链进行扩展。扩展后的模型如式(4):

$$MC = (X, T, G, A, \Psi) \tag{4}$$

其中 $T=\{t_1,t_2,..,t_n\}$ 表示 n 个时间划分，$G=\{g_1,g_2,..,g_n\}$ 表示客体的 n 个划分，$A=\{A(t_1,g_1),A(t_1,g_2),..,A(t_1,g_n),A(t_2,g_1),..,A(t_2,g_n),..,A(t_n,g_1),..,A(t_n,g_n)\}$ 即表示在不同时间划分下，各个客体划分对应的概率转移矩阵。Ψ 表示不同时间划分下的初始概率分布，即 $\Psi=\{\Psi(t_1),\Psi(t_2),\ldots,\Psi(t_n)\}$。

2.3 访问行为的预测

设在某个时刻 t 的主体所处的状态 $H(t)=[h_1 \quad h_2 \quad \ldots \quad h_n]$，因为主体在一个访问会话中，不可能同时访问不同的客体，因此，$h_1 \quad h_2 \quad \ldots \quad h_n$ 中只有一项不等于 0。预测 t+1 时刻状态的算法如下:

(1)判断时刻 t 所处的时间划分 st;

(2)判断 $H(t)$ 中不为 0 的项，记为 h_m，它对应的客体可以从线性表 L 中取得，即 $o_m = L[h_m]$;

(3)求 O_m 的客体划分 x:

在客体树 Tree 的所有子树 $sub-tree(i)$ 中，作如下的判断:

如果 O_m 只存在于一棵子树 $sub-tree(tp)$ 中，

记 x = tp;

否则，只考虑符合下面条件的子树 $sub-tree(tp)$:

对于 $sub-tree(tp)$ 中的客体所有 O_i，不存在 $Parent(o_i)=o_m$。

记 x = tp;

(4)根据 时间划分 st 和客体划分 x 获得相应的概率转移矩阵 $A(1)=A[st,x]$;

(5)计算 t 时刻状态对 t+1 时刻的预测结果: $V_1(t+1)=H(t)\times A(1)$，计算 t-1 时刻状态对 t+1 时刻的预测结果: $V_2(t+1)=H(t-1)\times A(2)$，依次类推。

(6)计算 t+1 时刻的综合预测值:

$$\begin{aligned} V(t+1) &= a_1\times V_1(t+1)+a_2\times V_2(t+1)+\ldots+a_n\times V_n(t+1) \\ &= a_1\times H(t)\times A(1)+a_2\times H(t-1)\times A(2)+\ldots+a_n\times H(t-n+1)\times A(n) \end{aligned}$$

其中，$\sum_{i=1}^{n} a_i = 1$，a_i 表示过去的每个预测值对 t+1 时刻的影响因子。对于有较强关联关系的两个客体之间的影响因子，可以取大一点。所谓的关联关系是指在执行一个客体之前，需要执行另一个客体的必要程度。

(7)取 $V(t+1)$ 向量中值最大的元素对应的客体在线性表 L 中的位置，并构成 $H(t+1)$。

3 日志文件的生成

3.1 确定主体的可疑度

从 Markov 模型得到 t+1 时刻主体访问各个客体可能性向量 :

$$V(t+1)=[v_1 \quad v_2 \quad \ldots \quad v_n]$$

定义主体 s 访问某个客体 O_i 的可疑度为: $SK(s)=1-v_i$，可疑度等级分为高可疑、中等可疑和低可疑 3 个等级。可按照下面的原则，将可疑度对应到某个可疑度等级。

(1) $SK(s) \le 0.4$， 可疑度等级为低可疑;

(2) $0.4 < SK(s) \le 0.7$，可疑度等级为中等可疑;

(3) $SK(s) > 0.7$，可疑度等级为高可疑。

3.2 日志结构

整个日志系统也分成 3 个不同的等级，分别存储处于不同可疑度等级下的主体行为记录，保存的内容即是主体进入相应客体之后所做的操作。整个日志文件系统如图 3 所示。

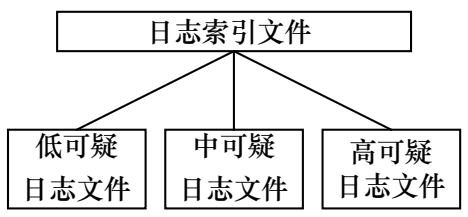

图 3 日志系统结构

日志索引文件记录了访问操作的记录号及其所在的文件等信息，而具体的客体访问信息则保存在相应等级的日志文件中，这些日志文件的记录与日志索引文件中的相应记录通过一个记录号关联。

日志文件的文件结构由文件头和文件记录组成，如下:

(1) 文件头

序号	内容
1	开始时间
2	结束时间
3	可疑度等级

(2) 文件记录

序号	内容
1	记录号
2	主体标识
3	功能层客体
4	访问时间
5	访问会话 ID
6	对功能层客体的所有操作记录
7	下一个操作记录号

日志文件经过这样处理之后，证据提取算法一般只需要在高可疑的日志文件中进行证据分析、提取，如果它需要到中等或低可疑的日志文件中寻找原始证据时，则可以通过记录号在日志索引文件中找到原始证据所在的文件。而高可疑等级的日志文件与简单的日志文件相比，其记录数大大减少。因此可以有效地缩短证据提取时间。

3.3 日志的写入

为了便于证据提取算法的分析，需要对写入日志的信息进行标准化。在一次访问会话过程中，可以定义若干个基本的操作原语，每种原语可以有不同的格式，并反映不同的语义。如:

INPUT NAME VALUE

GET

PRINT

不同可疑等级的日志文件的写入内容的区别: 对于高可疑的访问，记录详细的信息；对于低可疑访问行为只记录概括性记录，如什么时间进入客体、什么时间离开。

4 总结

为了用法律手段来解决入侵行为而造成的危害，运用计算机取证技术已经成为一种主要趋势。然而计算机取证过程主要是对日志文件的操作。目前的日志文件采取简单的组织

(下转第 153 页)

怀疑度或者支持度超过相应阈值的频繁闭模式，计算相应的怀疑度和支持度，向本域中的监控部件和其他域中的协作代理发送报警消息。

为了对报警消息进行关联分析和汇总，提出了一个基于频繁闭模式挖掘的报警关联与分析算法。由于篇幅所限，这里不详细介绍基于频繁闭模式挖掘的报警关联与分析算法。

2.3 消息交换

当协作代理经过分析后检测到一个入侵时，就自动生成一个 XML 报警消息，向本域中的监控部件和与入侵事件有关的其它域中的协作代理发送，同时将入侵事件存入入侵事件数据库中。监控部件接到报警信息后，可以使用适当的样式表将信息显示在屏幕上，并根据信息内容和危害程度决定是否采取响应措施。协作代理通过对来自其它域的报警信息的分析，可以检测到较复杂的入侵行为。上述基于 XML 消息交换实现协作的过程如图 3 所示。

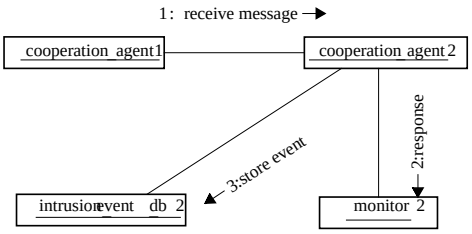

图 3 协作过程

由于基于 TCP/IP 的网络不能提供安全的消息传输通道，因此消息在传输前需要进行加密处理。但目前还没有正式的标准和相应的产品。W3C 提出的新型分布式对象计算协议—SOAP 协议(简单对象访问协议)[6]，为服务的请求、消息的格式定义了简单的规则，主要用来通过 XML 文档传递方法参数。一个 SOAP 消息就是一个 XML 文档，其文档元素为<Envelop>包括<Header>、<Body>和<Fault>。因此，可以利用 SOAP 协议通过在<Body>中放置加密的 XML 报警消息来实现消息的传输。

2.4 XML 消息与数据库之间的数据交换

协作代理可以将符合 IDMEF 标准数据格式的 XML 报警消息存入数据库中，也可以将数据库中的内容生成 XML 报警消息。这部分工作主要包括 XML 文档的解析与生成。需要把符合 IDMEF 的 XML 文档进行解析，数据存入数据库中；或者把数据库中的数据取出，生成符合 IDMEF 的 XML 文档，在网络中传输，以实现在不同域之间进行数据交换。

由于目前比较成熟的商用数据库系统都是关系数据库系统，而专门的 XML 数据库系统还处于研究阶段，没有实际产品可用。因此采用关系数据库来存储 XML 报警消息。按照 IDMEF 的数据模型结构，把 XML 数据拆分到不同的字段，存入数据库中。由于文档模式十分复杂，因此采用文档对象模型 DOM 对 XML 文档进行解析。在实现时，利用微软的.NET 框架中提供的 DOM 类来访问 XML 文档。

3 结语

由于入侵行为向分布式协作化入侵方向发展，因此对分布式协作入侵检测模型和分布式入侵检测与响应协作机制的研究在今天具有重要意义。本文提出的基于 XML 消息交换的分布式协作入侵检测模型，使用 XML 格式通过协作代理在不同的域间传递消息，并为此设计了相应的 XML Schema 文档。利用现有的 XML 相关技术，可以实现对 XML 文档的生成、解析、显示和传输。通过各入侵检测部件和入侵检测系统的互相协作，可以充分发挥各检测部件的优势，检测较复杂的协同攻击，增强对系统的保护。

参考文献

1 Spafford E H, Zamboni D. Intrusion Detection Using Autonomous Agents[J]. Computer Networks, 2000, 34(4): 547-570.

2 Porras P, Schnackenberg D, Staniford-Chen S, et al. The Common Intrusion Detection Framework Architecture[EB/OL]. http://www.isi.edu/gost/cidf/drafts/architecture.txt, 1999.

3 IETF. Intrusion Detection Message Exchange Format Data Model and Extensible Markup Language (XML) Document Type Definition [EB/OL]. http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-10.txt, 2003.

4 杨海松，李津生，洪佩琳. 分布开放式的入侵检测与响应架构——IDRA[J]. 计算机学报, 2003, 26(9): 1177-1182.

5 Pal P, Webber F, Schantz R E, et al. Survival by Defense-enabling[C]. Proceedings of the New Security Paradigms Workshop. New York: ACM Press, 2001: 71-78.

6 World Wide Web Consortium. SOAP 1.1[EB/OL]. http://www.w3.org/TR/#Notes.2001.

(上接第 150 页)

方式，使得日志文件规模很大，管理困难，而且从日志文件中寻找证据的时间长，取证难度大。针对这两个问题，本文在访问控制模型中使用 Markov 链对主体访问客体的行为进行建模，根据该模型可以预测主体在某个时刻访问客体的可能性，根据这种可能性确定主体访问行为的可疑程度，进而根据可疑程度决定将访问记录写入到相应等级的日志文件以及写入的内容。经过这样处理之后，日志记录将大大减小，从而可以有效缩短证据的分析提取时间。

参考文献

1 Tian Zhihong, Fang Binxing, Yun Xiaochun. An Architecture for Intrusion Detection Using Honey Pot[C]. 2003 International Conference on Machine Learning and Cybernetics, 2003,4: 2096

2 Civie V, Civie R. Future Technologies from Trends in Computer Forensic Science[C]. IEEE, Information Technology Conference, 1998: 105-108.

3 梁锦华，蒋建春，戴飞雁等. 计算机取证技术研究[J]. 计算机工程, 2002, 28(8).

4 钱桂琼，杨泽明，许榕生. 计算机取证的研究与设计[J]. 计算机工程, 2002, 28(6).

5 陈爱莉，张焕国. 一种支持计算机取证的日志系统的设计[J]. 计算机工程与应用, 2003,39(15).

6 Jiqiang L, Zhen H, Zengwei L. Secure Audit Logs Server to Support Computer Forensics in Criminal Investigations[C]. TENCON '02, Proceedings of IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering , 2002,1:180-183.

7 Abraham T, de Vel O. Investigative Profiling with Computer Forensic Log Data and Association Rules. Proceedings of 2002 IEEE International Conference on Data Mining, 2002:11-8.

8 李裕奇. 随机过程[M]. 北京：国防工业出版社, 2003.