Java语言的安全特性

罗　红，刘洪坤，程　建
(空军工程大学 电讯工程学院，陕西 西安　710077)

摘　要：讨论了Java语言的安全特性，指出applet的“沙盒”是Java语言安全特性的核心，它有效地阻止了病毒传染、信息窃取等非法操作。Java的安全模型是一种“三叉”结构，包括：类加载器、类校验器和安全管理器。

关键词：Java；applet；沙盒；语言安全特性
中图分类号：TP312JA　　**文献标识码**：A　　**文章编号**：1009-3516(2000)04-0052-03

1　Java安全模型

Java在网络环境中可将执行程序自动分布的功能，给人们带来了更大的安全担心。Java语言声称可以利用applet辨识应用程序，从而解决了这一问题。图1给出了Java的安全模型。Java程序以两种形式存在：第一种是applet，它能在互连网和内部网中作为网页的一部分进行传递，并在终端用户的浏览器内部运行。第二种是传统的应用程序。图1给出了Java的applet程序与应用程序和Java虚拟机JVM(Java Virtual Machine)之间的关系。

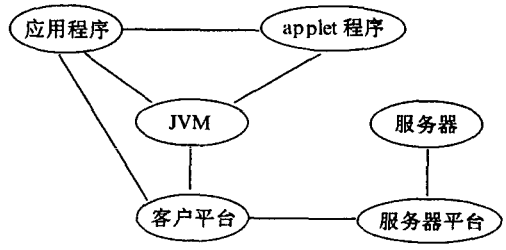

图1　Java安全模型

1.1　Java应用程序

Java应用程序没有安全机制。计算机安全要靠加强计算机管理制度和日常维护来解决。经常采取的计算机安全措施有，在全球网和单位内部网之间建立防火墙，终端用户不要从网络上下载可疑程序，不要随意使用外来软件，只能从可靠部门获取软件，另外还要经常利用杀毒软件和其它安全工具软件，来检查和维护计算机的安全。

1.2　Java applet

Java的applet采用特殊的“沙盒”安全机制。因为“可疑”程序大多是从全球网上下载下来的，并在本地计算机上进行了运行，而计算机用户经常未意识到他们正在执行恶意程序。“沙盒”正是为解决这一问题而设计的。

applet是Java程序的可执行形式，也是Java语言安全机制的重要核心。applet是包含在网页上能够在用户浏览器中运行的可执行程序的小片段。Java的安全机制允许一个用户从外部网络或内部网络上下载而且不破坏的运行applet程序。applet的活动被限制在“沙盒”中。applet程序在沙盒中可以做任何事情，但不能读写沙盒外的任何数据。沙盒模型的目的是在可靠环境中运行可疑程序，如果一个用户偶尔输入了一个恶意applet，它也不会对本地计算机造成破坏。

Java可自动地将applet限制在沙盒中。由于沙盒阻止了传染病毒、窃取信息的所有操作，并努力识别已感染了病毒的程序或潜在的黑客攻击，沙盒对新病毒的检测机制不需要定期修改。如上所述，applet的各种操作是受Java的安全机制限制的。如果一个applet是从网络上下载下来的，它还不允许进行如下的操作：

收稿日期：2000-03-17
作者简介：罗　红(1963-)，男，陕西西安人，讲师，硕士，主要从事计算机信息安全研究．

(1)从客户文件系统中读取文件。

(2)给客户文件系统中写入文件。

(3)删除客户文件系统中的文件。包括不能使用 File. delete( )方法、或调用操作系统的命令 rm 或 del。

(4)更改客户文件系统中文件名。包括不能使用 File. rename To( )方法、或调用操作系统的命令 mv。

(5)在客户文件系统上创建文件。包括不能使用 File. mkdirs( )方法、或调用操作系统的命令 mkdir。

(6)显示一个文件目录内容。

(7)查看一个文件是否存在。

(8)获取一个文件的信息。如文件大小、类型、修改时间等。

(9)除了可以连回发送它的计算机外,不能将网络连至其他任何计算机终端。

(10)在客户系统上听取或确认任何端口上的网络连接。

(11)创建带有可疑窗口标志的顶层窗口。

(12)利用任何工具获取用户名或主目录名。包括试图读取系统特性,如 use. name,user. home,user. dir,Java. home,Java. class. path 等等。

(13)定义任何系统属性。

(14)利用 Runtime. exec( )方法在客户系统上运行任何程序。

(15)利用 Runtime 或 System 类中的 Load( )、LoadLibrary( )方法在客户系统上装载动态库。

(16)用 applet 创建或生成非相同 ThreadGroup 组的线程。

(17)创建类加载器。

(18)创建安全管理器。

(19)指定任何网络控制,包括 ContentHandlerFactory,SocketImplFactory,URLStreamHandlerFactory。

(20)在客户系统中定义部分包类。

2　Java 安全机构

Java 的安全机构是由几个不同的系统操作一起构成的,Java 的沙盒安全模型可以用"三叉攻击"模型来描述。三叉攻击包括:

- 类加载器
- 类校验器
- 安全管理器

"三叉"这个词是用于描述安全模型的。而不是描述安全层次的。三个分叉中缺少一个都会给系统带来缺陷。我们不要只重视其中一个,而忽视另外两个。

2.1　类加载器

当从网络上下载一个 applet 并输入到计算机以后,网络浏览器就调用 applet 的类加载器进行处理。类加载器是 Java 安全机制链条中的第一个环节。除了要从网上下载获取 applet 可执行程序外,类加载器强行划分名称空间层次。一个名称空间管理着一个 applet 能访问 Java 虚拟机的其它内存空间。对于从本地盘上加载的可信程序,通过维护独立的名称空间,类加载器就可阻止可疑 applet 获取到系统的部分访问特权与信任。

从网络上下载的 applet 程序不能创建自己的类加载器,也不能引用系统的类加载器。

2.2　类校验器

在运行一个网络上下载的 applet 以前,类加载器先要调用校验器。校验器将检查 applet 对 Java 语言规格的确认,并检查是否有不符合 Java 语言规则和名称空间限制的内容。校验器还检查内存管理方面的错误,如堆栈的上溢/下溢,非法数据类型等。这方面的漏洞也会使恶意 applet 程序破坏系统的安全机制,或将系统的部分内容替换成它自己的代码。图 2 给出类加载器与类校验器之间的层次关系。

2.3　安全管理器

安全管理器定义沙盒的边界。SecurityManager 类对可疑程序的执行作出一定的限制。而正常的程序不使用安全管理器,安全管理器只用于网络浏览器,applet 阅读器,以及其他在控制环境中需要调用可疑程序

来使用。

如果一个 applet 程序要打扰本地计算机或进行访问信息的操作,Java 虚拟机首先会询问安全管理器这个操作是否安全。例如,从本地盘加载的一个可信 applet 程序试图读取磁盘,或者一个被加载的可疑 applet 试着连回它的原始服务器,安全管理器将会许可这些操作请求,Java 虚拟机然后再执行具体操作。否则,如果安全管理器不允许某一操作,虚拟机将提升一个安全等级,并给 Java 控制台输出一个错误信息。

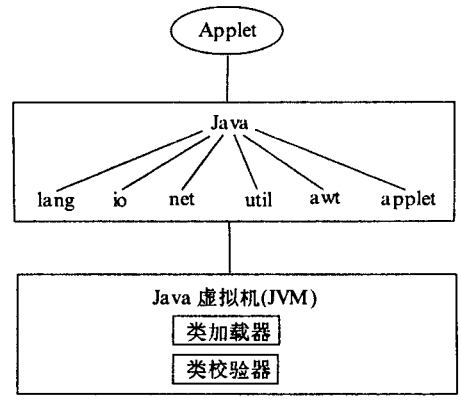

图 2 Java 语言的安全检验

安全管理器不允许可疑 applet 读取、写入或删除文件,不允许获取一个文件的相关信息,不能执行操作系统命令或运行自身程序、装载一个库、与其他机器建立网络连接(除了原始服务器以外)。

一个应用程序或一个网络浏览器只能有一个安全管理器。这就保证了所有的访问检查只能由一个安全管理器按唯一的安全策略进行安全检查。安全管理器是在计算机开机时加载的,系统不允许再对它扩充、修改或替换。道理很简单,applet 不能创建属于自己的安全管理器。

3 结束语

在网络环境中,Java 的安全状态可划分为如下几种:

(1)系统修改——Java 对该类型的进攻都可以很好地防护。

(2)窃取秘密——Java 对该类型的进攻都可以很好地防护。

(3)否定服务——Java 对该类型的进攻只能较弱地防护。

(4)对　　抗——Java 对该类型的进攻只能较弱地防护。

根据上面列出的情况看,Java 的安全机制主要是针对修改系统和窃取秘密的信息攻击的。Java 通过在 applet 中设置严格的限制,从而使用户计算机免遭破坏。这些限制防止了恶意 applet 程序窃取信息、传播病毒以及实施特洛伊木马行动。Java 还能禁止恶意 applet 程序与其他计算机的网络连接。这样就可阻止恶意 applet 探测存在于防火墙或操作系统中的安全缺陷。

参考文献:

[1] 姜　刚,胡全星. Java 语言程序设计[M]. 北京:人民邮电出版社,1998.

JAVA Security Architecture

LUO Hong, LIU Hong-kun, CHENG Jian
(The Telecommunication Engineering Institute, AFEU., Xi'an 710077, China)

Abstract: We discuss the Java security characteristic. The 'sandbox' mechanism is Java's major security concerns. It efficiently prevents the actions required to spread a virus or steal information. The Java security model can be described as a 'three-pronged'. It is made of the Class Loader, the Class Verifier and the Security Manager.

Key words: Java; applet; sandbox; security architecture