(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 105593870 A
(43)申请公布日 2016.05.18

(21)申请号 201480054678.6
(22)申请日 2014.09.25
(30)优先权数据
14/046,728 2013.10.04 US
(85)PCT国际申请进入国家阶段日
2016.04.01
(86)PCT国际申请的申请数据
PCT/RO2014/000027 2014.09.25
(87)PCT国际申请的公布数据
WO2015/050469 EN 2015.04.09
(71)申请人 比特梵德知识产权管理有限公司
地址 塞浦路斯尼科西亚
(72)发明人 山多尔·卢卡奇
劳尔-瓦西里·托萨
保罗-丹尼尔·博卡
格奥尔基-弗洛兰·哈嘉玛山
安德烈-弗拉德·卢察什
(74)专利代理机构 北京律盟知识产权代理有限责任公司 11287
代理人 张世俊
(51)Int.Cl.
G06F 21/56(2006.01)

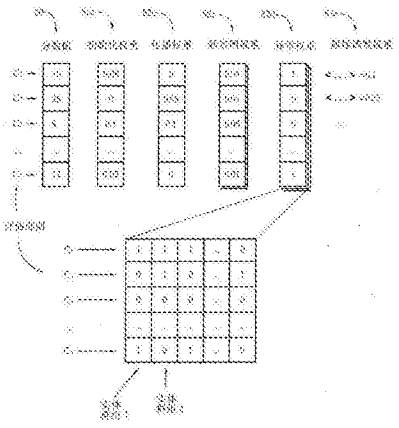
权利要求书3页 说明书14页 附图10页

(54)发明名称

用于恶意软件检测的复杂评分

(57)摘要

所描述系统及方法允许保护计算机系统免受例如病毒、特洛伊木马及间谍软件等恶意软件的影响。对于多个可执行实体（例如在所述计算机系统上执行的进程及线程）中的每一者，评分引擎记录多个评估分数，每一分数是根据相异评估准则确定的。每当实体满足评估准则（例如，执行行动）时，所述实体的所述相应分数便被更新。更新实体的分数可触发对与所述相应实体相关的实体的分数更新，甚至当所述相关实体被终止，即，不再作用时也如此。相关实体尤其包含所述相应实体的父代及/或将代码注入到所述相应实体中的实体。所述评分引擎根据实体的所述多个评估分数来确定所述相应实体是否为恶意的。



CN 105593870 A

1.一种包括至少一个处理器的主机系统，所述至少一个处理器经配置以执行实体管理模块、实体评估器及评分引擎，其中：

所述实体管理模块经配置以管理经评估软件实体的集合，其中管理所述集合包括：

识别所述集合的第一实体的一组后代实体；

确定所述第一实体是否被终止；

作为响应，当所述第一实体被终止时，确定所述组后代实体的所有成员是否均被终止；且

作为响应，当所述组后代实体的所有成员均被终止时，从所述集合移除所述第一实体；

所述实体评估器经配置以：

根据评估准则评估所述第一实体；且

作为响应，当所述第一实体满足所述评估准则时，将评估指示符发射到所述评分引擎；且

所述评分引擎经配置以：

记录针对所述集合的所述第一实体确定的第一分数及针对第二实体确定的第二分数，所述第一及第二分数是根据所述评估准则确定的；

响应于记录所述第一及第二分数，且响应于接收到所述评估指示符，根据所述评估指示符更新所述第二分数；且

作为响应，根据所述所更新第二分数来确定所述第二实体是否为恶意的。

2.根据权利要求1所述的主机系统，其中所述评分引擎进一步经配置以：

响应于接收到所述评估指示符，根据所述评估指示符更新所述第一分数；且

作为响应，根据所述所更新第一分数来确定所述第一实体是否为恶意的。

3.根据权利要求1所述的主机系统，其中所述第一实体为所述第二实体的子代。

4.根据权利要求1所述的主机系统，其中所述第二实体为所述第一实体的子代。

5.根据权利要求1所述的主机系统，其中所述第一实体包括由所述第二实体注入的代码区段。

6.根据权利要求1所述的主机系统，其中所述第二实体包括由所述第一实体注入的代码区段。

7.根据权利要求1所述的主机系统，其中更新所述第二分数包括将所述第二分数改变根据w·S确定的量，其中S为所述第一分数，且其中w为数值权重。

8.根据权利要求1所述的主机系统，其中管理所述经评估软件实体的集合进一步包括：

截获新软件实体的启动；且

作为响应，将所述新软件实体添加到所述集合。

9.一种存储指令的非暂时性计算机可读媒体，所述指令在被执行时配置主机系统的至少一个处理器以：

管理经评估软件实体的集合，其中管理所述集合包括：

识别所述集合的第一实体的一组后代实体；

确定所述第一实体是否被终止；

作为响应，当所述第一实体被终止时，确定所述组后代实体的所有成员是否均被终止；且

作为响应，当所述组后代实体的所有成员均被终止时，从所述集合移除选定实体；

记录针对所述集合的所述第一实体确定的第一分数及针对第二实体确定的第二分数，所述第一及第二分数是根据评估准则确定的；

响应于记录所述第一及第二分数，根据所述评估准则评估所述第一实体；

响应于评估所述第一实体，当所述第一实体满足所述评估准则时，更新所述第二分数；且

响应于更新所述第二分数，根据所述所更新第二分数确定所述第二实体是否为恶意的。

10.根据权利要求9所述的计算机可读媒体，其中所述至少一个处理器进一步经配置以：

响应于评估所述第一实体，当所述第一实体满足所述评估准则时，更新所述第一分数；且

作为响应，根据所述所更新第一分数来确定第一可执行实体是否为恶意的。

11.根据权利要求9所述的计算机可读媒体，其中所述第一实体为所述第二实体的子代。

12.根据权利要求9所述的计算机可读媒体，其中所述第二实体为所述第一实体的子代。

13.根据权利要求9所述的计算机可读媒体，其中所述第一实体包括由所述第二实体注入的代码区段。

14.根据权利要求9所述的计算机可读媒体，其中所述第二实体包括由所述第一实体注入的代码区段。

15.根据权利要求9所述的计算机可读媒体，其中更新所述第二分数包括将所述第二分数改变根据w·S确定的量，其中S为所述第一分数，且其中w为数值权重。

16.根据权利要求9所述的计算机可读媒体，其中管理所述经评估软件实体集合进一步包括：

截获新软件实体的启动；且

作为响应，将所述新软件实体添加到所述集合。

17.一种包括至少一个处理器的主机系统，所述至少一个处理器经配置以执行实体评估器及评分引擎，其中：

所述实体评估器经配置以：

根据评估准则评估第一软件实体，所述第一软件实体是在客户端系统上执行；且

作为响应，当所述第一软件实体满足所述评估准则时，将评估指示符发射到所述评分引擎；且

所述评分引擎经配置以：

响应于接收到所述评估指示符，根据所述评估指示符更新分数，其中所述分数是针对先前在所述主机系统上执行的第二软件实体确定的，所述第二软件实体在更新所述分数时被终止；且

作为响应，根据所述所更新第二分数来确定所述第二软件实体是否为恶意的。

18.根据权利要求17所述的主机系统，其中所述第一软件实体为所述第二软件实体的

子代。

19.根据权利要求17所述的主机系统，其中所述第一软件实体包括由所述第二软件实体注入的代码区段。

20.一种方法，其包括：

采用主机系统的至少一个处理器来确定在所述主机系统上执行的第一软件实体是否满足评估准则；

作为响应，当所述第一软件实体满足所述评估准则时，采用所述至少一个处理器来更新针对先前在所述主机系统上执行的第二软件实体确定的分数，所述第二软件实体在更新所述分数时被终止，所述分数是根据所述评估准则确定的；且

响应于更新所述第二分数，采用所述至少一个处理器来根据所述所更新第二分数确定所述第二软件实体是否为恶意的。

21.根据权利要求20所述的方法，其中所述第一软件实体为所述第二软件实体的子实体。

22.根据权利要求20所述的方法，其中所述第一软件实体包括由所述第二软件实体注入的代码区段。

用于恶意软件检测的复杂评分

背景技术

[0001] 本发明涉及用于保护计算机系统免受恶意软件影响的系统及方法。

[0002] 恶意软件(也被称为恶意软件(malware))在世界范围内影响大量计算机系统。在其许多形式中(例如计算机病毒、蠕虫、隐匿程序(rootkit)及间谍软件),恶意软件给数百万计算机用户呈现严重风险,使其易受数据及敏感信息丢失、身份盗用及生产力损失以及其它。

[0003] 安全软件可用于检测感染用户的计算机系统的恶意软件,且另外用于移除或停止此类恶意软件的执行。所属领域中已知数种恶意软件-检测技术。一些技术依赖于将恶意软件代理的代码片段与恶意软件-指示性特征库进行匹配。其它常规方法检测恶意软件代理的一组恶意软件-指示性行为。

[0004] 为规避检测及/或破坏安全软件的操作,一些恶意软件代理采用混淆技术,例如将其代码加密或在每一所感染计算机系统上使用稍微不同代码版本(多态性)。其它示范性检测规避方法将恶意活动划分为数个行动,每一行动由单独代理执行,可能具有时间延迟。在其它实例中,恶意软件可举例来说通过使用特权提升及/或通过覆写安全软件的代码而试图主动攻击并停用安全软件。

[0005] 为与一组迅速改变的恶意软件威胁保持同步,存在对开发稳健及可扩缩反恶意软件解决方案的强烈关注。

发明内容

[0006] 根据一个方面,一种主机系统包括至少一个处理器,所述至少一个处理器经配置以执行实体管理模块、实体评估器及评分引擎。所述实体管理模块经配置以管理经评估软件实体的集合,其中管理所述集合包括:识别所述集合的第一实体的一组后代实体;确定所述第一实体是否被终止;作为响应,当所述第一实体被终止时,确定所述组后代实体的所有成员是否均被终止;且作为响应,当所述组后代实体的所有成员均被终止时,从所述集合移除所述第一实体。所述实体评估器经配置以:根据评估准则评估所述第一实体;且作为响应,当所述第一实体满足所述评估准则时,将评估指示符发射到所述评分引擎。所述评分引擎经配置以:记录针对所述集合的所述第一实体确定的第一分数及针对第二实体确定的第二分数,所述第一及第二分数是根据所述评估准则确定的;响应于记录所述第一及第二分数,且响应于接收到所述评估指示符,根据所述评估指示符更新所述第二分数;且作为响应,根据所述所更新第二分数来确定所述第二实体是否为恶意的。

[0007] 根据另一方面,一种存储指令的非暂时性计算机可读媒体,所述指令在被执行时配置主机系统的至少一个处理器以管理经评估软件实体的集合,其中管理所述集合包括:识别所述集合的第一实体的一组后代实体;确定所述第一实体是否被终止;作为响应,当所述第一实体被终止时,确定所述组后代实体的所有成员是否均被终止;且作为响应,当所述组后代实体的所有成员均被终止时,从所述集合移除选定实体。所述指令进一步配置所述至少一个处理器以记录针对所述集合的所述第一实体确定的第一分数及针对第二实体确

定的第二分数，所述第一及第二分数是根据评估准则确定的。响应于记录所述第一及第二分数，所述指令进一步配置所述至少一个处理器以根据所述评估准则评估所述第一实体。响应于评估所述第一实体，所述指令进一步配置所述至少一个处理器以在所述第一实体满足所述评估准则时更新所述第二分数，且响应于更新所述第二分数，根据所述所更新第二分数确定所述第二实体是否为恶意的。

[0008] 根据另一方面，一种主机系统包括经配置以执行实体评估器及评分引擎的至少一个处理器。所述实体评估器经配置以：根据评估准则评估第一软件实体，所述第一软件实体是在客户端系统上执行，且作为响应，当所述第一软件实体满足所述评估准则时，将评估指示符发射到所述评分引擎。所述评分引擎经配置以响应于接收到所述评估指示符，根据所述评估指示符更新分数，其中所述分数是针对先前在所述主机系统上执行的第二软件实体确定的，所述第二软件实体在更新所述分数时被终止。所述评分引擎进一步经配置以响应于更新所述第二分数，根据所述所更新第二分数来确定所述第二软件实体是否为恶意的。

[0009] 根据另一方面，一种方法包括采用主机系统的至少一个处理器来确定在所述主机系统上执行的第一软件实体是否满足评估准则。所述方法进一步包括当所述第一软件实体满足所述评估准则时，采用所述至少一个处理器来更新针对先前在所述主机系统上执行的第二软件实体确定的分数，所述第二软件实体在更新所述分数时被终止，所述分数是根据所述评估准则确定的。所述方法进一步包括响应于更新所述第二分数，采用所述至少一个处理器来根据所述所更新第二分数确定所述第二软件实体是否为恶意的。

附图说明

[0010] 在阅读以下详细描述后且在参考图式后，本发明的前述方面及优点将立即变得更好理解，在图式中：

[0011] 图1展示根据本发明的一些实施例的经保护免受恶意软件影响的主机计算机系统的示范性硬件配置。

[0012] 图2-A展示根据本发明的一些实施例的包含在主机系统上执行的安全应用程序的一组示范性软件对象。

[0013] 图2-B展示一组示范性软件对象，所述示范性软件对象包含在经配置以支持虚拟化的主机系统中的虚拟机内执行的安全应用程序。

[0014] 图3图解说明根据本发明的一些实施例的以各种处理器特权层级在主机系统上执行的软件对象的示范性层次，所述软件对象包含一组反恶意软件对象。

[0015] 图4展示根据本发明的一些实施例的由图3的实体管理模块执行的示范性步骤序列。

[0016] 图5展示根据本发明的一些实施例的接收多个实体评估指示符的示范性评分引擎，所述多个实体评估指示符是针对软件实体由多个实体评估器模块确定的。

[0017] 图6图解说明在Windows®环境中的一组进程的示范性执行流程。实线箭头指示在不存在反恶意软件系统的情况下的示范性执行流程。虚线箭头指示对所述执行流程的修改，所述修改由根据本发明的一些实施例操作的多个实体评估器引入。

[0018] 图7展示根据本发明的一些实施例的由实体评估器模块执行的示范性步骤序列。

[0019] 图8展示根据本发明的一些实施例的多个示范性实体评分对象(ESO)，每一ESO是

针对相应软件实体确定的。ESO的示范性数据字段包含实体身份指示符EID、多个分数S_i及针对相应实体确定的聚合分数A以及其它。

[0020] 图9图解说明根据本发明的一些实施例的一组示范性分数值，及由评分引擎使用以对软件实体进行评分的各个组示范性权重。

[0021] 图10展示根据本发明的一些实施例的由评分引擎执行(图3到4)的示范性步骤序列。

[0022] 图11图解说明包括经由计算机网络连接到安全服务器的多个主机系统的示范性配置。

[0023] 图12展示根据本发明的一些实施例的主机系统与安全服务器之间的示范性反恶意软件交易。

具体实施方式

[0024] 在以下描述中，应理解，结构之间的所有所引用连接可为直接操作连接或通过中间结构的间接操作连接。一组元素包含一个或多个元素。对元素的任何引用应理解为指代至少一个元素。多个元素包含至少两个元素。除非另有需要，否则任何所描述方法步骤不需要一定以特定所图解说明次序执行。第一元素(例如数据)从第二元素导出涵盖第一元素等于第二元素，以及通过处理第二元素及任选地其它数据产生第一元素。根据参数做出确定或决策涵盖根据所述参数及任选地根据其它数据做出确定或决策。除非另外规定，否则一些数量/数据的指示符可为数量/数据本身，或与所述数量/数据本身不同的指示符。除非另外规定，否则进程表示计算机程序的实例，其中计算机程序为确定计算机系统执行规定任务的指令序列。计算机可读媒体涵盖非暂时性媒体(例如磁性媒体、光学媒体及半导体存储媒体(例如硬驱动器、光盘、快闪存储器、DRAM))以及通信链路(例如导电电缆及光纤光学链路)。根据一些实施例，本发明尤其提供计算机系统，所述计算机系统包括经编程以执行本文中所描述的方法的硬件(例如一个或多个处理器)，以及用以执行本文中所描述的方法的计算机可读媒体编码指令。

[0025] 以下描述以实例方式且未必以限制方式图解说明本发明的实施例。

[0026] 图1展示根据本发明的一些实施例的执行反恶意软件操作的主机系统10的示范性硬件配置。主机系统10可表示例如企业服务器的公司计算装置或者例如个人计算机或智能电话的最终用户装置以及其它。其它主机系统包含例如TV及游戏控制台的娱乐装置，或者具有支持虚拟化且需要恶意软件保护的存储器及处理器的任何其它装置。图1出于说明性目的展示计算机系统；例如移动电话或平板计算机的其它客户端装置可具有不同配置。在一些实施例中，系统10包括一组物理装置，所述物理装置包含处理器12、存储器单元14、一组输入装置16、一组输出装置18、一组存储装置20及一组网络适配器22，所述物理装置全部由一组总线24连接。

[0027] 在一些实施例中，处理器12包括经配置以用一组信号及/或数据执行计算及/或逻辑操作的物理装置(例如多核集成电路)。在一些实施例中，此类逻辑操作以处理器指令(例如机器代码或其它类型的软件)序列的形式递送到处理器12。存储器单元14可包括易失性计算机可读媒体(例如RAM)，所述易失性计算机可读媒体存储在实行指令的过程中由处理器12存取或产生的数据/信号。输入装置16可包含计算机键盘、鼠标及麦克风以及其它，其

包含允许用户将数据及/或指令引入到系统10中的相应硬件接口及/或适配器。输出装置18可包含允许系统10将数据传递到用户的显示装置(例如监视器及扬声器以及其它)以及硬件接口/适配器(例如图形卡)。在一些实施例中,输入装置16及输出装置18可共享一件共同硬件,如在触摸屏装置的情形中。存储装置20包含实现软件指令及/或数据的非易失性存储、读取及写入的计算机可读媒体。示范性存储装置20包含磁盘及光盘及快闪存储器装置以及可拆卸媒体(例如CD及/或DVD磁盘及驱动器)。所述组网络适配器22使系统10能够连接到计算机网络及/或连接到其它装置/计算机系统。总线24共同地表示实现主机系统10的装置12到22的相互通信的所述多个系统、外围装置及芯片集总线及/或所有其它电路。举例来说,总线24可包括将处理器12连接到存储器14的北桥(northbridge),及/或将处理器12连接到装置16到22的南桥(southbridge)以及其它。

[0028] 图2-A展示在不采用硬件虚拟化的配置中的主机系统10上执行的一组示范性软件对象。在一些实施例中,客体操作系统(OS)34包括软件,所述软件提供通向主机系统10的硬件的接口,且为一组软件应用程序42a到42c及44充当主机。OS 34可包括任何可广泛获得的操作系统,例如Windows®、MacOS®、Linux®、iOS®或Android™以及其它。应用程序42a到42c可包含字处理、图像处理、数据库、浏览器及电子通信应用程序以及其它。

[0029] 图2-B在使用硬件虚拟化的实施例中展示在主机系统10上执行的一组示范性软件对象。由虚拟机监控程序30暴露一组客体虚拟机32a到32b。虚拟机(VM)在所属领域中通常被称为实际物理机/计算机系统的软件仿真,其各自能够独立于其它VM运行其自身操作系统及软件。虚拟机监控程序30包括允许由主机系统10的硬件资源的多个虚拟机(例如处理器操作、存储器、存储装置、输入/输出及联网装置)进行多路复用(共享)的软件。在一些实施例中,虚拟机监控程序30使多个虚拟机及/或操作系统(OS)在具有各种程度的隔离的情况下同时在主机系统10上运行。为实现此类配置,形成虚拟机监控程序30的一部分的软件可创建多个虚拟化(即,软件仿真)装置,每一虚拟化装置仿真系统10的物理硬件装置(例如处理器12及存储器14以及其它),虚拟机监控程序30可进一步将一组虚拟装置指派给在主机系统10上操作的每一VM。因此,每一VM 32a到32b均操作如同其拥有其自身组的物理装置,即,作为较完整或较不完整的计算机系统。虚拟装置到虚拟机的创建及指派在所属领域中通常被称为暴露相应VM。流行虚拟机监控程序的实例包含来自威睿(Vmware)公司的VMware vSphere™及开源代码Xen虚拟机监控程序以及其它。

[0030] 在一些实施例中,虚拟机监控程序30包含存储器自检引擎40,所述存储器自检引擎经配置以执行如下文进一步描述的反恶意软件操作的。引擎40可并入到虚拟机监控程序30中,或可作为与虚拟机监控程序30相异且从虚拟机监控程序30独立但以与虚拟机监控程序30基本上类似的处理器特权层级执行的软件组件而递送。单个引擎40可经配置以对在主机系统10上执行的多个VM进行恶意软件保护。

[0031] 虽然图2-B为简单起见仅展示两个VM 32a到32b,但主机系统10可同时操作大量(例如,数以百计)VM,且此类VM的数目可在主机系统10的操作期间改变。在一些实施例中,每一VM 32a到32b同时且独立于在主机系统10上运行的其它VM分别执行客体操作系统34a到34b及/或一组软件应用程序42d到42e及42f。每一OS 34a到34b包括软件,所述软件提供到相应VM 32a到32b的(虚拟化)硬件的接口且充当用于计算在相应OS上执行的应用程序的主机。

[0032] 在一些实施例中，安全应用程序44经配置以执行如下文所详细描述的反恶意软件操作以保护主机系统10免受恶意软件的影响。在图2-B的实例中，应用程序44的实例可在每一VM 32a到32b上执行，每一此类实例均经配置以保护相应虚拟机。安全应用程序44可为独立程序，或可形成尤其包括反恶意软件、反垃圾邮件及反间谍软件组件以及其它的软件套件的一部分。

[0033] 图3图解说明根据本发明的一些实施例的在主机系统10上执行的软件对象的层次。图3展示经配置以在虚拟化环境中执行的示范性实施例；所属领域的技术人员可清楚，所图解说明实施例可经修改以直接执行于主机系统10上而非执行于VM 32内。图3是从处理器特权层级(在所属领域中也被称为层或保护环)的角度来表示。在一些实施例中，每一此类层或保护环均由一组指令表征，以相应处理器特权层级执行的软件对象经允许以执行所述组指令。当软件对象试图执行指令时(此在相应特权层级内不被允许)，所述尝试可触发处理器事件，例如异常、故障或虚拟机退出事件。在一些实施例中，可经由一组专用指令实现特权层级之间的切换。此类示范性指令包含SYSCALL/SYSENTER(其从用户层级切换到内核层级)、SYSRET/SYSEXIT(其从内核层级切换到用户层级)、VMCALL(其从用户层级或内核层级切换到根层级)及VMRESUME(其从根层级切换到内核层级或用户层级)。

[0034] 操作系统34的大部分组件以在所属领域中被称为内核层级或内核模式(例如，因特尔(Intel)平台上的环0)的处理器特权层级执行。应用程序42g以比OS 34弱的处理器特权(例如，环3或用户模式)执行。在支持虚拟化的实施例中，虚拟机监控程序30以最具特权的层级(也被称为根层级或根模式)(例如，环-1或Intel®平台上的VMX根)控制处理器12，从而将虚拟机32暴露于OS 34及例如应用程序42g的其它软件对象。

[0035] 在一些实施例中，安全应用程序44的部件可以用户层级的处理器特权(即，与应用程序42g相同的层级)执行。举例来说，此类部件可包括图形用户接口，其告知用户在相应VM上检测到的任何恶意软件或安全威胁，且从用户接收指示应用程序44的(例如)所要配置选项的输入。以用户层级执行的组件的另一实例为用户层级实体评估器50a，其如下文所详细描述而操作。在一些实施例中，用户层级实体评估器50a的一部分可在安全应用程序44内操作，而另一部分(例如挂钩模块)可在经评估应用程序(例如应用程序42g)内操作。应用程序44的其它部件可以内核特权层级执行。举例来说，应用程序44可安装反恶意软件驱动程序36、实体管理模块37及评分引擎38，其全部以内核模式操作。驱动程序36给反恶意软件应用程序44提供功能性(例如)以扫描存储器寻找恶意软件特征及/或检测进程及/或在OS 34上执行的其它软件对象的恶意软件-指示性的行为。在一些实施例中，反恶意软件驱动程序36包含内核层级实体评估器50b，其如下文所详细描述而操作。

[0036] 在一些实施例中，实体管理模块37管理在主机系统10(或VM 32)内执行的软件实体的集合。在一些实施例中，所述集合包括由例如55a到55b的实体评估模块评估所有实体以寻找恶意软件。为管理所述集合，模块37可响应于检测到寿命周期事件(例如实体启动及/或终止事件)的发生而添加及/或从所述集合移除实体，如在下文模式细节中所展示。模块37可进一步确定实体间关系，例如确定父实体的若干子实体(例如，子进程)，及/或确定选定实体是否已将软件对象(例如库)注入到另一实体中，或所述选定实体是否为由另一软件实体进行的注入的目标。子实体为由称作父实体的另一可执行实体创建的可执行实体，所述子实体独立于父实体执行。示范性子实体为子进程，举例来说，所述子进程是经由

Windows® OS的CreateProcess函数，或经由Linux®中的叉式(fork)机制创建。代码注入为所属领域中用以指示将代码序列(例如，动态链接库(DLL))引入到现有进程的存储器空间中以更改相应进程的原始功能性的方法族的通用术语。为执行例如检测进程的启动及/或检测代码注入等任务，模块37可采用所属领域中已知的任何方法，例如调用或挂起特定OS功能。举例来说，在运行Windows® OS的系统中，模块37可注册PsSetCreateProcessNotify例程回调以检测新进程的启动，及/或挂起CreateRemote线程函数以检测所注入代码的执行。

[0037] 图4展示根据本发明的一些实施例的由实体管理模块37执行的示范性步骤序列。在步骤序列250到252中，模块37使用(举例来说)上文所描述的方法截获实体寿命周期事件。当此事件已发生时，步骤254识别触发相应事件的实体。步骤258可包含确定相应实体的唯一实体识别指示符(EID)；此指示符可在对相应实体进行评分时使用，如在下文进一步展示。步骤256确定事件是否包括新实体(例如，新进程)的启动，且当不包括时，模块37前进到步骤260。当事件包括启动时，在步骤258中，模块37可将触发实体添加到经评估实体的集合。步骤260包括确定事件是否包括父实体繁衍子实体，且当不包括时，模块37可前进到步骤264。当包括时，在步骤262中，模块37可将相应子实体添加到经评估实体的集合。步骤262可进一步包含确定子实体的EID，及将触发实体与子实体之间的关系注册为父子(父代-子代)关系。

[0038] 在一些实施例中，步骤264确定事件是否包括代码的注入，且当不包括时，模块37可前进到步骤268。当包括时，模块37可识别代码注入的源实体及目标实体，其中所述源实体将代码注入到目标实体中。在步骤266中，模块37可注册源实体与目标实体之间的代码-注入类型的关系。

[0039] 在步骤268中，实体管理模块37确定事件是否包括触发实体的终止，且当不包括时，模块37返回到步骤250。在一些实施例中，当相应实体的所有组件已完成执行时，实体被视为已终止。举例来说，进程在相应进程的所有线程已完成执行时终止。当事件包括触发实体的终止时，在步骤270中，模块37可确定触发实体的一组后代实体。在一些实施例中，触发实体的后代实体包含相应实体的子实体，以及子实体的子实体(跨多代)。在一些实施例中，后代实体可递归地包含包括由触发实体注入的代码的目标实体，以及由目标实体定为目标的实体。在步骤272中，模块37可确定所述组后代实体的所有实体是否终止，且当不终止时，执行返回到步骤250。当所有后代均终止时，在步骤274中，实体管理模块37可从经评估实体的集合移除触发实体。

[0040] 在一些实施例中，评分引擎38经配置以从例如评估器50a到50b的多个实体评估器模块接收数据(所述数据是针对经评估软件实体确定的)，且根据相应数据确定所述相应实体是否为恶意的。在一些实施例中，由评分引擎38分析的软件实体尤其包含可执行对象，例如进程及执行线程。进程为计算机程序(例如应用程序或操作系统的一部分)的实例且通过具有至少一执行线程及由操作系统指派给所述进程的虚拟存储器的区段来表征，所述相应区段包括可执行代码。在一些实施例中，经评估软件实体可基本上在范围及复杂性方面变化，举例来说从个别线程变化到个别应用程序，变化到操作系统及/或虚拟机的整个实例。

[0041] 图5展示接收多个评估指示符52a到52d的示范性评分引擎38，每一指示符52a到52d是由实体评估器确定。在图5中，此类评估器包含用户层级实体评估器50a、内核层级实

体评估器50b及系统调用评估器50c以及其它。每一此类评估器模块可独立于其它评估器执行，且每一评估器模块可确定经评估软件实体的多个相异实体评估指示符。在实施硬件虚拟化的系统中，一些评估指示符(例如图5中的指示符52a到52c)由在VM 32内执行的组件确定，而其它评估指示符(例如52d)由在VM 32外部执行的组件(举例来说，由存储器自检引擎40)确定。在一些实施例中，每一评估指示符52a到52d均包括实体识别指示符，从而允许引擎38将相应评估指示符与确定评估指示符的软件实体唯一地相关联。

[0042] 一些评估指示符可为恶意软件-指示性的，即，可指示经评估实体为恶意的。一些评估指示符本身可并非恶意软件-指示性，但其可在与其它评估指示符组合时指示恶意。每一评估指示符52a到52d均可根据相异方法及/或准则确定。示范性评估准则包含行为准则，例如确定经评估实体是否执行特定行动(例如写入到磁盘文件)，编辑VM 32的系统注册密钥，或写入到属于被保护软件对象的存储器页。另一示范性准则可包含确定存储器的属于经评估实体的区段是否含有恶意软件-指示性特征。

[0043] 为图解说明实体评估器50a到50c的操作，图6展示根据本发明的一些实施例的一组软件实体70a到70b的示范性执行流程。为简单起见，所挑选实体70a到70b为在Windows®OS的实例中执行的进程；举例来说，可再现用于其它操作系统(例如Linux)的类似图式。实线箭头表示在不存在实体评估器的情况下(例如，在不存在安全应用程序44的情况下)的执行流程。虚线箭头表示由于根据本发明的一些实施例执行的实体评估器50a到50c的存在所致的对所述流程的修改。

[0044] 进程70a加载多个动态链接库(DLL)72a到72c；在图6的实例中，DLL 72c通过(可能为恶意的)进程70b而注入到进程70a中。当进程70a(或其所加载DLL中的一者)执行需要某一系统功能性(例如将某物写入到磁盘文件或编辑注册表项)的指令时，所述相应指令调用用户-模式API(例如KERNEL32.DLL或NTDLL.DLL)。在图6的实例中，相应用户-模式API调用由用户层级行为过滤器50a截获并分析。此类截获可通过例如DLL注入或挂起以及其它的方法实现。挂起为所属领域中使用的针对截获函数调用或消息或在软件组件之间传递的事件的方法的通用术语。一种示范性挂起方法包括通过插入将执行重新引导到第二函数的指令来更改目标函数的入口点。在此挂起之后，第二函数可替代目标函数或在目标函数之前执行。在图6的实例中，反恶意软件驱动程序36可挂起到KERNEL32.DLL或NTDLL.DLL的特定函数中以指示相应函数将执行重新引导到过滤器50a。因此，过滤器50a可检测到进程70a正试图执行根据所挂起函数识别的特定行为。当过滤器50a检测到此类行为时，过滤器50可制定评估指示符52a且将指示符52a发射到评分引擎38(例如，参见图5)。

[0045] 在典型执行流程中，由实体70a调用的用户-模式API函数可从操作系统的内核请求服务。在一些实施例中，通过在x86平台上发布系统调用(例如SYSCALL及SYSENTER)来进行此类操作。在图6的实例中，此类系统调用由系统调用重新评估器50c截获。在一些实施例中，此类截获举例来说包括通过改变存储在处理器12的模型特有寄存器(MSR)中的值来修改系统调用处理程序例程，此有效地将执行重新引导到过滤器50c。此类技术在所属领域中被称为MSR挂起，且可允许系统调用评估器50c检测经评估进程正试图执行特定系统调用。当截获此类系统调用时，系统调用过滤器50c可制定实体评估指示符52c且将指示符52c发射到评分引擎38。

[0046] 在系统调用之后，通常将对处理器的控制转交给OS 34的内核。在一些实施例中，

内核层级实体评估器50b经配置以截获OS内核的特定操作且因此确定经评估进程正试图执行可为恶意的特定操作。为截获此类操作，一些实施例可采用构建到OS 34中且由OS 34暴露的一组过滤机制。举例来说，在Windows OS中，可使用FltRegisterFilter来截获比如创建文件、打开文件、写入到文件及删除文件的操作。在另一实例中，评估器50b可使用ObRegisterCallback来截获、创建或复制对象-处理操作，或使用PsSetCreateProcessNotifyRoutine来截获新进程的创建。在又一实例中，可使用CmRegisterCallbackEx来截获例如创建及设定注册表项/值的Windows注册表操作。所属领域中已知用于其它操作系统(例如Linux®)的类似过滤机制。当内核-模式实体评估器50b截获此类操作时，评估器50b可制定实体评估指示符52b且将指示符52b发射到评分引擎38。

[0047] 为将例如实体评估指示符52a到52c的数据从评估器50a到50c发射到评分引擎38，所属领域的技术人员可采用任何进程间通信方法。举例来说，为在用户-模式组件与内核-模式组件之间通信，评估器50a到50c及引擎38可经配置以使用共享的存储器区段。当于在VM 32内执行的组件与在相应VM外部执行的组件之间需要数据交换时，可使用虚拟化领域中已知的任何方法进行此通信。举例来说，为将评估指示符52d从存储器自检引擎40发射到评分引擎38，一些实施例使用中断注入机制发信号到引擎38，使得从相应VM外部发射数据。举例来说，可通过上文所描述的共享存储器区段传送实际数据。

[0048] 图7展示根据本发明的一些实施例的由实体评估器(例如图4到5中的评估器50a到50c及/或存储器自检引擎40)执行的示范性步骤序列。在步骤序列302到304中，实体评估器等待主机系统10及/或虚拟机32内的触发事件的发生。示范性触发事件尤其包含软件实体执行特定行为(例如发布特定处理器指令)，试图使用一件特定硬件(例如存储装置20或网络适配器22)或试图写入到受保护存储器页。举例来说，对于评估器50c的触发事件可包含软件实体发布系统调用(例如，SYSENTER)。另一对于评估器50d的触发事件的实例可包含应用程序调用Url下载To文件API函数。为检测触发事件的发生，相应实体评估器可使用所属领域中已知的任何方法，例如代码注入及MSR挂起以及其它。上文关于图6描述触发事件截获的一些实例。

[0049] 当检测到触发事件时，在步骤306中，实体评估器可识别导致相应触发事件的软件实体(例如，进程)。在一些实施例中，实体评估器可依据由OS 34使用以表示目前正在执行中的每一进程及/或线程的数据结构来确定软件实体的身份。举例来说，在Windows中，每一进程均表示为执行体进程块(E进程)，其尤其包括对相应进程的线程中的每一者的处理及允许OS 34从多个执行进程识别相应进程的唯一进程ID。类似进程/线程表示可用于例如Linux的其它OS。

[0050] 在步骤308中，实体评估器可制定评估指示符，包含相应软件实体的识别符(例如，进程ID)及由相应软件实体执行且在步骤302到304中截获的行动/事件的种类的指示符。在一些实施例中，实体评估器可依据所截获触发事件的参数确定相应软件实体的行动及/或行为的类型。在操作的实例中，当进程试图从因特网下载文件时，用户层级实体评估器50a可截获所述尝试。除识别哪一进程正执行行动外，评估器50a还可确定行动(下载文件)的类型、从其下载文件的IP地址及所下载文件的磁盘地址，以及其它。此类数据可选择性地并入到评估指示符中，从而允许评分引擎38借助参数Z确定实体X已执行行动Y。在步骤310中，实体评估器将评估指示符发射到评分引擎38。

[0051] 在一些实施例中，评分引擎38及/或实体管理模块37维持经评估软件实体(例如在主机系统10(或VM 32)上执行的进程及线程)的集中式知识库。图8展示一组经评估实体70c到70e，每一经评估实体分别表示为示范性实体评分对象(ESO)74a到74c。每一ESO包括多个数据字段，所述多个数据字段中的一些数据字段在图8中图解说明。此类字段可包含唯一实体识别符(EID)76a、多个评估分数76b及聚合分数76d。在一些实施例中，评估分数76b由引擎38根据从个别实体评估器接收的评估指示符52a到52d确定。每一此类分数可根据由指示符76c识别的评估准则确定。在一些实施例中，评估分数76b与评估准则76c具有一对一对应性，使得根据相应准则赋予每一分数。举例来说，特定准则C_k可包括确定经评估实体是否从计算机网络(例如因特网)下载文件。在一个此类实例中，可仅在经评估实体试图进行下载的情况下奖励相应分数S_k。

[0052] 在一些实施例中，ESO 74a可进一步包括一组旗标76e。一些旗标76e可为二进制指示符(例如，0/1、是/否)。在一个此类实例中，旗标指示相应经评估实体E_1是否满足特定评估准则(例如，E_1是否是从因特网下载的可执行文件，E_1是否以命令线模式运行等)。另一示范性旗标指示实体E_1的分类，例如，E_1属于特定对象类别(例如特洛伊木马(Trojan)恶意软件、浏览器对象、PDF读取器应用程序等)的指示符。旗标的示范性用途包括其中实体E_1的评估分数S_i的更新触发E_1的另一评估分数S_j的更新的情况(参见下文)。旗标可用于打开及关闭此类共更新机制。举例来说，可已知，当E_1满足评估准则C_i(例如，如果实体执行特定行动)时，实体E_1也有可能满足准则C_j。因此，可为实体E_1设定指示关系$<C_i,C_j>$的旗标F_1，从而在分数S_i被更新时触发分数S_j的更新。

[0053] ESO 74a可进一步包含终止指示符76f，其指示相应实体目前是作用的还是已终止。此类终止指示符可允许评分引擎38记录及/或更新已终止实体的分数。ESO 74a可进一步包含与相应实体E_1相关的一组软件实体识别符。此类相关软件实体的实例可包括E_1的父实体(由识别符76g表示)，及E_1的一组子实体(由识别符76h表示)。ESO 74a可进一步包括一组注入目标实体指示符(项76j)(其识别E_1已将代码注入到其中的软件实体)及又一组注入源实体指示符(项76k)(其识别将代码注入到E_1中的软件实体)。

[0054] 在一些实施例中，对经评估软件实体的评分根据一组分数值且进一步根据额外参数继续进行。图9图解说明此类数据，其中一组分数值由项80表示。分数值由其对应评估准则C_1、…C_n索引。每一此类值可表示(举例来说)如果经评估实体满足相应评估准则(例如，如果其从因特网下载文件，如果其写入到MSWord®文档等)那么其所接收的预定数目个点。

[0055] 控制评分的示范性参数包含一组初始化权重82a、一组传播权重82b、一组新的实例权重82c、一组异常权重82d及一组旗标诱发权重82e。权重82a到82e由评估准则C_1、…C_n索引。一些类型的权重与评估准则一对一对应，使得对于每一C_i均存在一个权重值w_i。其它类型的权重与评估准则一对多对应。一个此类实例为图9中的异常权重82d，其中可存在对应于特定评估准则C_i的多个权重w_{ij}。权重可按实体的种类或类别分组，如由图9的实例所图解说明；举例来说，可存在可适用于字处理应用程序(例如，MSWord®)的第一权重值，可适用于web浏览器(例如，Firefox®及MS InternetExpolrer®)的第二权重值(可能相异于所述第一权重值)，及可适用于文件管理器应用程序(例如，Windows Explorer®)的第三权重值。在不同类别的实体当中加以区分可为有用的，这是因为一些评估准则可对一种类别的实体比

对其它实体更具恶意软件-指示性。更一般来说，每一评分权重可由元组<C_i,E_k,…>索引，其中C_i表示特定评估准则，且其中E_k表示特定经评估实体。用于存储及存取评分权重82a到82e的实际数据格式可在实施例当中变化。权重82a到82e可存储为矩阵、列表、关系数据库(RDB)或可扩展标记语言(XML)结构以及其它。将在下文讨论用于评分的权重的示范性用途。

[0056] (举例来说)由人类操作者预定分数值80及/或权重82a到82e。在一些实施例中，此类值可随时间改变，且可经调整以优化恶意软件检测。可将所更新分数值及/或权重值递送到主机系统10作为来自安全服务器的周期性及/或按需软件更新(参见下文，关于图10到11)。

[0057] 图10展示根据本发明的一些实施例的由评分引擎38执行的示范性步骤序列。在步骤302中，引擎38从实体评估器(举例来说，图5中的评估器50a到50c中的一者)接收实体评估指示符。在实施硬件虚拟化的一些实施例中，引擎38可从在相应虚拟机(例如，图5中的存储器自检引擎40)的外部执行的组件接收相应实体评估指示符。在步骤304中，引擎38可(例如)根据嵌入于相应评估指示符中的实体ID识别确定其相应实体评估指示符的软件实体(参见上文，关于图7)。

[0058] 接下来，评分引擎38针对在步骤304中识别的实体E以及针对与E相关的其它实体执行步骤框318到332。此类相关实体可包含E的父实体及子实体，E已将代码注入到其中的注入目标实体，及已将代码注入到E中的注入源实体，以及其它。以此方式，每次引擎38接收评估指示符(举例来说，指示实体E已执行特定行动)时，框318到332便可执行数次，从而不仅更新实体E的评估分数，而且更新与E相关的实体的评估分数。在一些实施例中，框318到332对E执行一次且对与E相关的每一实体E*执行一次。在替代实施例中，递归地执行框318到332，直到满足一些收敛准则为止。示范性收敛准则包括验证E及/或E*的评估分数是否在框318到332的连续执行之间改变，及当不存在此类改变时退出。在图10的示范性算法中，变量X用于指示目前正经历分数更新的实体。在步骤316中，将X设定到在步骤304中识别的实体E。

[0059] 在步骤318中，引擎38更新实体X(例如，图8中的实体76b)的评估分数。在一些实施例中，更新评估分数包括用新的值替换相应评估分数的所记录值：

[0060] $S_k^{(X)} \rightarrow S_k^{(X)} + \Delta S_k$， [1]

[0061] 其中$S_k^{(X)}$表示根据评估准则C_k针对实体X确定的评估分数，且其中ΔS_k表示增量，所述增量可为正或负(在一些实施例中，评估分数可在更新后即刻降低)。

[0062] 在一些实施例中，由评分引擎38根据在步骤312中接收的评估指示符确定分数增量ΔS_k。相应指示符可包含分数及/或指示在确定相应指示符时使用的评估准则。在一些实施例中，评分引擎38根据对应于相应评估准则C_k的分数值80(参见图9)确定分数增量ΔS_k，举例来说：

[0063] $\Delta S_k = V_k$ [2]

[0064] 其中V_k表示指派给准则C_k的分数值80。在一个此类实例中，其中准则C_k包括确定实体X是否从网络下载对象，且其中$V_k = 20$，每当实体X执行下载时，评估分数$S_k^{(X)}$便将增加20点。在一些实施例中，$\Delta S_k = \varepsilon V_k$，其中$\varepsilon$为二进制异常权重(例如，参见图9中的项82d)，迫使仅对经评估实体的子组更新分数S_k。此类异常权重可用于(举例来说)在各种类型的经评估

实体之间加以区分。举例来说，应允许浏览器接入无限数目个IP地址而不引起对恶意软件的怀疑；通过将浏览器类型的实体的异常权重设定为0，同时将其它类型的实体的异常权重保持为作用的(ε＝1)，可针对浏览器对象有效地关闭包含检测因特网接入的评估准则。

[0065] 在一些实施例中，根据针对与X相关的实体X*确定的评估分数确定在更新实体X的评估分数时使用的分数增量 ΔS_k，即，分数可从一个实体传播到相关实体，例如从子代传播到父代，或从注入目标传播到注入的源目标。在一个此类实例中，由子进程执行的行动可不仅触发执行所述行动(所述子进程)的实体的分数的更新，而且触发相应子进程的父进程的分数的更新。此类分数更新可包括根据以下方程式计算分数增量：

[0066] $$\Delta S_k = w_k S_k^{(X^*)}, \quad [3]$$

[0067] 其中 w_k 表示数值、准则特有权重，其指示实体X*的分数影响实体X的分数的强度。权重 w_k 可包含传播权重82b(图9)。一些实施例在多种此类传播权重当中加以区分，举例来说，用以将分数从子实体传播到父实体的权重的值可与用以将分数从父实体传播到子实体的权重不同。类似地，用以将分数从子实体传播到父实体的权重的值可与用以将分数从成为代码注入的目标的实体传播到执行代码注入的实体的权重不同。在一些实施例中，分数可从作用实体传播到已终止实体。举例来说，子进程的行动可使父进程的分数递增，甚至当父进程被终止时也如此。

[0068] 在一些实施例中，方程式[3]中的实体X*为实体X的另一实例。举例来说，X及X*可为同时执行的同一进程或线程的副本。在此类情形中，权重 w_k 可为新的实例权重(例如，图9中的项82c)或初始化权重(例如，项82a)。在一些实施例中，当启动实体X的新实例X'时，引擎38可使用新的实例权重 w_k 以将分数从X传播到X'来更新现有实体X的一些或所有评估分数。类似地，当启动X'时，引擎38可使用初始化权重 w_k 将分数从已执行的实体X传播到新实体X'来更新X'的一些或所有评估分数。

[0069] 在一些实施例中，更新评估分数 S_k 可触发相应实体的相异评估分数 S_m 的更新。举例来说，

[0070] $$S_k^{(X)} \rightarrow S_k^{(X)} + V_k \text{ 触发 } S_m^{(X)} \rightarrow S_m^{(x)} + F^{(X)} f_{km} V_m, \quad [4]$$

[0071] 其中 $F^{(X)}$ 为针对实体X设定的旗标(例如，参见图8中的项76e)，所述旗标指示评估准则 C_k 与 C_m 之间的联系，且其中 f_{km} 为旗标诱发权重(例如，参见图9中的项82e)，其指示实体X的 S_k 的更新对 S_m 的更新的影响强度。

[0072] 在步骤320中，评分引擎38可根据在步骤312中接收的评估指示符更新实体X的旗标(参见上文关于图8的关于旗标的讨论)。旗标可经设定以激活及/或去激活分数共更新机制，例如上文关于方程式[4]所描述。在一个此类实例中，经评估实体可根据评估指示符(步骤312)识别为web浏览器应用程序；此识别应对评分引擎38指示不对相应实体进行评分以用于未来从因特网下载。此可通过将针对相应实体的特定旗标F的值设定为0来实现，其中当实体从因特网下载对象时，旗标F对评分引擎38指示更新相应实体的评估分数。

[0073] 在步骤322中，评分引擎38可通过组合针对相应进程确定的个别评估分数来将实体X的聚合分数确定(举例来说)为和：

[0074] $$A^{(X)} = \sum_k S_k^{(X)} \quad [5]$$

[0075] 在步骤324中，引擎38可将聚合分数与预定阈值进行比较。当聚合分数不超过阈值

时，评分引擎38可继续进行到下文所描述的步骤326。在一些实施例中，可将阈值设定为根据用户输入确定的值。此类阈值可反映相应用户的安全偏好。举例来说，当用户选择严密安全时，可将所述阈值设定为相对低值；当用户偏好更具容忍性安全设定时，可将所述阈值设定为相对高值。在一些实施例中，可从远程安全服务器接收所述阈值，如下文关于图10到11所描述。

[0076] 在一些实施例中，在步骤322到324中，评分引擎38可确定多个聚合分数，且将每一聚合分数与(可能相异的)阈值进行比较。可根据评估分数的相异子组确定每一此聚合分数。在示范性实施例中，每一此分数子组及其对应评估准则子组可表示特定种类或类型的恶意软件(例如，特洛伊木马、隐匿程序等)。此可允许引擎38执行对所检测恶意软件的分类。在另一实施例中，评分引擎38采用多个阈值以便根据各种恶意程度(例如清白、可疑、危险及致命)将执行实体分类。

[0077] 当聚合分数超过阈值时，在步骤326中，引擎38可决定经评估进程为恶意的，且可采取反恶意软件行动。在一些实施例中，此类反恶意软件行动可尤其包含终止经评估进程、隔离经评估进程以及移除或停用经评估进程的资源(例如文件或存储器的区段)。在一些实施例中，反恶意软件行动可进一步包括(举例来说)通过经由连接到主机系统10(经由网络适配器22)的计算机网络将消息发送到系统管理员来警示主机系统10的用户及/或警示系统管理员。在一些实施例中，反恶意软件行动还可包括将安全报告发送到远程安全服务器，如下文关于图10到11所描述。

[0078] 在步骤序列328到330中，引擎38可识别与X相关的实体X*，其中X*的分数需要在X的目前分数更新之后更新。举例来说，X*可为X的父实体或子实体。在一些实施例中，可根据实体X的ESO的字段76g到76k识别实体X*(例如，参见图8)。当无此类实体X*存在时，或当所有此类实体X*均被视为已进行分数更新时，引擎38返回到步骤312。当存在至少一实体X*时，在步骤332中，评分引擎使X*成为目前实体并返回到步骤318。

[0079] 图3到4中所描绘的示范性评分引擎38在VM 32内以OS处理器特权层级(例如，内核模式)操作。在替代实施例中，评分引擎38可在VM 32内以用户模式，或甚至在VM 32外部以虚拟机监控程序30的处理器特权层级执行。

[0080] 在一些实施例中，自检引擎40基本上以与虚拟机监控程序30相同的特权层级执行，且经配置以执行虚拟机(例如VM 32(图3))的自检。VM或在相应VM上执行的软件实体的自检可包括：分析软件实体的行为、确定及/或存取此类实体的存储器地址、将特定进程的存取限制到定位于此类地址处的存储器的内容、分析此内容及确定相应实体的评估指示符(例如，图5中的指示符52d)以及其它。

[0081] 在一些实施例中，主机系统10可经配置以与远程安全服务器交换安全信息，例如关于恶意软件检测事件的细节。图11图解说明此示范性配置，其中多个主机系统10a到10c(例如上文所讨论的系统10)经由计算机网络26连接到安全服务器110。在示范性实施例中，主机系统10a到10c为由公司的员工使用的个别计算机，而安全服务器110可包括由相应公司的网络管理员配置以监视在系统10a到10c上发生的恶意软件威胁或安全事件的计算机系统。在另一实施例中，举例来说在其中每一主机系统10a到10c为托管十几个或数以百计虚拟机的服务器的基础设施即服务(IAAS)系统中，安全服务器110可包括经配置以从中央位置管理所有此类VM的反恶意软件操作的计算机系统。在又一实施例中，安全服务器110可

包括计算机系统，所述计算机系统由反恶意软件软件的提供者(例如，安全应用程序44的提供者以及其它)配置以接收关于在围绕网络26的各种系统上检测到的恶意软件的统计及/或行为数据。网络26可包含广域网(例如因特网)，而网络26的部分可包含局域网(LAN)。

[0082] 图12展示在如图11中所展示的实施例中的主机系统10与安全服务器110之间的示范性数据交换。主机系统10可经配置以将安全报告80发送到服务器110，且从服务器110接收一组安全设定82。在一些实施例中，安全报告80包括由在主机系统10上执行的实体评估器50a到50c及/或40确定的实体评估指示符52a到52d及/或分数，及/或由评分引擎38确定的聚合分数以及其它。安全报告80还可包括识别相应系统10及经评估实体的数据(例如，实体ID、名称、路径、散列或其它种类的实体识别符)，以及将实体评估指示符/分数与主机系统及确定其实体评估指示符/分数的实体相关联的指示符。在一些实施例中，报告80可进一步包括关于在主机系统10上执行的实体的统计及/或行为数据。系统10可经配置以在检测到恶意软件后即刻发送报告80，及/或根据调度(例如，每隔几分钟、每小时等)发送报告80。

[0083] 在一些实施例中，安全设定82可包含实体评估器的操作参数(例如，图5中的过滤器50a到50c的参数)及/或评分引擎38的参数。引擎38的示范性参数包含用于决定经评估进程是否为恶意的阈值，以及分数值80及权重82a到82e以及其它。

[0084] 在一些实施例中，服务器110运行优化算法以动态地调整此类参数以使恶意软件-检测性能最大化，举例来说增加检测率同时使误肯定最小化。优化算法可接收关于在所述多个主机系统10a到10c上执行的各种实体的统计及/或行为数据(其包含由各种实体评估器报告到评分引擎38的实体评估指示符/分数)，及针对参数确定最优值。接着经由网络26将所述值发射到相应主机系统。

[0085] 在一个此类优化实例中，改变分数值80可有效地改变相应评估准则相对于彼此的相关性。恶意软件威胁通常像波浪一样发生，其中世界范围内的大量计算机系统在短时间间隔内被同一恶意软件代理影响。通过从多个主机系统实时接收安全报告80，安全服务器110可与目前恶意软件威胁保持同步，且可将最优安全设定82及时地递送到相应主机系统，设定82包含(举例来说)经优化以用于检测目前恶意软件威胁的一组分数值80。

[0086] 上文所描述的示范性系统及方法允许保护主机系统(例如计算机系统)免受恶意软件(例如病毒、特洛伊木马及间谍软件)的影响。对于多个可执行实体(例如同时在主机系统上执行的进程及线程)中的每一者，评分引擎记录多个评估分数，每一分数是根据相异评估准则确定的。在一些实施例中，经评估软件实体可基本上在范围及复杂性方面变化，举例来说从个别执行线程变化到个别应用程序，变化到操作系统及/或虚拟机的整个实例。

[0087] 每当所监视实体满足评估准则(例如，执行行动)时，实体的相应分数便被更新。更新目标实体的分数可触发对与目标实体有关的其它实体的分数更新。此类相关实体尤其包含目标实体的子代、目标实体的父代、目标实体已将代码注入到其中的实体及已将代码注入到目标实体中的实体。

[0088] 常规反恶意软件系统通常将每一实体与其它实体单独地评分。一些恶意软件可设法通过以下方式来规避检测：将恶意活动分到数个相异代理当中(例如恶意进程的子进程)，使得个别代理中没有一者执行将被检测到的充分的恶意软件-指示性活动。相比之下，本发明的一些实施例将分数从一个实体传播到其它相关实体，因此跨越相关实体确证恶意软件-指示性数据。分数传播可确保检测到恶意活动中所涉及的代理中的至少一者。

[0089] 在一个示范性规避策略中，恶意软件代理可繁衍多个子进程并退出。恶意活动可被分到子进程当中，使得没有个别子进程的行动可独立地触发恶意软件警报。在本发明的一些实施例中，分数可从一个实体传播到另一实体，甚至当另一实体被终止时也如此。此配置可检测父进程为恶意的，但其可能未能检测子进程的恶意性。一些实施例维持目前正进行评估的实体的列表；所述列表可包含作用实体及已终止实体两者。仅当实体的所有后代均终止时才可从列表去掉相应实体。

[0090] 在常规反恶意软件系统中，通常为每一实体仅记录一个分数。通过保持多个每-实体分数(每一分数根据其相异准则计算)，本发明的一些实施例允许在每-准则基础上将分数在相关实体当中传播。此类分数可在传播后即刻增加或降低，从而允许贯穿每一实体的寿命周期对恶意性的较精确评价，具有较少误肯定检测。在一些实施例中，一个实体的分数对相关实体的分数的影响程度可经由数值传播权重调整。此类权重可在实体之间及/或在评估准则之间不同，从而允许分数传播的灵活及精确调谐。权重值可由人类操作者确定及/或经受旨在改进恶意软件检测性能的自动化优化。

[0091] 一些常规反恶意软件系统通过确定经评估实体是否执行恶意软件-指示性行为，及/或所述实体是否具有恶意软件-指示性特征(例如恶意软件-指示性代码序列)而确定相应实体是否为恶意。相比之下，在本发明的一些实施例中，实体评估准则独立地未必是恶意软件-指示性的。举例来说，一些准则包含确定实体是否执行良性行动，例如打开文件或接入IP地址。然而，当与其它行动组合时，此类行动可为恶意的(此类行动本身独立地可并非恶意软件-指示性的)。通过监视广泛多种实体行为及/或特征、随后记录大量(可能为数以百计)评估分数及以每-实体方式整合此类分数，本发明的一些实施例可增加检测率同时使误肯定最小化。

[0092] 本发明的一些实施例可保护虚拟化环境。在经配置以支持虚拟化的实施例中，本发明的一些组件可在虚拟机内执行，然而其它组件可在相应虚拟机外部执行(举例来说以暴露相应虚拟机的虚拟机监控程序的层级)。以虚拟机监控程序层级执行的此类组件可经配置以对同时在相应主机系统上执行的多个虚拟机执行反恶意软件操作。

[0093] 所属领域中的技术人员将清楚，可在不背离本发明的范围的情况下以许多方式更改以上实施例。因此，本发明的范围应由所附权利要求书及其法律等效内容来确定。

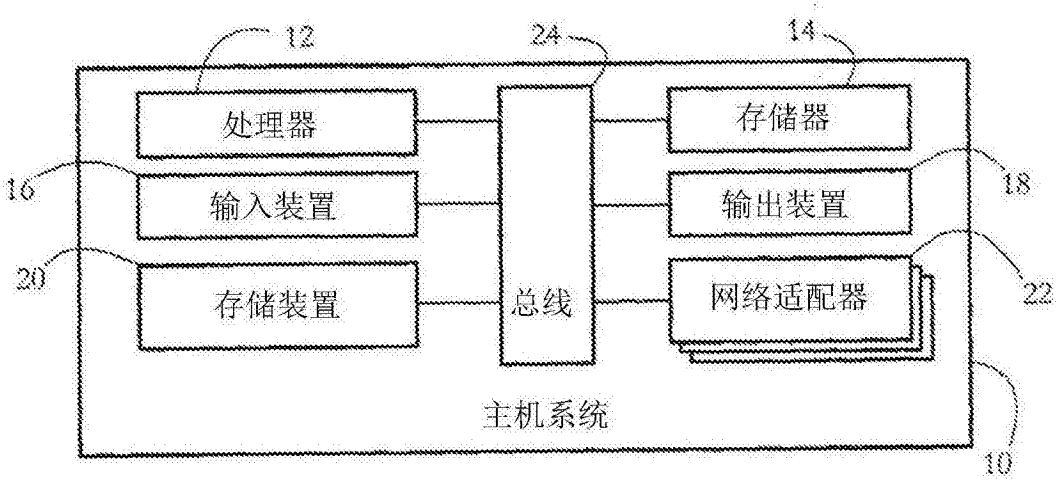

12
24
14
处理器
存储器
16
输入装置
输出装置
18
20
存储装置
总线
网络适配器
22
主机系统
10

图1

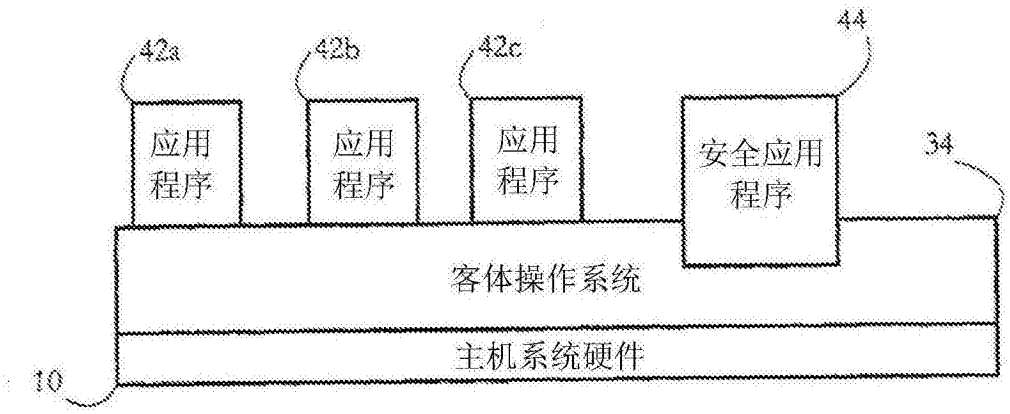

42a
42b
42c
44
应用程序
应用程序
应用程序
安全应用程序
34
客体操作系统
主机系统硬件
10

图2-A

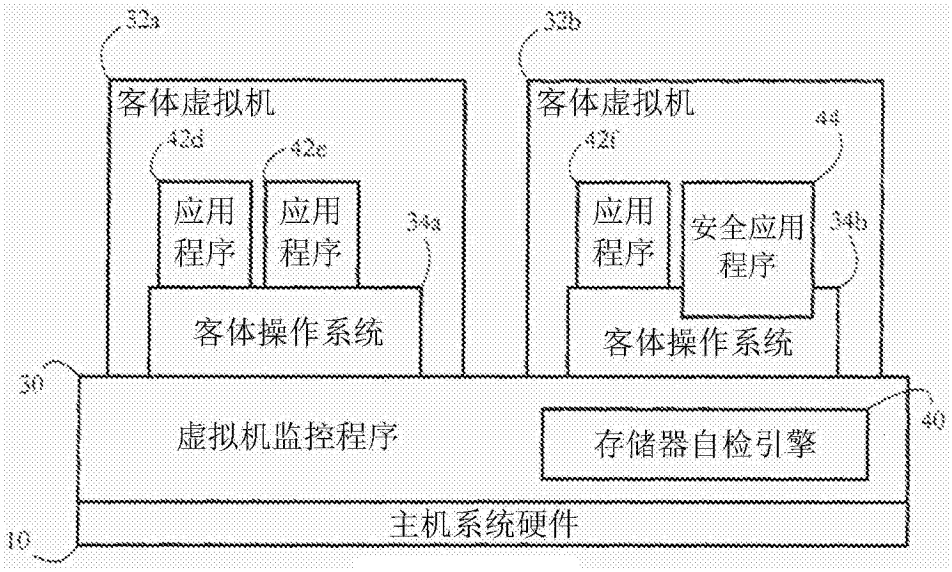

32a
32b
客体虚拟机
客体虚拟机
42d
42e
42f
44
应用程序
应用程序
应用程序
安全应用程序
34a
34b
客体操作系统
客体操作系统
30
虚拟机监控程序
存储器自检引擎
40
主机系统硬件
10

图2-B

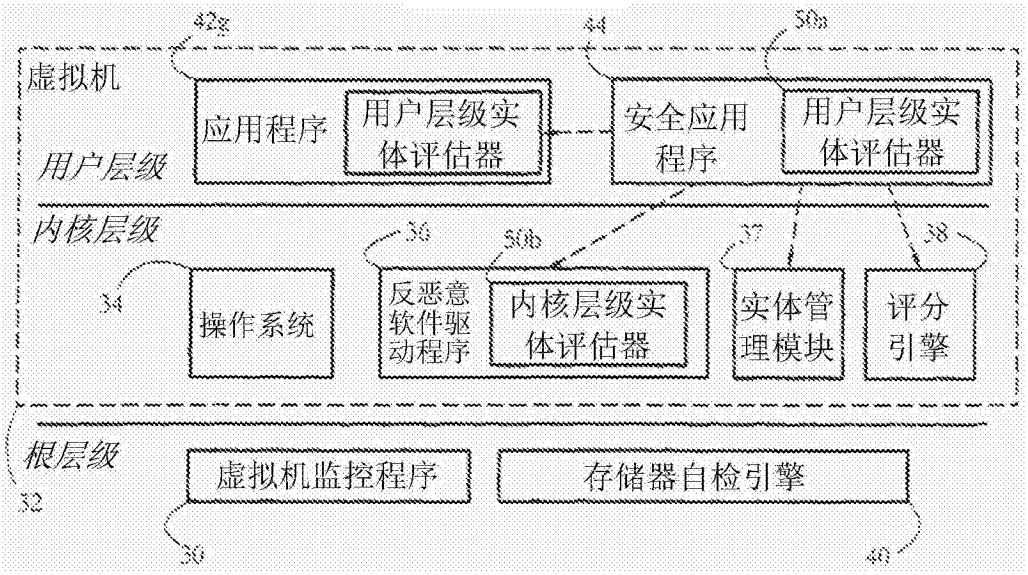

42g
44
50a
虚拟机
应用程序
用户层级实体评估器
安全应用程序
用户层级实体评估器
用户层级
内核层级
36
50b
37
38
34
操作系统
反恶意软件驱动程序
内核层级实体评估器
实体管理模块
评分引擎
根层级
32
虚拟机监控程序
存储器自检引擎
30
40

图3

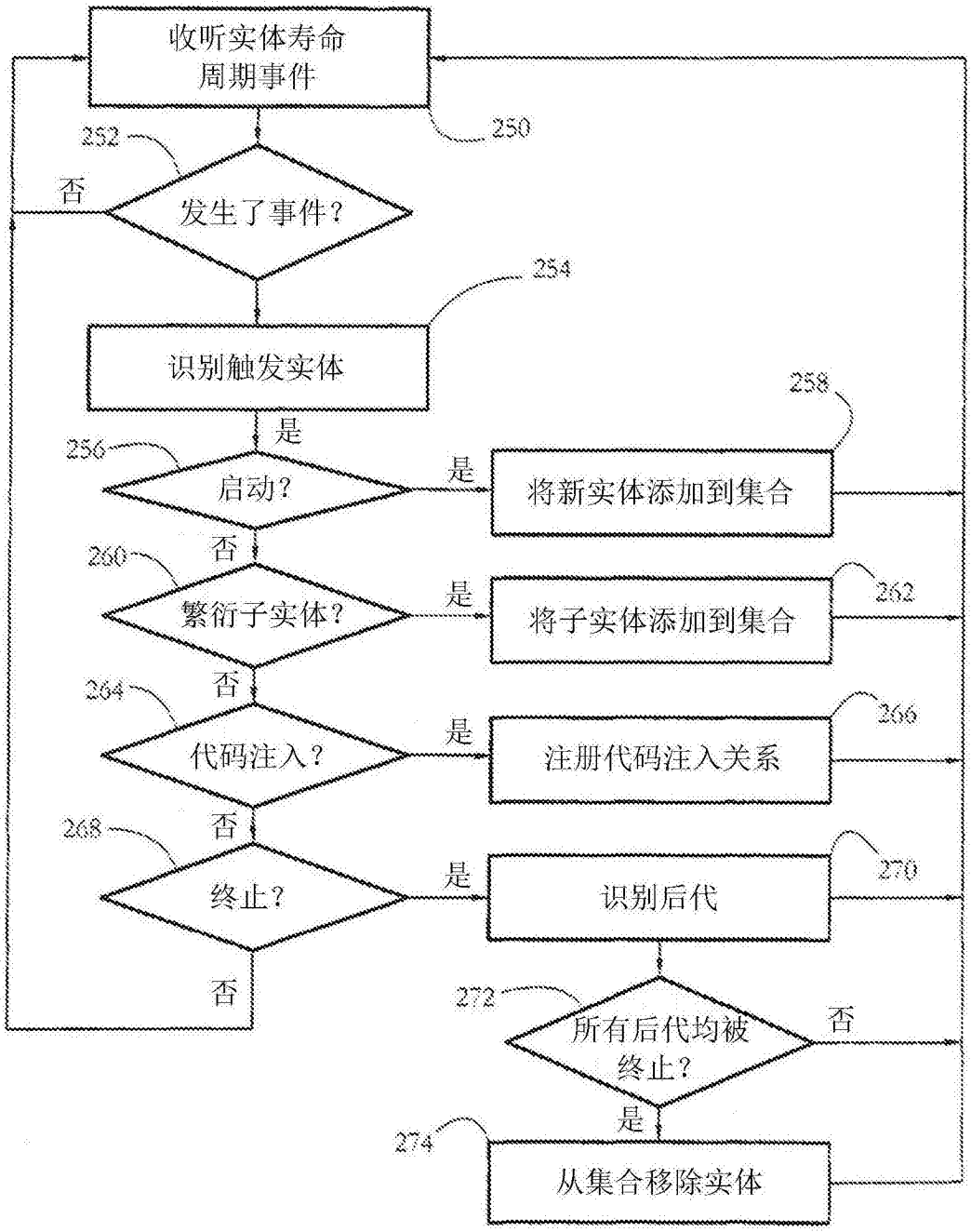

收听实体寿命周期事件
250
252
发生了事件？
否
254
识别触发实体
是
256
启动？
是
258
将新实体添加到集合
否
260
繁衍子实体？
是
262
将子实体添加到集合
否
264
代码注入？
是
266
注册代码注入关系
否
268
终止？
是
270
识别后代
否
272
所有后代均被终止？
否
是
274
从集合移除实体

图4

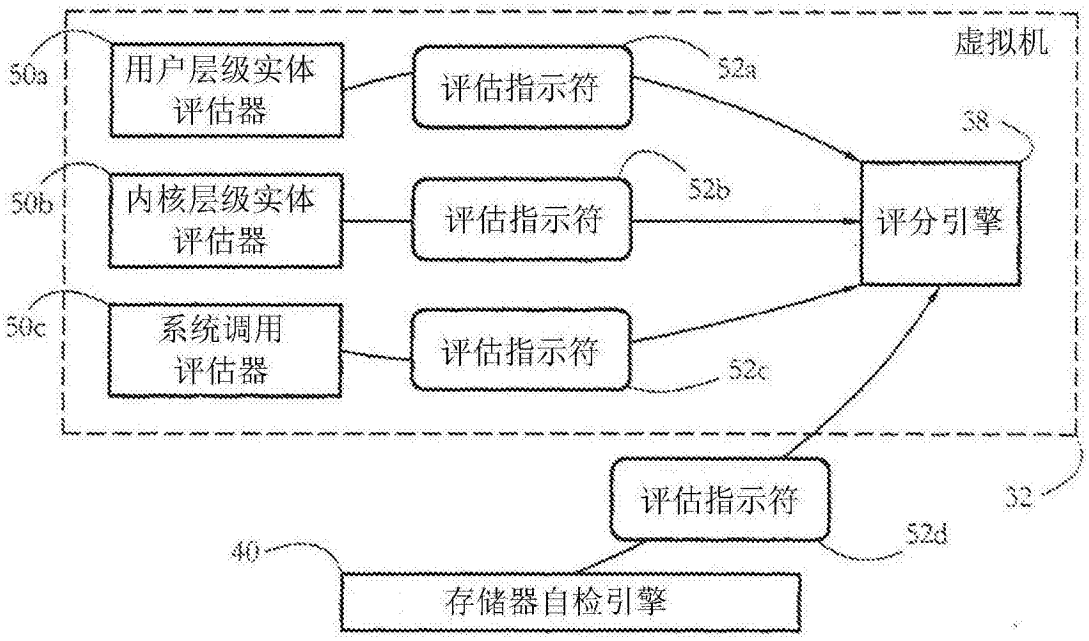

虚拟机
50a
用户层级实体评估器
评估指示符
52a
38
50b
内核层级实体评估器
评估指示符
52b
评分引擎
50c
系统调用评估器
评估指示符
52c
评估指示符
32
52d
40
存储器自检引擎

图5

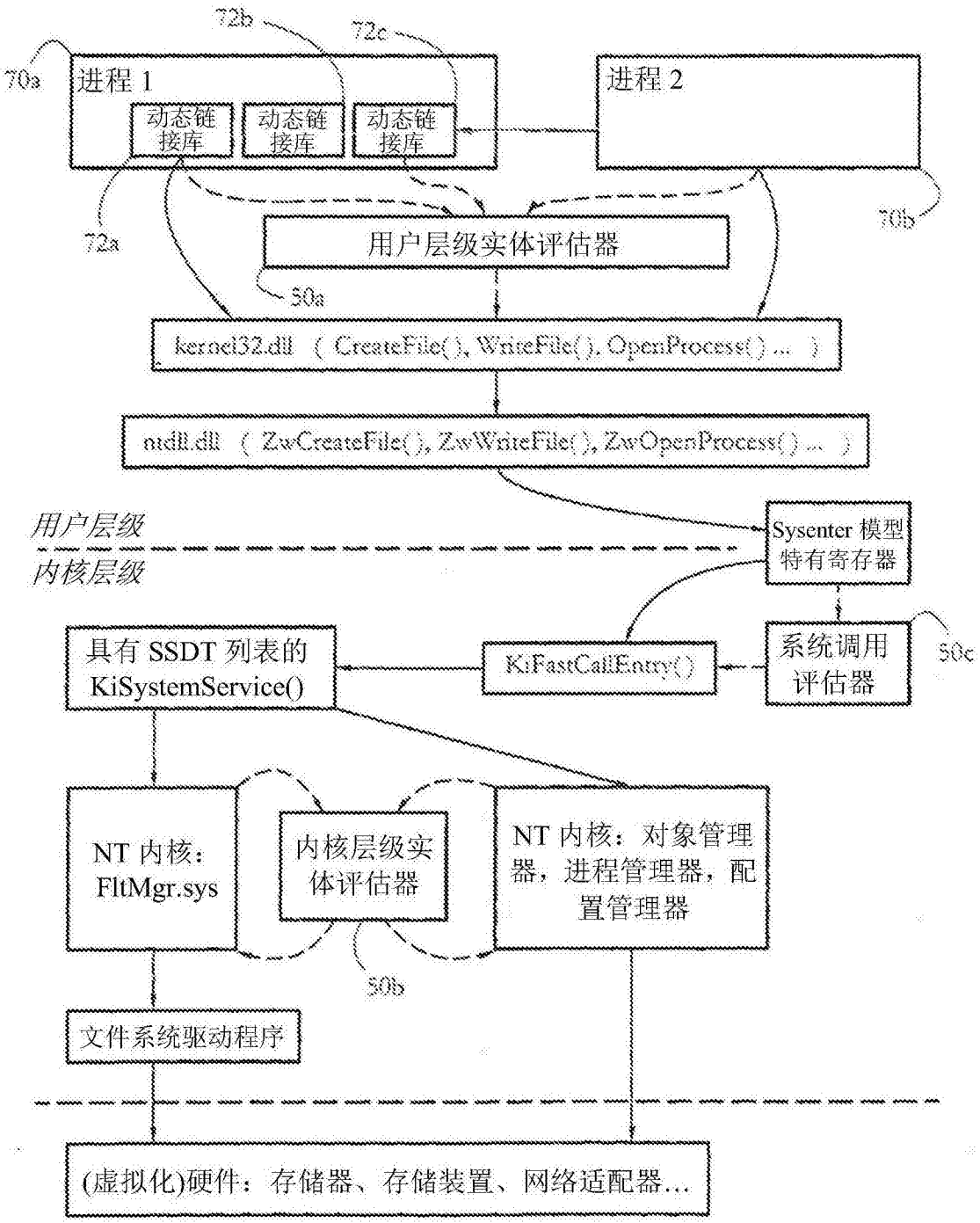

72b
72c
70a
进程 1
动态链接库
动态链接库
动态链接库
进程 2
72a
70b
用户层级实体评估器
50a
kernel32.dll （CreateFile(), WriteFile(), OpenProcess() ... ）
ntdll.dll （ZwCreateFile(), ZwWriteFile(), ZwOpenProcess() ... ）
用户层级
内核层级
Sysenter 模型特有寄存器
具有 SSDT 列表的 KiSystemService()
KiFastCallEntry()
系统调用评估器
50c
NT 内核：FltMgr.sys
内核层级实体评估器
NT 内核：对象管理器，进程管理器，配置管理器
50b
文件系统驱动程序
(虚拟化)硬件：存储器、存储装置、网络适配器...

图6

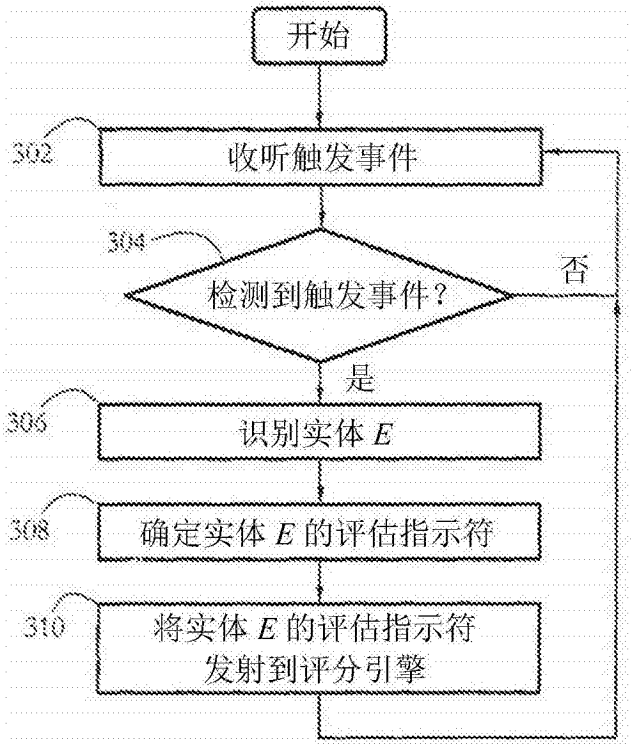

开始
302
收听触发事件
304
检测到触发事件？
否
是
306
识别实体 E
308
确定实体 E 的评估指示符
310
将实体 E 的评估指示符
发射到评分引擎

图7

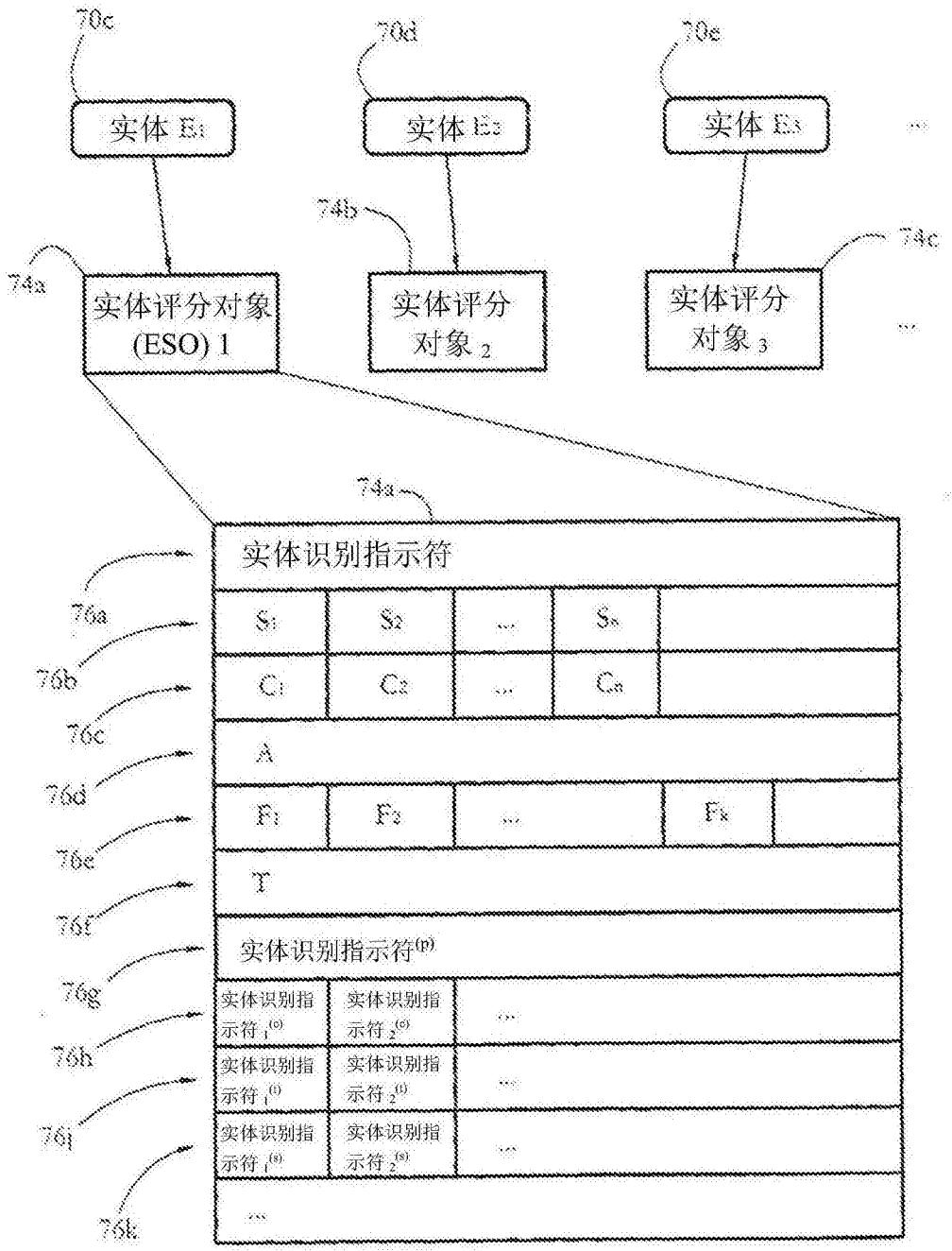

70c
70d
70e
实体 E1
实体 E2
实体 E3
74b
74a
74c
实体评分对象 (ESO) 1
实体评分对象 2
实体评分对象 3
74a
实体识别指示符
76a
S1
S2
...
Sn
76b
C1
C2
...
Cn
76c
A
76d
F1
F2
...
Fx
76e
T
76f
实体识别指示符(n)
76g
实体识别指示符1(o)
实体识别指示符2(o)
76h
实体识别指示符1(t)
实体识别指示符2(t)
76i
实体识别指示符1(s)
实体识别指示符2(s)
76k

图8

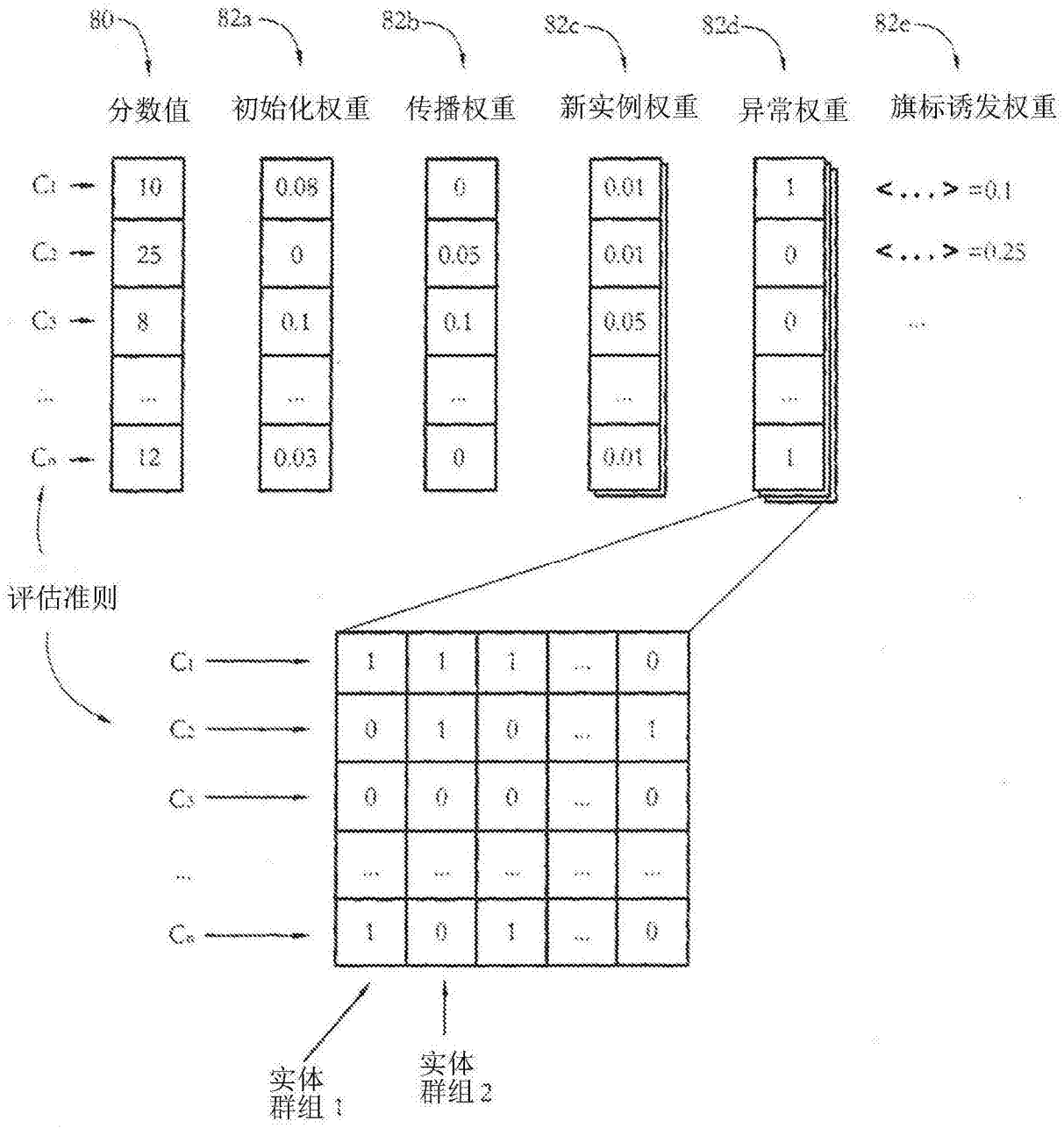

80
82a
82b
82c
82d
82e
分数值
初始化权重
传播权重
新实例权重
异常权重
旗标诱发权重
C1
C2
C3
...
Cn
10
25
8
...
12
0.08
0
0.1
...
0.03
0
0.05
0.1
...
0
0.01
0.01
0.05
...
0.01
1
0
0
...
1
<...> =0.1
<...> =0.25
...
评估准则
C1
C2
C3
...
Cn
1 1 1 ... 0
0 1 0 ... 1
0 0 0 ... 0
... ...
1 0 1 ... 0
实体群组 1
实体群组 2

图9

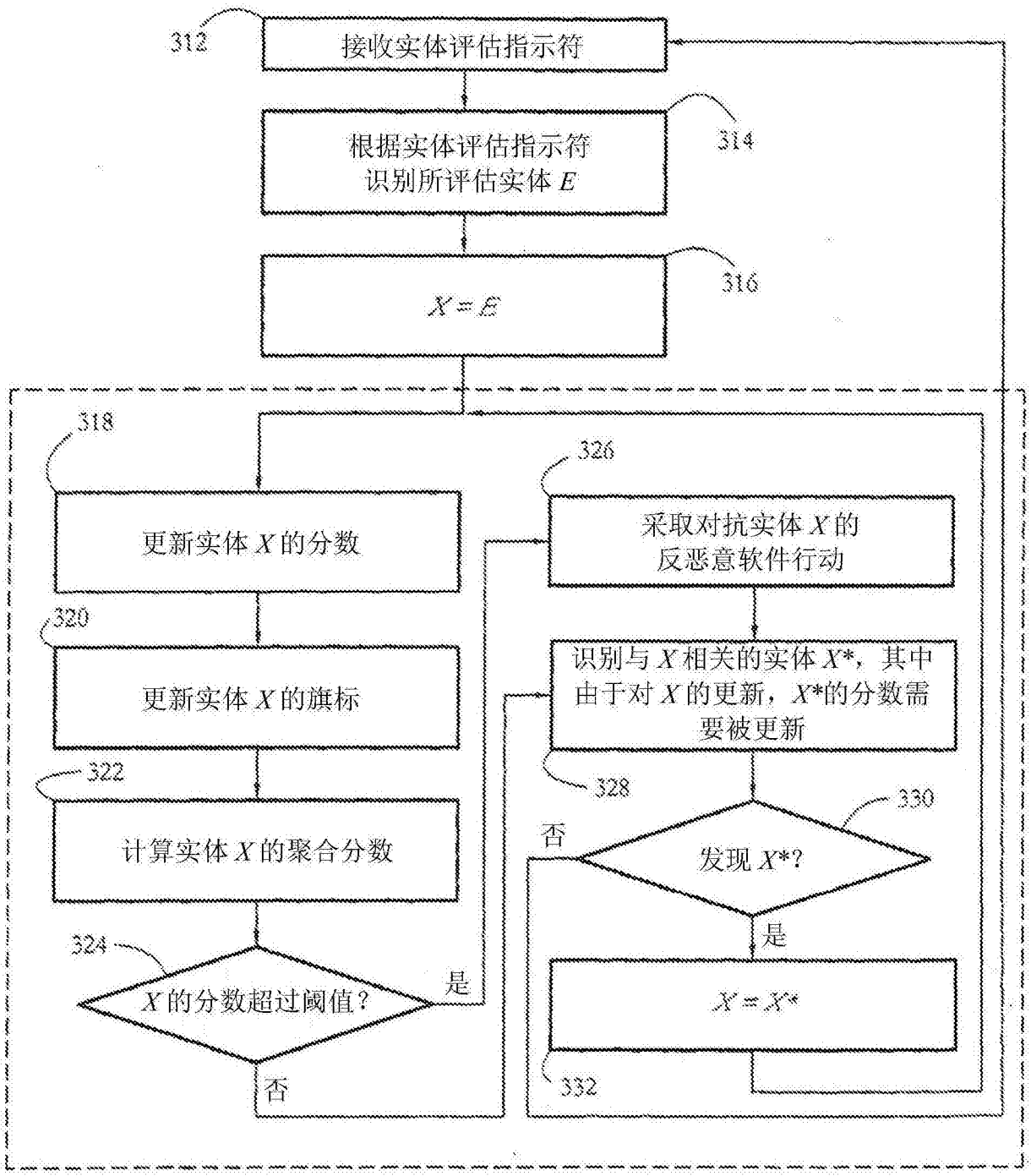

312
接收实体评估指示符
314
根据实体评估指示符
识别所评估实体 E
316
X = E
318
更新实体 X 的分数
320
更新实体 X 的旗标
322
计算实体 X 的聚合分数
324
X 的分数超过阈值？
是
否
326
采取对抗实体 X 的
反恶意软件行动
328
识别与 X 相关的实体 X*，其中
由于对 X 的更新，X*的分数需
要被更新
330
发现 X*？
否
是
X = X*
332

图10

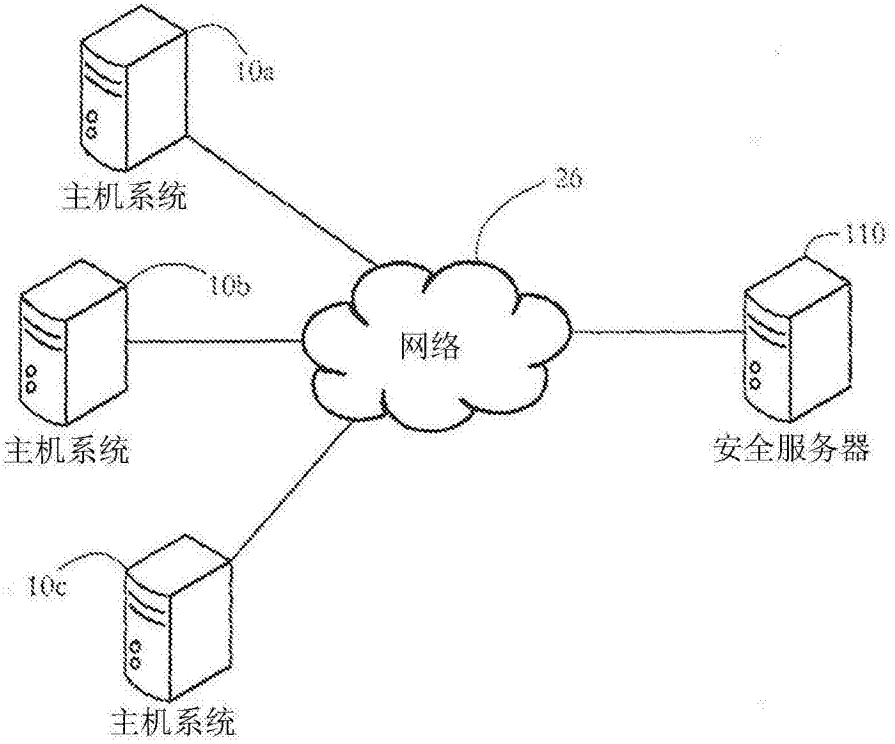

10a
主机系统
26
网络
110
安全服务器
10b
主机系统
10c
主机系统

图11

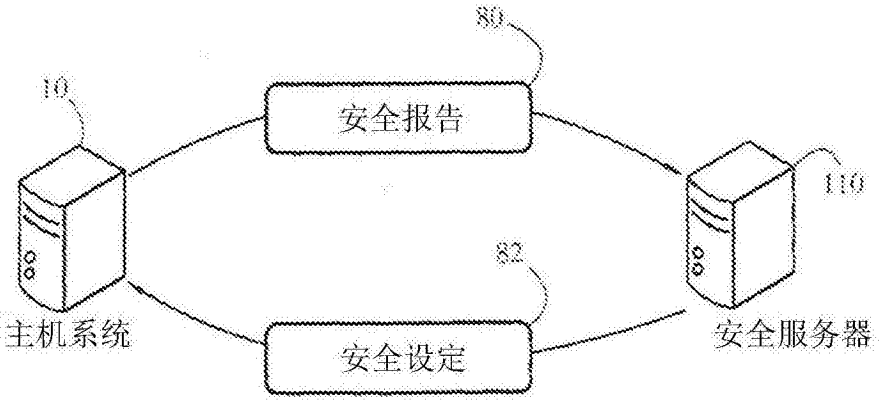

80
10
安全报告
110
82
主机系统
安全设定
安全服务器

图12