计算机科学1998 Vol.25 №.4

软件 Agent 的安全性研究*)

On Security for Software Agent

李 新 董 桓 丁俊华 吕 建

(南京大学计算机软件新技术国家重点实验室 计算机软件研究所 南京 210093)

摘 要 本文介绍 Internet 环境下 Agent 系统的安全性问题,首先概述 Agent 的概念及其特性,其次从四个方面剖析 Agent 系统的安全性,然后介绍我们的 Agent 系统中针对这四个方面问题提供的多种安全措施。

关键词 Agent,安全,Java

一、引言

随着 Internet 的飞速发展,网络上的资源日益增多,构成了广阔的信息空间(InfoSphere),但是分布式网络资源的一些固有特性阻碍了资源的有效利用。首先,网络资源未经组织,分布在世界各地的千万台主机上;第二,Internet 是动态变化的,资源的数量、类型和有效性都在不断变动;第三,同样的资源可以从不同的地方获取,由于信息源的变化,造成潜在的不一致性。而现今的网络应用基本上是基于静态主机之间同步或异步消息传递的远程过程调用(RPC)模式,网络上传递的主要是不能运行的消息,而可执行的程序仍然驻留在主机上,用户只能使用服务器上已预先设定好的服务。整个信息处理过程是被动的,不能根据用户的需求动态变化。所以,如何及时、有效地收集、整理和分析网络上的信息,利用这些资源,为人们的生产、生活和学习服务,逐渐成为新的课题。最早在人工智能领域中提出的 Agent 在这种背景下被赋予了新的内容,它作为用户授权的软件助理在 Internet 上的应用得到了广泛的研究。Agent 扩展了传统的静态模式,允许其在主机之间迁移,动态地适应新的环境,避免在不可靠的网络上进行远程交互和数据传输。利用 Agent 进行信息搜集、整理和电子商业交易等向人们展示了 Internet 应用和服务的新景观。尽管 Agent 有广阔的应用前景,但恶意的 Agent 和主机及不可靠的网络给安全性造成了很大威胁,如 Agent 占用大量资源、未经授权使用资源,用户窃取和篡改 Agent 中的数据、修改 Agent 的行为等等。Agent 真正要达到实用化,为人们所接受,必须解决 Intetnet 分布式环境下的安全问题。目前绝大多数试验系统都未能提供完善有效的安全措施。本文首先简要介绍 Agent 及其特性,然后分四个方面剖析 Agent 环境下的安全性,最后阐述我们用 Java 语言实现的 Agent 系统中的安全措施。

二、Agent 和安全性

1. Agent 定义和特性

Agent 一词现在被大量引用,不同领域的研究人员对 Agent 有不同的认识,分别给出了自己的定义和 Agent 性质的描述,离开特定的应用环境讨论 Agent 是很含糊的。Agent 的原意是一个人代表另一个人完成某件事,应用于计算机中,可认为 Agent 是相应用户授权的个人软件助理。人们根据自己的研究需要,在此基础上加入自己的见解。本文中所述的 Agent 可归结为软件 Agent 中的一种:流动 Agent (Mobile Agent),Gray 等的定义为:

流动 Agent 是一个执行程序,它能自主地通过异质网络从一台计算机迁移到另一台计算机。[1]

流动 Agent 一般应具有以下几个特性:

●自主性。Agent 是独立可运行的程序,能完成自身的目标,它根据自身所处的环境、本身状态及携带的数据和知识决定做什么,不需要用户经常干预。

●流动性。Agent 可以在任何时刻暂停执行,通过网络迁移到另一台计算机(以下称为站点)上继续运行。对于信息搜索和处理,流动性是强有力的支持,因为信息本身分布在不同的信息源,它们都有大量的信息,流动的 Agent 可以就近存取信息,而不是

*)本项研究受到国家杰出青年科学基金资助。
李 新 硕士生,研究方向为面向对象和软件 Agent。董 桓 硕士生,研究方向为面向对象和软件 Agent。丁俊华 博士生,研究方向为面向对象和软件 Agent。吕 建 教授,博士导师,研究方向为软件自动化,并行程序形式化方法,面向对象语言和环境等。

通过网络将信息传输到程序所在地再行处理。

●协作性。Agent 是独立但不孤立的程序，当它不能完全独立地完成某个任务时，应能通过某种通信机制和其它 Agent 或用户交互协作，能根据自身需要组织并发送消息给其它 Agent，也能理解和处理其它 Agent 发送的消息。现今的网络是人类社会的缩影，现实社会中人与人之间有着复杂的关系，协作是其中重要的一种。作为用户代表的 Agent 在网络环境中也要与代表其它用户的 Agent 或直接与人进行交互和协作。

●智能性。Agent 应具备学习功能，自身具备一定知识，并在与用户和其它 Agent 的交互中学习新知识。当迁徙到新的环境后，能自动适应外界环境的变化。

●安全性。对于恶意或有错的 Agent，站点应能够保护自己免遭破坏，同时 Agent 自身也要防范站点窃取机密数据，篡改代码和数据等不良行为，在诸如电子商贸和网络银行服务等应用中，安全便成为关键因素。

2. Agent 系统的安全性

我国安全专家缪道期将计算机系统安全定义为：计算机的硬件、软件和数据受到保护，不因偶然的和意外的原因而遭破坏、更改和显露，系统连续正常运行。据此，Agent 系统的安全就是保护运行 Agent 的主机、Agent 和 Agent 所代表的用户的权益不受恶意和错误的 Agent 和站点的攻击而能正常运行。Agent 环境下的安全问题可分解为四个相关方面：[3]

1. 保护站点免受 Agent 的破坏。Agent 到达站点后，站点应能对 Agent 验证身份，据此分配一定资源，防止 Agent 违反资源存取限制。

2. 保护 Agent 免受其它 Agent 的攻击。Agent 之间是平等关系，任意一个 Agent 不能干涉别的 Agent 运行。

3. 保护 Agent 免遭站点的破坏。恶意站点会篡取 Agent 所携带的关于用户隐私和机密信息，修改 Agent 的行为。

4. 保护一组主机。或许从单个站点来看，Agent 并无越轨行迹，但会给一群主机组成的网络造成危害。如 Agent 不断复制自身并在主机间来回流动，消耗大量资源，最终使网络瘫痪。

第二点其实是第一点的特例，只要保证站点的安全，不同的 Agent 有各自独立的地址空间，那么 Agent 之间不可能互相倾轧。对于第四点可采取简单的被动措施，如限制每个站点上可运行 Agent 的总数，若超过限量，则不接受新的 Agent。文[3]中提出了基于货币(Currency-based)的资源分配策略：每个 Agent 拥有有限货币，每个货币单位只能使用一次，Agent 使用货币存取资源，并且和它生成的子 Agent 共分所有的货币。这样 Agent 和它的子 Agent 就会因货币耗尽而终止运行。第一和第三点目前研究得较多，下面分别介绍。

3. 主机安全

根据 Agent 所代表的用户与站点用户的关系可将 Agent 分为两类：信赖的(Trusted)和不信赖的(Untrusted)。一般将本地的 Agent、站点用户认为可靠熟人的 Agent 作为信赖的，通过网络传输而来的 Agent 认为是不信赖的。区分的方法是身份验证，如传统的用户名加口令，通过身份验证的 Agent 拥有相应的资源存取权，否则拒绝服务。

这种方法有人称为硬性安全，意指 Agent 的可信度和存取权限在执行前就已确定[2]，它有两点不足：①信赖的 Agent 中可能会有错误，如死循环、误删文件等，仍然威胁站点安全。②在多数情况下，不可信赖的 Agent 并不意味着一定是恶意的，它们需要站点提供服务，但并不威胁其安全。对于 Internet 服务提供者(Internet Service Provider)来说，显然这种 Agent 更为重要。一概拒绝这些 Agent，大大缩小了服务范围。如果给不可靠的 Agent 一些受限制的资源，使其运行，直到结束或有不正当的举动时终止其运行，则能很好地解决这个问题。和硬性安全相对应，这种方法称为柔性安全(Soft Security)，关键是动态(即运行时刻)监视和控制 Agent，所用方法是入侵检测系统(Intrusion Detection System)，即在 Agent 行将对系统安全造成破坏的时候检测出来并采取相应措施。在我们设计的 Agent 系统中，利用 Java 语言提供的设施，提供了比较完善的动态监控措施，在第三节将详细介绍。

4. Agent 的安全

站点不信赖 Agent，同样 Agent 也不信赖站点。既然有恶意的 Agent，也就会有恶意的站点。如用户指派 Agent 去查询两家航空公司的机票，并订购一家较便宜的。当 Agent 已查到了第一家的情况，流动到第二家时，站点会将 Agent 中有关第一家的数据改掉，如提高票价，Agent 就会产生假象，以为第二家较便宜。更糟糕的是，本来 Agent 订一张票，站点修改 Agent 的行为，订了许多张，给用户造成损失。

显然，禁止站点对其上的 Agent 进行破坏几乎是不可能的，只能采取电子签名、PGP 和密码等技术，将机要部分加密，使得站点不能轻易修改数据和代码，并且在篡改过后的 Agent 流动到新的站点后，立即被发现并终止其运行[4]。Agent 不能将重要数据

以非加密形式经过不信赖的站点，重要数据的加密、解密和处理只能在信赖的主机上进行。此外，还要建立第三方的认证设施，使得 Agent 无法抵赖曾经发送消息给某个 Agent 或站点，Agent 无法否认曾经收到其它 Agent 的消息[6]，这在商务活动中是至关重要的。

三、我们的 Agent 系统中的安全措施

我们设计了一个基于 Internet 的流动 Agent 系统，始终重视 Agent 系统的安全性，使用安全的 Java 语言构造系统和用户 Agent，针对上述四个方面的问题，并采取了多种安全设施：通过识别与认证方法区分信赖关系；灵活、完备的访问控制对 Agent 进行严密监控以保护站点；Agent 则可采用密码技术保护自身。

1. 识别与认证

站点只能给信赖的 Agent 充分的资源。Agent 也只能在信赖的站点上对重要的数据进行加密、解密和处理。因此，Agent 和站点之间互相识别与验证显得十分重要，每个站点有一些信赖用户标识和密钥，凡是通过检验的 Agent 被认为是信赖的，赋予相应用户权限，没有通过的，作为不信赖的 Agent 仅给予少量受限资源；每个 Agent 也知道一些信赖站点和密钥，在流动到一个站点时，不仅要接受站点的身份验证，也要对站点进行验证，确信是否为信赖站点。推而广之，在 Agent 之间的交互，站点之间的合作中都将会进行身份验证。

2. 访问控制

我们的 Agent 系统采用柔性安全策略。为保证站点的安全，必须对开放给 Agent 的资源的存取权利加以监控，一旦出现危及安全的操作，立即终止 Agent 的运行。表1对计算机系统中典型的资源进行分类，其中"X"表示每类资源可能会受到的攻击[8]：

表1

资源	泄露		可用性		完整性		骚扰性	
文件系统	X	√	X	√	X	√	X	√
网络	X	√					X	√
随机存储器	X	√	X	X	X	√	X	X
输出设备							X	X
输入设备	X	√	X	√			X	√
过程控制			X	√			X	√
用户环境	X	√			X	√	X	√
系统调用	X	√	X	√	X	√	X	√

其中泄露：泄漏用户和站点的信息；可用性：受攻击时使合法的用户也得不到资源，如占据大量的磁盘空间；完整性：受攻击时损坏或修改数据；骚扰性：骚扰性攻击，如显示许多窗口，将屏幕遮蔽。

我们用 Java 语言较好地实现了访问控制。

(1)安全的 Java 语言　Java 语言较传统的程序设计语言从设计上更多地考虑了网络环境下的安全因素，提供多级安全设施[9~10]：编译器安全特性；运行时刻安全机制：Class Loader 安全检查；Bytecode 验证；存储管理和控制；Security Manager 检查。

(2)定制安全策略　java.lang 包中 SecurityManager 提供了一致和可控制的访问控制方法。这个类中有许多 checkX 方法，其中 X 代表操作，如 delete、connect 等，在 Java 程序即将执行某些涉及到系统安全的操作时，相关 check 方法被调用，判断是否违反访问控制限制。如 Agent 中调用 java.io 包 File 类中的 public boolean delete()方法删除文件时，只要在调用前设置了系统的 Security Manager，那么这时它的 checkDelete(String file)方法就会被调用，参数正是要删除文件的名。用户可重载此方法，制定自己的安全控制策略：首先确定欲删除文件的 Agent，然后查出对应的用户，再根据用户判断是否具有此权利，若允许则返回，否则引发安全异常。表1中"√"表明了采用这种访问控制方法后系统中大多数资源受到了很好的保护。

3. 密码技术

Agent 自身携带的数据和通信时发送的消息在需要的时候都可以加密，Java 类库中提供了多种加密方法。用户编制 Agent 时，可将 Agent 分为几个部分，对其中重要部分加密，只有到达信赖站点时，才对重要部分进行处理。

结论　本文首先分析 Agent 技术，从四个方面剖析 Agent 系统中的安全性问题，然后详细介绍我们设计的基于 Internet 的 Agent 系统中采取的多种方法，较好地实现了安全性，并在此基础上进行了 Internet 网上信息收集和数据库查询的实验[11]。我们的系统尚存在一些不足之处，如内存对象、输出设备保护不够等等，还需大量的探索、研究和实验，有待今后工作进一步完善。

参 考 文 献

[1]Daniela Rus et al.,Transportalbe Information Agents, 1996

[2]Andreas Rasmusson Sverker Jansson,Personal Security Assitance for Secure Internet Commerce, Swedish Institute of Computer Science Apr 1996

[3]Rober S. Gray, Agent Tcl: A flexible and secure mobile-agent system,1996

[4]Peter Wayner, Agents Unleashed: A public domain look at agent technology,1995

[5]William M. Farmer et al.,Security for Mobile Agents: Authentication and State Appraisal

[6]N. Asokan et al.,Server-Supported Signatures

[7]龚波，郑若忠，Internet 安全及三种技术，国防科技大学

[8]Joseph A. Bank, Java Security, jbank@mit.edu, 1995

[9]Sun Microsystems, HotJave(TM): The Security Story

[10]Jamie Jaworski, Sams Net, Java Developer's Guide

[11]孙圣强，一种软件 Agent 模型及其实现技术的研究，南京大学计算机系硕士毕业论文，1997