通用入侵检测知识自优化框架

韩 宏 卢显良 卢 军 陈 波
(电子科技大学计算机科学与工程学院 成都 610054)

Common Intrusion Detection Knowledge Self-optimization Frame

HAN Hong LU Xian-Liang LU Jun CHEN Bo
(College of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054)

Abstract For Intrusion Detection System, it is very important that system has enough and valid detection knowledge set. This heavily depends on experience of an individual administrator. If we could have the experience of individual shared by different systems, the collaborative systems will exchange the new detection knowledge automatically. It will dramatically improve the performance of systems as a whole. This paper presents a novel idea: Intrusion Detection Knowledge Self-Optimization, gives and implements a Common Intrusion Detection Knowledge Self-optimization Frame. The frame could manage different subsystems. The same type of subsystems could share and optimize detection knowledge automatically.

Keywords Distributed, Intrusion detection, Common intrusion detection knowledge self-optimization frame, Knowledge management

1 入侵检测知识管理的现状及问题

在计算机安全问题中,社会工程是一个核心要素,它往往成为系统潜在隐患[1],其最终表现形式之一就是不安全的系统配置和安全策略。在入侵检测领域,正确的系统配置很大程度取决于管理者检测知识的丰富程度。知识的丰富程度主要又取决于知识获取能力,特别是在分布式计算环境中,这种获取能力显得尤为重要。而在开放式入侵检测系统中,检测知识类型的多样性使该问题变得更为复杂。

安全系统中知识管理的目的是优化相关策略,消除认知缺陷造成的安全漏洞。它已引起了学界的关注:在病毒防御方面,简单的方式是客户端主动从中央服务器下载新病毒代码,比如 Norton Antivirus、金山毒霸等;复杂的方式有病毒免疫系统,它模拟生物免疫机制,在世界范围内共享新的病毒信息[2];在防火墙的管理上也提出了 firewall farm 的思想[3],这一思想在笔者从事的国防预研基金项目中已转化成更为完善的分布式防火墙构架;在入侵检测方面,免疫系统也是模拟了生物抗体信息自动传播的特性[4]。以上研究工作的重点就是使安全知识在系统中的传播和管理更为自动化。但它们大多只关心单一系统,而未关注不同系统间的检测知识交换。特别在入侵检测方面,更多的精力放在了检测方法的研究上,而不是整体检测知识的自优化。以流行的软件 Snort 来讲,知识管理涉及的工作最多就是在网站上给出新的规则,再由用户自行获取。而如何使用获得的规则就取决于用户相关知识水平,更不谈该用户是否能定期和及时地获取信息。入侵检测中没有一种方法能涵盖一切,为了整合集成各种方法,有人提出了通用入侵检测框架(CIDF)[5]。遗憾的是,CIDF 关注的是不同系统间检测信息的交互,并未关注不同检测方法带来的检测知识的管理问题。如果纯粹用人工管理,很难想象如何系统、有效、及时地完成任务。

为了解决上述问题,我们提出了开放性知识自优化概念,并给出了框架和具体实现。其核心思想就是以统一管理的方式,使子系统间能自动地交换更新知识,实现系统整体检测知识的自优化。并且在知识优化系统受损后能迅速恢复,继续自优化进程。

2. 开放性知识自优化框架

2.1 总体框架

开放性知识自优化框架的目的是管理不同检测系统的检测知识,使同类型的知识在不同系统间自由交换更新,完成整个系统知识的自动优化。框架的基本单位是管理域,由四部分组成:管理节点、终端节点、守护节点和认证层,见图 1。管理域通过级连形成管理树,我们称为知识社区,见图 2。

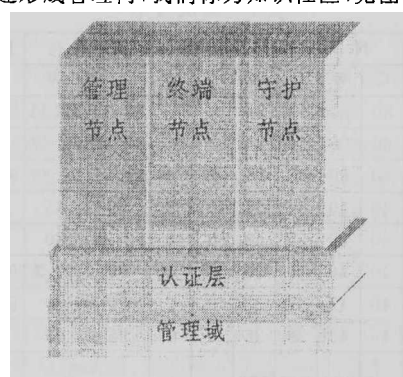

图 1 管理域结构

管理节点是管理域的核心组件,它负责所在域的知识更新、发放,终端节点类型的注册等功能。终端节点管理一个检测子系统的特定知识,比如一个典型的终端节点将管理一个 Snort 系统的检测规则集。守护节点的功能是保证管理节点的有效性,当管理节点失效时,守护节点将自动转化成管理节点以保证整个系统仍能正常运行。认证层向以上三个节点提供安全服务,保证通讯双方的真实性和通讯数据的安全性。从图 2 可见,管理域由管理节点和子节点(终端节点和管理节点)组成,一个管理域 A 可以通过将其管理节点加入另一个

域B的管理节点而成为B的子域。

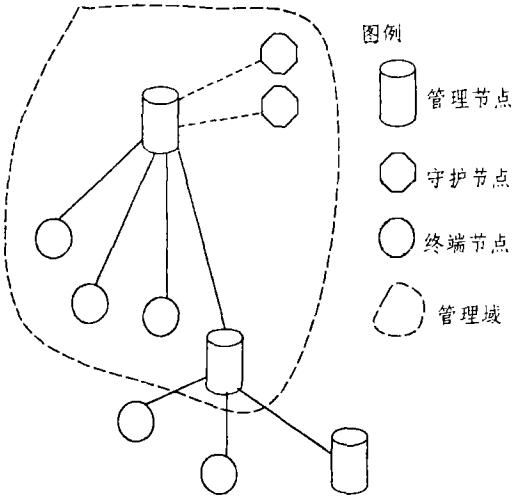

图2 知识自优化框架结构

框架主要工作过程如下：1. 当加入子节点(终端节点或管理节点)到域的管理节点时，子节点注册自己的位置信息、节点类型、知识类型和订购知识的子类型。2. 管理节点给子节点发送最新的所需模块，包括知识发现模块和知识具体化模块。3. 当被管理节点通过知识发现模块发现新检测知识时，提交本域的管理节点。4. 管理节点检查子节点提交的是否为新知识，如果是，则分发到订购知识的子节点。如果管理节点有父管理节点，则向其提交新知识。5. 子节点定期向管理节点查询并更新其订购的知识。

开放性知识自优化框架有以下优点：

· 伸缩性：管理域自由组成管理树这一方式可以容易地布置不同规模的系统。对于小规模系统，一个管理域就可以胜任；对于大规模系统，可以方便地将管理域进行分级处理。这种方式较符合实际系统中按责任单位来划分的惯例。由于系统并未事先区分管理树的深度和级别，理论上可以自由生成一棵无穷大的树，所以有很好的灵活性。

· 自治性：每一个管理域都是自主的工作单位，在其它域失效后，它仍能独立工作。

· 开放性：由于终端节点对检测子系统知识的具体管理独立于整个框架，所以该框架可容纳任何新的检测子系统。终端节点只是通过共同的通讯协议和管理节点产生互动。

· 抑制网络浪涌：当系统中多个节点向顶层节点提交新知识时，重复的知识提交可能引起网络的浪涌。管理域分级的方式能较好地解决这一问题，每一级的管理节点将过滤重复性知识提交，并汇总上报。

2.2 管理节点

管理节点是框架的核心组件，负责整个管理域的知识更新。其功能和服务见图3。管理节点提供三个基本服务：更新服务、注册服务、生存服务。

更新服务是关键服务，它由知识更新服务和模块更新服务组成。知识更新提供推拉两种方式；模块更新集中管理终端节点知识更新所需的模块，包括：知识发现和抽取(发现新的检测规则并提取通用部分)，通用知识的具体化(将收到的通用知识具体化到自己的运行环境中)。更新模块是不同检测子系统发现和更新自身知识的关键。框架本身并不规定更新模块的具体实现，只是负责模块的存储和更新。举例来讲，通过RegisterUpdateModule登记更新模块，知识发现和知识具体化模块共享一个ID，这一ID也是终端节点注册时知识类型的ID。当模块版本更新时，系统将根据终端节点注册的知识类型将对应模块自动发送到终端节点。终端节点和管理节点交互的关键是通讯协议，如图4。第0字节是命令类型，目前为K_Update(知识更新)和M_Update(模块更新)。1～6字节为知识类型的ID。7～8字节为负荷段的长度，剩余的是负荷字段。当命令字段为K_Update时，Payload为检测规则。知识更新采用推拉方式的结合：当有新知识提交时，管理节点将它推到子节点上；同时为了防止通讯原因造成管理节点无法推送，子节点将定时向管理节点查询新知识。图5为推和拉两种知识更新算法：

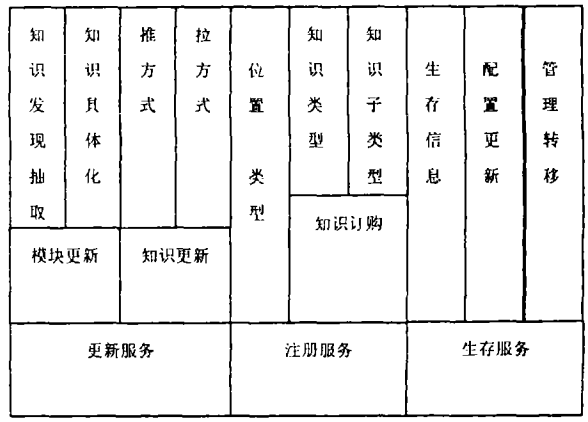

图3 管理节点提供的服务

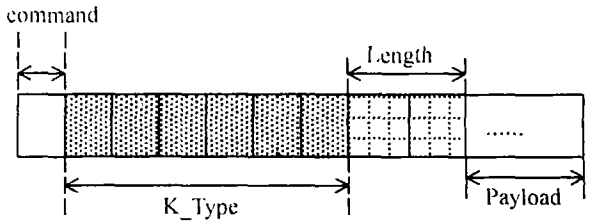

图4 知识更新协议

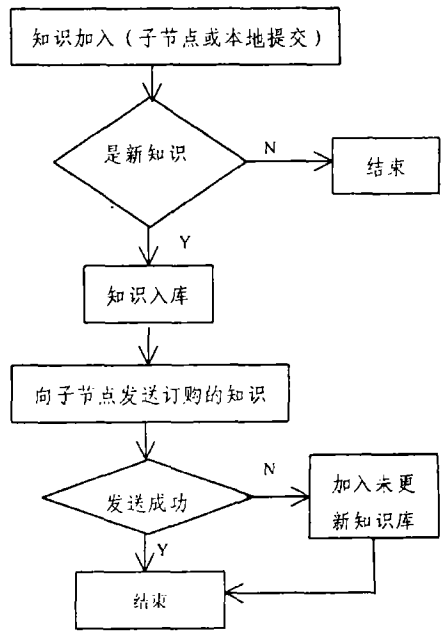

图5-a 推方式更新节点知识

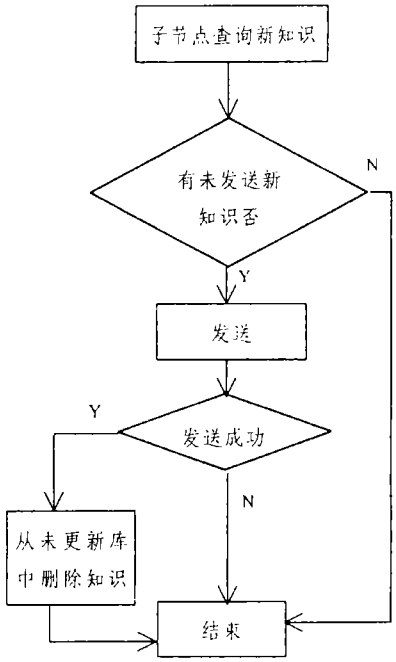

图 5-b 拉方式更新节点知识

注册服务是使子节点和管理节点耦合工作的重要服务，通过注册登记子节点的类型和位置信息(比如 IP 地址)，并指出终端节点的知识类型(比如 Snort 的检测规则)，管理节点就可将本管理域中新的相关知识(比如 Snort 的新规则)发送到对应的终端节点。知识子类型是指一个类型的检测规则子集，比如某节点只关注 FTP 的入侵检测规则。管理节点就可根据注册的要求只发送子集中的新规则。

生存服务是指在管理节点发生故障的情况下，重构新的管理节点所需的相关服务，见图 3。生存信息指示管理节点的存活状态，可采用在心跳信号的方式。配置更新是指管理节点主动向守护节点通告下属子节点的注册变动，使守护节点获知管理域的拓扑结构。管理转移服务涉及一个守护节点被转换成新管理节点的相关服务，它包括检测知识库的迁移。解决方式采用了一种简单的方法：将守护节点作为被管理的子节点加入管理域，其特殊处就是订购全部的新知识。这样系统并不作特别处理，守护节点将备份全部知识信息。

2.3 终端节点和守护节点

终端节点是知识的受体也是新知识的潜在提供者，其结构如图 6 所示。当终端节点加入新知识时，其发现模块将通过更新模块向管理节点提交新知识。发现模块所做的工作是抽取出通用的规则，然后和知识库中的知识比较，当发现新知识时，触发知识更新。更新模块的工作包括提交、接收和查询新知识。知识具体化和知识发现是一个相反的过程，即：将管理者发送来的知识具体配置到相应的检测环境中。以 Snort 来举例：收到的通用规则是 Alert any any -> any 7626 (msg "冰河攻击")。具体化的过程就是指出需要保护的服务器的地址，上面的规则就具体化为 Alert any any -> 202.115.14.145 7626(msg"冰河攻击")。

守护节点的功能是在管理节点失效后代替原节点重构系统，其结构如图 7 所示，包括：存活监测、配置更新、知识更新和域重构。2.2 节已简述了守护节点如何监测系统，并获取管理节点的知识库映像，这里主要叙述多个守护节点在管理节点失效后，如何通过仲裁选出一个守护节点充当新的管理节点。这里不采用集中式的方法，它将造成新的脆弱点。我们采取的是分布式的方法：因为守护节点知道其它守护节点的地址，就可以用 IP 地址的大小为优先权决定是否可以成为管理节点。算法如下：

1. 当管理节点无心跳信息时，守护节点各自计算自己的优先权。

2. 如果自己的 IP 地址对应的整数值为最大，则向其它守护节点发信息(包括自己的 IP 地址)通知自己将成为管理节点。

3. 当时限 T1 内收到其它守护节点的答复，则此节点正式成为管理节点；如有节点未答复，则再通知一次，在时限 T2 内仍无答复，则此节点正式成为管理节点。

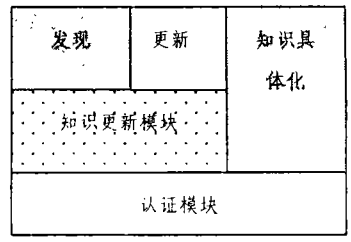

图 6 终端节点结构

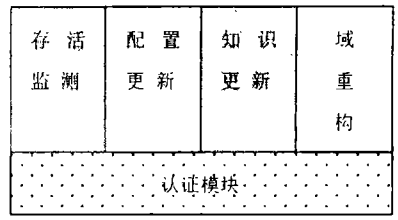

图 7 守护节点结构

当管理节点重新选出后，它将通知各子节点这一事件并告知其 IP 地址。

3 系统核心实现

为了验证框架的可行性，我们实现了一个叫 Vortex 的系统。目前管理了三种入侵检测的子系统：分布式 Snort、洪水攻击检测和扫描检测。分布式 Snort 是我们在 Snort 的基础上改善形成的系统，使其完成集中控制分布式检测的策略，能在不同的主机上分派检测规则集合。由于三种子系统都有各自不同的规则(检测知识)表达式，同时未来可能加入的子系统也允许具有任何的知识形式，这要求系统必须是开放的。此处主要给出管理节点知识更新部分的系统设计。我们利用了设计模式 SRATEGY[6]来实现插件的方式，即：将不同知识类型的新知识发现模块封装成 Strategy，利用面向对象方法的重载功能实现了管理节点的统一操作。对象图参见图 8。

作为不同的知识处理定义在 TKnowledgeProcessor 中，处理接口为 Process，子类通过重载 Process，封装了具体的处理策略，比如更新、抽取和具体化。TKnowledgeManager 分派具体知识的处理。其关键算法如下：1. TKnowledgeManager 收到提交的知识(Byte 数组)，将包格式中(见图 4)的 K_Type 字段的值作为知识类型参数传递给它管理的每一个 TKnowledgeProcessor 的子类。2. TKnowledgeProcessor 的子类在 Process 中判断传入的知识类型值是否等于自己处理的类型。如果是，则进行相关处理，否则退出。具体 TKnowle-

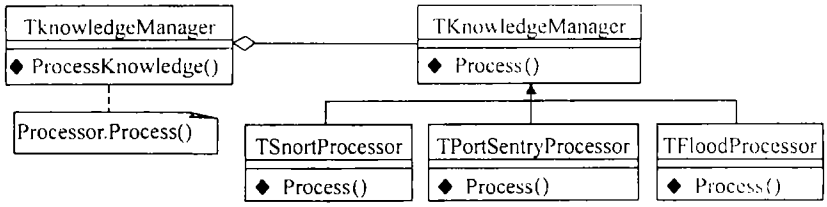

图 8　不同知识类型的统一处理模式

dgeProcessor 子类的实现可以用动态链接库，通过运行时加载，可以动态更新系统。如果用 Java 实现则是一种最自然的 OO 构建方式。以下用伪代码示(Object Pascal-like)示意模式中的关键部分：

```
TKnowledgeManager. ProcessKnowledge(Packet: String);
Begin
  Decode packet to get K_Type and the payload of knowledge;
  For I := 0 to Processors. Count - 1 do //Processors is a list of instances of subclass of TKnowledgeProcessor.
    Processors[I]. Process(K_Type, Knowledge);//different subclasses of TKnowledgeProcessor override method of Process.
End;
```

知识类型的 ID(K_Type)必须唯一，我们考虑可以采取两种方式：分散式管理(如 COM 中的 GUID)和集中式管理(如域名管理)。两种方式各有优缺点。

结论　在安全系统中安全知识(防火墙规则、入侵检测规则等)的及时更新是系统安全的重要保证。人工方式有诸多缺点，如果能以自优化的方式，形成一个知识社区(管理域节点形成的树)，则每一个社区中被管理的子系统都可以自动在社区中通告其新发现的知识(可能是管理员新加入的)。每一个子节点上的知识发现将在社区全局内形成共享。如此，系统将不断自优化和进化，保证了安全系统知识规则的及时有效更新。我们提出了入侵检测知识自优化的概念，给出了一个开放的、高伸缩性的自优化框架，并实现了一个实验性系统 Vortex，类似的工作在入侵检测领域尚未见报道。该框架实际上是一个通用的知识自优化框架，安全领域其它类型的知识，比如防火墙的规则也可纳入其中。未来的工作是在框架中加入评估，使得一个管理域可以评估新提交知识的有效性和重要性，让系统能以更有序的方式进化。

参 考 文 献

1　Waltz E. Information Warfare principles and operations. Artech House Boston . London

2　Kephart O, et al. Biologically inspired defenses against computer viruses. In: Proc. of IJCAI '95, 985-996, Montreal, Aug. 1995

3　www. gecities. com/researchtriangle/3372/firewall_farm. html

4　Kephart J O. A biologically inspired immune system for computers. In: R. A. Brooks, P. Maes, eds. Artificial Life IV. Proc. of the 4th Intl. Workshop on the Synthesis and Simulation of Living Systems, MIT Press 1994. 130～139

5　Communication in the Common Intrusion Detection Framework. http://www. isi. edu/gost/cidf/ drafts/communication. txt

6　Gamma E, Ralph Johnson Design Patterns: Elements of Reusable Object-Oriented Software

(上接第45页)

中的 a 表示系统给出判定为"yes"，而"yes"为正确判定的次数。

$$查全率=\frac{a}{a+c}\quad 查准率=\frac{a}{a+b}$$

对文本分类系统来说 a＋c 就是人工分入该类的文档数量，在封闭测试时，就是训练集中属于该类的文档数，成为机器归入文档数；列联表中的 a 表示机器正确地分入该类的文档数，成为正确归入文档数，因此对于文本分类系统来说查全率和查准率可定义为如下表示：

$$查准率=\frac{正确归入文档数}{机器归入文档数}*100\%$$

$$查全率=\frac{正确归入文档数}{应有文档数}*100\%$$

查准率和查全率这两个指标是相互矛盾的，有时为了提高系统的查准率就会使系统的查全率下降；为了提高系统的查全率，就会使系统的查准率下降，一个好的系统应该很好地兼顾这两个指标。

4.2　实验的设计及其描述

本文实验主要想验证仿人算法中所设计的计算聚类中心的新方法对分类器性能的影响。

实验的基准是：特征向量的特征项完全由单词组成，不包括任何短语，理想文献的计算方法用2中的式(2)，不对训练语料中的文本进行聚类分析；计算文本和理想文献的相似程度使用2中的公式(1)，本文所做的设计对分类器的影响通过查准率和查全率反映出来。

下表(表2)列出了本文对封闭语料和开放语料的实验结果：

表2　对封闭语料和开放语料的实验结果

	封闭语料		开放语料	
	查准率	查全率	查准率	查全率
基准	0.50	0.59	0.57	0.56
加入短语搭配之后	0.57	0.67	0.64	0.62
仿人算法	0.66	0.74	0.70	0.72

从实验的结果可以看出，把短语加入特征向量中之后，可以提高特征向量的聚类特性。本文设计的模仿人类专家的略读和跳读行为的文献聚类中心的计算方法，对系统性能有极大的提高，这给我们的一个启示是：探索人工分类的实质并把其应用到文献自动分类中去是一个系统是否能成功的关键。

参 考 文 献

1　康耀红著. 现代情报检索理论. 科学技术出版社，1990

2　朱兰娟. 中文文献的自动分类:[学位论文]. 上海交通大学，1987

3　曹素青，曾伏虎，等. 一个中文文本自动分类数学模型. 情报学报，1999，18(1)

4　张玥杰，姚天顺. 基于特征相关性的汉语文本自动分类模型的研究. 小型微型计算机系统，1998，19(8)

5　王永成，张坤. 中文文献自动分类研究. 情报学报，1997，16(5)

6　成颖，史九林. 自动分类研究现状与展望. 情报学报，1999，18(1)

7　Cohen, William W. Text Categorization and Relational Learning. Machine Learning. In: Proc. of the Twelfth Intl. Conf. (ML95)