入侵响应中基于事件相关性的攻击预测算法

王祖俪　程小平
（西南师范大学计算机与信息科学学院　重庆400715）

摘　要　目前的入侵检测系统(IDS)中的响应单元只对检测出的当前安全事件做出响应，而忽略了攻击事件间隐藏的关系及攻击的最终目的。本文针对上述问题在IDS的响应单元中提出了一个利用攻击事件间的相关性对攻击的最终目的进行预测的算法。实验证明该算法提高了网络的预警能力，减少了对误报的响应，并能发现分析引擎的漏报情况。

关键词　预测算法，前驱，后继，入侵响应

An Attack Predictive Algorithm Based on the Correlation of Intrusions Alerts in Intrusion Response

WANG Zu-Li　CHENG Xiao-Ping
(Faculty of Computer Science, South West China Normal University, Chongqing 400715)

Abstract　Traditional intrusion detection systems(IDSs)take actions to the alerts independently, and ignore the logical connections between them. In this paper an attack predictive algorithm in intrusion response part of the IDS is presented based on the prerequisites and consequences of intrusions. And an off-line simulation experiment shows that it can improve the prediction ability of the IDS, and reduce the false alert and find the potential attack.

Keywords　Predictive algorithm, Prerequisites, Consequences, Intrusion response

1　引言

入侵检测作为一种积极主动的安全防护技术，通过监视受保护系统的状态和活动，提供了对网内、网外攻击及误操作的实时检测和保护。其中的响应单元是对产生的安全事件进行及时的响应[1]。但目前的响应只限于对当前事件的响应，没有考虑事件间的相关性和分析引擎可能出现的误报，导致了对事件的响应仅仅局限在事后响应，也会因为误报而产生不必要或者错误的响应。随着攻击的形式越来越多样化、多层次和多步骤，当前的入侵检测系统的响应单元已经不能满足现有的需求[2]。

基于以上的原因，文[3]和[4]提出了在分析引擎中运用攻击树，以计算概率的方式预测攻击的下一步以及攻击的最终目的，但此方法在预测的同时增加了分析引擎的压力，容易造成负载的不平衡，加上计算概率的复杂度较高不适合做实时的预测。文[5]提出了一个基于主机的入侵检测系统中的预测算法，但该算法只能通过对本机日志内容的查询对攻击目的进行预测。

本文从网络入侵检测整个系统的效率出发，在现有检测技术能捕获大量的攻击事件的基础上提出了一个在响应单元中结合攻击事件的前后继关系对攻击进行实时预测的算法。本文第2节给出了该预测算法的具体过程，第3节用一个仿真实验证明该算法的正确性。最后对全文进行小结。

2　本文的预测算法

对于事件间的联系，我们将直接导致这个攻击事件成功的必要条件或者前提条件称为该事件的前驱(prerequisites)，相应地，这个事件成功后会生成一系列直接结果，使这些事件在下一时刻成为可能发生的事件，我们把那些在下一时刻可能发生的事件称为事件的后继(consequences)。

本文所提出的攻击预测算法的核心思想是：通过事件间的前驱后继关系，在响应当前已检测出的事件的同时推测在今后一段时间内同一攻击者对于同个IP可能的进一步攻击以及攻击的最终意图。

攻击者为了达到攻击目的，往往通过一定的攻击步骤来实施攻击，虽然这些步骤很灵活，但有些事件的发生存在明显的先后顺序，而这些顺序往往是不会改变的。比如要想获得某台机器上的资料，必须先获得它的访问权限。我们就是利用这些存在明显先后顺序的事件来建立我们的前驱后继表，以此作为预测攻击的基础。

2.1　事件建模

在该算法中会出现三个表，它们存放在事件数据库中。

静态表 P——用来存放攻击事件的前驱后继关系。

$P=(attack.name, attack.pre, attack.con)$

名称	前驱	后继

其中，*attack.name* 表示攻击事件的名称，*attack.pre* 表示攻击事件的前驱，*attack.con* 表示攻击事件的后继。在实际操作中我们只需要针对具体受保护的网络的保护要求、常遇到的攻击、操作系统等因素来制定符合自己的前驱后继集，而且由于这个表可以独立于系统存在，在实际使用过程中，用户可以根据情况不断调整前驱后继表。

动态表 I——预测结果的暂存表，该表的初始状态为空。

I = (*attack.name*, *attack.srcip*, *attack desip*, *at-*

王祖俪　硕士研究生，主要研究方向：计算机网络，信息安全技术；程小平　教授。

tack. time, tap, source)

名称	源地址	目标地址	时间	I表标志	来源
				1	

其中，*attack. name* 表示攻击名称，*attack. srcip* 表示攻击源地址，*attack. desip* 表示攻击目标地址，*attack. time* 表示攻击时间，*tap* 表示 *I* 表的标志(这里设为1)，*source* 表示事件的来源。

动态表 *H*——算法过程中的过程临时表 *H*，该表的初始状态为空。

H = (*attack. name*, *attack. srcip* *attack. desip*, *attack. time*, *tap*, *attack. pre*, *attack. con*)

名称	源地址	目标地址	时间	标志	前驱	后继
				2		

H 表中的前几个字段的定义与 *I* 表相同，*tap* 表示 *H* 表的标志(这里设为2)，*attack. pre*, *attack. con* 分别表示攻击事件的前驱和后继。

2.2 算法的主要过程

Step1：判断初始状态，进行阈值处理。来自分析引擎的事件，必须在某个时间内达到所规定的阈值才能进入预测状态，得到事件 a 后判断 *H* 表是否为空，若为空，则执行 step2；否则执行 step4。

Step2：激活事件。在 *P* 表中查找 a 的后继，将后继加入 *H* 表中。

Step3：进行预测，这是算法的核心。从 *P* 表中查找 *H* 表中各元素的后继的后继，若为空，则将 *H* 表中该元素及其后继的有关信息送给响应决策单元和 *I* 表，并从 *H* 中删除该元素。读下一事件，返回第一步。

Step4：进行二次响应即预测与实际情况相符时。若 *I* 表中存在事件 a 有该记录，则将记录的有关信息送给响应决策单元进行二次响应。读下一事件，返回第一步。

Step5：判断是否是已激活事件。若在 *H* 表中寻找到 a 记录，则将 a 的后继加入 *H*，从 *H* 表中删除 a，执行 step3。

Step6：发现漏报事件。若 a 的前驱不为空，则查找 a 的前驱是否在 *H* 中，若在，则将 a 的前驱的所有除 a 的后继加入 *H*，并从 *H* 表中删除 a 的前驱，然后执行 step2。否则，直接将 a 的前驱的所有除 a 的后继加入 *H*，执行 step3。

2.3 算法的流程图

在整个算法中，发送至 *I* 表和响应决策单元的事件的有关信息包括事件名称、目标地址、源地址、攻击时间和各表标志。算法的流程图如图1所示。

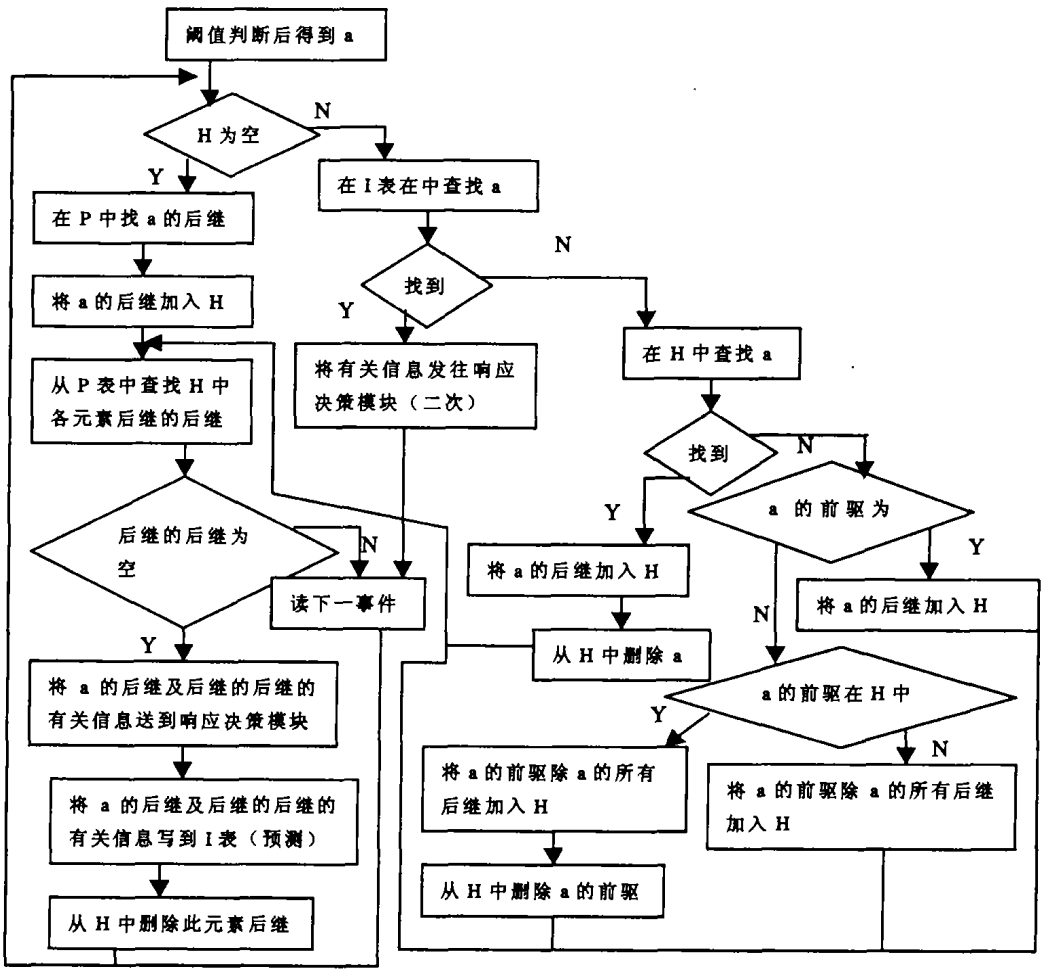

图1

3 仿真实验

3.1 实验概述

我们以 VB6.0及 SQLserver2000为开发工具，对算法进行一个离线仿真实验来检测它的正确性。为了表示算法的一般性，我们用一般节点代替了攻击的名称。实验模拟了10个不同事件(e1,e2,…,e10)在某一时段攻击某个子网的情况，整个过程共完成了三条路径的攻击：第一条路径 e1-e5-e7-e10-e9(源地址192.168.10.123，目标地址192.168.1.9)；第二条路径 e1-e3-e7-e9 (源地址 192.168.30.12，目标地址

192.168.33.111)；第三条路径 e1-e2-e5-e3-e7-e8-e9-e10(源地址192.168.44.11，目标地址192.168.49.123)。事件间的相互关系见表2，表1为模拟阈值处理后的来自分析引擎的事件，程序运行的结果为图2所示，图3、图4是与输入事件表1进行直观的比较。

3.2 实验结果

表1 模拟阈值处理后的来自分析引擎的事件

事件名称	源地址	目标地址	攻击时间
e1	192.168.10.123	192.168.1.9	11:08:50
e5	192.168.10.123	192.168.1.9	11:08:51
e7	192.168.10.123	192.168.1.9	11:08:51
e10	192.168.10.123	192.168.1.9	11:08:51
e9	192.168.10.123	192.168.1.9	11:08:52
e1	192.168.30.12	192.168.33.111	11:08:52
e3	192.168.30.12	192.168.33.111	11:08:53
e7	192.168.30.12	192.168.33.111	11:08:53
e9	192.168.30.12	192.168.33.111	11:08:54
e1	192.168.44.11	192.168.49.123	11:08:54
e2	192.168.44.11	192.168.49.123	11:08:55
e5	192.168.44.11	192.168.49.123	11:08:56
e3	192.168.44.11	192.168.49.123	11:08:57
e7	192.168.44.11	192.168.49.123	11:08:57
e8	192.168.44.11	192.168.49.123	11:08:58
e9	192.168.44.11	192.168.49.123	11:08:59
e10	192.168.44.11	192.168.49.123	11:08:59

表2 各事件间的前后继关系

名称	前驱	后继	名称	前驱	后继
e1	0	e2	e5	e3	e8
e1	0	e3	e6	e4	e10
e1	0	e4	e7	e3	e9
e2	e1	e5	e7	e4	e9
e3	e1	e5	e8	e5	e10
e4	e1	e6	e9	e7	0
e4	e1	e7	e10	e8	0
e5	e2	e8			

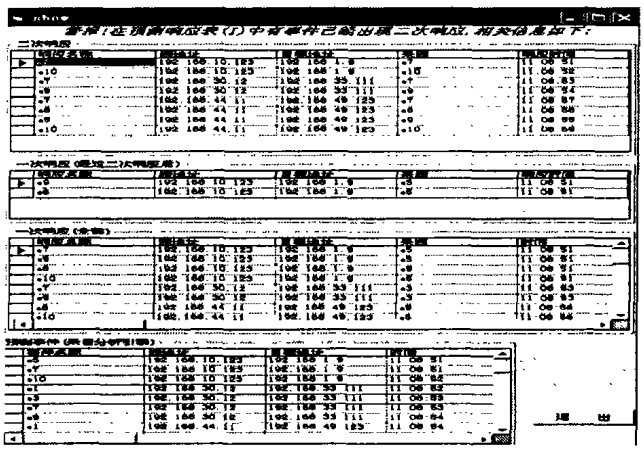

图2 实验运行的最后结果图

时间	输入事件（表F）			预测结果		
时间	事件名称	源地址	目标地址	事件名称	源地址	目标地址
11:08:51	e5	192.168.10.123	192.168.1.9	e7	192.168.10.123	192.168.1.9
11:08:51	e7	192.168.10.123	192.168.1.9	e10	192.168.10.123	192.168.1.9
11:08:51	--	--	--	e9	192.168.10.123	192.168.1.9
11:08:52	e10	192.168.10.123	192.168.1.9	--	--	--
11:08:52	e9	192.168.10.123	192.168.1.9	--	--	--
11:08:53	e3	192.168.30.12	192.168.33.111	e7	192.168.30.12	192.168.33.111
11:08:53	e7	192.168.30.12	192.168.33.111	e9	192.168.30.12	192.168.33.111
11:08:54	e9	192.168.30.12	192.168.33.111			
11:08:56	e5	192.168.44.11	192.168.49.123	e8	192.168.44.11	192.168.49.123
11:08:56	--	--	--	e10	192.168.44.11	192.168.49.123
11:08:57	e3	192.168.44.11	192.168.49.123	e9	192.168.44.11	192.168.49.123
11:08:57	e7	192.168.44.11	192.168.49.123			
11:08:58	e8	192.168.44.11	192.168.49.123			
11:08:59	e9	192.168.44.11	192.168.49.123			
11:08:59	e10	192.168.44.11	192.168.49.123			

图3 输入事件与预测结果的比较(检测算法正确性)

在图2中，预测事件块是来自表1的事件，一次响应(全部)块表示所有预测的结果，二次响应块表示预测结果与实际相同的事件集，以便进行再次响应。一次响应(经二次响应后)块显示已预测出但还为发生的事件。

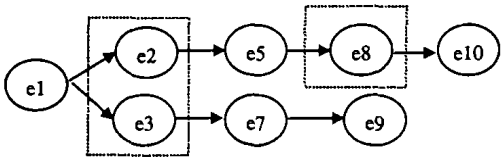

图4 对于分析引擎漏报的预测

图3中对在同一时间上产生的安全事件与算法预测的攻击下一步行为及目标进行了比较。图中箭头表示预测算法均在严重的安全事件产生前就已预测成功，即可采取措施进行预防。

图4是根据前驱后继表2画出的事件间的部分关系。在e1-e5-e7-e10-e9(源地址192.168.10.123，目标地址192.168.1.9)这条攻击路径中，图中虚线框部分是分析引擎漏报的事件，但从图1中可以看出当在11:08:51发现e5时，预测算法就推测e2或e3都有可能已经发生，在e5发生后推测出e8可能会发生，由此在11:08:51成功预测出可能会发生e7、e9，在11:08:52时刻正确预测到e10的发生。

小结 该预测算法在响应当前已检测到的事件的同时对攻击的目的做进一步预测，还可以及时发现由于漏报而形成的潜在隐患，及时予以响应。此外由于加入了事件阈值的处理也减少了对误报的不必要响应。

从实时性和攻击规模来考虑，该算法只对一些高级别的、危害性大的攻击事件进行预测，对于漏报的恢复工作只适合在一种攻击漏报一到两步的情况。由于事件之间的关系仍依靠事先总结的攻击路径模版进行预测，因此引入学习机制将是今后工作的重点。

参 考 文 献

1 邵先供，石鹏. 入侵检测响应系统的分析与研究[J]. 网络安全与应用，2003.9

一种基于无线传感器网络的密钥管理方案*)

郎为民[1] 杨宗凯[1] 吴世忠[2] 谭运猛[1]

(华中科技大学电子与信息工程系 武汉430074)[1] (中国国家信息安全测评中心 北京100091)[2]

摘 要 本文提出了一种无线传感器网络中的密钥管理方案,该方案支持密钥建立和更新等过程,并采用邻近节点概率否决投票的方法由基站撤消受节点的密钥集。同时,考虑到传感器节点资源有限的特点,方案中相关的协议和算法都比较简单,并完全避开了代价昂贵的公钥运算,从而使得方案的计算、存储和通信开销都比较小,执行效率大大提高。

关键词 无线传感器网络,密钥管理,安全方案

A Key Management Scheme for Wireless Sensor Network

LANG Wei-Min[1] YANG Zong-Kai[1] WU Shi-Zhong[2] TAN Yun-Meng[1]

(Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)[1]

(China National Information Security Testing Evaluation and Certification Center, Beijing 100091)[2]

Abstract This paper proposes a key management scheme for wireless sensor network, which supports key establishment, re-keying and revocation of the key ring owned by a compromised sensor node via probabilistic negative polling of its neighbors. Furthermore, the related protocols or algorithms are very simple and no expensive public-key operation is required in view of resource-starved sensor nodes, which minimizes the storage, computation, communication overhead and improves the efficiency of our scheme.

Keywords Wireless sensor network, Key management, Security scheme

1 引言

近年来,随着无线通信、集成电路、嵌入式计算及微机电系统等技术的飞速发展和日益成熟,具有感知能力、计算能力和通信能力的微型无线传感器开始在世界范围内出现。这些传感器具有低成本、低功耗、多功能等特点和无线通信、数据采集、信息处理、协同合作等功能。由这些微型传感器节点构成的传感器网络,能够协作地实时监测、感知和采集网络分布区域内的各种环境或监测对象的信息,并对这些数据进行处理,从而获得详尽而准确的信息,将其传送到需要这些信息的用户。因此,传感器网络[1]是信息感知和采集的一场革命,它能够在任何时间、任何地点和任何环境条件下提供大量详实可靠的信息,因而可以被广泛应用于军事斗争、国家安全、环境监测、交通管理、医疗卫生、制造业和反恐抗灾等领域。

由于传感器网络一般配置在恶劣环境、无人区域或敌方阵地中,加之无线网络本身固有的脆弱性,因而传感器网络安全[2,3]引起了人们的极大关注。传感器网络的许多应用(如军事目标的监测和跟踪等)在很大程度上取决于网络的安全运行,一旦传感器网络受到攻击或破坏,将可能导致灾难性的后果。如何在节点计算速度、电源能量、通信能力和存储空间非常有限的情况下,通过设计安全机制,提供机密性保护和身份认证功能,防止各种恶意攻击,为传感器网络创造一个相对安全的工作环境,是一个关系到传感器网络能否真正走向实用的关键性问题。

传感器网络的研究始于20世纪90年代末期,但安全问题的研究成果近几年才陆续出现,各种密钥管理方案[4~8]不断被提出。但由于传感器网络还未被真正地模型化和量化,无线传感器网络安全方案正处于理论研究阶段,距离实际应用和形成普遍接受的标准还相差甚远。

本文第2节给出了无线传感器网络的参考模型和基本工作流程;第3节全面描述了密钥管理方案中有关的协议和算法;第4节详细分析了方案的安全性和效率;最后对全文进行总结。

2 无线传感器网络的参考模型

传感器网络是由一组传感器以 Ad Hoc 方式构成的无线网络,其目的是协作地监测、感知、采集和处理网络覆盖区域中感知对象的信息,并发布给需要这些信息的用户。传感器网络是由大量体积小、成本低的传感器节点组成的,这些节点具有无线通信、传感、数据处理等功能。一个典型的传感器网络[1]通常包括传感器节点、簇头、基站、无线或有线网络、用户等部分,如图1所示。

集成有传感单元、数据处理单元和通信模块的传感器节点可以随机或者特定地散布在指定的感知区域内,通过自组

*)国家自然科学基金资助项目(60202005)资助.郎为民 博士研究生,讲师,研究方向为传感器网络、信息安全和应用密码学.杨宗凯 博士后,教授,博士生导师,研究方向为电子商务、远程教育和网络安全.吴世忠 教授,主要从事网络攻防技术、应用密码学等研究.谭运猛 博士,副教授,研究方向为电子支付、信息安全和应用密码学.

2 丁勇,虞平,龚俭.自动入侵响应系统的研究[J].计算机科学,2003(10)

3 Geib C W, Goldman R P. plan recognition in intrusion detection systems [J]. IEEE, 2001

4 Ye Nong, Li Xiaoyang. Probabilistic techniques for intrusion detection based on computes audit data [J]. IEEE Transactions on System, 2001.31

5 张永,陆余良.攻击树在多阶段入侵检测系统中的应用[J].计算机应用与软件,2004.8