移动 Agent 系统 JADE－S 的研究与改进

黄科文，郑洪源，丁秋林
（南京航空航天大学 信息科学与技术学院，江苏 南京 210016）

摘 要：提出了使用签名 jar 文件以及分离私钥和所属 Agent 的策略来建立一种安全的移动代理通信通道的方法。移动代理的安全性问题是目前研究十分广泛的一个复杂问题。文中对多 Agent 系统的安全性问题进行了全面的分析，并结合移动 Agent 平台 JADE－S，深入研究了该平台的特点和安全方案，指出了它存在的安全缺陷。在此基础上，提出了采用签名 jar 文件以及分离私钥和所属 Agent 的策略来完善 JADE－S 的安全性，最后指出了下一步工作重点。

关键词：安全；移动代理；反射；JADE－S
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1673－629X(2008)03－0165－04

Research and Improvement on JADE－S of Mobile Agent

HUANG Ke-wen, ZHENG Hong-yuan, DING Qiu-lin
(College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: The security of mobile agent system is one of the hot spots in recent research. In this paper, have a comprehensive analysis of multi－agent system's security problem and combined with mobile agent platform JADE－S, make a further study on the features and security architecture of this platform. Then, point out its security flaw. Based on this, put forward a method by applying signed jar file and detaching the private key from its affiliated agent to consummate JADE－S security. At last, point out the next key emphasis in work.

Key words: security; mobile agent; reflection; JADE－S

0 引 言

移动 Agent 技术能够有效地提高网络应用性能、减少网络流量、提高网上资源利用率，具有广泛的应用前景。随着人工智能和计算机网络的飞速发展，对 Agent 研究的深入，基于 Agent 和移动 Agent 系统的开发平台及应用也不断涌现，然而，Agent 系统的安全问题始终是阻碍它广泛应用的一个因素。

安全就是要保证数据的保密性、完整性、可靠性、可用性、抗抵赖性等，保证系统的正常运作。移动 Agent 的安全性，由于它自身的特殊性，使得要确保它的安全性变得更为复杂。显然，移动 Agent 系统能否广泛应用，这些安全问题的能否有效解决是决定因素之一。

文中研究了移动 Agent 平台面临的安全性问题，并对目前较为流行的一个移动 Agent 平台 JADE 以及它的安全扩展 JADE－S 进行了介绍和分析，指出了其中的不足并提出了一些可行的改进思路。

收稿日期：2007－06－08

作者简介：黄科文(1982－)，男，江苏张家港人，硕士研究生，研究方向为信息安全、人机交互；丁秋林，教授，博士生导师，研究方向为 CAD/CAM/MIS/CIMS。

1 移动 Agent 的安全性问题研究

在移动 Agent 系统中，Agent 的运行需要分布式系统中的宿主程序为其提供执行环境。机器的所有者、宿主程序的用户、Agent 所代表的用户以及软件的开发者都是不同的实体，这样必然会存在安全隐患。例如，宿主可能会被恶意的 Agent 程序侵入，类似于病毒或木马程序的 Agent 会泄露或毁坏敏感信息，侵占宿主的资源，从而干扰宿主的正常工作。反之，恶意的宿主也可能控制移动 Agent 的执行，并从中窃取敏感信息。因此，安全性问题是移动 Agent 系统中需要解决的最重要问题之一。

现在针对移动 Agent 安全的研究有基于移动代理平台(MAP, Mobile Agent Platform)和移动 Agent 模型的描述，也有一些具体的模型部分实现，比如：Aglet，JADE 等。通过对移动 Agent 模型框架(如图 1 所示)及应用场景的研究分析，移动 Agent 系统安全问题可以分为以下四类[1]：

1) 移动 Agent 平台遭受恶意 Agent 攻击(Agent→

Agent platform)。恶意 Agent 伪装成合法授权 Agent 进入 Agent Platform,提出没有意义的要求,干扰 MAP 正常工作。严重些可以导致 MAP 的信息泄露、被修改,指令代码等机密资料外泄。恶意 Agent 还可以根据所取得的权限关闭机器或者关闭 MAP。

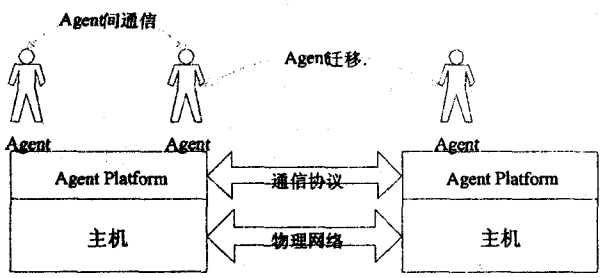

图 1 移动 Agent 模型框架

2) 移动 Agent 遭受其他 Agent 平台攻击(Agent←Agent platform)。移动 Agent 迁移到恶意 MAP,恶意 MAP 可以提取 Agent 的敏感信息,导致信息泄漏,还能干扰、修改 Agent 的代码或者状态,让 Agent 变成恶意 Agent 或改变 Agent 的计算结果。MAP 还可以拒绝 Agent 提出的服务要求甚至直接中断 Agent 继续执行。MAP 可以修改 Agent 与其它 Agent 的通讯,可以引诱更多 Agent 迁移过来。

3) 移动 Agent 遭受恶意 Agent 攻击(Agent→Agent)。Agent 在 MAP 上攻击其他的 Agent,它导致的危害很多,如:妨碍其他 Agent 活动;监听 Agent 与其他实体(Agent 或者 Agent platform)的通讯;对其他 Agent 的要求做出不正确的回应;伪装成其他 Agent 或中间人进而获得 Agent 要传给其他 Agent 或者 Host Platform 的敏感信息;恶意 Agent 也可以不停地发送垃圾信息给其他 Agent 导致其他 Agent 无法正常通讯和正常执行;如果 Agent platform 的安全性不高,恶意的 Agent 可以通过特殊手段存取或修改其他 Agent 的代码或者状态,利用 Agent platform 的漏洞干扰其他 Agent(比如:缓冲区溢出或把 Agent 的状态置为初值)。

4) 移动 Agent 系统遭受迁移等其他安全问题(Agent system←other Agent platform entities),主要是在迁移过程中遭受窃取、修改和重放等安全威胁,或者破坏或者干扰 Agent 之间或 Agent 与 platform 之间的通讯,重放、伪造和窃取信息。还有就是 Agent 系统的拒绝服务攻击(Agent system deny of services)。

目前移动 Agent 系统基本上都采用了解释语言(如 Tcl、Java)来实现,MAP 的部分安全问题可以由解释程序来解决,解释程序可以检查移动 Agent 执行的地址是否越界、检测系统请求的参数以及维护访问权限表等等。

2 移动 Agent 平台——JADE

JADE(Java Agent Development Environment)[2]是用 Java 编写的一个移动 Agent 系统,可以用来开发基于 Agent 的应用。JADE 遵循 FIPA(Foundation for Intelligent Physical Agents)规范,能实现与其他 FIPA 移动 Agent 系统间的互操作。

每一个运行的 JADE 执行期环境的实例被称为容器,因为它包含若干个 Agent,是 Agent 运行的环境。一个单独的特殊的主容器必须处于激活状态,所有其他容器都必须向主容器注册。主容器也称为 Agent 平台。主容器除了承担其它容器注册功能外,主容器区别于其他容器的特征在于它有两个特殊的 Agent:AMS(Agent Management System)负责控制平台内 Agent 的活动及外部应用程序与平台的交互;DF(Directory Facilitator)负责对平台内的 Agent 提供黄页服务,通过它 Agent 可以发现其它 Agent 及该 Agent 所需要的服务。图 2 是 JADE 平台的标准模型。在容器中,Agent 之间的交互都要通过信息传输系统,也称 ACC(Agent Communication Channel),控制平台内或不同平台之间的消息传输,它们的管理必须由主容器中的 AMS 完成,享受服务也必须通过主容器中的 DF,这些都是通过各容器之间的 MTS(Message Transport System)完成的。

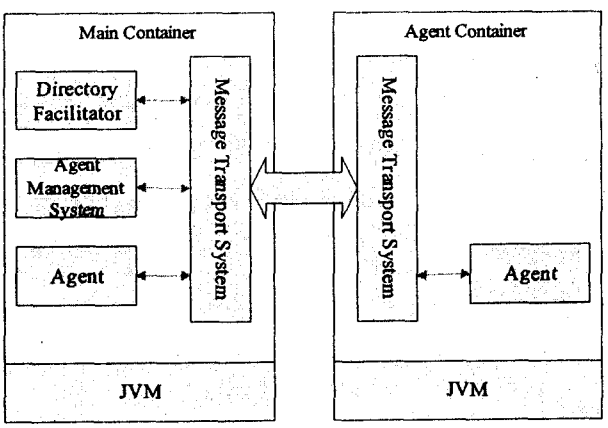

图 2 JADE 平台标准模型

Agent 在 JADE 中是作为一种自治的具有合作和通信能力的实体。为了安全 JADE 规定外部程序不能获得 Agent 的引用即不能直接存取 Agent 的属性。也不能直接指定 Agent 的行为。创建 Agent 的具体任务只能由容器来完成。返回结果也只是封装后的 Agent。JADE 平台提供了一些管理 Agent。如 AMS,DF,RMA(Remote Monitoring Agent)等。其中 RMA 提供了一个主要的系统管理界面。用户可以通过它观察到平台内的具体情况。

实际上,JADE 平台只提供了 Agent 的一个运行环

境,保证在其上的Agent能够正常运行,没有专门以维护系统安全出现的实体。虽然JADE平台没有直接集成安全实体,但是JADE提供了一个安全扩展——JADE-S。

3 JADE-S的安全性分析

为了满足一些JADE开发者的安全需求,JADE提供了安全扩展JADE-S[3]。它是一个JADE可选的能够提供一些安全保障的附件(JADE-S add-ons),在JADE的官方网站上可以下载到。

3.1 JADE-S

JADE-S是一个JADE的安全扩展,它给平台提供一些安全特性。当前的最新安全版本2完全替换了版本1。它在自定义砂箱(sandbox)的基础上扩展了Java安全模型,使用了下面的安全特性来满足移动Agent安全的需求:

* 用户鉴定(可靠性);
* 对Agent执行动作的授权(可用性);
* 消息的数字签名(完整性、抗抵赖性);
* 信息的加密(保密性)。

为了实现这几个安全特性,JADE-S提供了四个核心服务:

①安全服务(Security Service);

②许可证服务(Permission Service);

③数字签名服务(Signature Service);

④加密服务(Encryption Service)。

安全服务是JADE-S的一个基础服务,而其他的服务可以是有选择性构筑在这个基础服务之上,这三个服务之间是相互独立的。

我们知道JADE平台上的Agent容器通常分布在不同的网络主机上。在这样一个开放、分布的环境下为了能够安全有效地把它们关联,JADE-S引进了这样一个系统——多用户系统(multi-user system),在这个系统中,如果用户被平台超级管理员授权执行合适可靠的特权代码时,这个用户就能拥有一个平台上的所有的构件(包括Agents、容器)的访问权限。而且,每个Agent都会拥有一对自己的公私钥对用来加密和签名数据。

图3给出了JADE-S的体系结构[4]。一个JADE平台只有一个主容器,它有权进入密码仓库(password store),类似一个本地存储密码的文件或者一个远程LDAP服务器,并且拥有Agent平台许可文件(platform permission file)。然而,一个普通的容器(非主容器)只能有它本地的容器许可文件(container permission file)。每一个Agent都携带一对密钥对(公钥和私钥)和一个能够证明自己合法身份的证书。这个证书,由能够证明证书的有效性的认证权威AMS颁布,包含Agnet的内部名称和拥有者,容器和Agents的拥有者对应了某一个用户。

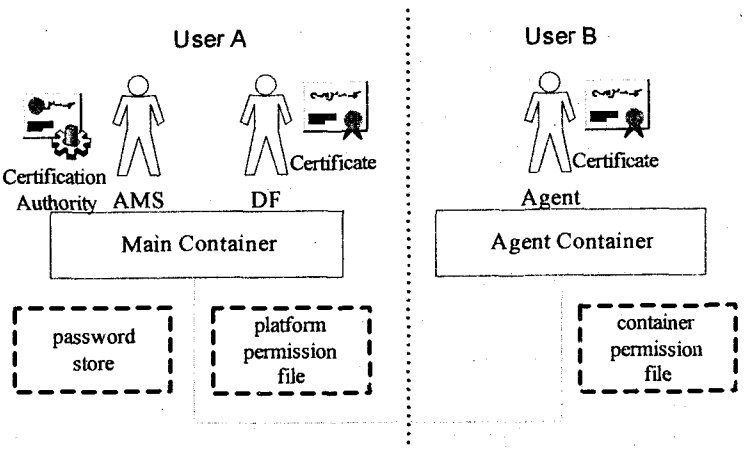

图3 JADE-S体系结构

3.2 JADE-S的不足

JADE-S提供了用户和Agents鉴定,权限管理,以及信息加密和签名,解决了一些JADE的安全问题。但是这些都是静态的安全措施,移动Agent的移动性和自主性的特征,使得它的安全问题变得很复杂,它需要一些动态的安全措施来完善JADE的安全性,JADE-S还存在一定的缺陷:

首先,JADE和JADE-S依赖于JVM(Java Virtual Machine)。虽然JVM与JADE没有直接的联系(JADE用Java开发,执行需要JVM支持),但是,在一个JVM上执行多个程序有可能会带来安全问题(其中有线程安全问题,Servlet就存在线程不安全问题)。

Java具有一个关键的性质——反射(Reflection),它是动态的(或准动态)。JADE就是利用了Java这一性质使得装载和执行Agents变得相对容易。JADE得到了反射便利性的同时,也给恶意用户留下了入口,使它们可以利用这个来侦查Agents,并且进行造假。这也是JADE-S还没有涉入安全保护内容的一个缺陷。先简单回顾一下获得处于JVM执行期的类和接口,用户可以做的事情[5]:

* 确定一个实例所属的类;
* 获得类以及它的父类的变量、属性、方法、构造函数等信息;
* 确定申明的变量、方法属于哪个接口或者类;
* 如果一个类的属性是公有的,就能够生成一个该执行期类的实例;
* 进入并且更改类的属性内容,因为在执行期能

够获得属性名称;

* 可以在编译期不知道对象名称的情况下,调用该对象的方法;

* 可以操作在编译期不知道大小和类型的数组。

JADE 在执行期用反射 API 来动态加载和执行 Agents,筛选对象等操作。以上提到的很多带来隐患的机制还可以应用到私有的类、属性、方法。在编译时,编译程序保证了私有成员的私有特性,从而,一个类的私有方法和私有成员变量不能被其他类静态引用。然而,反射机制使得在运行时可以查询以及访问变量和方法。由于反射是动态的,因此编译时的检查就不再适用了。一个攻击者利用反射机制只需要一个运行的实例,他就可以得到正在当前容器活动的一个 Agent 引用,然后通过 this.getContainerController()读取私有属性 myImpl。myImpl 包含产生当前被攻击 Agent 的容器的引用。得到了这个容器的引用,攻击者就可以进入、操纵该 Agent 的所有数据,包括私有的。这种攻击方法的应用是典型的 Agent 遭受恶意平台攻击。如何保护 JADE 平台使其不受这种攻击,或者让攻击损失降到最低,是值得研究的问题。

3.3 解决方案

恶意 Agent 平台可以利用 JADE-S 这一漏洞攻击 Agent,即 Agent 遭受 Agent 平台攻击,主要有两种可能性:

* 容器和主机是恶意的,专门捕获迁移过去的 Agent,导致安全问题(2);

* 容器和主机已经被破坏,成为了一个恶意主机。

解决这个安全问题,可以从两个角度来进行:恶意主机的识别和移动 Agent 的保护。

首先从恶意主机的识别的角度出发。前面提到,每一个容器加入 JADE 平台的时候首先要在主容器注册,注册的时候,可以进行一些处理。处理的目的有两个:一是识别该容器不是恶意的;二是确保该容器以后也不会变成恶意的(防止欺骗行为)。首先对该容器进行完整性等安全性检查,证明合法性后可以给该容器颁布一个证书,并对这些系统 jar 文件进行签名(防止系统文件被篡改)。每次 Agent 要迁移到某个容器的时候先检查该容器的证书,确保容器的合法性。

其次从移动 Agent 的保护的角度。关于 Agent 的安全保护问题一直是当前学术界研究的热点问题。这个问题跟传统的安全保护问题不同。传统的保护概念一般指的是保护系统安全,而 Agent 的保护问题是保护一段能够自由在网内移动的可执行代码集。当前提出的很多方案并不能完全支持许多 Agent 应用所要求的自由迁移和自治能力,很难广泛应用于实际。这里针对当前的应用,提出在 Agent 遭受恶意攻击时候能够尽量让损失控制在最小。从上面的论述可以看到,在 JADE-S 的安全机制下,Agent 携带了私钥,这样做在带来了效率的同时也带来了安全隐患。解决方法:一是可以对属性值进行重组[6],这样做防止 Agent 泄密;二是 Agent 只携带公钥,需要私钥的时候根据秘密约定向宿主容器请求私钥。

4 结 语

对当前流行的开源移动 Agent 平台 JADE(-S)的安全性问题进行了较为全面的分析。对它的缺陷提出了一些合理的改进方案。下一步的工作是实现研究方案中的具体细节,并测试其性能。为进一步研究移动 Agent 的安全和完善 JAD 的 E 安全提供了参考,有一定的实用价值和前景。

参考文献:

[1] Jansen W, Karygiannis T. Moblile Agent Security[M]. [s. l.]:NIST Special Publication,2000

[2] JADE Board. JADE Home Page[EB/OL]. 2007-05-21. http://jade.cselt.it/.

[3] JADE Board. JADE Security Guide: Usage Restricted According to License Agreement[EB/OL]. Copyright (C) 2004 TILAB S.p.A. 2005-02-28. http://jade.cselt.it/doc.

[4] Vila X, Schuster A, Riera A. Security for a Multi-Agent System based on JADE[J]. Computer & Security,2007,26:391-400.

[5] Campione M, Walrath K. The Java Tutorial, Second Edition: Object Oriented Programming for the Internet[M]. [s. l.]:Addison Wesley Publishing Company,1998.

[6] 孟 健,曹立明,王小平. Mobile Agent 的两种安全机制[J]. 计算机工程应用,2003,29(21):136-138.

(上接第 150 页)

(1):18-21.

[3] 毛 莉,刘广强. 基于.NET 的数据访问策略[J]. 微机发展,2004,14(10):52-54.

[4] 鄢爱兰,鹿江春. 数据库存储过程应用研究[J]. 南华大学学报,2006,20(2):100-102.

[5] MacDonald M. ASP.NET 完全手册[M]. 贾晓军,于秀山,吕嘉章,等译. 北京:电子工业出版社,2003.