



(12) 发明专利申请

(10) 申请公布号 CN 103136470 A

(43) 申请公布日 2013. 06. 05

(21) 申请号 201310079403. 5

(22) 申请日 2013. 03. 12

(71) 申请人 无锡江南计算技术研究所

地址 214083 江苏省无锡市滨湖区军东新村  
030 号

(72) 发明人 唐大国 季振宇 郑磊 叶俊  
李茜

(74) 专利代理机构 北京众合诚成知识产权代理  
有限公司 11246

代理人 龚燮英

(51) Int. Cl.

G06F 21/53(2013. 01)

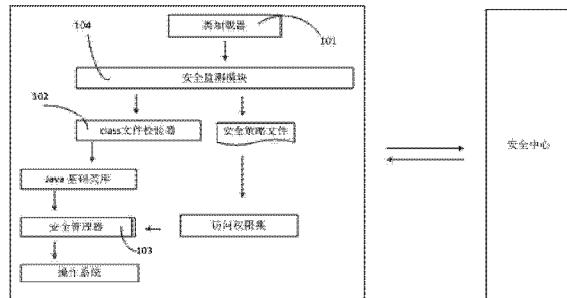
权利要求书1页 说明书4页 附图1页

(54) 发明名称

一种增强 Java 虚拟机安全的方法

(57) 摘要

一种增强 Java 虚拟机安全的方法，包括：在用户端运行应用时，用户端的安全管理器单元发起向安全中心建立连接请求；安全中心对连接请求进行验证并答复是否建立连接请求，安全中心针对连接请求验证用户端是否具备建立连接安全中心的权限，如果用户端的安全等级符合安全中心的认定，则建立连接请求。在安全中心通过对连接请求的验证从而建立连接的情况下，用户端的安全监测模块将签名后的应用摘要信息发送至安全中心进行认证；安全中心根据接收到的签名后的应用摘要信息对此应用进行认证，并答复是否同意用户端运行应用的请求。在安全中心没有通过对连接请求的验证从而不建立连接的情况下，用户端的安全监测模块执行抛异常处理并退出当前应用。



1. 一种增强 Java 虚拟机安全的方法,其特征在于包括 :

在用户端运行应用时,用户端的安全监测模块发起向安全中心建立连接请求;

安全中心对连接请求进行验证并答复是否建立连接请求,其中,安全中心针对连接请求验证用户端是否具备建立连接安全中心的权限,如果用户端的安全等级符合安全中心的认定,则建立连接请求。

2. 根据权利要求 1 所述的增强 Java 虚拟机安全的方法,其特征在于还包括 :

在安全中心通过对连接请求的验证从而建立连接的情况下,用户端的安全监测模块将签名后的应用摘要信息发送至安全中心进行认证;

安全中心根据接收到的签名后的应用摘要信息对此应用进行认证,并答复是否同意用户端运行应用的请求。

3. 根据权利要求 1 所述的增强 Java 虚拟机安全的方法,其特征在于还包括 :在安全中心没有通过对连接请求的验证从而不建立连接的情况下,用户端的安全监测模块执行抛异常处理并退出当前应用。

4. 根据权利要求 2 所述的增强 Java 虚拟机安全的方法,其特征在于还包括 :

用户端的类加载器单元初始化基础类库,并计算基础类库的签名信息,并将基础类库的签名信息交由安全监测模块发送至安全中心以请求对基础类库的签名信息进行认证;

安全中心对从用户端接收到的针对基础类库的签名信息的认证请求进行验证。

5. 根据权利要求 4 所述的增强 Java 虚拟机安全的方法,其特征在于,在安全中心对从用户端接收到的针对基础类库的签名信息的认证请求进行验证的步骤中,安全中心确认基础类库是否被修改,其中,通过判断基础类库的签名信息的版本和安全中心的签名信息的版本是否一致来确认基础类库是否被修改。

6. 根据权利要求 1 或 2 所述的增强 Java 虚拟机安全的方法,其特征在于还包括 :用户端向安全中心请求分发安全策略文件;安全中心在接收到分发安全策略文件的请求后验证用户端的资格,并据此决定是否发送相关安全策略文件。

7. 根据权利要求 1 或 2 所述的增强 Java 虚拟机安全的方法,其特征在于还包括 :用户端向安全中心发送相关共享库的有关认证请求;安全中心在接收到共享库的有关认证请求后对共享库的来源、版本进行认证。

## 一种增强 Java 虚拟机安全的方法

### 技术领域

[0001] 本发明涉及计算技术领域,更具体地说,本发明涉及一种增强 Java 虚拟机安全的方法。

### 背景技术

[0002] Java 语言是一种面向网络的软件技术,由于网络允许数据的共享和分布处理,使得计算机系统具有被入侵的潜在风险,故而 Java 需要解决其所面临的安全问题。

[0003] Java 的安全模型称为 Java 沙箱,Java 沙箱侧重于保护终端用户免受从网络上下载的、来自不可靠来源的、恶意程序的侵犯。Java 沙箱主要的基本组件有类加载器、class 文件校验器、安全管理器等。Java 沙箱通过对类加载器、安全策略的定制,可以根据应用本身的性质制定个性化安全策略。

[0004] Java 应用一般会利用不同的第三方类库,基于组件进行构建。但是,目前的安全机制对组件和第三方类库管理的不是很完善,Java 沙箱对其仅仅在于初始化时进行限制,对组件的运行时没有做更多的安全控制;另外,Java 应用通过调用本地共享库,不仅可以利用原有的代码迅速构建应用,而且可以获取更好的平台优势,充分发挥平台所提供的功能。

[0005] 但是,由于沙箱仅仅对本地共享库的载入进行控制,而不能对共享库本身的一些敏感操作进行安全控制,这容易引起严重的安全问题。

### 发明内容

[0006] 本发明所要解决的技术问题是针对现有技术中存在上述缺陷,提供一种融合安全中心与 Java 沙箱保护的安全机制,以增强 Java 自身的安全。

[0007] 根据本发明,提供了一种增强 Java 虚拟机安全的方法,其包括:在用户端运行应用时,用户端的安全监测模块发起向安全中心建立连接请求;安全中心对连接请求进行验证并答复是否建立连接请求,其中,安全中心针对连接请求验证用户端是否具备建立连接安全中心的权限,如果用户端的安全等级符合安全中心的认定,则建立连接请求。

[0008] 优选地,所述的增强 Java 虚拟机安全方法还包括:在安全中心通过对连接请求的验证从而建立连接的情况下,用户端的安全监测模块将签名后的应用摘要信息发送至安全中心进行认证;安全中心根据接收到的签名后的应用摘要信息对此应用进行认证,并答复是否同意用户端运行应用的请求。优选地,所述的增强 Java 虚拟机安全方法还包括:在安全中心没有通过对连接请求的验证从而不建立连接的情况下,用户端的安全监测模块执行抛异常处理并退出当前应用。

[0009] 优选地,所述的增强 Java 虚拟机安全方法还包括:用户端的类加载器单元初始化基础类库,并计算基础类库的签名信息,并将基础类库的签名信息交由安全监测模块发送至安全中心以请求对基础类库的签名信息进行认证;安全中心对从用户端接收到的针对基础类库的签名信息的认证请求进行验证。

[0010] 优选地,在安全中心对从用户端接收到的针对基础类库的签名信息的认证请求进

行验证的步骤中,安全中心确认基础类库是否被修改,其中,通过判断基础类库的签名信息的版本和安全中心的签名信息的版本是否一致来确认基础类库是否被修改。

[0011] 优选地,所述的增强 Java 虚拟机安全方法还包括:用户端向安全中心请求分发安全策略文件;安全中心在接收到分发安全策略文件的请求后验证用户端的资格,并据此决定是否发送相关安全策略文件。

[0012] 优选地,所述的增强 Java 虚拟机安全方法还包括:用户端向安全中心发送相关共享库的有关认证请求;安全中心在接收到共享库的有关认证请求后对共享库的来源、版本进行认证。

[0013] 本发明结合安全中心的安全机制在应用初始化和运行时两个阶段对类的安全进行控制,防止可信的类被破坏,保证不可信的类获取规定的权限,保证应用的操作不会突破沙箱的边界;通过安全中心解决使用公共密钥技术时的密码分发。通过统一制定、分发安全策略,保证了安全策略个性化的同时,集中控制应用的基础类库的安全。

[0014] 本发明结合安全中心,对应用、应用所使用的基础类库、第三方类库和共享的本地类库进行安全认证,保证应用所涉及的类库本身的安全性;通过在和安全中心交互所获取的类的基本信息,在运行时保证类的元信息不能被修改。同时,有效解决在鉴别和认证过程中的公共密钥技术的密码分发问题,有效地增强了 Java 的安全模型。

## 附图说明

[0015] 结合附图,并通过参考下面的详细描述,将会更容易地对本发明有更完整的理解并且更容易地理解其伴随的优点和特征,其中:

[0016] 图 1 是根据本发明实施例采用的 Java 虚拟机的增强型安全机制结构图。

[0017] 需要说明的是,附图用于说明本发明,而非限制本发明。注意,表示结构的附图可能并非按比例绘制。并且,附图中,相同或者类似的元件标有相同或者类似的标号。

## 具体实施方式

[0018] 为了使本发明的内容更加清楚和易懂,下面结合具体实施例和附图对本发明的内容进行详细描述。

[0019] 本发明分别在初始化和运行时两方面提供安全管理,以全局集中、节点自治的协同方式保证虚拟机的安全。“全局集中”即统一管理安全策略的制定与分发、集中控制应用的基础类库的安全;“节点自治”即融合虚拟机的类载入机制和安全执行控制机制以保证虚拟机安全。

[0020] 图 1 是根据本发明实施例采用的增强型安全机制结构图。

[0021] 如图 1 所示,本发明实施例采用的 Java 虚拟机的增强型安全机制结构包括:类加载器单元 101、class 文件校验器 102、安全管理器单元 103、基于安全中心的安全执行单元 104 等。

[0022] 类加载器单元 101 提供命名空间和保护域,剔除不可信类,保护可信任类的边界。

[0023] class 文件校验器 102 对载入的 class 文件进行校验,保证 class 文件的字节流符合 class 文件格式规范、字节码的语义描述符合 Java 语言规范的要求、虚拟机的安全运行不会为字节码所影响(如,类型转化是否有效,跳转指令的目标是否有效等)。

[0024] 安全管理器单元 103 定义沙箱的边界,保护虚拟机的外部资源,如网络、I/O、反射等,不被虚拟机内运行的恶意或有漏洞的代码侵犯,确保 Java 应用的行为发生在沙箱之中。

[0025] 作为基于安全中心的安全执行单元的安全监测模块 104 与安全中心进行交互,保证应用、应用所需基础类库、本地共享库的可信性与有效性。通过安全策略的统一分发,保证对系统资源操作的一致性、可控性。

[0026] 本发明实施例公开一种增强 Java 安全机制的方法,换言之,一种增强 Java 虚拟机安全的方法。该方法结合安全中心,对应用、应用的基础类库、第三方类库、以及共享库等进行管理。分别在应用的初始化和运行时两个阶段对 Java 安全模型进行增强。

[0027] 根据本发明实施例增强 Java 虚拟机安全的方法的包括用户端和安全中心交互流程,其示例的主要步骤如下:

[0028] <连接阶段>

[0029] 首先,在用户端(Java 虚拟机)运行应用时,用户端的安全监测模块单元 104 发起向安全中心建立连接请求。

[0030] 安全中心对连接请求进行验证并答复是否建立连接请求。具体地说,连接请求需要安全中心验证用户端是否具备建立连接安全中心的权限,如果用户端的安全等级符合安全中心的认定,则建立连接请求。

[0031] <应用摘要信息的认证>

[0032] 在安全中心通过对连接请求的验证从而建立连接的情况下,用户端的安全监测模块 104 将签名后的应用摘要信息发送至安全中心进行认证。

[0033] 具体地说,用户端的安全监测模块 104 依据安全中心的答复决定是否继续后续的步骤。如果安全中心允许建立连接请求,则发送应用摘要信息;否则,抛异常退出。即,在安全中心没有通过对连接请求的验证从而不建立连接的情况下,用户端的安全监测模块 104 执行抛异常处理并退出当前应用。

[0034] 安全中心根据接收到的签名后的应用摘要信息对此应用进行认证,并答复是否同意用户端运行应用的请求。

[0035] <用户端的基础类库的签名信息的认证>

[0036] 用户端的类加载器单元 101 初始化基础类库,并计算基础类库的签名信息,并将基础类库的签名信息交由安全管理器单元 103 发送至安全中心以请求对基础类库的签名信息进行认证。

[0037] 安全中心对从用户端接收到的针对基础类库的签名信息的认证请求进行验证。

[0038] 在上述步骤中,安全中心主要确认基础类库是否被修改,具体地说,通过判断基础类库的签名信息的版本和安全中心的签名信息的版本是否一致来确认基础类库是否被修改。安全中心对认证请求进行确认,如果符合要求(例如,如果基础类库的签名信息的版本和安全中心的签名信息的版本一致),则通过请求认证;如果不能通过请求,则需要将发送过来的基础类库发送回至用户端。

[0039] <策略文件的验证>

[0040] 用户端向安全中心请求分发安全策略文件。安全中心在接收到分发安全策略文件的请求后验证用户的资格,并据此决定是否发送相关安全策略文件。

[0041] <共享库的认证>

[0042] 用户端向安全中心发送相关共享库的有关认证请求。安全中心在接收到共享库的有关认证请求后需要对共享库的来源、版本进行认证。

[0043] 由此,本发明上述实施例结合安全中心的安全机制在应用初始化和运行时两个阶段对类的安全进行控制,防止可信的类被破坏,保证不可信的类获取规定的权限,保证应用的操作不会突破沙箱的边界;通过安全中心解决使用公共密钥技术时的密码分发。通过统一制定、分发安全策略,保证了安全策略个性化的同时,集中控制应用的基础类库的安全。

[0044] 本发明上述实施例结合安全中心,对应用、应用所使用的基础类库、第三方类库和共享的本地类库进行安全认证,保证应用所涉及的类库本身的安全性;通过在和安全中心交互所获取的类的基本信息,在运行时保证类的元信息不能被修改。同时,有效解决在鉴别和认证过程中的公共密钥技术的密码分发问题,有效地增强了 Java 的安全模型。

[0045] 此外,需要说明的是,除非特别指出,否则说明书中的术语“第一”、“第二”、“第三”等描述仅仅用于区分说明书中的各个组件、元素、步骤等,而不是用于表示各个组件、元素、步骤之间的逻辑关系或者顺序关系等。

[0046] 可以理解的是,虽然本发明已以较佳实施例披露如上,然而上述实施例并非用以限定本发明。对于任何熟悉本领域的技术人员而言,在不脱离本发明技术方案范围情况下,都可利用上述揭示的技术内容对本发明技术方案作出许多可能的变动和修饰,或修改为等同变化的等效实施例。因此,凡是未脱离本发明技术方案的内容,依据本发明的技术实质对以上实施例所做的任何简单修改、等同变化及修饰,均仍属于本发明技术方案保护的范围内。

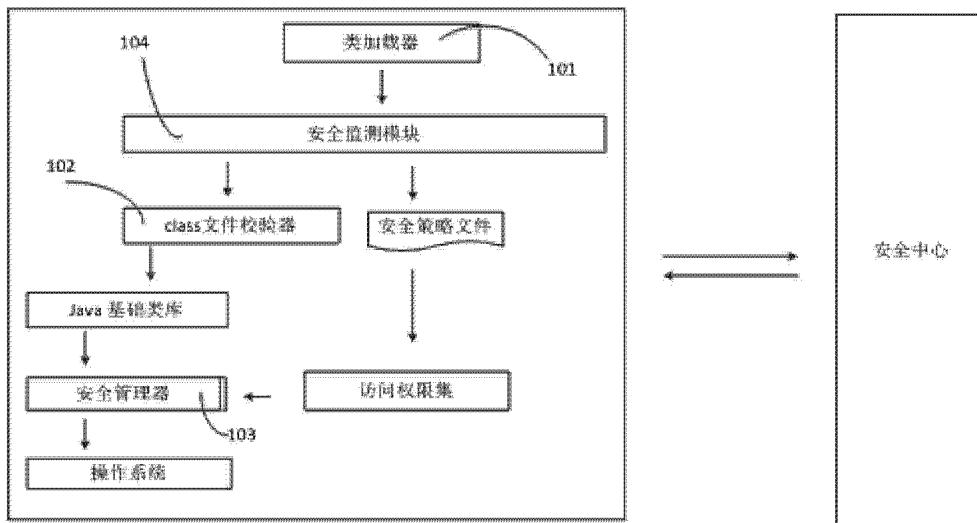


图 1