移动代理系统中恶意主机攻击模型的研究*)

郑 彦[1] 王汝传[1,2] 王海艳[1]
(南京邮电大学计算机科学与技术系 南京 210003)[1]
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 作为一种新的分布式计算模式,移动代理技术具有广泛的应用前景。但在目前的移动代理系统中,恶意主机问题,即保护代理免受恶意主机的攻击是很棘手的问题。为了更好地理解该问题,我们提出了基于抽象机器模型的攻击模型(RASPS)。该模型有助于提出有效的移动代理保护方案,并可以作为评价各种保护方案的基础。本论文首先分析了恶意主机的攻击行为,并在此基础上提出了恶意主机的攻击模型,最后分析了攻击实例程序。

关键词 移动代理,攻击模型,分布式计算

Malicious Hosts Problem and Model of Attacks against Mobile Agents in Mobile Agent Systems

ZHENG Yan[1] WANG Ru-Chuan[1,2] WANG Hai-Yan[1]
(Department of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003)[1]
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)[2]

Abstract As a new distributed computing technology, mobile agent has a wide application prospect. But in current mobile agent systems, malicious hosts problem, namely protection of mobile agent against malicious hosts is a tough problem. To better understand this problem, this paper puts forward an attack model. This model helps to find effective solutions of this problem and work as a basis to evaluate various solutions. First, this paper analyses the attack behaviors of malicious hosts; then puts forward the attack model ;finally interprets an attack instance program.

Keywords Mobile agent, Attack model, Distributed computing

1 引言

现代网络技术向分布式、动态化、智能化方向发展,移动代理技术就是在这一需求下应运而生的一种新的分布式计算模式。移动代理是一段独立于操作平台和操作系统的程序代码,它可以在计算机网络中漫游,代表用户在不同网络节点上进行交互工作。用户或代理控制中心向网络发出一个(或多个)移动代理,该代理按照特定的路线(用户预定义,或由它自己确定)在网络中漫游并与各服务器节点交互,完成任务后携带计算结果返回(初始用户或代理控制中心)。

移动代理可被看作是传统通信机制的一种扩展,由于能明确控制代码在何处运行,移动代理模式提供了特殊的优势,特别是当用于诸如万维网这样的动态异构网络环境的时候。移动代理拥有很多优点,如减少网络流量、增加客户机和服务器的异步性、便于负载均衡和容错、支持移动客户和服务定制等。它在通信网络管理、智能网领域、Internet 上的智能信息检索以及分布计算等领域有广泛的应用前景。

制约移动代理广泛应用的一个重要因素是代理的安全性。移动代理的安全问题主要有以下几个方面:传输过程中的移动代理受攻击;代理执行环境及主机对移动代理的攻击;移动代理对执行环境及主机的攻击。对于传输中的代理保护,可以用网络中的信息保护策略加以解决,目前已经有了较为成熟的解决方案。对于代理执行环境及主机的保护,目前已经进行了许多研究,并提出了一些有效可行的方法,例如移动代理的授权、身份认证、访问控制等。而对于移动代理的保护却是一个很棘手的问题,因为代理是完全暴露在目的代理系统中的,代理执行者可以很容易地孤立、捕捉代理,对代理进行攻击。

2 恶意主机问题

在已有的计算机系统中,主机攻击在其上运行的程序,这种情况几乎是不存在的,因为该程序是属于该主机的。而在开放的移动代理系统中,移动代理在绝大多数情况下都是由另一台主机所运行的。当一个目标主机接收到移动代理时,代理的拥有者就完全失去了对代理的控制。移动代理的每一行代码都要被目标上的代理执行环境解释、执行,也就是说,代理的代码对目标代理执行环境而言是完全可见的。这就导致了恶意主机的问题,而该问题是移动代理模式在开放系统中广泛使用必须解决的问题。恶意主机对代理的攻击有如下情形:

(1)窥视和篡改代码。

移动代理在主机上执行,主机必须能够读取其代码。恶

*)本课题得到国家自然科学基金(60573141 和 70271050)、江苏省自然科学基金(BK2005146)、江苏省高技术研究计划(BG2004004、BG2005037)、国家高科技 863 项目(2005AA775050)、江苏省计算机信息处理技术重点实验室基金(KJS050001)资助和江苏省高校自然科学研究计划(05KJB520092)资助。郑 彦 副教授,博士,主要研究方向为数据挖掘、信息安全以及移动代理等;王汝传 教授,博士生导师,主要研究方向是计算机软件、计算机网络和网格、信息安全、无线传感器网络、移动代理和虚拟现实技术等;王海艳 讲师,硕士,在读博士,主要研究方向为计算机软件、计算机网络、信息安全、移动代理等。

意主机可以通过逆向分析，如反汇编等，获取代理的意图、执行策略等。恶意主机可以篡改代码，如在代理中植入病毒、特洛伊木马或蠕虫。这是造成危害最大的操作，因为代码的修改将改变代理所执行的操作，将一个可信任的代理变为一个恶意代理，攻击网络上其他计算机。

(2)窥视和篡改数据。

恶意主机可以轻易地读取移动代理的私有数据而不留下任何痕迹。代理中的数据可以分为3类：

①执行状态数据：代理移动到新的平台上能够继续执行所需要的信息；

②用户的秘密信息：如用户的私钥，信用卡信息等；

③工作数据：代理完成任务所使用的信息，一般表现为局部变量和全局变量。例如，某一购物的代理收集了不同商家对某产品的报价；

恶意主机在仔细分析代码和数据后，可以对数据进行修改，例如提高其他商家对产品的报价，使代理的决策发生错误；

(3)窥视和操纵控制流。

一旦主机得知Agent的代码和数据，就可以很容易地确定任何时候Agent的动作，下一条指令是什么，在运行时刻全面监视Agent的执行，并在适当的时候，通过改变运行时刻的控制流来控制Agent的行为；

(4)拒绝执行代码。

恶意主机可能拒绝代理请求的服务，故意延迟甚至不执行代理。有些代理的任务和时间密切相关，这时这种攻击是很有效的。

(5)窥视代理与其它代理的交互。

恶意主机不但可以窥视Agent的交互，也可以直接更改交互的内容，甚至伪装成其它代理，中途接管交互过程；

(6)代理进行系统调用时返回错误的结果。

目前的移动代理保护技术还处于起步阶段，主要分为两个方面：

①基于检测的安全性措施：通过对运行环境进行检测来判断其是否安全，以及通过对移动代理的运行结果作检测来判断其是否受到了攻击。典型的方法有不让移动代理到不被信任的运行环境中去执行任务，要求提供一种检测和评价各个主机可信任度的手段；在移动代理中加入一个状态评价函数，移动代理将根据状态评价函数的运算结果决定下一步的行动，但该函数是由运行环境执行的，所以运算结果的可靠性仍然值得怀疑。

②主动的保护措施：基于检测的方式是被动的，并不能真正保护移动代理免受破坏。主动的保护措施有：黑匣子(Blackbox)保护方法，它的思想是从一个给定的代理规范来产生可执行的代理，如果在任何时间内Agent不会受到攻击，且只有它的输入和输出行为被观察到，则一个Agent可被看作一个黑匣子；加密函数保护方法，用于保护移动代理中部分关键数据和算法，如对某个计算函数进行加密，使攻击者无法了解函数的内部逻辑；共享秘密和互锁，由两个和两个以上的移动代理来共同完成一项任务，每个移动代理保持部分秘密，只有当它们达成一项协议后才可最终完成任务。一个完全不同的方法是使用可信任的、防攻击的硬件对代理进行保护，其核心思想是给代理系统配置额外的硬件，这个硬件不受主机控制，并且主机只能通过受限制的接口来访问硬件中的系统，因而能够为代理提供安全的执行环境。

我们提出恶意主机的攻击模型，就是为了帮助理解恶意主机的攻击行为，理解这种攻击行为发生的条件及原理，从而提出有效的保护方案。另外，该攻击模型还可以用于评估代理的保护方案。

4 恶意主机的攻击模型

4.1 攻击模型的要求

考虑到恶意主机对移动代理各种可能的攻击，攻击模型要符合一定的条件。模型中的攻击者必须能够读或者修改移动代理的数据、代码、状态、执行方式、与其他代理的通信以及系统调用的结果，而且攻击模型必须是抽象的，不仅要求对已有的各种代理语言是适用的，而且对未来的用于保护代理的语言也是适用的。

这种适用性的缺陷是，用目前已有的语言编写的代理可能更容易被恶意主机攻击，例如Java，Tcl，C++等。这主要是因为人们当初设计这些语言时主要用于提高程序执行的效率。为了保护移动代理免受主机攻击，代理的理想语言可能与提高程序执行效率的目标相反，执行环境应该对所执行的程序知道得尽可能地少，而对常规的程序设计语言而言，解释器应该对代码知道得尽可能地多。

4.2 攻击模型(RASPS)

我们将抽象计算机模型作为攻击模型的基础，该抽象计算机模型被称为随机访问存储程序和堆栈机器(RASPS，Random Access Stored Program plus Stack Machines)，它是计算机科学中用于解释计算机工作原理时所用的最基础的模型。RASPS包含一个内存单元向量、一个堆栈、堆栈指针、程序计数器。内存单元被用于存储装载的程序指令，堆栈单元被用于存储程序执行过程中的临时变量，它们都用一个简单的整数来编排地址。当下一条指令将要被执行时，机器根据程序计数器的值来取出相应的指令，解码并执行指令。

堆栈和内存单元都用于存放数据，它们都用一个从0开始的整数进行地址编排。当下一条指令要被执行时，机器根据程序计数器的值取出相应内存单元的数据。该数据是一条指令。机器解码并执行该指令。有些指令包含操作数，因此需要多个内存单元。例如，地址为2的内存单元中的数据值为250，翻译为指令为“Push x”，另外该指令需要操作数，该操作数放在另一个内存单元中。机器执行该指令的规则是将操作数压入堆栈。该指令执行完，程序计数器加上该指令所占的内存单元数。

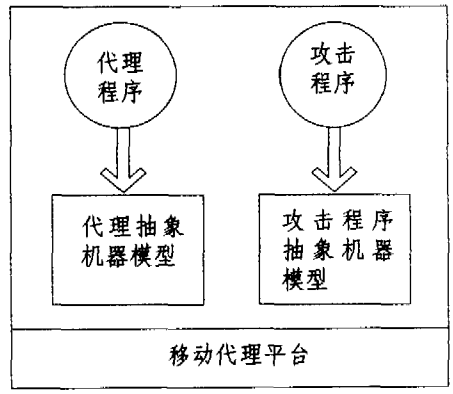

图1 恶意主机攻击模型

在我们的攻击模型中，攻击者被抽象为能够发起各种类型攻击的计算机程序；移动代理也是一个自包含的计算机程

序，在代理执行环境中执行。

代理平台初始化时，攻击程序被装载到攻击者的 RASPS 中。当移动代理到达该主机时，代理被装在到主机的内存，这种情形可以被抽象成代理被装载到代理的 RASPS 中，如图 1 所示。这样，攻击模型可以简化成两个抽象机器模型。一个机器用于装载代理程序，另一个装载攻击程序。

代理程序不能直接访问主机和系统信息，必须通过执行环境来间接访问。代理也不能直接和其他代理通信，必须使用执行环境所提供的通信方式来通信。这样，攻击程序能够通过控制执行环境达到攻击代理程序的目的，甚至攻击程序包含这个执行环境，如图 2 所示。

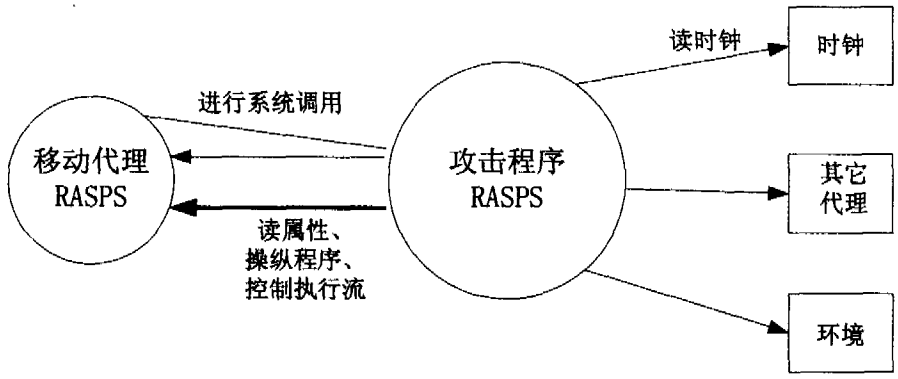

图 2　攻击模型

这样，由代理发起的系统调用被攻击 RASPS 所截获，并调用自己的程序，返回错误的结果。载有攻击程序的攻击 RASPS，能够存取代理 RASPS 的内存和堆栈，也能够存取代理 RASPS 的堆栈指针和程序计数器。通过这个模型，我们可以理解一个典型的攻击过程：

(1)在代理程序的下一条指令执行前，攻击 RASPS 取得代理 RASPS 的指令，并解码；

(2)攻击 RASPS 将取得的指令和参数存储到自己的堆栈里；

(3)攻击 RASPS 计算代理 RASPS 要执行的下一条指令的地址(即代理 RASPS 的程序计数器的值)，并将该地址存储到自己的堆栈里；

(4)攻击 RASPS 执行自己的相应攻击程序；

(5)攻击 RASPS 执行存储在自己堆栈中的代理指令；

(6)攻击 RASPS 将堆栈中的程序计数器的值存储到 RASPS 的程序计数器中；

(7)继续下一个循环。

这样，攻击 RASPS 可以控制代理 RASPS 运行的每一条指令。攻击 RASPS 可以允许代理 RASPS 正确的运行，删除或者修改代理 RASPS 内存中的指令。攻击程序也可以有选择地执行部分指令，并插入一些恶意指令到代理 RASPS 的内存中。这样，代理就完全受控于攻击者，代理的代码可以被任意修改、删除、增加或者有选择性的执行，代理在恶意主机环境中运行的脆弱性就完全反映了。

我们的攻击模型主要有两个用途：

①使用该模型便于说明恶意主机的问题。例如，移动代理不是仅仅受到恶意主机的某一种攻击，而是各种可能的攻击的集合，我们可以将这些攻击看成是该抽象模型的实例。到目前为止各种可能的攻击还没有被明确地证实。攻击模型可以被用于编写攻击程序，用于证实某一种攻击。

②该模型可以在证明某种保护方案的可行性时提供基础。我们已经知道，对于任何一种保护方案，不仅要求保护的算法是安全的，而且要求该算法的代码受到保护。因此，我们可以使用攻击模型来评价某个具体的保护方案。

4.3　攻击实例分析

```
Fetch 10      #10 = variable A
Fetch 11      #11 = variable B
Fetch 12      #12 = variable C
Call 40        #40 = buy
```

(a) 原始代码

```
A = (<230>XOR<194>)+<233>
B = (<193>Shift<231>)*<190>
C = (<191>-<234>)Shift<232>
D = <500>+<501>-<150>
```

(b) 代码转换

```
Fetch 230   Fetch 194   Xor         Fetch 233   Add
Fetch 190   Fetch 193   Fetch 231   Shift       Multiply
Fetch 232   Fetch 191   Fetch 234   Sub         Shift
Fetch 500   Fetch 501   Add          Fetch 150   Sub
Call
```

(c) 转换后的代码

图 3　攻击实例分析

为了说明攻击模型，我们将构建一个移动代理和一个攻击程序。移动代理将执行定义好的购买程序，购买程序的某个参数 A 包含电子货币，代理有权代表用户使用该电子货币，因此代理在移动过程中要尽力保护该数据。代理使用这样一种机制，将要保护的数据通过一个转换函数散布到一系

(下转第 109 页)

表 1 模拟参数

parameter	Value	parameter	value
test time	10s	size of database	500
number of nodes	5	number of sub-transaction	1—5
average message send time	316us	number of operations in each sub-transaction	5—15
data send period	10—100ms		

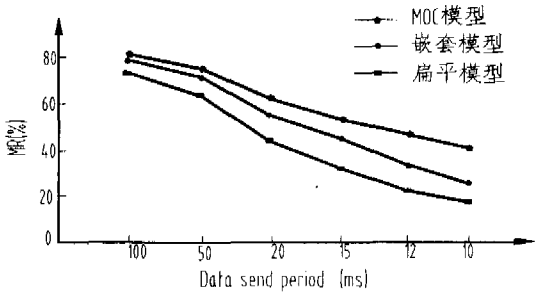

图 3 采样周期对 MR 的影响

结束语 传统的事务模型不能够较好的支持传感器网络中数据融合事务的执行。本文结合扁平事务、嵌套事务模型、数据流及事务工作流模型提出了一种基于事务及相关数据历史的控制域的较为适应该环境的 MOC 模型，并给出了相关定理和提交规则、回滚规则、可见规则。通过模拟实验进行测试，证明其可以提高监测数据的系统融合事务的实时性并能保证应用的松弛原子性。本系统在我们的原有分布式数据库 ARTs-Ⅱ的基础上正在研制过程中。

参 考 文 献

1 Tilak S, Abu-Ghazaleh N B, Heinzelman W. A taxonomy of wireless micro-sessor network models. Mobile Computing and Communications Review, 2002, 1(2): 1～8

2 Savarese C, Rabaey J. Robust positioning algorithms for distributed ad-hoc wireless sesor networks. In: Park Y, ed. Proceedings of the USENIX Technical Annual Conference. Monterey: USENIX, 2001. 317～328

3 Ratnasamy S, Karp B. GHT: A geographic hash table for data-centric storage. In: Reghavendrv CS, ed. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications. New York: ACM Press, 2002. 94～103

4 Girod L, Bychkovskiy V, Elson J, et al. Locating tiny sensors in time and space: A case study. In: Manoli Y, Kim KS, eds. Proceedings of the International Conference on Computer Design, Piscataway: IEEE Press, 2002. 195～204

5 Rentala P, Musunuri R, Gandham S, et al. Survey on sensor networks: [Technical Report]. UTDCS-33-02. University of Texas at Dallas, 2002

6 Madden SR, Szewczyk R, Franklin MJ, et al. Supporting aggregate queries over ad-hoc wireless sensor networks. In: Kindberg T. ed. Proceedings of the Workshop on Mobile Computing and Systems Applications. Los Alamitos: IEEE Computer Press, 2002. 49～58

7 Madden SR, Shah MA, Hellerstein JM, et al. Continuously adaptive continuous queries over streams. In: Franklin MJ, Moon B, Ailamakj A. eds. Proceeding of the SIGMOD Conference. New York: ACM Press, 2002. 49～60

8 Bonnet P, Gehrke JE, Seshadri P. Towards sensor database systems. In: Tan K-L, Franklin MJ, Lui JCS. eds. Proceedings of the 2nd International Conference on Mobile Data Management. Hong Kong: Springer-Verlag, 2001. 3～14

9 Jajodia S, Kerschberg L, eds. Advanced Transaction Models and Architectures. Kluwer, 1997

10 Whiting PG, Pascoer SV. A history of data-flow languages. IEEE Annals of the History of Computing, 1994, 16(4): 38～59

11 Garcia M H, Salem K. Sagas. In: Proceedings of SIGMOD International Conference on Management of Data. San Francisco, 1987. 249～259

12 Reuter A, Schwenkres F. ConTracts-a low level mechanism for building general purpose workflow management system. Data Engineering, 1995, 18(1): 4～10

(上接第 80 页)

列不同的变量中，如图 3(b)所示。

使用该转换函数，我们产生了一个简易的“黑匣子”机制，将原有的代码转换为黑匣子保护代码(如图 3(a)所示的原有代码和图 3(c)所示的黑匣子保护代码)。

攻击者试图获取电子货币的值。它知道代理的黑匣子保护代码，但不知道转换函数，所以并不知道原来的代码。由于黑匣子保护方案存在着一个严重的问题：即使原有的值通过转换函数已经变化了，但它仍然是明文，仍然可以作为程序调用的参数，因此攻击者可以等待代理执行调用购买子程序的指令，然后读出正确的调用参数，该参数即电子货币的值。攻击程序的代码如图 4 所示。

```
攻击子程序
10   push "Call 40"        //将指令所对应的数值放入堆栈;
13   equals?               //比较栈顶两个元素的值
14   if true  goto 17
     else goto 20
17   ...
18   ...
20   push "Call 40"
23   return
```

图 4 攻击程序

攻击 RASPS 将 Call 40 的指令放入到自己的堆栈中，然后将下一条指令“equals”放入到堆栈中，并在攻击程序执行前进行解释执行，该指令将代理所要执行的指令和 Call 40 指令进行比较，如果结果为 true，则这时攻击 RASPS 就获得了电子货币的值，因为此时电子货币的值刚好在堆栈的顶部。因为“equals”操作去掉了原有堆栈顶部的两个元素，而“if”操作又去掉了新放在堆栈顶部的布尔值，这样作为 buy 操作的 call 40 指令所需要的参数刚好在堆栈的顶部。

小结 恶意主机问题是移动代理系统获得广泛使用必须解决的问题。我们分析了恶意主机对代理的攻击，并比较了目前已有的各种代理保护方案，进而基于抽象机器模型，提出的恶意主机对代理的攻击模型，从理论上分析了攻击的可能性以及攻击的方法；另一个方面也有助于提出新的代理保护方案，并对各种保护方案进行评价。

参 考 文 献

1 王常杰，张方国，王育民. Internet 移动代理技术中的安全性研究. 西安电子科技大学学报(自然科学版)，2001，28(2)

2 朱向华，万燕，孙永强. 移动代理系统的安全机制. 计算机科学，2001，27(1)

3 何炎祥，陈苹萌. Agent 和多 Agent 系统的设计和应用. 武汉：武汉大学出版社

4 (美)Bruce Schneier 著. 应用密码学——协议、算法与 C 源程序. 吴世忠，祝世雄，张文政，等译. 北京：机械工业出版社，2000

5 Pfleeger C P. Security in Computing. Second Edition. Upper Saddle River, NJ: Prentice Hall, 1997

6 Chess D. Security Issues in Mobile Code Systems. In: Vigna G. Ed. Mobile Agent Security, LNCS 1419, Springer, 1998. 1～14