[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
*G06F 1/00 (2006.01)*

[12] 发明专利说明书

专利号 ZL 03817429.4

[45] 授权公告日 2007 年 5 月 30 日

[11] 授权公告号 CN 1318932C

[22] 申请日 2003.7.17 [21] 申请号 03817429.4

[30] 优先权

[32] 2002. 7.23 [33] US [31] 10/202,517

[86] 国际申请 PCT/GB2003/003112 2003.7.17

[87] 国际公布 WO2004/010269 英 2004.1.29

[85] 进入国家阶段日期 2005.1.21

[73] 专利权人 国际商业机器公司

地址 美国纽约

[72] 发明人 威廉·卡里斯利·阿诺尔德

戴维·迈克尔·切斯

约翰·弗雷德里克·莫拉尔

阿拉·西格尔

伊恩·尼古拉斯·沃利

斯蒂夫·理查德·怀特

[56] 参考文献

WO99/66386A1 1999.12.23

CN1314638A 2001.9.26

WO02/37740A2 2002.5.10

WO02/27440A2 2002.4.4

WO02/06928A2 2002.1.24

审查员 俞立文

[74] 专利代理机构 中国国际贸易促进委员会专利商标事务所

代理人 李德山

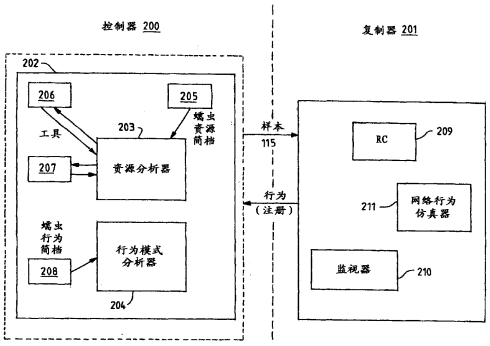
权利要求书 2 页 说明书 9 页 附图 4 页

[54] 发明名称

# 用于自动确定程序的潜在蠕虫样行为的方法与装置

[57] 摘要

用于自动确定怀疑具有蠕虫样特征的程序的行为简档的方法和系统，包括分析该程序所需的数据处理系统资源，如果所需资源不表示该程序具有蠕虫样特征，则在受控的非网络环境中运行该程序，同时监视并注册对系统资源的访问，从而确定该程序在非网络环境中的行为。 分析所观察行为的日志记录，确定该行为是否表示程序具有蠕虫样特征。该非网络环境可以向程序仿真网络的表象，但不仿真网络的操作。



1、一种用于自动确定程序的潜在蠕虫样行为的方法，包括：

在不仿真网络操作的环境中确定程序的行为简档；

比较所确定的行为简档与表示蠕虫样行为的简档；及

基于比较的结果提供潜在蠕虫样行为的指示；

其中确定行为简档的步骤包括：

在至少一种已知的非网络环境中执行该程序；

利用自动化方法检查环境，并且如果有变化的话就确定在环境中出现了什么变化；及

记录任何所确定的变化作为所述行为简档。

2、如权利要求 1 所述的方法，其中已知的非网络环境具有看起来能显示与网络相关的功能的能力。

3、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，对该程序企图确定环境是否具有网络功能作出响应，向该程序提供网络地址，作为该程序显示蠕虫样行为的诱饵。

4、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，已知的非网络环境向该程序显示不存在的本地与网络相关的资源和本地与网络相关的对象中的至少一种。

5、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，对该程序企图确定关于文件的信息作出响应，作出该文件好像存在的响应，作为该程序显示蠕虫样行为的诱饵。

6、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，对该程序企图确定关于文件的信息作出响应，在该文件返回该程序之前创建该文件，作为该程序显示蠕虫样行为的诱饵。

7、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，对该程序企图确定关于电子邮件程序的信息作出响应，向该程序返回该信息，作为该程序显示蠕虫样行为的诱饵。

8、如权利要求 1 所述的方法，其中，在至少一种已知的非网络环境中执行该程序期间，对该程序企图确定关于电子邮件地址本的信息作出响应，向该程序返回该信息，作为该程序显示蠕虫样行为的诱饵。

9、数据处理系统，包括至少一台计算机，该至少一台计算机用于执行所存储的程序，来自动确定一种程序的潜在蠕虫样行为，该系统包括:

用于在不仿真网络操作的环境中确定程序行为简档的装置;

用于比较所确定的行为简档与表示蠕虫样行为的存储简档的装置; 及

用于基于比较结果提供潜在蠕虫样行为的指示的装置;

其中确定行为简档的装置包括:

在至少一种已知的非网络环境中执行该程序的装置;

利用自动化方法检查环境，并且如果有变化的话就确定在环境中出现了什么变化的装置; 及

记录任何所确定的变化作为所述行为简档的装置。

# 用于自动确定程序的潜在蠕虫样行为的方法与装置

## 技术领域

本发明总体上涉及用于分析如那些称为蠕虫的不希望的软件实体的方法和装置，更具体而言，涉及用于自动识别软件程序中潜在的蠕虫样行为的方法与装置。

## 背景技术

计算机病毒可以定义为在计算机上以可能修改的方式而无需人介入地传播的自复制程序或软件例程。计算机蠕虫可以定义为暗中在计算机网络的计算机之间发送其自己的拷贝的程序，而且它利用网络服务进行复制。

在计算机病毒自动检测与分析领域，经常需要预测程序将显示什么类型的行为，由此程序可以在最适合程序的环境中被复制与分析。

软件可以被动态分析，以识别潜在的重要行为（如蠕虫样行为）。这种行为有可能只有当软件在软件能或看起来能访问生产网络和/或全球因特网的环境中执行时才显示。软件可以在实际的或包括监视组件和仿真组件的仿真网络环境中执行。监视组件用于捕捉和/或记录由软件和/或系统其它组件显示的行为，而仿真组件给被分析的软件一种它在访问生产网络和/或全球因特网的情况下执行的印象。被分析的软件被有效地限制于分析网络环境，实际上不能从任何生产网络或全球因特网读信息或改变在任何生产网络或全球因特网上的任何信息。

期望提供一种在这种环境以外指出计算机蠕虫身份的能力。尽管有可能利用这种环境来复制计算机软件病毒与蠕虫，但是由于蠕虫复制环境假定实际或仿真网络的存在，而实际中这实现起来很昂贵，因此是低效的。

因此，在网络环境以外预测软件样本是否是潜在蠕虫的能力将减少发送到蠕虫复制环境的样本数，并导致自动复制与分析系统效率的显著提高。

## 发明内容

因此，本发明提供用于自动确定怀疑具有蠕虫样特征的程序的行为简档的方法与装置。在第一方面，方法包括在不仿真网络操作的环境中确定程序的行为简档；比较所确定的行为简档与表示蠕虫样行为的简档；及基于比较的结果提供潜在蠕虫样行为的指示；确定行为简档的步骤包括：在至少一种已知的非网络环境中执行该程序；利用自动化方法检查环境，并且如果有变化的话就确定在环境中出现了什么变化；及记录任何所确定的变化作为所述行为简档。这样，分析所观察行为的注册记录，来确定其行为是否表示程序具有蠕虫样特征。非网络环境可以向程序仿真网络的出现，但不仿真网络的实际操作。

## 附图说明

现在参考附图，仅作为例子来描述本发明的优选实施方式，其中：

图 1 是本发明优选实施方式中数据处理系统的方框图；

图 2 是本发明优选实施方式中控制器单元的方框图；

图 3 是本发明优选实施方式中说明图 2 资源分析器组件运行的逻辑流程图；

图 4A 和 4B，统称为图 4，示出了本发明优选实施方式中说明图 2 复制器单元运行的逻辑流程图和说明图 2 行为模式分析器组件运行的逻辑流程图。

## 具体实施方式

在此所公开的方法基于蠕虫程序能将其向其它计算机传播的特征。这些特征包括但不限于以下一个或多个特征：（a）利用来自动态链接库（dll）的应用程序接口（API）向其它机器发电子邮件；（b）

通用邮件程序的自动化，该邮件程序例如但不限于 Microsoft Outlook™ 和 Outlook Express™；（c）使用地址本、系统注册表和其它系统资源确定蠕虫的潜在接收者和/或邮件程序的位置；（d）用蠕虫代码重写或代替系统库中的网络 API，该蠕虫代码如果执行，则将使得在所有从该机器发送的邮件中包含蠕虫代码作为附件或作为单独的邮件消息；（e）企图访问远程驱动器上的资源；及（f）删除将在系统重新启动后运行、并利用前面所述的一种方法使蠕虫代码发送到其它机器的程序。

蠕虫的一种简档是显示那些表示使用了上面提到的一种或多种方法的特征的简档。例如，蠕虫简档可以包括从网络 dll 导入发送方法或改变网络资源，或企图访问系统注册表来获得描述任何所安装的电子邮件程序的信息。

可疑程序（怀疑具有蠕虫样行为、特征或属性的程序）的行为简档的确定是分两个阶段执行的。

在第一阶段，确定程序所需的资源。表示潜在蠕虫样行为的资源的例子包括但不限于：（a）用于网络访问的动态链接库；（b）从这些库导入的表示可能企图发送电子邮件的方法；及（c）表示现有邮件程序自动化的动态链接库和方法，如 OLE 或 DDE。

在第二阶段，可疑程序在受控非网络环境中运行一次或多次，其中对系统资源的访问，有可能是全部访问，被监视并注册。为了缩小结果并减少误检的个数，该非网络环境可以配置成看起来象具有一些网络能力。例如，如果可疑蠕虫企图访问地址本或检查电子邮件程序的存在，那么为了得到更具体的蠕虫样响应，该环境将提供期望的信息。在有些情况下，为了方便对企图访问电子邮件程序的可疑程序的积极响应，在系统上安装电子邮件程序可能是有利的。

在完成一次或多次程序运行后，基于对系统所作的修改，例如但仅仅作为例子，对任何网络 dll 的修改及访问如包含邮件程序或地址本位置的注册表的特定资源的企图，创建该程序的行为简档。

本发明的一种优选实施方式运行在图 1 所示的一个或多个计算机

系统 100 上，图 1 示出了可以实现本发明该优选实施方式的典型计算机系统 100 的方框图。计算机系统 100 包括具有硬件单元 103 的计算机平台 102、在此也称为控制器 101 的实现下面所公开方法的软件分析程序（SAP）101。SAP 101 运行在计算机平台 102 和硬件单元 103 之上。硬件单元 103 一般包括一个或多个中央处理单元（CPU）104、可以包括随机访问存储器（RAM）的存储器 105 及输入/输出（I/O）接口 106。微指令码，如精简指令集，也可以包括在平台 102 中。各种外围组件 130 可以连接到计算机平台 102。一般提供的外围组件 130 包括显示器 109、存储该优选实施方式所使用数据及驻留蠕虫资源简档 205（见图 2）的外部数据存储设备（如磁带或磁盘）110，还有打印机 111。还可以包括链路 112，将系统 100 连接到一个或多个其它简单示为块 113 的类似计算机系统。链路 112 用于在计算机 100 和 113 之间发送数字信息。链路 112 还可以提供对全球因特网 113a 的访问。操作系统（OS）114 协调计算机系统 100 各种组件的运行，还负责管理各种对象和文件、用于记录关于该对象和文件的特定信息，如最后修改日期和时间、文件长度等。以传统方式与 OS 114 关联的是注册表 114A 及系统初始化文件（SYS_INIT_FILES）和其它文件，如 dll 114B，其中注册表 114A 的使用在下面描述。位于 OS 114 之上的是包含例如编译器、解释器和其它软件工具的软件工具层 114A。层 116 中的解释器、编译器和其它工具运行在操作系统 114 之上，并使得可以利用本领域已知的方法执行程序。

计算机系统 100 一种合适且不限制的例子是 IBM IntelliStation™（国际商用机器公司的商标）。合适的 CPU 例子是 Pentium™ III 处理器（Intel 公司的商标）；操作系统的例子是微软 Windows 2000（微软公司的商标）和 GNU/Linux 的 Redhat build; 解释器和编译器的例子是 Perl 解释器和 C++编译器。本领域技术人员应当认识到对于上面提到这些，可以替换成其它计算机系统、操作系统和工具的例子。

SAP 或控制器 101 根据本发明的教义运行，确定从执行可疑的恶

意程序或不期望软件实体，在此通称为样本 115，得到的可能蠕虫样行为。

SAP或控制器101一种目前优选但不限制的实施方式利用一个或多个计算机用于实现在图2中示出的控制器子系统或单元200和复制器子系统或单元201。构成控制器200一部分的蠕虫样行为分析器202利用一些目前存在的工具 114A，以确定可疑程序或样本 115 使用的dll和导入。

图2是本发明优选实施方式中控制器单元200和复制器单元201的具体方框图，并示出了蠕虫样行为分析器202和该实施方式运行的环境。该环境包括几个计算系统，一个是控制器 200，另一个是复制器201。

应当指出，尽管在此单元201称为复制器，但其主要功能不是复制蠕虫，即提供蠕虫另外的实例。相反，其主要目的是创建一种系统环境，更精确地说是仿真系统环境，其中以一种或多种途径演习样本115 一次或多次，从而确定样本 115 适合且使系统环境状态产生变化的行为，并获得样本115在（仿真）系统环境中运行时活动的记录或注册。系统状态的任何变化及所执行样本115的活动注册都同已知的与蠕虫样行为关联的系统状态变化和活动进行比较，如果出现匹配，则认为样本115显示蠕虫样行为。在这一点，可能期望尝试复制该可疑蠕虫以活动其更多实例进行分析及后续识别。

对于本发明，以多种“途径”演习样本115意味着通过不同的系统API（例如，系统和/或创建处理）运行该样本出现一次或几次，如果程序有GUI，还演习GUI。以多种途径演习样本115还可以通过运行该样本程序，然后重新启动系统并再次运行该程序来实现。这些技术并不是可以演习样本115的全部“途径”。

蠕虫样行为分析器202包括在此也称为静态分析器或静态确定单元的资源分析器 203，及在此也称为动态分析器或动态确定单元的行为模式分析器204。行为模式分析器204使用分别确定样本115所需的动态链接库114B列表及从动态链接库114B导入的方法的工具206

和 207。能用于确定哪个动态链接库 114B 是程序所需的工具 206、207 的例子被认为是微软 DEPENDS.EXE，在微软系统期刊发行的于 1997 年 2 月由 Matt Pietrek 所写的文章“Under the Hood”中描述，可以通过微软开发者网络活动。能用于确定样本程序 215 导入的工具 206、207 的例子被认为是 DUMPBIN.EXE，它构成微软开发者 Studio™ 版本 6.0 的一部分。其它用于执行相同或类似功能的工具可以由本领域技术人员找到或写出。

资源分析器 203（静态确定）利用这些工具创建由可疑程序或样本 115 所使用的资源简档，并将结果与蠕虫资源简档 205 的内容进行比较。典型的蠕虫资源简档 205 可以包括网络 dll 114B，例如但不限于 WSOCK32.DLL、INETMIB1.DLL、WINSOCK.DLL、MSWSOCK.DLL、WININET.DLL、MAPI32.DLL、MAPI.DLL 和 WS2_32.DLL，及表示使用 OLE 自动化的 DLL，如 OLE32.DLL，还有从这些动态链接库导入的方法列表。这些导入的方法包括但不限于从 WSOCK32.DLL 导入的“send”、“sendto”和 WSAsend 方法，从 OLE32.DLL 导入的 CoCreateInstance 和 CoCreateInstanceEx 方法，或者从 USER32.DLL 导入的 DDEConnect 方法。

行为模式分析器 204（动态确定）利用样本 115 在复制器 201 上的运行结果创建可疑程序的行为简档，并将这些结果与蠕虫行为简档 208 进行比较。典型的蠕虫行为简档 208 包括表示蠕虫样行为的一系列对系统的改变和/或对文件和注册表访问的企图。蠕虫行为简档 208 列表可以包括但不限于以下元素：（a）对一个或多个网络 dll 的改变，但不改变非网络 dll；（b）一个或多个带 VBS 后缀的文件的创建；（c）任何新文件的创建及将导致任何网络 dll 被新文件代替的对系统初始化文件 114B 的相应改变/创建。这后一种情形的例子是 Windows 系统中 Windows 目录下文件 wininit.ini 的创建，其中所创建的 wininit.int 文件包含如“WSOCK32.DLL=SOME.FILE”的指令，而 SOME.FILE 是由程序创建的新文件。蠕虫行为简档 208 列表还可以包括（d）企图访问地址本或/和对应于电子邮件程序位置的注册表键值的记录。

蠕虫资源简档 205 和蠕虫行为简档 208 优选地本质上都是静态的，优选地在样本 115 在复制器 201 上运行之前创建。

复制器 201 在上面所讨论的行为模式分析操作之前被控制器 200 调用。复制器 201 包括复制控制器（RC）209、行为监视器 210 及可选的网络行为仿真器 211。

网络行为仿真器 211 与行为监视器 210 一起运行，创建网络样行为的表象，从而得到样本 115 确定的蠕虫样行为。例如，当行为监视器 210 检测到样本 115 对 IP 地址的请求时，网络行为仿真器 211 就运行，向样本 215 提供错误的网络地址，如错误的 IP 地址。在这种情况下，样本 115 可能在显示蠕虫样行为之前请求本地 IP 地址，以确定该系统是否具有网络能力，因此可以向样本 115 提供本地 IP 地址，作为使样本 115 显示蠕虫样行为的诱饵。

以类似的方式，其中样本 115 运行的环境可以建成显示实际上不存在的系统资源和/或对象的存在。例如，样本 115 可能请求关于特定文件的信息，并且然后它可以有利于环境响应请求信息，就好像该文件存在一样，或者在它返回到样本 115 之前创建该文件作为样本显示蠕虫样行为的诱饵。即，已知的非网络环境可以建成向程序显示至少一种不存在的本地网络相关资源和本地网络相关对象。

图 3 和图 4 说明了在控制器 200 和复制器 201 执行过程中的总体控制流程。在图 3 中，样本 115 首先在步骤 301 传送到控制器 200，然后控制器 200 在步骤 302 将样本 115 传送到资源分析器 203。资源分析器 203 在步骤 303 确定哪些动态链接库 114B 是样本 115 所访问的，并将被访问的 dll 与蠕虫资源简档 205 中的那些进行比较（步骤 304）。

如果 dll 的使用与蠕虫资源简档 205 匹配，则在步骤 305 确定从这些 dll 导入的方法。如果这些方法与包含在蠕虫资源简档 205 中的那些匹配（步骤 306），则样本 115 被归为潜在的蠕虫。如果 dll 的使用或导入的方法都不与蠕虫资源简档 205 匹配，如在步骤 304 和 306 任一个中由 No 指示所指示的，则样本被传递到图 4A 所示的复制器

201，用于复制和后续行为模式确定。

图 4 说明了通过复制器 201（图 4A）和行为模式分析器（图 4B）的控制流程。在复制环境在步骤 401 初始化以后，样本在步骤 402 发送到复制器 201 并在步骤 403 执行。然后控制传递到行为模式分析器 204（图 4B），在步骤 404 检查对系统的任何改变，在步骤 405 比较检测出的结果和蠕虫行为简档 208 中的那些。如果存在匹配，则样本 115 被称为潜在的蠕虫，否则，行为模式分析器 204 在步骤 406 分析由行为监视器 210 报告的活动，并在步骤 407 试图匹配报告的样本行为与蠕虫行为简档 208 中列出的活动模式（蠕虫样行为模式）。

如果在步骤 404 分析的对系统的改变或在步骤 406 由行为监视器报告的活动包含任何在蠕虫行为简档 208 中列出的模式，则样本被归为潜在的蠕虫，否则，样本 115 被归为不是蠕虫。如果在步骤 407 归为蠕虫，则样本 115A 可以提供给蠕虫复制与分析系统，用于进一步特征化。

上述方法可以在计算机可读介质，如磁盘 110，上体现，用于实现自动确定怀疑具有蠕虫样特征的样本程序 115 的行为简档。计算机程序的执行使得计算机 100、200 可以分析样本程序 115 所需的计算机系统资源，如果所需资源不表示样本程序 115 具有蠕虫样特征，则计算机程序的进一步执行使得计算机 100、200 和 201 在受控的非网络环境中运行程序，同样监视并注册对系统资源的访问，以确定程序在非网络环境中的行为。计算机程序的进一步执行可以使计算机向样本程序仿真网络的出现，而不仿真网络的操作。

应当指出，前面的描述意味着只有如果样本的系统资源需求不表示潜在的蠕虫（静态确定），复制器 201 才运行（动态确定）。由于静态确定处理一般比动态确定处理在计算方面便宜得多，而且一般执行得更快，因此目前的优选实施方式是这种情况。但是，当潜在的蠕虫由第一处理或子系统表示时，两个处理或子系统都执行也是可能的，第二处理或子系统用于验证第一处理或子系统的结果。还应当指出，在有些情况下，由于可能会产生错误的负面结果，因此不期望需要静

态和动态确定处理都得出关于特定样本 115 蠕虫样本质的相同结论。因此，现在依赖静态或动态处理中的一种宣布特定样本是蠕虫样的是优选的，使得如果没有指明蠕虫样行为，则系统切换到作为潜在病毒来处理该样本。

如上面所指出的，对于由资源分析器 203 执行的静态确定（在仿真环境之外）通常比由行为模式分析器 204 执行的动态确定能更快地处理样本 115。

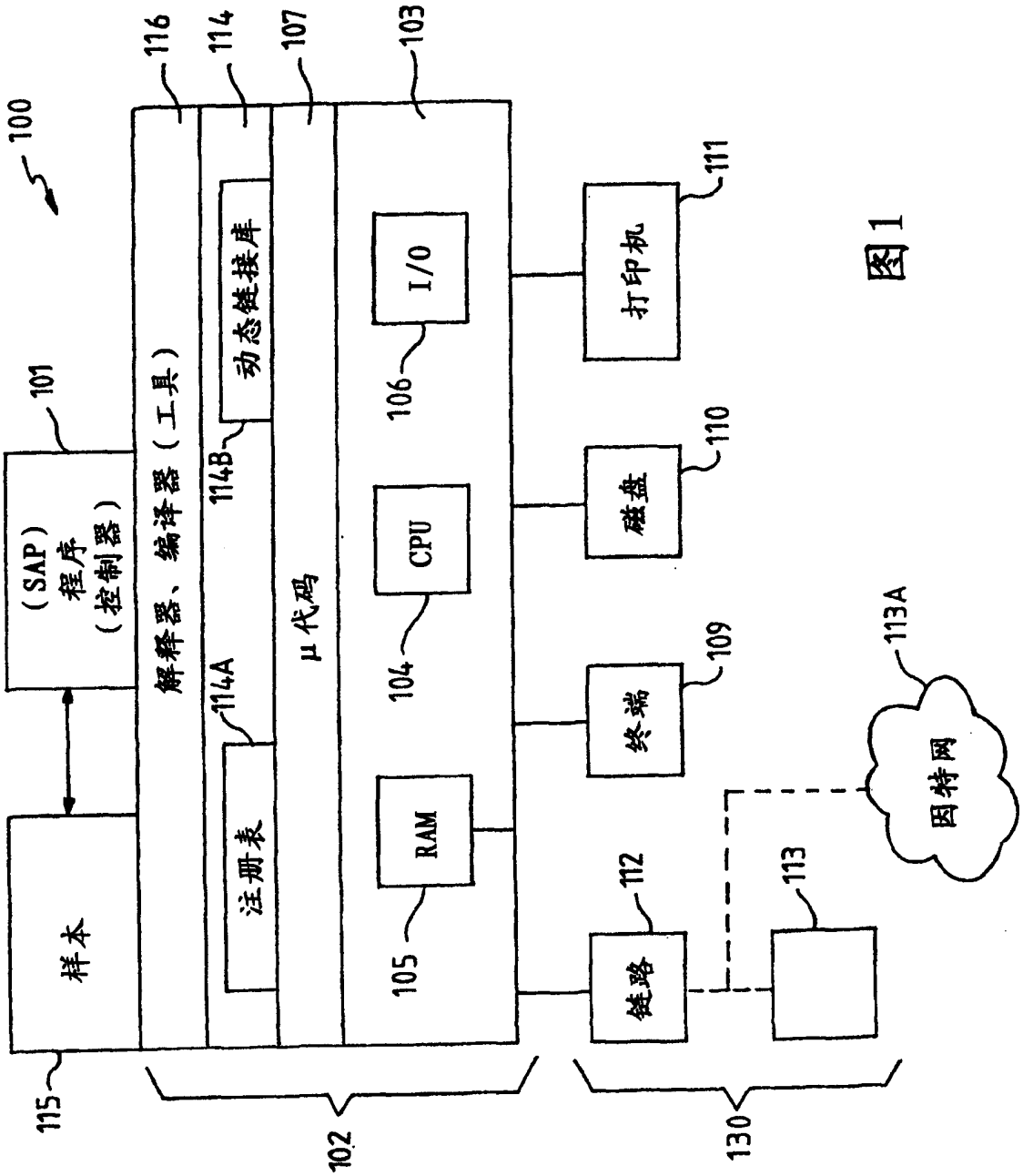

100
116
114
107
103
101
(SAP)
程序
(控制器)
解释器、编译器(工具)
动态链接库
114B
114A
注册表
μ代码
I/O
106
CPU
104
RAM
105
打印机
111
磁盘
110
终端
109
113A
因特网
链路
112
113
样本
115
102
130

图1

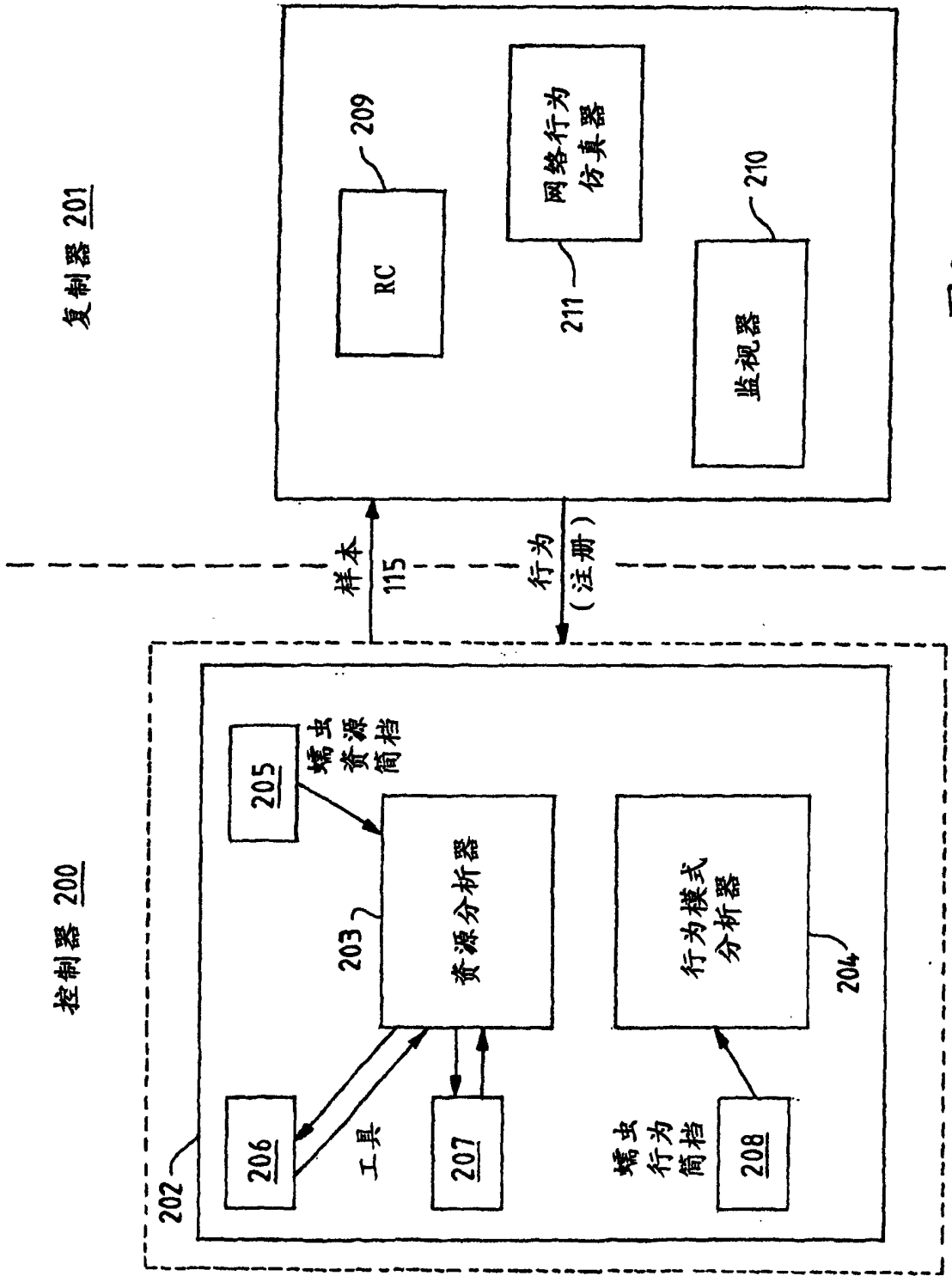

复制器 201
209
RC
网络行为仿真器
211
210
监视器
样本
115
行为（注册）
控制器 200
205
蠕虫资源简档
203
资源分析器
行为模式分析器
204
206
工具
207
蠕虫行为简档
208
202

图 2

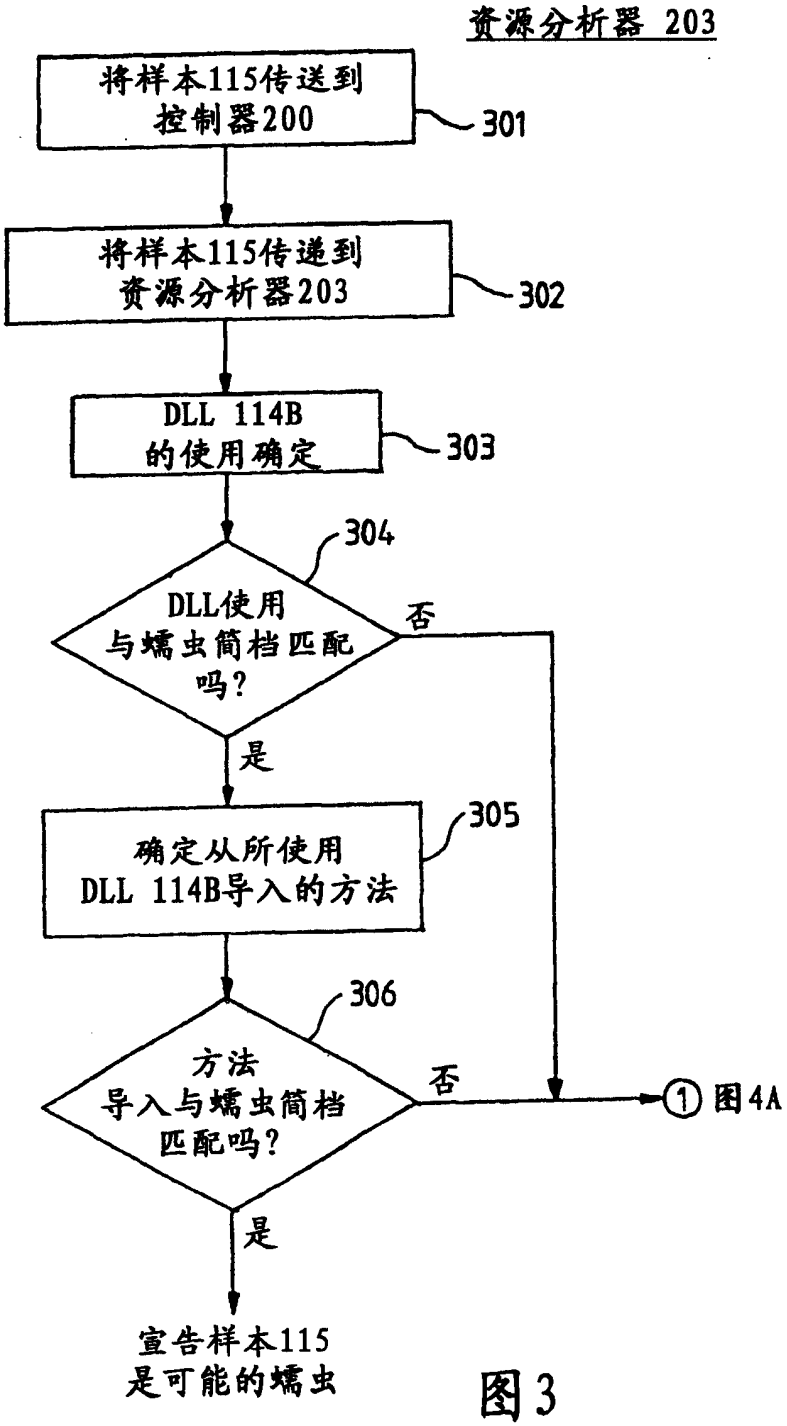

资源分析器 203
将样本115传送到控制器200
301
将样本115传递到资源分析器203
302
DLL 114B的使用确定
303
304
DLL使用与蠕虫简档匹配吗?
否
是
确定从所使用DLL 114B导入的方法
305
306
方法导入与蠕虫简档匹配吗?
否
① 图4A
是
宣告样本115是可能的蠕虫

图3

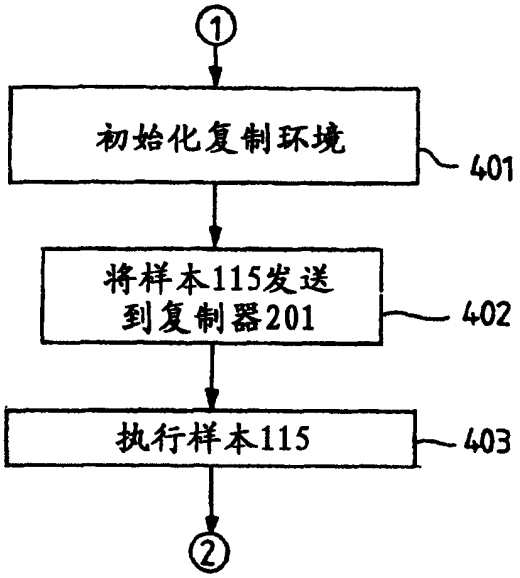

①
复制器 201
初始化复制环境
401
将样本115发送
到复制器201
402
执行样本115
403
②

图4A

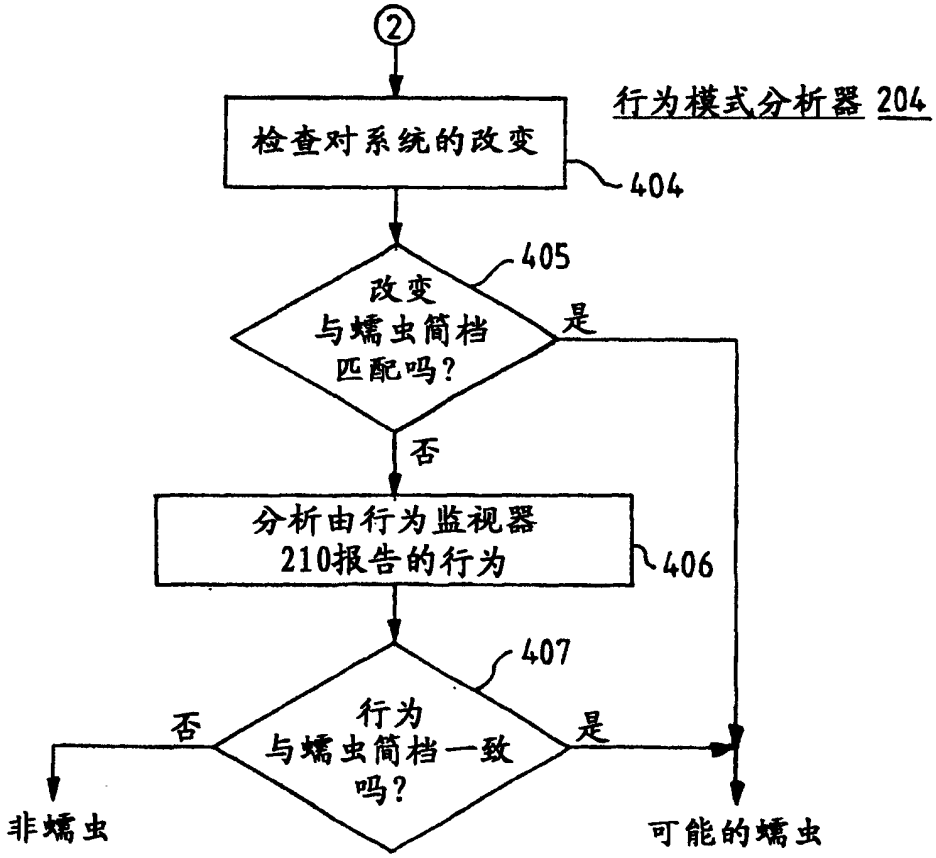

②
行为模式分析器 204
检查对系统的改变
404
405
改变
与蠕虫简档
匹配吗?
是
否
分析由行为监视器
210报告的行为
406
407
行为
与蠕虫简档一致
吗?
否
是
非蠕虫
可能的蠕虫

图4B