IDS决策过程中的时空一致性研究*)

罗光春　卢显良
(电子科技大学信息中心　成都610054)

摘　要　本文讨论了将数据融合技术运用到入侵检测系统中的方法,并提出了一个基于数据融合技术的入侵检测机制——DFIDM。本文重点研究了数据提取和对象提取过程中的时空一致性问题,在该机制中,有多个检测器搜集系统日志文件、网络流量信息、网络数据包等数据。这些数据首先由决策器进行本地决策,在通过数据提取和对象提取阶段的时间和空间校准之后,传送到融合中心进行决策。本文最后通过实验证明,采用了该机制的入侵检测系统能获得更好的性能。

关键词　入侵检测,时间一致性,空间一致性,性能

The Research on Consistency of Time and Space of IDS

LUO Guang-Chun　LU Xian-Liang
(Information Centre of UEST of China,Chengdu 610054)

Abstract　This paper introduces a method of Intrude Detection Based on Data Fusion,and presents a new mechanism—DFIDM. In this paper,we mostly research the consistency of time and space on Data Refinement and Object Refinement. In this mechanism,a few of sensors are configured to collect data,such as log file,information of Network Traffics and data package of network. First,these data will be made a local decision by decision-maker,and then adjusted in the period of Data Refinement and Object Refinement,last will be transferred to fusion center. As it showed by the research result this mechanism can improve the performance of IDS.

Keywords　Intrude detection,Consistency of time,Consistency of space,Performance

1　引言

对入侵行为的检测与分析不够准确,是目前各类入侵检测系统普遍存在的问题,表现为检测结果存在较大的虚警率和漏报率。为了解决这一问题,一般的思路是使用多个入侵检测系统或简单增加检测引擎,系统管理员根据多个系统或多个引擎的报告结果来评估是否存在攻击。但是,该办法中各个入侵检测系统不具备全局性的合理分工。更重要的是,对攻击的评估往往由系统管理员凭借主观经验来进行,增大了系统管理员的工作,同时降低了评估的可靠性。

下一代的IDS需要通过各种异质的分布式网络传感器来获取并融合数据,以形成对网络系统的态势估计(cyberspace situational awareness)[1,2],本文将所设计的基于数据融合技术的入侵检测机制称为DFIDM(Data Fusion Intrusion Detection Mechanism),在该机制中,采用了数据融合技术来实时评估攻击是否存在,攻击所属的种类以及如何对攻击进行响应。其理论依据不再赘述,可参见文[3]。为保证在检测过程中,目标系统内入侵行为整体状况和检测时序的准确性,DFIDM需要在目标系统全网内维持时空上的一致性,本文对此进行了设计与实现。

2　DFIDM的体系结构

本文所处理的融合对象具有一定层次,如图1所示。其中,判断有无入侵是最初步的分析,其次是入侵的特征、行为分类,在此基础上可对网络整体的状态进行评估,最终得出关于具体威胁的分析。由此可知,本文所设计的对数据进行提取、分析和处理的机制,是按照零散而大量的数据、有效信息、整体状态这一逻辑顺序来进行的。

分析类型	分析级别
威胁分析	高
状态评估	
入侵行为	中
入侵特性	
入侵的存在	低

图1　处理对象的层次模型

DFIDM的层次模型和功能布局如图2所示,按整个系统的功能划分为四个层次[4]:(1)首先通过遍布系统各处的分布式传感器获取原始数据(基于网络或基于主机的基元、标识符、入侵次数与频度和其它一些描述);(2)原始数据经过校准过滤后形成标准的原始数据记录库,并形成相关对象;(3)对象提取在时间(或空间)上相关联,其数据按统一标准校准,观测数据可以根据入侵检测基元(intrusion detection primitive)关联、配对、分类。对象通过配位的行为、依赖、共同的源点、共同的协议、共同的目标、相关的攻击率或其它高层次的属性而被检测出,形成一个基于对象的聚集的集合;(4)利用融合决策算法,对状态进行提取以形成对入侵行为的威胁评估;在以上层次结构以外,管理策略独立存在并管理、维护整个系统。

*)本文由电子科技大学青年基金支持。罗光春　副教授,博士;卢显良　教授,博士生导师。

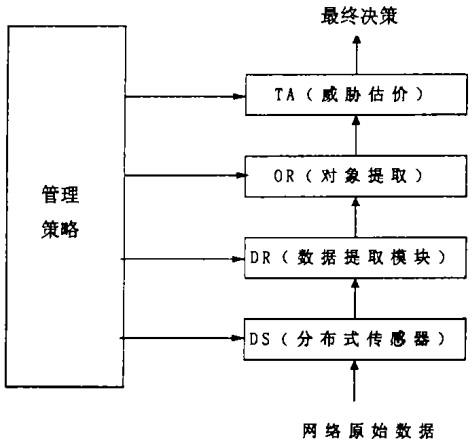


图2 DFIDM 的层次和功能模型

各部分功能说明如下：

(1)分布式传感器 DS(Distributing Sensors)布置在系统的各个部分，主要分为基于主机的传感器和基于网络的传感器两类，这些传感器用来获取来自网络或者主机的原始数据；(2)数据提取模块 DR(Data Refinement)应布置于传感器本地，以便于提高效率；(3)对象提取 OR(Object Refinement)和威胁估价 TA(Threat Assessment)两个模块可一同布置于融合中心 FS(Fusion Center)。这有利于对象提取时，通过校准形成一个基于对象的聚集的集合，同时也有利于状态提取和威胁评估时的数据集中处理。

3 数据提取过程中的一致性

如前所述，DS 布置在系统的各个部分来获取原始数据，这些数据可以是基元、标识符、入侵次数与频度其它一些描述，DS 的输出应该是完整的网络原始数据样本，DS 将其输出到 DR。DR 布置于传感器本地，以便于提高效率，原则上 DR 与 DS 应一一对应，从整个系统角度看，DR 的作用是对 DS 的原始数据进行预处理。主要功能为：(1)对 DS 所提供的数据进行过滤，保证所处理数据的规范性；(2)对 DS 提供的原始数据进行本地决策。将原始数据对照规则数据库中各元素进行遍历，对符合入侵规则的行为填写原始数据库，并形成相关对象。在此，对象可由对象标志符 OID 的方式表达、存储并用于传输；(3)对本地决策后形成的对象标志符 OID 加上时间标记 TT(Time Tag)和空间标记 ST(Space Tag)，由此形成带有时空标记的 OID，以便于后续处理时数据在时间和空间上的一致性。

3.1 时间一致性

DFIDM 需要在目标系统全网内维护统一的系统时间周期。其原因在于，整个网络上检测到并加以传输的数据可能具有异步性，如果不在时间上加以标记和对准，将无法对目标系统内入侵行为真实的整体状况和检测时序做出准确判断。分析一个简单例子：两个同时来自网络不同局部的本地决策对象 OID，由于网络延迟(Delay)等原因而实际到达时间相差很大。这种情况下，先到达的入侵检测对象 OID 所代表的入侵行为会被首先处理，而后到达的可能再次被系统处理，这就会造成数据处理不同步。而 DFIDM 的工作机制是在各本地决策的基础上进行最优融合决策，如果来自各监测器的本地决策在时空上出现混乱，也即最终决策的数据来源出现混乱，将会直接导致决策结果的不准确，甚至无效。综上，为保证从目标系统获取的数据具有时间上的一致性，DFIDM 在网络各节点处维护标准的系统时间周期 SSTC(System Standard Time Cycle)，该时间周期在系统内具有一致性和唯一性。

系统初始化时，对各节点时间统一初始化，同时各节点进入计时周期，在系统运行过程中，每个周期结束时，系统重新初始化 SSTC，以保证时间的一致性。SSTC 的取值不宜太大或太小，太大则一方面系统出现步调不同节点数的几率相应增加，而且单点不一致的时间长度会较大；SSTC 的值如果太小，会导致系统开销大幅度增加，故实际应用中 SSTC 值的设置应该适中。有了系统标准时间周期 SSTC，DR 就可对 OID 加上时间标记了。

3.2 空间一致性

系统一致性的另一个方面表现在空间一致性，即系统的最终决策必须对入侵行为发生的空间位置进行准确定位。一方面系统在后面的各级融合和决策中需要对入侵发生位置进行记载，更重要的是，只有这样系统才能对发生入侵的节点或网段进行准确响应。为此，系统维护一个传感器节点物理位置表 SLT(Sensors Location Table)，该表对所有节点统一编号，采用简单的一维数组即可实现。当节点新增或取消时，系统将根据变化情况修改 SLT。

可见，DR 主要完成数据预处理工作，在 DS 提供的原始数据基础上进行过滤、生成数据库和 OID、加上时空标记等工作。从整个系统来看，DR 输出的对象为本地决策 LDM(Local Decision-Making)，主要包括时间标记、空间标记和入侵类型等内容，如图3所示。

时间标记 TT	空间标记 ST	入侵类型 OID

图3 本地决策 LDM 示意图

4 对象提取过程中的一致性

OR 针对各节点传来的 LDM 进行处理，应布置于融合中心 FC。对象提取的作用是在时间(或空间)上对数据进行关联，其数据按统一的时间和空间标记校准、关联、配对、分类。各节点传输来的对象 LDM 按照相互依赖、共同的时间或空间、共同的协议、共同的目标、相关的攻击率或系统其它预设的高层次属性而被检测出，形成一个基于对象的聚集的集合。

具体有以下几个步骤：

(1)OR 管理和维护一个增量添加的初始数据缓存空间 ODC(Original Data Cache)，接收到各节点传来的 LDM 后首先按到达的时间先后顺序将其存入该数据空间，该数据空间是一个对象存储的集合，用数组或链表实现均可；

(2)对象的时间校准。这里对于网络故障如断网等因素暂不考虑，但即使在正常情况下，稍大一点的网络延迟也可能导致同时产生的对象到达 ODC 的时间略有不同，因此对 ODC 里的内容进行实时处理可能导致同一时刻数据的不完整。针对这一情况，系统采用了三级延迟缓存的设计思想。即对于初始数据缓存空间 ODC 存储的对象不作即时处理，ODC 作为第一级缓存；再设置二级缓存空间 SDC(Secondarily Data Cache)，用作时间校准。此外，系统还设置三级缓存空间 TDC (Tertiary Data Cache)，用作对最后按规则选出的输出对象集合进行缓存。

DFIDM 设计了两个时间数值，工作间隔时间 WT 和系统延迟数值 DV(Delay Value)，实际应用中可设为一个较小

的值，例如 WT＝1ms、DV＝10ms。系统每一个 WT（此处为1ms）对 ODC 中的对象进行一次处理，将时间标记相同的对象放入 SDC 里同一集合中，如果该集合已经存在，只需加入新的对象元素即可；如果不存在就创建新的集合，并加入该对象元素。这样，无论有无延迟，对象将从 ODC 中按时间标记不断被筛选加入到 SDC 的相对应集合中。同时，系统对 SDC 内的各集合不断轮询，当集合创建时间等于或大于 DV 时，认为该集合里存放了已消除时间延迟影响的数据，可进行下一步处理。

结合以上机制，考察一下延迟实际发生时的情况。当系统在1个 WT 到达时，对 ODC 中的对象进行时间判别，将其存入到 SDC 中相应集合。而由于网络延迟等问题，一个相同时间标记的对象在 n 个（$n<10$）WT 后才到达，此时，该对象仍能被存入到 SDC 对应的集合中。但是，如果延迟太大，当 $n>10$ 的情况下，该到达对象将被舍弃。也就是说，系统预设，如果在 DV 范围内的延迟是可以接受的，超过则该数据无效，将被丢弃。

(3)对于 SDC 中创建时间等于或大于 DV 的集合，OR 可认为已通过时间校准。此时 SDC 中一个即将输出到 TDC 的集合，为$\{O_1,O_2,\cdots,O_m\}$，其中 m 为该集合的对象个数，每个对象的时间标记都相同，设该时间标记为 t_0，如图4所示。

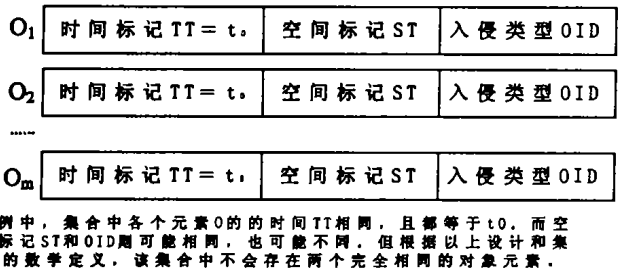
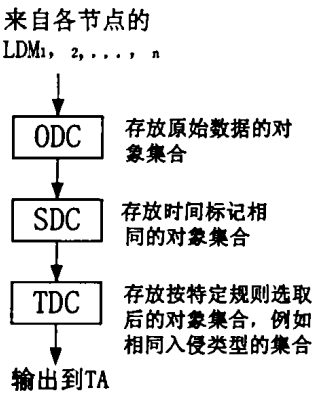


图4　即将从 SDC 传输到 TDC 的一个集合的情况



接下来可对其按若干规则进行处理，规则的选取由系统预设。主要可选择以下规则：共同的空间、共同的协议、共同的攻击类型、相似的攻击发生率等。按照不同规则生成的集合的聚集是不同的，其内容也可能重叠，既可以同时按多种规则，也可以仅选取一、两种最重要的规则。为简化问题起见，本文仅讨论选取一种规则的情况加以讨论，更多规则的情况与此相似，不再赘述。按此规则形成的最终集合即可存入三级缓存空间 TDC，作为 OR 的最终结果，并传输到 TA。

在对集合进行聚集时，最重要的规则无疑就是入侵类型。对于 DFIDM 而言，OR 输出的是经过时间校准后的、基于不同入侵类型、并包含 n 个集合的集合群（n 为该时刻入侵行为的类型数）。整个集合群所有集合内各元素的时间标记相同，等于 t_0，而每个集合内入侵类型 OID 相同，如图5。从系统初始化10ms 后，每1ms 产生 n 个集合（n 为该时刻入侵行为的类型数，数量可变），在没有入侵的情况下，OR 输出为空。

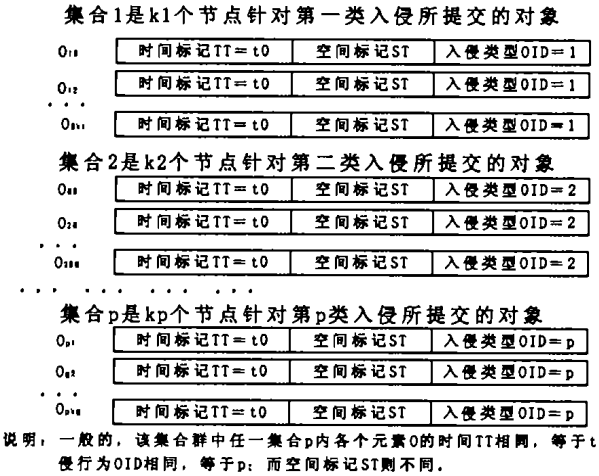


图5　OR 最终输出到 TA 的集合群

以上工作流程如图6所示。

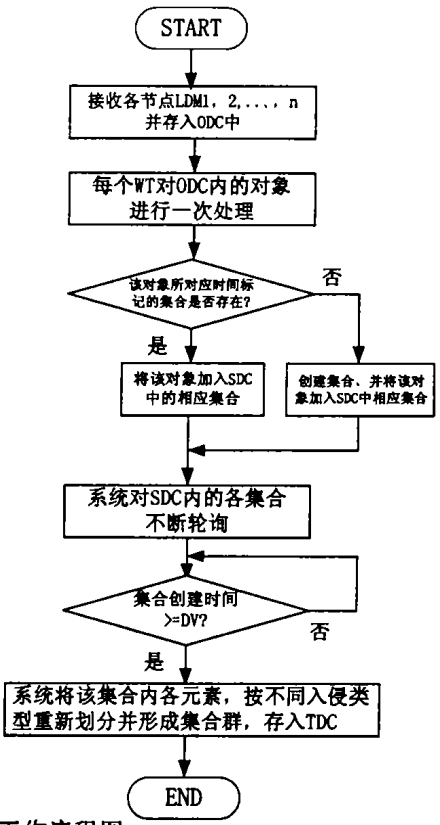


图6　OR 的工作流程图

（下转第139页）

$$A=\begin{bmatrix}-am_1 & a & 0\\ 1 & -1 & 1\\ 0 & -b & 0\end{bmatrix},B=\begin{bmatrix}-a(m_0-m_1)\\ 0\\ 0\end{bmatrix}$$

$$C=[1\quad 0\quad 0]\quad H=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

取从系统为：

$$\dot{z}=Az+B\phi(Cz)+u$$

$$q(t)=Hz(t)$$

其中，$u=G[p(t-\tau)-q(t-\tau)]$，τ 为时滞。

对于给定的时滞，应用上述定理，可以验证是否能找到使得主从系统同步的控制矩阵 G，进一步，应用定理2，可以估计到保证同步的最大时滞。为方便，在实验中取，$P=Q=Z=I$，为了利用 LMI 求解，先把定理2矩阵中的 G^TZG 用单位阵代替，经用 MATLAB 计算，得 $\tau=0.0912$，

$$X=1.0e+007\begin{pmatrix}-4.5558 & -0.0017 & -0.4109\\ -0.0017 & -4.9450 & -0.0026\\ -0.4109 & -0.0026 & -4.5425\end{pmatrix}$$

$$Y=1.0e+003\begin{pmatrix}-0.0032 & -0.0100 & -0.0000\\ -0.0054 & -8.8985 & 0.5847\\ 0.0003 & 0.5575 & -0.9626\end{pmatrix}$$

$$G=1.0e+003\begin{pmatrix}0.0022 & 0.0100 & 0.0000\\ -0.0000 & 8.8983 & -0.5774\\ -0.0000 & -0.5581 & 0.9612\end{pmatrix}$$

经过验证，此矩阵满足定理2中的不等式。需要指出的是，由于 τ 的计算依赖于定理中几个矩阵的选择，因此这里的最大时滞估计是有一定保守性的。

结论 本文讨论了主从 Lur'e 系统混沌同步的输出反馈控制问题，利用 Lyapunov 方法和矩阵不等式技巧，得到了一个依赖于时滞的混沌同步的充分条件，该条件可借助于 LMI 方法进行验证。同时，应用优化问题分求解，对同步的最大允许时滞做了一个保守的估计。最后进行了数值验证。

进一步的讨论可以考虑

$$C:u=G[p(t-\tau)-q(t)] \tag{11}$$

的情况，从实际应用的角度来看，这应该更合理。

参考文献

1 Wu C W, Chua L O. A unified framework for synchronization and control of dynamical systems. Int. J. Bifurcation and Chaos, 1994, 4:979～998

2 Curran P F, Chua L O. Absolute stability theory and the synchronization problem. Int. J. Bifurcation and Chaos, 1997, 6(7): 1375～1383

3 Suykens J A K, Curran P F, Chua L O. Master-slave synchronization using dynamic output feedback [J]. Int. J. Bifurcation and Chaos, 1997, 3(7): 671～679

4 Yalcin M E, Suykens J A K, Vandewalle J. Master-slave synchronization of Lur'e systems with time-delay. Int. J. Bifurcation and Chaos, 2001, 11:1707～1722

5 Liao Xiaoxin, Chen Guanrong. Chaos synchronization of general LUR'E systems via time-delay feedback control. Int. J. Bifurcation and Chaos, 2003, 13:207～213

6 PooGyeon P. A delay-dependent stability criterion for systems with uncertain time-invariant delays. IEEE Trans. Automat. Contr., 1999, 4(44):876～877

7 俞立. 鲁棒控制线性矩阵不等式处理方法. 清华大学出版社, 2002. 158～168

8 Chen G. http://www.ee.cityu.edu.hk/～gchen/chaos-bifur.html

（上接第123页）

5 实验结果

本文仅对入侵检测过程中的时空一致性进行了分析和设计。由其体系结构可知，作为一个完整的检测过程，还应包括最终融合和决策等工作。为此 DFIDM 设计了基于空间因素[5]、时序因素、历史记录、人工加权等因素的最终融合决策模块。限于篇幅，这些工作将在其它文章中加以阐述。

为验证检测过程中的时空一致性，本文对 DFIDM 进行了一系列实验，实验环境为1个融合中心 FC（Intel 服务器 1G），5个节点（PC 赛扬666），全部置于同一局域网中，并通过集线器连接，保证所有数据均同样流入5个节点，攻击数据由1台 PC 赛扬666提供。根据 DFIDM 的需要，设计并实现了 DR、OR 所对应的数据库和功能模块。开发环境为 Red Hat linux8.0，用 ANSI C 编写代码，数据库为 My SQL 4.0。入侵类型选择了 TCP Flood、UDP Flood、ICMP Flood、后门攻击、缓冲区溢出攻击等5种，通过下载相关攻击工具实现。分别对同时攻击类型为1、3、5的情况进行3组实验，每组实验分别进行了300次，共900次。实验结果表明，DFIDM 能完整、准确地记录攻击过程中入侵行为的空间标志，时序的准确性达到了98.56%，从而较好地满足了设计需要。

结论 本文通过对时间和空间一致性的专门处理，保证了系统在最终融合和决策时，入侵数据来源在时序和空间定位上的精确性。而传统的 IDS 在这方面往往有所忽略，从而导致数据来源的混乱。两相比较，提高数据来源的精确性从源头上保证了 DFIDM 的精确性。普通多机系统通常采用系统初始化时统一设定起始时间，其后不再校准的策略。而 DFIDM 中使用标准系统时间周期 SSTC 对此作了一定改进，保证在各节点时间按一定周期循环校准，从而进一步提高了时空精确性。

值得注意的是，本文中所提到的时间校准仍然难以确保系统各节点时间的绝对一致性，这是因为系统时间校准的工作也需要在目标系统内各节点间进行计算和传输，难保不出现因延迟等情况造成的误差。也就是说，可能系统中负责时间校准的控制部件进行了时间周期校准后，认定时间一致性已得到保证，但实际上个别节点上仍可能有误差存在。由于 DFIDM 中对时间一致性要求较高，以 ms 作为时间刻度，这一可能的、个别的、较小的误差仍可能对最终决策性能带来影响。在这样的情况下，DFIDM 所处理的看似达到时间一致性的目标系统，实际上可能是一个目标系统个别节点误差固定的情况，这使 DFIDM 在该时间周期内进行的时间校准、融合决策可能因时间一致性被破坏而出现性能降低。综上所述，对全网内各节点时间一致性的更好保证还需要进一步研究。

参考文献

1 Bass T. Multisensor Data Fusion for Next Generation Distributed Intrusion Detection Systems. In: 1999 IRIS National Symposium on Sensor and Data Fusion, May 1999

2 Bass T. Cyberspace Situational Awareness Demands Mimic Traditional Command Requirements. Signal Magazine, AFCEA, Feb. 2000

3 罗光春，卢显良，张骏，李炯. 一种基于多传感器数据融合的入侵检测机制. 电子科技大学学报（自然版），2004(1)

4 Bass T. Intrusion Detection Systems & Multisensor Data Fusion. Communications of the ACM, 2000, 43(4)

5 Bass T. Intrusion Detection Systems and Multisensor Data Fusion: Creating Cyberspace Situational Awareness. Communications of the ACM, 1999