基于多 Agent 协同的快速入侵检测系统*)

钱玉文　王　飞　孔建寿　王执铨
（南京理工大学自动化学院　南京 210094）

摘　要　在多 Agent 协同入侵检测系统中，不同检测 Agent 并行地检测网络包中不同的入侵特征，以提高系统的检测效率。使用消息、自定义通信协议等作为系统的协同通信机制，有效地避免了系统中的单点故障，并且，该机制使得各个 Agent 的检测结果可以有效融合。在分析了入侵的类型、特征后，使用 4 个检测 Agent 仿真了入侵检测的过程，并在检测精度、检测误差影响很小的情况下，使检测每条记录的时间大幅度减少。

关键词　多 Agent，入侵检测，协同

Fast Intrusion Detection Based on Muti-Agent

Qian Yu-wen　WANG Fei　KONG Jian-shou　WANG Zhi-quan
(School of Automation, NanJing University of Science and Technology, Nanjing 210094, China)

Abstract　In intrusion detection system based on multi-Agent, network packets were detected simultaneously by different Agents, which improved the efficiency of the system. Through using of the message mechanism, custom communication protocol mechanism, and other mechanisms, failure of single point was avoideds, and the detection results of different Agents were effectively integrated. In this research, four Agents were used to simulates intrusion detection based on multi-Agent, and the result was given which is that the time of detection was drastically reduced while accuracy and errors of detection were influenced little.

Keywords　Multi-Agent, Intrusion detection, Co-operation

1　引言

入侵检测系统(Intrusion Detection System, IDS)通过收集网络中的若干关键结点的信息，来检查其中是否有违反安全策略的行为和收集其中是否有遭到袭击的迹象，从而提供对内、外部攻击和误操作的实时保护[1]。然而，高速网络技术对入侵检测系统的效率提出了新的要求，如千兆以太网等的迅速发展，目前高速网络设备的背板交换能力已达到数 Gbps 或数十 Gbps，甚至上百 Gbps[2]。对网络上的实时入侵检测而言，单位时间内的数据流量越大，检测的工作量也越大。所以，必须对入侵检测系统的体系结构与检测方法进行改进和优化，以提高入侵检测系统的检测效率。

为了提高入侵检测系统的性能，Staniford 等人[3]提出了 CIDF(Common Intrusion Detection Framework)结构，该结构试图给出一个各种入侵检测系统之间相互协作的统一框架。但在这种系统中，中心检测部件的工作量依然很大，影响了整个系统的效率。文献[2]提出了负载平衡的方法来改善检测器的速度。然而，这种方法只针对字符匹配检测器有效，不能检测异常入侵。为了提高中心检测器的效率，大量研究工作集中在使用多代理来分解检测器的任务。具有代表性的工作是 Coast 实验室的研究人员提出的 AAFID 模型[4]和 Neumann 等人研发的 Emerald 系统[5]，它们在提高检测精度、降低系统误报率上等做了大量工作。然而，这些系统(或模型)在提高系统工作效率以及检测效率上的研究却较少。

为了有效提高入侵检测的性能，本文引入基于多 Agent 协同的入侵检测系统(以下简称 MAIDS)。由于 AAFID 系统使用 TCP/IP 协议传递消息的效率并不高，MAIDS 引入驱动层通信协议，提高 Agent 之间的传输速率；使用 Agent 消息机制实现多 Agent 推举、多 Agent 协商等协同行为，确保检测系统稳定地运行。系统在数据接受器上将接收到的高维网络数据分量，划分成多个低维数据分量，然后将这些低维数据分量分发到不同的检测代理(Detection Agent)上，利用协同机制使这些 Agent 并行地检测，以提高系统的检测效率。

2　基于 MAS 的入侵检测系统模型

2.1　基于 MAS 的 IDS 的体系结构

在局域网中，通过在交换设备可使网络数据包从 Internet 进入局域网时，定向到不同检测 Agent 的数据收集器上，这些检测 Agent 分布在计算机网路的不同主机上，这构成了基于多 Agent 的入侵检测系统的基础，其原理如图 1 所示。

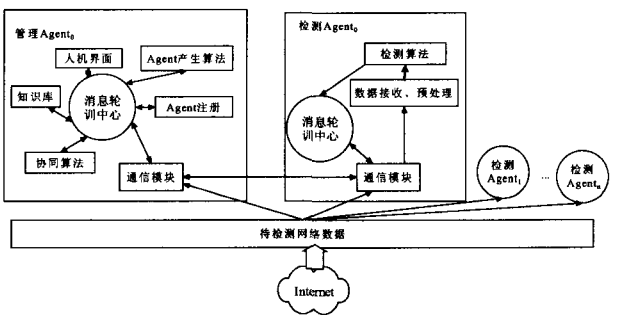

图 1　快速入侵检测系统构架

在图 1 中，基于 MAS 的入侵检测系统由多个入侵检测

*)本文受国家自然科学基金(60574082)支助。

Agent、管理 Agent 以及各种交换设备组成。每个检测 Agent 又可以分成入侵检测组件、通信组件和网络包采集组件等功能模块，都是自治的独立整体[6]。当网络包从 Internet 进入局域网络后，各个检测 Agent 不管该网络包目标地址是否是该 Agent 所在计算机地址，都将它保存下来。这样，所有检测 Agent 都可以得到网络中所有的包，并且所有 Agent 从网络上采集的包是相同的。然而，这些检测 Agent 对网络包中的不同行为特征（或者状态特征）敏感，而对其它特征不敏感。这就类似人体的感知系统，视觉器官能感受光线的特征，听觉器官可以感受声音特征。而这些特征需要经过数据预处理来获取。

2.2 Agent 的功能

管理 Agent 的目的是提供一些公共的信息服务，协同各个检测 Agent，主要的组件功能描述如下：

(1)通信组件：负责与其他 Agent 之间的可靠数据通信，这是基于消息的一种通信方式。通信模块还负责与本地浏览器等其他软件的通信。

(2)消息组件：负责存放和执行其他 Agent 传递过来的消息。

(3)协同控制组件：控制用户 Agent 之间的同步，解决并发冲突。

(4)知识库：存放共享信息，是指导与其他群体成员协同工作的核心，也是进行智能活动的依据。知识库的语法规则格式是：if 〈messages〉 then actions。

(5)数据库：存放系统运行参数、Agent 标识列表。

检测 Agent 的目的是为了接收网络数据并对其检测，它的各部分组件及其功能描述如下：

(1)通信组件：负责与协同 Agent 和本地浏览器的通信。

(2)消息组件：同协同 Agent 一样存放和执行其他 Agent 传送过来的消息。

(3)控制组件：协调各组件之间的操作。

(4)数据接收器及预处理组件：截获网络数据并对这些数据进行预处理。

(5)智能检测组件：调用 Bayes 决策算法对网络数据的不同维进行检测。

2.3 多 Agent 系统中的通信机制

多 Agent 协作是一种集体行为，指一个 Agent 在采取行动或做决定时，要受到别的 Agent 的存在或知识的影响[12]。多 Agent 系统中，协作则是通过进行 Agent 通信语言的交流来进行。这些 Agent 之间的通信语言独立于所有 Agent。MAIDS 采用的通信机制包括两个方面：一是定义一组能够使各个 Agent 协同起来的消息；二是实现系统中各个 Agent 之间的消息传输机制。

Agent 之间的通信采用基于消息的通信机制，本文利用消息完成服务请求和数据传递的任务。消息的原语主要有：inform，request，refuse，offer，accept 等。消息由一个消息名和一组消息参数组成，其定义可描述如下[13]：

```
Struct Message {
Agent ID SourceAgent;/ / 消息的来源 Agent 的名称；
Agent IP SourceAgent IP;/ / 源 Agent 所在机的 IP 地址；
Agent ID TargetAgent;/ / 消息的目的地 Agent 的名称；
Agent IP TargetAgent IP/ / 目的 Agent 所在机的 IP 地址；
Unsigned MessageType;/ / 消息的类型；
Cstring MessName;/ / 消息的名称；
Cstring MessCont;/ / 消息的内容；
Cstring MessPriority;// 消息中的动作被执行的优先级；
Tduration ActionTime;/ / 消息中的动作被执行的时间要求；
…}
```

在系统中，管理 Agent 和各个检测 Agent 通过发送、轮询收到的各类消息实现 Agent 协同工作。具有代表性的两个消息是 Agent 状态告知消息（tell）和推举管理 Agent 消息（elect）。各个 Agent 每隔一段时间 tell 消息告知对方自己的状态，一旦发现有某一个 Agent 不活动，便通知管理 Agent 重新创建一个与失效 Agent 相同的 Agent。另外，管理 Agent 是系统重要的部件，它影响系统的正常运行。各个 Agent 定期检测管理 Agent 是否活动，如果管理 Agent 不活动，则使用 elect 消息在活动的所有 Agent 中推选出一个性能最好的 Agent 作为管理 Agent。因此，系统中的每个 Agent 具有管理 Agent 的全部功能，但只有被推选者才可以作为管理 Agent。

由于 Agent 可以分布在同一台主机或者不同主机上，消息传输机制由本机进程通信以及网络通信两部分组成。为了重用代码，许多多 Agent 通信的工作使用 Socket 通信机制来实现[12]。在同一台主机上使用 Socket 进行通信，其实质是用硬盘上的临时文件作为数据交换区，这会降低系统的效率。在网络通信中，使用 Socket 通信，效率也不高，这是因为被传输的数据频繁地从核态空间拷贝到目态空间。为了提高系统的效率，在 MAIDS 中，开发了自定义的协议。在该协议中，在本机通信中采用内存共享的方法实现，这样避免内存与硬盘之间的拷贝；在网络通信中采用基于网络驱动中间层（NDIS）通信技术[13]，避免大量数据在内存中的频繁拷贝。在 NDIS 的 Miniport 接口进行数据发送，而在 NDIS 的 protocol 接口进行数据截获。Miniport 网卡和驱动层的接口，Protocol 是驱动层与协议层的接口。为了实现数据的稳定传输，通信中使用类似于 TCP 协议的三次握手机制。

2.4 基于多 Agent 协同检测算法

Agent 是一个自治的个体，它们可以分布在不同的主机上。每个 Agent 都采用行为检测的方法，而贝叶斯检测算法在样本足够的条件下，对异常行为具有较强的检测能力。因而，本文引入 Bayes 检测算法作为检测 Agent 的检测算法。

贝叶斯推理提供一种概率的推理手段，这对于入侵的近似判定十分合适，其原理如式(1)所示[14]。

$$V_{nb}=\underset{v_j\in V}{\operatorname{argmax}}P(v_j)\prod_i P(a_i|v_j) \tag{1}$$

在上式中，检测、训练每条数据都需要计算各个属性之间的相关性，所以使用贝叶斯分类器分类的效率很低。而在数据样本的子向量之间相关性较小的条件下，可将样本划分多个低维子向量，并将这些子向量分布到不同的计算节点上并行检测。这种并行的检测方法减少了每个检测节点的负担，又可避免计算高位向量的相关性，从而提高系统总体效率。于是，将网络包分成 N 个相对独立的数据分量，则就可以使用 n 个检测器分别对不同的分量进行检测，这 N 个检测器分布在 M 个 Agent 上。对于一个待识别入侵模式 $X=(x_1,x_2,\cdots,x_n)$，每个分类器的输出为 $Y_n(m=1,2,\cdots,N)$。这种模型用数学的方式描述如下：

$$Y=\begin{cases}0 & Y_1=Y_2=\cdots=Y_n=0\\ 1 & \text{else}\end{cases} \tag{2}$$

这样，可得到多分类器协同的决策算法，它类似于多个不同专家针对同一个事物不同方面分类进行决策，只要有一个专家认为是异常则可以断定是异常。类似地，多 Agent 协同

检测也就具有相同的机制，分布在不同 Agent 上的分类器针对不同的数据分量各自训练结束后，依据各自的识别性能确定其阈值，并分别对接收器送到的数据分量进行测试。如果有一个分量出现异常，即可判断出现入侵。

3 算法实现

3.1 数据预处理

为了有效地从网络数据包中提取特征，1998 年 DARPA 在 MIT 林肯实验室进行了一项入侵检测评估项目[9]。他们建立了模拟美国空军典型局域网的一个网络环境，仿真各种用户类型、各种不同的网络流量和攻击，使它就像一个真实的网络环境。他们将所有的 TCP/IP 网络包截获下来，事后将各个连接恢复出来，并对每一连接使用 41 个特征进行描述，从而形成了一个具有 41 维特征的数据集合。目前，关于许多入侵检测的研究工作是对这 41 个特征进行分析，例如文献[7]使用聚类分析的方法发现了 DOS 攻击的知识，文献[8]使用支持向量机的形式发现了不同的攻击方式。

在这些研究中，大量时间耗费在计算属性集内部所有特征的相关性上。然而，有些入侵无需所有的特征来判决，例如 DDOS 攻击只和流量的特性有关而与其它特性相关性很小。因而，有必要将整个数据集划分成几个相关性不大的子属性集，每个子属性集包含对应的子特征向量。入侵数据集合划分的方法有多种，最具代表性的工作是 Lee 等人采用数据挖掘的方法，将 41 个属性分成 4 个低维向量[11]。将这些子属性集分布到不同的检测 Agent 上并行地训练和检测，会提高系统训练和检测的速度。于是，系统预处理工作就是将待检测的数据向量 $X=(x_1,x_2,\cdots,x_{41})$ 分成 m 个子向量 $X_1,X_2,X_3,\cdots,X_m$（其中 $m\leqslant 41$），分别作为检测 Agent 检测输入。因此，各个 Agent 上需要不同的低维数据集合作为其训练集。事实上，各个检测 Agent 在训练和检测时，保留其能够检测的各维数据，而抛弃无需检测的各维数据。通过以上的预处理工作，各个 Agent 获取了训练数据，可以通过协同学习的方法来训练与检测入侵数据。

3.2 检测算法实现

入侵检测是一个二值分类问题，即正常与异常。对于样本空间中的每一个样本 X，均可以从中提取出感兴趣的一组特征集 F，定义 C 为分类集，即{正常，入侵}，令 $C1$ 表示正常，$C2$ 表示异常，它是一个随机变量。本文的目标是：在获得给定样本中的特征集后，判别出该样本是正常数据或是入侵数据的概率，即计算出 $P(C|F)$。实验中使用的具体算法如下：

(1)各个 Agent 上具有的训练、检测算法

训练算法

Step1　对训练集中全部样本的特征进行提取；

Step2　统计各特征量出现的频率；

Step3　得到不同分类器上分类规则。

分类算法

step1　提取出特征向量；

step2　计算：$\mathrm{argmax}(P_{NB}(C)P_{NB}(x|C))$；

step3　返回检测结果 cx，cx ∈ {正常，入侵}。

(2)不同 Agent 之间协同检测算法

Step1　由数据收集器收到从网络上获取网络包，并将高维向量分解成低维子向量；

Step2　数据收集器通过消息 MessageArrive 通知管理 Agent 有需要检测的数据到达；

Step3　数据收集器将需要检测的数据各个子向量分发给各个检测 Agent；

Step4　各个检测 Agent 上在驱动层得到数据，直接调用 Bayes 算法进行检测；

Step5　如果在检测过程中接收到其它 Agent 的入侵消息，退出检测；

Step6　如果检测出入侵，则通知入侵检测的管理模块并在网络中广播给其它 agent；

Step7　检测结果正常，退出检测。

4 仿真模型与结果分析

系统按照 Lee 提供的方法，将 DARPA CUP99 数据集划分成 4 个低维子向量，并设计了 5 个独立的检测 Agent，其中 4 个检测 Agent 对应地检测上述 4 个低维向量。这 4 个 Agent 是具有相互通信、Beyes 异常入侵检测和网络数据采集等功能的自治整体，剩下的一个 Agent 作为管理 Agent。实验在网络环境下实现，系统初始化时，这 5 个 Agent 分别配置到 5 个不同的网络主机上。在系统运行过程中，若某个检测 Agent 所在的主机出现故障，则管理 Agent 在正常的主机上创建相同的检测 Agent；若管理 Agent 所在机器出现故障，则从 4 个检测 Agent 中推举一个作为管理 Agent，并由当前的管理 Agent 再次创建一个检测 Agent。

在仿真实验中，完成如下工作：实验 1，分别从 DARPA CUP99 数据集中抽取 1 万、5 万、30 万和 137 万条数据进行了测试，确定检测速度与测试数据量、维数等的关系；实验 2，使用 Bayes 算法对 Darpa 数据检测，采用本文提出的多 Agent 算法对 Darpa 数据检测，比较各项检测指标。

在实验 1 中，首先对原数据集合进行训练与检测（即 41 维的 Darpa 数据集），将原数据集合使用 Lee 提供的方法划分成 4 个数据集，分别对应原数据集的不同子属性，使用 4 个 Agent 对这 4 个数据集合进行训练与检测。训练数据为 90%，测试数据为 10%。实验结果如图 2 所示。

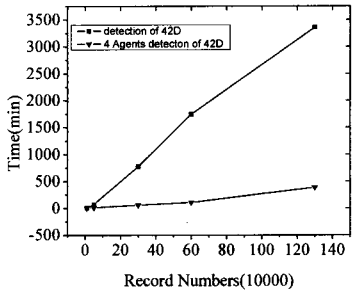

图 2　不同检测算法检测时间比较

检测时间与检测的样本数目并非是简单线性关系，这是因为数据越多消耗的系统资源就越多，从而检测的时间越长。另外，在对 137 万条数据的检测中，花费的时间为 3200 分钟，平均每一条数据的检测时间超过了 0.14 秒，这种检测速率对于高速计算机网络来说是远远不够的。然而，经过本文算法处理过后的算法对于数据的检测时间有了大幅度的提高，通过计算得知，经本文改进的多 Agent 检测算法对每条数据的检测时间约为 0.008 秒。

在实验 2 中，不使用 Agent 进行检测，而是使用一个普通的 Bayes 的检测算法，在设定阈值为 0.95 的条件下得到第一列的测试结果。对于各个 Agent 可以使用不同的阈值，来调

节各个检测参数。例如，在流量非常大的情况下可以将检测DDOS攻击的Agent的检测阈值适当降低，防止有误报发生，这样可以使用阈值来实现不同Agent的检测参数的调节。但在本实验中，为了和以前的工作比较，也使用了0.95的阈值。

表1 不同算法检测指标比较

检测方法	检测精度(%)	误报率(%)	漏报率(%)
非Agent算法	94.66	5.72	7.13
多Agent	94.01	6.87	12.11

由表1可知，单个主机对DARPA CUP99数据集中41维进行检测，检测的精度为94.66%，而多Agent进行检测准确率下降到94.01%。从各项指标可以看出，基于多Agent的检测方法检测效果要比使用高维数据进行检测的算法效果稍差。由此可知，使用MAS算法实现的多Agent协同入侵检测器大大地缩短了检测时间，缓解了网络流量对入侵检测带来的压力。

结束语 本文用MAS协同的方式同时对入侵的不同特征进行检测。协同入侵检测系统是一种全新的入侵检测系统，既可以有效地检测误用类型的入侵，又可以检测异常类型的检测。使用多Agent和并行机制，能够快速地、较为准确地对不同类型的入侵特征进行检测。这种提高了入侵检测效率的方法对用于检测多传感器的系统也有参考价值。由于将高网络数据向量划分成多个低维数据分量的相关性很小，使用多Agent协同检测方法不会对检测的精度产生较大影响。

参考文献

[1] Denning D E. An Intrusion Detection Model [J]. IEEE Transaction on Software Engineering, 1987, 13 (2): 2222-2321

[2] 李仁发，李红，喻飞，等. 入侵检测系统中负载均衡研究与仿真. 计算机仿真学报 2004, 16(7): 1444-14449

[3] Chen S, Cheung, et al. GrIDS: A Graph based Intrusion Detection System for Large Networks[C]// Proceeding of 19th National Information System Security Conference. 1996 (1): 361-370

[4] Barrus J. A Distributed Autonomous-agent Network-Intrusion-Detection and Response System// Procedings of the 1998

[5] Porras PA, Neumann PG. EMERALD: Event monitoring enabling responses to anomalous live disturbance [A]// Proceedings of the 20th National Information Systems Security Conference [C]. Baltimore Maryland USA, 1997: 353-365

[6] 朱永利，宋少群. 基于广域网和多智能体的自适应协调保护系统的研究. 中国电机工程学报, 2006, 26(16): 15-20

[7] 孙知信，唐益慰，张伟，等. 基于特征聚类的路由器异常流量过滤算法. 软件学报, 2006, 17(2): 295-304

[8] 肖云，韩崇昭，郑庆华，等. 一种基于多分类支持向量机的网络入侵检测方法. 西安交通大学学报, 2005, 39(6): 562-565

[9] http://kdd. ics. uci. edu/databases/kddcup99/kddcup. html. School of information and computer science university of California. KDD cup 1999 data [EB/OL]

[10] Lee W, Stolfo SJ, Mok KW. A data-mining frame-work for building intrusion detection models [A]// The 1999 IEEE Symposiumon Security and Privacy. Berkeley, USA, 1999

[11] 刘向军，刘世平，张洁，等. 多Agent系统通信与协作机制构造. 机械设计与制造工程, 2002, 31(2): 40-42

[12] 林杰，薛华成. 多agent协同工作模型在基于Web的群决策支持系统中的应用. 计算机集成制造系统, 2001, 17(6): 25-28

[13] 李明欣，余堃. 基于NDIS中间驱动的入侵检测. 计算机工程与设计, 2007, 28(1): 51-53

[14] 张波云，殷建平，蒿敬波，等. 基于多重朴素贝叶斯算法的未知病毒检测. 计算机工程, 2006, 32 (10): 11-13

(上接第50页)

系统吞吐量等相关跨层设计和性能分析。

参考文献

[1] Liu Q, Zhou S, Giannakis G B. Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links. IEEE Trans. Wireless Commun., 2004, 3(5): 1746-1755

[2] Liu Q, Zhou S, Giannakis G B. Queuing with adaptive modulation and coding over wireless links: Cross-layer analysis and design. IEEE Trans. Wireless Commun., 2005, 4(3): 1142-1153

[3] Wang X, Liu Q, Giannakis G B. Analyzing and Optimizing Adaptive Modulation Coding Jointly With ARQ for QoS-Guaranteed Traffic. IEEE Trans. Vehicular Technology, 2007, 56(2): 710-720

[4] Niyato D, Hossain E. A Queuing-theoretic and Optimization-based Model for Radio Resource Management in IEEE 802.16 Broadband Wireless Networks. IEEE Trans. On Computer, 2006, 55(1): 1478-1488

[5] IEEE Std 802.16a Local and Metropolitan Area Networks — Part 16. 2003

[6] Nakagami M. The m-distribution—A general formula of intensity distribution of rapid fading // Statistical Methods in Radio Wave Propagation. Oxford, UK: Pergamon, 1960: 3-36

[7] Biglieri E, Caire G, Taricco G. Limiting performance of blockfading channels with multiple antennas. IEEE Trans. Inf. Theory, 2001, 47(4): 1273-1289

[8] 3G PPTR 25.848 V4.0.0 Physical layer aspects of UTRA high speed downlink packet access (release 4). 2001

[9] Alouini M-S, Goldsmith A J. Adaptive modulation over Nakagami fading channels. Kluwer J. Wireless Commun., 2000, 13(1/2): 119-143

[10] Goeckel D L. Adaptive coding for time-varying channels using outdated fading estimates. IEEE Trans. Commun., 1999, 47: 844-855

[11] Goldsmith A J, Chua S-G. Adaptive coded modulation for fading channels. IEEE Trans. Commun., 1998, 46: 595-602

[12] Assaad M, Zeghlache D. Cross-layer design in HSDPA system to reduce the TCP effect. IEEE J. Sel. Areas Commun., 2006, 24(3): 614-625

[13] Maharshi A, Tong L, Swami A. Cross-layer designs of multi-channel reservation MAC under Rayleigh fading. IEEE Trans. Signal Process., 2003, 51(8): 2054-2067

[14] Kleinrock L. Queuing Systems vol. I. New York: Wiley, 1975