(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103620613 B
(45)授权公告日 2018.06.12

(21)申请号 201280016726.3

(22)申请日 2012.03.27

(65)同一申请的已公布的文献号
申请公布号 CN 103620613 A

(43)申请公布日 2014.03.05

(30)优先权数据
13/073,810 2011.03.28 US
13/073,791 2011.03.28 US
13/073,842 2011.03.28 US
13/073,853 2011.03.28 US
13/073,864 2011.03.28 US
13/074,741 2011.03.29 US
13/074,925 2011.03.29 US
13/074,947 2011.03.29 US
13/075,049 2011.03.29 US
13/075,072 2011.03.29 US
13/074,831 2011.03.29 US
13/075,101 2011.03.29 US
13/077,270 2011.03.31 US
13/076,537 2011.03.31 US
13/077,305 2011.03.31 US
13/077,227 2011.03.31 US
13/076,512 2011.03.31 US
13/076,480 2011.03.31 US
13/076,473 2011.03.31 US
13/076,493 2011.03.31 US

(85)PCT国际申请进入国家阶段日
2013.09.30

(86)PCT国际申请的申请数据
PCT/US2012/030702 2012.03.27

(87)PCT国际申请的公布数据
WO2012/135192 EN 2012.10.04

(73)专利权人 迈克菲股份有限公司
地址 美国加利福尼亚州

(72)发明人 A·S·萨兰

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 邢德杰

(51)Int.Cl.
G06F 21/56(2013.01)

审查员 李莎

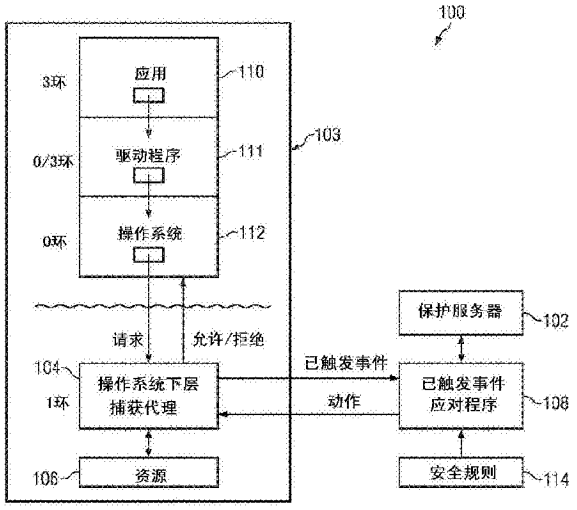
权利要求书4页 说明书153页 附图64页

(54)发明名称

用于基于虚拟机监视器的反恶意软件安全的系统和方法

(57)摘要

一种用于保护电子设备的系统包括存储器、处理器、驻留在存储器中以供由处理器执行的一个或多个操作系统、通信上耦合到操作系统的电子设备的资源、被配置为以低于访问资源的电子设备的所有操作系统的级别在电子设备上执行的虚拟机监视器、以及被配置为以低于访问资源的电子设备的所有操作系统的级别在电子设备上执行的安全代理。该虚拟机监视器被配置为截取从高于虚拟机监视器的级别做出的对资源的请求并向安全代理告知该请求。安全代理被配置为判断该请求是否指示恶意软件。



CN 103620613 B

1.一种用于针对恶意软件保护电子设备的系统,包括:

硬件处理器;

存储器,其与所述存储器通信耦合;

操作系统,用于在所述操作系统中加载和卸载驱动程序;

捕获代理,包含由所述处理器执行的所述存储器中的指令,并且用于捕获对所述操作系统的一个或多个资源的意图的访问,所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载,其中所述意图的访问是通过捕获包含用于系统功能的代码的存储器页面的执行来捕获的,所述系统功能用来加载或卸载所述驱动程序;和

已触发事件应对程序,包含由所述处理器执行的所述存储器中的指令;其中

所述捕获代理还用于将关于被捕获的尝试的信息发送给所述已触发事件应对程序,所述被捕获的尝试包括对驱动程序的加载或卸载;

所述已触发事件应对程序用于:

根据所述信息,访问一个或多个安全规则;

参照所述安全规则,评估意图的对驱动程序的加载或卸载;以及

将评价发送给所述捕获代理;并且

所述捕获代理还用于:

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时,采取矫正动作;以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时,允许所述意图的对驱动程序的加载或卸载;并且

所述捕获代理和所述已触发事件应对程序还用于以低于访问所述一个或多个资源的所述电子设备的所有操作系统的级别执行,包括在所述系统的存储器上运行但不使用操作系统。

2.如权利要求1所述的系统,其特征在于,所述捕获代理还用于捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行。

3.如权利要求1所述的系统,其特征在于,参照所述安全规则来评估意图的对驱动程序的加载或卸载包括:确定和评估将要加载的所述驱动程序的身份。

4.如权利要求1所述的系统,其特征在于,参照所述安全规则来评估意图的对驱动程序的加载或卸载包括:确定和评估尝试加载或卸载所述驱动程序的实体。

5.一种用于针对恶意软件保护电子设备的系统,包括:

硬件处理器;

存储器,其与所述存储器通信耦合;

操作系统,用于在所述操作系统中加载和卸载驱动程序;

捕获代理,包含由所述处理器执行的所述存储器中的指令,并且用于捕获对所述操作系统的一个或多个资源的意图的访问,所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载,其中所述意图的访问是通过捕获包含用于系统功能的代码的物理存储器地址的执行来捕获的,所述系统功能用来加载或卸载所述驱动程序;和

已触发事件应对程序,包含由所述处理器执行的所述存储器中的指令;其中

所述捕获代理还用于将关于被捕获的尝试的信息发送给所述已触发事件应对程序,所

述被捕获的尝试包括对驱动程序的加载或卸载；

所述已触发事件应对程序用于：

根据所述信息，访问一个或多个安全规则；

参照所述安全规则，评估意图的对驱动程序的加载或卸载；以及

将评价发送给所述捕获代理；并且

所述捕获代理还用于：

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时，采取矫正动作；以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时，允许所述意图的对驱动程序的加载或卸载；并且

所述捕获代理和所述已触发事件应对程序还用于在低于所述电子设备的所有操作系统的级别执行，包括访问所述系统的存储器但不使用操作系统。

6.如权利要求5所述的系统，其特征在于，所述捕获代理还用于捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行，所述系统功能是由所述操作系统提供的。

7.如权利要求5所述的系统，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估将要加载的所述驱动程序的身份。

8.如权利要求5所述的系统，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估尝试加载或卸载所述驱动程序的实体。

9.一种用于针对恶意软件保护电子设备的方法，包括：

捕获对操作系统的一个或多个资源的意图的访问，所述操作系统用于加载和卸载驱动程序，其中所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载，并且所述意图的访问是通过捕获包含用于系统功能的代码的存储器页面的执行来捕获的，所述系统功能用来加载或卸载所述驱动程序；

根据所述意图的访问，访问一个或多个安全规则；

参照所述安全规则，评价意图的对驱动程序的加载或卸载；

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时，采取矫正动作；以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时，允许所述意图的对驱动程序的加载或卸载，

其中所述捕获意图的访问的操作和所述评价意图的对驱动程序的加载或卸载的操作是在低于所述电子设备的所有操作系统的级别上执行的，包括访问所述电子设备的处理器但不使用操作系统。

10.如权利要求9所述的方法，其特征在于，所述捕获意图的访问还包括：捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行，所述系统功能是由所述操作系统提供的。

11.如权利要求9所述的方法，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估将要加载的所述驱动程序的身份。

12.如权利要求9所述的方法，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估尝试加载或卸载所述驱动程序的实体。

13.一种用于针对恶意软件保护电子设备的方法，包括：

捕获对操作系统的一个或多个资源的意图的访问，所述操作系统用于加载和卸载驱动程序，其中所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载，并且其中所述意图的访问是通过捕获包含用于系统功能的代码的物理存储器地址的执行来捕获的，所述系统功能用来加载或卸载所述驱动程序；

根据所述意图的访问，访问一个或多个安全规则；

参照所述安全规则，评价意图的对驱动程序的加载或卸载；

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时，采取矫正动作；以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时，允许所述意图的对驱动程序的加载或卸载，

其中所述捕获意图的访问的操作和所述评价意图的对驱动程序的加载或卸载的操作是在低于所述电子设备的所有操作系统的级别上执行的，包括访问所述电子设备的处理器但不使用操作系统。

14.如权利要求13所述的方法，其特征在于，所述捕获意图的访问还包括：捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行，所述系统功能是由所述操作系统提供的。

15.如权利要求13所述的方法，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估将要加载的所述驱动程序的身份。

16.如权利要求13所述的方法，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估尝试加载或卸载所述驱动程序的实体。

17.一种非瞬态计算机可读介质，包括存储于其上的计算机可执行指令，所述指令可被处理器读取，当所述指令被读取和执行时，使得所述处理器：

捕获对操作系统的一个或多个资源的意图的访问，所述操作系统用于加载和卸载驱动程序，其中所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载，并且所述意图的访问是通过捕获包含用于系统功能的代码的存储器页面的执行来捕获的，所述系统功能用来加载或卸载所述驱动程序；

根据所述意图的访问，访问一个或多个安全规则；

参照所述安全规则，评价意图的对驱动程序的加载或卸载；

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时，采取矫正动作；以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时，允许所述意图的对驱动程序的加载或卸载，

其中所述捕获意图的访问的操作和所述评价意图的对驱动程序的加载或卸载的操作是在低于电子设备的所有操作系统的级别上执行的，包括访问所述电子设备的处理器但不使用操作系统。

18.如权利要求17所述的非瞬态计算机可读介质，其特征在于，所述捕获意图的访问还包括：捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行，所述系统功能是由所述操作系统提供的。

19.如权利要求17所述的非瞬态计算机可读介质，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估将要加载的所述驱动程序的身份。

20.如权利要求17所述的非瞬态计算机可读介质，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估尝试加载或卸载所述驱动程序的实体。

21.一种非瞬态计算机可读介质，包括存储于其上的计算机可执行指令，所述指令可被处理器读取，当所述指令被读取和执行时，使得所述处理器：

捕获对操作系统的一个或多个资源的意图的访问，所述操作系统用于加载和卸载驱动程序，其中所述意图的访问包括在所述操作系统中意图的对驱动程序的加载或卸载，并且其中所述意图的访问是通过捕获包含用于系统功能的代码的物理存储器地址的执行来捕获的，所述系统功能用来加载或卸载所述驱动程序；

根据所述意图的访问，访问一个或多个安全规则；

参照所述安全规则，评价意图的对驱动程序的加载或卸载；

当所述评价包括意图的对驱动程序的加载或卸载为恶意的指示时，采取矫正动作；以及

当所述评价包括所述意图的对驱动程序的加载或卸载为安全时，允许所述意图的对驱动程序的加载或卸载，

其中所述捕获意图的访问的操作和所述评价意图的对驱动程序的加载或卸载的操作是在低于电子设备的所有操作系统的级别上执行的，包括访问所述电子设备的处理器但不使用操作系统。

22.如权利要求21所述的非瞬态计算机可读介质，其特征在于，所述捕获意图的访问还包括：捕获系统功能的用于加载或卸载所述驱动程序的子功能的意图的执行，所述系统功能是由所述操作系统提供的。

23.如权利要求21所述的非瞬态计算机可读介质，其特征在于，参照所述安全规则来评估意图的对驱动程序的加载或卸载包括：确定和评估将要加载的所述驱动程序的身份。

用于基于虚拟机监视器的反恶意软件安全的系统和方法

[0001] 优先权申请

[0002] 本申请要求以下美国申请的权益：于2011年3月28日提交的第13/073,791号、于2011年3月28日提交的第13/073,810号、于2011年3月28日提交的第13/073,842号、于2011年3月31日提交的第13/077,227号、于2011年3月28日提交的第13/073,853号、于2011年3月29日提交的第13/075,049号、于2011年3月31日提交的第13/076,493号、于2011年3月29日提交的第13/074,741号、于2011年3月31日提交的第13/077,305号、于2011年3月29日提交的第13/074,831号、于2011年3月29日提交的第13/074,925号、于2011年3月29日提交的第13/074,947号、于2011年3月31日提交的第13/077,270号、于2011年3月31日提交的第13/076,537号、于2011年3月28日提交的第13/073,864号、于2011年3月29日提交的第13/075,072号、于2011年3月29日提交的第13/075,101号、于2011年3月31日提交的第13/076,512号、于2011年3月31日提交的第13/076,480号以及于2011年3月31日提交的第13/076,473号，这些申请的内容通过引用整体合并于此。

技术领域

[0003] 本发明通常涉及计算机安全和恶意软件防护，且尤其涉及基于虚拟机监视器的反恶意软件安全的系统和方法。

背景技术

[0004] 本机操作系统服务可以防止安全软件在操作系统的内核内安装任意挂钩(hooking)。因而防止了安全软件过滤电子设备的所有行为，包括恶意软件的潜在恶意的动作。恶意软件可以包括但不限于间谍软件、rootkit、密码窃取器、垃圾邮件、网络钓鱼攻击源、拒绝服务攻击源、病毒、记录器、木马、广告软件或产生恶意活动的任何其他数字内容。

[0005] 由操作系统提供的过滤功能性可以受到限制，且仅在操作系统销售商决定的时间轴上可用。恶意软件可以以与安全软件相同的级别操作和驻留，尤其是在操作系统内核内，且因而危害操作系统和安全软件本身的完整性两者。

[0006] 多种形式的主动内核模式恶意软件篡改用户模式存储器来完成恶意任务，例如动态地注入恶意代码、修改用户模式代码段以便变更执行路径并重定向到恶意代码、以及修改用户模式数据结构以便使得安全软件失效。另外，一些恶意软件可以通过篡改进程存储器代码和数据片段以便欺骗检测逻辑来从内核攻击反恶意软件应用和进程。

[0007] 内核模式rootkit和其他恶意软件采用各种方法来对用户模式应用和内核模式设备驱动程序隐藏它们的存在。取决于感染发生的场所，所使用的技术可以改变。例如，恶意软件攻击操作系统的内核活动进程列表以便从列表中划去(delist)或解开(unlink)rootkit或其他恶意软件进程。其他恶意软件可以欺骗进程访问和枚举函数的代码段。

发明内容

[0008] 在一个实施例中，一种用于保护电子设备的系统，包括存储器、处理器、驻留在存

储器中以供由处理器执行的一个或多个操作系统、通信上耦合到操作系统的电子设备的资源、被配置为以低于访问资源的电子设备的所有操作系统的级别在电子设备上执行的虚拟机监视器、以及被配置为以低于访问资源的电子设备的所有操作系统的级别在电子设备上执行的安全代理。该虚拟机监视器被配置为截取从高于虚拟机监视器的级别做出的对资源的请求并向安全代理告知该请求。安全代理被配置为判断该请求是否指示恶意软件。

[0009] 在另一个实施例中，一种用于保护电子设备的系统，包括存储器、处理器、驻留在存储器中以供由处理器执行的一个或多个操作系统、通信上耦合到操作系统的电子设备的资源、被配置为以高于访问资源的电子设备的所有操作系统的优先级在电子设备上执行的虚拟机监视器、以及被配置为以高于访问资源的电子设备的所有操作系统的优先级在电子设备上执行的安全代理。优先级由处理器定义。虚拟机监视器被配置为截取从具有比虚拟机监视器低的优先级的实体做出的对资源的请求并向安全代理告知该请求。安全代理被配置为判断是否该请求指示恶意软件。

[0010] 在又一个实施例中，一种用于保护电子设备的系统，包括存储器、处理器、驻留在存储器中以供由处理器执行的一个或多个操作系统、耦合到操作系统的电子设备的资源、被配置为在比访问资源的电子设备的所有操作系统享有更多特权的执行环上在电子设备上执行的虚拟机监视器、被配置为在比电子设备的所有操作系统享有更多特权的执行环上在电子设备上执行的安全代理。虚拟机监视器被配置为截取对资源的请求，从比虚拟机监视器享有较少特权的执行环做出该请求并向安全代理告知该请求。安全代理被配置为判断该请求是否指示恶意软件。

[0011] 在再一个实施例中，一种用于保护电子设备的方法包括，以低于访问资源的电子设备的所有操作系统的级别，截取从较高级别做出的对电子设备的资源的请求并判断该请求是否指示恶意软件。该资源通信上耦合到操作系统。

[0012] 在进一步的实施例中，一种用于保护电子设备的方法包括，以比访问资源的电子设备的所有操作系统较高的优先级，截取从具有较低优先级的实体做出的对资源的请求并判断该请求是否指示恶意软件。这样的优先级由电子设备的处理器定义。

[0013] 在另一个进一步的实施例中，一种用于保护电子设备的方法包括，在比访问资源的电子设备的所有操作系统享有更多特权的执行环上，截取对资源的请求并判断该请求是否指示恶意软件。从享有较少特权的执行环做出该请求。

[0014] 在又一个进一步的实施例中，一种设备包括计算机可读介质和在计算机可读介质上携带的计算机可执行指令。指令可由处理器读取。在被读取和被执行时，指令用于使处理器以低于访问资源的电子设备的所有操作系统的级别截取从较高的级别做出的对电子设备的资源的请求并判断该请求是否指示恶意软件。该资源通信上耦合到操作系统。

[0015] 在再一个进一步的实施例中，一种设备包括计算机可读介质和在计算机可读介质上携带的计算机可执行指令。指令可由处理器读取。在被读取和被执行时，指令用于使处理器以比访问资源的电子设备的所有操作系统较高的优先级截取从具有较少的优先级的实体做出的对资源的请求并判断该请求是否指示恶意软件。优先级由处理器定义。

[0016] 在附加的实施例中，一种设备包括计算机可读介质和在计算机可读介质上携带的计算机可执行指令。指令可由处理器读取。在被读取和被执行时，指令用于使处理器在比访问资源的电子设备的所有操作系统享有更多特权的执行环上截取对资源的请求并判断该

请求是否指示恶意软件。从享有较少特权的执行环做出该请求。

附图说明

[0017] 为了更完整地理解本发明及其优点，现在参见结合附图阅读的以下撰写的描述，附图中：

[0018] 图1是用于保护电子设备免遭恶意软件的系统的示例实施例；

[0019] 图2是用于保护电子设备免遭恶意软件的基于虚拟机监视器的和基于安全规则的可配置安全解决方案的系统的示例实施例；

[0020] 图3是用于基于虚拟机监视器保护电子设备免遭恶意软件的方法的示例实施例；

[0021] 图4是用于保护电子设备免遭恶意软件的基于固件的和基于安全规则的系统的示例实施例；

[0022] 图5是用于保护电子设备免遭恶意软件的基于固件的解决方案的示例实施例的更详尽的视图；

[0023] 图6是基于固件的保护电子设备免遭恶意软件的方法的示例实施例；

[0024] 图7是用于针对恶意软件保护电子设备的基于微代码的系统的示例实施例；

[0025] 图8是基于微代码的保护电子设备免遭恶意软件的方法的示例实施例；

[0026] 图9是用于调节对电子设备上的安全敏感的处理器资源的软件访问的系统的示例实施例；

[0027] 图10是处理器资源控制结构的示例实施例；

[0028] 图11是用于调节对电子设备的安全敏感的处理器资源的软件访问的方法的示例实施例；

[0029] 图12用于调节软件访问的系统的示例实施例，该系统用于在电子设备上使用操作系统下层捕获(below-operating system trapping)来保护存储器；

[0030] 图13是存储器映射的示例实施例的阐释；

[0031] 图14是使用对电子设备的尝试访问的操作系统下层捕获来保护存储器的方法的示例实施例；

[0032] 图15是保护电子设备的操作系统内核的系统的示例实施例；

[0033] 图16是对操作系统的可信访问和可信驱动程序组件的访问映射的示例实施例；

[0034] 图17是进一步阐释图16的访问映射的虚拟存储器的示例实施例；

[0035] 图18是用于产生对操作系统的可信访问和可信驱动程序组件的访问映射的系统的示例实施例；以及

[0036] 图19是用于保护电子设备的操作系统内核的方法的示例实施例；

[0037] 图21是用于提供受保护操作系统执行环境的系统中的起动模块的示例实施例；

[0038] 图22是用于安全地执行操作系统的操作系统执行环境的示例实施例；

[0039] 图23是供用于提供受保护操作系统执行环境的系统或方法的盘映射位图的示例实施例；

[0040] 图24是用于起动受保护操作系统执行环境的方法的示例实施例；

[0041] 图25是提供用于安全地执行操作系统的操作系统执行环境的方法的示例实施例；

[0042] 图26是用于保护存储设备免遭未经授权的访问的系统的示例实施例；

[0043] 图27是供与用于保护存储设备免遭未经授权的访问的系统或方法一起使用的安全规则的示例实施例；

[0044] 图28是用于保护存储设备免遭未经授权的访问的方法的示例实施例；

[0045] 图29是用于保护在应用和输入/输出设备之间的写访问的输入/输出路径的系统的示例实施例；

[0046] 图30是用于保护在应用和输入/输出设备之间的写访问的输入/输出路径的方法的示例实施例；

[0047] 图31是用于保护在应用和输入/输出设备之间的读访问的输入/输出路径的系统的示例实施例；

[0048] 图32是用于保护在应用和输入/输出设备之间的读访问的输入/输出路径的方法的示例实施例；

[0049] 图33是用于检测和修复电子设备上的隐藏进程的系统的示例实施例；

[0050] 图34是用于检测和修复电子设备上的隐藏进程的方法的示例实施例；

[0051] 图35是用于检测和修复电子设备上的隐藏进程的另一系统的示例实施例；

[0052] 图36是用于检测和修复电子设备上的隐藏进程的另一方法的示例实施例；

[0053] 图37是用于检测和修复电子设备上的隐藏进程的又一方法的示例实施例；

[0054] 图38是用于保护对操作系统的系统调用的访问的系统的示例实施例；

[0055] 图39是供与保护对操作系统的系统调用的访问的系统或方法一起使用的系统调用表的示例实施例；

[0056] 图40是用于保护对操作系统的系统调用的访问的方法的示例实施例；

[0057] 图41是用于电子设备上恶意或潜在恶意的代码的调节和控制的系统的示例实施例；

[0058] 图42是用于电子设备上的自修改代码的调节和控制的方法的示例实施例；

[0059] 图43是用于电子设备上的恶意代码的修改的方法的示例实施例；

[0060] 图44是用于电子设备上的相关线程的监视和跟踪的方法的示例实施例；

[0061] 图45是用于保护电子设备的存储器和存储的系统的示例实施例；

[0062] 图46是用于保护电子设备的存储器和存储的方法的示例实施例；

[0063] 图47是用于保护对操作系统的对象的访问的系统的示例实施例；

[0064] 图48是供与保护对操作系统的对象的访问的系统或方法一起使用的行为状态映射的示例实施例；

[0065] 图49是用于保护对操作系统的对象的访问的方法的示例实施例；

[0066] 图50是用于保护在电子设备上的驱动程序之间的通信的系统的示例实施例；

[0067] 图51是驱动程序间通信的示例阐释；

[0068] 图52是O/S下层安全代理可以保护的电子设备的示例部分的附加阐释；

[0069] 图53是用于电子设备中的驱动程序间通信的操作系统下层捕获和保护的方法的示例实施例；

[0070] 图54是用于保护电子设备上驱动程序过滤器的附接和分开的系统的示例实施例；

[0071] 图55是示例设备栈区(device stack)的操作的更详尽的阐释；

[0072] 图56是可能已经受到进行附接或分开驱动程序过滤器的恶意软件危害的设备栈

区的示例阐释；

[0073] 图57是用于电子设备中的驱动程序过滤器附接的操作系统下层捕获的方法的示例实施例；

[0074] 图58是用于保护电子设备上的驱动程序的加载或卸载的系统的示例实施例；

[0075] 图59A和图59B是用于保护电子设备上的驱动程序的加载或卸载的方法的示例实施例；

[0076] 图60是用于操作系统下层捕获和保护把代码加载到存储器中的系统的示例实施例；

[0077] 图61是应用如何收集注入的代码以便放置在存储器中以供执行的示例阐释；

[0078] 图62A示出把应用的映像从磁盘加载到存储器的示例阐释；

[0079] 图62B示出在应用的映像被加载到存储器中之后实施的可能动作的示例阐释；

[0080] 图63阐释对所交换的内容恶意攻击以便注入代码的附加示例；

[0081] 图64是在一部分存储器已经被判断为恶意的之后的存储器映射的示例实施例；以及

[0082] 图65是存储器中的代码的加载和执行的操作系统下层捕获的方法的示例实施例。

具体实施方式

[0083] 图1是用于保护电子设备免遭恶意软件的系统100的示例实施例。系统100可以包括通信上耦合到已触发事件应对程序108的操作系统(“O/S”)下层捕获代理104。O/S下层捕获代理104可以被配置为捕获电子设备103的资源106的各种意图的(attempted)访问。O/S下层捕获代理104可以被配置为创建与已捕获的已尝试访问相关联的触发事件，并把已触发事件发送给触发事件应对程序108。触发事件应对程序108可以被配置为查阅一个或多个安全规则114或保护服务器102，以判断如何应对该触发事件。已触发事件应对程序108也可以被配置为评估已触发事件的倾向是恶意软件或破坏电子设备103的资源或操作的恶意尝试的指示。此外，已触发事件应对程序108可以被配置为向O/S下层捕获代理104提供应当允许还是拒绝已触发事件的判断，或可以被配置为产生另一矫正动作。

[0084] 可以在比电子设备103中的操作系统较低的功能级别实现O/S下层捕获代理104。例如，O/S下层捕获代理104可以截取操作系统112、驱动程序111或应用110对资源106的已尝试访问。O/S下层捕获代理104可以无需使用操作系统就运行在电子设备103的处理器上。在一个实施例中，O/S下层捕获代理104可以在裸机环境或执行级别上操作。另外，O/S下层捕获代理104可以运行在比电子设备103的所有操作系统高的执行优先级上，如电子设备103的处理器所定义的。例如，在其中较低的数字表示较高的优先级的使用保护环的分级保护域模型的上下文中，操作系统112可以在“0环(Ring0)”操作，同时O/S下层捕获代理104可以在“1环(Ring1)”操作。驱动程序111和应用110可以在“0环”或“3环(Ring3)”操作。在处理器的一些实施例中，“1环”的概念可以被称为“0环特权模式”，且“0环”的概念可以被称为“0环非特权模式”。“1环”或“0环特权模式”中的操作可以比“0环”或“0环非特权模式”需要附加的开销和支出。电子设备103的操作系统可以在0环运行。

[0085] O/S下层捕获代理104可以对在0环或更高的环运行的实体透明地操作。因而无论O/S下层捕获代理104是否存在都可以由操作系统112或另一实体以相同的方式请求对资源

106的尝试访问。在强加接收到的动作时，0/S下层捕获代理104可以允许该请求发生，可以拒绝该请求，或采取其他矫正动作。为了拒绝请求，0/S下层捕获代理104可以简单地不把请求传送给资源106或处理器，或可以向该请求提供欺骗的或假的应答以便使得操作系统112相信该动作已经发生。

[0086] 通过在“1环”、在比电子设备103的相关操作系统更高的优先级或低于电子设备103的相关操作系统运行，0/S下层捕获代理104可以避免困扰诸如操作系统112之类的操作系统的大多数恶意软件。恶意软件可以欺骗在“0环”运行的操作系统112或甚至反恶意软件的软件，这是因为恶意软件也可以在“0环”优先级运行。然而，如果要执行恶意活动，电子设备103上的恶意软件必须仍然做出对资源106的请求。因而，捕获被链接到敏感资源的操作可以由在低于电子设备103中的操作系统的级别下运行的捕获代理较好地完成。

[0087] 可以以任何合适的方式实现0/S下层捕获代理104。在一个实施例中，可以在虚拟机监视器中实现0/S下层捕获代理104。这样的实施例可以在低于操作系统的级别下操作，如对于0/S下层捕获代理104所描述的。例如，在下面的图2中的对安全虚拟机监视器216的讨论中可以找到这样的实施例的示例的描述。在另一实施例中，可以在固件中实现0/S下层捕获代理104。这样的实施例可以在低于操作系统的级别下操作，如对于0/S下层捕获代理104所描述的。例如，可以在下面的图4和图5中对固件安全代理440、516或PC固件安全代理444的讨论中找到这样的实施例的示例的描述。在又一个实施例中，可以在微代码中实现0/S下层捕获代理104。这样的实现可以在低于操作系统的级别下操作，如对于0/S下层捕获代理104所描述的。例如，在下面的图7中对微代码安全代理708的讨论中找到这样的实施例的示例的描述。可以在这些实施例的组合中实现0/S下层捕获代理104。

[0088] 已触发事件应对程序108可以由通信上耦合在一起的一个或多个事件应对程序或安全代理实现。可以在相同的安全代理中实现已触发事件应对程序108和0/S下层捕获代理104。在一个实施例中，已触发事件应对程序108可以在与0/S下层捕获代理相同的优先级环操作。在另一实施例中，已触发事件应对程序108可以在与操作系统112、驱动程序111或应用110相同的优先级操作。在再一个实施例中，已触发事件应对程序108可以由两个或更多个已触发事件应对程序实现，其中至少一个已触发事件应对程序在与0/S下层捕获代理相同的优先级环操作，且至少一个已触发事件应对程序在操作系统112、驱动程序111或应用110的级别操作。通过在0/S下层捕获代理104的级别运行，已触发事件应对程序108可以类似地避免“0环”或“3环”恶意软件感染代理本身的问题。然而，与操作系统112、驱动程序111或应用110一起在“0环”或“3环”运行的已触发事件应对程序108可能能够提供关于从“1环”代理的角度来看不可获得的对资源106的尝试访问的上下文信息。

[0089] 可以以任何合适的方式实现已触发事件应对程序108。在一个实施例中，可以在虚拟机监视器或虚拟机监视器安全代理中实现已触发事件应对程序108。这样的实施例可以在低于操作系统的级别下操作，如对于已触发事件应对程序108所描述的。例如，可以在下面的图2中对安全虚拟机监视器216或安全虚拟机监视器安全代理217的讨论中找到这样的实施例的示例的描述。在另一实施例中，可以完全地或部分地在固件实现已触发事件应对程序108。这样的实施例可以在低于操作系统的级别下操作，如对于已触发事件应对程序108所描述的。例如，可以在下面的图4和图5中对固件安全代理440、516或PC固件安全代理444的讨论中找到这样的实施例的示例的描述。也可以在图4中的0/S下层代理450中实现已

触发事件应对程序108，O/S下层代理450本身可以以像虚拟机监视器、固件或微代码那样的方式实现。在又一实施例中，可以在微代码中实现已触发事件应对程序108。这样的实现可以在低于操作系统的级别下操作，如对于已触发事件应对程序108所描述的。例如，可以在下面的图7中对微代码安全代理708的讨论中找到这样的实施例的示例的描述。也可以在图7的O/S下层代理712中实现已触发事件应对程序108，O/S下层代理712本身可以以像虚拟机监视器、固件或微代码那样的方式实现。可以在这些实施例的组合中实现已触发事件应对程序108。

[0090] 在一个实施例中，操作系统下层捕获代理104和/或已触发事件应对程序108可以在电子设备103的裸机层操作。操作系统下层捕获代理104和/或已触发事件应对程序108无需使用在它们和它们被配置为保护的资源106之间的操作系统就可以操作。资源106可以包括处理器、处理器的功能部件、存储器、诸如数据结构之类的驻留在存储器中的实体或诸如函数、进程或应用之类的驻留在存储器中以供由处理器执行的实体。操作系统下层捕获代理104和/或已触发事件应对程序108可以直接地在电子设备103的硬件上操作。操作系统下层捕获代理104和/或已触发事件应对程序108可以不要求使用诸如操作系统112之类的操作系统来执行，也不获得对资源106的完全访问。

[0091] 其他操作系统可以存在于电子设备103上，这些操作系统不参与在处于操作系统112、操作系统下层捕获代理104和已触发事件应对程序108的级别的实体与资源106之间的关系。例如，预引导操作系统可以安全地起动电子设备的各部分，但不参与电子设备在应对来自应用110、驱动程序111和操作系统112的对资源106做出的请求方面的正常操作。在另一示例中，电子设备103可以包含主板组件、插入式板卡、外围设备或其他组件，这些组件包含它们自己的各组操作系统和处理器，以执行在处于操作系统112、操作系统下层捕获代理104和已触发事件应对程序108的级别的实体与资源106之间的关系之外的功能。这些操作系统可以被嵌入到操作系统中。这些操作系统中的任何可以不被用来执行操作系统下层捕获代理104和已触发事件应对程序108。进一步，这些操作系统中的任何可以不访问受捕获代理104和已触发事件应对程序108保护的资源106。

[0092] 系统100可以包括一个或多个操作系统下层捕获代理104和一个或多个已触发事件应对程序108的任何组合。可以在对下面各图中的捕获代理、事件应对程序和安全代理的描述中找到操作系统下层捕获代理104和已触发事件应对程序108的描述。

[0093] 资源106可以包括电子设备的任何合适的资源。例如，资源106可以包括寄存器、存储器、控制器、或I/O设备。例如，可以在下面的图2的系统资源214、如图4中所示出的诸如显示器430和存储432之类的组件或图7的系统资源724的描述中找到资源106的示例实施例的描述。

[0094] 安全规则114可以包括任何合适的规则、逻辑、命令、指令、标志或用于向O/S下层捕获代理104告知要捕获什么动作或用于告知已触发事件应对程序108基于已捕获动作应对事件的其他机制。已触发事件应对程序108可以被配置为向O/S下层捕获代理提供一个或多个安全规则114。例如，可以在下面的图2的安全规则222、图4的安全规则422、434、436、438、图5的安全规则518、或图7的安全规则707、723的描述中找到安全规则114中的一些或全部的示例实施例的描述。

[0095] 可以以任何合适的方式实现诸如系统100的应用110、驱动程序111和操作系统112

之类的内核模式和用户模式实体。例如，可以在下面的图2的应用210、驱动程序211和操作系统212；图4的应用410、驱动程序411和操作系统412；以及图7的应用709、驱动程序711和操作系统713的描述中找到系统100的应用110、驱动程序111和操作系统112的示例实施例的描述。

[0096] 可以以任何合适的方式实现电子设备103，例如计算机、个人数字助理、电话、移动设备、服务器或可配置为解释和/或执行程序指令和/或进程数据的任何其他设备。例如，可以在图2的电子设备204、图4的电子设备404或图7的电子设备701的讨论中找到电子设备103的示例实施例的描述。

[0097] 可以在用于在低于电子设备103的操作系统的级别捕获对资源的尝试访问的任何合适的系统中实现系统100。也可以在用于通过查询安全规则以便判断尝试访问是否恶意来应对尝试访问的任何合适装置中实现系统100。例如，系统100可以由下面的图2-图8中所描述的系统和方法200、300、400、500、600、700和800实现。

[0098] 图2是用于保护电子设备免遭恶意软件的基于虚拟机监视器和基于安全规则的可配置安全解决方案的系统200的示例实施例。系统200可以是系统100的示例实施例，实现虚拟机监视器中的系统100的某些元素。系统200可以包括受可配置安全解决方案保护免遭恶意软件的电子设备204。系统200的可配置安全解决方案可以包括在所有操作系统下层运行的安全代理、安全虚拟机监视器、基于云的安全代理和O/S内部行为的安全代理。O/S下层安全代理和安全虚拟机监视器可以被配置为守护对电子设备204的系统资源(包括由O/S内部行为安全代理使用的资源)的访问。O/S下层安全代理可以在安全虚拟机监视器中运行。基于云的安全代理可以被配置为向O/S下层安全代理和O/S内部行为安全代理提供恶意软件检测信息，并从安全虚拟机监视器和O/S内部行为安全代理接收关于可能与恶意软件相关联的可疑行为的信息。O/S内部行为安全代理可以被配置为扫描电子设备204以便得到在电子设备上操作的恶意软件的痕迹。系统200可以包括一个或多个O/S下层安全代理，该一个或多个O/S下层安全代理被配置为捕获对电子设备204的资源的访问的尝试使用、产生对应于该尝试的已触发事件、查阅关于已触发事件的安全规则并且如果有必要则采取关于该尝试的矫正动作。

[0099] 在一个实施例中，系统200可以包括通信上耦合到一个或多个O/S内部安全代理218和安全虚拟机监视器(“SVMM”)安全代理217的保护服务器202。SVMM安全代理217可以驻留在SVMM 216中。SVMM 216可以在电子设备204上驻留并操作。O/S内部安全代理218和SVMM安全代理217可以通信上耦合。保护服务器202、O/S内部安全代理218、SVMM安全代理217和SVMM 216可以被配置为保护电子设备204免遭恶意软件的感染。

[0100] SVMM安全代理217可以是图1的已触发事件应对程序108的示例实施例。SVMM 216可以是图1的O/S下层捕获代理104的示例实施例。

[0101] 电子设备204可以包括被耦合到处理器206的存储器208。电子设备204可以包括出于任何合适的目的在电子设备上执行的一个或多个应用210或驱动程序211。电子设备204可以包括操作系统212。操作系统212可以被配置为向应用210或驱动程序211提供对电子设备204的系统资源214的访问。SVMM 216可以被配置为截取操作系统212对系统资源214的这样的调用。SVMM 216和SVMM安全代理217可以在低于操作系统212的级别操作。例如，SVMM 216和SVMM安全代理217可以直接在处理器206上以诸如“1环”之类的特权模式操作。

[0102] 处理器206可以包括例如微处理器、微控制器、数字信号处理器(DSP)、专用集成电路(ASIC)或被配置为解释和/或执行程序指令和/或进程数据的任何其他数字电路或模拟电路。在一些实施例中，处理器206可以解释和/或执行被存储在存储器208中的程序指令和/或进程数据。存储器208可以部分地或整体地被配置为应用存储器、系统存储器或两者。存储器208可以包括被配置为持有和/或容纳一个或多个存储器模块的任何系统、设备或装置；例如，存储器208可以包括只读存储器、随机存取存储器、固态存储器、或基于盘的存储器。每一存储器模块可以包括被配置为保留程序指令和/或数据一段时间的任何系统、设备或装置(例如，计算机可读的非暂态介质)。

[0103] 保护服务器202可以在网络244上操作。在网络244上操作的保护服务器202可以实现云计算方案。保护服务器202可以被配置为与电子设备204的元素通信以便更新恶意软件检测规则和信息。保护服务器202可以被配置为接收关于起源于电子设备204的可疑活动的信息并判断这样的可疑活动是否恶意软件感染的指示。操作系统212可以包括一个或多个O/S内部安全代理218。O/S内部安全代理218可以被配置为从保护服务器202接收监视和检测规则，例如O/S内部安全规则220。O/S内部安全代理218可以被配置为使用保护服务器202所接收的O/S内部安全规则220来监视和防止电子设备204上的可疑活动。O/S内部安全代理218可以被配置为向保护服务器202报告所检测到的可疑活动。O/S内部安全代理218可以被配置为阻止恶意软件操作并向保护服务器202报告这样的阻止。如果多于一个的O/S内部安全代理218存在于系统200中，则每一O/S内部安全代理218可以被配置为执行捕获确证或与O/S内部安全代理218相关联的其他任务的经指派部分。这样的部分可以由操作系统下层安全代理界定。例如，一个O/S内部安全代理218可以确证或调查MOV指令，同时另一O/S内部安全代理218可以确证或调查JMP指令。O/S内部安全代理218可以被配置为确定存储器中特定页面的生命周期。例如，O/S内部安全代理218可以知道通常由操作系统212用来分配存储器的页面的进程和步骤。类似地，O/S内部安全代理218可以知道通常由操作系统212用来把应用的映像加载到其加载器中的进程和步骤。这样的进程可以遵循静态的操作模式。因而，O/S内部安全代理218可以被配置为跟踪操作系统212的操作以便判断对于给定动作标准过程是否得到遵循。O/S内部安全代理218可以与SVMM安全代理217通信以便判断SVMM安全代理217所捕获的操作是否产生O/S内部安全代理218所观察到的相应预期动作。矛盾可以指示恶意软件已经尝试执行在操作系统212的正常操作之外的系统功能。因而，例如O/S内部安全代理218和SVMM安全代理217可以判断可疑的页面是直接由恶意软件加载到存储器中还是由操作系统加载器加载。这样的行为可以引起O/S内部安全代理218或SVMM安全代理217向保护服务器202报告信息，采用更主动的捕获和检查，或采取任何其他矫正措施。

[0104] 在一个实施例中，O/S内部安全代理219可以被配置为通过把自身嵌入在操作系统212内来提供上下文信息。例如，O/S内部安全代理219可以被配置为把自身或子部件寄存为驱动程序过滤器，并把自身附接到主驱动程序，以判断驱动程序看到或看不到什么。通过作为对NDIS.SYS的过滤器而附接，例如，O/S内部安全代理219可以被配置为报告操作系统212驱动程序所看见的文件I/O操作。

[0105] 在另一实施例中，O/S内部安全代理219可以被配置为向SVMM安全代理216或其他O/S下层安全代理提供从操作系统219内观察到的这样的信息，以供与在操作系统下观察到的信息进行比较。在两组信息之间的矛盾可以指示尝试隐藏自身的恶意软件的存在。例如，

O/S内部安全代理219可以钩住或过滤NDIS.SYS，并监视对特定文件的文件写入。SVMM安全代理216可以监视输入和输出命令。如果SVMM安全代理216基于由O/S内部安全代理219看见的函数调用列表确定比应该已经看见的更多的写入，那么，恶意软件可能在由操作系统212提供的函数外暗中写入到磁盘。

[0106] 可以以用于通信的任何合适的网络实现网络244，这样的网络诸如：因特网、内联网、广域网、局域网、回程网(back-haul-network)、对等网或其任何组合。保护服务器202可以使用从在各种电子设备204上运行的各种安全代理218提交的报告通过应用流行程度和口碑分析逻辑来进一步检测恶意软件。例如，可以把在电子设备204上所标识的可疑行为综合到规则中，供保护服务器202主动保护其他电子设备204。例如，基于可疑驱动程序已被报告的次数，可以确定这样的规则。例如，具有窄的或缓慢的分布模式的未知驱动程序可能与恶意软件相关联。另一方面，具有宽的和快速的分布的未知驱动程序可能与流行的和广泛可获得的应用的补丁相关联。在另一示例中，这样的检测到的驱动程序可能已经被在另一电子设备上运行的安全软件判断为已经访问宿主恶意软件已知的网站。这样的驱动程序可以被判断为与恶意软件相关联。

[0107] SVMM 216可以实现系统200的安全虚拟机监视函数中的一些或全部。SVMM 216可以被配置为截取在电子设备上运行的一个或多个操作系统对诸如寄存器、存储器或I/O设备之类的系统资源的访问。可以使用SVMM 216或被配置为根据本公开内容的教导保护电子设备204的任何其他虚拟机监视器来实现系统200的安全虚拟机监视函数。SVMM 216可以被配置为控制和过滤在操作系统212代表自身或代表在操作系统212上运行的应用210尝试访问系统资源214的同时由操作系统212采取的动作。SVMM 216可以在电子设备204上的操作系统212下运行且可以具有对操作系统212和应用210或驱动程序211可用的一些或全部处理器资源的控制权。应用210可以包括适合在电子设备204上运行的任何应用。驱动程序211可以包括适合在电子设备204上运行的任何驱动程序。可用于由SVMM 216控制的处理器资源可以包括被指派为用于虚拟化的那些资源。在一个实施例中，SVMM 216可以被配置为虚拟化系统资源214以供由操作系统212、应用210或驱动程序211访问。仅作为示例，这样的系统资源214可以包括输入-输出设备226、系统存储器228或处理器资源230。仅作为示例，处理器资源230可以包括常规寄存器232、调试寄存器234、存储器分段236、存储器分页238、中断240或标志242。I/O设备226可以包括对诸如键盘、显示器、鼠标或网卡之类的此类设备的访问。

[0108] SVMM 216可以被配置为捕获起源于操作系统212以便访问系统资源214的操作的执行。SVMM 216可以包括被配置为捕获对系统资源214的特定的尝试访问的控制结构。可以使用任何合适的控制结构。在一个实施例中，这样的控制结构可以包括虚拟机控制结构(“VMCS”)221。SVMM 216可以被配置为通过操纵在VMCS 221内的标志捕获这样的执行。SVMM 216可以被配置为捕获涉及对系统资源214的访问的操作系统212、应用210或驱动程序211的任何合适的操作。这样的所捕获的操作可以包括，例如：读取、写入和执行系统存储器228中的存储器的特定页面；从处理器寄存器230加载值和向其中存储值；或从I/O设备226读取和向其写入。任何这样的操作可以引起虚拟机退出(“VM Exit”)，这可以由SVMM 216捕获。SVMM 216可以被配置为捕获中断240的产生，中断240可以由处理器208产生或由操作系统212的元素发起。SVMM 216可以被配置为通过捕获IN(输入)和OUT(输出)指令来捕获从I/O

设备226尝试读取和向其写入。SVMM可以被配置为通过捕获对例如虚拟化技术直接I/O(Virtualization Technology Directed I/O,“VTd”)的机制的访问来捕获这样的指令。VTd可以根据处理器208允许I/O设备虚拟化。通过访问VTd设施,SVMM安全代理217可以被配置为确定由VTd连接的设备、确定来自操作系统212的元信息、I/O设备上的端口或其他合适的信息。SVMM安全代理217可以被配置为控制或捕获这样的虚拟化设备访问的操作。例如,SVMM安全代理217可以被配置为确定包含给予可编程I/O端口的I/O指派的I/O权限映射。SVMM安全代理217可以被配置为捕获可能由恶意软件做出的对这样的权限映射的访问,或使用这样的权限映射来判断操作系统212上的实体和I/O设备的请求的关系。

[0109] 在一个实施例中,SVMM安全代理217可以在SVMM 216中操作。在另一实施例中,SVMM安全代理217可以在SVMM 216外操作,但可以通信上耦合到SVMM 216。在这样的实施例中,SVMM安全代理217可以在低于诸如操作系统212之类的电子设备204的操作系统的级别下操作。SVMM安全代理217可以在与SVMM 216相同的级别和/或相同的优先级操作。SVMM安全代理217可以被配置为应对由SVMM 216触发或捕获的事件。SVMM安全代理217可以被配置为在低于操作系统212的级别下访问存储器228或盘的内容,以便检查内容免受内核级别rootkit的干扰。此外,SVMM安全代理217的一些操作可以由SVMM 216实现,且SVMM 216的一些操作可以由SVMM安全代理217实现。

[0110] 在关于什么动作将引起捕获或触发的方面,SVMM安全代理217可以被配置为设定SVMM 216的操作。在一个实施例中,SVMM 216可以被配置为把已捕获动作的检测传输给SVMM安全代理217。SVMM安全代理217可以被配置为查阅安全规则222以便判断该已捕获动作是否指示恶意软件或恶意活动,并且基于安全规则222可以把关于采取什么随后动作的指示提供给SVMM 216。这样的随后动作可以包括允许意图的动作、不允许意图的动作或采取其他矫正步骤。

[0111] 可以通过由O/S内部安全代理218收集的信息来协调SVMM 216和SVMM安全代理217捕获对系统资源214的意图的访问和执行的操作。O/S内部安全代理218可以被配置为把上下文提供给SVMM 216和SVMM安全代理217的捕获和应对操作。例如,特定的操作系统数据结构正常情况下只由特定的应用或服务写入。O/S内部安全代理218可以确定什么应用或进程当前在操作系统212上可见地运行并把该信息传输给SVMM安全代理217。如果特定的应用或服务不被列出为可见地运行,那么,对数据结构的尝试写入可能来自未经授权的应用或进程。

[0112] O/S内部安全代理218可以被配置为经由超级调用与SVMM 216和/或SVMM安全代理217通信。超级调用可以被实现为带有定义可以使用的可用请求的描述符表以及关联的输入和输出参数。这样的描述符表可以定义可能用于O/S内部安全代理218与SVMM 216和/或SVMM安全代理217通信的一个或多个请求。这样的描述符表也可以定义这样的请求的输入和输出参数可以位于存储器中的何处。

[0113] O/S内部安全代理218、SVMM安全代理217和保护服务器202可以被配置为相互认证。安全代理212、SVMM安全代理217和保护服务器202中的每一个可以被配置为不继续相互通信,除非各实体中的每一个都经过认证。SVMM216可以被配置为把O/S内部安全代理218映像定位在存储器206中,并使用密码签名算法来验证存储器206中的O/S内部安全代理218映像。在保护服务器202、O/S内部安全代理218和SVMM安全代理217之间的认证可以使用任何

合适的方法，包括密码散列和/或签名算法。在一个实施例中，这样的认证可以涉及私有密钥的交换。O/S内部安全代理218可以被配置为从保护服务器202接收密钥以便验证SVMM安全代理217的实例。

[0114] O/S内部安全代理218可以具有关于操作系统212的操作的上下文信息。O/S内部安全代理218可以被配置为与SVMM安全代理217通信以提供这样的上下文信息。SVMM安全代理217可以指示SVMM 216例如如何定义存储器的某些页面或捕获哪些寄存器。

[0115] SVMM 216可以被配置为捕获由SVMM安全代理217界定的对系统资源214的访问尝试。例如，为了捕获存储器访问，SVMM 216可以被配置为捕获诸如读、写或执行之类的操作。为了捕获对处理器寄存器230的访问，SVMM 216可以被指示为捕获包括加载、存储或读取寄存器值的操作。为了捕获I/O操作，I/O设备226、SVMM 216可以被指示捕获诸如对键盘、鼠标或其他外围设备的输入或输出之类的操作。与操作系统内部安全代理联合，下面各图中的SVMM安全代理217和/或其他操作系统下层安全代理可以被配置为对于I/O操作确定目标I/O设备226的身份、要在I/O设备226上执行的目标操作和要传输的数据。

[0116] SVMM安全代理217可以被配置为确定上下文信息，例如操作系统212的什么实体已经尝试访问电子设备204的资源，或者资源可以属于操作系统212的什么实体。SVMM安全代理217可以被配置为通过任何合适的方法做出这样的判定。在一个实施例中，SVMM安全代理217可以被配置为从操作系统内部安全代理218访问这样的判定的上下文信息。在另一实施例中，SVMM安全代理217可以被配置为直接地或间接地访问操作系统212的调用栈区和/或处理器208的执行栈区，以判断由操作系统212的不同进程或应用的调用次序。执行指令指针可以指向引起触发器的指令，同时执行栈区指针和执行基址指针可以指向栈区帧。通过栈区穿行执行基址指针，可以标识先前的函数调用，为即将到来的操作提供上下文。这样的栈区可以指示意图的操作以及源存储器位置。在又一实施例中，SVMM安全代理217可以被配置为结合安全规则222使用存储器映射来判断尝试是否恶意或指示恶意软件。例如，给定意图的访问的存储器位置，这样的存储器映射可以指示做出对资源的尝试访问的实体。例如，在虚拟存储器页面标识符和/或物理存储器地址中，可以定义这样的存储器映射。在另一示例中，这样的存储器映射可以指示对应于该尝试的目标的存储器位置的实体。使用存储器映射，SVMM安全代理217可以被配置为确定意图的访问的源和目标的身份或其实体所有者。在下面各图中，结合操作系统内部安全代理，通过监视系统的执行，可以部分地由SVMM安全代理217或其他O/S下层安全代理创建存储器映射。结合操作系统内部安全代理，下面各图中的SVMM安全代理217和/或其他操作系统下层安全代理可以为给定的存储器页面或物理地址判断这样的位置是否属于特定的代码部分或数据部分；它属于哪些模块、进程、应用、映像或其他实体；或者它是否与用户模式或内核模式条目相关联。结合操作系统内部安全代理，下面各图中的SVMM安全代理217和/或其他操作系统下层安全代理可以为虚拟存储器和物理存储器的映射确定指示在电子设备204上运行的各种实体的标识、位置和权限的元数据。类似地，下面各图中的SVMM安全代理217和/或其他操作系统下层安全代理可以使用大容量存储设备中的扇区的映射来判断这样的实体的映像在大容量存储设备中的位置。结合操作系统内部安全代理，下面各图中的SVMM安全代理217和/或其他操作系统下层安全代理可以为给定的实体确定它们可以驻留在上面的扇区、文件、目录和卷。

[0117] SVMM安全代理217可以被配置为分配存储器，例如O/S内部安全代理218、SVMM安全

代理217和SVMM 216的操作所要求的系统存储器228。SVMM安全代理217可以被配置为请求SVMM 216确保这样的已分配存储器免遭未经授权的读和写操作。SVMM 216可以被配置为在建立了存储器的保护之后初始化已分配存储器，以便消除恶意软件在由O/S内部安全代理218分配存储器和由SVMM216建立保护的时间之间添加恶意代码的机会。

[0118] SVMM安全代理217可以被配置为与保护服务器202通信以便安全地接收SVMM安全规则222。SVMM安全规则222可以包括指令、逻辑、规则、共享库、函数、模块或用于指示SVMM 216采用什么安全政策的任何其他合适的机制。SVMM安全代理217可以被配置为向保护服务器202传输关于来自电子设备204的可疑活动和已检测的恶意软件的信息。

[0119] O/S内部安全代理218可以被配置为与保护服务器202通信以便接收O/S内部安全规则220。O/S内部安全规则220可以包括指令、逻辑、规则、共享库、函数、模块或供O/S内部安全代理218检测电子设备204上的恶意软件的任何其他合适的机制。O/S内部安全代理218可以被配置为向保护服务器202传输关于电子设备204上的可疑活动和已检测恶意软件的信息。

[0120] O/S内部安全规则220和SVMM安全规则222均可以包括用于保护电子设备204免受恶意软件感染且用于检测可能包括恶意软件的可疑活动的保护规则。O/S内部安全代理安全规则可以包含可由O/S内部安全代理218执行且在O/S内部安全代理218内的规则。SVMM安全规则222可以包含可由SVMM 216和/或SVMM安全代理217执行且在SVMM 216和/或SVMM安全代理217内的规则。

[0121] SVMM安全规则222可以被配置为向SVMM安全代理217提供带有如何观察和检测电子设备204的恶意软件感染的定义的信息。例如，SVMM安全规则222可以包括来自诸如应用210或驱动程序211之类的实体的什么类型的函数调用或行为的分类，SVMM安全代理217可以监视这些函数调用或行为以便得到恶意软件的指示。作为另一示例，SVMM安全规则222可以包括SVMM安全代理217如何处理这样的已触发函数调用的定义，包括使用什么参数、如何从这样的调用提取值或者如何确证这样的调用的操作。此外，SVMM安全规则222可以包括用于SVMM内部安全代理217的关于如何监视诸如应用210或驱动程序211之类的电子设备的实体的行为的信息，以及这样的行为的检测规则的例外。作为又一示例，SVMM安全规则222可以包括用于SVMM安全代理217的关于如何防止和修复通过这样的行为的检测规则检测到的恶意行为的信息。SVMM安全规则222可以包括SVMM安全代理217应监视、收集什么数据并将其发送到保护服务器202的细节。

[0122] 类似地，O/S内部安全规则220可以被配置为向O/S内部安全代理218提供带有如何观察和检测电子设备204的恶意软件感染的定义以及如何与SVMM安全代理217协调这样的活动的信息。

[0123] SVMM安全规则222也可以包括关于SVMM 216将捕获的什么动作的规则。SVMM安全代理217可以被配置为把这样的规则应用到SVMM 216。例如，SVMM安全代理217可以被配置为转换要被捕获到存储器的可识别的虚拟或物理页面中的函数的地址、创建SVMM 216捕获这样的页面的执行的请求以及随后在捕获该执行之后调用安全代理217。SVMM安全代理217可以被配置为通过其与SVMM 216的接口接收SVMM安全规则222。这样的接口可以包括基于超级调用的接口。SVMM安全代理217可以被配置为通过相同的基于超级调用的接口把任何得到的检测或报告推送给SVMM 216。

[0124] 在一个实施例中，SVMM 216可以被配置为无需查询SVMM安全代理217就处理已触发动作。在这样的实施例中，SVMM 216可以被配置为安装在SVMM216内处理的附加触发器，该触发器可以不被传送给SVMM安全代理217。这样的附加触发器可以由SVMM安全规则222界定。在一个实施例中，SVMM安全规则222可以定义用于SVMM 216的存储器页面扫描规则。这样的规则可以包括哪些是恶意的且不应允许驻留在存储器中的实体或修正的列表。这样的规则也可以包括白名单，被配置为包括专门允许存在于系统存储器228内的页面的列表。在另一实施例中，SVMM安全规则222可以定义SVMM 216存储器页面访问规则。这样的规则可以包括允许什么代码页面或相反地禁止访问给定代码或数据页面的定义。因此，SVMM安全规则222可以被配置为指示SVMM 216按照存储器扫描器动作，和/或控制对存储器页面的访问。

[0125] SVMM 216可以被配置为通过阻止对系统资源214中它们各自的代码和数据页面的未经授权的读访问和写访问来保护SVMM安全代理217、SVMM 216和O/S内部安全代理218。例如，如果应用210或驱动程序211做出对系统存储器228、处理器寄存器230或I/O设备226中的一部分的、将引起影响SVMM安全代理217、SVMM 216和O/S内部安全代理218的完整性或操作的请求，那么，SVMM 216可以被配置为截取这样的尝试请求，且随后重新路由该请求、拒绝它或采取其他适当的动作。在另一示例中，SVMM 216可以被配置为授权对系统存储器228、处理器寄存器230或I/O设备226的一部分的读取访问，这些读取访问影响SVMM安全代理217、SVMM 216和用于诸如SVMM安全代理217本身之类的存储器安全软件应用或其他相应的或附属的程序的O/S内部安全代理218。这样的授权可以在SVMM安全规则222内定义，SVMM安全规则222可以定义SVMM 216如何应对对诸如系统存储器228之类的系统资源214的访问。在一个实施例中，SVMM安全规则222可以包括可信安全程序的白名单，该白名单可以包括SVMM安全代理217。

[0126] 为了与保护服务器202通信，SVMM 216可以包括受保护网络接口224。受保护网络接口224可以被配置为提供在诸如保护服务器202之类的网络服务器和诸如SVMM 216或SVMM安全代理217之类的电子设备204的元素之间的安全访问。SVMM 216可以包括可以实现受保护网络接口224的逻辑TCP/IP驱动程序或其他通信接口。保护服务器202可以被配置为经由受保护网络接口224通信以便指示SVMM 216或SVMM安全代理217更新自身，并且提供诸如SVMM安全规则222或O/S内部安全规则220之类的保护规则。保护服务器202可以被配置为递送用于特定电子设备204或特定SVMM 216的自定义规则。这样的定制可以包括电子设备204上已经报告的恶意活动的类型以及在电子设备204内的诸如抗病毒程序、防火墙或其他保护机制之类的其他保护机制。在一个实施例中，保护服务器202可以由电子设备204的管理员例如在本地网络上操作。在这样的情况中，管理员可以由从保护服务器202接收到的规则实现的、用于应对可疑行为的设置全局或个性化政策。SVMM 216可以包括告知SVMM 216或SVMM安全代理217如何通过经由保护服务器202安全递送的新映像来更新自身的更新引擎。

[0127] O/S内部安全规则220和SVMM安全规则222均可以被配置为请求把电子设备204上的特定的所观察到的动作或操作或者各类所观察到的动作或操作传送给保护服务器202。因此，保护服务器可以在允许动作在电子设备204上进行之前检查和验证观察结果。保护服务器202可以被配置为接受这样的动作以便同步地或异步地检查。在一个实施例中，O/S内

部安全代理218可以被配置为把有问题的活动、代码或数据的片段或动作传送给SVMM 216以供由保护服务器202验证。例如，O/S内部安全代理218可以通过检测在存储器内加载的未签名驱动程序来检测可疑的恶意软件实例。SVMM 216可以从O/S内部安全代理218接收关于可疑软件的信息，且可以把它提供给保护服务器202。

[0128] SVMM安全规则222可以被配置为允许或拒绝对电子设备的任何合适的系统资源的访问。可用于被监视的这样的资源可以取决于由处理器206公开的资源。例如，在一个实施例中，SVMM安全规则222可以被配置为允许SVMM 216限定对系统存储器228、I/O设备226和中断140的访问。这样的限制可以防止对诸如键盘、显示器或可移动盘之类的I/O设备的未经授权的访问。在另一实施例中，SVMM安全规则222可以被配置为允许SVMM 216限定对中断描述符表条目的访问，包括诸如中断240之类的在处理器寄存器中的条目。在又一实施例中，SVMM安全规则222可以被配置为允许SVMM 216限定对扩展页面表(“EPT”)或应对虚拟存储器(从客户操作系统的角度来看是真实存储器)到宿主物理存储器的映射任何其他机制的访问。

[0129] 如果除了处理器208之外电子设备204还包含支持虚拟化的一个或多个处理器，则SVMM 216或SVMM 216的另一实例可以被配置为截取对访问这样的其他处理器的虚拟化资源的尝试。如果电子设备204包含例如包含处理器208的四核处理器，则四核处理器的资源可受SVMM 216保护。如果一个或多个其他处理器不支持虚拟化，则SVMM 216不可以保护对它们的资源的访问。如果一个或多个其他处理器支持与处理器208不同的虚拟化技术，则SVMM 216可以被配置为保护对它们的资源的访问，但是以不同于保护处理器208的方式，这是由于虚拟化资源的方式不同。

[0130] 在操作中，保护服务器可以在网络244上运行。通过扫描电子设备204以便发现恶意软件，观察诸如电子设备204上的应用210和驱动程序211之类的实体的行为以便发现可疑行为，并通过修复所找到的任何这样的感染，O/S内部安全代理218可以在电子设备204上运行以便保护电子设备204免受恶意软件感染。O/S内部安全代理218可以运行在与操作系统212相同的优先级或级别，且可以运行在操作系统212中。SVMM 216可以在电子设备204上操作以便通过捕获对电子设备204的系统资源的意图的访问来保护电子设备204免受恶意软件感染。SVMM安全代理217可以运行在电子设备204或另一合适的电子设备上，以便设置SVMM 216的捕获操作并应对所捕获的对系统资源的尝试访问中的一些或全部。SVMM 216和SVMM安全代理217可以运行在低于具有优先级“1环”的操作系统212之下。SVMM安全代理217可以在SVMM 216上运行。

[0131] 保护服务器202可以把诸如SVMM安全规则222和O/S内部安全规则220之类的安全规则发送到电子设备204。这样的规则可以由SVMM安全代理217接收，这可以把O/S内部安全规则220提供给SVMM 216。这样的规则可以由O/S内部安全代理218接收。

[0132] 保护服务器202、安全代理218和SVMM安全代理217均可以相互认证。SVMM安全代理217可以在存储器中定位安全代理218的映像，并使用密码签名算法来验证驻留在存储器中的安全代理218的映像。保护服务器202和SVMM安全代理217可以使用密码散列和签名算法以便正确的相互标识来相互认证。SVMM安全代理217和保护服务器202也可以交换私有密钥以便认证相互的身份。安全代理218可以从保护服务器202接收密钥以便验证SVMM安全代理217的实例。可以不完全建立在安全代理218、SVMM安全代理217和SVMM安全代理202之间的

通信，除非代理中的每一个都相互认证。类似地，如果SVMM安全代理217和SVMM 216作为分离的实体而运行，则它们可以相互验证和认证。

[0133] SVMM 216和SVMM安全代理217可以在电子设备204的操作系统212和所有操作系统下运行。SVMM 216可以监视操作系统212、安全代理218、应用210和驱动程序211对包括I/O设备226、系统存储器228和处理器寄存器230在内的系统资源214的访问。SVMM 216可以捕获操作系统212、安全代理218、应用210、驱动程序211或电子设备204的任何其他实体所请求的关键操作的执行。SVMM 216可以通过操纵在VMCS 221中的标志来捕获这样的执行。当VMCS 221截取对受保护的资源的请求时，操作可以被移交给SVMM 216以供进一步操作、诊断和修复。在一个实施例中，操作可以随后由SVMM安全代理217操作。在另一实施例中，已捕获操作的应对可以由SVMM 216本身实施。SVMM 216可以捕获电子设备204的任何必要操作以提供针对恶意软件的保护。这样的操作可以包括但不限于：系统存储器228中特定的代码或数据页面的读取、写入和执行；从系统寄存器和处理器寄存器230加载和存储值；或者从I/O设备226读取或向其写入。将由SVMM 216捕获的特定的操作可以由SVMM安全规则222定义。

[0134] 保护服务器202可以与SVMM安全代理217或O/S内部安全代理218通信以便向每一个提供安全规则。在一个实施例中，保护服务器202可以把SVMM安全规则222递送给SVMM安全代理217。在另一实施例中，保护服务器202可以把O/S内部安全规则220递送给O/S内部安全代理218。在又一实施例中，保护服务器202可以把O/S内部安全规则220递送给SVMM安全代理217，SVMM安全代理217然后可以把规则提供给O/S内部安全代理218。

[0135] 应用210、驱动程序211或操作电子设备204的其他实体可以被O/S内部安全代理218观察到。O/S内部安全代理218可以使用O/S内部安全规则220来观察这样的处理实体的行为以便判断它们的行为是否构成了指示恶意软件的可能感染的可疑行为。一旦这样检测到可疑活动，O/S内部安全代理218可以把可疑信息提供给保护服务器202以供进一步分析和指示。O/S内部安全规则220可以向O/S内部安全代理218指出，这样的行为是可疑的，并且指出矫正动作。例如，应用210可以与宿主恶意软件已知的网络目的地通信。O/S内部安全代理218可以注意到应用210的活动，且随后阻止应用210对网络目的地的网络访问。O/S内部安全代理218也可以扫描电子设备204以便发现恶意软件。例如，O/S内部安全代理218可以检查存储器206或系统存储器228的内容，以便得到对应于恶意软件的签名的模式。这样的检查可以揭示，例如，应用210包含对应于恶意软件的已知片段的一块代码。然后，O/S内部安全代理218可以通过修复应用210、移除应用210或采取任何其他合适的动作来从电子设备204清除恶意软件的感染。O/S内部安全代理218可以就任何已检测可疑行为或恶意软件的其他指示与保护服务器202通信，且可以从保护服务器202接收关于如何处理这样的恶意软件的指示。

[0136] 在一个实施例中，SVMM安全代理217可以被配置为基于做出意图的操作的实体的起源评估已捕获操作。例如，如果驱动程序是从未知域下载的，或具有来自未知担保人的证书，那么，该驱动程序随后操作的能力受到限制。例如，可以对其状态未知的驱动程序否决把自身附接到另一驱动程序的能力。如果驱动程序是从宿主恶意软件已知的域下载的，或者包含欺诈凭证，那么，可以甚至不准许加载该驱动程序。类似地，如果已知驱动程序是来自特定的域或由特定的作者创建，那么，SVMM安全代理217可以被配置为识别电子设备204

中被授权为更新驱动程序的服务，并限制把驱动程序写入或存取到那些服务的能力。例如，来自公司X的内核驱动程序可以仅被写如到驻留在电子设备204上的公司X的更新服务软件。SVMM安全代理217可以被配置为确证更新服务的操作和完整性。在另一实施例中，SVMM安全代理217可以被配置为基于尝试的目标评估已捕获操作。例如，对于内核驱动程序可以捕获来自服务的更新软件的尝试，但不适用于应用软件。

[0137] 一旦实体已经被判断是可疑的，或尝试被判断为指示恶意软件，则可以链接引起尝试的该进程和容纳该进程的存储器。访问存储器的相同的部分的其他进程可以类似地被判断为恶意软件。可以存储访问资源的已捕获尝试，且可以根据原始事件评估访问受保护的资源的随后尝试。例如，恶意操作可以要求把代码写入到数据片段然后执行代码。因而，SVMM安全代理217可以捕获对该数据片段的原始写访问，允许写入，但是记录该写访问的源。随后，SVMM安全代理217可以捕获执行数据片段的随后尝试，并根据先前捕获的操作、尝试这种操作的实体或其他合适的取证信息评估该尝试的恶意状态。

[0138] SVMM安全代理217可以就SVMM 216要通过诸如VMCS 221之类的控制结构捕获哪些系统资源214的问题指示SVMM 216。然后，SVMM 216可以捕获起源于电子设备204的诸如操作系统212、应用210或驱动程序211之类的实体的对系统资源214的访问请求。例如，如果做出请求读取、写入或执行系统存储器228的部分，则SVMM 216可以通过在VMCS 221中设定用于系统存储器的已指派部分的标志来截取这样的请求。在另一示例中，可以由VMCS 221截取对I/O设备226做出的访问请求，例如输入或输出操作。在又一示例，诸如加载或存储命令之类的进程寄存器230中的请求可以由VMCS 221捕获。任何这样的捕获都可以导致向SVMM 216通知已尝试访问。一旦SVMM 216已经捕获对系统资源214的意图的操作，SVMM 216就可以把这样的已捕获执行传输给SVMM安全代理217。

[0139] O/S内部安全代理218和SVMM安全代理217可以通信以便确定在操作系统212内实施的操作的上下文。例如，来自操作系统212对电子设备204的特定资源的已捕获系统调用可以源自存储器的特定部分。SVMM安全代理217可以与O/S内部安全代理218通信，以判断什么应用、进程或其他实体驻留在存储器的该特定部分内。

[0140] 然后，基于SVMM安全规则222和来自O/S内部安全代理218的已捕获操作和/或上下文信息，SVMM安全代理217可以判断这样的访问是否构成可疑动作，例如指示恶意软件的感染的那些动作。例如，未经授权的应用尝试改变受保护存储器空间的系统存储器228可能是可疑活动，且因而由SVMM 216检测到的这样的尝试改变可以由SVMM安全代理217解释为恶意软件的操作。这样的活动可以被报告给保护服务器202以得到进一步指示，或者可以由O/S内部安全规则220指示动作。这样的检测的结果可以是阻止对系统存储器228的尝试改变，或触发对产生该尝试改变的电子设备204的实体的附加清除操作。

[0141] SVMM 216可以监视对系统资源214的附加调用以便保护SVMM 216、SVMM安全代理217和/或O/S内部安全代理218的完整性。SVMM 216可以实施由SVMM安全规则222定义的扫描操作，以便扫描系统存储器228的各部分，判断这样的存储器的各部分是否已经被恶意软件修改。SVMM 216可以利用签名、散列或指示已知存储器的给定模式是不安全的还是安全的其他规则。

[0142] 例如，SVMM 216可以通过阻止对对应于系统存储器228中的O/S内部安全代理218的代码和数据页面的未经授权的读取和写入访问来保护O/S内部安全代理218。一些恶意软

件可能通过对与系统存储器228相关联的系统资源214做出存储器修改或其他修改来尝试攻击O/S内部安全代理218。SVMM 216可以读取SVMM安全规则222中所包含的准许其变更代码或数据或对应于O/S内部安全代理218的其他系统资源214的已授权应用和电子设备204的其他实体的白名单。如果修改源自未被包含在白名单内的实体，那么，SVMM 216可以确定这样的修改与恶意软件相关联。对对应于O/S内部安全代理218的系统资源214的未经授权的访问可以由SVMM以任何合适的方式应对，这些方式包括阻止访问、创建蜜罐进程、向保护服务器202报告违规或任何其他合适的补救。

[0143] SVMM216也可以捕获对属于电子设备204的其他实体的系统资源214的访问。例如，系统存储器228中的目标存储器页面可以包含属于操作系统212的内核操作的一部分的样本代码或数据。SVMM216和SVMM安全规则222可以把对这样的目标页面的访问限制为仅经过授权的代码段。因此，如果系统存储器228中的代码页面尝试读取或变更目标存储器页面，且该代码页面属于电子设备204的未经授权实体，则这样的访问可以受到SVMM216阻止。因而，SVMM216可以用于控制对系统存储器228中的存储器页面的访问。

[0144] SVMM安全代理217可能能够通过联系保护服务器202以便得到经更新的规则来更新SVMM安全规则222或O/S内部安全规则220。保护服务器202可以基于所观察到的特定恶意软件、管理员设置或电子设备204的其他特性配置要递送给SVMM安全代理217的规则。SVMM安全代理217可以根据用户的要求、周期性地或根据重大事件的发生(例如遇到可以链接到恶意软件的新的可疑活动)更新电子设备204的规则。

[0145] SVMM安全代理217可以在对应于复合条件的VMCS中设置标志。可以跨越不同类型的资源捕获这样的标志。例如，VMCS可以被配置为捕获把某些值写入到存储器中的页面并随后把该页面移动到I/O设备的缓存的组合。

[0146] 系统200可以包含优于反恶意软件系统和软件的其他实现的一个或多个优点。例如，一些反恶意软件解决方案可以钩取操作系统的各种部分以便捕获和评估应用的低级操作。然而，这些解决方案自身可以在操作系统中操作或者在两个客户操作系统的情况中是在另一操作系统中操作。通过在操作系统的约束内操作，即使是以内核级的优先级操作，反恶意软件解决方案可能容易受到来自也运行在相同的操作系统上且也许运行在相同的优先级的恶意软件的恶意软件攻击的感染。如果在操作系统的级别实施对某些事件的捕获或触发，则这样的捕获或触发可能被运行在与操作系统相同的或较低的优先级的恶意软件钓鱼、钩取、逆向工程、危害或以另外方式挫败。例如，在操作系统上运行的检测和移除操作系统中的恶意挂钩的反恶意软件解决方案可以被在相同的优先级运行的恶意软件观察到。在另一示例中，作为过滤驱动程序寄存以便检测某些例程的操作的反恶意软件解决方案可以被在驱动程序栈区上比反恶意软件解决方案较低处寄存恶意过滤驱动程序的恶意软件挫败。类似地，如果某些已捕获或已触发事件的应对发生在操作系统的级别，则恶意软件可以影响这样的应对。例如，恶意软件可以撤销反恶意软件解决方案的修正，或甚至禁用反恶意软件解决方案的操作。

[0147] 在另一示例中，管理程序可以工作为虚拟化对诸如系统存储器228之类的系统资源的访问，但可能不会有条件地保护对系统资源的访问，且因而充当安全管理程序。这样的管理程序可以不拥有对诸如安全规则222中的行为规则之类的反恶意软件规则的访问权，以便标识恶意活动、实体或对系统资源的恶意的已尝试访问。这样的管理程序可以在操作

系统自身内运行，这可能易于遭受在与操作系统相同的优先级水平运行的恶意软件。这样的管理程序可以不是以“0环特权模式”运行，这是因为这样的模式可以要求管理程序截取太多对系统资源的尝试访问。可以给管理程序安排虚拟化客户操作系统的所有方面的任务，且这样的虚拟化的需求可能太过昂贵而不能同时地访问安全规则以便检查恶意行为。

[0148] 图3是用于基于虚拟机监视器的保护电子设备免遭恶意软件的方法300的示例实施例。在步骤305，可以认证O/S下层安全代理、O/S内部安全代理、保护服务器和虚拟机监视器的身份和安全。可以通过任何合适的方法完成这样的认证，包括通过定位和检验位于存储器中的每一个的映像，密码散列，或密钥。直到步骤305完成之前，可以停止其他步骤的操作。

[0149] 在步骤310，可以访问保护服务器以判断安全规则。这样的安全规则可以被用来在步骤315-380中做出判定。在步骤315，可以指示虚拟机监视器捕获对系统资源的访问。这样的访问可以源自在电子设备上运行的应用、驱动程序或操作系统。可以就要监视的电子设备的什么系统资源指示虚拟机监视器。也可以就要捕获的对所监视的系统资源的什么操作指示虚拟机监视器。例如，可以捕获对系统存储器的读、写或执行操作。在另一示例中，可以捕获对寄存器的加载或存储操作。在又一示例中，可以捕获对I/O设备的输入或输出动作。

[0150] 在步骤320，可以在诸如虚拟机控制结构之类的控制结构中设置对应于要捕获的这样的操作的标志。这样的已捕获操作可以产生VM退出，其中，在访问已标记资源时创建已触发事件。在步骤325，当系统存储器被分配给虚拟机监视器、O/S内部安全代理和O/S下层安全代理时，可以保护这样的存储器免遭未经授权的读和写操作。

[0151] 电子设备可以操作且通过在步骤330-340中捕获对系统资源的访问、在步骤345-355中扫描存储器以便发现恶意软件的存在以及在步骤360-365中扫描存储器以便发现已尝试存储器修改中的一种或多种来得到保护。捕获对系统资源的访问、扫描存储器以便发现恶意软件的存在以及扫描存储器以便发现已尝试存储器修改中的每一种可以并行实施。进一步，根据保护电子设备的操作的要求，可以重复这些中的每一种。

[0152] 在步骤330，可以捕获对诸如系统存储器、寄存器、或I/O设备之类的系统资源的访问。可以使用产生VM退出的VMCS标志来捕获访问。可以在低于在电子设备上运行的操作系统的级别实施这样的捕获。在步骤335，可以分析访问以便判断请求实体是否具有访问所请求的资源的权限。可以访问与已尝试访问相关联的上下文信息以便做出这样的判断。可以访问安全规则以便做出这样的判断。可以判断未经授权的访问是可疑的。可以在低于在电子设备上运行的操作系统的级别做出这样的应对和判定。如果访问是可疑的，那么，在步骤340，可以阻止对系统资源的可疑的已尝试访问。可以把这样的尝试报告给保护服务器。如果访问不是可疑的，那么，在步骤370可以允许访问。

[0153] 在步骤345，可以扫描电子设备的存储器页面以便发现恶意软件的存在。尽管扫描电子设备的存储器，但可以使用白名单来判断是否已知反映驻留在电子设备上的实体的存储器模式是安全的。如果遇到已知是安全的存储器模式，那么，在步骤370，可以允许存储器继续拥有对电子设备的访问权，且可以保持。尽管扫描电子设备的存储器，但可以使用黑名单来判断是否已知存储器模式包括恶意软件或与恶意软件相关联。可以通过访问安全规则来访问白名单和黑名单。在步骤350，如果找到了已知与恶意软件相关联的存储器模式，那么，在步骤375可以通过修复、移除或废止拒绝该存储器模式访问电子设备。

[0154] 在步骤355，可以扫描存储器以便判断是否已经或者正在尝试修改存储器。可以在低于电子设备中的操作系统的级别进行这样的扫描。这样的存储器可以包括内核存储器、系统数据结构或可以被恶意软件修改的电子设备的存储器的任何其他部分。例如，可以修改在电子设备上运行的活动线程的列表以便隐藏恶意进程的存在。如果检测到修改，那么，在步骤365可以判断这样的修改是否得到许可。这样的修改是否得到许可可以由安全规则定义。例如，可以保护反恶意软件进程的代码或数据页面免遭任何其他进程的修改或访问。如果存储器修改被视为经过授权，那么，在步骤370，可以允许修改。如果判断存储器修改未经授权且不被允许，那么，在步骤375，可以拒绝修改。

[0155] 在步骤370，如果允许访问或修改，那么，可以存储访问或修改以供稍后引用。对恶意软件的一些检测可以利用关于过往访问或修改的信息来判断这样的过往访问和目前所检测的访问一起是否包括对资源的恶意访问。

[0156] 在步骤375，如果拒绝修改、访问或其他操作，那么，在步骤380可以向保护服务器报告这样的事件。这样的报告可以包括关于任何关联的恶意软件或可疑行为的信息。

[0157] 根据保护电子设备的需要，可以连续地、周期性地或根据需求重复方法300的各步骤。

[0158] 图4是用于保护电子设备404免遭恶意软件的基于固件的和基于安全规则的系统400的示例实施例。系统400可以是系统100的示例实施例，其中，以固件实现系统100的某些元素。可以在低于电子设备404的操作系统的级别实施捕获系统400的操作。系统400可以包括被配置为捕获用于使用或访问电子设备404的资源的、诸如I/O命令之类的请求的一个或多个O/S下层安全代理。这样的O/S下层安全代理可以被配置为管理在设备之间或者与电子设备404的主处理器的输入和输出数据的交换。可以在电子设备404的诸如设备控制器之类的组件的固件中或在电子设备404自身的固件中实现这样的O/S下层安全代理。这样的固件可以驻留在非易失性存储器中。电子设备404的这样的资源可以包括图1的系统资源106或其各种可能的实施例，或者是被耦合到系统400中的设备或在其中实现的资源。系统400可以包括一个或多个O/S下层安全代理，O/S下层安全代理被配置为捕获对电子设备404的资源的访问的尝试使用、产生对应于尝试的已触发事件、查阅关于已触发事件的安全规则以及如果必要的话采取关于尝试的矫正动作。

[0159] 在一个实施例中，可以仅在电子设备404的组件的固件中实现系统400的O/S下层安全代理，如下面以及在图5的讨论中所描述的。在另一实施例中，可以在诸如主PC固件428之类的电子设备404自身的固件中实现系统400的O/S下层安全代理。在这样的实施例中，可以在电子设备404的主板上实现主PC固件428。在又一实施例中，也可以在O/S下层代理450中实现系统400的O/S下层安全代理。O/S下层代理450可以在低于电子设备404的诸如操作系统412之类的操作系统的级别以任何合适的方式实现以便提供对资源的访问的触发或应对这样的触发。例如，O/S下层代理450可以是图2的SVMM 216或SVMM安全代理217的实施例。O/S下层代理450可以包括安全规则422。

[0160] 电子设备404可以包括用于实施来自电子设备404的输入和输出操作的一个或多个组件。电子设备404可以包括任何合适的数量的这样的组件和任何合适的类型的组件。这样的组件可以由设备实现，且它们自己的处理器、存储器、和软件嵌入到固件中。这样的组件的示例实施例可以是图5的I/O设备502。

[0161] 电子设备404可以包括，例如，显示器424和存储426。每一个这样的组件424、426可以包括固件430、432。固件430、432均可以实现图5的固件504。如上所述，每一个这样的组件424、426可以包括基于固件的安全代理，例如固件安全代理440、442。固件安全代理440、442均可以部分地或完全地实现图5的固件安全代理516。在一个实施例中，固件安全代理440、442中的每一个可以在它们各自的固件430、432中实现。在另一实施例中，固件安全代理440、442中的每一个可以在它们各自的组件424、426中的每一个中的固件430、432外实现。这样的设备固件安全代理440、442中的每一个可以通信上耦合到各自的一组安全规则434、436。每一这样的安全规则434、436可以实现图5的安全规则518。

[0162] 电子设备404可以包括固件。在一个实施例中，电子设备404可以包括主PC固件428。主PC固件428可以由基本输入/输出系统(“BIOS”)实现。在一个实施例中，主PC固件428可以被配置成计算机的BIOS。在这样的情况中，主PC固件428可以被配置为初始化计算机的处理器406的操作。主PC固件428可以被配置为允许主处理器406与诸如显示器424和存储426之类的I/O设备通信。在这样的实施例中，计算机也可以包含可编程I/O控制器，可编程I/O控制器可以由固件或BIOS编程，且与诸如424和存储426之类的I/O设备的固件通信。

[0163] 主PC固件428可以包括O/S下层安全代理。在一个实施例中，主PC固件428可以包括PC固件安全代理444。PC固件安全代理444可以被配置为截取对系统资源414的请求。为了完成这样的功能，PC固件安全代理444可以完全地或部分地实现图2的SVMM安全代理217或SVMM 216、和/或图5的固件安全代理516的功能。PC固件安全代理444可以实现图2的SVMM安全代理217或SVMM 216的功能以便完成对系统资源414的访问的O/S下层触发和应对、O/S下层代理和诸如O/S内部安全代理418之类的O/S内部安全代理的验证和确证和诸如安全规则420、422之类的安全规则的分发。PC固件安全代理444可以实现图5的固件安全代理516的功能以便完成固件中的O/S下层触发和应对、安全规则的更新并且评估被发送给电子设备404的各部分的IN和OUT命令。

[0164] 电子设备404可以包括安全规则438。安全规则438可以是图1的安全规则114的示例实施例。在一个实施例中，安全规则438可以驻留在主PC固件428中。在另一实施例中，安全规则438可以驻留在主PC固件428外，且PC固件安全代理444可以被耦合到安全规则438。

[0165] 系统400的安全代理可以被配置为一起工作以便防止恶意软件及其恶意操作。可以捕获对资源的尝试访问，且随后的事件被触发，以便在诸如显示器424或存储426之类的设备中或在主PC固件428中的固件安全代理中应对。这样的设备或固件中的固件安全代理可以被配置为应对已触发事件或把已触发事件传送到另一安全代理以供应对。由于有限的执行和更新能力，一些固件安全代理可以被限制为应对它们自己的已触发事件，且因而把这样的已触发事件传送给其他安全代理是有益的。固件安全代理可以向其传送事件的安全代理可以包括，例如，诸如O/S内部安全代理418之类的O/S内部安全代理、诸如O/S下层安全代理450之类的O/S下层安全代理或诸如PC固件安全代理444之类的另一固件安全代理。这些其他安全代理可以被配置为接收已触发事件、查阅安全规则、上下文信息或权限，并把要实现的所得到的动作发送回来。

[0166] 因此，尽管图4阐释用于基于固件的安全代理实施O/S下层触发和应对的示例数量的元素，但在各种实施例中可以使用更多或更少的元素。在使用更多或更少的元素时，每一元素和系统400的功能性可以据此改变。在一个实施例中，低于操作系统412的级别的系统

400的安全代理可以限于一个或多个O/S内部安全代理418和固件安全代理440、442。在这样的示例中，固件安全代理440、442可以依赖于保护服务器402以便更新到安全规则434、436。固件安全代理440、442可以依赖于O/S内部安全代理418以便更新或应对已触发事件，但是O/S内部安全代理418的操作是较不安全的，除非O/S下层安全代理确证O/S内部安全代理。固件安全代理440、442可以基于在安装、生产或配置建立的固件安全规则434提供触发。这样的安全规则可以是相对静态的。在这样的情况中，借助于少量分析，固件安全代理440、442可以被配置为提供相对基本的事件触发。这样的固件安全代理440、442却是有用的，这是由于在电子设备404的操作系统下面完成这样的触发，因而较好检测一些恶意的或可疑的操作。

[0167] 在另一实施例中，系统400的安全代理可以包括PC固件安全代理444或O/S下层代理450中的任一种，但不是两者。在这样的情况中，PC固件安全代理444的功能性可以由O/S下层代理450实现，且反之亦然。PC固件代理444或O/S下层代理450中的任一种可以被耦合到保护服务器402并被配置为获得诸如安全规则420、422、438、434、436之类的信息，并与系统400中的其他安全代理共享这样的信息。出于通信、更新或存储成本的目的，这样的安全规则可以适应每一各自的安全代理。PC固件代理444或O/S下层代理450中的任一种可以被配置为从诸如固件安全代理440、442之类的其他安全代理接收已触发事件、应用安全规则和其他信息，并采取矫正动作，例如把所得到的事件发送给固件安全代理440、442，或者把信息发送给保护服务器402。PC固件代理444或O/S下层代理450中的任一种可以被配置为捕获对系统资源414的尝试访问。PC固件代理444或O/S下层代理450中的任一种可以被配置为与O/S内部安全代理418通信以判断已触发事件的上下文。如果多于一个的O/S内部安全代理418出现在系统400中，则每一O/S内部安全代理418可以被配置为执行捕获、确证或与O/S内部安全代理418相关联的其他任务的经指派部分。这样的部分可以由操作系统下层安全代理定义。例如，一个O/S内部安全代理418可以确证或调查MOV指令，同时另一O/S内部安全代理418可以确证或调查JMP指令。

[0168] 在又一实施例中，系统400的安全代理可以包括PC固件安全代理444和O/S下层代理450两者。然而，在这样的实施例中，PC固件安全代理444的功能性中的一些或全部可以由O/S下层代理450实现，且反之亦然。在PC固件安全代理444和O/S下层代理450之间的任务划分可以考虑多种因素。例如，诸如PC固件安全代理444之类的在固件内的安全代理的操作另一O/S下层代理450的操作更加安全。然而，更新安全规则和O/S下层代理450的软件可以比在PC固件安全代理444中更加简单和快捷。

[0169] 在再一个实施例中，一个或多个固件安全代理440、442可以驻留在独立于PC固件安全代理444或操作系统下层代理422的系统400上。在这样的示例中，固件安全代理440、442可以确证操作系统内部安全代理418的实例。

[0170] 固件安全代理440、442、444中的每一个可以被配置为驻留在固件逻辑内，足以能够监视和控制固件逻辑以便发现外部通信。固件安全代理440、442、444因而可以被配置为捕获特定信息和/或与特定的其他实体通信特定信息。固件安全代理440、442、444可以被配置为确定所接收到的操作请求以及要发送或接收的数据。此外，固件安全代理440、442、444可以被配置为控制要发送或接收的数据，且可以被配置为引起对数据的附加操作，例如加密、压缩、在数据中嵌入水印或解码数据中的水印。与固件安全代理440、442、444通信的系

统400的其他安全代理可以被配置为在要由固件安全代理440、442、444捕获的数据中嵌入水印，或解码由固件安全代理440、442、444放置到数据中的水印。

[0171] 可以例如通过可编程的输入-输出中断或可编程的输入-输出寄存器来实施与固件安全代理440、442或PC固件安全代理444的通信。这样的中断或寄存器可以由固件安全代理440、442、444驻留在其中的固件或设备的生产商定义和提供。

[0172] 系统400的O/S下层安全代理中的一个或多个可以被配置为充当主安全代理以便协调电子设备404的基于固件的安全代理的反恶意软件活动。在一个实施例中，PC固件安全代理444可以被配置成主系统400的安全代理。在另一实施例中，O/S下层代理450可以被配置为充当主安全代理。安全代理可以被配置为应对来自固件安全代理440、442的已触发事件。主安全代理可以被配置为确证固件安全代理440、442诸如O/S内部安全代理418等的其他安全代理的操作以及。主安全代理可以被配置为向其他安全代理通知安全代理中的一个是否已经注意到可疑行为或已检测的恶意软件、系统400是否受到恶意软件攻击或者系统400的管理员是否已经改变影响安全的偏好或设置。主安全代理可以与系统400的其他安全代理共享关于攻击的信息。

[0173] 通过在低于系统400的操作系统的级别捕获对系统400的资源的访问和/或应对所得到的已触发事件，系统400可以提供针对恶意软件的增强安全。固件中的安全代理的操作可以减少恶意软件影响安全代理的操作的机会。在固件中或者在设备级别捕获操作可以减少恶意软件欺骗或钓鱼系统400的元素以便伪装其操作的能力。例如，无论恶意软件危害操作系统412的什么部分，不可以对设备自身隐瞒对组件424、426的请求。

[0174] 图5是用于保护电子设备免遭恶意软件的基于固件的解决方案的示例实施例的更详尽的视图。诸如I/O设备502之类的设备可以被配置为接收和捕获对使用或访问设备的资源的请求。在一个实施例中，I/O设备502可以被配置为处理这样的已捕获请求来判断该请求是否指示恶意软件的存在。在另一实施例中，I/O设备502可以被配置为把这样的已捕获请求作为已触发事件传送给I/O设备驻留其中的系统的另一部分。系统的这样的另一部分可以包括O/S下层安全代理。I/O设备502可以包括被耦合到存储器508的固件504和处理器506，其中，固件504可以包括驻留在存储器508以供由处理器506执行的指令。

[0175] I/O设备502可以包括电子设备的用于控制对电子设备的资源的访问的任何合适的部分。在一个实施例中，I/O设备502可以实现电子设备的外围设备的一些或全部。I/O设备502可以例如由显示控制卡、计算机总线控制器、高速缓存设备、I/O控制器设备、盘控制器、存储器设备、网络控制器、主板、或键盘控制器实现。I/O设备502可以驻留在电子设备中。在一个实施例中，I/O设备502可以被耦合到物理组件。仅仅作为示例，这样的物理组件可以包括显示器、计算机总线、存储器、I/O控制器、盘、网卡或键盘。在另一实施例中，I/O设备502可以与所耦合的物理组件分离地驻留。例如，键盘控制器可以通过串行接口与键盘耦合。在这样的实施例中，I/O设备502可以驻留在电子设备中，同时这样的物理组件可以通信上耦合到电子设备但驻留在电子设备外。

[0176] 固件504可以被配置为控制I/O设备502的操作。固件504可以包括被配置为捕获对资源的请求的O/S下层安全代理516，在低于I/O设备502或I/O设备502驻留在其中的系统中的操作系统的级别下操作。O/S下层安全代理516可以被配置为应对从已捕获请求得到的事件以判断是否允许、拒绝或以另外方式应对请求，以便保护I/O设备502或I/O设备502驻留

在其中的系统免遭恶意软件。在一个实施例中，固件504可以包括固件安全代理516。固件安全代理516可以合并图2的SVMM 216或SVMM安全代理217的功能性中的一些或全部，但是在固件504中实现。在这样的情况中，SVMM 216或SVMM安全代理217的诸如捕获对资源的访问和/或应对已捕获请求等的功能性可以由固件安全代理516实施。在一个实施例中，固件安全代理516可以被配置为驻留在固件504中。

[0177] 固件504可以包括I/O命令510、数据传送引擎12和编程逻辑514。I/O命令510可以包括用于向设备发送信息或从其接收信息的指令。这样的命令可以包括IN或OUT命令的变体。I/O命令510的执行可以操作为执行所期望的设备动作。由设备接收到的请求可以被转换成I/O命令。根据对资源的特定请求的捕获或触发可以通过根据关联的I/O命令510的捕获或触发来完成。数据传送引擎512可以被配置为应对把请求传输到设备和随后的响应。数据传送引擎512可以被耦合到处理器506和在I/O总线上的可编程I/O控制器，在I/O总线上交换I/O命令510和数据。可编程逻辑514可以被配置为提供指令以供固件504操作I/O命令510和数据传送引擎512。编程逻辑514可以被加载到诸如处理器506之类的处理器中。

[0178] 固件安全代理516可以被配置为修改编程逻辑514的操作以便检测已尝试恶意操作。固件安全代理516也可以被配置为监视把请求传输到设备以便通过数据传送引擎512截取对I/O设备502的请求并判断这样的请求是否恶意的。固件安全代理516可以包括控制结构，在控制结构中，可以把标志设置为对应于要捕获的操作。在一个实施例中，可以在结构中根据要捕获的命令的存储器地址设置标志。固件安全代理516可以被配置为设置用于截取对I/O设备502的请求的标志。这样的标志可以对应于，例如，I/O命令510的特定命令或这样的特定命令与特定参数的组合。这样的标志可以被配置为截取特定的请求或请求类别。一旦触发对应于已捕获的I/O命令510的已尝试操作的特定标志，固件安全代理516可以被配置为处理事件并采取所得到的动作，通过数据传送引擎512把所得到的信息传送给另一安全代理，或通过数据传送引擎512传送已触发事件。

[0179] I/O设备502也可以包括安全规则518。安全规则518可以实现图2的安全规则222中的一些或全部。安全规则518可以在存储器508中实现。在一个实施例中，安全规则518可以驻留在固件504外。在另一实施例中，安全规则518可以驻留在固件504中。固件安全代理516可以通信上耦合到安全规则518并被配置为访问安全规则518，以判断在固件504中设置什么标志以便捕获对I/O设备502做出以便访问其资源的特定请求或请求类别。例如，固件安全代理516可以被配置为访问安全规则518以便判断已触发事件是恶意的还是非恶意的。在一个实施例中，安全规则518可以包含供固件安全代理516处理已触发事件的指令。固件安全代理可以被配置为使用这样的指令来判断是否允许或拒绝请求，或者采取另一矫正动作。在另一实施例中，固件安全代理516可以被配置为使用这样的指令来判断是否向另一安全代理报告该请求。这样的矫正动作也可以包括等待来自其他安全代理的可以包含关于是否允许或拒绝请求的指令的响应。

[0180] 在一些实施例中，固件安全代理516可以驻留在固件504中，这可以使得相对难以更新固件安全代理516。另外，恶意软件攻击的不断改变的本质要求反恶意软件解决方案是灵活的。因此，固件安全代理516可以使用用于接收信息的任何合适的机制，以便确定捕获对I/O设备的什么请求，以及采取什么随后动作。

[0181] 在一个这样的实施例中，这样的机制可以包括如上所述的访问安全规则518。固件

安全代理516可以被配置为从其他安全代理或保护服务器接收新的和经更新的安全规则518。为了实现灵活性，固件安全代理516可以被配置为把安全规则518存储在与固件分离的存储器508中，例如如果把这样的规则存储在固件504中将使得难以更新安全规则518。

[0182] 在另一个这样的实施例中，固件安全代理516可以被配置为依据固件的更新或闪现(flash)来更新安全规则518。在这样的实施例中，更新要捕获的请求的灵活性可能受到限制。因此，安全规则518可以针对非常特定的受保护资源。例如，盘设备的安全规则518可以包括捕获对设备的引导扇区的所有写请求的指令。在一些情况中，在与其他安全代理的通信廉价的场合，安全规则518可以包括捕获各种各样的请求的指令，其中可以把处理大部分卸载给其他安全代理。

[0183] 在又一这样的实施例中，固件安全代理516可以被配置为从其他安全代理接收指令。在一种情况中，这样的指令可以采取固件504或固件安全代理516的函数调用的参数的形式。例如，另一安全代理可以调用固件安全代理516的被命名为“UpdateRule(trigger,action)(更新规则(触发,动作))”的函数，其中，在trigger(触发)中详述要捕获的请求，且在action中详述要采取的随后动作。固件安全代理516因而可以通过接收关于对安全规则的更新的指令来更新安全规则518。在另一情况中，另一安全代理可以把安全规则518的更新写到设备502的已保留存储器空间，该存储器空间随后可以被固件安全代理516访问。从其他安全代理接收到的指令也可以指示固件安全代理516使用一组特定的安全规则518。例如，在时序要求严格(time-critical)的操作期间，固件安全代理516可以被这样的指令配置为使用最小的一组核心安全规则518。如果I/O设备502是盘设备，这样的最小的一组核心规则可以包括捕获对盘的引导扇区的访问的指令。在另一示例中，如果目前不实施时序要求严格的操作，则固件安全代理516可以被这样的指令配置为采用来自安全规则518的规则以便捕获广泛得多的访问尝试并把相应的事件发送给其他安全代理以供应对。

[0184] 固件安全代理516可以被配置为控制I/O命令510、扫描接收到的或要发送的内容或数据并对命令和内容应用访问控制。固件安全代理516可以被实现为现有的设备固件的扩展。

[0185] 固件安全代理516的实现可以取决于设备502的类型。例如，显示设备和盘设备可以因不同种类的内容或意图的命令而触发。各种设备中的固件安全代理516的创建可以适应与设备的特定种类的接口。例如，如果设备502被配置为通过串行高级技术附件(“SATA”)总线通信，则类似于通过SATA总线通信的其他设备，它可以配备有固件安全代理516。固件安全代理516可以被定制为支持设备502的体系结构、支持设备502的外部总线I/O或设备502的其他接口。

[0186] 固件安全代理516可以被配置为通过截取特定的读命令和写命令来捕获对设备502中的资源的尝试访问，这可构成对资源的请求的一部分。可以截取、评估和基于诸如安全规则518中的一个之类的规则阻止或允许读或写命令。用于固件安全代理516的安全规则518可以包括用于检测恶意软件的迹象的任何合适的规则。这样的读命令和写命令可以例如是对驱动程序的函数调用或中断的结果。

[0187] 例如，安全规则518可以包括供固件安全代理516扫描要被写到设备的数据的规则。可以评估数据的内容或数据的散列以便判断该数据是否对应于恶意软件数据或代码。这样的评估可以通过把内容和白名单或黑名单中的数据或签名来进行。接连的写入可能必

须被一起评估以适当地评估要写入的数据或内容的完整范围，以便正确地把内容或数据标识为恶意软件或不是恶意软件。例如，可以在对设备502的重复接连调用中写入文件。要写入的数据可以排队，以使得可以评估对写命令的内容的适当扫描。

[0188] 在另一示例中，安全规则518可以包括供固件安全代理516扫描设备中的现有数据的规则。设备502可以包含从系统外例如在网卡中接收到的内容。当驻留在设备502中时，可以扫描所接收的信息的内容以便发现恶意软件的迹象。固件安全代理516可以通过把内容与白名单或黑名单中的数据或签名进行比较来做出评估。

[0189] 在又一示例中，安全规则518可以包括供固件安全代理516基于时间或权限评估命令的规则。在不应实施合法活动的时间期间，可以保护诸如网络设备或盘之类的设备502免遭读取或写入。例如，某些恶意软件可以在引导期间攻击盘驱动程序。因而，固件安全代理516可以在引导盘的时间期间防止对设备的任何写入。类似地，可以由设备502驻留在其中的系统的管理员设置关于何时或如何使用设备或系统的权限。例如，设备502驻留在其中的系统的管理员可以把设备设置为在营业时间之外不可用。系统上的网络设备没有合法目的在营业时间之外传输活动，且因而基于安全规则518中的权限，对网络设备的读取和写入可受到固件安全代理516阻止。这样的使用可以阻止，例如，设备的实际用户的蓄意活动或恶意软件使用网络设备来实施拒绝服务攻击的蓄意活动。

[0190] 在再一个示例中，安全规则518可以包括供固件安全代理516基于与I/O命令一起使用的参数评估命令的规则。这样的参数可以包括，例如，写命令将写入的地址。安全规则518可以包括指示盘设备的特定部分是只读的规则。因而，固件安全代理516可以检查与把数据写到盘的OUT命令相关联的参数以判断数据将被写入的地址，并且如果所尝试的写是针对受到安全规则518中的规则的写保护的盘的部分则阻止该命令。固件安全代理516可以与诸如发起调用的内容或实体之类的其他基础结合考虑这样的参数。例如，扫描要写入的数据的内容是昂贵的，且因此安全规则518可以把固件安全代理516配置为仅当数据被写入到某些地址范围时扫描要写入的数据。在另一示例中，诸如安全规则518之类的安全规则可以仅允许某些调用实体向盘设备的某些部分写入或从其读取。因而，固件安全代理516可以捕获所尝试的写或读且在可以安全地判断调用实体的身份之前不允许该尝试。可以通过评估被用来调用设备函数的参数中的信息做出这样的判断，这是由于一些这样的函数可以标识调用的设备驱动程序或应用。在这样的情况中，固件安全代理516可以采取任何适当的步骤来判断调用的有效性。在一个实施例中，固件安全代理516可以查阅安全规则518中的白名单或黑名单以便判断调用实体是否被授权为做出这样的调用。在另一实施例中，固件安全代理516可以与包含设备502的系统中的其他安全代理通信以便判断调用应用或设备驱动程序是否有效。这样的其他安全代理已经确证调用应用或设备驱动程序的操作，或者可以与已经验证这样的操作的O/S内部安全代理通信。在又一示例中，对诸如设备502之类的设备的现有驱动程序调用可以不标识调用实体。因此，没有可用的参数。在这样的示例中，固件安全代理516可以被配置为传送已触发事件或以另外方式查阅系统中的其他安全代理，以判断产生已尝试访问的调用的上下文。这样的其他安全代理可以为调用提供合适的上下文，以判断是否经授权的实体做出该尝试。

[0191] 在进一步的示例中，安全规则518可以包括供固件安全代理516基于来自设备502驻留在其中的环境的信息来评估命令的规则。系统中的其他安全代理可能已经检测到难以

移除的恶意软件感染，或者可能要求来自管理员的直接干预以便清除。系统中的其他安全代理可以具有观察到的可疑行为，且还没有完全分析行为的本质。在这样的情况中，固件安全代理516可以从其他安全代理接收这样的现有威胁的通知。取决于感染的类型，安全规则518因而可以规定固件安全代理516的预防性动作。例如，键盘设备中的固件安全代理516可以接收到已经检测到已知用于键盘记录的特定类型的恶意软件的迹象但还不能移除的通知。安全规则518因而可以规定固件安全代理516不允许来自键盘设备的所有读和写以便防止利用键盘传输的信息的危害。

[0192] 固件安全代理516可以以不同的方式保护不同类型的设备的I/O。例如，取决于恶意软件威胁，显示设备的固件安全代理516可以关闭显示器的各部分。固件安全代理516可以阻止引起在屏幕上产生水印的某些图案的显示。固件安全代理516可以捕获特定图案的尝试显示。固件安全代理516可以截取从设备尝试读取信息以便防止抓屏。

[0193] 在另一示例中，与系统的剩余部分通信，用于键盘设备的固件安全代理516可以可选地编码或解码其结果。这样的加密可以由固件安全代理516在出现了诸如键盘记录器之类的恶意软件威胁的通知时设定。

[0194] 在又一示例中，用于网络设备的固件安全代理516可以基于源因特网协议(“IP”)地址、源端口号、要发送或接收的数据、目的地IP地址或目的地端口号来捕获。一旦捕获到使用网络设备的这样的尝试，固件安全代理516可以扫描要发送或接收的分组的数据净荷以便发现恶意软件的迹象。在一个实施例中，这样的数据净荷可以被发送给另一安全代理或保护服务器，其中，可以扫描内容以便发现恶意软件的迹象。数据净荷的内容可以经过加密，以使得分组嗅探器不能成功地截取内容。可以捕获到由于和不安全的网络目的地的通信相关联的安全风险引起的对网络设备的尝试操作，其中，与恶意目的地的网络通信可以危害设备502驻留在其中的系统的安全。由于诸如银行业务网站之类的特定数据集的敏感本质，可以捕获意图的操作。在这样的情况中，当从这样的网站接收数据时，可以由固件安全代理516在传送到另一安全代理或调用实体之前加密数据。这样的加密可以防止分组嗅探器或设备502的系统中的过滤器成功地截取信息。

[0195] 要截取的特定的I/O命令510可以取决于特定的设备和该设备的操作。因而，设备502的生产商可以决定如何配置用于特定设备502的固件安全代理516的操作。设备502的生产商可以决定向其他安全代理公开多少设备502的功能性。例如，设备502可以被配置为在把已触发事件交接给其他安全代理之前要求确证这样的安全代理。

[0196] 在操作中，一个或多个O/S下层安全代理可以是在系统400的固件或系统400的组件的固件中运行。固件安全代理440可以在显示器424中操作，固件安全代理442可以在存储426中操作，且PC固件安全代理444可以在主PC固件408中操作。O/S下层代理450和O/S内部代理412可以在系统400中操作。每一安全代理可以与系统400中的一个或多个其他安全代理通信。每一这样的安全代理可以在接受通信之前确证另一安全代理的实例。在确证安全代理之后，保护服务器402可以与安全代理中的一个或多个通信。

[0197] PC固件安全代理444或O/S下层代理可以被指派为主安全代理。主安全代理可以与保护服务器402通信以判断安全规则。主安全代理可以把安全规则本地存储到主安全代理。主安全代理可以把安全规则分发给每一安全代理，其中，可以把安全规则本地存储到安全代理。可以为设备的类型、构造或模型定制安全规则，以便减少一大组安全规则的代价。

[0198] 在接收诸如规则434之类的安全规则时，诸如显示器424之类的设备可以在设备固件430内的控制结构中设置对应于要捕获的设备的操作的标志。类似的任务可以由存储426执行。

[0199] 应用410或驱动程序411可以试图访问诸如显示器424或存储426之类的设备。应用或驱动程序411可以通过调用操作系统412的内核来做出这样的尝试，操作系统412的内核又可以调用操作系统设备驱动程序，操作系统设备驱动程序又可以向组件424、426发送请求。

[0200] 请求可以到达诸如存储426之类的设备。在设备上运行的固件安全代理442可以通过监视具有控制结构的存储426的数据传送引擎412过滤这样的请求。请求可以采取对存储426可用的I/O命令510的形式。如果请求匹配已经由固件安全代理442设置的任何标志，则可以捕获该请求，且可以触发所得到的事件。固件安全代理442可以查阅安全规则436以判断如何应对已触发事件。

[0201] 在一个实施例中，已触发事件可以由固件安全代理442应对，且基于诸如关联的数据、命令、上下文信息、时间或环境信息之类的可用信息，可以采取矫正动作。这样的矫正动作可以包括允许或拒绝请求、移除恶意代码或数据或加密要传输的数据。其他矫正动作可以包括向保护服务器402发送关于已捕获事件的要传送的信息。固件安全代理442可以告知其他安全代理关于已捕获事件的状态，以使得其他这样的代理也可以在查询它们各自的安全规则之后采取矫正动作。例如，如果固件安全代理442检测到未知来源的恶意软件攻击，则固件安全代理440可以封锁对显示器424的附加访问。

[0202] 在另一实施例中，已触发事件可以被传递给诸如O/S内部安全代理418、PC固件安全代理444或O/S下层代理450之类的另一安全代理以供应对。接收安全代理，例如PC固件安全代理444，可以通过查询安全规则438来应对已触发事件。基于诸如数据、命令、上下文信息、时间或环境信息之类的可用信息，PC固件安全代理444可以允许或拒绝由已触发事件表示的请求。PC固件安全代理444可以与O/S内部安全代理418通信以判断关于对资源的尝试访问的上下文信息。PC固件安全代理444可以与保护服务器402通信以便得到关于如何应对已触发事件的附加信息。PC固件安全代理444可以把用于所得到的动作的指令发送回到来源固件安全代理442。PC固件安全代理444可以把关于已触发事件信息发送给保护服务器402以便分析或记录。可以在未知已触发事件的恶意本质时实施这样的分析或记录。PC固件安全代理444可以通知系统400的安全代理已经检测到特定种类的恶意软件、已经检测到一种可疑活动或者系统400受到恶意软件攻击。

[0203] 在从PC固件安全代理444接收到信息时，固件安全代理440可以采取矫正动作。这样的动作可以包括允许或拒绝已尝试访问、加密要传输的数据或移除恶意代码或数据。

[0204] 图6是用于保护电子设备免遭恶意软件的基于固件的可配置保护的方法600的示例实施例。在步骤605，可以认证O/S下层安全代理、O/S内部安全代理、保护服务器和固件安全代理的身份和安全。可以通过通过任何合适的方法来完成这样的认证，包括通过定位和检验位于存储器中的每一个的映像、密码散列或密钥。直到步骤605完成，可以停止其他步骤的操作。

[0205] 在步骤610，可以访问保护服务器以判断安全规则。这样的安全规则可以被用来在下列的步骤中做出判定。在步骤615，可以指示固件安全代理捕获对系统资源的访问。这样

的访问可以源自在电子设备上运行的应用、驱动程序或操作系统。可以指示固件安全代理要监视电子设备的什么系统资源。也可以指示固件安全代理要捕获所监视的系统资源上的什么操作。例如，可以标识要捕获对固件安全代理在其上运行的设备的读命令和写命令。在步骤620，可以在控制结构中设置对应于要捕获的这样的操作的标志。这样的已捕获操作可以产生已触发事件。

[0206] 电子设备在步骤630-675可以操作并通过一次或多次捕获对系统资源的访问受到保护，或在步骤680-685扫描数据以便发现恶意软件的存在。每一次捕获对系统资源的访问和扫描数据以便发现恶意软件的存在可以被并行实施。进一步，根据保护电子设备的操作的需要，这些中的每一个都可以重复。

[0207] 在步骤630，可以捕获对诸如系统存储器、寄存器或I/O设备之类的系统资源的访问。可以在低于在电子设备上运行的操作系统的级别实施这样的捕获。可以在固件内实施这样的捕获。在步骤632，可以产生与已捕获的尝试相关联的所得到的已触发事件以及任何关联信息。在步骤635，可以判断已触发事件目前是否应当被应对或者被传送给另一安全代理以供应对。可以通过访问一个或多个安全规则来做出这样的判断。如果已触发事件目前应当被应对，那么，在步骤640可以访问安全规则以基于已捕获事件和诸如关联数据、命令、上下文信息、时间或环境信息之类的其他信息判断采取什么动作。例如，可以扫描要写或读的数据以便发现敏感内容或恶意内容；可以标识调用实体以便查看实体是否具有权限；可以检查被用来调用命令的参数；或者可以引用来自其他安全代理的关于系统中的恶意软件的警报。

[0208] 在步骤642，可以判断已尝试访问是可疑的还是不可疑的。如果访问安全规则和与已尝试访问相关联的信息的组合产生已尝试访问是不可疑的判断，那么，在步骤645可以允许尝试。如果判断这样的尝试是可疑的，那么，在步骤647可以采取矫正动作。这样的矫正动作可以包括从数据移除恶意内容、告知保护服务器或其他安全代理关于恶意尝试的存在、不允许已尝试访问或加密要传输的数据。如果尝试是不可疑的，那么，在步骤650可以允许已触发事件。

[0209] 在步骤655，如果判断另一安全代理要应对已触发事件，则把已触发事件传送给另一安全代理以供应对。在步骤670，可以接收到来自安全代理的指示应采取适当动作的响应。在步骤675，可以采取这样的动作，例如矫正动作或允许已触发事件的操作。

[0210] 在步骤680，可以扫描设备存储器以便发现恶意软件的存在。这样的存储器可以包含从诸如另一网卡或先前执行的文件读取的结果之类的另一实体到达的内容。如果已知存储器的内容是是恶意的、可疑的或未知的，那么，在步骤685，可以移除存储器的内容。

[0211] 在步骤690，如果拒绝了已尝试访问，或如果找到了可疑的内容，那么，可以把这样的事件报告给另一安全代理或保护服务器。这样的报告可以包括关于任何关联的恶意软件或可疑行为的信息。

[0212] 根据保护电子设备的需要，可以连续地、周期性地或根据需求重复方法600的各步骤。

[0213] 图7是用于保护电子设备204免遭恶意软件的基于微代码的系统700的示例实施例。系统700可以是系统100的示例实施例，以微代码实现系统100的某些元素。可以在电子设备701的操作系统下面实施系统700的捕获操作。系统700可以包括被配置为捕获对电子

设备204的资源的访问的已尝试使用、产生对应于尝试的已触发事件、查阅关于已触发事件的安全规则并且如果必要的话采取关于尝试的矫正动作的一个或多个O/S下层安全代理。这样的O/S下层安全代理可以被配置为截取从电子设备701的资源产生的信息、产生对应于该产生的已触发事件、查阅关于已触发事件的安全规则并且如果必要的话采取关于尝试的矫正动作。可以在系统700的处理器中完全地或部分地实现这样的O/S下层安全代理中的一个或多个。可以在这样的处理器的微代码(“μC”)中完全地或部分地实现O/S下层安全代理。可以受到系统700保护的电子设备701的系统资源724可以包括，例如，类似于图2的系统资源224的资源、物理存储器714、处理器标志716、异常718、寄存器720或中断722。

[0214] 系统700可以包括基于微代码的O/S下层安全代理，例如微代码安全代理708。微代码安全代理708可以驻留在诸如处理器704之类的处理器的微代码708内。在一个实施例中，微代码安全代理708可以被配置为捕获由诸如应用710、驱动程序711或操作系统713之类的系统700的各部分做出的对系统资源724的尝试访问。微代码安全代理708可以被配置为基于这样的对系统资源724的尝试访问创建已触发事件。例如，操作系统713可以通过尝试执行物理存储器714中的地址中的代码片段来尝试起动程序。在另一示例中，操作系统713可以尝试读或写物理存储器714中的地址。尽管示出了物理存储器714，但微代码安全代理可以被配置为捕获对访问虚拟存储器的尝试。在另一实施例中，微代码安全代理708可以被配置为捕获来自处理器702的诸如微代码模块710之类的其他部分的尝试传输信息。微代码模块710可以包括被配置为实施处理器702的操作以便执行指令的处理器702的其他部分。这样的尝试传输信息可以包括来自系统资源724的操作的结果。例如，在代码处理期间，除以零的操作可以由微代码模块710截取且可以尝试产生和传输异常718。

[0215] 微代码706可以包括硬件级别的指令用于执行从诸如操作系统713之类的系统700的元素接收到的更高级别的指令。微代码706可以把这样的更高级别的指令转换成电路级别的指令以便由处理器702执行。微代码706可专用于电子电路或由处理器702实现的处理器的类型。在创建处理器702时，微代码706可以配置有微代码706的特定内容。更新或重新编程处理器702上的微代码706的能力可能受到限制。微代码706可以驻留在内部处理器存储器704中。内部处理器存储器704可以是与诸如存储器703之类的系统700的系统存储器分离的高速存储器。在一个实施例中，内部处理器存储器704可以是只读存储器。在另一实施例中，微代码706可以驻留在内部处理器存储器704中包含的可编程逻辑阵列中。在又一实施例中，内部处理器存储器704可以包括或被实现为存储器存储或控制存储。在这样的实施例中，内部处理器存储器704可以部分或完全由静态随机存取存储器或闪速存储器实现。在这样的实施例中，微代码706可以被配置为作为处理器702的初始化的一部分从诸如存储器703之类的某些其他存储介质加载到存储器存储，且可以被配置为通过写到存储器存储的数据被更新、被重新安装或接收诸如安全规则或机器指令之类的新信息。

[0216] 微代码安全代理708可以被配置为访问安全规则707以判断要捕获什么操作、命令、通信或其他动作。安全规则707可以驻留在微代码706，或处理器702或系统700的另一合适的部分内。安全规则707可以由来自诸如做出对微代码安全代理708的调用并通过参数传送信息的其他安全代理之类的在处理器702外的实体的函数调用实现。微代码安全代理708可以通信上耦合到安全规则707。在一个示例中，安全规则707可以具有这样的逻辑：

[0217] -如果地址(x)由虚拟存储器范围(X1-->X2)或物理存储器范围(Y1-->Y2)中的代

码执行，那么向O/S下层代理产生已触发事件以供应对；

[0218] -如果地址(x)由物理存储器范围(Z1-->Z2)中的代码执行，那么，跳过指令；

[0219] -如果A、B和C；那么，存储器范围(Y1-->Y2)可以访问存储器范围(X1-->X2)；以及

[0220] -仅来自存储器范围(Y1->Y2)和(T1->T2)的代码可以写到(Z1-->Z2)。

[0221] 微代码706可以包括理解已经接收到的指令的上下文的状态机。执行某些安全规则707(例如在相互的上下文内评估接连操作的安全规则)需要这样的信息。这样的信息可以随已触发事件一起传送。

[0222] 也可以在O/S下层代理712中实现系统700的O/S下层安全代理中的一个或多个。O/S下层代理712可以以任何合适的方式实现以便在低于诸如操作系统713之类的电子设备701的操作系统的级别提供对资源的访问的触发或这样的触发的应对。O/S下层代理712可以实现图2的SVMM 216或SVMM安全代理217；图4的固件安全代理440、442或PC固件安全代理444；或图5的固件安全代理516的功能性中的一些或全部。O/S下层代理712可以通信上耦合到安全规则723。

[0223] 在一个实施例中，系统700的O/S下层安全代理中的一个或多个，例如O/S下层代理712，可以被配置为应对由诸如微代码安全代理708之类的基于微代码的安全代理产生的已触发事件。O/S下层代理712可以被配置为也以与图1-2和4-5中的O/S下层代理相似的方式捕获对资源的访问或应对已触发事件。O/S下层代理712和微代码安全代理708可以通信上耦合。微代码安全代理708可以被配置为把已触发事件发送给O/S下层代理712。O/S下层代理712可以通信上耦合到诸如O/S内部安全代理719之类的其他安全代理，且可以通信上耦合到保护服务器202。O/S下层代理712可以被配置为从诸如O/S内部安全代理719之类的其他安全代理接收上下文信息。这样的信息可以提供关于产生对系统资源724的尝试访问的实体的信息。如果多于一个的O/S内部安全代理719出现在系统700中，则每一O/S内部安全代理719可以被配置为执行捕获、确证或与O/S内部安全代理719相关联的其他任务的已指派部分。这样的部分可以由操作系统下层安全代理定义。例如，一个O/S内部安全代理719可以确证或调查MOV指令，同时另一O/S内部安全代理719可以确证或调查JMP指令。

[0224] O/S下层代理712也可以被配置为从保护服务器202接收安全规则或准实时信息。此外，O/S下层代理712可以被配置为查阅诸如安全规则723之类的安全规则、任何从诸如O/S内部安全代理719之类的其他安全代理或保护服务器202接收的上下文信息，以便判断如何应对从微代码安全代理708接收的已触发事件。

[0225] 在特定的实施例中，O/S下层代理712可以包含理解系统700中遇到的操作的上下文的行为状态机。然后，O/S下层代理712可以被配置为基于上下文判断要由微代码安全代理708执行的适当的动作。这样的动作可以包括在安全规则的要求的推动下的矫正动作、允许操作、拒绝操作或采取其他步骤。微代码安全代理708可以被配置为采取从O/S下层代理712接收的这样的动作。

[0226] O/S下层代理712可以是也被配置为确定由诸如O/S内部安全代理719之类的另一安全代理执行的适当的动作。例如，如果来自微代码安全代理708的已触发事件指示特定的种类的恶意软件威胁，或对电子设备701的内核或用户模式的特定部分的威胁，则O/S下层代理712可以被配置为指示O/S内部安全代理719采取矫正动作。因而，O/S下层代理712可以控制O/S内部安全代理719。

[0227] O/S下层代理712可以被配置为确证微代码安全代理708的实例，且反之亦然。O/S下层代理712可以被配置为与微代码安全代理708通信以便共享或设置要在安全规则707中实现的诸如来自安全规则723的那些之类的安全规则，关于系统700、管理员或环境设置和偏好的状态信息，或供微代码安全代理708捕获操作、产生触发以及应对这样的触发或把它们发送给其他安全代理的其他合适的信息。

[0228] O/S下层代理712可以被配置为通过任何合适的机制把这样的信息传输给微代码安全代理708。O/S下层代理712可以调用处理器702、微代码706或微代码安全代理708的函数，并把信息作为参数传送给函数。这样的函数可以被专门创建为把这样的改变传送给微代码安全代理708。例如，为了禁止来自另一物理存储器范围“B”的存储器的任何实体操作对物理存储器范围“A”的访问，可以使用诸如“Bar_Memory(A,B)(禁止_存储器(A,B))”之类的函数。作为这一函数被调用的结果，微代码安全代理708可以被配置为设置在微代码706内的参数。调用这样的微代码指令可以享有特权，以使得微代码安全代理708可以被配置为在代表O/S下层代理712调用这样的微代码指令之前确证O/S下层代理712。在另一示例中，O/S下层代理712或微代码安全代理708可以通过把数据写到存储器存储、控制存储或处理器702或微代码706的其他可写入部分来传输这样的信息。

[0229] 处理器702可能具有有限资源供微代码安全代理708完全地实现所有必要的捕获和应对以便保护系统700免遭恶意软件。在一个实施例中，微代码安全代理708可以被配置为仅实现要由处理器702实施的动作的捕获，且可以把与这样的捕获相关联的触发卸载给系统700的其他安全代理或组件以供随后应对。微代码安全代理708可以采取随后的动作，例如允许或不允许请求或通信，或可以采取其他动作，例如报告信息。在另一实施例中，微代码安全代理708可以被配置为实现已触发事件的少部分的应对。用于这样的应对的合适的已触发事件可以包括不要求显著的上下文信息的那些。例如微代码安全代理708可以通过安全规则707接收特定的范围的存储器地址免遭所有读和写的信息，除非O/S下层代理712的实例已经得到确证。可以实现这样的安全规则，因为内容是非常敏感的，且没有O/S下层代理712的操作帮助，就不能识别访问存储器内容的实体的身份。因而，在确证O/S下层代理的实例和操作之后，微代码安全代理708可以设置指示这样的确证的位。如果对存储器的尝试访问被触发，且仍然未设置该位，那么，微代码安全代理708可以被配置为不允许该存储器范围的内容的读、写或执行。如果已经设置该位，那么，微代码安全代理708可以被配置为随后捕获对存储器范围的尝试访问，产生要发送给O/S下层代理712的已触发事件，O/S下层代理712将根据上下文信息和其他设置评估是否允许该调用实体访问存储器范围。然后，O/S下层代理712可以把也许指示是允许还是拒绝访问的所得到的动作发送回到微代码安全代理708。

[0230] 已触发事件可以包括可以用来标识已尝试动作的源、方法或目的地的任何合适的信息。已触发事件可以被微代码安全代理708或O/S下层安全代理712用来应用安全规则。已触发事件可以由微代码安全代理708产生。例如，已触发事件可以精确地详述访问什么资源、调用什么指令、使用什么指令操作数、尝试或指令来自什么存储器地址(即源存储器)、要把操作的结果存储到什么存储器中(即目标存储器)或什么存储器将受到影响，或引起已尝试动作的源、方法或目的地的标识的任何其他信息。微代码安全代理708也可以被配置为包括关于处理器702的信息，例如活动、睡眠、空闲、停机和重启的处理器状态；处理器间通

信;以及功率消耗。

[0231] 诸如O/S下层代理712之类的另一安全代理可以被配置为在已触发事件中把这样的信息用来在应用安全规则722时判断事件的范围。O/S下层代理712可以具有对附加线索的访问权,这些附加线索例如关于在操作系统713中操作的实体的信息、保护服务器202中的新信息、其他安全代理检测到的恶意软件或其他威胁、管理员设置等等。例如,假定已捕获请求起源于物理存储器中的特定地址,则O/S下层代理712可以确定与该特定地址相关联的线程、进程或应用。然后,O/S下层代理712可以被配置为判断这样的实体是否经过授权来采取所考虑的动作。O/S下层代理712可以被配置为确定实体的身份。O/S下层代理712可以被配置为把实体分类为已知是安全的(例如,通过查询白名单)、已知是恶意的(例如,通过观察行为或查询已知的恶意软件的黑名单)或未知。O/S下层代理712可以被配置为把关于未知实体和恶意实体的信息报告给保护服务器202。

[0232] 出于捕获目的,微代码安全代理708可以拥有对某些处理器702资源和其他安全代理不可用的其他系统资源724的访问权。在一个实施例中,在微代码706内微代码安全代理708的实现可以避免通过把这样的资源的公开限制为在处理器外的调用实体来创建的限制。例如,虚拟机监视器可以被限制为捕获对已经由处理器702出于虚拟化目的而公开的资源的操作。把捕获对存储器所尝试的读、写或执行的能力作为进一步的示例。基于虚拟机监视器安全代理可以仅拥有对可用于虚拟化的存储器的访问权,并且,因而,可以仅可以跟踪对存储器页面的意图的读、写或执行尝试。相反,微代码安全代理708可以截取和应对对特定的物理存储器地址的读、写或执行请求,并基于安全规则707评估该请求。在提供系统700中的安全解决方案时,较小的粒度可以提供较大的灵活性。对在带有特定的物理存储器地址的上下文中使用什么指令的指令级别的知晓告知系统700,哪一实体调用什么资源,且不仅仅是访问了存储器页面。这灵活性可以是非常有价值的。例如,微代码安全代理708可以监视两个邻近的存储器地址以便发现读、写或执行尝试,但基于访问了两个存储器地址中的哪一个可以由安全规则707指示为采取完全地不同的动作。由于仅对对其作出尝试的存储器页面的观察,可能无法应用规则中的这样的区分。在另一示例中,管理程序用于监视和设置调试寄存器的其他方法不具有被用来访问调试寄存器的指令的上下文,系统700也是如此。另外,用于设置或观察这样的调试寄存器的一些其他实体不在低于操作系统的级别运行,这使得它们更可能是恶意软件。最终,用于设置或观察这样的调试寄存器一些其他实体并不针对安全,且不能够访问安全规则、评估访问和采取矫正动作。

[0233] 要由微代码安全代理708采取的矫正动作可以包括由安全规则707确定的或者从O/S下层代理712接收到的任何合适的动作。可以允许或拒绝的命令或指令。可以允许或抑制从微代码模块710产生的信息。可以修改任何这样的命令、指令或信息。

[0234] 微代码安全代理708可以被配置为捕获中断的产生。可以通过捕获例如“INT”指令的执行以及后面跟随读取与中断相关联的宿主信息已知的相关寄存器来捕获中断。例如,可以读取通用寄存器以便知晓中断的代码标识符以及被用来调用它的参数。例如,中断13可以是盘中断,且一组已知的寄存器可以把中断标识为读或写,以及数据的相关的扇区和位置。

[0235] 微代码安全代理708可以被配置为捕获被写到处理器702的输入和输出端口的值。微代码安全代理708可以被配置为捕获被写到处理器702的输入和输出设备的值。微代码安

全代理708可以被配置为捕获用于做出这样的写或读的指令。

[0236] 微代码安全代理708也可以被配置为捕获处理器702的算术逻辑单元(“ALU”)的特定操作。可以捕获对应于受保护的散列算法的步骤的处理器上的系列操作以判断对函数的未经授权的访问。一些算术操作被恶意软件用来伪装自身或使得自身变形。某些算术指令、按位指令或MOV指令都是可以引起存储器页面或地址范围的内容改变的指令。通过捕获这样的指令,可以记录对代码部分或数据部分的改变。如果随后的分析显示代码部分或数据部分被修改为自修改恶意软件的一部分,那么,已捕获和已记录的指令可以被用来跟踪恶意软件所使用的加密算法。例如,可以判断,恶意软件使用带有特定密钥的XOR函数来使得自己变形。这样的信息可以产生用于检测自修改恶意软件的较好安全规则。进一步,通过保持存储器修改的跟踪,可以通过反转指令的应用来是实现修复逻辑。

[0237] 另外,微代码安全代理708可以被配置为实施数字权限管理操作。例如,微代码安全代理708可以被配置为指示要求运行特定的程序的授权的接收安全规则707。该特定程序可以位于存储器中的特定地址。这样的授权可以采取微代码安全代理708从O/S下层安全代理712接收例如授权码、密钥或字节的形式。这样的授权可以通过微代码安全代理708捕获对存储器的已尝试访问或程序指令的加载来完成,并把已触发事件发送给O/S下层安全代理712,O/S下层安全代理712又可以拥有对授权码、密钥或字节的访问权。O/S下层安全代理712可以把决定返回给微代码安全代理712。因而,基于授权码可以允许或不允许程序的操作。

[0238] 此外,微代码安全代理708可以被配置为基于存储器散列或校验和停止存储器中的特定代码的执行。这样的散列或校验和可以由安全规则707指示为是恶意的。当从存储器加载代码时,微代码安全代理708可以实施内容的散列或校验和,把它与已知的恶意代码的那些进行比较,且然后,拒绝对加载的尝试并加载修复函数以便消除违规代码。

[0239] O/S下层代理712可以被配置为告知包括微代码安全代理706的系统700的其他安全代理,它已经判断系统700已经感染了恶意软件、遇到可疑行为或以另外方式被危害。在这样的情况中,微代码安全代理706可以被配置为禁用处理器702的各部分的操作。微代码安全代理706可以被配置为通过捕获和拒绝对特定的系统资源724的请求或来自微代码模块710的已产生通信来禁止这样的操作。可以禁用处理器702的各部分是因为它们是敏感的或者可能被恶意软件滥用。

[0240] 微代码安全代理706可以被配置为保护存储器地址或存储器地址的范围免遭尝试加载、读、写或执行尝试。这样的存储器可以包括敏感数据,或者可以是受限的、敏感的或受保护的函数的初始化点。在不存在访问软件是安全的或者是中性的验证的场合,微代码安全代理706可以防止访问这样的存储器。在这样的情况中,诸如O/S下层代理712之类的安全代理可以把特定的存储器地址标识为受保护,也许是因为这样的存储器地址可以对应于示例敏感信息或受保护例程。O/S下层代理712可以给微代码安全代理708发送诸如安全规则707之类的关于保护哪些地址的信息。微代码安全代理708可以捕获对这样的存储器地址的意图的加载、执行、读取或写,并把相应的已触发事件发送给O/S下层代理712。O/S下层代理712可以根据安全规则723、来自保护服务器202的信息、白名单或任何其他合适的信息源判断调用软件是安全的还是中性的。O/S下层代理712可以把要实现的动作返回给微代码安全代理708。微代码安全代理706可以被配置为保护虚拟存储器中的页面或范围和/或物理存

储器中的地址或范围。微代码安全代理706可以被配置为把虚拟存储器页面、位置、或地址转换成物理存储器位置或地址。因而，给定要捕获的虚拟存储器位置，或从其中发起尝试的虚拟存储器位置，微代码安全代理706可以被配置为确定相应的物理存储器位置，或反之亦然。

[0241] 此外，微代码安全代理708可以被配置为保护对敏感代码的访问。在一个实施例中，微代码安全代理708可以被配置为通过监视对特定地址的访问以上面所描述的方式保护对敏感代码的访问，其中，当代码被存储在存储器中时，该地址表示代码的开始。在另一实施例中，微代码安全代理708可以被配置为监视“JMP”或相似的分支指令的执行，分支指令将把处理器304的操作移动到敏感数据或代码的中间。在这样的情况中，微代码安全代理708可以被配置为捕获“JMP”指令的执行以及与敏感内容范围的组合。微代码安全代理708可以被配置为分析“JMP”指令源于何处。微代码安全代理708可以被配置为产生对应于已捕获的“JMP”的尝试执行的已触发事件，该已触发事件可由O/S下层代理712来应对。O/S下层代理712可以被配置为考虑“JMP”指令源于何处，且“JMP”指令所起源的这样的存储器是否得到授权来访问所考虑的存储器。

[0242] 微代码安全代理708自身或其中的捕获功能性也可以被配置为由系统700的其他部分允许或禁用。如果捕获和应对事件是昂贵的，因而可能地损害系统性能，则这样的能力是有用的。这样的允许和禁用可以是基于特别敏感的程序或数据的使用、恶意软件威胁的检测、管理偏好或任何其他合适的原因。在一个实施例中，微代码安全代理706可以被配置为从O/S下层代理712接收MSAOn信号、VMXOn信号或其他指令以便开始安全处理和捕获。微代码安全代理708可以接收MSAOff信号、“VMWrite VMXOff”信号或其他指令来停止安全处理和捕获。在开始或停止安全处理和捕获之前，微代码安全代理708可以确证做出请求的安全代理的身份和实例。

[0243] 此外，微代码安全代理708可以被配置为截取在处理器702和电子设备701的其他处理器之间的处理器间消息和命令。这样的处理器间命令可以由适当的微代码模块710接收或由访问特定的系统资源724的电子设备701的实体尝试。在一个实施例中，处理器间命令可以被访问处理器702的软件从操作系统713通过机器状态寄存器发送。恶意软件可能试图发送这样的消息，例如，以便关闭处理器或把它们置于睡眠模式。微代码安全代理708可以被配置为捕获对例如对应于处理器间命令的MSR寄存器的尝试写入。已捕获命令的已触发事件可以被发送到O/S下层代理712以供应对以便验证尝试的源。

[0244] 微代码安全代理708可以被配置为截取来自处理器的诸如软件中断722之类的消息的产生和通信。微代码安全代理708可以被配置为控制中断的执行以使得它们仅被经授权软件访问。例如，将不被允许没有已知身份(例如通过散列、驱动程序在存储器中的源等等确定)或具有恶意身份的驱动程序执行软件中断。微代码安全代理708可以捕获对中断的访问并把已触发事件传送给O/S下层代理712以供应对。

[0245] 在另一示例中，微代码安全代理708可以被配置为捕获处理器702产生异常718。异常可以包括例如除以零操作、页面故障和调试信号。对包含这些的存储器地址的读访问可以被微代码安全代理708捕获并由O/S下层代理712应对。

[0246] 微代码安全代理708可以被配置为保护处理器702的各种数据结构。例如，恶意软件可以攻击中断描述符表(“IDT”)。在一个实施例中，微代码安全代理708可以捕获对包含

IDT自身的存储器位置的写访问尝试。在另一实施例中，微代码安全代理708可以保护诸如“LOAD IDT(加载IDT)”和“STORE IDT(存储IDT)”之类的用于改变IDT的函数被存储在其中的存储器位置。在另一示例中，微代码安全代理708可以被配置为保护EFLABS或类似的数据结构，或与中断应对程序相关联的标志。恶意软件可以通过由未经授权的源更改这样的资源来尝试破坏中断应对程序的操作。

[0247] 尽管微代码安全代理708可能专用于特定类型的处理器的特定实例，这是由于不同的电路布置可能需要不同的微代码指令，但一组安全规则707可能对使用给定的指令集的所有处理器来说是有效的。这是可能的，因为微代码安全代理708可以捕获某些指令，这些指令在实现相同指令集的不同处理器之间将不改变，但是电路可以改变且可关联资源取决于电路。例如，主台式中央处理单元(“CPU”)和嵌入式系统CPU两者都可以是来自相同生产商的ISA处理器，且因而安全规则707可以至少部分地在两种类型的处理器之间共享。相反，图形处理器上的图形处理单元或具有不同的指令集的车载嵌入式处理器不可以共享安全规则707。

[0248] 在操作中，微代码安全代理708可以在电子设备701的处理器702中运行，且O/S下层代理712可以在低于电子设备104的操作系统的级别运行。微代码安全代理708和O/S下层代理712可以相互认证。微代码安全代理708可以发起对系统资源724的访问的捕获以及由微代码模块710产生的输出或通信。因此可以根据来自O/S下层代理712的需求、根据安全规则707或在处理器702启动时发起微代码安全代理708。O/S下层代理712可以因为在系统700中的事件、管理员或系统设置或因为已触发的安全规则723而把安全允许请求发送给微代码安全代理708。例如，因为要执行特定的程序、要访问敏感数据或在系统700中的其他地方已检测到恶意软件威胁，可以产生这样的请求。O/S内部安全代理719和/或O/S系统下层代理712可以向微代码安全代理708认证自身。为了认证自身，O/S内部安全代理719和/或O/S系统下层代理可以调用由处理器702提供的特权指令来发起认证进程。该调用可以引起微代码安全代理708借助于签名或散列测量和认证例如O/S内部安全代理719和/或O/S系统下层代理712。

[0249] 微代码安全代理708可以从O/S下层代理712接收安全规则707。可以通过函数调用或通过写入到诸如存储器存储之类的共享存储器来更新微代码安全代理708。微代码安全代理708可以基于安全规则707把标志应用到被配置为捕获特定的指令、这样的指令的操作数、目标地址、源地址或其任何组合的微代码706的控制结构。微代码安全代理708可以捕获诸如操作系统713、应用710、或驱动程序711之类的在处理器上运行的实体对系统资源的尝试访问。微代码安全代理708的操作对这样的实体来说可以是透明的。微代码安全代理708可以捕获诸如来自其他微代码模块710的实例的输出之类的信息的产生。这样的微代码模块710可以包括被配置为给处理器702执行各种任务的微代码的其他部分。例如，微代码模块710中的一些可以检测何时产生处理器异常或中断、如何路由输入和输出数据或执行操作。微代码安全代理708的操作对这样的模块来说可以是透明的。微代码安全代理708可以使用状态机来执行基于所观察到的先前事件预测的某种捕获。

[0250] 在捕获对资源的访问或信息的产生时，微代码安全代理708可以创建与捕获相关联的已触发事件。这样的已触发事件可以包含关于捕获的信息，包括诸如所捕获的指令、所使用的参数、发起的存储器位置和目标存储器位置之类的上下文信息。

[0251] 在一个实施例中，微代码安全代理708可以应对已触发事件。在另一实施例中，微代码安全代理708可以把已触发事件传送给0/S下层代理712或另一安全代理以供应对。微代码安全代理708可以查阅安全规则707以便判断是否以及如何应对已触发事件，或把已触发事件传送给0/S下层代理712。微代码安全代理708可以等待来自0/S下层代理712的应答，或者如果安全规则707不要求后续措施则可以允许已捕获动作。微代码安全代理708可以基于安全规则707采取矫正动作，例如允许或拒绝指令，或替换要执行的值或参数。

[0252] 0/S下层代理712可以从微代码安全代理708接收已触发事件。0/S下层代理712可以查阅诸如安全规则723之类的安全规则以基于已触发事件判断要采取的适当动作。0/S下层代理712可以使用来自微代码安全代理708的已触发事件信息、来自0/S内部安全代理719的上下文信息、来自保护服务器202的信息、来自其他安全代理的判定、管理员设置、时间或其他信息来判断应采取的适当的动作。0/S下层代理712可以把要采取的动作发送给0/S内部安全代理719和/或微代码安全代理708。0/S下层代理712可以把关于已触发事件的信息和所得到的动作发送给保护服务器202。

[0253] 微代码安全代理708可以从诸如0/S下层代理712之类的另一安全代理接收要采取的动作。微代码安全代理708可以执行所接收的动作，例如允许或拒绝指令，或替换要执行的值或参数。

[0254] 图8是用于基于微代码的个性化的和可配置的保护电子设备免遭恶意软件的方法800的示例实施例。在步骤805，可以确证微代码安全代理的实例。在步骤810，可以确证另一安全代理的实例。这样的安全代理可以包括0/S下层安全代理。在步骤815，可以获得、发送或接收用于在处理器内的微代码级别捕获的一个或多个安全规则。可以通过例如函数调用或通过把参数写到共享存储器空间来传输这样的安全规则。在步骤820，可以发起在微代码级别的资源安全捕获。在一个实施例中，这样的发起可以源自接收到开始安全捕获的信号。在这样的实施例中，可以接收到信号，这是因为已经检测到对系统的恶意攻击，或因为敏感数据可能出现在系统中。在另一实施例中，这样的发起可以起因于咨询安全规则。在又一实施例中，这样的发起可以起因于处理器的启动。

[0255] 在步骤825，可以在微代码中设置对应于要捕获的操作的标志。这样的标志可以对应于特定的指令、这样的指令的操作数、目标地址，源地址，或其任何组合。这样的标志可以由所接收到的安全规则定义。在步骤830，可以接收到要执行的指令并与捕获标志比较。在步骤835，可以接收到所产生的且要从微代码发送的信息并与捕获标志比较。可以通过状态机步骤830和835，其中，可以重复各步骤，且可以记录来自各步骤的多次迭代的结果并与标志或安全规则比较。

[0256] 在步骤840，可以判断是否已经捕获到指令或信息。如果没有捕获到任何事物，则该方法可以返回到在步骤830和835监视指令和所产生的信息。如果捕获到某种事件，那么，在步骤845可以创建与捕获相关联的已触发事件。这样的已触发事件可以包含关于捕获的信息，包括诸如所捕获的指令、所使用的参数、发起的存储器位置和目标存储器位置之类的上下文信息。

[0257] 在步骤850，可以判断是否在微代码内应对已触发事件或者在微代码外的安全代理是否应该应对已触发事件。如果要在微代码内应对已触发事件，那么，在步骤855可以采取已触发事件的适当动作。这样的动作可以通过查询安全规则来定义。这样的动作可以包

括允许要执行的指令或要发送的信息、拒绝指令或通信、替换存储器中的值或参数中的值或所要求的任何其他矫正动作。然后，在步骤830和835，方法800可以继续安全监视。

[0258] 如果要在微代码外应对已触发事件，那么，在步骤860可以把已触发事件发送给安全代理以供应对已触发事件。在步骤865，可以收集与已触发事件相关的附加信息。这样的信息可以包括设置、偏好、上下文信息或恶意软件状态。在步骤870可以使用这样的信息来把安全规则应用到已触发事件。这样的应用可以产生相对于已触发事件要采取的一连串动作。在步骤875，可以指定这样的一连串动作并将其传递给可以实现指定动作的各种安全代理。这样的动作可以包括矫正动作、允许操作或通信发生、向保护服务器报告事件或任何其他合适的结果。在步骤880，可以采取在步骤875指定的动作。然后，方法800可以在步骤830和835继续安全监视。

[0259] 图9是用于调节对电子设备901上的安全敏感的处理器资源的软件访问的系统900的示例实施例。系统900可以包括O/S下层捕获代理920和已触发事件应对程序922，已触发事件应对程序922被配置为在电子设备901上操作，以便检测来自在诸如操作系统913之类的电子设备901的操作系统中运行的基于软件实体的对访问处理器资源924的恶意尝试。此外，O/S下层捕获代理920和已触发事件应对程序922可以被配置为使用一个或多个安全规则908来判断捕获什么已尝试操作或信息的产生以及如何应对对应于已捕获操作或信息所创建的已触发事件。O/S下层捕获代理920和已触发事件应对程序922可以被配置为允许、拒绝或为已触发事件采取其他矫正动作。

[0260] 电子设备901可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701和/或其任何组合来实现，或者被配置为实现它们的功能性。电子设备901可以包括被耦合到存储器903的一个或多个处理器902。处理器902可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702或其任何组合实现，或者被配置为实现它们的功能性。存储器903可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备901可以包括操作系统913，操作系统913可以包括被耦合到一个或多个安全规则921的O/S内部安全代理919。操作系统913可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理919可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418和/或图7的O/S内部安全代理719或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0261] O/S下层捕获代理920可以由图1的O/S下层捕获代理104、图2的SVMM216、图4的固件安全代理440、442或PC固件安全代理444、图5的固件安全代理516、图7的微代码安全代理708和/或其任何组合实现，或者被配置为实现它们的功能性。已触发事件应对程序922可以由图1的已触发事件应对程序108、图2的SVMM安全代理217、图4的O/S下层代理450、图7的O/S下层代理712和/或其任何组合实现，或者被配置为实现它们的功能性。在各种实施例中，O/S下层捕获代理920的功能性中的一些可以由已触发事件应对程序922完成，或者已触发事件应对程序922的功能性中的一些可以由O/S下层捕获代理920完成。此外，可以在相同的软件模块中实现O/S下层捕获代理920和已触发事件应对程序922。

[0262] 安全规则908可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、

436、438、图5的安全规则518、图7的安全规则707、723和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则921可以由图2的安全规则220、图4的安全规则420、图7的安全规则721和/或其任何组合实现，或者被配置为实现它们的功能性。

[0263] O/S下层捕获代理920可以被配置为截取对诸如处理器资源924之类的任何合适的资源的访问或来自诸如处理器资源924之类的任何合适的资源的信息。例如，处理器资源924可以由图1的资源106、图2的系统资源214、图4的诸如显示器424和存储426之类的组件的部分或图7的系统资源实现，或者被配置为实现它们的功能性。处理器资源924可以包括诸如处理器902之类的处理器可用的资源，以便允许处理器加载和执行指令。这样的资源可以包括，例如，数据寄存器928、控制寄存器930、高速缓存934、处理器标志936、处理器核心938、处理器异常940或处理器中断942。对这样的资源的已尝试访问可以包括诸如带有操作数的汇编语言指令之类的指令。捕获对其可用的处理器资源924可以取决于由处理器902公开的资源。例如，如果在虚拟机监视器中实现O/S下层捕获代理920，则可用于供O/S下层捕获代理920捕获的处理器资源924可以限于由处理器902出于虚拟化的目的公开的处理器资源924。在这样的情况中，处理器902可以包括用于处理器资源924中的一些的虚拟化扩展。在另一示例中，如果在微代码安全代理中实现O/S下层捕获代理920，那么，处理器902已经使得处理器902的几乎所有资源都可用于捕获。

[0264] O/S下层捕获代理920可以包括处理器资源控制结构(“PRCS”)926。可以以记录、数据结构、表或任何其他合适的结构实现PRCS 926。PRCS 926可以包含指定应捕获处理器资源924的哪些指令、信息或已尝试访问的信息。O/S下层捕获代理920或已触发事件应对程序922可以被配置为在PRCS 926中设置对应于要捕获的敏感的操作、信息或资源的标志。O/S下层捕获代理920或已触发事件应对程序922可以被配置为根据被包含在安全规则908内的信息在PRCS 926中设置这样的标志。

[0265] 图10是PRCS 1000的示例实施例。PRCS 1000可以是图9的PRCS 926的示例实施例。PRCS 1000可以包括要捕获的各种处理器资源的条目1014的表。每一条目可以具有标识资源和可以产生已触发事件的条件的一个或多个字段1004、1006、1008、1010、1012。例如，PRCS 1000可以具有用于触发标志1002、资源的标识符1004、与资源相关联的类型1006、触发器类型1008、关于何时触发事件的何时触发条件1010和其中触发事件的执行阶段1012的字段。PRCS1000的实现可以取决于其资源被标识的处理器的本质，包括体系结构(例如工业标准体系结构“ISA”)或由处理器902公开的资源。

[0266] 触发标志1002可以包括关联条目1014的捕获和触发开启还是关闭的指示。这样的标志可以允许把捕获条件作为条目1014加载到PRCS 1000中但更保持蛰伏。因而，PRCS 1000可以加载有安全规则的实施例而不主动地强加它们。触发标志1002可以被配置为由诸如图9的O/S下层捕获代理920之类的实体设置。这样的操作可以允许使用PRCS 1000的反恶意软件系统相比于将要求每次允许或禁用对特定资源或条件的捕获时增殖和减少PRCS 1000的系统操作得快得多。开启和关闭条目1014的能力可以允许反恶意软件系统有选择地捕获某些操作。如果特定的捕获操作在时间或执行方面是昂贵的，则这样的选择性是有益的，且因而仅在检测到特定的条件时允许条目1014。例如，如果系统正常地多次写到特定的寄存器，可以关闭对该寄存器的访问的捕获，直到反恶意软件系统的另一部分检测到指示可能的恶意软件感染的可疑行为。在这样的情况中，对应于写寄存器的条目1014的触发标

志1002可以被设置为“ON”以便捕捉攻击资源的任何附加恶意尝试。

[0267] 资源标识符1004可以包括要捕获的处理器的特定资源的标识。例如，标识符1004可以显示，资源是寄存器，例如特定的数据寄存器、诸如EAX之类的地址寄存器、栈区寄存器、控制寄存器、矢量寄存器、诸如ESP之类的栈区指针、指令寄存器、程序计数器、指令寄存器、程序状态字、常量寄存器、浮点寄存器、或条件寄存器。作为其他示例，标识符1004可以标识，资源是指令，例如“JMP”、“JZ”(如果条件等于零则跳转)、“JNZ”(如果条件不于零则跳转)、“MOV”(移动值)或“SysEnter”(快速调用0环过程)。作为又一示例，标识符1004可以标识，资源是类似于以下的其他资源中的一种：诸如转换后备缓冲器之类的高速缓存；诸如时间戳计数器之类的计数器；诸如系统的处理器0、处理器1...处理器N之类的逻辑核心；或诸如“DIV/0”之类的处理器异常或诸如处理器间中断之类的中断或其他全局变量。资源标识符1004可以被转换成指令的地址、寄存器或由资源标识符1004表示的其他资源的表示。资源类型1006可以包括条目1014包括的资源的类或类型的标识。PRCS 1000的一些条目可以适用于特定类型的所有资源。

[0268] 触发器类型1008可以包括所得到的已触发事件的应对是同步还是异步的标识。同步触发可以引起已捕获资源的执行或通信停机，直到例如判断该尝试是否指示恶意软件。异步触发可以允许已捕获资源的执行或通信继续，同时例如记录触发器以供将来评估。在一个实施例中，异步触发的对资源的已尝试访问可以被用来构建较大系列动作的评估，且在可以做出判断之前这样的系列动作的适当评估可以要求多个数据点。例如，无论指令指针寄存器的特定读取自身可能不是恶意的，但是所返回的信息的随后使用可能是恶意的。因而，状态机可以被用来首先异步捕获指令指针寄存器的读取，但是然后同步地捕获其在另一指令中的使用。

[0269] 何时触发条件1010可以包括逻辑规则或条件，在这些规则和条件下将基于对资源的访问产生已触发事件。例如，在写入或读取资源时将为寄存器产生已触发事件。在执行诸如“JMP”之类的指令时将为指令产生已触发事件。在高速缓存失效时，可以为诸如转换后备缓冲器之类的高速缓存产生已触发事件。取决于处理器的状态，例如在核心是空闲的时，可以为处理器核心产生已触发事件。在设置或写入标志或异常时，可以触发处理器异常或处理器标志。何时触发条件1010可以包括组合逻辑条件，例如对单个资源的多个条件(例如值范围)、对多个资源的条件(因而在多个条目1014中连结)或两者的组合。

[0270] 何时触发条件1010可以包含根据要捕获的资源的类型的条件。例如，在它被写入、用特定的值写入或读取时，可以触发寄存器。在另一示例中，在它被写入、用特定的值写入或读取时，可以类似地触发高速缓存或指针。在又一示例中，在核心是空闲的时，可以触发处理核心。在再一个示例中，在发送中断之前(在对中断表的全局空间的尝试访问时)或在发送中断之后(在写中断表之后)，可以触发处理器间中断，例如被用来命令处理器核心停机、睡眠或激活的处理器间中断。

[0271] 触发器1012的执行阶段可以包括在指令执行的哪一阶段中将捕获已尝试访问且产生已触发事件的指示。触发器1012的执行阶段与何时触发条件1010组合用作对捕获给定资源的附加要求。为了捕获给定条目，在关联指令到达触发器1012的执行阶段中指定的执行阶段时，可以评估何时触发条件1010。触发器1012的执行阶段可以包括例如对应于处理器的五个指令执行阶段或步骤的条目。在一个实施例中，五个这样的指令执行阶段可以包

括1)提取指令,2)解码指令,3)执行,4)访问存储器位置以便得到结果,以及5)把返回值写回到存储器、寄存器或另一位置。在这样的实施例中,触发器1012的执行阶段可以包括在五阶段中的任何阶段之前或之后触发的能力。这提供总共六个不同的示例触发选项—在提取之前、在解码之后(且因而在执行之前)、在执行之后(且因而在访问存储器位置之前)、在访问存储器位置之后(且因而在写返回值之前)以及在写返回值之后。基于执行阶段捕获的能力可以提供其他反恶意软件系统不可获得的显著灵活性。例如,执行特定指令的结果预先未知,且因而反恶意软件系统可以把触发器1012的执行阶段的值设置为在访问存储器位置以便得到结果之后,但在把返回值写回到寄存器之前,如该指令所命令的。这可以允许反恶意软件系统评估操作结果而无需允许写入它。如果该结果指示恶意操作,那么,可以把哑元值而不是从第四执行阶段返回的值写回到寄存器。基于已尝试执行,可以把关于已尝试执行的信息提供给已触发事件的应对程序,以便帮助判断该尝试是否恶意的。

[0272] PRCS 1000的每一资源1004可以具有多个条目对应于对资源1004与另一1004的访问的组合。这样的访问组合可以包括要捕获的两个步骤或更多进程。例如,条目1014可以包括用于以下的分离的条目:a)对对应于中断描述符表("IDT")的存储器位置的访问与对控制寄存器的访问的组合,以及b)对对应于中断描述符表的存储器位置的访问与对通用寄存器的访问的组合。此外,在图9中,这样的分离的条目可以由系统900的分离部分应对。例如,特定的O/S内部捕获代理919可以应对收集已捕获IDT的上下文信息——通用寄存器访问,同时其他O/S内部捕获代理919可以应对收集已捕获IDT的上下文信息——控制寄存器访问。

[0273] 返回到图9,O/S下层捕获代理920可以被配置为在PRCS 926中设置标志或添加条目。O/S下层捕获代理920可以被配置为访问诸如安全规则908之类的一个或多个安全规则以判断这样的标志或条目。在一个实施例中,O/S下层捕获代理920可以被配置为从已触发事件应对程序922接收设置这样的标志或条目的指令,已触发事件应对程序922可以在查询安全规则908或保护服务器202之后调用O/S下层捕获代理920。可以由处理器902和/或O/S下层捕获代理920提供用于设置标志或向PRCS 926添加条目的一组特定的特权例程。

[0274] 如果电子设备901包括多于一个的处理器,则每一个这样的处理器可以具有相应的PRCS 926。在一个实施例中,系统900可以包括用于每一个这样的PRCS926的O/S下层捕获代理920。在另一实施例中,O/S下层捕获代理920可以被配置为捕获在每一个这样的PRCS 926中表示的资源。

[0275] 如果系统900支持虚拟化,那么,PRCS 926自身可以被虚拟化。虚拟化PRCS926的内容可以限于被相应的处理器902虚拟化的那些资源。这样的被虚拟化PRCS 926可以被包括在虚拟机监视器中。在这样的情况中,O/S下层捕获代理920或已触发事件应对程序922可以被配置为在这样的虚拟机监视器中控制PRCS 926。在另一实施例中,O/S下层捕获代理920可以被配置为捕获在每一个这样的PRCS 926中表示的资源。此外,以每PRCS或每虚拟化处理器为基础,可以在每一个这样的虚拟化PRCS 926中创建条目1014,且可以在每一个这样的虚拟化PRCS 926中设置触发标志1002。

[0276] O/S下层捕获代理920可以被配置为把从已捕获尝试或通信得到的已触发事件发送给已触发事件应对程序922。已触发事件应对程序922可以被配置为基于已触发事件的信息和一个或多个安全规则908执行任何合适的随后动作。例如,已触发事件应对程序922可

以被配置为允许已尝试指令的执行，但是要求在执行之后通知结果。在另一示例中，已触发事件应对程序922可以被配置为完全跳过命令的执行或通信。如果没有要求返回值，则可以应用这样的示例。在又一示例中，例如通过使用“JMP”指令来把执行发送给修复例程的地址，可以把执行传递到新的位置。

[0277] 在操作中，O/S下层捕获代理920和已触发事件应对程序922可以在电子设备901上操作。O/S下层捕获代理920可以在低于电子设备901的操作系统的级别操作。此外，已触发事件应对程序922也可以在低于电子设备901的操作系统的级别操作。已触发事件应对程序922可以查阅安全规则908或保护服务器202以判断在PRCS 926中设置什么标志1002或条目1014。已触发事件应对程序922可以向O/S下层捕获代理920指示在PRCS 926中设置什么标志1002或条目1014。取决于检测到的各种条件，例如正在使用的应用910、所检测的恶意软件的其他指示、先前已触发事件或电子设备901的管理员设置，O/S下层捕获代理920和已触发事件应对程序922可以在电子设备901操作期间动态地改变触发标志1002或在PRCS 926中添加新的条目1014。作为这样的动态改变的基础的信息可以来自，例如，O/S下层捕获代理920或O/S内部代理919。可以根据资源1004或资源类型1006标识PRCS 926中的条目1014。触发器类型1008可以被设置为把随后的已捕获事件配置成同步的或异步的。何时触发条件1010可以被设置为配置在什么环境下已截取的请求将产生已触发事件，如同触发器1012的执行阶段那样。

[0278] 取决于系统900遇到的各种条件，可以动态地允许或禁用PRCS 926中的条目。例如，O/S下层捕获代理920可以禁用昂贵的捕获操作，这是因为直到已触发事件应对程序922接收到电子设备901受到恶意软件攻击的指示的这样的时间之前，所捕获的已尝试访问频繁地发生，带有许多假阳性。然后，O/S下层捕获代理920可以允许捕获操作。在一个实施例中，在这样的条件下可以允许一个或多个处理器资源924上的广泛捕获以便防止未知的恶意软件动作进一步危害电子设备901。这样的广泛捕获可以扩展到本质上关闭处理器、虚拟化处理器、线程、进程或应用的整个执行环境。

[0279] 对处理器资源924的请求可以起因于处于系统900的操作系统的级别的实体，例如起因于应用910、驱动程序911或操作系统913。请求可以被传送给处理器资源924但被O/S下层捕获代理920截取。此外，通过各种处理器资源924可以从处理器产生信息或通信。信息或通信可以由O/S下层捕获代理920截取。

[0280] 如果信息或通信匹配PRCS 926中的条目1014的任何何时触发1010字段，则O/S下层捕获代理920可以使用PRCS 926来捕获对资源的访问，且随后，产生已触发事件。被设置为“ON”的触发标志1002已经允许的条目1014可以被匹配为已尝试访问或信息或通信。可以把要访问的资源与资源字段1004和/或资源类型字段1006进行比较。如果要访问的资源匹配这样的字段，那么，可以评估何时触发条件1010。如果何时触发条件1010匹配系统信息或关于该请求的信息，那么，PRCS 926可以产生已触发事件。触发器1012的执行阶段可以被用来判断在何时产生已触发事件。例如，可以在在指令提取之前、在指令提取之后、在执行之后、在访问存储器以供随后写入、或在访问诸如寄存器之类的另一资源以供写回之后创建已触发事件。此外，可以为已尝试通信或诸如处理器间中断之类的信息的产生而产生已触发事件，类似于在中断被发送或被写到中断表之前或之后的“Interrupt_Sleep(中断_睡眠)”。取决于触发器类型1008，所产生的已触发事件可以是同步的或异步的。如果产生了同

步的已触发事件，则O/S下层捕获代理920可以停止对资源的已尝试访问或通信的产生的执行，等待事件的应对。如果产生了异步的已触发事件，则O/S下层捕获代理920可以允许对资源的已尝试访问或通信的产生的执行。O/S下层捕获代理920可以把关于尝试的附加上下文信息添加到已触发事件，例如该尝试从其中起源的存储器地址、把结果写到何处或任何其他合适的信息。

[0281] 出于决定已触发事件是否可疑的目的，O/S下层捕获代理920可以包括与已触发事件相关的信息。例如，O/S下层捕获代理920可以确定诸如判断存储器的什么部分做出已尝试访问之类的信息。可以由已触发事件应对程序922把该存储器部分与在电子设备903上运行的已知的进程、应用或程序进行相关。如果已尝试访问起因于未知的或未经授权的进程、应用或程序，那么，该尝试可能是可疑的。已触发事件应对程序922可以使用来自O/S内部安全代理919的信息来判断这样的相关。在另一示例中，O/S下层捕获代理920可以提供关于先前已触发事件的信息，例如状态机中所记录的那些。与目前的已触发事件相关的这样的先前已触发事件可以提供关于该尝试是否可疑的上下文信息。

[0282] O/S下层捕获代理920可以把已触发事件传送给已触发事件应对程序922，已触发事件应对程序922可以通过根据安全规则908评估已触发事件中的信息和/或来自O/S内部代理919的上下文信息来应对该事件。可以判断所得到的适当的动作并将其发送回到O/S下层捕获代理920以便应用到已捕获尝试。这样的动作可以包括允许尝试、拒绝指令执行或代入不同的数据或指令以便规避恶意软件的操作。

[0283] O/S下层捕获代理920可以存储已触发事件以供在捕获将来的尝试访问时随后引用。例如，恶意操作可能要求要由处理器资源924执行多个指令。因而，这样的恶意行为的每一步骤可以被反映在PRCS 926中的分离的条目1014中。O/S下层捕获代理920可以捕获恶意操作的第一步骤，该第一步骤自身可以不是恶意的但仅在与随后的步骤组合时是恶意的。在这样的情况中，可以把这样的步骤的条目1014设置为异步触发，这是由于该条件仅仅被记录到状态机中以使得O/S下层捕获代理920或PRCS 926可以知晓先前所应对的尝试。恶意操作的第二步骤的捕获可以具有如同何时触发条件1010的对第一步骤的捕获。

[0284] 图11是用于调节对电子设备上的安全敏感的处理器资源的软件访问的方法1100的示例实施例。在步骤1105，可以访问安全规则以在步骤1110判断要保护什么处理器资源或处理器通信。在低于电子设备中的操作系统的级别操作的捕获代理可以判断要捕获什么资源和通信。这样的捕获代理可以在例如虚拟机监视器、固件或处理器的微代码中操作。

[0285] 在步骤1115，可以把对应于要捕获的资源或通信的条目写到处理器资源控制结构，处理器资源控制结构可以被配置为在指定条件下捕获对已指派资源或通信的操作、访问或其他使用。PRCS中的条目可以写上资源的标识、资源类型、将触发事件的条件、触发器将是异步还是同步以及已尝试访问或通信应在什么执行阶段(如果有的话)产生已触发事件。在步骤1120，PRCS中的条目也可以写上触发器或允许标志，其指示条目是否被激活以供是否捕获。如果不设置触发标志，那么，该条目可以蛰伏且不被用来捕获对资源的已尝试访问。

[0286] 在步骤1125，可以监视对资源的访问或通信的产生。这样的监视可以通过PRCS发送。电子设备中的实体可以试图尝试产生处理器通信或尝试访问处理器资源。对访问资源的这样的尝试可以起源于电子设备的操作系统的级别。如果指令、命令或对访问资源的其

他尝试匹配PRCS中的条目的资源标识符，其中该条目已经被激活，那么，可以捕获该尝试。类似地，如果产生了匹配PRCS中的条目的资源标识符的处理器通信，其中该条目已经被激活，那么，可以捕获该尝试。在一个实施例中，如果满足指定何时触发的附加准则，则可以捕获对访问资源的尝试或通信的产生。例如，在曾经写入控制寄存器时可以捕获对控制寄存器的尝试写入。在另一示例中，在控制寄存器写有特定的值时可以捕获对控制寄存器的尝试写入。

[0287] 在步骤1130，可以判断是否捕获了已尝试访问或通信。如果没有捕获到尝试，那么，在步骤1140可以判断是否需要调整PRCS中的条目。这样的调整可以包括允许或禁用这样的条目、添加新的条目或调整准则或条目的设置。然后，方法1100可以返回到步骤1125。这样的调整可以是基于例如电子设备中所检测到的新的恶意软件、时间的流逝、先前已捕获尝试或管理员的设置。

[0288] 在步骤1145，如果已经捕获到尝试，则可以判断所得到的已触发事件应该是同步的还是异步的。如果触发器类型是不同步的，那么，方法1100可以返回到步骤1125，与进行到步骤1150并行发生。如果触发器类型是同步的，那么，在步骤1150可以存储关于已捕获尝试的信息。这样的信息可以例如被状态机用于将来判断已捕获尝试是否应该产生已触发事件。在步骤1155，可以判断是否满足触发器的所有条件。这样的条件可以要求例如把某些值写入到资源，或请求起源于(或不起源于)存储器中的特定位置。此外，这样的条件可以要求先前捕获了其他尝试。可以访问关于这样的尝试的信息并将其存储在状态机中。如果没有满足触发的所有条件，那么，方法1100可以返回到步骤1125。

[0289] 如果满足了触发的所有条件，那么，在步骤1155可以判断在哪一特定执行阶段(如果有的话)应产生已触发事件。这样的阶段可以包括，例如，在提取尝试中的指令之前、在提取指令之后、在执行指令之后、在访问存储器以便读取结构之后或在写回值之后。此外，这样的阶段可以包括在执行处理器间中断之前或之后。一旦完成了所指派的执行阶段，则在步骤1165可以产生该尝试的已触发事件。在步骤1170，在已触发事件中可包括诸如该尝试的源地址或目的地地址之类的上下文信息或所涉及的资源，以便在步骤1175传递给应对程序。

[0290] 在步骤1180，可以查询安全规则，以便在步骤1185判断已触发事件是否可疑的、不为管理员设置所准许、或指示恶意软件。诸如已触发事件的上下文信息之类的上下文信息、电子设备的操作系统中的其他事件或管理员设置可以被用来评估安全规则对已触发事件的应用。如果已触发事件是不可疑的，那么，在步骤1187可以通知捕获代理，且方法1100可以返回到步骤1125。如果已触发事件是可疑的，那么，在步骤1190所得到的矫正动作可以被发送给捕获代理。这样的矫正动作可以取决于对访问资源或产生处理器通信的特定尝试。例如，恶意指令可以欺骗要读或写的值，或可以把跳转指令定向到修复例程。在步骤1195，可以应用矫正动作。方法1100可以返回到步骤1125。

[0291] 图12是使用电子设备1201上的操作系统下层捕获来调节软件访问以便保护存储器的系统1200的示例实施例。系统1200可以包括O/S下层安全代理1220，O/S下层安全代理1220被配置为在电子设备1201上操作以便检测对访问存储器的恶意尝试，该恶意尝试来自在电子设备1201的诸如操作系统1213之类的操作系统中运行的基于软件的实体。此外，O/S下层安全代理1220可以被配置为使用一个或多个安全规则1208和存储器映射1206来判断

捕获对存储器的什么已尝试访问以及如何应对对应于已捕获操作而创建的已触发事件。O/S下层安全代理1220可以被配置为允许已触发事件、拒绝已触发事件或为已触发事件采取其他矫正动作。

[0292] 电子设备1201可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901和/或其任何组合实现，或者配置为实现它们的功能性。电子设备1201可以包括被耦合到诸如物理存储器1203之类的存储器的一个或多个处理器1202。处理器1202可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器902或其任何组合实现，或者被配置为实现它们的功能性。物理存储器1203可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备1201可以包括操作系统1213，操作系统1213可以包括被耦合到一个或多个安全规则1221的O/S内部安全代理1219。操作系统1213可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理1219可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418、和/或图7的O/S内部安全代理719、图9的安全规则908或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0293] O/S下层安全代理1220可以由图1的O/S下层捕获代理104或已触发事件应对程序108、图2的SVMM 216或SVMM安全代理217、图4的固件安全代理440、442、O/S下层代理450或PC固件安全代理444、图5的固件安全代理516、或图7的微代码安全代理708或O/S下层代理712、图9的O/S下层捕获代理920或已触发事件应对程序922和/或其任何组合实现，或者被配置为实现它们的功能性。

[0294] 安全规则1208可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、图7的安全规则707、723、图9的安全规则908和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则1221可以由图2的安全规则220、图4的安全规则420、图7的安全规则721、图9的安全规则921和/或其任何组合实现，或者被配置为实现它们的功能性。

[0295] O/S下层安全1220可以被配置为截取对电子设备1201的存储器的访问。这样的存储器可以包括，例如，对物理存储器1203的地址的已尝试访问或对虚拟存储器1204的页面的已尝试访问。这样的尝试访问可以起源于操作系统1213或利用操作系统1213来在电子设备1201上运行的实体，例如应用1210或驱动程序1211。

[0296] 在一个实施例中，受O/S下层安全1220保护的存储器可以包括虚拟存储器1204。虚拟存储器1204可以包括诸如操作系统1213、应用1210或驱动程序1211之类的实体可用的存储器，已经从物理存储器和/或存储提取出该存储器。虚拟存储器1204可以是对诸如操作系统1213、应用1210或驱动程序1211之类的实体来说作为连续的一块存储器而出现，尽管所使用的实际空间可以跨越诸如物理存储器1203之类的实际物理存储器而分离地散布和/或在诸如盘的存储中。虚拟存储器1204可以根据处理器1202的扩展来虚拟化。虚拟存储器1204的地址空间可以被分割成存储器页面。存储器页面可以是每一相等的大小，例如四千字节。电子设备1201可以被配置为使用页面表来把虚拟存储器1204的虚拟地址转换成诸如物理存储器1203之类的存储器的物理地址或存储的地址。电子设备1201可以包括存储器管

理单元1214(“MMU”),存储器管理单元1214被配置为把虚拟存储器1204的虚拟地址转换成诸如物理存储器1203之类的存储器的物理地址和/或存储的地址。可以索引虚拟存储器1204的页面。对虚拟存储器1204页面的已尝试访问可以包括页面的意图的读、写或执行,且O/S下层安全代理1220可以被配置为捕获该尝试。在一个实施例中,虚拟存储器1204的页面可以对应于物理存储器地址或存储的地址。在另一实施例中,每一虚拟存储器1204的页面可以对应于物理存储器地址。在又一实施例中,包含诸如操作系统1213的特定部分之类的某些内容页面可以被固定且在电子设备1201的操作期间可以不改变。

[0297] 在另一实施例中,受O/S下层安全代理1220保护的存储器可以包括物理存储器1203。可以通过物理存储器的地址访问物理存储器1203,如标记(A)、(B)、(C)、(D)、(E)、(F)、(G)、(H)、(I)、(J)和(K)中所示出的,物理存储器的地址指示物理存储器1203中的特定地址,该地址可以是包含已定义元素的存储器范围的基地址。可以通过对特定的存储器地址的意图的读、写或执行来访问物理存储器1203,且O/S下层安全代理1220可以被配置为捕获该尝试。例如,意图的写可以采取指令“MOV Addr1,Value(移动地址1,值)”的形式,其中,由变量“Value(值)”表示的值被写到由“Addr1(地址1)”表示的特定存储器地址。可以使用写到物理存储器1203地址的任何指令。意图的读可以采取诸如“MOV Value,Addr1(移动值,地址1)”之类的指令的形式,其中,由“Addr1”表示的特定存储器地址读取由变量“Value(值)”表示的值。可以使用从物理存储器1203地址读取的任何指令。已尝试执行可以采取用物理存储器1203地址加载例如诸如“EIP”之类的指令指针寄存器的指令的形式,例如“MOV EIP,Addr1(移动EIP,地址1)”。这样的指令可以被配置为执行在由“Addr1”表示的地址开始的代码。可以使用用于执行存储器中的地址的任何指令。

[0298] O/S下层安全代理1220可以被配置为截取对虚拟存储器1204的已尝试访问。此外,O/S下层安全代理1220可以是被配置为截取对物理存储器1203的已尝试访问。在一个实施例中,可以不截取对虚拟存储器1204的请求,但截取在MMU已经把虚拟存储器1204页面转换成物理存储器1203地址之后的对物理存储器1203的随后的相应尝试访问,O/S下层安全代理1220可以被配置为截取对物理存储器的已尝试访问。在另一实施例中,可以直接做出对物理存储器1203的已尝试访问而不需要通过虚拟存储器1204转换,且O/S下层安全代理1220可以被配置为截取该已尝试访问。在再一个实施例中,可以截取对虚拟存储器1204做出的已尝试访问,但O/S下层安全代理1220可以不被配置为截取对物理存储器1203地址的随后访问。

[0299] O/S下层安全代理1220可以通信上耦合到O/S内部安全代理1219。O/S下层安全代理1220可以被配置为从O/S内部安全代理1219接收关于对电子设备1201的存储器的已尝试访问的上下文信息。由O/S内部安全代理1219提供的上下文信息可以包括已经尝试对电子设备1201的存储器的特定访问的实体的身份。

[0300] O/S下层安全代理1220可以通信上耦合到或包括存储器映射1206。可以以文件、记录、数据结构或其他合适的实体实现存储器映射1206。存储器映射1206可以包括关于电子设备1201的各种实体在存储器中的位置的信息。例如,如果进程被加载到电子设备1201的存储器中以供执行,则存储器映射1206可以包括关于虚拟存储器1204中的哪些存储器页面或物理存储器1203中的哪些地址范围包含该进程的信息。取决于电子设备1201中的存储器的虚拟化的实现,该进程的所有内容可以或者不可以被加载到物理存储器1203中,这是因

为一些内容可以被加载到诸如盘之类的存储中。对于要访问的这样的内容，它们可以被加载到物理存储器1203中。在这样的情况中，存储器映射1206可以包含关于内容被存储在其中的地址的信息，无论是在物理存储器1203中还是在诸如盘之类的存储中。O/S下层安全代理1220可以被配置为使用存储器映射1206来判断虚拟存储器1204页面或物理存储器1203地址中的任何给定内容的身份或所有者。例如通过剖析操作系统1213的操作，且然后判断各种敏感的组件位于存储器中的何处，O/S下层安全代理1220可以构建存储器映射1206。当做出对访问存储器的尝试时——例如加载操作系统1213内核，或执行内核模式指令——O/S下层安全代理1220可以被配置为与O/S内部安全代理1219通信以便判断加载或执行操作系统1213的什么部分。在另一示例中，O/S下层安全代理1220可以被配置为判断这样的虚拟存储器1204页面的存储器范围的内容的散列或数字签名。可以把散列或数字签名与安全规则1208中所包含的或从保护服务器202获得的已知值进行比较。已知的值可以是先前的表征的结果，其中例如已经标识操作系统1213的各部分。要映射的元素可以由安全规则1208确定。O/S下层安全代理1220可以被配置为在把元素从电子设备1201中的存储器的一个位置复制到另一位置时跟踪元素在存储器映射1206中的移动。

[0301] 图13是存储器映射的示例实施例的阐释。在一个实施例中，虚拟存储器映射1302可以包括要通过它们在虚拟存储器中的位置跟踪的元素的映射。在另一实施例中，物理存储器映射1304可以包括要通过它们在物理存储器中的位置跟踪的元素的映射。在各种实施例中，虚拟存储器映射1302和物理存储器映射1304可以被映射到一起，以使得可以在两种映射中跟踪元素。

[0302] 虚拟存储器映射1302可以反映十个不同的虚拟存储器页面。虚拟存储器映射1302可以阐释，例如，可以在存储器页面1和存储器页面2中找到这样的页面目录中的内核操作系统数据结构。在另一示例中，可以在存储器页面4-6中找到被称为“Fn1”的特定的进程、函数或例程的元素。在又一示例中，可以在页面8中找到用于系统服务分派表(“SSDT”)的权限的数据结构。在再一个示例中，可以在存储器页面8和存储器页面9中找到被称为“Fn2”的特定的进程、函数或例程的元素。

[0303] 物理存储器映射1304可以用物理存储器反映元素的位置。物理存储器中的元素的各部分可以跨越存储器散布在不邻近的片段或块中。此外，物理存储器中的元素的各部分可以跨越存储器以任意的次序散布。每一片段的大小在大小方面可以改变。片段可以在偏移基地址的地址处开始。图13中示出的示例基地址是00x000，在地址FFxFFF终止。表示物理存储器的各种片段的开始的地址被表示为(A)-(O)。对于被包含在物理存储器的多个片段内的元素，可以注意元素的次序。在物理存储器中，元素的多个片段可以由指针链接在一起，其中元素的一个片段的结束可以指向元素的下一片段。

[0304] 例如，Fn1可以被映射到在(A)和(B)、(J)和(K)以及(M)和(N)之间的片段。在另一示例中，SSDT权限可以被映射到在(G)和(H)之间的片段。在又一示例中，页面目录数据结构可以被映射到在(O)和FFxFFF、(F)和(G)以及(I)和(J)之间的片段。在再一个示例中，Fn2可以被映射到在(H)和(I)以及(B)和(C)之间的片段。

[0305] 返回到图12，O/S下层安全代理1220可以被配置为查阅安全规则1208以判断要保护存储器的什么部分以及如何保护它们。例如，安全规则1208可以被配置为指示页面目录数据结构仅可以由电子设备1201的某些经授权实体写入。因而，可以捕获对写入页面目录

数据结构的尝试，且可以检查尝试写入的元素可以以便判断它们是安全的、未知的还是已知是不安全的。O/S下层安全代理1220可以被配置为查阅存储器映射1206以判断页面目录数据结构位于存储器中何处。如果例如完全地或部分地在虚拟机监视器中实现O/S下层安全代理1220，则O/S下层安全代理1220可以被配置为在控制结构中设置标志以便捕获对虚拟存储器1204的存储器页面1和/或2的任何尝试写入。在另一示例中，如果完全地或部分地以微代码实现O/S下层安全代理1220，则O/S下层安全代理1220可以被配置为在控制结构中设置标志以便捕获对在物理存储器1203的地址(O)和FFxFFF、(F)和(G)和(I)和(J)之间的地址范围内的存储器地址的任何尝试写入。

[0306] 在另一示例中，安全规则1208可以被配置为指示Fn1仅可以由电子设备的某些经授权实体调用。因而，可以捕获对执行Fn1的尝试，且可以检查调用Fn1元素以便判断它们是安全的、未知的还是已知为不安全的。O/S下层安全代理1220可以被配置为查阅存储器映射1206以判断Fn1驻留在存储器中何处。如果例如完全地或部分地在虚拟机监视器中实现O/S下层安全代理1220，则O/S下层安全代理1220可以被配置为在控制结构中设置标志以便捕获对虚拟存储器1204的存储器页面4、5和/或6的已尝试执行。在另一示例中，如果完全地或部分地以微代码实现O/S下层安全代理1220，则O/S下层安全代理1220可以被配置为在控制结构中设置标志以便捕获对物理存储器1203的存储器地址(A)的任何已尝试执行。在其中可以分离地执行Fn1的不同部分的一些情况中，O/S下层安全代理1220可以被配置为捕获对在物理存储器1203的(A)和(B)、(M)和(N)、地址(O)和FFxFFF、(F)和(G)、(J)和(K)或(I)和(J)之间的范围内的任何存储器地址的任何已尝试执行。

[0307] 在一个实施例中，O/S下层安全代理1220可以被配置为查阅O/S内部安全代理1219以判断什么实体已经做出对写入存储器的调用，然后这被用来判断实体是否得到授权来做出写入。在另一实施例中，O/S下层安全代理1220可以是被配置为判断请求来自其中的虚拟存储器1204的存储器页面并查阅存储器映射1206以便判断这样的存储器页面是否与其中映射的任何元素相关联。在又一实施例中，O/S下层安全代理1220可以被配置为确定请求元素的存储器页面的散列或签名并将其与已知实体的散列和签名进行比较。

[0308] 如果O/S下层安全代理1220完全地或部分地由微代码实现，则O/S下层安全代理1220可以被配置为判断尝试写入的指令的地址。在一个实施例中，O/S下层安全代理1220可以被配置为通过检查指令指针以判断在物理存储器1203中的何处做出该指令来做出这样的判断。在另一实施例中，通过访问存储器映射1206，O/S下层安全代理1220可以被配置为从与地址相关联的存储器映射1206确定元素。在又一实施例中，O/S下层安全代理1220可以被配置为确定请求元素的散列或签名并将其与已知实体的散列和签名比较。

[0309] 一旦已经捕获对存储器的已尝试访问，则O/S下层安全代理1220可以被配置为访问安全规则1208以基于已标识的请求实体判断如何应对已捕获尝试。安全规则1208可以定义，例如，仅操作系统1213的某些指定的内核部分可以调用和执行Fn1，或者仅已知为安全的且在白名单上的实体可以写入SSDT的权限。然后，O/S下层安全代理1220可以被配置为采取任何适当的动作，例如允许请求进行、拒绝请求、欺骗响应或所写的值或执行矫正的进程。

[0310] 在操作中，O/S下层安全代理1220可以在低于诸如操作系统1213之类的电子设备1201的操作系统的级别运行。O/S下层安全代理1220可以访问安全规则1208以判断保护电

子设备1201的什么存储器资源。O/S下层安全代理1220可以确定、开发和/或增殖存储器映射1206的内容，为了这样做，O/S下层安全代理1220可以访问安全规则1208、保护服务器202或任何其他合适的信息源以便在存储器映射1206中增殖信息。O/S下层安全代理1220可以截取来自诸如操作系统1213、应用1210或驱动程序1211之类的处于操作系统级别的实体的对物理存储器1203或虚拟存储器1204的请求，以便在存储器映射1206中映射存储器的所有权和内容。O/S下层安全代理1220可以访问O/S内部安全代理1219以判断什么实体被加载到存储器中以使得可以增殖存储器映射1206。存储器映射1206可以包含用于物理存储器1203、虚拟存储器1204和/或在两者之间的映射的存储器映射。

[0311] O/S下层安全代理1220可以查阅安全规则1208以判断保护虚拟存储器1204和/或物理存储器1203的什么部分。安全规则1208可以指定，在动态的基础上保护存储器的一些部分，其中，可以由O/S下层安全代理1220取决于各种考虑允许或禁用对存储器的保护。这样的考虑可以包括，例如，管理员设置、恶意行为或可疑行为的检测、时间、先前所检测的对存储器的访问或任何其他合适的准则。如果保护电子设备1201的存储器在计算资源方面是昂贵的，这样的动态允许和禁用可以允许O/S下层安全代理1220更好地保护电子设备1201的存储器的关键部分，同时减少对电子设备1201执行其他任务的能力的副作用。例如，存储器包含操作系统1213的内核代码的内容可以总是受到O/S下层安全代理1220的保护，同时包含第三方应用1210的代码的内容的存储器可以仅在其他指示恶意软件存在或可以影响第三方应用1210时受到保护。

[0312] O/S下层安全代理1220可以在控制结构中设置标志以便捕获对物理存储器1203和/或虚拟存储器1204的已尝试访问。在一个实施例中，当从操作系统1213中的实体做出对被指派为要捕获的虚拟存储器1204中的存储器页面的请求时，O/S下层安全代理1220可以截取所尝试的请求。在另一实施例中，当做出对虚拟存储器1204中的存储器页面的请求时，O/S下层安全代理可以允许由MMU 1214把请求转换成对物理存储器1203中的地址的请求，在此O/S下层安全代理可以截取所尝试的请求。在又一实施例中，当直接地做出对物理存储器1203中的地址的、来自操作系统1213中的实体的请求时，O/S下层安全代理1220可以截取所尝试的请求。

[0313] 一旦已经截取了请求，O/S下层安全代理1220就可以使用任何合适的机制来评估所截取的对存储器的请求。安全规则1208可以被用来判断该尝试是否可疑，指示恶意软件对使用电子设备1201的资源的恶意尝试。安全规则1208可以包括以下考虑：例如，是否尝试读、写或执行；什么实体做出尝试；所访问的存储器地址或页面；相同的请求者的先前的尝试或动作；电子设备1201的管理员的安全设置，例如基于电子设备1201的用户或多或少带有限制性的规则；或由存储器位置和/或数字签名或散列确定或根据相关的页面或存储器地址的请求者的身份。

[0314] 例如，对虚拟存储器1204的页面2中的或物理存储器1203的地址(J)处的页面目录数据结构尝试写入可以由O/S下层安全代理1220截取。如果写入是来自未知的进程的存储器的部分，则O/S下层安全代理1220可以判断该写入是可疑的。然而，如果该尝试写入是来自操作系统1213内核的已知的经验证部分，那么，可以判断该尝试不是可疑的。同样地，可以截取对在虚拟存储器1204的页面8或在物理存储器1203的地址(H)处的Fn2的已尝试执行。如果该已尝试执行是从用户输入做出的，那么，可以判断该执行不是可疑的。如果已尝

试执行是从另一程序的存储器做出的，且该程序不在经核准列表上，那么，可以判断该尝试是可疑的或恶意的。

[0315] 在另一示例中，如果Fn1是出于互操作性的目的通常向其他应用公开其高速缓存的web浏览器，则O/S下层安全代理1220可以允许Fn1的存储器页面或存储器地址的指定部分由其他应用读取。然而，如果Fn1包含应保持私有的元数据或其他信息，那么，O/S下层安全代理1220可以保护Fn1的存储器页面或存储器地址的那些部分免遭来自不同于Fn1自身的任何进程的读取。

[0316] 一旦已经判断程序是可疑的、恶意的或以另外方式指示恶意软件，那么，O/S下层安全代理1220可以采取任何合适的矫正动作。O/S下层安全代理1220可以，例如，拒绝对虚拟存储器1204的存储器页面2或物理存储器1203的地址(J)的写请求，但仍返回该值被写入的所得到的指示。可以监视产生该请求的进程以便发现对访问电子设备1201的资源的附加尝试，可以停止该进程，或者可以从电子设备1201清除该进程。在另一示例中，对虚拟存储器1204的页面8或物理存储器1203的地址(H)的已尝试执行可以改为涉及蜜罐(honeypot)进程或清除进程的执行。

[0317] 受O/S下层安全代理1220保护的存储器的内容可以包括可能受到恶意软件攻击的数据、代码或任何其他有用的系统资源。O/S下层安全代理1220可以保护存储器的内容免遭恶意软件，该恶意软件尝试例如读、写或钩住显示在电子设备1201上运行的进程的机制、把其代码注射到被加载在存储器中的应用的部分、或改变虚拟存储器1204的映射表的权限和访问标志。通过在低于操作系统1213的级别操作，O/S下层安全代理1220可以避免在操作系统1213中以内核模式级别运行的恶意软件。O/S下层安全代理1220可以完成零日检测，这是由于在一些情况中不需要请求实体的身份先前已经被判断为是恶意的知识——实体是未知的这一事实可以被用来拒绝对电子设备1201的存储器的一些部分的访问。如果操作系统1213或在操作系统1213中运行的反病毒或反恶意软件措施被完全危害，则可以完全锁定存储器，使之免遭在操作系统的级别运行的实体。

[0318] O/S下层安全代理1220的一个应用可以是甚至在尝试对特定内容读、写或执行之前就通过检测对特定存储器页面的权限的改变来检测对虚拟存储器1204的内容的已尝试访问。MMU 1214所使用的存储器表可以驻留在存储器中、在虚拟存储器1204的页面自身中和/或在物理存储器1203的地址中。对改变存储器表的值例如把进程的代码部分的权限从“读”改变为“写”的尝试，本身可以被O/S下层安全代理1220捕获。存储器虚拟存储器1204的页面或物理存储器1203的地址可以受O/S下层安全代理1220保护，且对于把新值写到这样的位置的权限的已捕获尝试，O/S下层安全代理1220可以判断是否允许该尝试的请求者做出这样的改变。例如，如果改变进程的代码部分的权限的请求起因于不同的进程，则可以拒绝对权限的尝试改变。

[0319] 图14是使用对电子设备的已尝试访问的操作系统下层捕获来保护存储器的方法1400的示例实施例。在步骤1405，可以映射电子设备的虚拟存储器或物理存储器以便确定存储器的内容的身份或所有者。为了映射存储器，例如，可以访问保护服务器；可以跟踪存储器的读、写和执行；和/或扫描存储器的内容并为该内容产生和签名。

[0320] 在步骤1410，可以访问安全规则以便在步骤1415确定要保护的物理存储器的地址或虚拟存储器的页面。要保护的存储器可以取决于，例如，安全规则，电子设备的用户，诸如

恶意软件的指示、对访问受保护存储器的先前尝试之类的在电子设备中其他观察到的行为，或管理员设置。要保护的存储器可以动态地改变，这是由于电子设备的操作条件可以改变。安全规则可以指定要保护的电子设备的实体，且可以通过访问存储器映射来确定该实体在物理或虚拟存储器中的位置。

[0321] 在步骤1420，可以根据安全规则要求在控制结构中设置标志以便捕获对存储器的已尝试访问。可以为虚拟存储器的页面和/或物理存储器的地址设置这样的标志。标志可以包含要保护的存储器的指示以及要标识的访问方法的种类(例如—读、写或执行)。在步骤1425，可以监视对受保护存储器的访问以便查看是否已经对已指派的地址或页面做出对所指派的类型的已尝试访问。在步骤1430，可以判断是否已经捕获对访问存储器的尝试。如果不是，那么，在步骤1435可以判断要保护的存储器的标志是否要求改变。如果是，那么，方法1400可以返回到步骤1410以便访问安全规则以更新用户保护对存储器的访问的标志。如果不是，那么，方法1400可以返回到步骤1425以便监视对受保护存储器的已尝试访问。

[0322] 如果已经捕获对访问存储器的尝试，那么，在步骤1440开始，可以评估已捕获尝试。为了评估该尝试，可以查询存储器映射以判断从何处做出请求，并标识请求者。可以确定和评估要写的数据的值以便发现它们的内容。可以考虑尝试的本质——读、写或执行。可以与安全规则联合使用这些示例考虑，以便在步骤1445判断已尝试访问是否指示恶意软件。如果已尝试访问指示恶意软件，那么，在步骤1450，可以采取矫正动作。这样的矫正动作可以包括拒绝所请求的访问、返回经欺骗的值或发起蜜罐进程或矫正进程。如果已尝试访问不指示恶意软件，那么，在步骤1455可以允许该请求。方法1400可以根据继续保护电子设备的存储器的要求返回到步骤1425。

[0323] 图15是用于保护电子设备1504的操作系统1512内核的系统的示例实施例。系统1500可以包括O/S下层安全代理1516，O/S下层安全代理1516被配置为在电子设备1504上操作，以便防备恶意软件对访问操作系统1512的组件(例如，函数、数据和/或其他组件)以及与操作系统相关联的可信驱动程序的尝试。此外，O/S下层安全代理1516可以被配置为使用一个或多个安全规则1522来判断捕获什么尝试操作以及如何响应这样的已捕获操作。O/S下层安全代理1516可以被配置为对已捕获操作允许、拒绝或采取其他矫正动作。

[0324] 如图15中所示出，电子设备1504可以包括被耦合到存储器1508的处理器1506、一个或多个应用1510、一个或多个驱动程序1511、操作系统1512、O/S下层安全代理1516和安全规则1522。电子设备1504可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图1201的电子设备和/或其任何组合实现，或者被配置为实现它们的功能性。处理器1506可以被全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器902、图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。存储器1508可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、物理存储器1203或图12的虚拟存储器和/或其任何组合实现，或者被配置为实现它们的功能性。应用1510可以全部地或部分地由图1的应用110、图2的应用210、图4的应用410、图7的应用709、图9的应用910、图12的应用1210和/或其任何组合实现，或者被配置为实现它们的功能性。驱动程序1511可以全部地或部分地由图1的驱动程序111、图2的驱动程序211、图4的驱动程序411、图7的驱动程序711、图9的驱动程序911、图12的驱动程序1211和/或其任何组合实现，或者被配置为

实现它们的功能性。操作系统1512可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理1516可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。

[0325] 如图15中所示出，O/S下层安全代理1516可以包括安全规则1522。安全规则1522可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、或图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。可以以任何合适的方式(例如，由电子设备1504的用户设置的政策、包括电子设备1504的企业的管理员设置的政策、O/S下层安全代理1516的创建者设置的政策等等)建立安全规则1522。在一些实施例中，O/S下层安全代理1516可以从保护服务器202经由网络244请求和/或接收对安全规则1522的更新或修改(例如，由于对恶意软件定义的更新)。

[0326] 如图15中所示出，安全规则1522可以包括访问映射1562和政策1564。访问映射1562可以包括陈述操作系统1512的各种个体组件(例如，函数，数据和/或其他组件)和驱动程序1511的一个或多个可信访问以及关于一个或多个可信访问的上下文信息的日志、列表、映射或其他数据结构。图16是访问映射1562的示例实施例。在某些实施例中，可以通过仿真另一电子设备(例如，图18的电子设备1800)上充分地免遭恶意软件的操作系统(例如，图18的操作系统1812)及其充分地免遭恶意软件的可信驱动程序(例如，可信驱动程序1811)的执行来创建访问映射1562。下面相对于图18和19进一步详述根据这样的实施例的访问映射1562的创建。如图16中所示出，访问映射1562可以包括一个或多个函数访问子映射1602、一个或多个数据访问子映射1604和/或一个或多个栈区访问映射1606。

[0327] 对于操作系统1512或可信驱动程序1511的特定函数，函数访问子映射1602可以定义其他可信函数对特定函数的可信访问。函数访问子映射1602也可以包括与对函数的这样的可信访问相关联的上下文信息，在一些实施例中，这样的上下文信息包括特定驱动程序内可信访问的调用函数位于其中的代码段(例如，如由存储器位置所标识的)。

[0328] 对于操作系统1512或驱动程序1511的特定数据项，数据访问子映射1604可以定义可信函数对特定数据的可信访问。数据访问子映射1604也可以包括与对数据项的这样的可信访问相关联的上下文信息，在一些实施例中，这样的上下文信息包括与可信函数相关联的特定存储器位置，在其中可信函数位于特定的驱动程序内和/或可信访问是读访问还是写访问的代码段(例如，如由存储器位置所标识的)。

[0329] 栈区访问子映射1606可以定义描述在多个函数当中的经准许的调用关系的函数栈区。在栈区访问子映射1606中，栈区中的每一特定函数访问在函数栈区中出现在它下面的函数是受到信任的。栈区访问子映射1606可以包括与函数访问子映射1602的上下文信息类似的上下文信息。栈区访问子映射1606可以示出，例如，特定的函数F2可以调用函数F3，且函数F3可以调用F4，而F4调用F3以及F3调用F2不是可信的函数调用路径。

[0330] 在访问映射1562中陈述的各种函数、数据、代码段、驱动程序和其他实体的身份可

以由在存储器中特定的函数、数据、代码部分、驱动程序或实体被存储在其中的存储器位置(例如,物理存储器地址或虚拟存储器地址)来定义。图17是进一步阐释在图16的示例访问映射1562中所定义的函数和数据当中的相互关系的虚拟存储器1700的示例实施例。如图17中所叙述的,存储器1700可以包括分别位于存储器地址1701、1706、1710和1714的驱动程序Y1、Y2、Y3和Y4。驱动程序Y1可以包括在地址1702处的代码部分X1内的地址1703处的函数F1。驱动程序Y1也可以包括在地址1704处的数据部分D1内的数据指针1705。驱动程序Y2可以包括在地址1707处的代码部分X2内的地址1708处的函数F2。驱动程序Y3可以包括在地址1711处的代码部分X3内的地址1712处的函数F3。驱动程序Y4可以包括在地址1715处的代码部分X3内的地址1716处的函数F4。函数F2的存储器地址Z2可以驻留在存储器位置1709。函数F3的存储器地址Z3可以驻留在存储器位置1713。函数F4的存储器地址Z4可以驻留在存储器位置1717。图17中各种箭头描绘在图16的访问映射1562陈述的函数和数据当中的可信访问。例如,函数访问子映射1602可信访问由以下箭头叙述:指示在地址1708处的函数F2和地址1703处的函数F1之间的可信执行调用的箭头、指示在地址1712处的函数F3和地址1703处的函数F1之间的可信执行调用的箭头、以及指示在地址1716处的函数F4和地址1703处的函数F1之间的可信执行调用的箭头。

[0331] 返回到图15,政策1564可以包括陈述要应用的政策以便定义要由O/S下层安全代理1516捕获的事件和/或已捕获事件的应对的日志、列表或其他数据结构。在特定的实施例中,政策可以规定,响应于驱动程序函数对操作系统1512或可信驱动程序1511的存储器存储组件的一部分的已尝试访问(例如,读、写、执行、函数调用,如果访问映射1562中的条目指示这样的驱动程序函数拥有对这样的组件的访问权(包括,在一些实施例中,指示驱动程序函数出现在驱动程序的特定代码部分中,如访问映射1562中所定义的),则O/S下层安全代理1516可以允许这样的尝试访问。在相同的或替代的实施例中,政策可以规定,响应于驱动程序函数对操作系统1512或可信驱动程序1511的存储器存储组件的一部分的已尝试访问(例如,读、写、执行、函数调用),如果访问映射1562中没有条目指示这样的驱动程序函数拥有对这样的组件的访问权(包括,在一些实施例中,指示驱动程序函数出现在驱动程序的特定代码部分中,如访问映射1562中所定义的),则O/S下层安全代理1516可以拒绝这样的尝试访问。在这些和其他实施例中,政策可以规定,对于未知的驱动程序函数对操作系统1512或可信驱动程序1511的组件的已尝试访问,可以允许某些已尝试访问,且拒绝其他已尝试访问,和/或关于这样的访问的信息可以作为取证迹象而被传输到保护服务器202,以供进一步分析。

[0332] 在操作中,O/S下层安全代理1516可以根据在本公开内容中陈述的任何捕获技术捕获对操作系统1512和驱动程序1511的组件的已尝试访问。在一些实施例中,O/S下层安全代理1516可以根据安全规则1522捕获事件。响应于捕获对操作系统1512和驱动程序1511的组件的已尝试访问,O/S下层安全代理1516可以把与已尝试访问相关联的上下文信息与访问映射1562进行比较,以判断该已尝试访问是否可信。如果已尝试访问是可信的(例如,如果已尝试访问在访问映射1562中具有相应的条目),则O/S下层安全代理1516可以允许访问。如果已尝试访问是不可信的(例如,如果已尝试访问在访问映射1562中不具有相应的条目),O/S下层安全代理1516可以发起矫正动作。矫正动作可以包括拒绝已尝试访问、查询政策1564以便判断允许还是拒绝访问、和/或把关于这样的访问取证数据(例如,上下文信息)

报告给保护服务器202以供进一步处理。因此，O/S下层安全代理1516连同访问映射1562和政策1564，可以防备对操作系统1512和驱动程序1511的组件的恶意攻击。

[0333] 在一些实施例中，政策1564可以规定，由O/S下层安全代理1516响应于非可信尝试访问而发起的矫正动作可以取决于是潜在地非恶意的实体还是潜在地恶意的实体实施已尝试访问。潜在地恶意的实体可以是O/S下层安全代理1516未知(例如，不出现在白名单或黑名单中的任何中)的、发起非可信已尝试访问(例如，在访问映射1562中不具有相应的条目)的以及表现出指示潜在的恶意软件的存在行为(例如，尝试访问电子设备1504的敏感资源，尝试访问子函数而没有使用操作系统1512提供的函数路线等等)的应用、驱动程序或其他实体。潜在地非恶意的实体可以是不以另外方式被认为是潜在地恶意的实体的任何实体。在潜在地非恶意的实体情况中，政策1564可以允许某些已尝试访问而拒绝其他。例如，对于潜在地非恶意的实体，可以允许网络调用和文件系统调用，而可以拒绝对修改内部网络分派例程指针、修改内部网络驱动程序接口规范(NDIS)指针、或写入到内核代码部分、数据部分或系统服务分派表(SSDT)的已尝试访问。另一方面，对于潜在地恶意的实体，可以拒绝所有尝试访问。

[0334] 在其他实施例中，政策1564可以规定，可以允许未知实体(例如，不出现在白名单或黑名单中的实体)有限制地执行一次已尝试访问，在此之后可以把关于访问的信息传输给保护服务器202且进一步评估以判断任何进一步的矫正动作。

[0335] 图18是用于产生访问映射1562的系统1800的示例实施例。系统1800可以包括O/S下层安全代理1816，O/S下层安全代理1816被配置为在电子设备1804上操作以便基于所观察到的操作系统1812和可信驱动程序1811的行为在访问映射1562中产生条目。如图18中所示出的，电子设备1804可以包括处理器1806、存储器1808、驱动程序1811、操作系统1812和O/S下层安全代理1816。电子设备1804可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图1201的电子设备和/或其任何组合实现，或者被配置为实现它们的功能性。

[0336] 处理器1806可以包括，例如微处理器、微控制器、数字信号处理器(DSP)、专用集成电路(ASIC)或被配置为解释和/或执行程序指令和/或进程数据的任何其他数字或模拟电路。在一些实施例中，处理器1806可以解释和/或执行被存储在存储器1808中的程序指令和/或处理数据。存储器1808可以部分地或整体地被配置为应用存储器、系统存储器或两者。存储器1808可以包括被配置为持有和/或容纳一个或多个存储器模块的任何系统、设备或装置；例如，存储器1808可以包括只读存储器、随机存取存储器、固态存储器、或基于盘的存储器。每一存储器模块可以包括被配置为在一段时间内保留程序指令和/或数据的任何系统、设备或装置(例如，计算机可读的非暂态介质)。

[0337] O/S下层安全代理1816可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444、或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220、图15的O/S下层安全代理1516和/或其任何组合实现，或者被配置为实现它们的功能性。

[0338] 操作系统1812可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213、图15的操作

系统1512和/或任何组合实现，或者被配置为它们的实现功能性。可信驱动程序1811可以全部地或部分地由图1的驱动程序111、图2的驱动程序211、图4的驱动程序411、图7的驱动程序711、图9的驱动程序911、图12的驱动程序1211、图15的驱动程序1511和/或其任何组合实现，或者被配置为实现它们的功能性。然而，结合在电子设备1804中的使用，操作系统1812可以免遭恶意软件且可信驱动程序1811可以仅包括已知是非恶意的且免遭恶意软件的那些驱动程序。例如，可以对电子设备1804多加小心，以便确保操作1812和可信驱动程序1811不包括恶意实体。作为特定的示例，操作系统1812和可信驱动程序1811可以被安装在电子设备1804的空的或新近格式化的计算机可读介质上，且可以小心保证没有不同于0/S下层安全代理1816的其他实体被安装在电子设备1804。

[0339] 在操作中，0/S下层安全代理1816可以根据在本公开内容中陈述的任何捕获技术捕获对操作系统1812和可信驱动程序1811的组件的已尝试访问。响应于捕获对系统1812和可信驱动程序1811的组件的访问，0/S下层安全代理1816可以确定与访问相关联的上下文信息并存储访问的记录和上下文信息(例如，作为函数访问子映射1602、数据访问子映射1604、函数栈区访问子映射1606或其他合适的方式的一部分)。因而，充分地免遭恶意软件的电子设备1804的执行和在操作系统1812及其可信驱动程序1811当中的可信信赖可以由0/S下层安全代理1816观察到，以产生访问映射1562的条目，其中，每一条目定义对操作系统1812或可信驱动程序1811的组件的可信访问。因为基于已知充分地免遭恶意软件的实体的仿真执行产生访问映射1562的条目，访问映射1562可以包括操作系统1812及其可信驱动程序1811的标准预期行为的表示，而没有附加实体。因而，访问映射1562可以仅包括拥有对操作系统1812及其可信驱动程序1811的组件的合法、非恶意的访问权的条目。

[0340] 因此，一旦电子设备1804的0/S下层安全代理1816产生访问映射1562，就可以使得访问映射1562对0/S下层安全代理1516可用(例如，通过经由网络244访问访问映射1562，通过把访问映射下载到电子设备1504，通过经由计算机可读存储介质传递到电子设备1504等等)，其中如上所述，0/S下层安全代理1516可以捕获对操作系统1512和/或驱动程序1511的组件的已尝试访问，以判断已尝试访问中哪些是可信的或非可信的，并基于这样的判断采取进一步的动作。结果，0/S下层安全代理可以保护操作系统1512和可信驱动程序1511免遭恶意访问。

[0341] 图19是用于保护电子设备的操作系统内核的方法1900的示例实施例。在方法1900，在其上安装有充分地免遭恶意软件的操作系统和关联可信驱动程序的第一电子设备上执行的第一0/S下层安全代理可以被用来创建访问映射(例如参见步骤1905-1910)。另外，在第二电子设备上执行的第二0/S下层安全代理可以通过引用访问映射来保护被安装在第二电子设备上的第二操作系统及其相关的驱动程序的组件(例如参见步骤1915-1930)。

[0342] 在步骤1905，在其上安装有充分地免遭恶意软件的操作系统和关联可信驱动程序的第一电子设备上运行的第一0/S下层安全代理可以捕获对操作系统和/或可信驱动程序的组件(例如，函数和数据)的访问。在步骤1910，第一0/S下层安全代理可以把关于访问的信息(包括与访问相关联的上下文信息)记录到访问映射。相对于另一调用函数对函数的访问，这样的上下文信息可以包括其中可信访问的调用函数位于特定的驱动程序内的代码段(例如，由存储器位置所标识的)。相对于调用函数对数据项的访问，这样的上下文信息可以

包括与可信函数相关联的特定存储器寄存器、其中可信函数位于特定的驱动程序内和/或可信访问是读访问还是写访问的代码段。

[0343] 在步骤1915，在第二电子设备上执行的第二O/S下层安全代理可以捕获对在第二电子设备上执行的操作系统和/或驱动程序的组件的已尝试访问。在步骤1920，响应于捕获对组件的已尝试访问，第二O/S下层安全代理可以把与已尝试访问相关联的上下文信息与访问映射进行比较，以判断已尝试访问是否可信。如果已尝试访问在访问映射中具有相应的条目，则已尝试访问可以是可信的。如果已尝试访问是可信的，则法1900可以进行到步骤1925。如果已尝试访问是不可信的，则法1900可以进行到步骤1930。

[0344] 在步骤1925，响应于判断已尝试访问是可信的，第二O/S下层安全代理可以允许已尝试访问。在完成步骤1925之后，方法1900可以再次进行到步骤1915。

[0345] 在步骤1930，响应于判断已尝试访问是不可信的，第二O/S下层安全代理可以发起矫正动作。矫正动作可以包括拒绝已尝试访问、查询政策以便判断允许还是拒绝访问和/或把关于这样的访问的取证数据(forensic data)(例如，上下文信息)报告给保护服务器以供进一步处理。在完成步骤1925之后，方法1900可以再次进行到步骤1915。

[0346] 图20是提供操作系统执行环境以便安全地执行操作系统的、被配置为保护电子设备2001免遭恶意软件的系统2000的示例实施例。来自图20的元素可以与图21和图22中它们的同名配对物相同。系统2000可以包括被配置为提供操作系统执行环境2008(“OSEE”)的受保护起动的起动模块2020。起动模块2020可以被配置为通过确保诸如操作系统(“O/S”)下层安全代理2004、操作系统2012和O/S内部安全代理2016之类的OSEE 2008的组件在起动之前不受恶意软件禁用来提供OSEE 2008的受保护起动。在起动模块2020成功地提供OSEE2008的安全起动之后，OSEE 2008的诸如O/S下层安全代理2004和O/S内部安全代理2016之类的组件，可以协作以便防止恶意软件感染电子设备2001的诸如起动模块2020之类的组件。

[0347] 电子设备2001可以包括被配置为提供OSEE 2008的受保护起动的起动模块2020。OSEE 2008可以包括O/S下层安全代理2004和O/S内部安全代理2016以提供用于执行一个或多个操作系统2012的安全环境。电子设备2001也可以通信上耦合到保护服务器2022，以便辅助提供用于执行一个或多个操作系统2012的安全环境。保护服务器2022可以包括备份存储设备2024。电子设备2001可以全部地或部分地由图1的电子设备103、图2的电子设备104、图4的电子设备404、图7的电子设备701和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备2001可以包括资源2026，例如一个或更多处理器2002、存储器2003或存储设备2006。处理器2002可以全部地或部分地由图2的处理器208、图4的处理器406、图7的处理器702和/或其任何组合实现，或者被配置为实现它们的功能性。存储器2003可以全部地或部分地由图2的存储器207、图4的存储器408、图7的存储器703和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统2012可以由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713和/或其任何组合实现，或者被配置为实现它们的功能性。可以在图22的O/S内部安全代理2206的讨论中找到O/S内部安全代理2016的示例实施例的描述。可以在图22的O/S下层安全代理2208的讨论中找到O/S下层安全代理2004的示例实施例的描述。

[0348] 存储设备2006可以由图1的资源106、图2的系统资源214、图4的存储426、图5的I/O

设备502和/或其任何组合实现，或者被配置为实现它们的功能性。存储设备2006可以包括用于存储数据或其他信息的任何合适的资源。例如，存储设备2006可以包括但不限于直接访问存储设备(例如，硬盘驱动器或软盘)、连续访问存储设备(例如，磁带磁盘驱动器)、紧致盘、CD-ROM、DVD、随机存取存储器(RAM)和/或闪速存储器(例如，基于闪存的固态驱动器)。存储设备2006可以被分割成一个或多个扇区，每一扇区能够存储固定量的数据。例如，存储设备2006可以被分割成每个512字节的扇区，但可以使用任何合适的扇区大小。在各种实施例中，存储设备2006可以被定位为远离电子设备2001，例如在保护服务器2022上。在其他实施例中，存储设备2006可以是电子设备2001的本地资源2026。

[0349] 备份存储设备2024可以包括用于存储数据或其他信息的任何合适的资源。例如，备份存储设备2024可以由存储设备2006实现，或者被配置为实现它的功能性。备份存储设备2024可以由电子设备2001的诸如存储设备2006之类的本地存储设备实现。在其他实施例中，备份存储设备2024可以由位于网络上的远程存储设备实现，例如在保护服务器2022上。如果备份存储设备2024位于网络上，则O/S下层安全代理2004可以使用网络连接来访问备份存储设备2024。可以在低于操作系统2012的优先级水平实现网络连接以便避免使用操作系统内核的网络设备驱动程序，网络设备驱动程序可能感染恶意软件。可以使用主动管理技术(AMT)来实现网络连接，主动管理技术可以允许通过直接地访问电子设备2001的网卡使用HTTPS、iSCSI、NFS或CIFS客户机来访问备份存储设备2024。在这样的实施例中，尽管访问备份存储设备2024要求有网络连接，但备份存储设备2024可以与在电子设备2001的操作系统2012上执行的任何恶意软件隔离。

[0350] 保护服务器2022可以被定位为远离电子设备2001且可以被配置为与电子设备2001的诸如起动模块2020、O/S下层安全代理2004和O/S内部安全代理2001之类的组件通信，以提供安全规则2018或发送和接收其他信息。例如，保护服务器2022可以接收关于对访问资源2026的可疑尝试的信息且可以存储这种信息以供随后分析。保护服务器2022可以由图1的保护服务器102、图2的保护服务器202和/或其任何组合实现，或者被配置为实现它们的功能性。

[0351] 安全规则2018可以包括任何合适的规则、逻辑、命令、指令、标志或用于指定要求捕获的事件和每一事件的适当响应的其他机制。安全规则2018可以由图1的安全规则114、图2的安全规则220、222、图4的安全规则420、422、434、436、438、图5的安全规则518、图7的安全规则707、721、723和/或其任何组合实现，或者被配置为实现它们的功能性。

[0352] 起动模块2020可以被配置为通过确保OSEE 2008的诸如O/S下层安全代理2004、操作系统2012和O/S内部安全代理2016之类的组件在起动之前不被恶意软件禁用来提供OSEE 2008的受保护起动。起动模块2020可以通过检验与O/S下层安全代理2004、操作系统2012和O/S内部安全代理2016相关联的一个或多个受保护文件的完整性来估定O/S下层安全代理2004、操作系统2012和O/S内部安全代理2016是否被恶意软件禁用。如果起动模块2020在任何受保护文件中检测到恶意软件，则起动模块2020可以被配置为从备份副本还原受保护文件。在起动模块2020验证OSEE 2008的组件不被恶意软件禁用或起动模块2020成功地恢复被恶意软件禁用的OSEE 2008的任何组件之后，起动模块2020可以起动OSEE 2008。在起动OSEE 2008时，起动模块2020可以在起动诸如操作系统2012之类的OSEE 2008的其他组件之前起动O/S下层安全代理2004。

[0353] 在起动模块2020成功地提供OSEE 2008的安全起动之后，诸如O/S下层安全代理2004和O/S内部安全代理2016之类的OSEE 2008的组件可以协作以便防止恶意软件感染电子设备2001的资源2026。例如，O/S下层安全代理2004和/或O/S内部安全代理2016可以被配置为截取对访问存储设备2026上的各种受保护文件的尝试，如安全规则2018所指定的。受保护文件可以包括与起动模块2020、O/S下层安全代理2004、或O/S内部安全代理2016、或操作系统2012的核心文件相关联的文件。保护这些文件免遭恶意软件可以帮助确保这些组件所采用的防护设施不被恶意软件破坏。例如，通过在操作系统2012执行的同时保护起动模块2020免遭恶意软件，电子设备2001的下次启动起动模块2020将免遭恶意软件。以这种方式，在电子设备2001被引导时可以由起动模块2020检查诸如O/S下层安全代理2004、O/S内部安全代理2016和操作系统2012之类的OSEE2008的组件以便发现恶意软件，且在操作系统2012执行的同时可以由OSEE2008的组件保护起动模块2020免遭恶意软件。

[0354] 图21是用于提供受保护操作系统执行环境的系统中的起动模块2102的示例实施例。来自图21的元素可以与图20和图22中它们的同名配对物相同。例如，可以使用起动模块2102来实现来自图20的系统的起动模块2020或来自图22的系统的起动模块2226的功能性。起动模块2102可以被配置为通过安全地起动O/S下层安全代理2128、操作系统2124和O/S内部安全代理2126提供受保护操作系统执行环境2122。

[0355] 起动模块2102可以包括引导代理2104、受保护起动代理2110和恢复代理2112。引导代理2104可以被配置为确保，在电子设备2101启动时，在操作系统2124和任何其他软件(例如，恶意软件)之前引导受保护起动代理2110。受保护起动代理2110可以被配置为安全地起动OSEE 2122。OSEE 2122可以是用于安全地执行操作系统2124的执行环境。通过利用安全规则2116来判断O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126是否已经感染了恶意软件，受保护起动代理2110可以提供OSEE 2122的受保护起动。例如，通过扫描存储设备2114上每一组件的盘映像以便发现已知的模式的恶意软件、通过比较每一组件的盘映像的密码散列值、和/或通过使用检测恶意软件的任何其他合适的方法，受保护起动代理2110可以检查OSEE 2122的组件以便发现恶意软件。如果受保护起动代理2110检测到恶意软件感染，则可以利用恢复代理2112来从恶意软件感染恢复。如果受保护起动代理2110没有检测到恶意软件感染，或者如果恢复代理2112完成了成功恢复，则受保护起动代理2110可以被配置为起动O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126。O/S下层安全代理2128可以由图22的O/S下层安全代理2208实现，或者被配置为实现它的功能性。O/S内部安全代理2126可以由图22的O/S内部安全代理2206实现，或者被配置为实现它的功能性。操作系统2124可以由图20的操作系统2012实现，或者被配置为实现它的功能性。存储设备2114可以由图20的存储设备2006实现，或者被配置为实现它的功能性。安全规则2116可以由图20的安全规则2018实现，或者被配置为实现它的功能性。

[0356] 引导代理2104可以包括主引导记录(“MBR”)管理器2106和引导程序加载器2108，且可以被配置为确保在电子设备2101启动时，在操作系统2124和诸如恶意软件之类的任何其他软件之前引导受保护起动代理2110。MBR管理器2106可以被配置为用引导程序加载器2108替换存储设备2114上的现有MBR2130。MBR 2130可以位于存储设备的第一扇区(即，扇区0)，且可以负责在电子设备2101启动时引导操作系统2124或其他软件。通过用引导程序加载器2108替换MBR 2130，引导程序加载器2108可以变成新的MBR 2130。将不执行原始MBR

2130，且因此，将不引导与原始MBR 2130相关联的操作系统2124或其他软件。相反，在电子设备2101启动时，由于引导程序加载器2108已经变成了新的MBR 2130，将执行引导程序加载器2108。引导程序加载器2108可以被配置为引导受保护起动代理2110，受保护起动代理2110负责起动OSEE2122。以这种方式，可以在操作系统2124和/或任何其他软件之前引导受保护起动代理2110，允许受保护起动代理2110在加载O/S下层安全代理2128、O/S内部安全代理2126和/或操作系统2124之前检查恶意软件。

[0357] 受保护起动代理2110可以被配置为起动OSEE 2122。OSEE 2122可以被配置成用于安全地执行操作系统2124的执行环境，且可以包括O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126。受保护起动代理2110可以由能够提供盘I/O功能性、网络I/O功能性、和基本控制台I/O功能性的瘦嵌入式操作系统实现。在另一实施例中，受保护起动代理2110可以由O/S下层安全代理2128实现。受保护起动代理2110可以被配置为检测O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126是否已经感染了恶意软件。为了检测恶意软件感染，受保护起动代理2110可以使用密码散列算法来验证与O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126相关联的各种受保护文件2120的完整性。受保护文件可以包括，例如，MBR 2130、操作系统2124的核心文件和O/S下层安全代理2128和/或O/S内部安全代理2126的可执行映像。为了验证受保护文件2120的完整性，受保护起动代理2110可以使用散列算法来计算受保护文件2120的散列值。然后，可以把所计算散列值与先前产生的受保护文件2120的散列值进行比较。如果散列值不同，那么，受保护文件2120可能已经被恶意软件修改或变更。在各种实施例中，安全代理2110可以利用盘映射位图（“DMB”）2118来验证受保护文件2120的完整性。盘映射位图2118可以指定每一受保护文件2120在存储设备2114上的位置，且也可以提供先前产生的每一受保护文件2120的散列值。可以在来自图23的盘映射位图2301的讨论中找到盘映射位图2118的示例实施例的描述。受保护起动代理2110可以查阅盘映射位图2118，以便标识受保护文件2120在存储设备2114上的位置、计算受保护文件2120的散列值并将所计算的散列值与由盘映射位图2118提供的先前所产生的散列值进行比较。如果受保护文件2120的散列值不匹配，则受保护文件2120可能已经被恶意软件变更或修改。受保护起动代理2110可以起动恢复代理2112以便从潜在的恶意软件感染恢复。如果没有检测到潜在的恶意软件感染，或者如果由恢复代理2112成功地恢复所有潜在被感染的文件，则受保护起动代理2110可以进行到加载O/S下层安全代理2128、操作系统2124和O/S内部安全代理2126。安全起动代理2110可以被配置为在起动OSEE 2122之后终止。

[0358] 恢复代理2112可以被配置为从与O/S下层安全代理2128、操作系统2124和/或O/S内部安全代理2126相关联的一个或多个受保护文件2120的恶意软件感染恢复。为了从恶意软件感染恢复，恢复代理2112可以被配置为从备份存储设备检索备份文件并用相应的备份文件替换受感染的受保护文件2120。备份文件可以本地存储在电子设备2101上，例如在存储设备2114上。备份文件也可以是被存储在远离电子设备2101的位置。例如，备份文件可以存储在网络上，例如在来自图20的保护服务器2022的备份存储设备2024上。可以维护备份文件的元数据，且其可以包括修订版本号以及创建备份文件的日期和时间。在把备份文件用于恢复受保护文件2120之前，恢复代理2112可以被配置为验证备份文件的完整性以便确保备份文件没有感染恶意软件。恢复代理2112可以以与受保护起动代理2110验证受保护文

件2120的完整性类似的方式验证备份文件的完整性。例如，恢复代理2112可以计算备份文件的散列值且可以把所计算的散列值与来自盘映射位图2118的备份文件的相应散列值进行比较。如果散列值的比较指示备份文件可能感染了恶意软件，则不可以使用该备份文件和/或可以使用较旧的备份文件。恢复代理2112可以被配置为向受保护起动代理2110告知成功恢复，以便允许受保护起动代理2110进行起动O/S下层安全代理2128、操作系统2124、和O/S内部安全代理2126。

[0359] 图22是用于安全地执行操作系统的操作系统执行环境(“OSEE”)2202的示例实施例。来自图22的元素可以与图20和图21中它们的同名配对物相同。例如，可以使用OSEE 2202来实现来自图20的OSEE 2008或来自图21的OSEE 2122的功能性。OSEE 2202可以被配置成用于安全地执行操作系统2204的执行环境，且可以包括操作系统2204、O/S下层安全代理2208、O/S内部安全代理2206和/或盘安全代理2214。OSEE 2202可以由起动模块2226安全地起动。此后，诸如O/S下层安全代理2208、O/S内部安全代理2206和盘安全代理2214之类的OSEE 2202的组件可以协作以便防止恶意软件禁用电子设备2201的组件。例如，OSEE 2202的组件可以协作以便保护起动模块2226免遭恶意软件。以这种方式保护起动模块2226可以帮助确保在电子设备2201的下次初始化中，不会破坏起动模块2226所使用的防护设施以便允许起动感染恶意软件的操作系统2204、O/S下层安全代理2208和/或O/S内部安全代理2206。

[0360] OSEE 2202可以包括O/S下层安全代理2208、操作系统2204、O/S内部安全代理2206和/或盘安全代理2214。OSEE 2202可以由起动模块2226安全地起动。在起动模块2226成功地提供OSEE 2202的安全起动之后，诸如O/S下层安全代理2208、O/S内部安全代理2206和盘安全代理2214之类的OSEE 2202的组件可以协作以便防止恶意软件禁用诸如起动模块2226之类的电子设备2201的组件。

[0361] O/S下层安全代理2208可以包括O/S下层捕获代理2210和已触发事件应对程序2212。O/S下层捕获代理2210可以由图1的O/S下层捕获代理104、图2的SVMM 216、图4的固件安全代理440、442或PC固件安全代理444、图5的固件安全代理516、图7的微代码安全代理708和/或其任何组合实现，或者被配置为实现它们的功能性。已触发事件应对程序2010可以由图1的已触发事件应对程序108、图2的SVMM安全代理217、图4的O/S下层代理450、图7的O/S下层代理712和/或其任何组合实现，或者被配置为实现它们的功能性。在各种实施例中，O/S下层捕获代理2210的功能性中的一些可以由已触发事件应对程序2212实现，或已触发事件应对程序2212的功能性中的一些可以由O/S下层捕获代理2210实现。在一个实施例中，已触发事件应对程序2212可以与O/S下层安全代理2208相同的优先级水平操作。在另一实施例中，已触发事件应对程序2212可以被实现为O/S内部安全代理2206的部分，且可以在操作系统2204的优先级水平操作或者在高于操作系统2204的优先级水平操作。在再一个实施例中，已触发事件应对程序2212可以由两个或更多个已触发事件应对程序实现，其中，至少一个已触发事件应对程序在与O/S下层安全代理2208相同的优先级水平操作，且至少一个已触发事件应对程序在操作系统2204的优先级水平操作或在高于操作系统2204的优先级水平操作。

[0362] O/S下层安全代理2208可以被配置为使用O/S下层捕获代理2210来截取对访问诸如存储设备2218之类的电子设备2201的资源的请求。在截取对访问存储设备2218的请求

时，O/S下层捕获代理2210可以被配置为创建与已捕获访问尝试相关联的已触发事件，且可以被配置为把已触发事件发送给已触发事件应对程序2212以判断相对于该事件采取的适当动作。已触发事件可以包括诸如与请求相关联的存储设备2218的区域(例如，扇区和/或文件)、请求实体和所请求的访问的类型之类的信息。请求实体是负责发起请求的实体，例如操作系统2204、驱动程序2228或应用2230。所请求的访问的类型可以包括对读、写或执行来自存储设备2218的代码的请求。

[0363] 已触发事件应对程序2212可以被配置为接收和处理来自O/S下层捕获代理2210的已触发事件。已触发事件可以包含关于已经被O/S下层捕获代理2210捕获的、对访问存储设备2218的请求的信息。已触发事件应对程序2212可以被配置为结合与已触发事件相关联的上下文信息利用一个或多个安全规则2216来标识对访问存储设备2218的受保护区域的尝试，并且判断适当的响应。在标识对访问诸如受保护的扇区和/或文件之类的受保护区域的尝试之后，已触发事件应对程序2212可以被配置为查阅安全规则2216以便判断对访问受保护区域的尝试是否得到授权。已触发事件应对程序2212还可以被配置为向O/S下层安全代理2208提供适当动作的判断。例如，已触发事件应对程序2212可以告知O/S下层安全代理2208应当允许还是拒绝已触发事件，或者是否应当采取其他矫正动作。

[0364] O/S内部安全代理2206可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理719和/或其任何合适的组合实现，或者被配置为实现它们的功能性。O/S内部安全代理2206可以在操作系统2204的优先级水平执行或在高于操作系统2204的优先级水平执行，且可以被配置为查阅一个或多个安全规则2216以便保护电子设备2201免遭恶意软件。例如，安全规则2216可以要求O/S内部安全代理2206截取对访问存储设备2218上的某些受保护文件2222的尝试。安全规则2216还可以指定对访问受保护文件2222的特定尝试是否得到授权。然而，因为O/S内部安全代理2206在操作系统2204的优先级水平执行或在高于操作系统2204的优先级水平执行，O/S内部安全代理2206自身可能感染了在操作系统2204上执行的恶意软件，且O/S内部安全2206的防护设施可能被规避了。为了帮助防止这种可能性，O/S下层安全代理2208可以被配置为保护O/S内部安全代理2206免遭恶意软件。

[0365] 盘安全代理2214可以包括DMB产生器2232和盘保护器2234，且可以被用来保护诸如起动模块2226和OSEE 2202的组件之类的电子设备2201的组件免遭恶意软件。盘安全代理2214可以以任何合适的方式实现。在一个实施例中，盘安全代理2214可以被实现为O/S下层安全代理2208的部分和/或可以在与O/S下层安全代理2208相同的优先级水平执行。在另一实施例中，盘安全代理2214可以被实现为O/S内部安全代理2206的部分和/或可以是在操作系统2204的优先级水平操作或在高于操作系统2204的优先级水平操作。在再一个实施例中，盘安全代理2214可以由两个或更多个盘安全代理实现，其中，至少一个盘安全代理在与O/S下层安全代理2208相同的优先级水平操作，且至少一个盘安全代理在操作系统2204的优先级水平操作或在高于操作系统2204的优先级水平操作。

[0366] 盘保护器2234可以被配置为通过截取对访问与这些组件相关联的各种受保护文件2222的未经授权的尝试来保护起动模块2226和OSEE 2202的组件免遭恶意软件。受保护文件2222可以包括核心操作系统文件(例如，操作系统内核文件)、核心安全代理文件(例如，O/S下层安全代理2208和O/S内部安全代理2206的可执行映像)和/或这些文件的备份副

本。通过截取对访问其中存储有受保护文件2222的存储设备2218的扇区的未经授权的尝试，盘保护器2234可以防止对受保护文件2222的未经授权的访问。在一些实施例中，盘保护器2234可以使用盘映射位图2220来标识受保护文件2222以及其中存储有受保护文件2222的存储设备2218上的扇区。可以在图23的盘映射位图2301的讨论中找到盘映射位图2220的示例实施例的描述。盘映射位图2220可以包含与各种受保护文件相关联的信息，例如，包括其中存储有每一受保护文件的存储设备的扇区或多个扇区。盘保护器2234可以查阅盘映射位图2220以便标识其中存储有受保护文件2222的存储设备2218的扇区。然后，盘保护器2234可以截取对访问与受保护文件2222相关联的扇区的尝试，且可以查阅安全规则2216以便判断该尝试是否得到授权。例如，安全规则2216可以指定，除非该请求是来自操作系统2204，否则应当拒绝对写入核心操作系统文件的请求。

[0367] 在一些实施例中，盘保护器2234的功能性可以由O/S下层安全代理2208的组件实现。通过把盘保护器2234实现为O/S下层安全代理2208的组件，盘保护器2234可以在低于操作系统2204的级别执行，且可以避免烦扰操作系统2204的大多数恶意软件。例如，可以由O/S下层捕获代理2210和已触发事件应对程序2212实现盘保护器2234的功能性。O/S下层捕获代理2210可以被配置为查阅盘映射位图2220以便标识要求保护的存储设备2218的扇区。O/S下层捕获代理可以还被配置为捕获对访问存储设备2218的已标识扇区的尝试，且可以利用安全规则2216来判断该尝试是否得到授权。以这种方式，可以保护由盘映射位图2220标识的受保护文件2222免遭未经授权的访问。

[0368] 在其他实施例中，盘保护器2234的功能性可以被实现为O/S内部安全代理2206的组件。例如，O/S内部安全代理2206可以包括盘过滤驱动程序以便实现盘保护器1133的功能性。过滤驱动程序可以是驱动程序2228，驱动程序2228可被插入到用于操作系统2204的特定设备的现有驱动程序栈区，且可以被用来补充先前存在的驱动程序功能性。例如，盘过滤驱动程序可以被插入到用于盘(例如，存储设备2218)的现有驱动程序栈区，且可以补充先前存在的盘驱动程序的功能性。通过查询盘映射位图2220以便标识要求保护的存储设备2218的扇区，盘过滤驱动程序可以实现盘保护器1133的功能性。然后，盘过滤驱动程序可以截取对访问存储设备2218的受保护扇区的尝试，且可以利用安全规则2216来判断该尝试是否得到授权。以这种方式，将保护由盘映射位图2220标识的受保护文件2222免遭未经授权的访问。然而，因为盘过滤驱动程序在操作系统2204的优先级水平执行或者在高于操作系统2204的优先级水平执行，盘过滤驱动程序可能自身感染了在操作系统2204上执行的恶意软件，且盘过滤驱动程序的防护设施可能被规避了。因此，在一些实施例中，盘保护器2234的功能性可以由O/S下层安全代理2208和O/S内部安全代理2206两者实现。例如，如上所述，O/S内部安全代理2206可以被配置为使用盘过滤驱动程序来截取对访问存储设备2218的未经授权的尝试，且O/S下层安全代理2208可以被实现为防止对修改存储器中或存储设备2218上的盘过滤驱动程序映像的未经授权的尝试，由此保护盘过滤驱动程序免遭在与操作系统2204相同的优先级水平运行的恶意软件的破坏。

[0369] 盘保护器2234还可以被配置为在关闭电子设备2201之前验证MBR的完整性。例如，在启动电子设备2201的关闭时，盘保护器2234可以被配置为计算MBR 2224的散列值。然后，盘保护器2234可以查阅盘映射位图2220以便获得先前所产生的MBR 2224的散列值且可以将所计算的散列值与先前所产生的散列值进行比较。如果散列值不同，那么，MBR 2224可能

已经被恶意软件变更，且盘保护器2234可以被配置为用备份副本替换MBR 2224。以这种方式，在下次启动电子设备2201时，将不引导感染恶意软件的MBR 2224。

[0370] DMB产生器2232可以被配置为产生和更新盘映射位图2220。例如，DMB产生器2232可以被配置为确定其中存储每一受保护文件2222的存储设备2218上的扇区，且可以进一步被配置为产生每一受保护文件2222的散列值。DMB产生器2232可以把每一受保护文件2222的相应扇区和散列值存储在盘映射位图2220中。DMB产生器2220可以以任何合适的方式实现。例如，DMB产生器2220的功能性可以被实现为O/S下层安全代理2208或O/S内部安全代理2206的部分，或DMB产生器2220的功能性可以由O/S下层安全代理2208和O/S内部安全代理2206两者实现。

[0371] 在一个实施例中，DMB产生器2232可以通过截取对访问受保护文件2222的请求产生盘映射位图2220。例如，O/S内部安全代理2206可以包括文件系统过滤驱动程序，该文件系统过滤驱动程序被配置为截取对访问受保护文件2222的请求。文件系统过滤驱动程序截取针对文件系统或另一文件系统过滤驱动程序的请求。通过在请求到达其预期目标之前截取该请求，过滤驱动程序可以扩展或替换请求的原始目标提供的功能性。来自O/S内部安全代理2206的文件系统过滤驱动程序可以截取涉及受保护文件2222的文件I/O请求。然后，过滤驱动程序可以查询文件系统以便获得其中存储有受保护文件2222的内容的存储设备2218上的扇区。然后，过滤驱动程序可以访问文件系统的主格式表(MFT)以判断受保护文件2222的盘扇区布局。可以更新盘映射位图2220以便指定其中存储有受保护文件2222的所标识的扇区。如果没有为受保护文件2222产生散列值，则可以产生散列值，且可以更新盘映射位图2220以便包括新散列值。如果受保护文件2222被更新，则也可以产生新散列值并将其存储在盘映射位图2220中。例如，如果文件系统过滤驱动程序截取对写入受保护文件2222的请求，则需要使用受保护文件2222的经修改内容来产生新的散列值。

[0372] 图23是在用于提供受保护操作系统执行环境的系统或方法中使用的盘映射位图2301的示例实施例。例如，盘映射位图2301可以被用来实现图21的盘映射位图2118、图22的盘映射位图2220或图26的盘映射位图2628的功能性。盘映射位图2301可以是文件且可以包含与各种受保护的文件2302相关联的信息。例如，盘映射位图2301可以标识其中存储有每一受保护文件2302的存储设备的扇区2304，且可以包括每一受保护文件2302的散列值2306。盘映射位图2301可以被用来验证各种受保护文件2302的完整性。例如，图21的受保护起动代理2110和/或恢复代理2112可以使用来自盘映射位图2301的信息验证受保护文件2302的完整性。例如，可以由来自图22的DMB产生器2232产生盘映射位图2301。

[0373] 盘映射位图2301可以被存储在存储设备上的已指派扇区。已指派扇区可以驻留在被用来实现操作系统的文件系统的存储设备的相同部分。可以把已指派扇区标志为被占据以便防止各扇区被操作系统使用。也可以把存储设备进行分区，以便允许把盘映射位图2301存储在与操作系统不同的分区的已指派扇区。盘映射位图2301也可以是被存储在位于网络上的远程存储设备。例如，盘映射位图2301可以被存储在诸如来自图20的保护服务器2022或来自图26的保护服务器2602之类的保护服务器上。

[0374] 盘映射位图2301可以标识每一受保护文件2302、其中存储有受保护文件2302的存储设备的扇区或多个扇区2304和受保护文件2302的散列值2306。由盘映射位图2301标识的受保护文件2302可以包括核心安全代理文件2308、核心操作系统文件2310和备份文件

2312。核心安全代理文件2308可以包括MBR和可执行的O/S下层安全代理和O/S内部安全代理。核心操作系统文件2310可以包括操作系统内核文件和其他操作系统文件。例如，如果操作系统是微软Windows™的变种，则核心操作系统文件2310可以包括ntoskrnl.exe、hal.sys、win32k.sys、ntfs.sys、disk.sys和/或tcpip.sys。核心操作系统文件2310可以取决于特定的操作系统而改变。备份文件2312可以包括每一核心安全代理文件2308和每一核心操作系统文件2310的备份副本。在各种实施例中，备份文件2312可以不被存储在与核心安全代理文件2308和核心操作系统文件2310相同的存储设备上。在这样的实施例中，盘映射位图2301也可以标识其中存储有备份文件2312的特定存储设备。替代地，分离的盘映射位图2301可以被用来存储与备份文件2312相关联的信息，例如扇区2304和散列值2306。

[0375] 对于每一受保护文件2302，盘映射位图2301可以存储使用密码散列算法产生的散列值2306。散列算法可以包括可以接收作为输入的数据块且可以产生作为输出的位串或散列值的算法。不同数据集的散列值通常是不同的。把每一受保护文件2302的内容用作对散列算法的输入，产生每一受保护文件2302的散列值2306。可以使用任何合适的密码散列算法，例如，包括安全散列算法2(“SHA-2”)或消息摘要算法5(“MD5”)。

[0376] 盘映射位图2301可以例如由图21的受保护起动代理2110和/或恢复代理2112、或来自图22的O/S下层安全代理2208、O/S内部安全代理2206和/或盘安全代理2214、图26的O/S下层安全2616和/或O/S内部安全代理2618用来检测受保护文件2301的潜在恶意软件感染。为了检测受保护文件2302的潜在恶意软件感染，可以使用散列算法用来验证受保护文件2302的完整性。可以查询盘映射位图2304以便标识其中存储有受保护文件2302的存储设备上的扇区2304，且然后可以从存储设备的适当扇区2304检索受保护文件的内容。然后，可以使用诸如SHA-2或MD5之类的所选择的散列算法来使用受保护文件2302的内容产生散列值，且可以把所产生的散列值与来自盘映射位图2301的相应散列值2306进行比较。如果散列值不同，那么，受保护文件2302可能已经被恶意软件修改或变更。

[0377] 可以以任何合适的方式产生盘映射位图2301。在一个实施例中，可以通过截取对访问受保护文件2302的请求、获得与受保护文件2302相关联的信息以及用关于受保护文件信息更新盘映射位图2301来产生盘映射位图2301。在一些实施例中，可以由诸如例如来自图26的O/S下层安全代理2616之类的在比操作系统较低的优先级环执行的软件截取请求。在其他实施例中，可以由诸如例如来自图26的O/S内部安全代理2618之类的在与操作系统相同的优先级环执行的软件截取请求。例如，来自图26的O/S内部安全代理2618可以包括文件系统过滤驱动程序。文件系统过滤驱动程序可以截取涉及受保护文件2302的文件I/O请求。然后，过滤驱动程序可以查询文件系统以便获得其中存储有受保护文件2302的内容的存储设备上的扇区2304。然后，过滤驱动程序可以访问文件系统的主文件表(MFT)以判断受保护文件2302的盘扇区布局。可以更新盘映射位图2301以便指定其中存储有受保护文件2302的已标识扇区2304。如果文件系统过滤驱动程序截取对写入受保护文件2302的请求，则可以使用受保护文件2302的经更新内容来产生新的散列值，且可以更新盘映射位图2301以便存储新的散列值。

[0378] 图24是用于起动受保护操作系统执行环境的方法的示例实施例。在步骤2410，可以用被配置为引导受保护起动环境的替代MBR来替换存储设备的现有MBR。MBR可以位于存储设备的第一扇区(即，扇区0)且可以在电子设备的启动时就执行。以这种方式，在启动电

子设备时，可以不执行原始MBR，且因此，可以不加载与原始MBR相关联的操作系统或其他软件。相反，可以执行替代MBR且其可以加载受保护起动环境。在步骤2420，可以启动电子设备，且因此可以执行来自步骤2410的替代MBR。替代MBR可以进行到加载受保护起动环境。

[0379] 在步骤2430，可以获得安全规则。安全规则可以被本地存储在存储设备上，或者可以远程存储例如在保护服务器上。这样的安全规则可以被用来在步骤2440-2480做出判定。在步骤2440，可以判断是否已经创建了各种受保护文件的备份副本。可以在安全规则中指定要求备份的受保护文件。备份文件可以包括例如替代MBR、与受保护起动环境相关联的文件、与一个或多个安全代理相关联的文件和核心操作系统文件。如果还没有创建备份副本，那么，在步骤2450创建备份副本。备份副本可以被本地存储在存储设备上，或者可以远程存储例如在保护服务器上。

[0380] 在步骤2460可以判断安全代理或操作系统是否感染了恶意软件。安全代理可以包括O/S下层安全代理和/或和O/S内部安全代理。在一个实施例中，可以通过检验与安全代理和操作系统相关联的各种受保护文件的完整性来检查安全代理和操作系统以便发现恶意软件。散列算法可以被用来验证受保护文件的完整性。例如，可以使用受保护文件的内容来计算每一受保护文件的散列值，且可以把所计算的散列值与先前所产生的受保护文件的散列值进行比较。如果受保护文件的散列值不同，那么，受保护文件可能已经被恶意软件修改。在一些实施例中，盘映射位图可以标识存储设备上的其中存储有每一受保护文件的扇区，且也可以包括先前所产生的每一受保护文件的散列值。在这样的实施例中，可以查询盘映射位图以判断其中存储有受保护文件的内容的扇区，并且可以使用受保护文件的内容来计算散列值。也可以查询盘映射位图以检索先前所产生的受保护文件的散列值以使得可以把先前所产生的散列值与所计算的散列值进行比较。如果受保护文件的散列值不同，那么，可以假定是恶意软件感染，且在步骤2470，可以从潜在的恶意软件感染恢复受保护文件。如果受保护文件的散列值匹配，那么，受保护文件没有受到变更，且因此没有感染恶意软件。在这种情况中，该方法可以进行到步骤2480，步骤2480中可以加载安全代理和操作系统。

[0381] 在步骤2470，可以对潜在的恶意软件感染执行恢复。通过检索已经受到感染的每一受保护文件的备份副本并用相应的备份副本替换潜在地受感染的受保护文件，执行恢复。备份副本可以位于本地存储设备上或可以远程位于例如保护服务器上。在使用备份副本来替换潜在地受感染的受保护文件之前，也可以验证备份文件的完整性以便确保备份文件它们自身没有感染恶意软件。

[0382] 在已经使用相应的备份副本来恢复受保护文件之后，在步骤2480，可以加载安全代理和操作系统。安全代理可以包括O/S下层安全代理和/或O/S内部安全代理。O/S下层安全代理可以在低于操作系统的优先级水平执行，且O/S内部安全代理可以执行在操作系统的优先级水平执行或在高于操作系统的优先级水平执行。O/S下层安全代理和O/S内部安全代理可以协作以便保护电子设备免遭恶意软件。例如，O/S下层安全代理和/或O/S内部安全代理可以保护诸如存储设备之类的电子设备的资源免遭未经授权的访问。在一些实施例中，可以给可以负责提供O/S下层安全代理、O/S内部安全代理和/或操作系统的安全起动的电子设备的组件提供保护。例如，O/S下层安全代理和/或O/S内部安全代理可以保护负责执行步骤2410-2470的那些组件。以这种方式，在下一次启动电子设备时，在步骤2420加载的受保护起动环境可以不受恶意软件禁用。

[0383] 根据保护存储设备的需要，可以连续地、周期性地、根据需求或在事件触发时重复来自图24的方法的步骤，事件的触发可以包括恶意软件和/或其他可疑行为的检测。

[0384] 图25是提供用于安全地执行操作系统的操作系统执行环境的方法2500的示例实施例。在步骤2505，可以认证O/S下层安全代理、O/S内部安全代理和保护服务器的身份和安全。可以使用任何合适的方法执行这样的认证，包括通过使用密码散列和/或使用密钥来定位和检验每一组件的存储器中的映像。直到完成步骤2505，可以停止其他步骤的操作。在步骤2510，可以获得安全规则。安全规则可以被O/S下层安全代理和/或O/S内部安全代理本地存储在存储设备上，或者可以远程存储在例如保护服务器上。这样的安全规则可以被用来在步骤2515-1475做出判定。

[0385] 在步骤2515，可以截取访问受保护文件的尝试。所截取的尝试可以在操作系统级别发生或者在高于操作系统级别发生，例如由O/S内部安全代理进行，或者它可以在低于操作系统的级别发生，例如由O/S下层安全代理进行。受保护文件可以包括MBR、与一个或多个安全代理相关联的文件、用来起动一个或多个安全代理(例如，来自图21的加载模块2102)的文件和核心操作系统文件。可以由安全规则指定受保护文件。在步骤2520，可以判断是否需要把受保护文件的条目添加到盘映射位图。盘映射位图可以被实现为文件或其他数据结构，且可以存储关于受保护文件的某些信息，例如每一受保护文件位于其中的存储设备上的扇区和与每一受保护文件相关联的散列值。如果盘映射位图不包含在步骤2515正在被访问的受保护文件的这种信息，可以把受保护文件的条目添加到盘映射位图。例如，盘映射位图可以不指定其中存储有受保护文件的扇区，或者可以不指定受保护文件的散列值。如果盘映射位图丢失了这种信息，那么，在步骤2525可以更新盘映射位图以便包括这信息。为了更新盘映射位图，可以标识存储受保护文件的内容的扇区，并且可以使用受保护文件的内容来产生散列值。确定其中存储有受保护文件的存储设备上的扇区可以涉及查询文件系统和访问主格式表(MFT)以判断受保护文件的扇区布局。然后，可以从存储设备的适当扇区检索受保护文件的内容，且然后可以把受保护文件的内容用作密码散列算法的输入来计算散列值。然后，可以把受保护文件的相应扇区和所计算散列值存储在盘映射位图中。

[0386] 在步骤2530，可以判断对受保护文件的访问是否得到授权。这种判断可以在操作系统级别发生或者在高于操作系统级别发生，例如由O/S内部安全代理进行，或者它可以在低于操作系统的级别发生，例如由O/S下层安全代理进行。可以结合安全规则分析与访问受保护文件的尝试请求相关联的上下文信息，以便判断是否可以授权请求实体访问受保护文件。例如，安全规则可以指定可以或不可以授权操作系统、特定应用和/或特定设备驱动程序访问受保护文件。安全规则也可以指定可以得到授权访问受保护文件的请求实体的访问权限，例如读、写或执行。如果对受保护文件的访问没有得到授权，那么，在步骤2555，可以拒绝访问。如果对受保护文件的访问得到授权，那么，在步骤2535，可以判断受保护文件是否已更新。如果受保护文件已更新，那么，在步骤2540，也可以更新盘映射位图。例如，如果对受保护文件更新引起被用来存储文件的存储设备上的扇区的改变，可以更新盘映射位图以便标识用来存储受保护文件的适当扇区。另外，可以产生受保护文件的新散列值并将其存储在盘映射位图中。在步骤2545，也可以更新受保护文件的备份副本以便反映对受保护文件的近来更新。

[0387] 如果对受保护文件的访问得到授权，那么，在步骤2550，可以允许对受保护文件的

访问。如果对受保护文件的访问没有得到授权，那么，在步骤2555，可以拒绝访问，并且在步骤2560，可以把关于该访问尝试的任何可疑信息报告给保护服务器。

[0388] 在步骤2565，可以判断是否检测到电子设备的关机。如果没有检测到关机，那么，该方法可以在步骤2515再次继续，以便继续截取对访问受保护文件的尝试。如果检测到关机，那么，在步骤2570可以验证MBR的完整性，以便确保在电子设备下次启动不引导感染恶意软件的MBR。可以通过使用MBR的内容计算散列值并把所计算的散列值与来自盘映射位图的先前所产生的散列值进行比较来验证MBR的完整性。如果散列不同，则MBR已经被变更，且可以用备份副本替换。在验证了MBR的完整性之后，在步骤2575，可以关闭电子设备。

[0389] 根据保护存储设备的要求，可以连续地、周期性地、根据需求或在事件触发时重复来自图25的方法的步骤。

[0390] 图26是用于保护存储设备2606免遭未经授权的访问的系统900的示例实施例。系统900可以包括通信上耦合到已触发事件应对程序2608的操作系统(“O/S”)下层安全代理2616。O/S下层安全代理2616可以包括被配置为捕获对访问电子设备2601的存储设备2606的尝试的O/S下层捕获代理2604。O/S下层捕获代理2604可以被配置为创建与已捕获的访问请求相关联的已触发事件并把已触发事件发送给已触发事件应对程序2608。已触发事件应对程序2608可以被配置为查阅一个或多个安全规则2614或保护服务器2602以判断如何应对已触发事件。已触发事件应对程序2608也可以被配置为评估已触发事件的倾向是恶意软件或破坏存储设备2606的恶意尝试的指示。此外，已触发事件应对程序2608可以被配置为向O/S下层捕获代理2604提供应当允许还是拒绝已触发事件的判断，或者可以被配置为产生另一矫正动作。O/S下层安全代理2616可以通信上耦合到在操作系统2612中运行的O/S内部安全代理2618。系统900可以被配置为使用备份存储设备2620还原存储设备2606上的数据。

[0391] 电子设备2601可以全部地或部分地由图1的电子设备103、图2的电子设备104、图4的电子设备404和/或图7的电子设备701或其任何组合实现，或者被配置为实现它们的功能性。电子设备2601可以包括被耦合到存储器2603的一个或多个处理器2602。处理器2602可以全部地或部分地由图2的处理器208、图4的处理器406和/或图7的处理器702或其任何组合实现，或者被配置为实现它们的功能性。存储器2603可以全部地或部分地由图2的存储器207、图4的存储器408和/或图7的存储器703或其任何组合实现，或者被配置为实现它们的功能性。电子设备2601可以包括操作系统2612，操作系统2612可以包括O/S内部安全代理2618。操作系统2612可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412和/或图7的操作系统713或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理2618可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418和/或图7的O/S内部安全代理719或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0392] 存储设备2606可以由图1的资源106、图2的系统资源214、图4的存储426、或图5的I/O设备502实现，或者被配置为实现它们的功能性。存储设备2606可以包括用于存储数据或其他信息的任何合适的资源。例如，存储设备2606可以包括但不限于直接访问存储设备(例如，硬盘驱动器或软盘)、连续访问存储设备(例如，磁带磁盘驱动器)、紧致盘、CD-ROM、DVD、随机存取存储器(RAM)盘和/或闪速存储器(例如，基于闪存的固态驱动器)。存储设备

2606可以包括大容量存储设备。不考虑与系统总线的连接类型或接口方法，存储设备2606可以包括被连接到电子设备2601的存储设备，系统总线可以包括但不限于PCI、串行ATA、USB或火线。存储设备2606可以包括永久的块设备。存储设备2606可以被分割成一个或多个扇区924，每一扇区能够存储固定量的数据。例如，存储设备2606可以被分割成每个512字节的扇区，但可以使用任何合适的扇区大小。存储设备2606上的扇区924可以是静态的或动态的。静态扇区的位置固定，而动态扇区不固定。例如，主引导记录2626(MBR)是静态的且位于扇区0，即存储设备2606上的第一扇区。要求保护的动态扇区包括存储主文件表(即，包含与存储在文件系统上的所有文件相关联的元数据的文件)、操作系统内核文件、设备驱动程序和诸如O/S下层安全代理2616或O/S内部安全代理2618之类的反恶意软件应用的扇区。因为动态扇区不固定，必须把存储在动态扇区上的文件从它们在文件系统上的概念存在映射到来自文件的数据驻留在其中的存储设备2606的物理扇区。

[0393] O/S下层安全代理2616可以由图1的O/S下层捕获代理104、图2的SVMM 216、图4的固件安全代理440、442或PC固件安全代理444、图5的固件安全代理516或图7的微代码安全代理708实现，或者被配置为实现它们的功能性。在借助于图4的固件安全代理440或442或图5的固件安全代理516的功能性来实现O/S下层安全代理2616的实施例中，可以在存储设备2606的固件中实现O/S下层安全代理2616。已触发事件应对程序2608可以由图1的已触发事件应对程序108、图2的SVMM安全代理217、图4的O/S下层代理450或图7的O/S下层代理712实现，或者被配置为实现它们的功能性。在借助于图4的固件安全代理440或442或图5的固件安全代理516的功能性来实现已触发事件应对程序2608的实施例中，可以在存储设备2606的固件中实现已触发事件应对程序2608。在各种实施例中，O/S下层安全代理2616的功能性中的一些可以由已触发事件应对程序2608实现，或已触发事件应对程序2608的功能性中的一些可以由O/S下层安全代理2616实现。此外，O/S下层安全代理2616和已触发事件应对程序2608可以在相同的软件模块中实现。

[0394] 可以在比电子设备2601的操作系统2612较低的功能的级别实现O/S下层安全代理2616。例如，O/S下层安全代理2616可以截取操作系统2612、驱动程序2611或应用2610对存储设备2606的已尝试访问。O/S下层安全代理2616可以在电子设备2601的处理器上运行而无需使用操作系统。在一个实施例中，O/S下层安全代理2616可以在裸机环境或执行级别上操作。另外，O/S下层安全代理2616可以在比电子设备2601的所有操作系统2612更高的优先级环上执行，如由电子设备2601的处理器所定义的。例如，在其中较低的数字表示较高的优先级的使用保护环的分级保护域模型的上下文中，操作系统2612可以是在“0环”操作而O/S下层安全代理2616可以“1环”操作。驱动程序2611和应用2610可以在“0环”或“3环”操作。电子设备2601的操作系统可以在0环运行。

[0395] 通过在“1环”运行，O/S下层安全代理2616可以避免烦扰诸如操作系统2612之类的操作系统的大多数恶意软件。O/S下层安全代理2616可以对在0环或更高运行的实体透明地操作。因而，可以由操作系统2612或另一实体以相同的方式请求访问存储设备2606的尝试，而不考虑是否存在O/S下层安全代理2616。在强加访问存储设备2606的请求时，O/S下层安全代理2616可以允许请求、拒绝请求、破坏存储设备2606上的数据、破坏存储设备2606的介质表面、加密存储设备2606上的数据或采取其他矫正动作。为了拒绝请求，O/S下层安全代理2616可以简单地防止请求到达存储设备2606或处理器2602，或者可以向该请求提供经欺

骗的或哑元应答以便使得操作系统2612相信该动作已经发生。为了允许请求，0/S下层安全代理2616可以简单地把请求传送给存储设备2606或处理器2602。为了破坏数据，0/S下层安全代理2616可以被配置为重写或以另外方式移除存储设备2606上的数据。为了破坏存储设备2606的介质表面，0/S下层安全代理2616可以执行动作以便把存储设备2606呈现为不可操作为读或写数据。为了加密存储设备2606上的数据，0/S下层安全代理2616可以使用任何合适的加密算法来加密存储设备2606上的数据并用经加密的数据替换存储设备2606上的未经加密的数据。

[0396] 0/S下层安全代理2616可以包括被配置为捕获对访问存储设备2606的请求的0/S下层捕获代理2604。可以由操作系统2612、驱动程序2611或应用2610发起对访问存储设备2606的请求。0/S下层捕获代理2604可以被配置为标识负责发起请求的请求实体。0/S下层捕获代理2604还可以被配置为创建与已捕获访问尝试相关联的已触发事件并把已触发事件发送给已触发事件应对程序2608以便判断相对于该事件要采取的适当动作。已触发事件可以包括诸如与请求相关联的存储设备2606的区域(例如，扇区和/或文件)、请求实体和所请求的访问的类型之类的信息。与请求相关联的存储设备2606的区域可以是存储设备2606的一个或多个扇区，或者可以是被存储在存储设备2606上的文件。请求实体可以是操作系统2612、驱动程序2611或应用2610。例如，如果应用2610或驱动程序2611请求访问存储设备2606，则已触发事件可以指示请求访问的特定的应用2610或驱动程序2611。如果请求是来自操作系统2612而非特定的应用2610或驱动程序2611，则已触发事件可以指示该请求是来自操作系统2612。所请求的访问的类型可以包括从存储设备2606读取、向存储设备2606写入或执行存储设备2606上的代码的请求。

[0397] 在一个实施例中，0/S下层捕获代理2604可以被配置为仅在检测到诸如对系统的攻击、恶意软件感染或任何其他潜在的安全威胁之类的事件之后捕获对访问存储设备2606的请求。在这样的实施例中，保存系统100的资源直到已经检测到潜在的安全威胁。在另一实施例中，0/S下层捕获代理2604可以被配置为在所有时刻捕获对访问存储设备2606的请求，而不考虑是否已经检测到潜在的安全威胁。

[0398] 在另一实施例中，0/S下层安全代理2616可以被配置为通过捕获对用于文件输入和输出的驱动程序或系统函数的调用的执行来捕获对存储设备2606的已尝试访问。这样的调用的捕获可以虚拟存储器页面级别完成，其中，可以由0/S下层安全代理2616标识和保护包含这样的驱动程序或系统函数的存储器页面。在这样的情况中，0/S下层安全代理2616可以部分地或完全地例如由虚拟机监视器或在微代码中实现。这样的调用的捕获可以在物理存储器地址级别完成，其中，可以由0/S下层安全代理2616标识和保护这样的驱动程序或系统函数的代码段的存储器地址。在这样的情况中，0/S下层安全代理2616可以完全地或部分地例如以微代码实现。恶意软件可以直接调用这样的函数，在这种情况中，0/S下层安全代理2616可以确定这样的函数的调用者，以便标识该调用者是否拥有访问存储设备2606的特定部分的权限。通过例如调用未经建档的文件函数的子函数或者直接地分支到函数的代码部分而根本无需调用函数，恶意软件可以间接地调用这样的函数。这样的尝试可以被用来隐藏调用者的身份或以另外方式模糊恶意软件对文件I/O的使用。在这样的情况中，通过捕获子函数的执行或通过捕获通向文件I/O函数的代码部分的JMP或分支指令，0/S下层安全代理2616可以捕获已尝试文件I/O。这样的行为本身是可疑的，因此即使调用者是未知的，

O/S下层安全代理2616也可以被配置为判断这样的已尝试访问的宿主是可疑的且该尝试可以指示恶意软件。

[0399] 在又一实施例中，O/S下层安全代理2616可以被配置为通过捕获为访问盘而产生的中断来捕获对存储设备2606的已尝试访问。这样的中断可以由正常的文件I/O函数调用，或者可以由避免使用函数并尝试直接写入到存储设备2606恶意软件产生。O/S下层安全代理2616可以被配置为确定中断的源、标识中断的本质、标识任何上下文信息或参数、标识中断的目标、并判断该尝试是否可疑。例如，该尝试是否可疑的判定可以包括调用者的身份，或者动作自身是可疑的。例如，恶意软件可以执行一系列指令，其中，可以把要写的扇区的计数(例如“MOV al,count(移动al,计数)”)、要写的磁道的标识(例如“MOV ch,track(移动ch,磁道)”)、要写的扇区的标识(例如“MOV cl,sector(移动cl,扇区)”)，要写的首部的标识(例如“MOV dh,head(移动dh,首部)”)，要写的卷的标识(例如“MOV dl,drive(移动dl,驱动程序)”)、要执行的文件的类型的标识(例如“MOV ah,03h”)以及要写到文件的数据的存储器位置(例如“MOV bx,buf”)移动到通用寄存器。把这样的信息指派给特定的通用寄存器可以是用于加载用于随后的文件I/O中断的信息的已知方法。可以用“MOV(移动)”指令做出这些指派。随后，可以执行产生中断13的指令，例如“INT 13h”。O/S下层安全代理2616可以被配置为捕获命令并检查关联寄存器的内容以判断已尝试文件I/O的本质以及存储设备2606的所针对的部分。O/S下层安全代理2616可以被配置为查阅安全规则以便判断这样的操作的调用者是否拥有写入到存储设备2606的指定部分的权限。O/S下层安全代理2616可以被配置为检查执行历史以便判断这样的命令序列是否起因于经授权文件I/O驱动程序，或者它们是否由未知的或恶意进程直接执行。在这样的情况中，基于这样的行为，即使先前不已知调用者的状态是恶意的，也可以判断调用者是恶意的。最终，即使做出标准文件I/O驱动程序调用以便执行中断，也可以标识驱动程序的调用者，且O/S下层安全代理2616可以被配置为判断调用者是否拥有访问所考虑的存储设备2606的部分的权限。

[0400] O/S下层安全代理2616可以包括映射代理2622。映射代理2622可以被配置为把文件从其在文件系统上的概念存在映射到该文件存储在其中的存储设备2606的扇区924。在一个实施例中，映射代理2622可以在与O/S下层安全代理2616相同的优先级环操作。在另一实施例中，映射代理2622可以被实现为O/S内部安全代理2618的部分且可以在与操作系统2612、驱动程序2611或应用2610相同的优先级环是操作。在再一个实施例中，映射代理2622可以由两个或更多个映射代理实现，其中，至少一个映射代理在与O/S下层安全代理2616相同的优先级环操作，且至少一个映射代理在操作系统2612、驱动程序2611或应用2610的优先级环操作。映射代理2622可以接收对映射来自O/S下层捕获代理2604或已触发事件应对程序2608的文件的请求，且可以响应通过提供该文件存储在其中的存储设备2606上的扇区。这样的实施例可以允许O/S下层捕获代理2604和/或已触发事件应对程序2608标识对访问不总是被存储在存储设备2606的相同扇区上的动态定位的文件或数据的请求。例如，主文件表、操作系统内核文件、设备驱动程序和反恶意软件软件的位置可以不总是在存储设备2606的相同扇区924上，且映射代理2622可以被用来标识这些文件存储在其中的扇区。在一些实施例中，映射代理2622可以查询文件系统以判断受保护文件存储在其中的扇区。映射代理2622也可以使用盘映射位图2628来标识受保护文件存储在其中的存储设备2606上的扇区924。盘映射位图2628可以由图23的盘映射位图2301实现，或者被配置为实现它的功

能性。盘映射位图2628可以包含与各种受保护文件相关联的信息，例如，包括其中每一受保护文件存储在其中的存储设备的扇区或多个扇区。如果受保护文件经过更新，则也可以更新来自盘映射位图2628的信息。以这种方式，在映射代理2622接收对把受保护文件从其在文件系统上的概念存在映射到该文件驻留在其中的存储设备2606的扇区924的请求时，映射代理2622可以查阅盘映射位图2628以便标识对应于受保护文件的扇区924。

[0401] 已触发事件应对程序2608可以由通信上耦合到一起的一个或多个事件应对程序或安全代理实现。可以在相同的安全代理中实现已触发事件应对程序2608和O/S下层捕获代理2604。在一个实施例中，已触发事件应对程序2608可以在与O/S下层捕获代理2604相同的优先级环操作。在另一实施例中，已触发事件应对程序2608可以被实现为O/S内部安全代理2618的部分且可以在与操作系统2612、驱动程序2611或应用2610相同的优先级环操作。在再一个实施例中，已触发事件应对程序2608可以由两个或更多个已触发事件应对程序实现，其中，至少一个已触发事件应对程序在与O/S下层安全代理2616相同的优先级环操作，且至少一个已触发事件应对程序操作系统2612、驱动程序2611或应用2610的优先级环操作。通过在O/S下层捕获代理2604的优先级环运行，已触发事件应对程序2608可以类似地避免“0环”或“3环”恶意软件感染该代理自身的问题。然而，“0环”或“3环”与操作系统2612、驱动程序2611或应用2610一起运行的已触发事件应对程序2608可以提供从“1环”代理的角度来看不可用的关于对存储设备2606的已尝试访问的上下文信息。

[0402] 已触发事件应对程序2608可以被配置为接收和处理来自O/S下层捕获代理2604的已触发事件。已触发事件应对程序2608也可以被配置为把安全规则2614提供给O/S下层安全代理2616和/或O/S下层捕获代理2604。已触发事件可以包含关于已经被O/S下层捕获代理2604捕获的、对访问存储设备2606的请求的信息。已触发事件应对程序2608可以被配置为结合与已触发事件相关联的上下文信息利用一个或多个安全规则2614或保护服务器2602来标识对存储设备2606的访问受保护区域尝试并判断适当的响应。例如，已触发事件应对程序2608可以使用安全规则2614来标识对访问诸如受保护的扇区和/或文件之类的存储设备2606的受保护区域的尝试。已触发事件应对程序2608可以使用映射代理2622来帮助标识对访问受保护文件的请求。例如，已触发事件应对程序2608可以向映射代理2622发送对把受保护文件映射到存储设备2606上的相应扇区的请求。映射代理2622可以用对应于受保护文件的扇区响应。已触发事件应对程序2608可以通过标识对对应于文件的访问扇区的尝试来标识对访问受保护文件的尝试。在标识对访问诸如受保护的扇区和/或文件之类的受保护区域的尝试之后，已触发事件应对程序2608可以被配置为查阅安全规则2614以便判断对访问受保护区域的尝试是否得到授权。已触发事件应对程序2608还可以被配置为向O/S下层安全代理2616提供适当动作的判断。例如，已触发事件应对程序2608可以告知O/S下层安全代理2616应当允许还是拒绝已触发事件，是否应当破坏特定的数据或介质表面，或者是否应当加密数据。

[0403] 单独地或与诸如已触发事件应对程序2608或O/S内部安全代理2618之类的组件结合，O/S下层安全代理2616可以被配置为确定访问存储设备2606的典型、可信的方法。例如，通常通过文件I/O驱动程序的调用做出对扇区存储设备2620的写或读。因而，可以由检查被用来做出尝试过程或函数的O/S下层安全代理2616评估对写入受保护扇区的已捕获尝试。可以观察和评估访问扇区时对预期行为的背离，以便指示恶意软件。如果例如O/S下层安全

代理2616判断，通过直接调用中断13而不使用正常的文件I/O函数或驱动程序来做出对受保护扇区的尝试写入，那么，这样的尝试写入是可疑的。

[0404] 备份存储设备2620可以被用来备份和还原存储设备2606上的数据。例如，O/S下层安全代理2616和/或O/S内部安全代理2618可以被配置为备份来自存储设备2606的数据并在各种环境下还原数据。安全规则2614可以指定经授权为要备份的存储设备2606的特定扇区924。当来自存储设备2606的数据要求还原时，可以使用来自备份存储设备2620的相应扇区的数据来写存储设备2606的适当扇区。如果必要，在还原过程期间期间可以使用对存储设备2606的多次写入。在一些实施例中，如果判断数据损坏或以另外方式感染了恶意软件，则可以恢复来自存储设备2606的数据。可以通过扫描存储设备2606的扇区以便检测恶意软件的存在来执行这种判断。在扫描存储设备2606的扇区的同时，可以使用黑名单来标识已知包括恶意软件或与恶意软件相关联的数据的模式。黑名单可以由安全规则2614定义。如果找到了已知与恶意软件相关联的数据的模式，那么，从备份存储设备2620恢复受感染扇区。在一些实施例中，盘映射位图2628可以被用来判断各种受保护文件是否感染了恶意软件。例如，在图23的盘映射位图2301的讨论中可以找到盘映射位图2628的示例实施例的描述。盘映射位图2628可以指定存储设备2606上受保护文件的位置且也可以提供先前所产生的受保护文件的散列值。可以查询盘映射位图2628以便标识受保护文件的位置，可以使用受保护文件的内容来计算的散列，并且可以把所计算的散列与来自盘映射位图2628的先前所产生的散列值进行比较。如果散列值不匹配，则受保护文件可能已经被恶意软件变更，且从恢复备份存储设备2620文件。在一些实施例中，在被用来还原存储设备2606上的数据之前也检查备份存储设备2620以便发现恶意软件。如果备份存储设备2620被感染，则不可以使用来自备份存储设备2620的备份数据和/或可以使用较旧的备份可以被使用，或者可以拒绝对访问存储设备2606的请求。

[0405] 可以由O/S下层安全代理2616把来自备份存储设备2620的数据写到存储设备2606，以便避免使用可能感染了恶意软件的操作系统2612的文件系统机制。然而，可以使用任何其他安全进程来用来自备份存储设备2620的数据把数据还原到存储设备2606。可以维护每一备份的元数据，且其可以包括修订版本号、创建备份的日期和时间以及与该备份相关联的应用2610或其他实体。备份存储设备2620可以被定位为远离存储设备2606，例如在网络上。例如，备份存储设备2620可以与保护服务器2602相关联。如果备份存储设备2620位于网络上，则O/S下层安全代理2616可以使用带外网络连接来访问备份存储设备2620，以便避免使用可能感染了恶意软件的操作系统内核网络设备驱动程序。在一个实施例中，可以使用主动管理技术(AMT)来实现这种带外网络连接，主动管理技术可以允许通过直接地访问电子设备2601的网卡使用HTTPS、iSCSI、NFS或CIFS客户机来访问备份存储设备22620。

[0406] 保护服务器2602可以在网络上操作且可以实现云计算方案。保护服务器2602可以被配置为存储安全规则2614并与诸如O/S下层安全代理2616、O/S内部安全代理2618和/或已触发事件应对程序2608之类的系统900的元素通信，以提供安全规则2614和其他信息。保护服务器2602可以包括备份存储设备2620。备份存储设备2620可以用于存储安全规则2614和/或备份来自存储设备2606的数据。

[0407] 安全规则2614可以由图1的安全规则114、图2的安全规则220、222、图4的安全规则420、422、434、436、438、图5的安全规则518或图7的安全规则707、721、723实现，或者被配置

为实现它们的功能性。可以在下面的图27的讨论中找到安全规则2614的示例实施例的描述。

[0408] 图27是与用于保护存储设备免遭未经授权的访问的系统或方法一起使用的安全规则的示例实施例。安全规则2700可以包括用于指定要求捕获的事件和每一事件的适当响应的任何合适的规则、逻辑、命令、指令、标志或其他机制。例如，安全规则2700可以由来自图26的O/S下层安全代理2616、O/S内部安全代理2618和/或已触发事件应对程序2608用来标识要求捕获的事件并判断每一事件的适当响应。安全规则2700可以要求捕获对访问存储设备的所有请求，或者可以仅要求捕获特定类型的请求，例如读、写和/或执行请求。安全规则2700还可以包括指定存储设备的受保护区域2702的规则，受保护区域2702例如要求保护的存储设备的特定扇区或存储设备上的文件。对于每一受保护区域2702，安全规则2700可以指定诸如操作系统、应用或驱动程序之类的可以授权或者不可以授权访问每一受保护区域2702的请求实体2704。安全规则2700也可以指定经授权为访问受保护区域2702的每一实体2704对受保护区域2702的访问权限2706，例如读2706a、写2706b或执行2706c。

[0409] 某些安全规则2700可以是应用无关的或应用专用的。应用无关的规则不考虑请求访问存储设备的受保护区域2702的应用就适用。应用专用的规则可以取决于发起请求的应用而授权或禁止对受保护区域2702的访问。规则2710a是指定应当拒绝任何实体写入主引导记录的请求的应用无关的规则的示例。规则2710b是允许安全代理写入存储设备上其自己的映像同时禁止任何其他实体写入存储设备上的安全代理映像的应用专用的规则的示例。规则2710c和2710d也是应用专用的规则的示例。规则2710c指定任何实体不可以写入因特网浏览器应用的代码页面。规则2710d指定如果请求是来自因特网浏览器应用则可以允许写入因特网浏览器应用的数据页面的请求，而将会拒绝来自任何其他实体的写入因特网浏览器应用的数据页面的请求。

[0410] 安全规则2700可以由应用或操作系统定义为允许应用和操作系统指定诸如数据或代码页面之类的它们各自的信息的必要保护。安全规则2700也可以由管理员设置，且可以被远程存储，例如存储在来自图26的保护服务器2602上。可以从远程位置检索和/或更新安全规则2700。

[0411] 在一些实施例中，在允许访问存储设备之前，安全规则2700可以要求到诸如来自图26的保护服务器2602之类的保护服务器的网络连接。如果到保护服务器的连接不可用，则安全规则2700可以禁止访问存储设备且可以破坏存储设备的数据或介质表面。例如，安全规则2700可以指定，如果诸如来自图26的O/S下层安全代理2616之类的安全代理不能够连接到保护服务器且已经持续指定的天数，则安全代理可以假设存储设备的安全已经受到危害。在这样的实施例中，即使存储设备上的数据物理上已经受到危害，也保护存储设备上的数据。

[0412] 图28是用于保护电子设备的存储设备免遭未经授权的访问的方法的示例实施例。在步骤2805，可以认证O/S下层安全代理、O/S内部安全代理、已触发事件应对程序和保护服务器的身份和安全。可以使用任何合适的方法执行这样的认证，包括通过定位和检验每一组件的存储器中的映像、密码散列或密钥。直到完成了步骤2805之前，在某些实施例中可以停止其他步骤的操作。

[0413] 在步骤2810，获得安全规则。安全规则可以由O/S下层安全代理、O/S内部安全代理

或已触发事件应对程序本地存储，或者安全规则可以远程存储，例如在保护服务器上。这样的安全规则可以被用来在步骤2815-2860做出判定。在步骤2815，判断是否已经检测到安全威胁。例如，O/S下层安全代理和/或O/S内部安全代理可以标识电子设备上的恶意软件，或者可以标识恶意软件感染电子设备的尝试。如果还没有检测到安全威胁，那么，可以不采取动作。如果已经检测到安全威胁，那么，在步骤2820，指示O/S下层安全代理捕获对存储设备的访问。在一些实施例中，指示O/S下层安全代理在认证进程之后捕获对存储设备的访问，而不考虑是否检测到安全威胁。

[0414] 在步骤2825，捕获到对访问存储设备的请求。可以由在比电子设备上运行的操作系统较低的优先级环执行的软件实施这样的捕获。例如，O/S下层安全代理可以执行捕获功能性。在步骤2830，判断与已捕获请求相关联的存储设备的扇区是否已经感染了恶意软件。可以通过扫描存储设备的扇区以便检测恶意软件的存在来执行这种判断。在扫描存储设备的扇区的同时，可以使用黑名单来标识已知包括恶意软件或与恶意软件相关联的数据的模式。如果找到了数据已知与恶意软件相关联的数据的模式，那么，在步骤2835从备份存储设备恢复受感染扇区。在一些实施例中，在使用备份存储设备来还原存储设备之前也扫描备份存储设备以便发现恶意软件。如果备份存储设备受感染，则可以不适用该备份和/或可以使用较旧的备份，或者可以拒绝对访问存储设备的请求。

[0415] 在步骤2840，判断是否已经请求了访问存储设备的受保护扇区。受保护扇区由安全规则定义。安全规则可以要求保护特定扇区，或者可以要求保护动态定位的特定的文件和/或数据。例如，安全规则可以要求保护主引导记录，主引导记录是静态的且位于存储设备的第一扇区(扇区0)。作为另一示例，安全规则也可以要求保护主文件表、操作系统内核文件、设备驱动程序或反恶意软件软件。这些文件可以具有动态的位置且不总被存储在相同的扇区上。如果动态存储的文件或数据需要保护，则把文件或数据从其在文件系统上的概念存在转换到的文件或数据驻留在其中的存储设备的实际扇区。如果对访问存储设备的请求不涉及受保护扇区，那么，在步骤2850，允许对访问存储设备的请求。如果对访问存储设备的请求涉及受保护扇区，那么，在步骤2845判断对受保护扇区的访问是否得到授权。可以结合安全规则分析与对访问存储设备的尝试请求相关联的上下文信息，以便判断是否授权该请求实体访问受保护扇区。例如，安全规则可以指定，可以或不可以授权操作系统、特定的应用或特定的设备驱动程序访问受保护扇区。安全规则也可以为经授权访问受保护扇区的请求实体指定诸如读、写或执行之类的访问权限。

[0416] 如果对受保护扇区的访问得到授权，那么，在步骤2850，允许对访问存储设备的请求。如果访问受保护扇区未被授权，那么，在步骤2855，拒绝对访问存储设备的请求。在一些实施例中，可以采取其他矫正动作。例如，可以破坏或加密存储设备上的数据，或者可以破坏存储设备的介质表面。如果对访问存储设备的请求没有得到授权，则在步骤2860把对访问存储设备的尝试报告给保护服务器。这样的报告可以包括关于任何关联的恶意软件或可疑行为的信息。

[0417] 根据保护存储设备的要求，可以连续地、周期性地、根据需求或在事件触发时重复来自图28的方法的步骤。

[0418] 图29是用于保护在应用和输入/输出设备之间的写访问的输入/输出路径的系统2900的示例实施例。系统2900可以包括电子设备2904，要保护电子设备2904免遭对电子设

备2904的应用输入/输出(I/O)路径的恶意软件攻击。电子设备2904可以包括操作系统下层安全代理2916、I/O设备2926、应用2910、操作系统2912和驱动程序2911。电子设备2904可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备401、图7的电子设备701和/或其任何组合实现，或者被配置为实现它们的功能性。

[0419] O/S下层安全代理2916可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理2916可以被配置为保护电子设备2904的应用I/O路径免遭恶意软件。I/O设备2926可以全部地或部分地由图2的设备226、图4的显示器424或存储426、图5输入-输出设备502和/或其任何组合实现，或者被配置为实现它们的功能性。应用2910可以全部地或部分地由图1的应用110、图2的应用210、图4的应用410、图7的应用709和/或其任何组合实现，或者被配置为实现它们的功能性。驱动程序2911可以全部地或部分地由图1的驱动程序111、图2的驱动程序211、图4的驱动程序411、图7的驱动程序711和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统2912可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713和/或其任何组合实现，或者被配置为实现它们的功能性。

[0420] 如图29中的箭头所示出的，且如结合下面的图30所描述的，O/S下层安全代理2916可以捕获经由应用I/O路径的数据传送。一旦捕获，O/S下层安全代理2916可以截取结合I/O写访问要从应用2910递送到I/O设备2926的内容。O/S下层安全代理2916可以修改所截取的I/O内容并通过正常的I/O路径(例如，经由操作系统3112和驱动程序3111)传送经修改内容。这样的经修改内容可以包括“经欺骗的”或“哑元”内容，以使得能够截取I/O路径数据的任何恶意软件将截取哑元数据而不是实际的用户数据。当经修改的I/O内容到达I/O设备2926的设备驱动程序时，O/S下层安全代理2916可以截取经修改的I/O内容，并用原始内容来替换它，因而保护I/O传送免遭恶意软件攻击。另外，基于规则(例如，安全规则114、220、222、438、434、436、518、707、721和/或723)，O/S下层安全代理2916可以检测在正常的I/O路径上传送的经修改内容是否受到类似恶意软件的行为(例如，指示经修改内容被嗅探、被钩住和/或以另外方式受到攻击的行为)影响，并且如果检测到类似恶意软件的行为就采取矫正动作。在一些实施例中，O/S下层安全代理2916可以把关于发生类似恶意软件的行为的信息传输给保护服务器202。例如，O/S下层安全代理2916可以把取证信息传输给保护服务器202，取证信息可以辅助保护服务器202标识引起类似恶意软件的行为的恶意软件和/或防止进一步感染电子设备2904和/或其他电子设备。这样的取证信息可以包括但不限于在其中发生该行为的电子设备的身份、标识类似恶意软件的行为的O/S下层安全代理、其中发生类似恶意软件的行为的设备2926和/或应用I/O路径、被O/S下层安全代理放置到I/O路径的经修改内容和/或已截取的经修改数据(指示潜在恶意软件对经修改数据做出的修改)。

[0421] 尽管在应用2910和输入/输出设备2926之间的路径被示出为带有一定数量的元素，但这样的路径可以包括实现在应用2910和输入/输出设备2926之间的输入或输出路径所需要的多个组件。例如，操作系统2912和驱动程序111可以包括多个子组件以便在应用2910和输入/输出设备2926之间传送信息。操作系统2912和驱动程序111和它们的子组件可以被配置为使用系统定义的函数或驱动程序定义的函数来相互调用。O/S下层安全代理

2916可以被配置为捕获沿着在应用2910和输入/输出设备2926之间的输入/输出路径的任何这样的通信或操作。例如，在Windows™环境中，为了把映像放置在设备2926中，应用2910可以被配置为使用BitBlt函数来调用gdi32.dll，gdi32.dll可以被配置为使用NtGDIBitBlt函数来调用ndtll.dll，ndtll.dll可以被配置为使用NtGDIBitBlt函数来调用win32k.sys，win32k.sys可以被配置为调用图形I/O驱动程序，I/O驱动程序可以处理到由设备2926实现的显示器的输入和输出。O/S下层安全代理2916可以被配置为捕获任何这样的函数调用的执行，例如通过捕获包含这样的函数的代码段的存储器位置的执行。存储器位置可以包括，例如，虚拟存储器页面或物理存储器的地址范围。

[0422] O/S下层安全代理2916可以被配置为判断用于沿着在应用2910和设备2926之间的路径传输命令或信息的这样的函数的调用者，并判断它们是否已经被经授权实体执行。例如，驱动程序2911函数可以被恶意进程直接地调用，而不是使用由系统提供的方法(例如操作系统2912中的函数)来访问2911的函数。O/S下层安全代理2916可以被配置为捕获驱动程序2911的函数的执行，并基于该访问起源于其中的存储器地址判断，例如，应用2910被驱动程序2911直接地调用，并且该调用不起源于在操作系统2912内的经授权实体。这样的访问可能已经完成，以便避免在操作系统2912内的安全设置。O/S下层安全代理2916可以被配置判断这样的访问指示恶意软件，并拒绝已尝试访问。

[0423] 此外，O/S下层安全代理2916可以被配置为通过捕获对对应于输入和输出缓冲器的存储器位置的尝试读或写信息来捕获在应用2910和设备2926之间的信息通信。例如，操作系统2912可以把信息写到I/O缓冲器并调用驱动程序2911的函数来检索在缓冲器内要发送给设备2926信息。由于信息大小，可以使用这样的缓冲器，而不是直接作为参数传送信息。因而，O/S下层安全代理2916可以被配置为捕获例如对虚拟存储器页面或I/O缓冲器的物理地址范围的读或写访问。O/S下层安全代理2916可以被配置为判断访问I/O缓冲器的实体的身份，以便判断该实体是否得到授权读或写I/O缓冲器。例如，O/S下层安全代理2916可以被配置为捕获对键盘数据缓冲器的已尝试访问。如果应用2910尝试直接(即在通过操作系统2912的正常调用链的范围之外)从缓冲器读信息，则O/S下层安全代理2916可以被配置为拒绝访问，这是由于对缓冲器中的键盘数据的已尝试直接访问指示诸如键盘记录器之类的恶意软件。在另一示例中，可以保护显示器数据的缓冲器免遭访问以便防止抓屏恶意软件。在又一示例中，可以保护网络输出的缓冲器免遭访问以便防止拒绝服务攻击产生或分组修改。

[0424] 因而，在一些实施例中，O/S下层安全代理2916可以被配置为阻止对没有已知得到授权读或写缓冲器的访问I/O缓冲器的实体的所有访问。在这样的实施例中，可以阻止其恶意软件状态是未知的应用2910或其他实体，即使先前通过扫描实体以便发现恶意签名没有把该实体标识为恶意软件。在其他实施例中，O/S下层安全代理2916可以被配置为使得对缓冲器的访问仅限于在已知的调用链内直接在该缓冲器的下面或上面的驱动程序、接口、应用或其他实体。类似地，O/S下层安全代理2916可以被配置为使得对驱动程序2911或操作系统2912的函数的访问仅限于仅在已知的调用链内直接在该缓冲器的下面或上面的驱动程序、接口、应用或其他实体。通过观察已知安全系统的典型操作以便理解、概况分析(profile)和测试什么实体沿着在应用2910和设备2926之间的路径相互调用，可以定义这样的已知的调用链。可以在O/S下层安全代理2916可访问的安全规则中实现这样的已知安

全操作的表征。可以拒绝在这样的已知链路外对诸如驱动程序2911或操作系统2912之类的驱动程序的组件的任何调用或对I/O缓冲器的调用。

[0425] O/S下层安全代理2916可以捕获在应用2910和设备2926之间的路径内的调用、读取要传送的数据、加密数据、把数据重新插入到路径并允许操作进行。在一个实施例中，设备2926可以包含被配置为解密这样的数据的固件安全代理。这样的固件安全代理和O/S下层安全代理2916可以通信上耦合以便协调这样的加密，和/或均可以具有协调这样的加密的相似安全规则。相反地，O/S下层安全代理2916可以被配置为捕获在路径内的调用、解密来自设备的数据、把数据重新插入到路径并允许操作进行。在另一实施例中，O/S下层安全代理2916可以被配置为捕获进一步在路径下面和上面的调用、读取要传送的数据、解密数据、把数据重新插入到路径并允许操作进行。

[0426] 此外，O/S下层安全代理2916可以被配置为检查要沿着在应用2910和设备2926之间的路径传送的数据并扫描数据以便发现恶意软件的指示。O/S下层安全代理2916可以被配置为控制在路径内的实体之间传送的数据或替换作为参数而传送的数据(例如哑元数据)。

[0427] 图30是用于保护在应用和输入/输出设备之间的写访问的输入/输出路径的方法3000的示例实施例。在步骤3002，O/S下层安全代理可以判断应用I/O路径是否易受恶意软件攻击的攻击。因为在此公开的用于保护应用I/O路径的系统和方法可以消耗显著的处理器、存储器和/或其他资源，期望仅在应用I/O路径特别易受恶意软件攻击感染时采用这样的系统和方法。在应用或操作系统正在执行其中可以传输潜在敏感信息的I/O操作时，应用I/O路径可以易受恶意软件攻击的攻击的。例如，如果应用正在访问银行业务或其他金融网站，这可能在应用I/O路径上暴露敏感信息，例如财务数据、企业人事数据、账户号、用户名、密码、社会保障号和/或电子设备的用户的其他标识数据，则O/S下层安全代理可以判断应用I/O路径易受恶意软件攻击的攻击。

[0428] 在步骤3003，如果判断应用I/O路径易受攻击，则方法3000可以进行到步骤3005。否则，方法3000可以返回到步骤3002，且可以不采取应用I/O路径保护，直到判断应用I/O路径易受攻击的时刻。

[0429] 在步骤3005，O/S下层安全代理可以捕获来自应用的对设备(例如，显示器、盘驱动器、键盘等等)的I/O写访问。例如，如果I/O写访问包括在Windows操作系统中把数据从应用传送到显示设备，则O/S下层安全代理可以捕获应用对位块传送操作(例如，BitBlt)的调用或对显示器I/O函数的库(例如，gdi32.dll，ntdll.dll等等)的调用的执行。I/O写或读访问可以包括一系列或一连串的、对驱动程序和驱动程序的函数并且在驱动程序和驱动程序的函数之间的调用，以便到达最终设备。例如，在Windows™中，应用可以使用BitBlt函数来调用gdi32.dll，gdi32.dll可以使用NtGDIBitBlt函数来调用ntdll.dll，ntdll.dll可以使用NtGDIBitBlt来调用win32k.sys，win32k.sys可以调用图形I/O驱动程序，图形I/O驱动程序可以访问显示设备。

[0430] 在步骤3010，O/S下层安全代理可以截取I/O操作的内容(例如，在显示设备上显示的图像、写到盘驱动器的数据等等)。

[0431] 在步骤3015，O/S下层安全代理可以修改I/O内容。例如，可以用“经欺骗的”或“哑元”内容来修改内容，以使得尝试攻击应用I/O路径的恶意软件仅可以访问经修改内容而不

是构成原始内容的敏感信息。O/S下层安全代理可以以任何合适的方式修改I/O内容。例如，为了替换要显示给显示设备的图像，可以把经修改内容代替原始内容作为参数传送给位块传送操作。在特定的示例中，O/S下层安全代理可以用预先确定的哑元内容替换敏感的文件或电子邮件的文本内容。

[0432] 在步骤3020，O/S下层安全代理可以通过应用I/O路径传送用于正常操作的经修改内容，包括该应用在其上执行的操作系统的操作，以及在操作系统和设备之间的驱动程序。在这一步骤期间，影响应用I/O路径恶意软件会尝试利用I/O内容。然而，所利用的任何数据可以是由O/S下层安全代理插入的经修改的哑元内容，因而保护原始内容免遭利用。

[0433] 在步骤3025，O/S下层安全代理可以在经修改内容到达I/O设备(例如，在设备的通信端口或具有I/O设备的电子设备处)时截取经修改内容。在步骤3030，O/S下层安全代理可以用原始内容替换经修改内容。例如，如果I/O写访问包括在Windows操作系统中把数据从应用传送到显示设备，则可以通过钩住具有显示设备的电子设备的I/O端口、图形I/O驱动程序的存储器挂钩或通过钩住或触发来自图形I/O驱动程序的显示命令的执行来实现替换映像。因此，可以在应用和设备之间带外传输原始内容，保持受到保护免遭尝试在传统应用I/O路径中利用该内容的恶意软件。

[0434] 在步骤3035，O/S下层安全代理可以判断经修改内容是否受到类似恶意软件的行为影响。例如，基于规则(例如，安全规则114、220、222、438、434、436、518、707、721和/或723)，O/S下层安全代理可以判断已截取的经修改内容是否具有指示其受到恶意软件影响的特性(例如，当已修改数据自身在应用I/O路径中被修改时，如果已修改数据通过应用I/O路径)。另外，如果O/S下层安全代理判断经修改内容受类似恶意软件的行为影响，则O/S下层安全代理可以采取矫正动作(例如，移除、隔离和/或以另外方式使恶意软件失效的动作)。另外，在一些实施例中，O/S下层安全代理可以把关于发生类似恶意软件的行为的信息(例如，取证信息)传输给保护服务器。

[0435] 图31是用于保护在应用和输入/输出设备之间的读访问的输入/输出路径的系统3100的示例实施例。系统3100可以包括电子设备3104，要保护电子设备3104免遭对电子设备2104的应用输入/输出(I/O)路径的恶意软件攻击。电子设备3104可以包括操作系统下层安全代理3116、I/O设备3126、应用3110、操作系统3112和驱动程序3111。系统3100可以包括电子设备3104，要保护电子设备3104免遭对电子设备2104的应用输入/输出(I/O)路径的恶意软件攻击。电子设备3104可以包括操作系统下层安全代理3116、I/O设备3126、应用3110、操作系统3112和驱动程序3111。电子设备3104可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备401、图7的电子设备701、图29的电子设备2904和/或其任何组合实现，或者被配置为实现它们的功能性。

[0436] O/S下层安全代理3116可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图29的O/S下层安全代理2916和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理3116可以被配置为保护电子设备3104的应用I/O路径免遭恶意软件。I/O设备3126可以全部地或部分地由图2的设备226、图4的显示器424或存储426、图5的输入-输出设备502、图29的I/O设备2926和/或其任何组合实现，或者被配置为实现它们的功能性。应用3110可以全部地或部分

地由图1的应用110、图2的应用210、图4的应用410、图7的应用709、图29的应用2910和/或其任何组合实现，或者被配置为实现它们的功能性。驱动程序3111可以全部地或部分地由图1的驱动程序111、图2的驱动程序211、图4的驱动程序411、图7的驱动程序711、图29的驱动程序2911和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统3112可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图29的操作系统2912和/或其任何组合实现，或者被配置为实现它们的功能性。

[0437] O/S下层安全代理3116可以被配置为保护电子设备3104的应用I/O路径免遭恶意软件。如图31中的箭头所示出的，以及如结合下面的图32所描述的，O/S下层安全代理3116可以经由应用I/O路径捕获数据的传送。一旦捕获，O/S下层安全代理3116可以截取结合I/O读访问要从I/O设备3126递送到应用3110的内容。O/S下层安全代理3116可以修改已截取的I/O内容并通过正常的I/O路径传送经修改内容(例如，经由操作系统3112和驱动程序3111)。这样的经修改内容可以包括“经欺骗的”或“哑元”内容，以使得能够截取I/O路径数据的任何恶意软件将截取哑元数据而不是实际的用户数据。当经修改的I/O内容到达I/O设备3126的设备驱动程序时，O/S下层安全代理3116可以截取经修改的I/O内容，并用原始内容来替换它，因而保护I/O传送免遭恶意软件攻击。另外，基于规则(例如、安全规则114、220、222、438、434、436、518、707、721和/或723)，O/S下层安全代理3116可以检测在正常的I/O路径上传送的经修改内容是否受到类似恶意软件的行为(例如，指示经修改内容被嗅探、被钩住和/或以另外方式受到攻击的行为)影响，并且如果检测到类似恶意软件的行为就采取矫正动作。在一些实施例中，O/S下层安全代理3116把关于发生类似恶意软件的行为的信息传输给保护服务器202。例如，O/S下层安全代理3116可以把取证信息传输给保护服务器202，取证信息可以辅助保护服务器202标识引起类似恶意软件的行为的恶意软件和/或防止进一步感染电子设备3104和/或其他电子设备。这样的取证信息可以包括但不限于在其中发生该行为的电子设备的身份、标识类似恶意软件的行为的O/S下层安全代理、其中发生类似恶意软件的行为的设备3126和/或应用I/O路径、被O/S下层安全代理放置到I/O路径的经修改内容和/或已截取的经修改数据(指示潜在恶意软件对经修改数据做出的修改)。

[0438] 图32是用于保护在应用和输入/输出设备之间的读访问的输入/输出路径的方法3200的示例实施例。在步骤3202，O/S下层安全代理可以判断应用I/O路径是否易受恶意软件攻击的攻击。步骤3202可以类似于方法3000的步骤3002。在步骤3203，如果判断应用I/O路径易受攻击，则方法3200可以进行到步骤3205。否则，方法3200可以返回到步骤3202，且可以不采取应用I/O路径保护，直到判断应用I/O路径易受攻击的时刻。步骤3203可以类似于方法3000的步骤3003。

[0439] 在步骤3205，O/S下层安全代理可以捕获来自设备的对应用(例如，显示器、盘驱动器、键盘等等)的I/O读访问。在步骤3210，O/S下层安全代理可以截取I/O操作的内容(例如，从键盘接收到的击键、要从盘驱动器读取的数据等等)。

[0440] 在步骤3215，O/S下层安全代理可以修改I/O内容。例如，用“经欺骗的”或“哑元”内容来修改内容，以使得尝试攻击应用I/O路径的恶意软件仅可以访问经修改内容而不是构成原始内容的敏感信息。O/S下层安全代理可以以任何合适的方式修改I/O内容。

[0441] 在步骤3220，O/S下层安全代理可以通过应用I/O路径传送用于正常操作的经修改

内容，包括该应用在其上执行的操作系统的操作，以及在操作系统和设备之间的驱动程序。在这一步骤期间，影响应用I/O路径恶意软件会尝试利用I/O内容。然而，所利用的任何数据可以是由O/S下层安全代理插入的经修改的哑元内容，因而保护原始内容免遭利用。

[0442] 在步骤3225，O/S下层安全代理可以在经修改内容到达应用时截取经修改内容。在步骤3230，O/S下层安全代理可以用原始内容替换经修改内容。因此，可以在应用和设备之间带外传输原始内容，保持受到保护免遭尝试在传统应用I/O路径中利用该内容的恶意软件。

[0443] 在步骤3235，O/S下层安全代理可以判断经修改内容是否受到类似恶意软件的行为影响(例如，当已修改数据自身在应用I/O路径中被修改时，如果已修改数据通过应用I/O路径)。例如，基于规则(例如、安全规则114、220、222、438、434、436、518、707、721和/或723)，O/S下层安全代理可以判断已截取的经修改内容是否具有指示其受到恶意软件影响的特性。另外，如果O/S下层安全代理判断经修改内容受类似恶意软件的行为影响，O/S下层安全代理可以采取矫正动作(例如，例如，移除、隔离和/或以另外方式使恶意软件失效的动作)。另外，在一些实施例中，O/S下层安全代理可以把关于发生类似恶意软件的行为的信息(例如，取证信息)传输给保护服务器。

[0444] 另外，在一些实施例中，在应用I/O路径上传送的哑元数据(例如，在方法3000的步骤3015和3020和/或方法3200的步骤3215和3220中)可以被用来跟踪电子设备2904和/或电子设备3104上恶意软件的存在。例如，当在第一设备的应用在网络(例如，由第二电子设备主控的银行业务或其他金融网站)上把敏感信息传输给第二电子设备，O/S下层安全代理可以把哑元信息插入到I/O路径中，该哑元信息可以欺骗对第二电子设备的访问(例如，可以把“假的”用户名和密码提供给银行业务网站的哑元信息)。第二电子设备可以包括其自己的安全代理，使得在以这种方式欺骗第二设备时，第二电子设备的安全代理可以被配置为跟踪在访问期间采取的动作(例如，在欺骗访问期间在银行业务网站采取的动作，例如改变概况信息或其他动作)，以判断是否已经发生了类似恶意软件的行为。如果第二电子设备处的安全代理判断已经发生了类似恶意软件的行为，那么，第二电子设备可以采取矫正动作。例如，第二电子设备处的安全代理可以传输适当的消息(例如，传输到通信上耦合到第二电子设备的保护服务器102)以便指示类似恶意软件的行为的存在。这样的消息可以包括取证迹象，取证迹象包括，例如，类似恶意软件的行为和/或第一电子设备的身份(例如，因特网协议地址或其他标识信息)的描述。

[0445] 图33是用于检测和修复电子设备3304上的隐藏进程的系统3300的示例实施例。O/S内部安全代理3318和/或O/S下层安全代理3316可以在电子设备3304上操作以便检测和修复恶意感染，例如被配置为隐藏电子设备3304上正在运行的进程的操作恶意软件。电子设备3304可以包括被耦合到存储器3308的处理器3306、操作系统3312和一个或多个进程3373。电子设备3304可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701和/或其任何组合实现，或者被配置为实现它们的功能性。处理器3306可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702和/或其任何组合实现，或者被配置为实现它们的功能性。存储器3308可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统3312可以全部地或部分地由图1的操作系统112、图2的操作系

统212、图4的操作系统412、图7的操作系统713和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理3318可以全部地或部分地由图2的O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理719和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理3316可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708和/或其任何组合实现，或者被配置为实现它们的功能性。

[0446] 进程3373可以被配置为在电子设备3304上操作。在电子设备3304上操作的一个或多个进程3373可以是与恶意软件相关联的恶意进程。电子设备3304上的恶意软件可以操作为掩饰一个或多个恶意进程3373的存在，以便避免反恶意软件软件的检测。例如，操作系统3312可以包括操作系统内核存储器3380。操作系统内核存储器3380可以包括用于跟踪电子设备3304上的进程的执行的一个或多个机制。在一个示例中，这样的机制可以包括活动进程列表3384。活动进程列表3384可以以数据结构、记录、文件或用于跟踪在电子设备3304上操作的进程的任何其他合适的方法实现。例如，如果进程3373b是与恶意软件相关联的恶意进程，则电子设备3304上的恶意软件可以修改活动进程列表3384以便移除对进程3373b的引用。因而，在判断哪些进程在电子设备上3304活动运行且应检查以便发现恶意软件时，在电子设备3304上运行的安全软件不把进程3373b识别为活动进程以供检查。

[0447] 操作系统3312可以包括就绪队列3322。就绪队列3322可以包括一个或多个合适的数据结构(例如，数组、表、列表等等)陈述在电子设备3304上操作的活动线程。活动进程3373可以包括一个或多个个体线程。线程可以被认为是在活动进程3373内的、可以与从活动进程3373的其他线程分离地独立调度以便由电子设备3304执行的处理单元(例如，一个或多个指令)。作为Windows™操作系统中的就绪队列3322的说明性示例，就绪队列3322可以由被称为KiDispatcherReadyListHead变量实现。就绪队列3322也可以包括关于活动线程的各种元数据，例如，包括该线程的进程的标识符，这样的进程的映像名称、开始地址、用户模式地址、设备对象和/或其他合适的信息。在Windows™操作系统中，这样的活动进程信息可以被包括在与该线程相关联的执行线程(“ETHREAD”)数据结构中。

[0448] 在系统3300的操作期间，O/S内部安全代理3318和/或O/S下层安全代理3316可以引起安全设备驱动程序3370在操作系统3312上执行。安全设备驱动程序3370可以以驱动程序、模块、可执行程序、DLL或用于提供内核模式设备驱动程序服务的任何其他合适的机制实现。安全设备驱动程序3370可以被配置为调用操作系统3312的各种部分来列举在电子设备3304上运行的进程。例如，安全设备驱动程序3370可以被配置为检查内核存储器3380或活动进程列表3384。安全设备驱动程序3370可以被配置为传输安全设备驱动程序3370可以检测的活动进程的第一列表3373(例如，活动进程列表3384)。安全设备驱动程序3370可以被配置为把活动进程列表3384传输给O/S内部安全代理3318和/或O/S下层安全代理3316。在一个实施例中，安全设备驱动程序3370可以被配置为经由超级调用把与给定的已检测进程相关联的执行进程(“EPROCESS”)结构传送给O/S下层安全代理3316。因为安全设备驱动程序3370在与操作系统相同的特权的执行环或比操作系统较少特权的执行环运行，由安全设备驱动程序3370列举的活动进程可以限于出现在活动进程列表3384上的那些活动进程，这意味着已经修改活动进程列表3384以便移除对它们自身的引用的恶意进程不会被安全

设备驱动程序3370列举。在Windows™操作系统中，安全设备驱动程序3370可以被配置为使用函数ZwQuerySystemInformation来从操作系统请求进程列表，标识要判断的SystemProcessInformation。O/S下层安全代理3316可以被配置为也使用这样的函数，且在执行这样的动作时更加安全。安全设备驱动程序3370可以把所列举的进程放置在活动进程的第一列表3385。在某些实施例中，第一列表3385可以基本上等效于活动进程列表3384。在其他实施例中，可以不创建分离的第一列表3385，且安全设备驱动程序3370可以改为使用活动进程列表3384代替这样的第一列表3385。

[0449] 相反，O/S内部安全代理3318和/或O/S下层安全代理3316可以在与操作系统3312相同特权的执行环或比操作系统3312更多特权的执行环运行，且因此，可以列举在电子设备3304上执行的个体线程。至少基于这样的已列举线程，O/S内部安全代理3318和/或O/S下层安全代理3316可以确定在电子设备3304上执行的所有活动进程3373，包括已经从活动进程列表3384移除对它们自身的引用的恶意进程。例如，在某些实施例中，O/S内部安全代理3318和/或O/S下层安全代理3316可以扫描就绪队列3322并列举就绪队列3322中的所有线程，把线程放置在列表中。对于每一线程，O/S内部安全代理3318和/或O/S下层安全代理3316可以定位持有该线程的进程和关于这样的进程的其他信息(例如，通过引用元数据，例如与该线程相关联的ETHREAD信息)，因而允许O/S内部安全代理3318列举活动进程3373的第二列表3386，包括已经从活动进程列表3384移除对它们自身的引用的恶意进程。

[0450] 为了进一步阐释特定的实施例，与线程相关联的ETHREAD数据结构可以包括多个元数据字段，包括ThreadsProcess字段、StartAddress字段、DeviceToVerify字段、Win32StartAddress字段和ThreadListEntry字段。通过分析ThreadsProcess字段，O/S内部安全代理3318和/或O/S下层安全代理3316可以标识持有线程的进程，从中可以确定该进程的进程标识符和映像名称。从StartAddress和Win32StartAddress，O/S内部安全代理3318和/或O/S下层安全代理3316可以标识存储器中的哪些代码正在执行进程，因而允许如果发现持有该线程的进程是可疑的则进一步标识可疑驱动程序、应用和/或其他程序。基于DeviceToVerify，O/S内部安全代理3318和/或O/S下层安全代理3316可以判断设备对象是否与线程相关联，且因而，如果持有该线程的进程被发现是可疑的则标识驱动程序对象和恶意软件驱动程序。ThreadListEntry可以辅助在相同的进程内列举线程。

[0451] 然后，O/S内部安全代理3318可以比较活动进程3373的第一列表3385和活动进程3373的第二列表3386，并把出现在第二列表3386中且不出现在第一列表3385的活动进程3373标识标识为可疑进程。这样可疑进程的迹象可以是对在电子设备3304上运行的防病毒或反恶意软件软件以及操作系统3312隐藏的恶意软件的迹象。

[0452] 在其他实施例中，至少基于已标识线程，O/S下层安全代理3316可以扫描就绪队列3322以便列举线程并列举活动进程3373的第二列表3386(例如，通过引用元数据，例如与线程相关联的ETHREAD信息)。在这样的实施例中，O/S下层安全代理3316可以从O/S内部安全代理3318接收由安全设备驱动程序3370产生的活动进程的第一列表3385，或者可以通过直接从存储器读取访问活动进程3373的第一列表3385。然后，O/S下层安全代理3316可以比较活动进程3373的第一列表3385和活动进程3373的第二列表3386，并把出现在第二列表3386中且不出现在第一列表3385中的活动进程3373标识可疑进程。这样可疑进程可以对在电子设备3304上运行的防病毒或反恶意软件软件以及操作系统3312隐藏。这样可疑进程的迹象

可以是对在电子设备3304上运行的防病毒或反恶意软件软件以及操作系统3312隐藏的恶意软件的迹象。

[0453] 如果O/S内部安全代理3318和/或O/S下层安全代理3316判断存在在电子设备3304上运行的隐藏进程的迹象，O/S内部安全代理3318和/或O/S下层安全代理3316可以被配置为扫描操作系统3312、操作系统内核存储器3380或电子设备3304的其他元素，以便判断是否已经做出与这样的进程相关的任何修改。例如，O/S内部安全代理3318和/或O/S下层安全代理3316可以被配置为扫描以便发现已知由恶意软件实施的任何存储器修改。在一些实施例中，O/S内部安全代理3318或O/S下层安全代理3316可以被配置为扫描操作系统代码部分3382以及活动进程列表3384。在这些和其他实施例中，O/S内部安全代理3318和/或O/S下层安全代理3316可以利用与可疑进程相关联的线程的线程元数据(例如，ETHREAD信息)来确定要扫描以便发现修改的电子设备3304的元素和/或其部分。

[0454] 如果找到了恶意修改，则O/S内部安全代理3318或O/S下层安全代理3316可以采取矫正动作。例如，O/S内部安全代理3318或O/S下层安全代理3316可以被配置为修复操作系统内核存储器3380中找到的任何恶意修改。作为另一示例，O/S内部安全代理3318或O/S下层安全代理3316可以被配置为移除通过其对操作系统内核存储器380中的存储器修改的观察所判断的任何检测到的rootkit感染。作为进一步的示例，O/S内部安全代理3318或O/S下层安全代理3316可以被配置为修复对任何内部数据结构或代码段的任何感染。在这些和其他实施例中，O/S内部安全代理3318和/或O/S下层安全代理3316可以利用与可疑进程相关联的线程的线程元数据(例如，ETHREAD信息)来判断要采取的矫正动作(例如，这样的元数据可以标识恶意软件进程的具体存储器位置、负责可疑行为的驱动程序等等)。O/S内部安全代理3318或O/S下层安全代理3316可以被配置为对在由安全设备驱动程序3370确定的进程的第一列表3385和从就绪队列3322中出现的线程元数据的分析确定的进程的第二列表3386之间出现的每一矛盾重复扫描过程以便发现隐藏进程的修改。

[0455] 图34是用于检测和修复电子设备上的隐藏进程的方法的示例实施例。在步骤3405，一个或多个安全代理(例如，O/S内部安全代理3318和/或O/S下层安全代理3316)可以引起安全设备驱动程序在电子设备的操作系统上执行。在步骤3410，安全设备驱动程序可以经由对电子设备的操作系统的标准系统调用列举在电子设备上运行的活动进程。安全设备驱动程序可以把所列举的进程放置在活动进程的第一列表。因为安全设备驱动程序可以在与操作系统相同特权的执行环运行或在比操作系统较少特权的执行环运行，安全设备驱动程序所列举的活动进程可以限于出现在操作系统的活动进程列表上的那些活动进程，这意味着已经修改活动进程列表以便移除对它们自身的引用的恶意进程不会被安全设备驱动程序列举。

[0456] 在步骤3415，一个或多个安全代理可以扫描线程就绪队列，且至少基于这样的扫描，列举在电子设备上执行的个体线程并把它们放置字在线程列表中。在步骤3420，至少基于与该线程相关联的元数据(例如，与线程相关联的ETHREAD信息或陈述持有线程的进程的其他元数据)，一个或多个安全代理可以定位持有线程的进程并产生活动进程的第二列表。第二列表可以包括已经从活动进程列表移除对它们自身的引用的恶意进程。

[0457] 在步骤3425，一个或多个安全代理可以比较活动进程的第一列表和活动进程的第二列表。在步骤3430，一个或多个安全代理可以把出现在第二列表且不出现在第一列表的

活动进程标识为可疑进程。这样可疑进程的迹象的可以是对在电子设备上运行的防病毒或反恶意软件软件和/或在电子设备上执行的操作系统隐藏的恶意软件的迹象。

[0458] 在步骤3435，如果一个或多个安全代理判断存在在电子设备上运行的隐藏进程的迹象，一个或多个安全代理可以判断是否已经由可疑进程做出对电子设备的部分的修改。为了判断是否已经做出修改，一个或多个安全代理可以扫描操作系统和/或操作系统内核存储器，以便判断是否已经做出与这样的进程相关的任何修改。例如，一个或多个安全代理可以扫描以便发现由恶意软件实施的任何存储器修改，和/或可以扫描操作系统内核存储器的操作系统代码部分和/或活动进程列表。

[0459] 在步骤3440，如果已经找到修改，则一个或多个安全代理可以采取矫正动作。例如，一个或多个安全代理可以修复在操作系统内核存储器中找到的任何恶意修改。作为另一示例，一个或多个安全代理可以移除通过其对操作系统内核存储器中的存储器修改的观察确定的任何检测到的rootkit感染。作为进一步的示例，一个或多个安全代理可以修复对任何内部数据结构或代码段的任何感染。可以对每一已标识的可疑进程重复的方法3400的各部分。因此，一个或多个安全代理可以对在由安全设备驱动程序3370确定的进程的第一列表3385和从就绪队列3322中出现的线程元数据的分析确定的进程的第二列表3386之间出现的每一矛盾重复扫描过程以便发现隐藏进程的修改。

[0460] 有利地，以上所描述的方法和系统可以提供对rootkit和/或其他恶意软件的标识，而不要求对操作系统内核的任何函数的挂钩或捕获。

[0461] 图35是用于检测和修复电子设备3504上的隐藏进程的系统3500的示例实施例。O/S下层安全代理3516、安全设备驱动程序3570和安全动态链接库(DLL)3572可以在电子设备3504上操作以便检测和修复恶意感染，例如被配置为隐藏在电子设备3504上正在运行的进程的操作的恶意软件。电子设备3504可以包括被耦合到存储器3508的处理器3506、操作系统3512、安全DLL 3572、O/S下层安全代理3516、虚拟机控制结构3552(“VMCS”)一个或多个进程3573(例如，进程3573a、3573b和3573c)、与这样的进程相关联的地址空间3587(例如，地址空间3587a、3587b和3587c)和一个或多个系统资源，例如类似于CR3控制寄存器3560的控制寄存器。处理器寄存器3530可以包括诸如例如CR3寄存器3560或任何其他寄存器3568的此类寄存器。尽管作为处理器寄存器3530的示例给出CR3，但可以使用任何合适的控制寄存器。CR3寄存器3560可以是被配置为控制或改变电子设备3504上的CPU的一般行为的处理器寄存器。CR3寄存器3560可以被配置为允许诸如在电子设备3504上运行的处理器3506之类的处理器把虚拟存储器地址转换成物理存储器地址。CR3寄存器3560可以被配置为定位用于当前所请求的任务的页面目录和页面表，当前所请求的任务例如驻留在栈区中且被选择为供O/S调度程序操作的任务。可以以任何合适的虚拟定址控制寄存器实现CR3寄存器3560。根据电子设备3504的特定的设计或实现，其他寄存器268可以出现在处理器寄存器3530。处理器寄存器3530可以与处理器3506或电子设备3504的另一处理器相关联。

[0462] 电子设备3504可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图33d的电子设备3304和/或其任何组合实现，或者被配置为实现它们的功能性。处理器3506可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图33的处理器3306和/或其任何组合实现，或者被配置为实现它们的功能性。存储器3508可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存

储器703、图33的存储器3308和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统3512可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图33的操作系统3312和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理3516可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图33的O/S下层安全代理3316和/或其任何组合实现，或者被配置为实现它们的功能性。

[0463] 电子设备3504可以包括虚拟机控制结构1152。在一个实施例中，虚拟机控制结构1152可以驻留在O/S下层安全代理3516中。在另一实施例中，虚拟机控制结构1152可以通信上耦合到O/S下层安全代理3516。在这样的实施例中，虚拟机控制结构1152的功能性中的一些或全部可以由O/S下层安全代理3516实现。此外，在这样的实施例中，O/S下层安全代理3516的功能性中的一些或全部可以由虚拟机控制结构1152实现。虚拟机控制结构1152可以完全地或部分地由图1的O/S下层捕获代理104、图2的VMCS、图4的固件安全代理440、442或PC固件安全代理444、图5的固件安全代理516、或图7的微代码安全代理706实现。虚拟机控制结构1152可以在数据结构、记录、文件、模块或用于捕获对诸如处理器寄存器3530或其他资源之类的资源的请求的任何其他合适的实体实现。在一个实施例中，例如其中系统3500可以完全地或部分地由图2的系统200实现的实施例，虚拟机控制结构1152和O/S下层安全代理3516可以被配置为虚拟化对诸如处理器寄存器3530或任何其他合适的系统资源之类的电子设备3504的系统资源的访问。

[0464] 虚拟机控制结构1152可以包括用于捕获操作系统3512对诸如处理器寄存器3530之类的系统资源所请求的操作的一个或多个标志1154。标志1154可以包括捕获的标志，例如，进程上下文切换标志3556和/或读函数标志3558。标志1154可以包括适用于捕获对诸如处理器寄存器3530之类的系统资源的访问的任何标志。O/S下层安全代理3516可以被配置为设置虚拟机控制结构1152的哪些标志1154将被用来捕获对系统资源的访问。可以被虚拟机控制结构1152和O/S下层安全代理3516捕获和/或守护的系统资源可以包括但不限于处理器寄存器3530。

[0465] 进程3573可以被配置为在电子设备3504上操作。电子设备3504上的一个或多个进程3573操作可以是与恶意软件相关联的恶意进程。电子设备3504上的恶意软件可以操作为掩饰进程3573的一个或多个恶意进程的存在以便避免被反恶意软件软件检测。例如，操作系统3512可以包括操作系统内核存储器3580。操作系统内核存储器3580可以包括用于跟踪电子设备3504上的进程的执行的一个或多个机制。在一个示例中，这样的机制可以包括活动进程列表3584。活动进程列表3584可以以数据结构、记录、文件或用于跟踪在电子设备3504上操作的进程的任何其他合适的方法实现。例如，如果进程3573b是与恶意软件相关联的恶意进程，则电子设备3504上的恶意软件可以修改活动进程列表3584以便移除对进程3573b的引用。因而，在判断哪些进程在电子设备上3504活动运行且应检查以便发现恶意软件时，在电子设备3504上运行的安全软件将不把进程3573b识别为活动进程以供检查。

[0466] 进程3573或在电子设备3504上操作的其他实体，在使用虚拟存储器时，作为正常操作的一部分，可以要求使用与进程3573中的一个相关联的进程上下文切换。为了促进虚拟存储器的使用，操作系统3512可以被配置为实施进程上下文切换、读取或附接到给定进

程。这样的动作可以要求操作系统3512尝试访问系统资源，包括诸如CR3寄存器3560之类的控制寄存器。操作系统3512可以产生以命令“move value,CR3(移动值,CR3)”的形式的对CR3寄存器3560的读取。操作系统3512可以被配置为以命令“move CR3,value(移动CR3,值)”的形式尝试改变CR3寄存器3560的值。

[0467] 虚拟机控制结构1152可以被配置为截取操作系统3512对访问包括寄存器3530的电子设备3504的系统资源的尝试。虚拟机控制结构1152可以被配置为尝试捕获操作系统3512访问电子设备3504的系统资源的某些尝试命令。虚拟机控制结构1152可以被配置为使用标志来截取操作系统3512的命令。在一个实施例中，虚拟机控制结构1152可以包括标志3556-3558以便截取进程上下文切换和对CR3寄存器3560的读命令。O/S下层安全代理3516可以被配置为在虚拟机控制结构1152中设置这样的标志3556-3558。虚拟机控制结构1152可以被配置为产生VM退出，这是对诸如与CR3寄存器3560相关联的读取或进程上下文切换命令之类的已标记操作的截取所产生的事件。在一个实施例中，虚拟机控制结构1152可以被配置为为对与虚拟存储器相关联的控制寄存器的任何已尝试访问产生VM退出。每当在电子设备3504上运行的进程3573中的一个尝试实施进程上下文切换，或读取与进程相关联的进程空间，虚拟机控制结构1152可以被配置为产生VM退出并把关于所尝试的命令信息传递给O/S下层安全代理3516。为了阐释，O/S下层安全代理3516可以被配置为把对CR3寄存器3560(或另一寄存器3568)的所有这样的动作记录在寄存器改变3576中。寄存器改变3576可以以文件、结构、数据结构、记录或用于把改变的历史存储到CR3寄存器3560或另一寄存器3568的任何其他合适的机制实现。通过记录对CR3寄存器3560的所有访问，O/S下层安全代理3516因而可以拥有已经在电子设备3504中尝试进程上下文切换的所有进程3573的记录。作为寄存器改变3576的这样的改变记录可以被配置为充当在电子设备3504上运行的进程的记录。O/S下层安全代理3516可以被配置为从寄存器改变3576列表确定正在运行的进程3586，包括在电子设备3504上运行的所有进程。

[0468] 通过截取对CR3寄存器的访问，O/S下层安全代理3516可以通过拒绝在进出执行时交换进程的能力来冻结执行。通过冻结执行，O/S下层安全代理3516可以被配置为收集正在运行的进程的列表而无需进入与恶意软件的竞态条件，恶意软件可以工作为避免或破坏矫正措施或检测措施。

[0469] 替代地，O/S下层安全代理3516可以被配置为通过监视用于上下文交换的操作系统函数(例如，Windows™函数SwapContext)确定运行进程3586的列表。对操作系统上下文交换函数的监视可以经由与函数调用的代码段相关联的存储器的处理器级别监视或通过观察指向这样的代码段执行指令指针(“EIP”)执行。例如，可以在安全规则中映射和描述SwapContext驻留在其中的物理存储器位置或虚拟存储器位置。SwapContext驻留在其中的物理地址或虚拟存储器地址的标志可以被设置为以使得可以捕获对存储器的任何已尝试执行。例如，如果O/S下层安全代理3516完全地或部分地由图2的SVMM安全代理217实现，那么，O/S下层安全代理3516可以设置VMCS 1152以便捕获对SwapContext驻留在其中的虚拟存储器页面的已尝试执行。在另一示例中，如果O/S下层安全代理3516完全地或部分地由图7的微代码安全代理708实现，那么，O/S下层安全代理3516可以设置VMCS 1152以便捕获对开始SwapContext函数的代码部分的物理存储器地址的已尝试执行。

[0470] 操作系统3512可以包括安全设备驱动程序3570。安全设备驱动程序3570可以以驱

动程序、模块、可执行程序、DLL或用于提供内核模式设备驱动程序服务的任何其他合适的机制实现。安全设备驱动程序3570可以被配置为调用操作系统3512的各种部分以便列举在电子设备3504上运行的运行进程。例如，安全设备驱动程序3570可以被配置为检查内核存储器3580或活动进程列表3584。安全设备驱动程序3570可以被配置为传输安全设备驱动程序3570可以检测的运行进程列表3580。安全设备驱动程序3570可以被配置为把运行进程列表3580传输给O/S下层安全代理3516。在一个实施例中，安全设备驱动程序3570可以被配置为经由超级调用把与给定的已检测进程相关联的EPROCESS结构传送给O/S下层安全代理3516。一旦接收到安全设备驱动程序3570所检测的活动进程的EPROCESS结构，O/S下层安全代理3516可以被配置为计算与从安全设备驱动程序3570接收到的每一个这样的EPROCESS结构相关联的CR3寄存器3560值(或其他寄存器3568值)。O/S下层安全代理3516可以被配置为把从安全设备驱动程序3570接收到的运行进程列表3580与O/S下层安全代理3516从寄存器改变3576确定的运行进程列表3586进行比较。O/S下层安全代理3516可以被配置为判断，相比于O/S下层安全代理3516已经编译的寄存器改变3576，是否丢失了从来自安全设备驱动程序3570的运行进程列表3580导出的任何CR3值。这样的矛盾可以指示隐藏进程。这样的进程可以对安全设备驱动程序3570、在电子设备3504上运行的任何防病毒或反恶意软件软件以及操作系统3512隐藏。然而，这样的进程的迹象对O/S下层安全代理3516来说是可见的，这是因为这样的隐藏进程尝试例如进程上下文切换或进程地址空间的读取。隐藏进程的这样的迹象可以是经由内核根设备驱动程序的内核模式感染的迹象，因为内核模式安全设备驱动程序3570不能检测隐藏进程。

[0471] O/S下层安全代理3516可以判断，丢失的进程不是隐藏进程，而是在编译列表编译的时间之间的正常执行过程中被删除的进程。为了帮助防止把这样的线程错误标识为隐藏进程，O/S下层安全代理3516可以被配置为监视用于创建和删除进程的函数的执行。这样的函数可以包括例如pspProcessCreate或pspTerminateProcess。O/S下层安全代理3516可以被配置为做出其列举动作和观察创建或删除函数的时间戳记录，以使得如果进程丢失，则可以判断在进程被标识为丢失之前是否用删除函数删除了该进程。

[0472] 如果O/S下层安全代理3516判断存在在电子设备3504上运行的隐藏进程的迹象，则O/S下层安全代理3516可以被配置为扫描操作系统3512和操作系统内核存储器3580，以便判断是否已经做出与这样的进程相关的任何修改。O/S下层安全代理3516可以被配置为扫描以便发现已知由恶意软件实施的任何存储器修改。在一些实施例中，O/S下层安全代理3516可以被配置为扫描操作系统代码部分3582以及活动进程列表3584。O/S下层安全代理3516可以被配置为修复在操作系统内核存储器3580中找到的任何恶意修改。O/S下层安全代理3516可以被配置为移除移除通过其对操作系统内核存储器3580中的存储器修改的观察确定的任何检测到的rootkit感染。O/S下层安全代理3516可以被配置为修复对任何内部数据结构或代码段的任何感染。O/S下层安全代理3516可以被配置为对在由O/S下层安全代理3516和安全设备驱动程序3570确定的进程之间出现的每一矛盾重复扫描过程以便发现隐藏进程的存储器修改。安全设备驱动程序3570可以被配置为从O/S下层安全代理3516接收最终的进程列表，例如运行进程列表3586。安全设备驱动程序3570可以被配置为通过对O/S下层安全代理3516的超级调用访问运行进程列表3586。

[0473] 安全DLL 3572可以被配置为在电子设备3504上操作。安全DLL 3572可以以动态链

接库(DLL)、共享库、可执行程序或用于如下所示执行其函数的任何其他合适的机制实现。安全设备驱动程序3570可以被配置为把安全DLL3572或对安全DLL 3572的引用注入到在电子设备3504上运行的每一进程的地址空间，例如进程地址空间3587。进程地址空间3587中的每一个因而可以包含指向安全DLL 3572的指针。安全DLL 3572可以被配置为列举在电子设备3504上运行的所有用户模式进程。安全DLL 3572可以被配置为使用任何合适的技术来列举用户模式进程。例如，安全DLL 3572可以被配置为使用Windows™操作系统中的NtQuerySystemlnformation函数，并使用ProcessFirst和ProcessNext函数导航进程。这样的函数也可以由O/S下层安全代理执行。安全DLL 3572可以被配置为把其结果编译到运行进程列表3588中。安全DLL 3572可以被配置为把运行进程列表3588发送给安全设备驱动程序3570。安全DLL 3572可以被配置为经由受保护输入和输出调用发送这样的运行进程列表3588。安全设备驱动程序3570可以被配置为把所接收的运行进程列表3588与它从O/S下层安全代理3516接收运行进程列表3586的列表进行比较。安全设备驱动程序3570可以被配置为判断这样的进程列表中的任何差异可以包括用户模式rootkit攻击。安全设备驱动程序3570或安全DLL 3572可以被配置为检查与进程地址空间3587相关联的存储器中的进程代码和数据部分，以便判断是否已经做出任何存储器修改。安全设备驱动程序3570可以被配置为修复在进程地址空间3587中做出的任何存储器修改。安全设备驱动程序3570可以被配置为对在来自安全DLL 3572的运行进程列表3588和来自O/S下层安全代理3516的运行进程列表3586之间出现的每一矛盾重复扫描进程地址空间3587、检测存储器修改、和修复这样的存储器修改的过程。

[0474] 在操作中，进程3573可以在电子设备3504上操作。进程3573中的一个或多个可以被隐藏。例如，进程3573b可以与恶意软件相关联，且可以对在电子设备3504上运行的防病毒或反恶意软件软件隐藏以便掩饰其恶意操作。进程3573可以通过操作系统3512访问电子设备3504的系统资源。为了访问存储器的不同部分，或为了由处理器3508执行，进程3573可以要求访问电子设备3504的控制寄存器。这样的访问可以包括引起进程上下文切换或进程地址空间的读取。这样的要求可以由操作系统3512应对，其中，操作系统3512访问诸如CR3寄存器3560之类的寄存器。虚拟机控制结构1152可以截取这样的请求并为该请求产生VM退出。虚拟机控制结构1152可以把与这样的尝试相关联的信息提供给O/S下层安全代理3516。O/S下层安全代理3516可以设置标志，例如在虚拟机控制结构1152上捕获指令“move CR3, value”3556或“move value,CR3”3558的那些标志。O/S下层安全代理3516可以记录对CR3寄存器3560和寄存器改变3576的所有尝试读取或改变。

[0475] 为了判断在电子设备3504上运行的一个或多个进程3573是否被隐藏，安全设备驱动程序3570可以从操作系统3512判断什么内核模式进程正在操作系统3512上运行。安全设备驱动程序3570可以扫描诸如活动进程列表3584之类的操作系统内核存储器3580的部分来确定这样的进程。安全设备驱动程序3570因而可以具有运行进程列表3580的列表，该列表可以检测电子设备3504的内核模式中的操作。安全设备驱动程序3570可以把运行进程列表3580发送给O/S下层安全代理3516。通过经由超级调用把运行进程列表3580所检测的每一进程的EPROCESS结构传送给O/S下层安全代理3516，安全设备驱动程序3570可以把运行进程列表3580发送给O/S下层安全代理3516。O/S下层安全代理3516可以计算在运行进程列表3580内包含的每一这样的EPROCESS的CR3值。O/S然后，下层安全代理3516可以把运行进

程列表3580中的起因于安全设备驱动程序3570的CR3值与它在电子设备3504的操作期间编译的寄存器改变3576进行比较。在运行进程列表3580和寄存器改变3576之间任何矛盾可以是在电子设备3504上隐藏了进程3573中的一个或多个的结果。

[0476] 如果O/S下层安全代理3516判断存在在电子设备3504上运行的隐藏进程的迹象，则O/S下层安全代理3516可以扫描操作系统3512和操作系统内核存储器3580，以便判断是否已经做出与这样的进程相关的任何修改。O/S下层安全代理3516可以扫描以便发现已知由恶意软件实施的任何存储器修改。在一个实施例中，O/S下层安全代理3516可以扫描操作系统代码部分3582以及活动进程列表3584以便发现存储器修改。O/S下层安全代理3516可以修复在操作系统内核存储器3580中找到的任何恶意修改。O/S下层安全代理3516可以移除通过其对操作系统内核存储器3580中的存储器修改的观察所确定的任何检测到的rootkit感染，或修复对任何内部数据结构或代码段的任何感染。O/S下层安全代理3516可以被配置为对在由O/S下层安全代理3516和安全设备驱动程序3570确定的进程之间出现的每一矛盾重复扫描过程，以便发现隐藏进程的存储器修改。O/S下层安全代理3516可以产生诸如运行进程列表3586之类的最终进程列表并把这样的列表发送给安全设备驱动程序3570。

[0477] 安全设备驱动程序3570可以把安全DLL 3572或对安全DLL 3572的引用注入到在电子设备3504上运行的每一进程的地址空间，例如运行进程列表3586。进程地址空间3587中的每一个因而可以包含指向安全DLL 3572的指针。安全DLL 3572可以列举在电子设备3504上运行的所有用户模式进程。安全DLL3572可以把其结果编译到运行进程列表3588并把运行进程列表3588发送到安全设备驱动程序3570。

[0478] 安全设备驱动程序3570可以把所接收的运行进程列表3588与从O/S下层安全代理3516接收到的运行进程列表3586的列表进行比较。安全设备驱动程序3570可以判断，这样的进程列表中的任何差异可以指示恶意软件感染，例如用户模式rootkit攻击。安全设备驱动程序3570可以检查与进程地址空间3587相关联的存储器中的进程代码和数据部分，以便判断是否已经做出任何存储器修改，并做出任何必要的修复。安全设备驱动程序3570可以对在来自安全DLL 3572的运行进程列表3588和来自O/S下层安全代理3516的运行进程列表3586之间出现的每一矛盾重复扫描进程地址空间3587、检测存储器修改和修复这样的存储器修改的过程。

[0479] 图36是用于检测和修复电子设备上的隐藏进程的方法3600的示例实施例。

[0480] 在步骤3605，可以截取和记录对控制寄存器的已尝试访问。这样的控制寄存器可以是CR3控制寄存器。可以通过捕获VM退出来截取这样的尝试访问。可以通过在虚拟机控制结构中设置标志来实现这样的截取。可以重复步骤3605，以便在步骤3610构建O/S级别下层的进程列表，该列表可以包括给定时间周期内访问控制寄存器的所有进程的记录。在使用虚拟存储器的系统中，可以访问控制寄存器以便切换和访问这样的虚拟存储器。可以借助于O/S下层安全代理的辅助来实现步骤3605-3610。在方法3600的操作期间根据需要，可以周期性地或按需重复步骤3605-3610，以便为方法3600中要做出的各种比较提供经更新的基准。

[0481] 在步骤3615，可以从操作系统的内核模式的角度确定在电子设备的操作系统上运行的进程。可以通过使用操作系统的内核模式的列举函数来确定这样的进程。例如，可以访

问操作系统的活动进程列表以确定运行进程。在步骤3620，这样的进程可以被用来构建O/S级别的进程列表。在步骤3625，可以计算来自O/S级别的进程列表中的每一进程的EPROCESS结构的控制寄存器值。这样的寄存器值可以允许在O/S级别的进程列表中交叉引用该进程。

[0482] 在步骤3630，可以比较O/S级别下层的进程列表和O/S级别的进程列表，以便判断是否存在任何矛盾。如果存在在中O/S级别的进程列表丢失但在O/S级别下层的进程列表中存在的任何进程，则在步骤3635，可以判断这样的进程被隐藏且因而是恶意的。

[0483] 在步骤3640，可以扫描操作系统和系统存储器以便发现与隐藏进程相关联的存储器修改。在一个实施例中，可以扫描这样的资源以便发现进程列举相关的变更。例如，可以扫描操作系统代码段和/或操作系统活动进程列表。在步骤3645，可以修复任何检测到的存储器修改。在步骤3650，可以对所有隐藏进程重复步骤3605-3645，直到在O/S级别下层的进程列表和O/S级别的进程列表的元素之间不存在矛盾。

[0484] 图37是用于检测和修复电子设备上的隐藏进程的方法3700的示例实施例。方法3700与方法3600的不同之处在于，方法3600涉及包括内核模式进程和用户模式进程两者的列表的创建和比较，而方法3700涉及内核模式进程列表和用户模式处理器列表的单独的创建和比较。通过比较进程在一个列表中存在且在另一个中不存在，可以确定恶意软件进程的性质，例如恶意软件是用户模式rootkit还是内核模式rootkit。而且，rootkit可能已经感染一个或许多进程。

[0485] 在步骤3705，可以截取和访问对控制寄存器的已尝试访问。这样的控制寄存器可以是CR3控制寄存器。可以通过捕获VM退出来截取这样的已尝试访问。可以通过在虚拟机控制结构中设置标志来实现这样的截取。可以重复步骤3705，以便在步骤3710构建O/S级别下层的进程列表该列表可以包括给定时间周期内访问控制寄存器的所有进程的记录。在使用虚拟存储器的系统中，可以访问控制寄存器以便切换和访问这样的虚拟存储器。可以借助于O/S下层安全代理的辅助来实现步骤3705-3710。在方法3700的操作期间根据需要，可以周期性地或按需重复步骤3705-3710，以便为方法3700中要做出的各种比较提供经更新的基准。

[0486] 在步骤3715，可以从操作系统的内核模式的角度确定在电子设备的操作系统的内核模式中运行的进程。可以通过使用操作系统的内核模式的列举函数来确定这样的进程。例如，可以访问操作系统的活动进程列表以判断在内核模式中运行的进程。在步骤3720，这样的进程可以被用来构建O/S级别的进程列表。在步骤3725，可以计算来自O/S级别的进程列表中的每一进程的EPROCESS结构的控制寄存器值。这样的寄存器值可以允许在O/S级别的进程列表中交叉引用该进程。

[0487] 在步骤3730，可以把O/S级别下层的和O/S级别的进程列表进行比较，以判断是否存在任何矛盾。如果存在在O/S级别的进程列表丢失但出现在O/S级别下层的进程列表中的任何进程，则在步骤3735可以判断这样的进程可以被隐藏的且因而是恶意的，可能采取内核模式rootkit的形式。在一个实施例中，替代地，可以判断，在O/S级别的进程列表中丢失的进程事实上是用户模式进程。在这样的实施例中，步骤3715可能尚未列举电子设备的用户模式进程。

[0488] 在步骤3740，可以扫描操作系统和系统存储器以便发现与隐藏进程相关联的存储器修改。在一个实施例中，可以扫描这样的资源以便发现进程列举相关的变更。例如，可以

扫描操作系统代码段和/或操作系统活动进程列表。在步骤3745，可以修复任何检测到的存储器修改。在步骤3750，可以对所有隐藏的内核模式进程重复步骤3735-3745，直到在O/S级别下层的进程列表和O/S级别的进程列表的内核模式元素之间不存在矛盾。

[0489] 在步骤3755，可以确定和列举电子设备的用户模式进程。可以通过把共享库注入到每一运行进程的地址空间中来实现步骤3755。共享库可以调用操作系统的用户模式进程列举函数。在步骤3760，可以利用步骤3755的结果来创建用户级别进程列表。在步骤3765，可以判断在O/S级别下层的进程列表和用户级别进程列表之间的矛盾。不出现在用户级别进程列表中且先前不出现在O/S级别的进程列表中的任何进程，可能是隐藏用户模式进程，且因而与恶意软件相关联的。在步骤3770，可以扫描用户模式的应用和进程空间以便发现存储器修改。可以检查共享库被注入到其中的地址空间的进程代码以便发现这样的存储器修改。在步骤3775可以扫描主控用户模式进程代码以及数据部分的存储器的部分。在步骤3780，可以修复任何检测到的存储器修改。在步骤3785，可以重复步骤3755-3780，直到在O/S级别下层的进程列表和用户级别进程列表的用户模式元素之间不存在矛盾。

[0490] 图38是用于保护对在电子设备3801上执行的操作系统3813的系统调用的访问的系统3800的示例实施例。系统3800可以包括O/S下层捕获代理3820和已触发事件应对程序3822，它们被配置为在电子设备3801上操作，以便检测来自在诸如操作系统3813之类的电子设备3801的操作系统中运行基于软件的实体的对访问系统调用和/或系统调用表的恶意尝试。此外，O/S下层捕获代理3820和已触发事件应对程序3822可以被配置为使用一个或多个安全规则3808来判断何时捕获对系统调用和/或系统调用表3804的访问以及如何应对与已捕获操作相关联的已触发事件。O/S下层捕获代理3820和已触发事件应对程序3822可以被配置为对已触发事件进行允许、拒绝，或采取其他矫正动作。

[0491] 电子设备3801可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备3801可以包括被耦合到存储器3803的一个或多个处理器3802。处理器3802可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器902、图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。存储器3803可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的存储器1203和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备3801可以包括操作系统3813，操作系统3813可以包括系统调用表3804、虚拟存储器页面表3806和O/S内部安全代理3819。操作系统3813可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213、和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理3819可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219和/或其任何合适的组合实现，或者被配置为实现它们的功能性。安全规则3808可以是由图1的安全规则114、图2的安全规则220、222、图4的安全规则420、434、436、438、图5的安全规则518、图7的安全规则707、721、723、图9的安全规则908、921、图12的安全规则1208、1221和/或其任何组合实现，或者被配置为实现它们的功能性。保护服务器3818可以全部地或部分地由图1的保护服务器102、图2的保护服务器202和/或

其任何组合实现，或者被配置为实现它们的功能性。

[0492] O/S下层捕获代理3820可以由图1的O/S下层捕获代理104、图2的SVMM216、图4的固件安全代理440、442和/或PC固件安全代理444、图5的固件安全代理516和/或图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。已触发事件应对程序3822可以由图1的已触发事件应对程序108、图2的SVMM安全代理217、图4的O/S下层代理450、图7的O/S下层代理712、图9的已触发事件应对程序922和/或其任何组合实现，或者被配置为实现它们的功能性。在各种实施例中，O/S下层捕获代理3820的功能性中的一些可以由已触发事件应对程序3822实现，和/或已触发事件应对程序3822的功能性中的一些可以由O/S下层捕获代理3820实现。此外，O/S下层捕获代理3820和已触发事件应对程序3822可以在相同的软件模块中实现。

[0493] 页面表3806可以被实现为数据结构且可以被用来实现操作系统3813的虚拟存储器系统。虚拟存储器系统是把对电子设备3801的存储器3803的访问虚拟化的存储器管理系统。在虚拟存储器系统中，给在操作系统3813上执行的软件进程提供虚拟地址空间，进程可以把该虚拟地址空间看作是连续的存储器块。实际上，虚拟地址空间可以跨越不同的物理存储器区域而散布。在进程请求访问存储器时时，操作系统3813可以负责把进程的虚拟地址映射成数据实际上存储于其中的存储器3803的物理地址。虚拟地址空间可以被分割成被称为虚拟存储器页面的固定大小的连续的虚拟存储器地址块。页面表3806可以被用来存储从虚拟存储器页面到虚拟存储器页面存储在其中的存储器3803中其相应的物理地址的映射。页面表3806可以包括各种访问权限，例如读、写和/或执行，以便指定对给定虚拟存储器页面授权的访问类型。在一些实施例中，O/S下层捕获代理3820和/或已触发事件应对程序3822可以被配置为捕集任何已产生的异常或意图的读、写或执行操作并使用安全规则3808来判断对访问存储器3803的未经授权的请求是否指示恶意软件。

[0494] 系统调用表3804可以是由操作系统3813用来实现系统调用的数据结构。系统调用可以是由操作系统3813提供的例程和/或系统服务。系统调用表3804可以提供在应用3810和操作系统3813之间的接口，允许应用3810请求操作系统3813执行未授权应用3810执行的操作。可以使用对系统调用表3804的索引标识每一系统调用，系统调用表3804中，可以存储特定的系统调用的条目。系统调用表3804中的每一条目可以存储其中可以存储对应于特定的系统调用代码的存储器3803中的地址。这样的条目可以被实现为指针。可以通过向操作系统3813告知适当的索引和随后把控制权传递给操作系统3813来执行系统调用。然后，操作系统3813可以查阅系统调用表3804以便标识其中存储对应于特定的系统调用的代码的存储器3803中的位置。然后，操作系统3813可以执行代码并把控制权返回给负责请求系统调用的软件组件。可以在来自下面的图39的系统调用表3901的描述中找到系统调用表3804的示例实施例的描述。

[0495] O/S下层捕获代理3820可以被配置为截取对与系统调用相关联的诸如例如存储器3803和/或处理器3802的任何合适的资源3816的访问或来自与系统调用相关联的任何合适的资源3816的信息。例如，资源3816可以由图1的资源106、图2的系统资源214、图7的系统资源、图9的处理器资源924、图12的虚拟存储器1204和/或物理存储器1203和/或其任何组合实现，或者被配置为实现它们的功能性。资源3816可以包括对诸如处理器3802之类的处理器可用以便允许处理器加载和执行指令的资源。这样的资源可以包括，例如，数据寄存器、

控制寄存器、高速缓存、处理器标志、处理器核心、处理器异常、和/或处理器中断。资源3816也可以包括虚拟存储器和/或物理存储器3803。对这样的资源的已尝试访问可以包括指令，例如带有操作数的汇编语言指令，且可以通过捕获指令的执行来捕获这样的尝试访问。

[0496] O/S下层捕获代理3820可以被配置为截取对诸如存储器3803和/或处理器3802的资源之类的任何合适的资源的访问或来自任何合适的资源的信息。O/S下层捕获代理3820可以包括可以被用来捕获对访问系统调用和/或系统调用表3804的尝试的系统调用捕获器3814。然后，系统调用捕获器3814和/或已触发事件应对程序3822可以结合安全规则3808使用与已捕获尝试相关联的上下文信息，以便判断是否允许尝试、拒绝尝试和/或把尝试报告给一个或多个用户3812。上下文信息可以包括已捕获访问尝试的请求实体、所讨论的特定系统调用和/或所尝试的特定类型的访问(例如，执行系统调用的尝试或读/写系统调用表3804中的条目的尝试)。

[0497] 系统调用捕获器3814可以是O/S下层捕获代理3820的模块和/或组件，且可以被配置为以任何合适的方式捕获对系统调用的访问。例如，系统调用捕获器3814可以被配置为捕获诸如把控制权传递给操作系统3813以供执行系统调用的指令之类的被用来实现系统调用的汇编语言指令的执行。要捕获的特定指令可以取决于电子设备3801的特定处理器3802和/或操作系统3813。作为示例，在使用在支持x86指令集体系结构(“ISA”)的处理器3802上执行的微软Windows™执行的变种时，系统调用捕获器3814可以捕获对执行‘SysEnter’和/或‘KiFastSysCall’指令的尝试。这些指令用于把控制权传递给操作系统3813以便执行系统调用。捕获对执行‘SysEnter’指令的尝试可以仅捕获来自在“3环”优先级执行的软件的尝试，而捕获对执行‘KiFastSysCall’指令的尝试可以捕获捕获来自在“0环”或“3环”优先级执行的软件的尝试。在一些实施例中，可以通过捕获对执行对应于特定指令存储在其中的物理存储器位置的虚拟存储器页面的尝试来捕获对执行‘SysEnter’和/或‘KiFastSysCall’指令的尝试。

[0498] 在另一实施例中，系统调用捕获器3814可以被配置为捕获对访问系统调用表3804的尝试。被用来捕获对访问系统调用表3804的尝试的特定方法可以取决于电子设备3801的特定处理器3802和/或操作系统3813。在使用支持x86ISA的处理器3802时，系统调用捕获器3814可以通过捕获对执行MOV指令的某种尝试来捕获对读或写系统调用表3804的尝试。例如，系统调用捕获器3814可以通过捕获指令“MOV syscall_table_address,EAX”来捕获对写入系统调用表3804的尝试。这一指令可以操作为把来自EAX寄存器的值入到在syscall_table_address所指定的存储器地址处的系统调用表中的条目。通过捕获对写如系统调用表3804的尝试，系统调用捕获器3814可以防止恶意软件用包含恶意代码的存储器地址盖写系统调用表3804中的条目。类似地，系统调用捕获器3814可以捕获对从系统调用表3804读取的尝试，通过捕获指令“MOV EAX,syscall_table_address”。这一指令可以操作为从在syscall_table_address所指定的存储器地址处的系统调用表3804中的条目读取值。捕获对读取系统调用表3804中的条目的尝试将有效地捕获对执行与系统调用表3804中的条目相关联的系统调用的所有尝试，这是由于必须读取系统调用表3804才能允许操作系统3813标识在对应于系统调用的代码存储器中的位置。另外，捕获对读取系统调用表3804中的条目的尝试将捕获例如恶意软件对读取系统调用表3804的任何直接尝试。在一些实施例中，捕获对读取系统调用表3804的所有尝试。在一些实施例中，可以通过捕获对访问对应于系

统调用表3804存储在其中的物理存储器位置的虚拟存储器页面的尝试来捕获对访问系统调用表3804的尝试。

[0499] 在又一实施例中，通过捕获对执行在系统调用的代码驻留在其中的存储器位置处的代码的尝试，系统调用捕获器3814可以被配置为捕获对执行系统调用的尝试。可以查询系统调用表3804，以便标识特定系统调用的代码驻留在其中的存储器位置。被用来捕获对执行系统调用的尝试的特定方法可以取决于电子设备3801中的处理器3802的类型。在一个实施例中，可以使用基于例如指令指针(IP)寄存器的值的触发器捕获对执行系统调用的尝试。在一些实施例中，IP寄存器可以被称为程序计数器(PC)寄存器。取决于具体的处理器，IP寄存器可以被用来存储当前正在执行的指令的地址或的接下来要执行的指令地址。在使用支持x86ISA的处理器3802时，通过监视IP寄存器的值以及在在IP寄存器的值包含系统调用的地址时捕获执行，系统调用捕获器3814可以捕获对执行特定的系统调用的尝试。在另一实施例中，通过捕获对执行对应于系统调用的代码存储在其中的物理存储器位置的虚拟存储器页面的尝试，可以捕获对执行系统调用的尝试。可以在对应于要捕获的特定的系统调用的索引处查询系统调用表3804，以便标识对应于系统调用的代码在存储器中的位置。在一个实施例中，可以通过捕获用于传递控制的指令(例如被定向到已知是在系统调用表3804内的位置的“JMP”例程)来捕获包含被链接到系统调用表3804的例程和函数的存储器位置的执行。

[0500] 用于捕获对系统调用和系统调用表3804的访问的上面的方法的具体实现可以取决于O/S下层捕获代理3820和/或系统调用捕获器3814的具体实现。例如，如果在虚拟机监视器中实现O/S下层捕获代理3820和/或系统调用捕获器3814，则基于特定存储器地址的任何捕获(例如，基于对读/写系统调用表中的条目的捕获和/或基于在包含系统调用的代码的存储器位置处执行的捕获)尝试可以基于虚拟存储器地址，这是由于可能还没有把存储器地址从虚拟存储器地址转换成物理存储器地址。作为另一示例，如果在微代码安全代理中实现O/S下层捕获代理3820和/或系统调用捕获器3814，则基于特定存储器地址的任何捕获可以基于物理存储器地址，因为可能已经在微代码级别执行从虚拟地址到物理地址的转换。

[0501] 在捕获尝试期间，可以检测做出对访问系统调用表3804或与系统调用表3804相关联的函数的尝试的指令的存储器位置。可以分析该存储器位置以便判断做出尝试的实体。

[0502] 在捕获特定的尝试之后，系统调用捕获器3814可以创建与尝试相关联的已触发事件并将其发送给已触发事件应对程序3822。然后，已触发事件应对程序可以结合安全规则3808使用与已捕获事件相关联的上下文信息来判断是否允许该事件、拒绝该事件和/或把该事件报告给一个或多个用户3812。上下文信息可以包括已捕获事件的请求实体、所讨论的特定系统调用和/或所请求的动作(例如，对执行系统调用的尝试和/或对读/写系统调用表3804中的条目的尝试)。例如，可以仅允许已知是安全的和免遭恶意软件的某些实体写入系统调用表3804。基于对写入系统调用表3804的未经授权的尝试，可以判断先前未知其恶意软件状态的实体是恶意软件。在另一示例中，可以捕获链接到系统调用表3804的函数的执行，且仅当做出对函数的调用的实体通过系统调用表3804做出这样的尝试时允许该执行。可以拒绝先前未知其恶意软件状态的实体对执行这样的函数的直接访问的尝试。另外，可以拒绝通过黑名单或其他判断被判断为恶意软件的、尝试访问系统调用表3804或其相关

的函数的实体的访问，且可以采取其他合适的矫正动作。

[0503] 用户3812可以包括供与和对访问系统调用表3804的已捕获尝试和/或对执行系统调用的已捕获尝试相关联的信息一起使用的任何实体。用户3812可以包括电子设备3801的应用3810和/或安全代理，和/或可以包括第三方应用或其他软件。例如，用户3812可以包括在电子设备3801上执行的安全软件，例如O/S下层捕获代理3820、已触发事件应对程序3822和/或O/S内部安全代理3819，它们可以把与已捕获尝试相关联的上下文信息用于检测恶意软件。在一些实施例中，每一用户3812可以提供在与O/S下层捕获代理3820相同的优先级水平操作的其自己的安全代理，诸如例如固件安全代理。用户3812也可以包括远程地执行例如在保护服务器3818上执行的安全软件。作为另一示例，用户3812可以包括特定资源的制造者，例如由电子设备3801使用的任何I/O设备。制造者可以对经由对系统调用表的访问和/或诸如与资源相关联的系统调用之类的系统调用的执行来危害资源的任何可疑尝试感兴趣。作为另一示例，用户3812可以包括数字权限管理(“DRM”)系统的管理员。DRM系统可以限制和/或控制数字内容的使用，且通常被用来保护受版权保护的数字内容，例如视频和/或音乐内容。DRM系统的管理员可以对知道何时和如何访问各种数字受保护文件感兴趣，且可以通过跟踪可以被用来访问受保护文件的各种系统调用来实现这一点。可以给用户3812提供应用程序编程接口(“API”)，以便允许用户3812访问与对执行系统调用的已捕获尝试和/或对访问系统调用表3804的已捕获尝试相关联的信息。

[0504] 图39是供与保护对操作系统的系统调用的访问的系统和/或方法一起使用的系统调用表3901的示例实施例。系统调用表3901可以被操作系统用来把地址3906存储在每一系统调用3904的代码所驻留的存储器3908中。例如，可以使用系统调用表3901来实现图38的系统调用表3804的功能性。系统调用表3901可以由表、记录和/或其他合适的数据结构实现。在带有微软Windows™操作系统的变种的实施例中，系统调用表3901可以由系统服务描述符表(“SSDT”)实现。系统调用3904可以是操作系统提供的例程和/或系统服务。典型的系统调用3904可以包括，例如，用于操作和/或执行文件的打开、读、写、关闭和/或执行、用于创建新的进程的ntCreateProcess和/或用于加载新的驱动程序的ntLoadDriver和ZwLoadDriver。

[0505] 系统调用3904可以提供在应用和操作系统之间的接口，这允许应用请求操作系统执行未授权该应用执行的操作。例如，通常在“3环”优先级执行的应用可能需要访问盘上的文件但不拥有执行盘I/O操作的权限。该应用可以使用诸如读或写文件系统调用之类的系统调用3904来把控制权传递给操作系统，以便允许操作系统满足来自该应用的请求。可以在“0环”优先级执行的操作系统可以提供与特定的系统调用3904相关联的服务，且然后可以把控制权传递回去给该应用。例如，操作系统可以访问系统调用表3901以便标识标识对应于系统调用3904的代码位于其中的存储器地址3906。然后，操作系统可以执行在存储器3908中的指定地址3906处的代码，且然后，可以把控制权传递回去给该应用。以这种方式，该应用可以利用通常仅对诸如操作系统之类的在“0环”优先级执行的软件可用的某些服务。

[0506] 可以使用索引3902来把每一系统调用3904引用到其中存储有系统调用3904的条目的系统调用表3901中。例如，系统调用表3901总共由N个条目，且使用在0到N-1范围内的索引3902来引用每一条目。通过向操作系统告知适当的索引3902并把控制权传递给操作系

统，可以执行系统调用3904。在一些实施例中，软件组件可以通过把索引放置到处理器的寄存器中来指定适当的索引3902，且然后可以执行指令以便把控制权传递给操作系统，以供执行系统调用3904。例如，在一个实施例中，使用x86指令集体系结构(“ISA”)，可以使用下列指令来为应用实现系统调用：

[0507] “MOV EAX,index”

[0508] “SysEnter”

[0509] 第一指令把“index(索引)”移动到处理器的EAX寄存器中，其中’index’是对应于特定的系统调用3904的条目驻留在其中的系统调用表3901中的索引3902的整数。然后，’SysEnter’指令把控制权传递给操作系统，且操作系统可以访问在EAX寄存器中所指定的索引3902处的系统调用表3901。系统调用表3901的特定索引3902处的条目可以指定存储器地址3906，存储器地址3906指向特定的系统调用3904的代码驻留在其中的存储器3908中的位置。然后，处理器可以执行位于存储器3908中的指定地址3906的代码。系统调用3904可以由包括应用、操作系统和/或驱动程序的任何软件组件执行。作为x86ISA上的示例，操作系统和/或驱动程序可以以类似于应用的方式执行系统调用3904，所不同的是使用’KiFastSysCall’指令。

[0510] 可以向系统调用表3901添加和/或从中移除系统调用3904。例如，如果把新设备添加到电子设备，则需要操作系统加载新设备的设备驱动程序，且需要把系统调用3904添加到系统调用表3901，以便允许应用利用新设备的功能性。可以把新的系统调用的代码加载到存储器3908中，且可以把新的系统调用3904的条目添加到系统调用表3901的末尾，指定系统调用的代码驻留在其中的存储器3908中的地址3906。

[0511] 上面所描述的用于实现系统调用3904的实施例仅仅是多种可能的实施例中的一些。系统调用3904和/或系统调用表3901可以以任何合适的方式实现。系统调用3904和/或系统调用表3901的具体实现可以取决于电子设备的特定的处理器和/或操作系统。

[0512] 图40是用于保护对在电子设备上执行的操作系统的系统调用的访问的方法4000的示例实施例。在步骤4005，可以认证O/S下层安全代理、O/S内部安全代理、已触发事件应对程序和保护服务器的身份和安全。可以使用任何合适的方法来执行这样的认证，包括通过定位和检验每一组件的存储器中的映像、使用密码散列和/或使用密钥。直到步骤4005完成之前，可以停止其他步骤的操作。在步骤4010获得安全规则。安全规则可以由O/S下层安全代理、O/S内部安全代理和/或已触发事件应对程序本地存储，和/或可以远程存储，例如在保护服务器上。这样的安全规则可以被用来在步骤4015-4040做出判定。

[0513] 在步骤4015，可以截取对执行系统调用和/或访问系统调用表的尝试。在一些实施例中，可以通过捕获对执行被用来实现系统调用的控制权传递指令的尝试来截取对执行系统调用的尝试。例如，一些处理器和/或操作系统可以使用诸如SysEnter和/或KiFastSysCall指令之类的控制权传递指令来实现系统调用，并且，可以通过捕获适当的控制权传递指令的执行来截取对执行系统调用的尝试。也可以通过捕获对执行对应于特定的控制权传递指令存储在其中的物理存储器位置的虚拟存储器页面的尝试来截取对执行系统调用的尝试。在一些实施例中，可以通过捕获对执行包含系统调用的代码的存储器位置处的代码的尝试来截取对执行系统调用的尝试。在这样的实施例中，捕获可以基于IP寄存器的值。例如，可以查询系统调用表或存储器映射，以便标识包含系统调用的代码的存储器

位置，并且在IP寄存器包含特定的系统调用的存储器位置的地址时可以发生捕获。在其他实施例中，可以通过捕获对执行对应于特定的系统调用的代码存储在其中的物理存储器位置的虚拟存储器页面的尝试，可以截取对执行系统调用的尝试。在一些实施例中，也可以截取对读或写系统调用表的尝试。在这样的实施例中，可以通过捕获用来读或写系统调用表中的存储器位置的指令的执行来截取该尝试。例如，在x86指令集体系结构上，在使用MOV指令来读或写系统调用表中的位置时，可以捕获MOV指令。在一些实施例中，通过捕获对访问对应于系统调用表存储在其中的物理存储器位置的虚拟存储器页面的尝试，也可以截取对读或写系统调用表的尝试。

[0514] 在步骤4020，标识已尝试访问的源。例如，已尝试访问可以来自应用，驱动程序、O/S内部安全代理、操作系统和/或其他软件实体。在步骤4025，判断该尝试是否得到授权。可以结合与该尝试相关联的上下文信息使用安全规则来判断可以还是不可以授权特定的尝试。上下文信息可以包括已尝试访问的源和/或访问的具体类型。例如，安全规则可以指定，仅操作系统可以写入系统调用表。作为另一示例，安全规则可以指定，带签名的驱动程序或与条目相关联的其他软件组件可以写其自己的条目。如果尝试得到授权，那么，在步骤4030允许访问。如果尝试未得到授权，那么，在步骤4035拒绝访问。最终，在步骤4040，判断是否应当把该尝试报告给一个或多个用户。是否应当报告尝试可以取决于所讨论的具体系统调用和与已尝试访问相关联的上下文信息。安全规则可以指定何时应把对执行系统调用和/或访问系统调用表的尝试报告给一个或多个用户。

[0515] 根据保护存储设备的要求，可以连续地、周期性地、根据需求或在事件触发时重复来自图40的方法的步骤。

[0516] 图41是用于调节和控制电子设备4104上的恶意的或潜在恶意的代码的系统4100的示例实施例。例如，系统4100可以被用于调节和控制电子设备4104上的自修改代码。系统4100可以包括被配置为在电子设备4104上操作以便防备恶意软件自修改自身以便逃避检测的尝试的O/S下层安全代理4116。作为另一示例，系统4100可以被用于修改电子设备4104上的恶意代码。系统4100可以包括O/S下层安全代理4116被配置为在电子设备4104上操作以便修改恶意代码从而使得所检测到的恶意软件失效。作为进一步的示例，系统4100可以被用于监视和跟踪线程以便标识包括潜在恶意的代码的线程族。系统4100可以包括O/S下层安全代理4116被配置为在电子设备4104上操作以便监视和跟踪在各线程当中关系。

[0517] 此外，O/S下层安全代理4116可以被配置为使用一个或多个安全规则4122来判断捕获什么已尝试操作以及如何响应这样的已捕获操作。O/S下层安全代理4116可以被配置为允许已捕获操作、拒绝已捕获操作或对已捕获操作采取其他矫正动作。

[0518] 如图4100中所示出，电子设备4104可以包括被耦合到存储器的4108处理器4106、操作系统4112、O/S下层安全代理4116和安全规则4122。电子设备4104可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。处理器4106可以全部地或部分地由图2的处理器208，图4的处理器408，图7的处理器702，图9的处理器902，图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。存储器4108可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的物理存储器1203或虚拟存储器和/或其任何组合实现，或者被

配置为实现它们的功能性。操作系统4112可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理4116全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则4122可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、或图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。可以以任何合适的方式建立安全规则4122(例如，由电子设备4104的用户设置的政策、由包括电子设备4104的企业的管理员设置的政策、由O/S下层安全代理4116的创建者设置的政策等等)。在一些实施例中，O/S下层安全代理4116可以经由网络244请求和/或接收来自保护服务器202的对安全规则4122的更新或修改(例如，由于对恶意软件定义的更新)。

[0519] 操作系统4112可以包括O/S内部安全代理4118。O/S内部安全代理4118可以全部地或部分地由图2的O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理718、图9的O/S内部安全代理919、O/S内部安全代理1219和/或其任何组合实现，或者被配置为实现它们的功能性。

[0520] 如图41中所示出，存储器4108可以包括权限标志4136和历史4140。权限标志4136可以维护标志、变量或建立与被存储在存储器4108中的内容相关的权限的其他数据。例如，权限标志4136可以指示，对于存储器4108的特定的位置(例如，页面或地址)，在电子设备4104上执行的实体是否可以读、写和/或执行被存储在特定位置的内容。在一些实施例中，可以在存储器4108的页面表条目(PTE)和/或页面目录条目(PDE)中实现权限标志4136。权限标志4136可以被存储在特定存储器位置(例如，页面或地址范围)。

[0521] 历史4140可以包括日志、列表、高速缓存和/或用于记录已捕获的已尝试访问和与已捕获的已尝试访问相关联的信息(例如，已尝试访问的类型、与已捕获已尝试访问相关联的存储器位置等等)的其他合适的数据结构。为了防止恶意软件尝试经由对历史4140的尝试恶意访问规避O/S下层安全代理4116的有效性，可以根据在此描述为用于保护存储器的一种或多种方法来保护历史4140的内容免遭已尝试访问。例如，O/S下层安全代理4116可以捕获不同于O/S下层安全代理4116的实体对历史4140驻留在其中的存储器页面或存储器地址范围的已尝试访问，并拒绝这样的已捕获的已尝试访问。

[0522] 如上所述，O/S下层安全代理4116可以基于安全规则4122检测恶意代码的存在。O/S下层安全代理4116可以经由以上所描述的任何方法和/或以任何其他合适的方式检测恶意代码的存在。例如，O/S下层安全代理4116可以通过捕获对存储器4108或电子设备4104的其他资源的访问检测恶意代码的存在。作为另一示例，O/S下层安全代理4116可以通过扫描存储器4108的页面和/或存储4126以便发现恶意代码来检测恶意代码的存在。作为进一步的示例，O/S下层安全代理4116可以通过从O/S内部安全代理4118接收O/S内部安全代理4118已经检测到恶意代码的存在的通信来检测恶意代码在存储器中的存在。

[0523] 尤其，O/S下层安全代理4116可以基于安全规则4122捕获对存储器的一个或多个

已尝试访问，这些已尝试访问可以单独地或整体地指示自修改恶意软件的存在。作为示例，对权限标志4136中所陈述的存储器位置的权限的改变(例如，从读改变成读/写或从读/写改变成读/写/执行)可以(例如，单独地或与其他已尝试存储器访问一起整体地)指示恶意软件的存在。因此，O/S下层安全代理4116可以捕获一就被检测改变to权限标志4136。例如，在一些实施例中，O/S下层安全代理4116可以根据安全规则4122捕获对包括权限标志4136的存储器4106的位置(例如，页面或地址)的已尝试访问。在相同的或替代的实施例中，O/S下层安全代理4116可以根据安全规则4122捕获对包括操作系统4112的用于修改存储器权限(例如，MiSetProtectionOnSection，AllocateVirtualMemory()，MiProtectVirtualMemory()和/或Windows™中的其他合适的函数调用)的调用和/或函数的存储器4108的位置(例如，页面或地址)的和/或包括操作系统4112的相应权限标志(例如，NTProtectVirtualMemory，ZwProtectVirtualMemory，ProtectVirtualMemory和/或Windows™中的其他合适的标志)的存储器4108的位置(例如，页面或地址)的已尝试访问。

[0524] 作为另一示例，把内容从存储器4106的一个位置复制到另一位置可以指示(例如，单独地或与其他已尝试存储器访问一起整体地)恶意软件的存在。因此，O/S下层安全代理4116可以捕获与在存储器位置之间复制内容相关联的已尝试访问。例如，在一些实施例中，O/S下层安全代理4116可以根据安全规则4122捕获用于把内容从存储器的一个位置复制到另一位置的处理器函数。在相同的或替代的实施例中，O/S下层安全代理4116可以根据安全规则4122捕获对包括用于操作系统4112的复制数据的调用和/或函数例如Windows™中的MemUICopy函数的存储器4108的位置(例如，页面或地址)的已尝试访问。

[0525] 作为进一步的示例，被存储在存储器4106中的内容的修改或“就地写入(writing-in-place)”可以(例如，单独地或与其他已尝试存储器访问一起整体地)指示恶意软件的存在。因此，O/S下层安全代理4116可以捕获与存储器4108中的内容的就地写入相关联的已尝试访问。例如，在一些实施例中，O/S下层安全代理4116可以根据安全规则4122捕获用于就地修改存储器4108中的内容的处理器函数。在相同的或替代的实施例中，O/S下层安全代理4116可以根据安全规则4122捕获对包括操作系统4112的用于就地修改内容的调用和/或函数的存储器4108中的位置(例如，页面或地址)的已尝试访问。

[0526] 作为进一步的示例，被存储在存储器中的经复制或经修改内容的执行可以(例如，单独地或与其他已尝试存储器访问一起整体地)指示恶意软件的存在。因此，O/S下层安全代理4116可以捕获与存储器4108中的内容的执行相关联的已尝试访问。例如，在一些实施例中，O/S下层安全代理4116可以根据安全规则4122捕获用于执行存储器4108中的内容的处理器函数。在相同的或替代的实施例中，O/S下层安全代理4116可以根据安全规则4122捕获对包括用于执行内容的的操作系统4112的调用和/或函数的存储器4108的位置(例如，页面或地址)的已尝试访问。

[0527] 作为进一步的示例，把内容加载到存储器中可以(例如，单独地或与其他已尝试存储器访问一起整体地)指示恶意软件的存在。因此，O/S下层安全代理4116可以捕获与把代码加载到存储器4108中相关联的已尝试访问。例如，在一些实施例中，O/S下层安全代理4116可以根据安全规则4122捕获用于把代码加载到存储器4108中的处理器函数或系统函数。O/S下层安全代理4116可以被配置为确定用于把代码加载到存储器4108中的安全的或规范的方法，例如操作系统加载器。可以测试或映射这样的安全的或规范的方法，以使得操

作系统4112所采用的逻辑或步骤可以是已知的。一旦捕获对把代码加载到存储器4108中的尝试，O/S下层安全代理4116可以判断这样的尝试是否匹配用于加载代码的已知方法。例如，如果该尝试涉及把代码加载到存储器的已分配部分，并尝试通过用直接写存储器回避操作系统加载器来这样做，则可以判断该尝试是恶意的。

[0528] 如果包含代码的页面或存储器范围已经被修改，则O/S下层安全代理4116可以被配置为跟踪修改。如果允许继续执行，则可以跟踪和记录已修改代码的随后操作。然而，O/S下层安全代理4116可以给予这样的代码较少特权，例如通过捕获和拒绝这样的经修改代码访问属于包含其他内核模式实体或操作系统的存储器的特权位置的尝试。经修改代码的恶意软件状态可以是未知的，并且直到确实判断它是安全的之前，O/S下层安全代理4116可以拒绝经修改代码访问内核函数或例程。

[0529] O/S下层安全代理4116可以把关于一个或多个已捕获的已尝试访问信息记录在历史4140中中。O/S下层安全代理4116可以不时地分析历史4140以判断是否已经发生相对于特定存储器位置的可疑行为。在其分析期间，O/S下层安全代理4116可以查阅规则4122以判断历史4140中实现的特定存储器位置的行为是否指示可以证明自修改恶意软件代码的潜在存在的可疑行为。例如，如果对历史4140的分析指示在第一存储器位置处的内容被复制到第二位置，在第二位置处被修改，且然后，已经发生对第二位置的内容的已尝试执行，则这样的指示可以是自修改恶意软件代码的潜在存在的迹象。作为另一示例，如果对历史4140的分析指示均具有在第三位置处的共同起源的、在第一存储器位置和第二存储器位置处的内容都是已尝试执行的目标，则这样的指示可以是自修改恶意软件代码的潜在存在的迹象。作为进一步的示例，如果对历史4140的分析指示在特定存储器位置处的内容具有在多个其他存储器位置处的起源，则这样的指示可以是自修改恶意软件代码的潜在存在的迹象。此外，历史4140可以记录在分层结构中的级别和实体之间所做出的修改。

[0530] 如果在其他存储器位置处的内容是特定存储器位置的内容的经复制或经修改版本，则在此所使用的在特定存储器位置处的内容对于在另一位置处的内容来说是“起源”，且包括其中在其他存储器位置处的内容除了在特定存储器地址处的内容之外的一个或多个中间起源的衍生物的情况。

[0531] 因为如果被应用到每一存储器位置则这样的历史4140的记录可以消耗电子设备4104的处理资源的显著部分，O/S下层安全代理4116可以仅在发生可以指示特定存储器位置易受恶意软件感染的已尝试访问时记录特定存储器位置的历史4140。例如，O/S下层安全代理4116可以在捕获对特定存储器位置的权限(如权限标志4136中所实现的)的修改时开始记录特定存储器位置的历史4140。

[0532] 另外，因为如果被应用到每一存储器位置和/或已尝试访问则分析历史4140以便判断是否已经发生可疑行为可以消耗电子设备4104的处理资源的显著部分，O/S下层安全代理4116可以仅在发生与特定存储器位置相关联的特定的已捕获的已尝试访问时分析相对于特定存储器位置的历史。例如，在一些实施例中，O/S下层安全代理4116可以在捕获对特定存储器位置处的内容的已尝试访问时发起与特定存储器位置相关联的对历史4140的分析。

[0533] 在某些实施例中，单个已捕获的已尝试访问的发生可以指示可疑行为而无需对历史4140的分析。例如，对权限标志4136中所陈述的某些存储器位置的权限的改变(例如，从

读到读/写或从读/写到读/写/执行)可以指示恶意软件的存在。例如,存储操作系统内核或安全应用的存储器位置的权限的改变可以指示证明潜在的恶意软件的存在的可疑行为。

[0534] 如果O/S下层安全代理4116检测到证明潜在的恶意软件的存在的可疑行为(例如,基于单个已捕获的已尝试访问或对历史4140的分析),则O/S下层安全代理4116可以发起矫正动作(例如,根据安全规则4122)。例如,在一些实施例中,O/S下层安全代理4116可以在检测到可疑行为时把被存储在与所检测的可疑行为相关联的特定存储器位置内容与已知的恶意软件和/或已知的可信/可靠的进程比较,以便判断该内容是不是恶意的。可以通过把内容的散列、指纹或其他签名与已知进程的散列、指纹或其他签名进行比较来实现这样的比较。

[0535] 替代地或另外,如果O/S下层安全代理4116检测到证明潜在的恶意软件的存在的可疑行为(例如,基于单个已捕获的已尝试访问或对历史4140的分析),O/S下层安全代理4116可以把与可疑行为相关联的取证迹象(例如,存储器位置的内容、与存储器位置相关联的历史4140等等)报告给保护服务器202以供进一步分析。在一些实施例中,然后,保护服务器202可以产生与内容相关联的签名(例如,散列或指纹)、产生与签名相关联的政策或黑名单条目并把这样的信息传输给在其他电子设备上执行的安全代理。在相同的或替代的实施例中,保护服务器202可以进一步分析可疑行为(例如,连同从其他电子设备接收的取证迹象一起),以判断该可疑行为是否实际上指示恶意软件,且如果是,则把关于类似的行为是否是恶意软件存在的迹象的指令(例如,以安全规则4122的形式)传输给电子设备。

[0536] 如果O/S下层安全代理4116判断,与可疑行为相关联的存储器位置的内容是恶意的(例如,通过把该内容与已知进程、从保护服务器202接收到的信息、对安全规则4122的引用和/或其他判断进行比较),O/S下层安全代理4116可以采取进一步的矫正动作(例如,根据安全规则4122)。这样的矫正动作可以包括但不限于不允许执行内容、撤销对内容的改变(例如,如历史4140中所陈述的内容的修改和复制)、修复内容、用无害内容替换该内容和/或禁用与该内容相关联的进程。

[0537] 在以上所描述的各种实施例中,当在存储器4108的不同部分之间传递内容时,可以过渡地应用被应用到存储器4108的特定部分的安全规则4122和保护。因而,例如,如果一组特定的安全规则4122应用到存储器4108的特定部分中的内容,则在把这样的内容传递到存储器4108的另一部分时,O/S下层安全代理4116可以更新安全规则4122以便应用到存储器4108的目的地部分。

[0538] 如上所述,O/S下层安全代理4116可以基于安全规则4122检测恶意代码的存在。O/S下层安全代理4116可以经由以上所描述的任何方法和/或以任何其他合适的方式检测恶意代码的存在。例如,O/S下层安全代理4116可以通过捕获对存储器4108或电子设备4104的其他资源的访问来检测恶意代码的存在。作为另一示例,O/S下层安全代理4116可以通过扫描存储器4108的页面和/或存储4126以便发现恶意代码来检测恶意代码的存在。作为进一步的示例,O/S下层安全代理4116可以通过从O/S内部安全代理4118接收O/S内部安全代理4118已经检测到恶意代码的存在的通信来在存储器中检测恶意代码的存在。

[0539] 响应于检测电子设备4104上的恶意代码(无论这样的代码是自修改代码还是其他恶意代码),O/S下层安全代理4116可以采取矫正动作,包括修改恶意代码。在此使用的恶意代码的“修改(modifying)”或“修改(modification)”可以包括但不限于如存储器4108中所

实现的恶意代码的修改、如存储4126中所实现的恶意代码的修改和/或恶意代码对存储器4108和电子设备4104的其他资源的访问的修改。恶意代码的修改是有益的，这是因为包括恶意代码的存储器4108的部分(例如，页面)可以属于恶意软件或甚至是没有意识到感染的程序。例如，这样的恶意代码可以被嵌入到字处理文档、操作系统内核的部分或恶意软件自身中。

[0540] 在修改存储器4108中包含的恶意代码时，O/S下层安全代理4116可以修改恶意代码以使得包括恶意代码的程序可以自我终止和/或把执行传递给可以使恶意代码失效(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的可信代码。例如，O/S下层安全代理4116可以把对操作系统4112的“exit(退出)”函数的调用插入到存储器4108中的恶意代码，以使得可以最终终止恶意代码的执行。作为另一示例，O/S下层安全代理4116可以把指令(例如，“JUMP(跳转)”指令)插入到存储器4108中的恶意代码，该指令可以把恶意代码的执行重定向到其中存储有可以使恶意代码失效(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的代码的已知、可信的部分的存储器4108的另一部分。作为进一步的示例，如果恶意代码目前正在执行，则O/S下层安全代理4116可以变更存储器4108中的指令指针值，以便引起把执行的控制权传递给可以使失效恶意代码(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的代码的已知可信部分。

[0541] 在一些实例中，不期望简单地终止由恶意代码实现的恶意进程。举例来说，终止和删除可以不适用于其中删除或终止具有不期望的副作用的操作系统的已感染部分或其他以另外方式可信的应用。因此，O/S下层安全代理4116可以修改恶意代码以使得修复恶意代码，允许已感染应用如同感染没有发生的那样有效地执行。例如，O/S下层安全代理4116可以用已知的或可信的代码替换恶意代码。作为特定的示例，如果操作系统的已知部分在特定存储器页面受到感染，则O/S下层安全代理4116可以用操作系统的这样的部分的已知代码重写该特定存储器页面。在一个实施例中，O/S下层安全代理4116可以从保护服务器获得替代页面。可以按需产生这样的替代页面，或可以将其配置为替换操作系统组件、驱动程序或其他模块的已知部分。替代地，O/S下层安全代理4116可以修改存储器4108中的指令指针，以使得在另一存储器位置处的受感染代码的经恢复版本处继续执行。

[0542] 在修改存储4126中包含的恶意代码时，O/S下层安全代理4116可以变更或删除恶意代码。例如，通过捕获在存储器4108和存储4126之间且反之亦然的内容传递，O/S下层安全代理4116可以收集关于被存储在存储器4108中的内容与被存储在存储4126中的相应内容的关系的信息并将其存储在日志、列表、高速缓存或其他数据结构中。因此，如果O/S下层安全代理4116标识存储器4108中的恶意代码，则它可以引用所收集的关于被存储在存储器4108中的内容与被存储在存储4126中的相应内容的关系的信息并修改具有恶意代码的、对应于存储器4108的位置的存储4126的位置中的内容。这样的修改可以包括但不限于删除存储4126中的相应内容，或修改内容以引起存储4126和/或存储器4108中的恶意代码的自我终止或删除。

[0543] 在修改恶意代码对存储器4108和电子设备4104的其他资源的访问时，O/S下层安全代理4116可以拒绝恶意代码片段对存储器4108或电子设备4104的其他资源的任何访问。这样拒绝存储器4108和其他资源可以引起包括恶意代码的进程崩溃或以另外方式使其失

效。例如，一旦已经标识恶意代码，则O/S下层安全代理4116可以捕获包括恶意代码的进程对存储器4108或电子设备4104的资源的已尝试访问并拒绝这样的访问。

[0544] 根据以上所描述的修改技术中的一些，O/S下层安全代理4116可以在保证恶意代码完好的同时使得恶意代码失效。在这样的场景中，O/S下层安全代理可以隔离恶意代码并将其作为取证迹象传递给保护服务器202以供进一步分析。然后，保护服务器202可以产生与恶意代码相关联的签名(例如，散列或指纹)、产生与签名相关联的政策或黑名单条目并把这样的信息传输给在其他电子设备上执行的安全代理。

[0545] 在一些实施例中，存储器4108的特定部分中存在的恶意代码的标识可以允许O/S下层安全代理4116标识具有恶意代码的存储器4108的其他部分。例如，在检测到表现出类似恶意软件的行为的线程时，O/S下层安全代理4116可以确定线程的执行地址和/或恶意代码在存储器页面内的位置。在虚拟存储器配置中，可以连续地列出应用代码，而在物理存储器中，应用代码基本上不是连续的。因而，通过利用由操作系统维护的在存储器4108中的物理存储器地址和存储4126中的虚拟存储器地址之间的映射，O/S下层安全代理4116可以标识与已标识恶意代码连续的、也可能包括恶意代码的虚拟存储器的部分，并把这样的虚拟存储器部分映射回到可能受感染的物理存储器地址。因此，O/S下层安全代理可以进一步监视在这样的物理存储器地址处的代码的执行以便发现恶意代码的存在。

[0546] 另外，O/S下层安全代理4116也可以被配置为监视与线程的执行和/或线程对资源的使用相关的活动，并基于这样的监视确定在各种线程之间的关系。因此，当特定线程被标识为恶意时，O/S下层安全代理4116可以确定与恶意线程相关线程(例如，起源线程、派生线程、同胞线程等等)并相对于除了恶意线程以外还有相关线程采取矫正动作。

[0547] 为了执行这样的监视和跟踪，O/S下层安全代理4116可以监视对存储器4108、存储4126、网络244和/或电子设备4104的其他资源的访问；监视操作系统服务、与线程执行和/或线程对资源的使用相关的调用和/或函数；和/或使用在此描述的技术中的一种或多种来检测可疑行为。例如，O/S下层安全代理4116可以实现图12的O/S下层安全代理1216的功能性，以便捕获(例如，基于安全规则4122)对读、写和/或执行存储器4108、存储4126和/或电子设备4104的其他组件上的代码的已尝试访问、对权限标志4136的已尝试改变和/或可以(例如，单独地或与其他存储器访问一起整体地)指示可疑行为的其他已尝试访问，并把关于这样的尝试访问信息记录到历史4140。

[0548] 作为进一步的示例，O/S下层安全代理4116可以实现O/S下层安全代理712、微代码安全代理708和/或O/S下层捕获代理920的功能性，以便捕获(例如，基于安全规则)意图的操作系统服务、与线程执行和/或线程对资源的使用相关的、可以证明可疑行为的调用和/或函数，并把关于这样的尝试访问的信息记录到历史4140。另外，在一些实施例中，O/S内部安全代理4118可以被配置为捕获可以证明可疑行为的线程执行和/或线程对资源的使用的操作系统4112的用户模式函数或内核模式函数，并把关于这样的尝试访问的信息记录到历史4140和/或把这样的信息传输给O/S下层安全代理4116。

[0549] 为了判断各线程之间的关系，O/S下层安全代理4116可以从存储器的角度监视对操作系统的线程同步对象的已尝试访问。为了阐释，初始线程可以产生第二线程，然后，第二线程开始操作(且变成该进程的主线程)，同时初始线程终止自身。作为另一阐释，线程可以操作为通过进程间通信(IPC)调用相互创建、终止或挂起。因而，线程可以跨越多个进程，

且一个进程中的线程可以对其他进程中的线程做出IPC调用，以便创建、终止或挂起。O/S内部安全代理4118可以通过捕获用于发起这样的IPC调用的操作系统调用(例如，在Windows™实施例中，诸如NTCreateThread、NTSuspendThread或NTTerminateThread之类的调用)来跟踪IPC调用。

[0550] 然而，使用O/S内部安全代理捕获这样的IPC调用可能被恶意软件危害或规避。因此，O/S下层安全代理4118可以通过捕获对与发起IPC调用相关联的存储器或处理器资源的已尝试访问来监视这样的尝试访问。例如，O/S下层安全代理4116可以实现图9的O/S下层捕获代理920的功能性，以便捕获对与发起IPC调用相关联的处理器资源的已尝试访问。作为另一示例，O/S下层安全代理4116实现图12的O/S下层安全代理1220的功能性，以便捕获对这样的IPC调用的可执行代码存储在其中的存储器位置(例如，页面或地址)的已尝试访问。在捕获与IPC调用相关联的事件时，O/S下层安全代理4116可以把关于这样的事件的信息(例如，线程标识符)记录到历史940。

[0551] 为了标识与IPC调用相关联的线程，O/S下层安全代理4116可以访问一个或多个处理器资源(例如，诸如在图7中被标识为系统资源724的那些)以便获取关于特定线程的信息。例如，对与在Windows™操作系统中执行的线程，处理器寄存器(例如，FS寄存器)可以指向每一处理器的存储器中被称为处理器控制块(PCB)的结构。PCB包括由线程调度程序用来管理处理器上的线程的信息，包括用于当前在处理器上执行的线程的ETHREAD数据结构以及用于已调度线程的ETHREAD列表。与线程相关联的ETHREAD数据结构可以包括多个元数据字段，包括线程的标识符。因此，在对Windows™应用保护时，O/S下层安全代理4116可以访问处理器资源中的信息以判断处理器的PCB的存储器位置，然后，访问PCB以便获得特定线程的ETHREAD信息。

[0552] 基于被存储在历史4140中的关于IPC调用的信息，O/S下层安全代理4116可以分析历史4140以判断在各种线程当中的关系。在其分析期间，O/S下层安全代理4116可以查阅规则4122以判断历史4140中陈述的线程行为是否指示在两个或更多个线程当中的关系。因此，如果判断特定线程或其宿主应用是恶意的，则O/S下层安全代理4116可以确定与特定线程相关的一个或多个线程并相对于这样的相关线程采取矫正动作。例如，矫正动作可以包括O/S下层安全代理4116检查、扫描和/或分析这样的线程(例如，使用在本公开内容中的其他地方描述的一种或多种技术)以判断这样的相关线程是否包括恶意代码。作为另一示例，如果判断这样的线程是恶意的，则矫正动作可以包括O/S下层安全代理4116终止、删除、修改或以另外方式中和这样的一个或多个相关线程(例如，使用在本公开内容中的其他地方描述的一种或多种技术)。作为附加示例，矫正动作可以包括O/S下层安全代理4116把与特定线程及其相关线程相关联的取证迹象传输给保护服务器202以供进一步分析。保护服务器202可以分析信息并把关于要采取的任何附加矫正动作的指令(例如，以安全规则4122的形式)传输给电子设备4104。作为进一步的示例，O/S下层安全代理4116可以尝试修复包括恶意线程的存储器的部分(例如，页面、存储器地址等等)。

[0553] 为了执行这样的修复，O/S下层安全代理4116可以不时产生存储器4106或其特定部分(例如，存储操作系统、安全应用或关键驱动程序的存储器的部分)的快照，并存储这样的快照(例如，存储在历史4140中)。快照可以与诸如日期和时间快照、与快照相关联的实体(例如，操作系统、应用或驱动程序)、与存储器页面相关联的线程标识符、虚拟存储器中存

储器的地址位置等等的上下文信息一起存储。如果定位了恶意线程或线程族，则可以至少部分地基于与快照相关联的上下文信息通过用适当的快照替换具有恶意线程的存储器的部分来修复包括这样的线程的存储器的部分。在一些实施例中，O/S下层安全代理4116也可以（例如，在历史4140中）记录由可信实体在产生快照之后对快照的存储器位置做出的改变，以使得响应于检测到恶意软件对快照反转将不撤销合法改变。

[0554] 因为如果被应用于所有线程则相关线程和线程的潜在恶意行为的这样的监视可以消耗电子设备的处理资源的显著部分，O/S下层安全代理4116可以仅在发生可以指示特定存储器位置易感染恶意软件和/或特定存储器存储重要的或关键的代码或数据（例如，操作系统或安全应用）的已尝试访问时执行这样的监视。例如，O/S下层安全代理4116可以在捕获对特定存储器位置的权限（如权限标志4136中所实现的）的修改时开始对特定存储器位置监视线程行为和线程关系。

[0555] 图42是用于调节和控制电子设备上的自修改代码的方法4200的示例实施例。在步骤4205，O/S下层安全代理可以捕获对存储器的已尝试访问，其中，每一这样的尝试访问可以单独地或集体地指示自修改恶意软件的存在。可以根据安全规则确定所捕获的已尝试访问。潜在地指示恶意软件的已尝试访问可以包括但不限于对存储器权限的改变、把内容从一个存储器位置复制到另一存储器位置、修改存储器位置的内容和执行存储器位置。

[0556] 在步骤4210，O/S下层安全代理可以把关于已捕获的已尝试访问的信息记录在历史中（例如，已尝试访问的类型、与已捕获的已尝试访问相关联的存储器位置等等）。因为如果被应用到每一存储器位置则对历史的这样的记录可以消耗电子设备的处理资源的显著部分，O/S下层安全代理可以在发生可以指示特定存储器位置易感染恶意软件（例如，基于建立用于发起特定存储器地址的历史的触发事件的安全规则）的已尝试存储器访问时发起特定存储器位置的历史记录。例如，O/S下层安全代理可以在捕获对特定存储器位置的权限（例如，如存储器位置的权限标志中所实现的）的修改时开始记录特定存储器位置的历史。

[0557] 在步骤4215，O/S下层安全代理可以监视（例如，根据安全规则）可以触发对特定存储器位置的历史的分析的启动的已尝试访问。因为如果被应用到每一存储器位置和/或已尝试访问，则分析历史以判断是否已经发生可疑行为可以消耗电子设备的处理资源的显著部分，O/S下层安全代理可以在发生与特定存储器位置相关联的特定的已捕获的已尝试访问对特定存储器位置发起对历史的分析（例如，在下面的步骤4220中）。例如，在一些实施例中，O/S下层安全代理可以在捕获对特定存储器位置处的内容的已尝试访问时触发对与特定存储器位置相关联的历史的分析的启动。

[0558] 在步骤4220，O/S下层安全代理可以分析历史，以便对特定存储器位置判断是否已经发生可疑行为。在其分析期间，O/S下层安全代理可以查阅安全规则，以便判断历史中实现的特定存储器位置的行为是否指示可以证明自修改恶意软件代码的潜在存在的可疑行为。例如，如果对历史的分析指示在第一存储器位置处的内容被复制到第二位置、在第二位置处被修改且然后已经发生对第二位置的内容的已尝试执行，则这样的指示可以是自修改恶意软件代码的潜在存在的迹象。作为另一示例，如果对历史的分析指示具有在第三位置处的共同起源的、在第一存储器位置和第二存储器位置处的内容都是已尝试执行的目标，则这样的指示可以是自修改恶意软件代码的潜在存在的迹象。作为进一步的示例，如果对历史的分析指示在特定存储器位置处的内容具有在多个其他存储器位置处的起源，则这样

的指示可以是自修改恶意软件代码的潜在存在的迹象。

[0559] 在步骤4225，0/S下层安全代理可以判断是否已经检测到证明潜在恶意软件的存在的可疑行为(例如，基于单个已捕获的已尝试访问或对历史的分析)。如果已经检测到可疑行为，则方法4200可以进行到步骤4230。否则，方法4200可以再次进行到步骤4205。在步骤4230，响应于检测到证明潜在的恶意软件的存在的可疑行为(例如，基于单个已捕获的已尝试访问或对历史的分析)，0/S下层安全代理可以发起矫正动作(例如，根据安全规则4122)。例如，在一些实施例中，0/S下层安全代理可以在检测到可疑行为时把在与所检测的可疑行为相关联的特定存储器位置处存储的内容与已知的恶意软件和/或已知的可信/可靠的进程进行比较，以判断该内容是不是恶意的。可以通过把内容的散列、指纹或其他签名与已知进程的散列、指纹或其他签名进行比较来实现这样的比较。作为另一示例，0/S下层安全代理可以在检测到可疑行为时把与可疑行为相关联的取证迹象(例如，存储器位置的内容、与存储器位置相关联的历史等等)报告给保护服务器以供进一步分析。

[0560] 在步骤4235，0/S下层安全代理可以判断与可疑行为相关联的存储器位置的内容是不是恶意的(例如，通过把该内容与已知进程、从保护服务器接收到的信息、对安全规则的引用和/或其他判断进行比较)。如果内容是恶意的，则方法4200可以进行到步骤4240。否则，方法4200可以再次进行到步骤4205。在步骤4240，响应于判断内容是恶意的，0/S下层安全代理可以采取进一步矫正动作(例如，根据安全规则)。这样的矫正动作可以包括但不限于不允许执行内容、撤销对内容的改变(例如，如历史陈述的对内容的修改和复制)、修复内容、用无害内容替换该内容和/或禁用与该内容相关联的进程。在完成步骤4240之后，方法4200可以再次进行到步骤4205。

[0561] 图43是用于修改电子设备上的恶意代码的方法4305的示例实施例。在步骤4305，0/S下层安全代理可以检测电子设备上恶意代码的存在。例如，0/S下层安全代理可以通过捕获对电子设备的存储器或电子设备的其他资源的访问来检测恶意代码的存在。作为另一示例，0/S下层安全代理可以通过扫描存储器的页面和/或电子设备的存储以便发现恶意代码来检测恶意代码的存在。作为进一步的示例，0/S下层安全代理可以通过从检测到恶意代码的存在的0/S内部安全代理接收通信来检测存储器中恶意代码的存在。

[0562] 在步骤4310-4320，响应于在电子设备上检测到恶意代码，0/S下层安全代理可以采取矫正动作，包括修改恶意代码。举例来说，在步骤4310，0/S下层安全代理可以修改恶意代码以使得包括恶意代码的程序可以自我终止和/或把执行传递给可以使恶意代码失效(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的可信代码。例如，0/S下层安全代理可以把对操作系统的“exit”函数的调用插入到电子设备的存储器中的恶意代码，以使得可以最终终止恶意代码的执行。作为另一示例，0/S下层安全代理可以把指令(例如，“JUMP”指令)插入到电子设备的存储器中的恶意代码，该指令可以把恶意代码的执行重定向到其中存储有可以使恶意代码失效(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的代码的已知、可信的部分的存储器4108的另一部分。作为进一步的示例，如果恶意代码目前正在执行，则0/S下层安全代理可以变更存储器中的指令指针值，以便引起把执行的控制权传递给可以使失效恶意代码(例如，通过擦除和与恶意代码片段相关联的线程或进程相关联的所有代码和数据)的代码的已知可信部分。替代地，0/S下层安全代理可以修改恶意代码以使得以使得修复恶意代码，

允许已感染应用如同感染没有发生的那样有效地执行。通过用已知的或可信的代码替换恶意代码或修改存储器中的指令指针，以使得在另一存储器位置处的受感染代码的经恢复版本处继续执行。

[0563] 在步骤4315，O/S下层安全代理可以修改存储中包含的恶意代码。例如，通过捕获在电子设备的存储器和存储之间且反之亦然的内容传递，O/S下层安全代理可以收集关于被存储在存储器中的内容与被存储在存储中的相应内容的关系的信息并将其存储在日志、列表、高速缓存或其他数据结构中。因此，如果O/S下层安全代理标识存储器中的恶意代码，则它可以引用所收集的关于被存储在存储器中的内容与被存储在存储中的相应内容的关系的信息并修改具有恶意代码的、对应于存储器的位置的存储的位置中的内容。这样的修改可以包括但不限于删除存储中的相应内容，或修改内容以引起存储和/或存储器中的恶意代码的自我终止或删除。

[0564] 在步骤4320，O/S下层安全代理可以修改恶意代码对存储器和电子设备的其他资源的访问，例如，以便拒绝恶意代码片段对电子设备的存储器或其他资源的任何访问。对存储器和其他资源的这样的拒绝可以引起包括恶意代码的进程崩溃或以另外方式使其失效。例如，一旦已经标识恶意代码，则O/S下层安全代理可以捕获包括恶意代码的进程对存储器或电子设备的资源的已尝试访问并拒绝这样的访问。

[0565] 在步骤4325，O/S下层安全代理可以基于所检测的恶意代码的物理存储器地址标识潜在地具有恶意代码的存储器的其他部分。例如，在检测到表现出类似恶意软件的行为的线程时，O/S下层安全代理可以确定线程的执行地址和/或恶意代码在存储器页面内的位置。在虚拟存储器配置中，可以连续地列出应用代码，而在物理存储器中，应用代码基本上不是连续的。因而，通过利用由操作系统维护的在存储器中的物理存储器地址和存储中的虚拟存储器地址之间的映射，O/S下层安全代理可以标识对应于已标识的恶意代码、也可能包括恶意代码的虚拟存储器的部分，并把这样的虚拟存储器部分映射回到可能受感染的物理存储器地址。因此，O/S下层安全代理可以进一步监视在这样的物理存储器地址处的代码的执行以便发现恶意代码的存在。

[0566] 在步骤4330，O/S下层安全代理可以隔离恶意代码并将其作为取证迹象传递给保护服务器以供进一步分析。

[0567] 图44是用于监视和跟踪电子设备上的相关线程的方法4400的示例实施例。在步骤4405，O/S下层安全代理可以捕获对和与另一线程创建、挂起或终止一个线程相关联的线程同步对象的函数调用相关联的存储器或处理器资源的已尝试访问。例如，O/S下层安全代理可以捕获对与进程间通信(IPC)调用相关联的处理器资源的存储器的已尝试访问。在步骤4410，O/S下层安全代理可以把与这样的已捕获的已尝试访问相关联的信息(例如，线程标识符)存储到历史。

[0568] 在步骤4415，O/S下层安全代理可以捕获对存储器或处理器资源的已尝试访问，其中，每一这样的尝试访问可以单独地或集体地指示恶意软件的存在。可以根据安全规则确定所捕获的已尝试访问。潜在地指示恶意软件的已尝试访问可以包括但不限于对存储器权限的改变、把内容从一个存储器位置复制到另一存储器位置、修改存储器位置的内容和执行存储器位置。在步骤4420，O/S下层安全代理可以把与这样的已捕获的已尝试访问相关联的信息(例如，线程5标识符)存储到历史。在步骤4410和4420，O/S下层安全代理可以访问处

理器资源中的信息以判断线程元数据的存储器位置，且基于线程元数据，获得特定线程的线程标识符以便作为该信息的一部分存储在历史中。

[0569] 在步骤4425，O/S下层安全代理可以分析历史(例如，根据安全规则)以判断相对于特定线程是否已经发生与恶意软件感染一致的行为。在步骤4430，如果已经发生与恶意软件感染一致的行为，则O/S下层安全代理可以分析历史，以便确定与已经标识了恶意软件的特定线程相关的一个或多个线程。

[0570] 在步骤4435，O/S下层安全代理可以对特定线程和一个或多个相关线程采取矫正动作。例如，矫正动作可以包括O/S下层安全代理检查、扫描、和/或分析这样的线程以判断这样的相关线程是否包括恶意代码。作为另一示例，矫正动作可以包括如果判断这样的线程是恶意的则O/S下层安全代理终止、删除、修改或以另外方式中和这样的一个或多个相关线程。作为附加示例，矫正动作可以包括O/S下层安全代理把与特定线程及其相关线程相关联的取证迹象传输给保护服务器以供进一步分析。

[0571] 图45是用于保护电子设备4504的存储器和存储的系统4500的示例实施例。系统4500可以包括O/S下层安全代理4516，O/S下层安全代理4516被配置为在电子设备4504上操作以便防备对访问电子设备4504的存储器4508和存储4526的恶意尝试。此外，O/S下层安全代理4516可以被配置为使用一个或多个安全规则4522来判断要捕获什么已尝试操作以及如何响应这样的已捕获操作。O/S下层安全代理可以被配置为允许已捕获操作、拒绝已捕获操作或对已捕获操作采取其他矫正动作。

[0572] 如图45中所示出，电子设备4504可以包括被耦合到存储器4508的处理器4506、应用4510、驱动程序4511、操作5系统4512、操作系统下层安全代理4516、存储4526和应用资产4548。电子设备4504可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。处理器4506可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器902、图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。存储器4508可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的物理存储器1203或虚拟存储器和/或其任何组合实现，或者被配置为实现它们的功能性。应用4510可以全部地或部分地由图1的应用110、图2的应用210、图4的应用410、图7的应用709、图9的应用910、图12的应用1210和/或其任何组合实现，或者被配置为实现它们的功能性。驱动程序4511可以全部地或部分地由图1的驱动程序111、图2的驱动程序211、图4的驱动程序411、图7的驱动程序711、图9的驱动程序911、图12的驱动程序1211和/或其任何组合实现，或者被配置为实现它们的功能性。操作系统4512可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S下层安全代理4516可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全5代理4518可以全部地或部分地由O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理718、图9的O/S内

部安全代理919、O/S内部安全代理1219和/或其任何组合实现，或者被配置为实现它们的功能性。存储4526可以全部地或部分地由图4的存储426实现，或者被配置为实现它的功能性。

[0573] 安全规则4522可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、或图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。可以以任何合适的方式建立安全规则4122(例如，由电子设备4504的用户设置的政策、由包括电子设备4504的企业的管理员设置的政策、由O/S下层安全代理4516的创建者设置的政策等等)。在一些实施例中，O/S下层安全代理4516可以经由网络244请求和/或接收来自保护服务器202的对安全规则4522的更新或修改(例如，例如，由于对恶意软件定义的更新)。

[0574] O/S下层安全代理4516可以包括存储器跟踪设施4542、存储跟踪设施4544和存储器/存储安全层4546。存储器跟踪设施4542可以与存储器4508连接以便监视对存储器4508的访问。例如，存储器跟踪设施4542可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理442、图7的微代码安全代理708、图9的O/S下层安全代理920、和/或图12的O/S下层安全代理1220实现，或者被配置为实现它们的功能性，以便在应用4510、驱动程序4511和/或操作系统4512对读、写或执行存储器4508中的特定页面的已尝试访问(例如，如页面表标志和/或位标志所指示的)发生时捕获或触发。作为另一示例，存储器跟踪设施4542可以全部地或部分地由O/S下层安全代理712和/或微代码安全代理708实现，或者被配置为实现它们的功能性，以便在应用4510、驱动程序4511和/或操作系统4512对读、写或执行存储器4508中的特定地址的已尝试访问发生时捕获或触发。因此，存储器跟踪设施4542可以跟踪内容在存储器4508中从一个位置到另一位置的移动(例如，从一个页面到另一页面或从一个地址到另一地址)或在存储器4508和存储4526之间的移动(例如，连同虚拟存储器上下文交换或从存储4526加载可执行代码)。另外，存储器跟踪设施4542可以把关于所跟踪的移动的信息存储在日志、列表、高速缓存或存储器跟踪设施4542和/或存储器/存储安全层4546可访问的其他合适的数据结构。

[0575] 存储跟踪设施4544可以与存储4526连接以便监视内容从存储4526中的一个位置移动到另一位置或在存储器4508和存储4526之间移动。例如，存储跟踪设施4544可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理442、图7的微代码安全代理708、图9的O/S下层安全代理920和/或图12的O/S下层安全代理1220实现，或者被配置为实现它们的功能性，以便在应用4510、驱动程序4511和/或操作系统4512对读、写或执行存储4526中的特定扇区的已尝试访问发生时捕获或触发。作为另一示例，存储跟踪设施4544可以全部地或部分地由O/S下层安全代理712和/或微代码安全代理708实现，或者被配置为实现它们的功能性，以便在应用4510、驱动程序4511和/或操作系统4512对读、写或执行存储4526中的特定地址的已尝试访问时捕获或触发。因此，存储跟踪设施4544可以跟踪内容从存储4526中的一个位置移动到另一位置(例如，从一个扇区移动到另一扇区或从一个地址移动到另一地址)或在存储器4508和存储4526之间移动(例如，连同虚拟存储器上下文交换或加载来自存储4526的可执行代码一起)。

[0576] 在操作中，存储器/存储安全层4546可以接收安全规则4522并把安全规则4522传输给存储器跟踪设施4542和存储跟踪设施4544。因而，存储器跟踪设施4542和存储跟踪设施4544的监视可以基于安全规则4522，安全规则4522可以指示监视是否得到允许和/或标

识要监视存储器4508和/或存储4526的哪些部分。

[0577] 存储器跟踪设施4542和存储跟踪设施4544可以向存储器/存储安全层4546通知存储器4508和/或存储4526的已尝试访问(例如,对在存储器4508或存储4526内或在存储器和存储4526之间的内容的尝试移动)。存储器/存储安全层4546可以全部地或部分地由图1的O/S下层捕获代理104、图2的SVMM安全代理217或SVMM 216、图4的固件安全代理440、442、PC固件安全代理444或O/S系统下层代理450、图5的固件安全代理516、图7的微代码安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理和/或其任何组合实现,或者被配置为实现它们的功能性。存储器/存储安全层4546可以根据安全规则4522和/或应用资产4548分析由存储器跟踪设施4542和/或存储跟踪设施4544报告的存储器4508和/或存储4526的已尝试访问,以便判断这样的尝试访问是否指示恶意软件攻击,如下面更详细地描述的。在一些实施例中,存储器/存储安全层4546可以把如图45的活动4532和活动4534所指示的、存储器跟踪设施4542和存储跟踪设施4544所报告的访问的日志、列表或其他指示存储到存储器4508和/或存储4526。因而,除了分析单个对存储器4508和/或存储4526的已尝试访问之外,存储器/存储安全层4546可以根据安全规则4522分析活动4532和/或活动4534中所包括的活动的历史,以便判断该访问的历史行为是否指示恶意软件的存在。

[0578] 在特定的实施例中,存储器/存储安全层4546可以全部地或部分地由图2的SVMM 216实现,或者被配置为实现它的功能性,存储器跟踪设施4542可以全部地或部分地由图7的微代码安全代理708实现,或者被配置为实现它的功能性,且存储跟踪设施4544可以全部地或部分地由图4的固件安全代理442实现,或者被配置为实现它的功能性。在这样的实施例中,存储器跟踪设施4542可以捕获特定存储器访问,且存储跟踪设施4544可以捕获特定存储访问,且每一个都可以向存储器/存储安全层4546通知这样的已捕获事件。然后,存储器/存储安全层4546可以分析单个对存储器和/或存储的已尝试访问,和/或根据安全规则4522分析活动的历史,以便判断访问的历史行为是否指示恶意软件的存在。

[0579] 在另一特定的实施例中,存储器/存储安全层4546、存储器跟踪设施4542和存储跟踪设施4544中的每一个都可以全部地或部分地由图2的单个SVMM216实现,或者被配置为实现它的功能性。在这样的实施例中,SVMM 216可以捕获特定存储器访问、捕获特定的存储访问并分析对存储器和/或存储的个体已尝试访问,和/或根据安全规则4522分析活动的历史,以便判断访问的历史行为是否指示恶意软件的存在。

[0580] 应用4510、驱动程序4511、操作系统4512和/或另一实体的应用资产4548可以表示指示这样的实体及其组件如何驻留在存储器4508和/或存储4526内的映射、表、列表和/或其他数据结构。应用资产4548可以标识实体可以存储到其中的存储器4508和/或存储4526的部分(例如,存储器页面、存储器地址范围、盘扇区、盘地址分级等等)。如上所述,基于应用资产4548和/或安全规则4522,存储器/存储安全代理4516可以判断来自存储器跟踪设施4542和/或存储跟踪设施4544的关于对存储器4508和/或存储4526的已尝试访问的通知是否指示恶意软件攻击。例如,在其中应用4510是便携式可执行文件的实施例中,应用资产4548可以标识包括应用4510的可执行代码的、存储器4508和/或存储4526中所存储的应用4510的部分,和/或标识包括应用4510的数据(包括应用4510的组件存储在其中的存储器4508和/或存储4526的位置)的、存储器4508和/或存储4526中所存储的应用4510的部分。安全规则4522可以规定,对于应用4510的这样的示例,起源于不同于应用4510的程序的、对包

括应用4510的可执行代码的存储器4508和/或存储4526的部分的写访问指示恶意软件攻击。另外或替代地，安全规则4522可以规定，这样的示例对于应用4510的这样的示例，起源于不同于应用4510的程序的、对包括应用4510的数据的存储器4508和/或存储4526的部分的读或写访问指示恶意软件攻击。

[0581] 作为另一示例，在其中应用4510是字处理程序的实施例中，应用资产4548可以标识包括应用4510的可执行代码的、存储器4508和/或存储4526中所存储的应用4510的部分、标识存储器4508和/或包括脚本、映像、格式化文本、笔记和应用4510的其他数据(包括应用4510的组件存储在其中的存储器4508和/或存储4526的位置)的、存储4526中所存储的应用4510的部分。安全规则4522可以规定，对于应用4510的这样的示例，可以允许对包括应用4510的数据的存储器4508和/或存储4526的部分的读或写访问(例如，起源于操作系统、反恶意软件应用等等的访问)，并且不同于该组特定程序的程序的访问可以指示恶意软件攻击。

[0582] 应用资产4548可以由应用4510、驱动程序4511、操作系统4512和/或其他程序的创建者(例如，应用销售商、程序员或创建者)、电子设备4504的用户、包括电子设备4504的企业的管理员、O/S下层安全代理4516的创建者和/或另一合适的个体创建或定义。在一些实施例中，对于程序，应用资产4548可以包括在程序的存储上结构和程序的存储器中结构的之间的关系(例如，在存储器4508和存储4526的程序的组件之间的映射)。

[0583] 为了聚集应用资产4548，O/S内部安全代理4518和/或O/S下层安全代理4516可以使用任何数量的合适的技术。例如，O/S内部安全代理4518和/或O/S下层安全代理4516可以收集可以与由操作系统4512结合虚拟存储器操作产生的虚拟存储器页面交换相关联的信息。例如，在Windows™中，O/S内部安全代理4518可以访问原型页表项(PTE)并把这样的信息传输给O/S下层安全代理4516。在其他实施例中，O/S下层安全代理4516可以在执行访问的任何时刻为存储器4508中的页面和/或盘上的扇区4526产生散列、指纹或其他唯一标识符，并维护这样的标识符的高速缓存(例如，要存储在存储器4508和/或存储4526中的这样的高速缓存)。在这样的场景中，O/S下层安全代理4516可以应用简单比较以判断存储4526的哪一扇区被加载到存储器4508的哪一页面，且反之亦然。这样的映射可以允许安全代理4516和/或4518跟踪存储器4508和/或存储4526中的实体的特定信息的位置。

[0584] 因为如果被应用到对存储器4508和存储4526的所有访问则存储器/存储安全层4546、存储器跟踪设施4542和/或存储跟踪设施4544所执行的监视和分析可以消耗电子设备的处理资源的显著部分4504，可以仅在特定的已定义倾向中允许对存储器4508和存储4526的监视和分析。例如，在一些实施例中，安全规则4522可以提供，存储器跟踪设施4542和/或存储跟踪设施4544仅监视存储器4508和/或存储4526的特定部分(例如，包括操作系统或关键驱动程序或应用的那些部分)。作为另一示例，在相同的或替代的实施例中，安全规则4522可以提供，如果其他指示示出存储器4508和/或存储4526的特定部分中的程序是可疑的和/或其他指示示出恶意软件攻击已经发生，则存储器跟踪设施4542和/或存储跟踪设施4544监视该程序。作为进一步的示例，如同虚拟存储器上下文交换或加载来自存储4526的可执行代码的情况一样，存储器跟踪设施4542和/或存储跟踪设施4544可以除了在把内容从存储器4508加载到存储4526或反之亦然时之外放弃存储器捕获和存储捕获。

[0585] 在操作中，如上所述，存储器/存储安全层4546可以通过根据安全规则4522和/或

应用资产4548分析所报告的对存储器4508和/或存储4526的访问来保护存储器4508和/或存储4526。在接收对存储器4508和/或5存储4526的已尝试访问的通知之后，存储器/存储安全层4546可以确定请求已尝试访问的实体的身份(例如，操作系统451、驱动程序4511或应用4510)。例如，O/S内部安全代理4518可以从操作系统4512收集与特定存储器4508和/或存储请求4526的请求实体相关的上下文信息，并把这样的信息传输给存储器/存储安全层4546。另外，存储器/存储安全层4546可以验证请求实体的身份并判断该实体是否已被恶意软件危害(例如，通过把被存储在存储器中的实体的映射或散列与实体的已知预期映射或散列进行比较，或扫描被存储在存储器中的实体以便发现恶意软件的存在)。此外，存储器/存储安全层4546可以判断实体是否得到授权做出请求(例如，基于安全规则4522和/或应用资产4548判断是否授权实体访问存储器4508或存储4526的特定部分)。此外，存储器/存储安全层4546可以扫描与尝试访问相关的内容(例如，正在被读、写或执行的数据或可执行代码是)以便判断该内容是否包含恶意软件。而且，存储器/存储安全层4546可以判断对访问历史(例如，如活动4532和/或活动4534中所存储的)的行为分析是否指示恶意软件的存在(例如，未经授权的实体对O/S 4512的受保护部分的已尝试访问)。如果判断已尝试访问是恶意软件相关的，则存储器/存储安全层4546可以采取矫正动作。矫正动作可以包括阻止已尝试访问、终止请求实体、修复请求实体、把恶意软件相关事件的发生传输给保护服务器202和/或任何其他合适的动作。

[0586] 作为特定的示例，响应于对存储4526的特定扇区的请求(如由来自存储跟踪设施4544的通知所指示的)，存储器/存储安全层4546可以至少基于安全规则4522判断是否要访问该特定扇区。另外，存储器/存储安全层4546可以扫描与尝试访问相关的内容(例如，正在被读、写或执行的数据或可执行代码)以便判断该内容是否免遭潜在的恶意软件感染。判断是否内容。此外，存储器/存储安全层4546可以至少基于安全规则4522判断是否授权请求已尝试访问的实体访问特定的扇区。如果这样的判断指示已尝试访问不是恶意软件相关的，则存储器/存储安全层4546可以批准已尝试访问。

[0587] 作为另一特定的示例，响应于对存储器的特定页面的请求(如来自存储器跟踪设施4542的通知所指示的)，存储器/存储安全层4546可以扫描与已尝试访问相关的内容(例如，结合已尝试访问(例如，正在被读、写或执行的数据或可执行代码))以便判断该内容是否免遭潜在的恶意软件感染。另外，存储器/存储安全层4546可以至少基于安全规则4522判断是否授权请求已尝试访问的实体访问特定页面。此外，如果已尝试访问是从存储4526到存储器4508的传递，则存储器/存储安全层4546可以至少基于安全规则4522判断要从中传递内容的存储4526的特定部分是否可信源。如果这样的判断指示已尝试访问不是恶意软件相关的，则存储器/存储安全层4546可以批准已尝试访问。

[0588] 另外，当在存储器4508和存储4526之间、在存储器4508的不同部分之间或者在存储4526的不同部分之间传递内容时可以过渡地应用被应用到存储器4508或存储4526的特定部分的安全规则4522和保护。因而，例如，如果一组特定的安全规则4522应用到存储器4508的特定部分中的内容，则在把这样的内容传递到存储器4508的另一部分或传递到存储4526时，存储器/存储安全层4546可以更新安全规则4522以便应用到存储器4508或存储4526的目的地部分。

[0589] 图46是用于保护电子设备的存储器和存储的方法4600的示例实施例。在步骤

4605，存储器/存储安全层可以把安全规则传输给存储器跟踪设施和存储跟踪设施。因为在此公开的用于保护存储器和存储免遭恶意软件的系统和方法可以消耗显著的处理器、存储器和/或其他资源，期望仅在存储器或存储的特定位置尤其易受恶意软件攻击感染时采用这样的系统和方法。例如，如果存储器或存储的部分包括操作系统或安全应用的部分，或如果已经在电子设备上看见或检测到攻击的先前指示，则存储器或存储的部分易受恶意软件攻击感染。

[0590] 在步骤4610，存储器跟踪设施和存储跟踪设施可以根据安全规则监视访问。为了监视，存储器跟踪设施和存储跟踪设施可以在对从存储器/存储安全层接收的安全规则所标识的存储器或存储的特定部分时的已尝试访问(例如，意图的读、写或执行)发生时捕获或触发。

[0591] 在步骤4615，存储器跟踪设施和/或存储跟踪设施可以把对存储器和/或存储的已尝试访问的通知传输给存储器/存储安全层。

[0592] 在步骤4620，存储器/存储安全层可以确定请求在存储器/存储安全层的通知中所标识的已尝试访问的实体的身份(例如，操作系统、驱动程序或应用)。例如，与存储器/存储安全层通信的O/S内部安全代理可以从操作系统收集与特定存储器和/或存储请求的请求实体相关的上下文信息，并将这样的信息传输给存储器/存储安全层。

[0593] 在步骤4625，存储器/存储安全层可以验证请求实体的身份并判断该实体是否已经被恶意软件危害。例如，存储器/存储安全层可以把存储器中所存储的实体的映射或散列与实体的已知预期映射或散列进行比较。作为另一示例，存储器/存储安全层可以扫描存储器中所存储的实体以便发现恶意软件的存在。

[0594] 在步骤4630，存储器/存储安全层可以判断是否授权该实体做出请求。例如，存储器/存储安全层可以查阅安全规则和/或应用资产以便判断是否授权该实体访问存储器4508或存储4526的特定部分。在步骤4635，存储器/存储安全层可以分析与已尝试访问相关联的内容(例如，正在被读、写或执行的数据或可执行代码)。例如，存储器/存储安全层可以扫描与已尝试访问相关的内容以便判断该内容是否包含恶意软件。

[0595] 在步骤4640，存储器/存储安全层可以分析对存储器和/或存储的访问的历史。这样的历史可以在电子设备的存储器和/或存储中被存储为日志或列表。这样的分析可以包括对访问存储器和/或存储的历史的行为分析以判断该历史是否指示恶意软件的存在。

[0596] 在步骤4645，存储器/存储安全层可以判断(例如，基于步骤4620-4640中的一个或多个的分析和判断)由存储器跟踪设施和/或存储跟踪设施报告的对存储器和/或存储的已尝试访问是否指示该已尝试访问受到恶意软件影响。另外，如果存储器/存储安全层判断经修改内容受到类似恶意软件的行为影响，则存储器/存储安全层可以采取矫正动作(例如，移除、隔离和/或以另外方式使恶意软件失效的动作)。另外，在一些实施例中，存储器/存储安全层可以把关于类似恶意软件的行为的发生的信息(例如，取证信息)传输给保护服务器。

[0597] 在步骤4650，存储器/存储安全层可以把访问的通知添加到被存储在电子设备的存储器和/或存储上的访问的日志或列表。为了执行对访问历史的行为分析，存储器/存储安全层可以稍后访问已存储的日志或列表。在完成步骤4650之后，方法4600可以再次返回到步骤4605。

[0598] 图47是用于保护对在电子设备4701上执行的操作系统4713的对象的访问的系统4700的示例实施例。系统4700可以包括O/S下层捕获代理4720和已触发事件应对程序4722，O/S下层捕获代理4720和已触发事件应对程序4722被配置为在电子设备4701上操作以便检测在操作系统4713上执行的基于软件的实体访问对象4706和/或对象管理器4704的恶意尝试。此外，O/S下层捕获代理4720和已触发事件应对程序4722可以被配置为使用一个或多个安全规则4708来判断何时捕获对对象4706和/或对象管理器4704的访问以及如何应对与已捕获操作相关联的已触发事件。O/S下层捕获代理4720和已触发事件应对程序4722可以被配置为允许已触发事件、拒绝已触发事件或对已触发事件采取其他矫正动作。

[0599] 电子设备4701可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备4701可以包括被耦合到存储器4703的一个或多个处理器4702。处理器4702可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器902、图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备4701可以包括操作系统4713，操作系统4713可以包括O/S内部安全代理4719和用于管理对象4706的对象管理器4704。操作系统4713可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理4719可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418、图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219和/或其任何合适的组合实现，或者被配置为实现它们的功能性。安全规则4708可以由图1的安全规则114、图2的安全规则220、222、图4的安全规则420、434、436、438、图5的安全规则518、图7的安全规则707、721、723、图9的安全规则908、921、图12的安全规则1208、1221和/或其任何组合实现，或者被配置为实现它们的功能性。保护服务器4714可以全部地或部分地由图1的保护服务器102、图2的保护服务器202和/或其任何组合实现，或者被配置为实现它们的功能性。

[0600] 存储器4703可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的存储器1203和/或其任何组合实现，或者被配置为实现它们的功能性。存储器4703可以使用被配置为虚拟化对存储器4703的访问的虚拟存储器系统来实现。在虚拟存储器系统中，可以给在操作系统4713上执行的软件进程提供该进程可以将其看作是连续的存储器块的虚拟地址空间。实际上，虚拟地址空间可以跨越物理存储器的不同区域而散布。在进程请求访问存储器时，操作系统4713可以负责把进程的虚拟存储器地址映射到数据实际上存储在其中的存储器4703中的物理地址。虚拟地址空间可以被分割成被称为虚拟存储器页面的固定大小的连续的虚拟存储器地址块。页面表可以被用来存储从虚拟存储器页面到虚拟存储器页面存储在其中的存储器4703中相应的物理地址的映射。页面表可以包括各种访问权限，例如读、写和/或执行，以便指定对给定虚拟存储器页面授权的访问类型。在进程尝试以相关的虚拟存储器页面的访问权限不授权的方式来访问虚拟存储器地址时，可以拒绝该尝试。

[0601] O/S下层捕获代理4720可以由图1的O/S下层捕获代理104、图2的SVMM216、图4的固件安全代理440、442和/或PC固件安全代理444、图5的固件安全代理516和/或图7的微代码

安全代理708、图9的O/S下层捕获代理920、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。已触发事件应对程序4722可以由图1的已触发事件应对程序108、图2的SVMM安全代理217、图4的O/S下层代理450、图7的O/S下层代理712、图9的已触发事件应对程序922和/或其任何组合实现，或者被配置为实现它们的功能性。在各种实施例中，O/S下层捕获代理4720的功能性中的一些可以由已触发事件应对程序4722实现，并且/或者已触发事件应对程序4722的功能性中的一些可以由O/S下层捕获代理4720实现。此外，可以在相同的软件模块中实现O/S下层捕获代理4720和已触发事件应对程序4722。

[0602] 使用对象4706来表示与操作系统4713相关联的每一资源，可以把操作系统4713实现为对象面向环境。例如，操作系统4713可以具有对象，这些对象表示驱动程序4711、应用4710、进程、存储器4703、文件和/或文件夹、物理设备和/或由操作系统4713使用的任何其他资源。另外，在操作系统4713上执行的每一应用4710和/或其他软件进程执行也可以使用对象4706来表示由特定的应用4710或软件进程使用的资源。对象4706可以包括对象函数4730，对象函数4730对特定类型的对象4706来说是唯一的，且可以被用来操作特定对象4706的数据。对象4706可以由包括首部和主体的数据结构来表示。对象4706的首部可以包括所有对象公用的管理字段。这些字段可以由对象管理器4704用来管理对象4706。对象4706的首部可以包括，例如，标识对象4706的对象名称和/或安全描述符指定与对象4706相关联的访问权限。对象4706的主体可以包含对特定类型的对象4706来说是唯一的对象专用数据字段。

[0603] 对象管理器4704可以被配置为在操作系统4713中执行以便管理操作系统4713的对象4706。可以使用可以被用来管理对象4706各种对象管理器函数4726来实现对象管理器4704。例如，对象管理器函数4726可以包括函数，这些函数被用来创建、删除、修改数据和/或修改对象4706的设置。对象管理器函数4726可以使用一个或多个子函数4728来实现。微软Windows™操作系统的对象管理器4704所使用的对象管理器函数4726的示例可见表1。

[0604]

ObAssignObjectSecurityDescriptor	ObFindHandleForObject	ObQueryObjectAuditingByHandle
ObAssignSecurity	ObFreeObjectCreateInfoBuffer	ObQuerySecurityDescriptorInfo
ObAuditInheritedHandleProcedure	ObGetHandleInformation	ObQueryTypeInfo
ObCheckCreateObjectAccess	ObGetHandleInformationEx	ObQueryTypeName
ObCheckObjectAccess	ObGetObjectInformation	ObReferenceFileObjectForWrite
ObClearProcessHandleTable	ObGetObjectSecurity	ObReferenceObjectByHandle
ObCloseHandle	ObGetProcessHandleCount	ObReferenceObjectByName
ObCreateObject	ObGetSecurityMode	ObReferenceObjectByPointer
ObCreateObjectType	ObInheritDeviceMap	ObReferenceObjectEx

[0605]

ObDeassignSecurity	ObInitProcess	ObReferenceObjectSafe
ObDeleteCapturedInsertInfo	ObInitSystem	ObReferenceProcessHandleTable
ObDereferenceDeviceMap	ObInitializeFastReference	ObReferenceSecurityDescriptor
ObDereferenceObject	ObInsertObject	ObReleaseObjectSecurity
ObDereferenceObjectDeferDelete	ObIsLUIDDeviceMapsEnabled	ObSetDeviceMap
ObDereferenceObjectEx	ObIsObjectDeletionInline	ObSetDirectoryDeviceMap
ObDereferenceProcessHandleTable	ObKillProcess	ObSetHandleAttributes
ObDereferenceSecurityDescriptor	ObLogSecurityDescriptor	ObSetSecurityDescriptorInfo
ObDupHandleProcedure	ObMakeTemporaryObject	ObSetSecurityObjectByPointer
ObDuplicateObject	ObOpenObjectByName	ObShutdownSystem
ObEnumerateObjectsByType	ObOpenObjectByPointer	ObSwapObjectNames
ObFastDereferenceObject	ObPerfDumpHandleEntry	ObValidateSecurityQuota
ObFastReferenceObject	ObPerfHandleTableWalk	ObWaitForSingleObject
ObFastReferenceObjectLocked	ObQueryDeviceMapInformation	
ObFastReplace	ObQueryNameString	

[0606] 表1:微软Windows对象管理器函数的示例

[0607] 存储器映射4718可以由图12的存储器映射1206实现,或者被配置为实现它的功能性。存储器映射4718可以在文件、记录、数据结构或任何其他合适的实体中实现。存储器映射4718可以被包括为O/S下层捕获代理4720的部分,或者可以通信上耦合到O/S下层捕获代理4720。存储器映射4718可以包括关于各种对象资源4734在存储器4703中的位置的信息。对象资源4734可以包括,例如,对象管理器4704、对象管理器函数4726和/或子函数4728、对象4706和/或对象函数4730。存储器映射4718可以包括关于虚拟存储器中的存储器页面、物理存储器中的地址范围和/或特定的对象资源4734可以存储在其上的盘的位置的信息。O/S下层捕获代理4720可以被配置为使用存储器映射4718来判断虚拟存储器页面或物理存储器地址中的任何给定内容的身份或所有者。

[0608] O/S下层捕获代理4720可以确定、开发和/或增殖存储器映射4718的内容。为了这样做,O/S下层捕获代理4720可以访问安全规则4708、保护服务器4714或用于增殖存储器映射4718中的信息的任何其他合适的信息源。例如通过剖析操作系统4713的操作且然后判断各种对象资源4734位于存储器中何处,O/S下层捕获代理4720可以构建存储器映射4718。O/S下层捕获代理4720可以通过遍历各个函数的执行栈区结合O/S内部安全代理4719标识对象函数4730、对象管理器函数4726和/或对象管理器子函数4728在存储器中的位置。O/S下层捕获代理4720可以截取来自诸如操作系统4713、应用4710或驱动程序4711之类的在操作系统级别的实体对物理存储器或虚拟存储器的请求,以便在存储器映射4718中映射存储器的中所有权和内容。例如,当捕获到对访问对象资源4734的尝试时,O/S下层捕获代理4720可以被配置为与O/S内部安全代理4719通信,以便判断正在访问什么对象资源4734和/或什

么实体负责访问特定对象资源4734。O/S下层捕获代理4720可以与O/S内部安全代理4719通信，以便判断正在把什么实体加载到存储器中以使得可以增殖存储器映射4718。存储器映射4718可以包含用于物理存储器、虚拟存储器和/或在两者之间映射的存储器映射。

[0609] 因为对象面向操作系统4713的几乎所有资源可以由对象4706表示，恶意软件可以通过攻击对象4706尝试危害操作系统4713的安全。O/S下层捕获代理4720和/或已触发事件应对程序4722可以被配置为保护对象4706免遭恶意软件。恶意软件对对象4706的攻击可以包括对盗用对象资源4734的任何尝试，例如操纵对象4706和/或对象管理器4704的未经授权的尝试。例如，操作系统4713可以包括表示在操作系统4713上执行的每一软件进程的对象4706，且恶意软件可以删除与在操作系统4713上执行的安全应用相关联的特定的进程对象4706。以这种方式，安全应用的执行可以停止，这允许恶意软件破坏安全软件的防护设施并执行进一步的恶意活动。作为另一示例，恶意软件可以编辑其自己的对象4706的字段，例如对象名称，以便掩饰自身免受恶意软件扫描器。恶意软件也可以尝试修改对象4706的安全设置。例如，恶意软件可以尝试修改表示核心操作系统4713文件的对象4706的访问权限，以便允许文件可由任何实体访问。恶意软件可以通过调用对象管理器函数4726、对象管理器子函数4728和/或对象函数4730间接地执行上面所描述的攻击。恶意软件可以在调用函数之前改变对象的安全设置，以便隐蔽自己作为特定函数的调用者的身份。如果保护对象管理器函数4726免遭未经授权的执行但一个或多个子函数4728不受保护，则恶意软件可以调用对象管理器子函数4728而不是对象管理器函数4730。以这种方式，恶意软件可以通过调用对象管理器函数4726的一个或多个不受保护的子函数4728规避对象管理器函数4726的保护。恶意软件也可以通过访问对象4706和/或对象管理器4704存储在其中的存储器4703中的位置直接地攻击对象4706。

[0610] O/S下层捕获代理4720和/或已触发事件应对程序4722可以被配置为通过保护对访问对象资源4734的尝试来防止对对象4706的恶意软件攻击。例如，O/S下层捕获代理4720可以被配置为捕获对读、写或执行对象资源4734的尝试，且已触发事件应对程序4722可以被配置为判断已捕获尝试是否指示恶意软件。对访问对象资源4734的已捕获尝试可以包括对执行对象管理器4704的函数4726和/或子函数4728的尝试、对执行对象4706的函数4730的尝试和/或对直接访问对象资源4734存储在其中的存储器4703中的位置的尝试。O/S下层捕获代理4720的存储器捕获功能性可以由图12的O/S下层安全代理1220实现，或者被配置为实现它的功能性。

[0611] O/S下层捕获代理4720可以被配置为以任何合适的方式捕获对执行对象函数4730、对象管理器函数4726和/或对象管理器子函数4728的代码的尝试。例如，O/S下层捕获代理4720可以被配置为捕获对执行位于特定函数的代码可以存储在其中的存储器位置处的代码的尝试。O/S下层捕获代理4720可以被配置为查阅存储器映射4718以便标识要求捕获的函数在存储器4703中的位置。可以在虚拟存储器级别或物理存储器级别捕获对执行代码的已捕获尝试。例如，O/S下层捕获代理4720可以被配置为捕获对执行与特定函数的代码相关联的虚拟存储器页面的尝试。O/S下层捕获代理4720也可以被配置为捕获对执行位于对应于特定函数的代码可以存储在其中的物理存储器地址的虚拟存储器地址处的代码的尝试。这样的捕获可以在从虚拟存储器地址转换到物理存储器地址之前发生。在又一实施例中，O/S下层捕获代理4720可以被配置为捕获对执行位于特定函数的代码可以存储在其

中的物理存储器地址处的代码的尝试。这样的捕获可以在从虚拟存储器地址转换到物理存储器地址之后发生，或者也可以在直接尝试执行在物理存储器地址处的代码而无需首先通过通过虚拟存储器转换之后发生。在捕获对执行对象函数4730、对象管理器函数4726和/或对象管理器子函数4728的尝试之后，O/S下层捕获代理4720可以创建与已捕获尝试相关联的已触发事件并将其发送给已触发事件应对程序4722，以供应对已捕获尝试。

[0612] O/S下层捕获代理4720可以被配置为捕获对访问存储器4703中的对象资源4734的尝试。例如，O/S下层捕获代理4720可以被配置为捕获对访问被用来存储对象4706和/或对象管理器4704的存储器位置的尝试。O/S下层捕获代理4720也可以被配置为捕获对写入被用来存储对象函数4730、对象管理器函数4726和/或对象管理器子函数4728的代码的存储器位置的尝试。这样的捕获将防止恶意软件用恶意代码盖写对象函数4730、对象管理器函数4726和/或对象管理器子函数4728的代码。在一些实施例中，O/S下层捕获代理4720可以使用存储器映射4718来标识对象资源4734在存储器4703中的位置。在一个实施例中，O/S下层捕获代理4720可以被配置为捕获对访问对应于对象资源4734的虚拟存储器地址的虚拟存储器页面的尝试。在另一实施例中，O/S下层捕获代理4720可以被配置为捕获对访问对应于对象资源4734可以存储在其中的物理存储器地址的虚拟存储器地址的尝试。这样的捕获可以在从虚拟存储器地址转换到物理存储器地址之前发生。在又一实施例中，O/S下层捕获代理4720可以被配置为捕获对访问对象资源4734可以存储在其中的物理存储器地址的尝试。这样的捕获可以在从虚拟存储器地址转换到物理存储器地址之后发生，或者也可以在直接尝试访问物理存储器地址而无需首先通过虚拟存储器转换之后发生。

[0613] 在一个实施例中，O/S下层捕获代理4720可以被配置为查阅O/S内部安全代理4719以确定负责请求访问特定对象资源4734的请求实体。在另一实施例中，O/S下层捕获代理4720可以被配置为确定该请求所来自的虚拟存储器页面并查阅存储器映射4718以便判断这样的存储器页面是否与其中映射的任何元素相关联。在又一实施例中，O/S下层捕获代理4720可以被配置为确定请求元素的虚拟存储器页面的散列或签名并将其与已知实体的散列和签名进行比较。在捕获对访问对象资源4734的尝试并标识请求实体之后，O/S下层捕获代理4720可以创建包含与已捕获尝试相关联的信息的已触发事件，包括请求的特定对象资源4734、访问的类型和请求实体。O/S下层捕获代理4720可以把已触发事件发送给已触发事件应对程序4722以便应对已捕获尝试。

[0614] 已触发事件应对程序4722可以被配置为从O/S下层捕获代理4720接收与已捕获尝试相关联的已触发事件。已触发事件应对程序4722可以结合安全规则4708使用与已触发事件相关联的上下文信息，以确定要对已触发事件采取的适当动作。在一些实施例中，已触发事件应对程序4722可以与O/S内部安全代理4719协作以便标识与已触发事件相关联的的上下文信息。上下文信息可以包括已捕获尝试的请求实体、与已捕获尝试相关联的特定对象4706和/或相对于特定对象4706的所请求的访问的类型。安全规则4708可以指定，例如，与安全应用相关联的进程对象仅可以由安全应用自身删除。作为另一示例，安全规则4708可以授权来自操作系统4713的尝试创建新的对象4706，且可以要求O/S下层捕获代理4720捕获对访问新近创建的对象4706的将来的尝试。

[0615] O/S下层捕获代理4720可以被配置为监视对对象资源4734的访问以便创建表示操作系统4713的行为的行为状态映射4732。例如，O/S下层捕获代理4720可以通过捕获对访问

对象资源4734的尝试创建行为状态映射4732，并且更新行为状态映射4732以便表示每一已捕获操作。行为状态映射4732可以被用来实现行为分析系统4716，以便主动检测和阻止未知的零日恶意软件的攻击。可以在下面的图48的行为状态映射4802和行为分析系统4804的讨论中找到行为状态映射4732和行为分析系统4716的示例实施例的描述。

[0616] 图48是供与保护对操作系统的对象的访问的系统或方法一起使用的行为状态映射4802的示例实施例。例如，行为状态映射4802可以被实现为图47的行为状态映射4732，且可以由图47的行为分析系统4716、O/S下层捕获代理4720和/或已触发事件应对程序4722产生和/或利用。在面向对象环境中，操作系统及其全部资源，包括文件、应用、处理器、驱动程序和/或设备，可以被实现为对象。行为状态映射4802可以基于在操作系统的对象当中的操作和/或交互提供面向对象操作系统的行为的表示。行为状态映射4802可以被用于标识通常与恶意软件相关联的对象交互的模式。

[0617] 行为状态映射4802可以使用包括图和/或映射的任何合适的数据结构来实现。在使用图的一个实施例中，每一节点可以表示操作系统的对象，且在每一节点之间的边可以表示在各对象当中的操作和/或交互。例如，使用节点来表示操作系统对象并使用节点来表示进程对象，操作系统执行进程可以由行为状态映射表示。行为状态映射可以包括从操作系统对象到进程对象的边，该边表示该进程由操作系统执行。然后，如果进程打开文件，则行为状态映射可以被更新为包括节点表示特定的文件对象且可以包括从进程对象到文件对象的边，该边表示特定的文件由该进程打开。可以以这种方式为在各对象当中执行的操作连续更新行为状态映射。在一些实施例中，可以把行为状态映射实现为表示整个操作系统的行为，或者仅表示诸如在操作系统上执行的特定的应用、驱动程序和/或进程之类的操作系统的特定组件的行为。

[0618] 行为状态映射4802是与感染了恶意软件的操作系统相关联的的行为状态映射的示例实施例。行为状态映射4802包括表示操作系统4806、对象管理器4816、安全应用4808和恶意软件4810的对象的节点。从操作系统4806到安全应用4808的边表示操作系统4806执行安全应用4808，且从操作系统4806到恶意软件4810的边表示操作系统4806执行恶意软件4810。从操作系统4806到对象管理器4816的边表示操作系统4806创建对象管理器4816。与多条边相关联的恶意软件4810表示恶意软件4810所执行的恶意活动。从恶意软件4806到操作系统文件4814的两条边表示恶意软件4810打开操作系统文件4814并写到操作系统文件4814。作为示例，操作系统文件4814可以被用来指定在初始化操作系统4806时可以执行的应用，且恶意软件4810可以写入到这些文件以便把自身包括为这些应用中的一个。从恶意软件4810到安全应用4808的边表示恶意软件4810尝试终止安全应用4808。从恶意软件4810到系统调用表4812的边表示恶意软件4810写入系统调用表4812。恶意软件4810可以写入系统调用表4812，例如，以便修改特定的系统调用的条目。以这种方式，在执行系统调用时，恶意软件4810的恶意代码可取代预期系统调用来执行。从恶意软件4810到对象管理器4816的边表示恶意软件尝试调用对象管理器4810的特定函数。例如，恶意软件4810可以尝试通过调用对象管理器4816的删除对象函数删除操作系统4716的对象。行为状态映射4802仅表示行为状态映射的一个可能的实施例。可以以适于描绘操作系统的对象的操作和/或交互的任何方式实现行为状态映射4802。

[0619] 返回到图47，在一些实施例中行为状态映射4732可以由O/S下层捕获代理4720产

生。在其他实施例中，先前已经产生行为状态映射4732且可以将其用于主动检测和阻止未知的零日恶意软件的攻击。

[0620] 可以通过监视在操作系统4713的对象4706当中的交互和/或操作来产生行为状态映射4732。例如，O/S下层捕获代理4720可以捕获对访问对象资源4734的尝试，且可以更新行为状态映射4732以便反映每一已捕获操作。在一些实施例中，可以使用感染了恶意软件的操作系统4713来产生行为状态映射4732。在其他实施例中，可以使用免遭恶意软件的操作系统4713来产生行为状态映射4732。在一些实施例中，在产生了行为状态映射4732之后，可以分析它以便隔离与恶意软件相关联的行为模式和/或隔离安全行为的模式。在这样的实施例中，可以更新行为状态映射4732以便仅表示已隔离行为，或者可以创建新的行为状态映射以便仅表示已隔离行为。以这种方式，行为状态映射4732可以提供已知与恶意软件相关联的对象行为模型和/或已知是安全的对象行为模型。例如，如果在感染了恶意软件的操作系统4713上产生行为状态映射4732，则可以分析行为状态映射4732以便隔离恶意行为。通常由恶意软件执行的恶意行为包括修改核心操作系统文件、访问系统调用表和/或结束与安全应用相关联的进程以及其他。通过分析感染了恶意软件的操作系统4713的行为状态映射4732，可以在对象级别分析恶意行为。在对象级别分析恶意行为可以允许使得特定的恶意活动与在负责执行恶意活动的对象4706当中的操作模式相互关联。类似地，如果在免遭恶意软件的操作系统4713上产生行为状态映射4732，则行为状态映射4732可以被用来在对象级别分析安全行为，以便使得已知的安全行为与对象操作模式相互关联。

[0621] 在一些实施例中，行为状态映射4732可以用来主动检测和阻止未知的零日恶意软件的攻击。在这样的实施例中，先前已经产生行为状态映射4732，且它可以提供通常与恶意软件相关联的行为模型和/或已知是安全的行为模型。在这样的实施例中，行为状态映射4732可以由行为分析系统4716用来标识通常与恶意软件相关联的操作系统4713的行为。行为分析系统4716可以由O/S下层捕获代理4720实现，或者可以由已触发事件应对程序4722实现，或者在一些实施例中，行为分析系统4716的功能性可以部分地由O/S下层捕获代理4720且部分地由已触发事件应对程序4722实现。O/S下层捕获代理4720可以被配置为捕获对访问对象资源4734的尝试，且行为分析系统4716可以被用来判断已捕获尝试是否指示恶意软件。行为分析系统4716可以比较对行为状态映射4732的已尝试访问。在其中行为状态映射4732表示已知的安全行为的实施例中，行为分析系统4716可以从行为状态映射4732判断已捕获尝试是否匹配任何安全行为。如果发现匹配，则行为分析系统可以判断已捕获尝试是安全的且可以决定允许尝试。在其中行为状态映射4732表示与恶意软件相关联的行为的实施例中，行为分析系统4716可以从行为状态映射4732判断已捕获尝试是否匹配与恶意软件相关联的任何行为。如果发现匹配，行为分析系统4716可以判断已捕获尝试是不安全的且可以决定拒绝尝试。

[0622] 在一些实施例中，多个行为状态映射4732可以被使用。例如，O/S下层捕获代理4720可以包括当前行为状态映射和模型行为状态映射。当前行为状态映射可以表示操作系统4713的当前行为。模型行为状态映射可以是先前产生的提供通常与恶意软件相关联的模型行为和/或已知是安全的模型行为的状态映射。O/S下层捕获代理4720可以捕获对对象资源4734的已尝试访问且可以更新当前行为状态映射以便反映该已尝试访问。然后，行为分析系统4716可以把模型行为状态映射与当前行为状态映射进行比较。以这种方式，行为分

析系统4716可以结合先前行为从当前行为状态映射分析已捕获尝试，以判断已捕获尝试是否与恶意软件相关联。这可以允许行为分析系统4716更有效地评估已捕获尝试。

[0623] 图49是用于保护对操作系统的对象的访问的方法4900的示例实施例。在步骤4905，可以认证O/S下层安全代理、O/S内部安全代理、已触发事件应对程序和保护服务器的身份和安全。可以使用任何合适的方法来执行这样的认证，包括通过定位和检验每一组件在存储器中的映像、使用密码散列和/或使用密钥。直到步骤4905完成之前，可以停止其他步骤的操作。在步骤4910获得安全规则。安全规则可以由O/S下层安全代理、O/S内部安全代理和/或已触发事件应对程序本地存储，和/或可以远程存储，例如在保护服务器上。在步骤4915-4945，这样的安全规则可以被用来做出判定。

[0624] 在步骤4915，可以截取对访问与操作系统的对象相关联的资源的尝试。与操作系统的对象相关联的资源可以包括，例如，对象管理器、对象管理器函数和/或子函数、对象自身和/或对象的函数。已截取尝试可以包括对执行存储对象函数、对象管理器函数和/或对象管理器函数的子函数的存储器的位置处的代码的尝试。已截取尝试也可以包括对访问对象和/或对象管理器存储在其中的存储器中的位置的尝试。在一些实施例中，可以在虚拟存储器级别在把虚拟存储器地址转换到物理存储器地址之前截取尝试。在其他实施例中，可以在物理地址级别截取尝试。在一些实施例中，存储器映射可以被用来把对象资源在存储器中的位置指定为受到保护。

[0625] 在步骤4920，标识已截取尝试的请求实体。例如，已截取尝试可以是来自应用、驱动程序、O/S内部安全代理、操作系统，/或其他软件实体。在一些实施例中，可以通过查询包含在操作系统上执行的实体的地址的存储器映射来标识请求实体。

[0626] 在步骤4925，可以更新操作系统的当前行为状态映射。当前行为状态映射可以是基于在操作系统的对象当中的交互和/或操作描绘操作系统的行为的数据结构。对于每一已截取的对访问对象资源的尝试，可以更新当前行为状态映射以便反映与已截取尝试对应的操作。在步骤4930，把当前行为状态映射与模型行为状态映射进行比较。模型行为状态映射可以表示通常与恶意软件相关联的行为和/或通常已知是安全的行为。比较可以允许标识与恶意软件相关联的对象操作模式或可以允许标识已知是安全的对象操作模式。

[0627] 在步骤4935，判断是否授权已截取尝试。如果基于步骤4930当前行为状态映射与模型状态映射的比较标识了恶意软件，那么，不可以授权尝试。如果在步骤4930的比较没有标识恶意软件，那么，可以结合与已截取尝试相关联的上下文信息的使用安全规则来判断是否授权特定的尝试。上下文信息可以包括已截取尝试的请求实体、与已截取尝试相关联的特定对象和/或所请求的访问的类型。例如，安全规则可以指定，与安全应用相关联的进程对象仅可以由安全应用自身删除。如果判断授权了尝试，那么，在步骤4940可以允许访问。如果不授权尝试，那么，在步骤4945可以拒绝访问。

[0628] 根据保护电子设备的需要，可以连续地、周期性地、按需和/或在触发事件时重复图49的方法的步骤。

[0629] 图50是用于保护在电子设备5001上的驱动程序之间的通信的系统5000的示例实施例。系统5000可以包括O/S下层安全代理5020，O/S下层安全代理5020被配置为在电子设备5001上操作以便检测对截取或破坏在诸如电子设备5001上的操作系统5012之类的操作系统的驱动程序之间的通信的恶意尝试。此外，O/S下层安全代理5020可以被配置为使用一

个或多个安全规则5008来判断，例如，要捕获什么所尝试的驱动程序间通信、要捕获对驱动程序间通信设施的什么已尝试访问或基于尝试和所涉及的实体是否授权该尝试。O/S下层安全代理5020可以被配置为允许、拒绝或采取已捕获尝试的其他矫正动作。

[0630] 电子设备5001可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备5001可以包括被耦合到诸如存储器5003之类的存储器的一个或多个处理器5002。处理器5002可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器1202、图12的处理器1202、或其任何组合实现，或者被配置为实现它们的功能性。存储器5003可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、物理存储器1203或图12的虚拟存储器1204和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备5001可以包括操作系统5012，操作系统5012可以包括被耦合到一个或多个安全规则5021的O/S内部安全代理5019。操作系统5012可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理5019可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418和/或图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0631] O/S下层安全代理5020可以由图1的O/S下层捕获代理104或已触发事件应对程序108、图2的SVMM 216或SVMM安全代理217、图4的固件安全代理440、442、O/S下层代理450或PC固件安全代理444、图5的固件安全代理516或图7的微代码安全代理708或O/S下层代理712、图9的O/S下层捕获代理920或已触发事件应对程序922、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。

[0632] 安全规则5008可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则5021可以是由图2的安全规则220、图4的安全规则420、图7的安全规则721、图9的安全规则921、图12的安全规则1221和/或其任何组合实现，或者被配置为实现它们的功能性。

[0633] 电子设备5001可以包括一个或多个应用、驱动程序或其他实体——例如，应用5026或驱动程序2 5028——其可以尝试访问电子设备5001的资源以便与诸如驱动程序5029之类的驱动程序或驱动程序5029的驱动程序间通信设施通信或使用它们。应用5026或驱动程序2 5028可以包括任何进程、应用、程序或驱动程序。应用5026或驱动程序2 5028可以尝试调用驱动程序5029，且因而发起得到在驱动程序5029和另一驱动程序之间的驱动程序间通信的一系列调用。应用5026或驱动程序2 5028可以直接地或通过调用其他例程来尝试访问驱动程序间通信设施。应用5026或驱动程序2 5028可以尝试调用驱动程序子函数5030。可以通过对存储器5003的已尝试读、写或执行操作结合处理器5002来尝试对驱动程序间通信设施的所有这样的调用和访问。操作系统下层安全代理5020可以被配置为截取对驱动程序间通信设施的的这样的已尝试调用或访问、查阅来自O/S内部安全代理5019的安全规则5008和/或上下文信息以便判断该尝试是否指示恶意软件，并采取任何适当的矫正

动作。操作系统下层安全代理5020可以被配置为通过捕获对存储器5003访问和/或对处理器5002的使用做出这样的截取。操作系统下层安全代理5020可以被配置为访问安全规则5008并判断将捕获对存储器5003的什么已尝试访问和/或对处理器5002的什么使用。操作系统下层安全代理5020可以被配置为在控制结构中设置对应于要捕获的动作的标志。

[0634] 在一个实施例中，应用5026或驱动程序2 5028可以尝试通过存储器页面访问与驱动程序间通信相关联的存储器5003的部分，其中，存储器5003已经被操作系统5012虚拟化。在这样的实施例中，O/S下层安全代理5020可以被配置为基于存储器页面捕获对存储器5003的已尝试访问或执行。在另一实施例中，应用5026或驱动程序2 5028可以尝试访问与驱动程序间通信相关联的存储器5003的物理部分。在这样的实施例中，O/S下层安全代理5020可以被配置为基于存储器地址捕获对存储器5003的已尝试访问或执行。

[0635] 电子设备5001的操作系统5012和驱动程序可以提供用于驱动程序间通信的设施。例如，诸如NTFS.SYS 5031之类的驱动程序可以包含分派例程指针5032、导出地址表5034、导入地址表5036或快速I/O例程指针5038。分派例程指针5032可以包括指向驱动程序的函数的指针，例如在诸如代码部分1、代码部分2或恶意软件代码部分5046之类的代码部分中实现的函数。导出地址表5034可以包括指向与函数相关联的代码部分的指针，其中，该指针可以由其他驱动程序用来调用驱动程序的函数。导入地址表5036可以包含用于一个或多个其他驱动程序、用于要调用的驱动程序的函数的指针的列表。这样的导入地址表5036可以是导入另一驱动程序的导出地址表的结果。可以为对驱动程序的数据的访问提供用于驱动程序间通信的这样的设施。驱动程序的这样的数据可以是性质专用的，且构成驱动程序。例如，NTFS.SYS 5031可以包括用于可以包含指向所有打开的文件的指针的打开文件应对列表5040的数据部分或结构或可以包含指向在操作系统5012内挂载的每一存储卷的指针的已挂载卷列表5042。驱动程序间通信设施可以倾向于受恶意软件攻击，且因而O/S下层安全代理5020可以捕获使用这些设施、它们下面的机制以及或诸如驱动程序数据之类的这些设施的目标的已尝试访问。

[0636] 图51是驱动程序间通信的示例阐释。应用5102可以尝试做出请求，例如对网络接口卡(“NIC”)5116的网络请求或对盘5128的文件请求。为了实现这样的设备，可以通过操作系统5104处理该请求。操作系统5104的输入和输出请求可以由操作系统输入/输出管理器5106处理。

[0637] 操作系统输入/输出管理器5106可以通过调用一系列驱动程序中可用的函数来发送网络请求。操作系统输入/输出管理器5106可以调用SOCKET_DRIVER_AFS.SYS 5108，SOCKET_DRIVER_AFS.SYS 5108可以调用类型需求接口(“TDI”)协议驱动程序5110，类型需求接口(“TDI”)协议驱动程序5110可以调用网络驱动程序接口规范(“NDIS”)驱动程序5112，网络驱动程序接口规范(“NDIS”)驱动程序5112又可以调用NDIS.SYS驱动程序，NDIS.SYS驱动程序对NIC卡5116的硬件来说是专用的。应答可以遵循相同的一连串驱动程序到应用5102。

[0638] 同样地，操作系统输入/输出管理器5106可以通过调用一系列驱动程序中可用的函数发送文件请求。操作系统输入/输出管理器5106可以调用附加有文件系统过滤驱动程序5118的文件系统驱动程序5120，文件系统驱动程序5120又可以调用附加有盘过滤驱动程序5122的盘驱动程序5124，盘驱动程序5124又可以调用对盘5128来说是专用的盘驱动程

序，例如DISK.SYS，DISK.SYS可以应对到盘5128的物理输入和输出。应答可以遵循相同的一连串驱动程序到应用5102。

[0639] 可以借助于系统范围内的调用或者通过由驱动程序自身指定的调用来实施在图51内的驱动程序和内核模块的每次调用。恶意软件可以尝试挂钩、破坏、劫持、欺骗或以另外方式攻击在图51中示出的每一元素之间的调用。因而，这些通信表示图50的O/S下层安全代理5020可以被配置为在这样的通信发生时以及在保护允许这样的通信的机制时要保护的示例通信。

[0640] 图52是电子设备的示例部分的附加阐释，诸如图50的O/S下层安全代理5020之类的O/S下层安全代理可以被配置为结合驱动程序间通信保护这样的示例部分。驱动程序间通信可以由起源于诸如应用5202之类的用户模式的请求引起，或由来自诸如驱动程序2 5204之类的另一驱动程序的请求引起。应用5202可以被配置为做出对电子设备的部分的设备请求命令5208。设备请求命令5208可以被系统服务分派表(“SSDT”)5210转换成设备函数5212。设备函数5212可以被配置为把对应于应用5202所做出的请求的I/O请求包(“IRP”)发送给与所讨论的设备相关联的驱动程序。在图52中，这样的驱动程序可以是驱动程序1 5206，驱动程序1 5206可以应对目录控制的I/O请求。可以由起源于诸如驱动程序2 5204之类的另一内核模式驱动程序请求引起驱动程序间通信。

[0641] 图50的O/S下层安全代理5020可以被配置为保护电子设备5001免遭恶意软件，恶意软件可以以任何合适的方式攻击驱动程序间通信。例如，恶意软件可以挂钩用于诸如命令IoCallDriver 5220之类的发送或接收IRP命令的函数。这样的恶意软件可以引起该函数误导预期用于已指派目标(例如包含驱动程序可以执行的系统定义函数的分派例程5209)的IRP。相反，恶意软件挂钩5224可以被安装在IoCallDriver 5220上以便截取请求。在恶意软件挂钩5224后面可以的恶意软件检查在把请求发送到其原始目的地IRP_MJ_READ 5226之前或在改为运行恶意代码5228之前检查该请求。因而，O/S下层安全代理可以被配置为保护包含用于IoCallDriver 5220的指令的存储器或检查IoCallDriver 5220的执行以便发现的对驱动程序或回调例程的恶意调用。O/S下层安全代理5020可以被配置为捕获所导出的函数或分派例程。O/S下层安全代理5020可以被配置为捕获在操作系统构造内用于这样的函数或例程的指针，以及函数自身的存储器位置的执行。例如，恶意软件可以尝试改变值导出地址表(下面进一步详述的“EAT”)中的指针的存储器位置，或尝试改变函数自身的代码部分的内容(例如，以便插入到恶意代码的“JMP”)。通过捕获访问对针和函数的访问，可以解码已捕获尝试以判断函数的调用者。

[0642] 在另一示例中，驱动程序1 5206可以维护由对自身来说专用的驱动程序15206提供的、可以由诸如驱动程序之类的其他实体调用的函数的EAT 5211。EAT5211可以包括指向用于执行所指派的函数的代码部分的位置的函数指针的列表或数组。恶意软件可以改变这样的指针的值，以使得EAT中的条目不再指向正确的代码部分。可以使得这些指针改为指向潜在地恶意的代码部分以使得在由另一驱动程序通过引用EAT 5211中的指针调用该驱动程序函数时，执行恶意代码。例如，EAT 5211可以正常地包含函数驱动程序1Fn1以及函数驱动程序1Fn2的指针，函数驱动程序1Fn1的指针可以指向驱动程序1Fn1代码部分5214，函数驱动程序1Fn2的指针通常可以指向驱动程序1Fn2代码部分5216。然而，恶意软件已经改变第二指针以使得驱动程序1Fn2现在指向恶意软件代码部分5218。因而，O/S下层安全代理可

以被配置为保护EAT 5211驻留在其中的存储器空间、截取写请求并拒绝对写EAT 5211的这样的已捕获尝试，除非验证了写入者。这样的验证可以包括，例如，驱动程序1 5206自身更新其函数。O/S下层安全代理也可以被配置为捕获对写、改变或设置EAT 5211的任何已尝试函数的执行。O/S下层安全代理也可以验证该尝试的调用者得到授权执行这样的函数且该调用者不破坏标准过程，例如通过调用未建档子程序以便改变EAT 5211。

[0643] 在又一示例中，诸如驱动程序2 5204之类的另一驱动程序可以导入驱动程序1 5206的EAT 5211并把表主控为与驱动程序1 5206的函数相关联的其自己的导入地址表(“IAT”)5222。使用IAT 5222，驱动程序2 5204可以被配置为调用驱动程序1 5206的函数。IAT 5222 5可以由操作系统加载器填写。恶意软件可以以多种方式影响IAT。可以改变IAT 5222中的值，以使得诸如驱动程序2Fn2之类的函数现在指向恶意的代码部分，例如恶意软件代码部分5218。因而，O/S下层安全代理可以被配置为保护IAT 5222驻留在其中的存储器空间，截取写请求并拒绝对写入IAT 5222的这样的已捕获尝试，除非验证了写入者。这样的验证可以包括，例如，操作系统加载器加载IAT 5222。O/S下层安全代理也可以被配置为捕获对写、改变或设置IAT 5222的任何已尝试函数的执行。O/S下层安全代理也可以验证该尝试的调用者得到授权执行这样的函数且该调用者不破坏标准过程，例如通过调用未建档子程序以便改变IAT 5222。

[0644] 在再一个示例中，一旦已经调用诸如驱动程序1Fn1之类的驱动程序函数，则诸如驱动程序1Fn1代码部分5212中的代码之类的代码可以开始执行。恶意软件可以重写或注入这样的代码部分的各部分，以使得在调用例程时，执行恶意代码。因而，O/S下层安全代理可以被配置为保护驱动程序函数的代码驻留在其中的存储器空间、捕获写请求并拒绝写入驱动程序的代码部分的这样的已捕获尝试，除非验证了写入者。这样的验证可以包括，例如，判断该写入起源于用补丁更新自身的驱动程序。O/S下层安全代理也可以被配置为捕获对写、改变或设置驱动程序函数的代码部分的任何已尝试函数的执行。O/S下层安全代理也可以验证该尝试的调用者得到授权执行这样的函数且该调用者不破坏标准过程，例如通过调用未建档子程序以便改变驱动程序函数的代码部分。

[0645] 在进一步的示例中，驱动程序函数的代码可以被恶意软件直接调用，而不是通过访问已授权的导入或导出地址表。因而，O/S下层安全代理可以被配置为保护诸如驱动程序1Fn2代码部分5216之类的驱动程序的函数的执行免遭恶意代码5228直接执行。这样的O/S下层安全代理可以捕获函数的执行。O/S下层安全代理可以通过判断什么驱动程序由操作系统在它们各自的IAT 5222中这样更新来从上下文信息判断什么驱动程序已经接收到执行驱动程序1 5206上的函数的权限。O/S下层安全代理可以判断从何处做出调用，且如果这样的位置不对应于已知的经授权驱动程序，则可以拒绝该尝试。在一个实施例中，图50的O/S内部安全代理5019可以作为驱动程序或驱动程序过滤器而寄存在操作系统中，以提供上下文信息。例如，rootkit驱动程序可以避免调用文件I/O的NTFS.SYS。O/S内部安全代理5019可以作为过滤器而寄存在NTFS.SYS上以便看到对NTFS.SYS做出或者从NTFS.SYS做出的所有调用，且然后，告知O/S下层安全代理关于什么函数调用(如果有的话)是被rootkit用于文件I/O。

[0646] 返回到图50，在诸如驱动程序5029和驱动程序2 5028之类的操作驱动程序中，可以通过任何合适的方法通信。O/S下层安全代理5020可以捕获这样的通信或对允许这样的

通信的机制的尝试改变。在一个实施例中，O/S下层安全代理5020可以捕获和评估对图52中所描述的驱动程序间通信的恶意干扰的任何示例。

[0647] 在一个示例中，这样的通信可以包括经由IRP发送的I/O控制代码。O/S下层安全代理5020可以捕获对应于用于经由IRP发送I/O控制代码的函数调用的代码的执行、确证是否授权发送器和根据要求采取任何矫正动作。

[0648] 在另一示例中，这样的通信可以包括调用驱动程序的函数的代码部分，例如代码部分1。O/S下层安全代理5020可以捕获已尝试执行代码部分1。O/S下层安全代理5020可以判断已尝试执行是否起因于使用合法的访问函数的方式的合法的源。O/S下层安全代理5020可以标识调用者，并判断该调用者是否已知，以及任何规则是否基于已确定的调用者的身份防止函数的执行。例如，代码部分2执行可以限于已知且拥有数字证书的驱动程序。O/S下层安全代理5020可以判断已经发起该访问的驱动程序2 502是否根据白名单已知是安全的并具有数字证书。在另一示例中，O/S下层安全代理5020可以判断是否通过驱动程序5029做出该调用，或者是否通过驱动程序5030的未建档子函数做出该调用而没有访问驱动程序5029(以及其中采取的可能的安全措施)。相关的示例可以是捕获例如应用5026对直接跳转或分支到代码部分1而不使用任何指定的驱动程序函数机制的尝试。即使通过扫描或其签名不知道应用5026是恶意的，这样的行为也是高度可疑的，并指示恶意软件，且因而O/S下层安全代理5020可以判断该访问指示恶意软件。

[0649] 在又一示例中，回调例程5044可以被寄存在驱动程序中，例如在驱动程序的数据空间中。可以触发回调例程以便在驱动程序或特定的驱动程序函数退出时执行。这样的回调例程5044可以是恶意的。因而，O/S下层安全代理5020可以通过检测对在存储器内的驱动程序5031的代码部分或数据部分的已尝试写入来捕获任何回调例程的已尝试创建。如果已知意图的写入者是恶意的，那么，可以拒绝该尝试。如果意图的写入者是未知的，那么，可以允许写入，但可以捕获回调例程自身的后续执行以便判断要执行的动作是否恶意的。例如，记录器可以安装其中可以把副本网络分组发送给恶意服务器的回调例程5044。可以观察和评估回调例程的后续行为以便发现恶意软件的附加指示。

[0650] 在再一个示例中，应用5026可以尝试从EAT的5034读取地址，且然后，直接地执行相应函数。O/S下层安全代理5020可以捕获对EAT 5034的已尝试读取，并判断是否授权读取者做出对诸如代码部分1之类的函数的这样的尝试和后续执行。这样的尝试可以指示恶意软件已经尝试直接地读取EAT 5034而不是使用由操作系统5012提供的标准化方法，例如寄存为相依的驱动程序并通过其自己的导入地址表接收函数指针列表。

[0651] 在进一步的示例中，驱动程序2 5028可以尝试直接地操纵诸如NTFS.SYS 5031之类的驱动程序的数据部分。O/S下层安全代理5020可以捕获对驱动程序的数据部分的任何已尝试操纵以便防止对驱动程序间通信的恶意攻击。例如，O/S下层安全代理5020可以捕获对快速I/O路由指针5038的已尝试写入，并评估该尝试是否起因于NTFS.SYS 5031自身或操作系统5012。否则，O/S下层安全代理5020可以拒绝被判断为起因于诸如驱动程序2之类的另一驱动程序的这样的已捕获尝试。类似地，如果任何这样的关键数据由内核操作系统5012持有，那么，O/S下层安全代理5020可以被配置为捕获对包含这样的数据的存储器的已尝试访问。

[0652] 在又一进一步的示例中，驱动程序2 5028可以尝试通过对导入地址表5036的尝试

读取从驱动程序信息获得关于其他第三方的信息。O/S下层安全代理5020可以捕获对导入地址表5036的尝试读取，并拒绝不起源于诸如NTFS.SYS之类的驱动程序自身、从中导入该地址表的第三方或操作系统5012的任何尝试。

[0653] 在再一个进一步的示例中，可以钩取用于访问驱动程序的部分的函数调用，这允许恶意软件获得对电子设备5001的各种部分的访问权。O/S下层安全代理5020可以通过保护这样的函数调用驻留在其中的存储器、捕获对把恶意挂钩添加到系统函数的尝试写入来防备这样的攻击。类似地，O/S下层安全代理5020可以保护函数的代码部分免遭可以直接地访问代码部分以便注入恶意代码的恶意软件。例如，O/S下层安全代理5020可以捕获对代码部分2中所容纳的函数的代码的尝试写入，以便防止添加所注入的代码。

[0654] 因为捕获与驱动程序间调用相关联的各种资源可能是昂贵的，O/S下层安全代理5020可以根据要求允许或禁用对这样的资源的捕获。对于每一已捕获尝试，O/S下层安全代理5020可以标识动作的驱动程序或模块、标识目标驱动程序、并标识访问类型。这样的类型可以包括读、写或执行类型。O/S下层安全代理5020可以考虑这些元素，以及用于评估对访问电子设备5001的资源的尝试是否恶意的任何其他合适的准则。

[0655] 图53是用于电子设备中的驱动程序间通信的操作系统下层捕获方法5300的示例实施例。在步骤5305，可以访问安全规则以判断与要保护的驱动程序间通信相关联的资源。这样的安全规则可以标识资源，以及按照其来捕获和评估对资源的已尝试访问的准则。

[0656] 在步骤5310，可以在控制结构中在低于电子设备内的操作系统的级别设置标志。可以例如为用于捕获驱动程序间通信函数的已尝试执行、加载驱动程序间通信子函数的执行、读取或写入存储器中加载的驱动程序的数据或代码部分和/或跳转、分支或用于驱动程序间通信的驱动程序的代码部分的其他直接执行设置标志。可以通过存储器页面为虚拟存储器访问和/或通过与以上所描述的尝试相对应的存储器地址通过物理存储器访问来设置标志。

[0657] 在步骤5315，可以监视电子设备以便发现对访问与驱动程序间通信相关联的资源的已捕获尝试。在步骤5320，如果没有捕获尝试，那么，进程5300可以进行到步骤5315以继续监视以便发现已捕获尝试。如果已经捕获了尝试，那么，可以在步骤5325开始应对该尝试。可以在低于电子设备的操作系统的级别实施这样的应对。在步骤5325，可以收集可用于分析该尝试是否恶意的信息。例如，可以确定做出尝试的进程、应用、驱动程序或例程。可以从O/S内部安全代理获得来自电子设备的操作系统中的上下文信息。

[0658] 在步骤5330，可以判断对与驱动程序间通信相关的驱动程序的数据部分的已尝试访问是否未经授权。这样的数据部分内容可以包括EAT、IAT或任何其他合适的信息。如果是，那么，在步骤5360可以判断该尝试是否可疑和/或恶意，且可以拒绝该尝试。

[0659] 如果不是，在步骤5335，可以判断是否直接地访问用于驱动程序间通信的函数的内容无需使用被授权函数。在一个实施例中，可以判断调用进程或例程是否未经授权访问驱动程序的这样的部分。如果是，那么，在步骤5360可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。可以采取任何合适的矫正动作。如果不是，那么，在步骤5345可以判断直接地执行驱动程序间通信子函数而不使用被指派为用于这样的访问的函数。在一个实施例中，可以判断调用进程或例程是否未经授权做出这样的尝试。如果是，那么，在步骤5360可以判断改尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果不是，那么，在步骤5350可

以判断是否由经授权实体调用驱动程序间通信函数，或者是否由经授权实体调用所尝试的分支、跳转或其他直接执行。如果不是，那么，在步骤5360可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果是，在步骤5355可以允许该尝试。

[0660] 如果允许该尝试，且该尝试用于寄存回调函数，那么，在步骤5365可以标记为新近添加的回调函数分配的存储器以用于附加捕获。如果尝试寄存回调函数的实体是未知的，或者如果寄存回调的实体的恶意软件状态没有最后确定，则尤其可以采用这样的步骤。因而，可以捕获和评估回调函数的代码的后续的读、写或执行。否则，在步骤5385可以允许驱动程序执行。方法5300可以可选地返回到步骤5315，以继续监视以便发现对用于驱动程序间通信的电子设备的资源的已尝试访问。

[0661] 图54是用于保护电子设备5401上的驱动程序过滤器的附接和分开的系统5400的示例实施例。系统5400可以包括O/S下层安全代理5420，O/S下层安全代理5420被配置为在电子设备5401上操作以便检测对在诸如操作系统5412之类的电子设备5401的操作系统中附接或分开驱动程序过滤器的恶意尝试。此外，O/S下层安全代理5420可以被配置为使用一个或多个安全规则5408来判断什么驱动程序过滤器的已尝试附接或分开可以对应于已捕获操作，并基于该尝试和实施该尝试的实体判断该尝试是否得到授权。O/S下层安全代理5420可以被配置为允许已捕获事件、拒绝已捕获事件或对已捕获事件采取其他矫正动作。

[0662] 电子设备5401可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备1201可以包括被耦合到诸如物理存储器1203之类的存储器的一个或多个处理器1202。处理器5402可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器1202、图12的处理器1202或其任何组合实现，或者被配置为实现它们的功能性。存储器5403可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的物理存储器1203或虚拟存储器1204和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备5401可以包括操作系统5412，操作系统5412可以包括被耦合到一个或多个安全规则5421的O/S内部安全代理5419。操作系统5412可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理5419可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418、和/或图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0663] O/S下层安全代理5420可以由图1的O/S下层捕获代理104或已触发事件应对程序108、SVMM 216或图2的SVMM安全代理217、图4的固件安全代理440、442、O/S下层代理450、或PC固件安全5代理444、图5的固件安全代理516、或图7的微代码安全代理708或O/S下层代理712、图9的O/S下层捕获代理920或已触发事件应对程序922、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。

[0664] 安全规则5408可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则5421可以由图2的

安全规则220、图4的安全规则420、图7的安全规则721、图9的安全规则921、图12的安全规则1221和/或其任何组合实现，或者被配置为实现它们的功能性。

[0665] 电子设备5401可以包括一个或多个应用、驱动程序或可以尝试使用驱动程序访问电子设备5401的资源的其他实体——例如，“应用1”。在一个实施例中，这样的资源可以是I/O设备5430。I/O设备5430可以包括，例如，存储设备、显示设备、外围设备、键盘或任何其他设备或供电子设备5401用于输入和输出的组件。在一个实施例中，I/O设备5430可以是虚拟设备。操作系统5412可以应对对资源的请求。在一个实施例中，操作系统5412可以包括这样的请求的应对程序，例如I/O管理器5422。I/O管理器5422可以被配置为解析和应对对资源的请求并把该请求引导到适当的驱动程序以供进一步应对该请求。例如，I/O管理器5422可以把I/O请求从应用1发送到I/O驱动程序5428。操作系统5412可以包括任何合适的数量和种类的驱动程序以便管理和转换资源的请求，这些资源包括但不限于显示器、键盘、盘存储、串行、通用串行总线(USB)、火线、IEEE-488、插件板、打印机、计算机总线或联网。诸如I/O驱动程序5428之类的驱动程序可以被配置为执行到例如I/O设备的直接定址。在一个实施例中，操作系统5412可以包括可以仿真硬件设备的虚拟设备驱动程序。

[0666] 通过使用诸如设备栈区5424之类的结构，I/O驱动程序5428是可访问的。设备栈区5424可以是包括驱动程序和任何附加的驱动程序过滤器的结构。例如，设备栈区5424可以包括驻留在I/O驱动程序5428的顶部上的一个或多个I/O过滤器。诸如I/O请求之类的请求可以通过驱动程序栈区5424被发送给驱动程序5428，但由I/O过滤器5426截取。操作系统5412可以包括任何合适的数量和种类的驱动程序过滤器，以执行用于驱动程序或资源的专用操作。例如，诸如I/O驱动程序过滤器5426之类的驱动程序过滤器可以条件化或格式化请求、提供优化、缓存结果或执行任何其他合适的函数。诸如I/O驱动程序过滤器5426之类的驱动程序过滤器的具体实现可以取决于驱动程序自身的本质和/或身份。例如，一些驱动程序过滤器可以适用于特定种类的所有驱动程序，例如显示器、键盘或文件存储，而一些驱动程序过滤器可以适用于特定的品牌或型号的特定驱动程序。在接收诸如I/O请求之类的请求之后，诸如I/O过滤器5426之类的过滤器可以对该请求实施操作或代表该请求实施操作，且然后把经过滤的请求传送给诸如I/O驱动程序5428之类的驱动程序。I/O驱动程序5428可以与诸如I/O设备5430之类的设备通信并接收返回中的原始结果。通过诸如I/O过滤器5426之类的相同的过滤器把原始结果发送回去。I/O过滤器5426可以出于格式化、内容、呈现或任何其他合适的目的过滤结果。然后，可以把经过滤的结果传送回去给诸如I/O管理器5422之类的调用设备栈区的实体，或最终传送回去给应用1。

[0667] 图55是示例设备栈区5500的操作的更详尽的阐释。出于说明性目的，设备栈区5500可以被配置成用于使得应用与存储盘上的文件交互的文件I/O驱动程序栈区。设备栈区5500可以包括被配置为过滤对文件I/O驱动程序5506的请求和来自文件I/O驱动程序5506的请求的“过滤器1”5502和反恶意软件文件I/O过滤器5504。设备栈区5500的基础可以是文件I/O驱动程序5506，且请求可以进入设备栈区5500，被向下传送给驱动程序，且向上返回结果，并设备栈区5500的顶部访问该结果。例如，请求可以被过滤器1 5502接收到，被传送到反恶意软件文件I/O过滤器5510，然后，被传送到文件I/O驱动程序5506。每一过滤器可以被配置为在栈区中向上或向下传送请求之前执行其个体过滤操作。文件I/O驱动程序5506可以被配置为实施经过滤的I/O请求并把结果(如果有的话)返回给反恶意软件文件I/

0过滤器5504，反恶意软件文件I/O过滤器5504可以被配置为把其自己的经过滤结果返回给过滤器1 5502。过滤器1和反恶意软件文件I/O过滤器5504可以均被配置为对结果执行过滤操作。设备栈区5500的过滤器可以被配置为过滤任何合适的请求，例如读、写或执行。

[0668] 设备栈区5500可以包括用于组织栈区和促进在设备栈区5500内的过滤器和驱动程序之间的通信的任何合适的机制。例如，设备栈区5500可以包括用于标识设备栈区5500的基础以及标识过滤器的次序的数据结构。用于排序设备栈区5500的示例数据结构可以包括指针5508、5510、5512或5514。每一指针可以包括在栈区中被定位在上面或下面的下一实体的地址。例如，反恶意软件文件I/O过滤器5510可以包括指向在栈区下的下一实体的指针5512(该实体可以是文件I/O驱动程序5514)，以及指向栈区上的下一实体的指针5510(该实体可以是过滤器1 5502)。这样的示例数据结构可以由设备对象实现。

[0669] 过滤器1 5502可以被配置为接收文件I/O请求，例如“把Malware.DLL写到Kernel.DLL”，这是指示文件“Kernel.DLL”的内容要被另一文件“Malware.DLL”盖写的命令。过滤器1 5502可以接收该请求、对请求执行其操作并把该请求传送给栈区下的下一实体，该实体可以是反恶意软件文件I/O过滤器5504。反恶意软件文件I/O过滤器5504可以被配置为保护系统的核心文件免遭未经授权的程序的篡改。例如，恶意软件可以尝试改变或删除某些系统文件，例如内核操作系统内容、主引导记录或反恶意软件软件文件。在图55的示例中，该请求可以是尝试用未知的、潜在恶意的文件“Malware.DLL”写诸如“Kernel.DLL”之类的这样的受保护文件，和/或请求可以起源于不同于操作系统的内核进程的进程。因而，例如，反恶意软件文件I/O过滤器5504可以被配置为过滤这样的请求并阻止请求使其不能到达文件I/O驱动程序5506。反恶意软件文件I/O过滤器5504可以被配置为把受阻止的请求发送给在系统上运行的反恶意软件代理5516以供进一步分析。在一个实施例中，反恶意软件代理可以完全地或部分地由图54的O/S内部安全代理5419实现，或者由任何其他合适的反恶意软件模块、软件、系统或进程实现。反恶意软件文件I/O过滤器5504可以被配置为把已欺骗的应答返回到栈区上，以使得尝试该请求的实体可以如同写入成功了的那样进行。否则，如果该写尝试不被视为可疑的或恶意的，那么，反恶意软件文件I/O过滤器5504可以被配置为把该请求传送给文件I/O驱动程序5506，并将结果返回到过滤器1 5502。

[0670] 图56是已经受到作用为附接或分开驱动程序过滤器的恶意软件危害的设备栈区的示例阐释。文件I/O设备栈区5602是如图54的设备栈区5424或图55的设备栈区5500所描述的设备栈区的实现。文件I/O设备栈区5602可以被配置为提供对文件I/O驱动程序5610的访问，并包括诸如“过滤器1”5606之类的过滤器和反恶意软件文件I/O过滤器5608。反恶意软件文件I/O过滤器5608是图55的反恶意软件文件I/O过滤器5504的实现。已经从驱动程序栈区5602分开反恶意软件文件I/O过滤器5608。这样的分开已经由用于分开或移除驱动程序过滤器的系统函数调用完成，或者通过直接操纵文件I/O设备栈区5602的数据结构完成。过滤器1 5606不再通过反恶意软件文件I/O过滤器5608传送请求，且可以改为回避它并把请求直接地发送给诸如文件I/O驱动程序5610之类的下一实体。指针5612已经被修改为改为越过反恶意软件文件I/O过滤器5608指向下一实体。文件I/O驱动程序5610不再接收由反恶意软件文件I/O过滤器5608过滤的请求。响应于所接收的请求，文件I/O驱动程序5610可以把应答发送回去给过滤器1，这是由于文件I/O驱动程序5610的指针5614会被修改，以使得反恶意软件文件I/O过滤器5608不再处于过滤器的基础。因而，可以有效地从文件I/O设

备栈区5602移除反恶意软件文件I/O过滤器5608。

[0671] 反恶意软件文件I/O过滤器5608的移除可能已经得到授权，或者可能是恶意软件攻击的结果。在与反恶意软件文件I/O过滤器5608相同的执行优先级(例如0环)操作的恶意软件可以成功地分开过滤器而无需检测。尽管作为示例示出反恶意软件文件I/O过滤器5608，但其他驱动程序过滤器类似地受到攻击。

[0672] 键盘I/O设备栈区5604可以是如图54的设备栈区5424或图55的设备栈区5500所描述的设备栈区的完全或部分的实现。键盘I/O设备栈区5604可以被配置为提供对系统的键盘设备的访问。在一个实施例中，键盘I/O设备栈区5604最初包括在键盘驱动程序5620的顶部上的诸如过滤器1 5616之类的过滤器。然而，驱动程序过滤器附接操作已经把恶意键盘记录器过滤器5618附加到键盘I/O设备栈区5604中。诸如过滤器1 5616的指针5622或键盘驱动程序5620的指针5628之类的栈区的数据结构已经被修改为允许在过滤器1 5616和键盘驱动程序5620之间插入恶意键盘记录器过滤器5618。已经通过附接驱动程序过滤器的系统函数调用或者通过直接操纵键盘I/O设备栈区5604的数据结构完成这样的操作。恶意键盘记录器过滤器5618可以被配置为捕捉用户击键并把它们保存到文件或远程服务器。

[0673] 恶意键盘记录器过滤器5618或其他潜在恶意的过滤器可以被安装在栈区中免遭反恶意软件软件检测的位置。例如，潜在恶意的过滤器可以被安装在栈区中比反恶意软件过滤器低的位置，以使得反恶意软件过滤器完成的无论什么矫正动作都可以由恶意过滤器撤销。此外，恶意过滤器可以代替可信过滤器被插入到栈区中，以便掩饰恶意过滤器的操作。

[0674] 返回到图54，在操作系统5412的级别运行的反恶意软件软件可能不能完全解决驱动程序过滤器的恶意附接和分开，这是因为实施这样的活动的恶意软件也可以在相同的优先级水平运行。

[0675] O/S下层安全代理5420可以被配置为截取对低于操作系统5412的级别附接或分开驱动程序过滤器的尝试。O/S下层安全代理5420可以被配置为查阅安全规则5408以确定与驱动程序过滤器的附接和分开相关联的资源并捕获对这样的资源的已尝试访问。这样的资源可以包括，例如，存储器5403的各部分。在捕获这样的尝试访问之后，O/S下层安全代理5420可以被配置为基于安全规则5408判断尝试访问的实体是否得到授权采取动作。O/S下层安全代理5420可以被配置为允许或拒绝请求或采取另一适当的动作。

[0676] 在一个实施例中，存储器5403可以包括虚拟存储器。在这样的实施例中，存储器5403可以包含存储器页面，存储器页面包括：用于附接函数5436、附接子函数5438、分开函数5440和/或分开子函数5442的代码；驱动程序数据结构的数据表权限5444；和/或驱动程序数据结构5446自身。附接函数5436和分开函数5440可以由操作系统5412提供，供在电子设备5401内的实体附接或分开驱动程序。这样的实体通常可以调用附接函数5436或分开函数5440。通过提供附接函数5436和分开函数5440，操作系统5412可以提供受控、受保护和有效的机制，供电子设备5401的实体添加或移除驱动程序过滤器。然而，附接子函数5438和分开子函数5442未经建档，或以另外方式阻止操作系统5412使用它们。附接子函数5438和分开子函数5442预期仅由关联的附接函数5436和分开函数5440使用。恶意软件可以通过调用附接子函数5438和分开子函数5442的个体实例回避附接函数5436和分开函数5440的安全和控制机制。

[0677] 在另一实施例中，存储器5403可以包括物理存储器。在这样的实施例中，存储器5403可以包括存储器地址，存储器地址包括：用于系统的附接函数5436、附接子函数5438、分开函数5440、分开子函数5442的代码；关于驱动程序数据结构的权限5444的数据；和/或驱动程序数据结构5446自身。

[0678] 用于附接函数5436的代码可以包括用于由系统5400或操作系统5412指派的、供应用或驱动程序激活驱动程序过滤器的任何函数的任何代码。这样的函数可以包括用于把驱动程序过滤器附接到驱动程序栈区的函数，例如把的I/O过滤器5426中的一个附接到设备栈区5424。这些函数又可以调用子程序或其他函数来执行激活驱动程序过滤器时的特定任务。包含用于分开函数5440的代码的存储器可以包括用于由系统5400或操作系统5412指派的、供应用或驱动程序禁用驱动程序过滤器任何函数的任何代码。这样的函数可以包括用于从驱动程序栈区分开驱动程序过滤器的函数，例如从设备栈区5424分开I/O过滤器5426中的一个。这些函数又可以调用子程序或其他函数来执行在禁用驱动程序过滤器时的特定任务。在一个示例中，Windows™附接函数5436可以包括但不限于：IoAttachDevice()、IoAttachDeviceByPointer()、IoAttachDeviceToDeviceStack()和IoAttachDeviceToDeviceStackSafe()。在另一示例中，Windows™分开函数5440可以包括但不限于：IoDeleteDevice()和IoDetachDevice()。

[0679] 用于这样的子程序或其他函数的代码可以被包括在包含用于附接子函数5438或分开子函数5442的代码的存储器中。恶意软件可以直接地调用子函数以便避免被操作系统5412检测。因而，如果系统5400的实体已经直接地调用这样的子函数而不使用存储器中用于诸如附接函数5436或分开函数5440之类的标准函数的代码，那么，可以判断已尝试访问是可疑的。在一个示例中，Windows™附接子函数可以包括但不限于IopAttachDevicetoDeviceStackSafe()。这样的子函数可以由Windows™附接函数5436的实例中的每一个调用。

[0680] 驱动程序数据结构的权限5444可以包括用于设置读、写或执行与诸如设备栈区5424之类的设备栈区相关联的数据结构的能力的表、标志或任何其他合适的数据结构或指示。例如，这样的权限5444可以控制写或读设备栈区中的、诸如图55的指针5508、5510、5512或5514或图56的指针5612、5614、5622、5624、5626或5628之类的指针的能力。改变存储器5403中的权限5444的未经授权的尝试可以指示恶意软件尝试恶意地附接或分开诸如I/O过滤器5426中的一个之类的驱动程序过滤器的第一步骤。

[0681] 驱动程序数据结构5446可以包括用于组织诸如I/O驱动程序5428或设备栈区5424之类的驱动程序或设备栈区的任何合适的数据结构。例如，驱动程序数据结构5446可以包括图55的指针5508、5510、5512或5514或图56的指针5612、5614、5622、5624、5626或5628。即使恶意软件已经彻底地掩饰其调用例程，但附接或分开驱动程序过滤器要求改变驱动程序数据结构5446中的值。因而，对改变驱动程序数据结构5446的值的未经授权的尝试可以指示恶意软件。

[0682] O/S下层安全代理5420可以被配置为基于物理存储器和/或虚拟存储器保护存储器5403中与驱动程序过滤器相关的内容。例如，O/S下层安全代理5420可以被配置为截取尝试读、写或执行包含以下的存储器页面的请求：用于附接函数5436、附接子函数5438、分开函数5440、分开子函数5442的代码；驱动程序数据结构的数据表权限5444；和/或驱动程序

数据结构5446自身。在这样的示例中，可以完全地或部分地在虚拟机监视器中实现O/S下层安全代理5420。在另一示例中，O/S下层安全代理5420可以被配置为截取尝试读、写或执行包含以下的存储器地址的请求：用于附接函数5436、附接子函数5438、分开函数5440、和/或分开子函数5442的代码；驱动程序数据结构的数据表权限5444；和/或驱动程序数据结构5446自身。

[0683] O/S下层安全代理5420可以被配置为截取对存储器5403中与驱动程序过滤器相关的内容的这样的请求并根据上下文信息评估该请求。这样的上下文信息可以包括做出请求的实体、请求的本质(例如读、写或执行)、尝试写的值、实体做出请求的方式、先前对请求存储器5403中与驱动程序过滤器相关的内容的尝试，和/或来自O/S内部安全代理5419的、关于已经尝试访问存储器5403的、在操作系统5412的级别的实体的操作的信息。

[0684] 基于该请求的评估，O/S下层安全代理5420可以被配置为允许请求、拒绝请求、把经欺骗的响应发送回去给调用实体，或采取任何其他合适的矫正动作。

[0685] 在操作中，O/S下层安全代理5420可以在电子设备5401上操作以便安全地附接和/或分开驱动程序过滤器。应用、驱动程序或诸如“应用2”之类的其他实体可以发起过滤器附接或分开尝试。例如，应用2可以以用户模式、以内核模式、在与操作系统5412相同的级别或者在比操作系统5412高的级别操作。O/S下层安全代理5420可以访问安全规则5408以判断如何保护电子设备5401上的驱动程序过滤器的附接和分开。O/S下层安全代理5420可以设置控制结构标志以便捕捉，例如：对用于附接函数5436、附接子函数5438、分开函数5440和/或分开子函数5442的代码的存储器页面或地址的已尝试执行；对用于驱动程序数据结构的权限5444的存储器页面或地址的尝试写入；和/或对驱动程序数据结构5446的尝试读取或写入。应用2可以尝试通过各种机制访问诸如I/O过滤器5426之类的驱动程序过滤器，包括通过访问与这样的驱动程序过滤器相关联的存储器5403的内容。

[0686] 在一个实施例中，例如，应用2可以通过直接操纵存储器5403中的值，例如尝试写入驱动程序数据结构的权限5444，来尝试过滤器附接或分开。这样的已尝试写入可以是对把驱动程序数据结构上的只读权限改变成读/写的尝试，以使得以后可以重写数据结构的值，以便附接或分开驱动程序过滤器。这样的尝试可以回避用于访问诸如I/O过滤器5426之类的驱动程序过滤器的标准化的和受保护的机制。通过回避这样的机制，该尝试可以对操作系统5412的安全措施掩饰、对其隐藏或以另外方式挫败操作系统5412的安全措施。

[0687] 在另一实施例中，应用2可以通过调用和运行由操作系统5412提供的用于过滤器附接或分开的操作的附接或分开函数5432来尝试过滤器附接或分开。这样的附接或分开函数5432又可以调用和运行附接或分开子函数5434a的实例。附接或分开子函数5434a可以执行引起对驱动程序数据结构5446的已尝试访问或对这样的驱动程序数据结构的权限5444的访问的特定调用。附接或分开函数5432可以由操作系统5412提供为访问诸如I/O过滤器5426之类的驱动程序过滤器的标准化的、受保护的机制。附接或分开函数5432可以受到保护，以使得仅操作系统5412的某些进程可以使用函数来访问驱动程序过滤器。

[0688] 在又一实施例中，应用2可以通过直接地调用和运行附接或分开子函数5434b的实例来尝试过滤器附接，而不使用诸如由操作系统5412提供的用于对驱动程序数据结构5446的标准化的、受保护的访问的附接或分开函数5432之类的函数。如果操作系统5412不包括用于保护和批准对附接或分开子函数5434b的使用的机制(对附接或分开函数5432来说包

括这样的机制)，那么，子函数5434b的直接使用可以被恶意软件用来隐藏或掩饰自身或以另外方式挫败操作系统5412的安全措施。

[0689] O/S下层安全代理5420可以捕获对存储器5403中与驱动程序过滤器相关的内容的已尝试访问。O/S下层安全代理5420可以包括控制结构，以判断如何处理已截取访问。O/S下层安全代理5420可以访问安全规则5408或保护服务器以判断如何应对这样的尝试访问。

[0690] 例如，可以捕获应用2对执行附接函数5436或分开函数5440的尝试。对可以由操作系统提供为用于访问I/O过滤器5426的标准化的或受保护的方法的这样的函数的使用可以限于，例如，经数字签署的驱动程序。因而，在一个实施例中，O/S下层安全代理5420可以访问安全规则5408以便判断要求驱动程序经过数字签署的规则、判断调用应用或驱动程序以及判断驱动程序是否经过签署。O/S下层安全代理5420可以访问O/S内部安全代理5419，O/S内部安全代理5419可以访问操作系统5412以判断应用2的签署状态。可以通过检查操作系统5412的调用栈区来做出这样的访问。在另一实施例中，O/S下层安全代理5420可以例如基于应用2的散列判断应用2是否在黑名单、白名单上或者未知其恶意状态。O/S下层安全代理5420可以判断，如果应用2是未知的，则作为预防可以阻止应用2，或者也许可以把关于应用2的信息报告给保护服务器。此外，如果应用是已知的，则O/S下层安全代理5420可以通过捕获更多的应用2的操作以更严格的审查监视应用2的操作。O/S下层安全代理5420可以判断应用2包括恶意软件、阻止应用2、清洁应用2的电子设备5401或采取其他矫正动作。

[0691] 在另一示例中，可以捕获应用2对执行附接子函数5438或分开子函数5442的尝试。通常仅通过使用诸如附接函数5436或分开函数5438之类的标准化的或受保护的机制来实施对这样的函数的使用。因而，在一个实施例中，如果调用附接子函数5438或分开子函数5442的例程不是已知的或可列举的标准化的或受保护的机制中的一种，O/S下层安全代理5420可以基于该例程捕获。在另一实施例中，O/S下层安全代理5420可以捕获对附接子函数5438或分开子函数5442的所有已尝试访问，然后确定调用例程，并且如果调用例程标准化的或受保护的机制中的一种则拒绝该请求。例如，通过设备栈区5424中的信息、驱动程序数据结构5446中的信息或通过判断从存储器5403中的哪一存储器页面或存储器地址做出执行子函数的命令并使得页面或地址与存储器映射相关，可以确定调用例程。如果判断该调用例程是附接函数5436或分开函数5442的实例，那么，可以如先前所描述的验证调用这些函数的例程。O/S下层安全代理5420可以拒绝不起源于经授权的附接或分开函数的调用的对附接子函数5438或分开子函数5442的任何调用。

[0692] 在又一示例中，O/S下层安全代理5420可以捕获应用2对写驱动程序数据结构的权限5444或者读或写驱动程序数据结构5446的尝试。对所有这样的尝试的捕获可以包括捕获起源于分开或附接函数的执行的这样的尝试。因而，在捕获这样的尝试时，O/S下层安全代理5420可以判断从存储器的什么部分或从什么实体做出这样的尝试。如果这样的尝试是从经授权函数做出的，那么，可以允许该尝试。可以如先前所描述的验证经授权函数自身的调用者。如果这样的尝试不是从经授权函数做出的，那么，该尝试可以指示应用2对直接地操纵设备栈区5424的恶意尝试且可以阻止该尝试。

[0693] O/S下层安全代理5420可以在判断尝试是否恶意时考虑设备的类型。例如，虚拟盘卷可以特别倾向于被过滤器利用。因而，O/S下层安全代理5420可以访问诸如对象之类的驱动程序数据结构5446以便确定设备的类型，并且，如果类型是“FILE_VIRTUAL_VOLUME”则要

求请求者经过数字签署。这样的要求可以独立于操作系统5412的要求。在判断是否捕获该尝试或判断该尝试是否恶意时可以考虑的其他类型的设备对象可以包括但不限于：写一次介质、虚拟卷、可移动介质、远程设备、软盘、只读设备、已挂载设备、即插即用设备或具有自动产生的名称的设备。安全规则5408可以包括对这样的类型的设备对象的考虑。例如，如果调用驱动程序是未知的，则可以保护调制解调器类型的设备免遭所有附接。这可以允许O/S下层安全代理5420防止恶意的驱动程序嗅探传真和调制解调器操作。在另一示例中，如果对于扫描仪设备的驱动程序不存在已知的合法过滤器使用，则可以保护设备扫描仪的驱动程序免遭所有附接。

[0694] 如上所述，O/S下层安全代理5420可以基于尝试访问I/O过滤器5426的实体触发或应对该尝试。此外，O/S下层安全代理5420可以确定要访问的设备的自有驱动程序，并在在判断尝试是否恶意时考虑这样的信息。驱动程序数据结构5446或设备数据结构均可以包含链接驱动程序和设备的信息。可以访问这样的数据结构，以判断在驱动程序和设备之间的关系。例如，如果由应用2做出对访问I/O过滤器5426的尝试但判断应用1占用I/O设备5430，那么可以拒绝该请求。

[0695] 图57是用于电子设备中的驱动程序过滤器附接的操作系统下层捕获的方法5700的示例实施例。在步骤5705，可以访问安全规则以判断要保护的与驱动程序过滤器附接相关联的资源。这样的安全规则可以标识资源以及捕获和评估对资源的已尝试访问的准则。

[0696] 在步骤5710，可以在控制结构中在低于电子设备内的操作系统的级别设置标志。例如，可以为捕获对附接函数或分开函数的已尝试执行、附接子函数或分开子函数的执行、写到驱动程序和设备数据结构的读/写/执行权限和/或读取或写入数据结构自身设置标志。可以为通过对应于以上所描述的尝试的存储器地址、通过存储器页面和/或通过物理存储器访问的虚拟存储器访问设置标志。

[0697] 在步骤5715，可以监视电子设备以便发现对访问与驱动程序过滤器的附接相关联的资源的已捕获尝试。在步骤5720，如果没有捕获尝试，那么，进程5700可以进行到步骤5715以继续监视以便发现已捕获尝试。如果已经捕获了尝试，那么，可以在在步骤5725开始应对该尝试。可以在低于电子设备的操作系统的级别实施这样的应对。在步骤5725，可以收集可用于分析该尝试是否恶意的信息。例如，可以确定做出该尝试的进程、应用、驱动程序或例程。可以从O/S内部安全代理获得来自电子设备的操作系统中的上下文信息。可以确定与该尝试相关联的设备的设备类型，也可以确定设备的自有驱动程序。

[0698] 在步骤5735，可以判断是否直接访问设备对象或驱动程序栈区的数据结构而无需使用经授权函数。在一个实施例中，可以判断调用进程或例程是否未经授权访问这样的数据结构。如果已经直接地访问各数据结构，那么，在步骤5760可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。可以采取任何合适的矫正动作。如果没有直接地访问各数据结构，那么，在步骤5740可以判断是否尝试直接写入设备对象或设备栈区的数据结构的权限。在一个实施例中，可以判断调用进程或例程是否未经授权做出这样的写入尝试。如果已经直接写入这样的数据结构，那么，在步骤5760可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果没有直接写入这样的数据结构，那么，在步骤5745可以判断是否直接地执行附接或分开子函数而无需使用指派为用于这样的访问的函数。在一个实施例中，可以判断调用进程或例程是否未经授权做出这样的尝试。如果已经直接地执行这样的子函

数，那么，在步骤5760可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果没有直接地执行这样的子函数，那么，在步骤5750可以判断附接函数或分开函数是否由经授权实体调用。如果没有由经授权条目调用该函数，那么，在步骤5760可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果已经由经授权条目调用该函数，则在步骤5755可以允许该尝试。

[0699] 在执行步骤5755或5760之后，方法5700可以可选地返回到步骤5715以继续监视以便发现对用于驱动程序过滤器附接的电子设备的资源的已尝试访问。

[0700] 图58是用于保护电子设备5801上的驱动程序的加载或卸载的系统5800的示例实施例5800。系统5800可以包括被配置为在电子设备5801上操作以便检测对加载或卸载诸如操作系统5812之类的电子设备5801的操作系统中的驱动程序的恶意尝试的O/S下层安全代理582。此外，O/S下层安全代理5820可以被配置为使用一个或多个安全规则5808来判断驱动程序的什么已尝试加载或卸载可以对应于已捕获操作以及基于该尝试和实施该尝试实体判断是否授权该尝试。O/S下层安全代理5820可以被配置为允许已捕获事件、拒绝已捕获事件或对已捕获事件采取其他矫正动作。

[0701] 电子设备5801可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备5801可以包括被耦合到诸如存储器5803之类的存储器的一个或多个处理器5802。处理器5802可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器1202、图12的处理器1202或其任何组合实现，或者被配置为实现它们的功能性。存储器5803可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的物理存储器1203或虚拟存储器1204和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备5801可以包括操作系统5812，操作系统5812可以包括被耦合到一个或多个安全规则5821的O/S内部安全代理5819。操作系统5812可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理5819可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418和/或图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0702] O/S下层安全代理5820可以由图1的O/S下层捕获代理104或已触发事件应对程序108、SVMM 216或图2的SVMM安全代理217、图4的固件安全代理440、442、O/S下层代理450或PC固件安全代理444、图5的固件安全代理516或图7的微代码安全代理708或O/S下层代理712、图9的O/S下层捕获代理920或已触发事件应对程序922、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。

[0703] 安全规则5808可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则5821可以由图2的安全规则220、图4的安全规则420、图7的安全规则721、图9的安全规则921、图12的安全规则1221和/或其任何组合实现，或者被配置为实现它们的功能性。

[0704] 电子设备5801可以包括可以尝试访问电子设备5801的资源以便加载或卸载驱动程序的一个或多个应用、驱动程序或其他实体——例如，应用5826。应用5826可以包括任何进程、应用、程序或驱动程序。应用5826可以尝试访问诸如存储器5803之类的资源。在一个实施例中，应用5826可以尝试通过存储器页面访问存储器5803，其中，存储器5803已经由操作系统5812虚拟化。在另一实施例中，应用5826可以通过访问物理存储器的地址尝试访问存储器5803。应用5826可以尝试使用处理器5802来执行存储器5803中的指令。

[0705] 操作系统5812可以提供供在电子设备5801内的诸如应用5826之类的实体加载和卸载驱动程序的函数。这样的实体通常可以调用加载和卸载函数。通过提供这样的函数，操作系统5812可以提供供电子设备5801的实体加载或卸载驱动程序的受控制的、受保护的和有效的机制。这样的函数又可以依赖于加载子函数和卸载子函数的组合。这样的加载和卸载子函数可以未经建档，或以另外方式阻止操作系统5812使用它们。加载和卸载子函数预期仅由关联的加载和卸载函数使用。恶意软件可以通过调用加载子函数和卸载子函数的个体实例回避加载和卸载函数的安全和控制机制。此外，恶意软件可以通过直接地跳转到这样的函数或子函数的代码部分回避加载和卸载函数的安全和控制机制。另外，当把驱动程序加载到存储器中以供执行时，恶意软件可以尝试变更驱动程序，其中，盘上的驱动程序上的映像没有恶意软件，但在把代码加载到驱动程序时被恶意软件注入，得到损坏的驱动程序。

[0706] 操作系统5812所提供的任何合适的加载或卸载函数可以被用来加载或卸载驱动程序。例如，由Windows™操作系统实现的操作系统5812可以利用函数ZwLoadDriver()来加载驱动程序或使用函数ZwUnloadDriver()来卸载驱动程序。这样的加载或卸载函数可以调用任何合适的数量或种类的子函数。例如，ZwLoadDriver()可以调用NtLoadDriver()，NtLoadDriver()又可以调用IopLoadUnloadDriver()。IopLoadUnloadDriver()又可以调用IopUnloadDriver()以便加载驱动程序或调用IopUnloadDriver以便卸载驱动程序。因而，可以使用函数调用的层次结构来加载或卸载驱动程序。在一个实施例中，作为其他操作的副作用，可以加载驱动程序。例如，由Windows™操作系统实现的操作系统5812可以利用函数MmLoadSystemImage()来加载整个系统映像，因此在此期间可以加载驱动程序。由这一函数加载的映像可以包含驱动程序的相对地址，且必须基于再定位表来再定位这样的相对地址。为了完成这样的任务，对MmLoadSystemImage()的调用可以使用子函数LdrRelocateImage。在另一实施例中，某些辅助函数可以被用来加载驱动程序，例如MiMapViewOfDataSection()、MiMapViewOfImageSection()或MiMapViewOfPhysicalSection()。

[0707] 为了保护驱动程序的加载和卸载，O/S下层安全代理5820可以被配置为访问安全规则5808以判断应当捕获电子设备5801的资源的什么部分以及应当确定与尝试相关联的什么上下文信息。O/S下层安全代理5820可以被配置为捕获对任何合适的函数或子函数的已尝试执行。O/S下层安全代理5820可以被配置为以任何合适的方式捕获对这些函数或子函数的已尝试执行。

[0708] 在一个实施例中，O/S下层安全代理5820可以被配置为在控制结构中设置用于在存储器5803中执行这样的函数的标志。例如，O/S下层安全代理5820可以被配置为捕获对在存储器5803中的地址(D)处的ZwLoadDriver()的函数入口点的已尝试执行。在另一示例

中，O/S下层安全代理5820可以被配置为捕获对在存储器5803中的地址(E)处的NtLoadDriver()的函数入口点的已尝试执行。

[0709] 在另一实施例中，O/S下层安全代理5820可以被配置为捕获对这样的函数的代码部分的直接执行，这可以强烈指示恶意软件。例如，O/S下层安全代理5820可以被配置为捕获任何意图的“JMP”指令或导致对在存储器5803中的地址(A)处的的NtLoadDriver()的代码部分的直接访问的类似指令。

[0710] 在尝试加载驱动程序时，O/S下层安全代理5820可以被配置为在准许加载之前判断加载驱动程序的潜在影响。O/S下层安全代理5820可以被配置为检查存储中的驱动程序的映像，例如在盘5824上。例如，在捕获新驱动程序5830的尝试加载时，O/S下层安全代理5820可以被配置为扫描在盘5824上的新驱动程序5830的映像的内容。O/S下层安全代理5820可以被配置为确定盘5824上的新驱动程序5830的映像的内容的散列或这样的内容的部分(例如，代码部分)的数字签名和/或散列或签名。O/S下层安全代理5820可以被配置为根据白名单、黑名单或诸如安全规则5808之类的其他信息判断是否已知新驱动程序5830是安全的、恶意的还是未知的。O/S下层安全代理5820可以被配置为在新驱动程序5830驻留在盘5824上时评估新驱动程序5830的布局，以便确认新驱动程序5830的身份。O/S下层安全代理5820可以被配置为评估由新驱动程序5830的创建者提供的数字签名，例如签署者名称、签名信息或签名日期。O/S下层安全代理5820可以被配置为在映像新驱动程序5830的文件驻留在盘5824中时评估映像新驱动程序5830的文件的名称。

[0711] 在一个实施例中，O/S下层安全代理5820可以被配置为通过允许加载操作但实现附加行为的监视来有条件地允许尝试加载驱动程序。例如，在捕获应用5826对新驱动程序5830的尝试加载时，O/S下层安全代理820可以确定应用5826和盘5824上的新驱动程序5830的映像都不是已知的，但要加载的驱动程序最初没有显示与操作系统5812的关键部分交互的迹象。为了防止假阳性阻止访问，O/S下层安全代理5820可以被配置为允许加载新驱动程序5830。然而，由于还没有确定地判断驱动程序的真正本质，O/S下层安全代理5820可以被配置为在存储器5803中的地址(B)处在为新驱动程序5830分配的存储器空间中分配附加的标志或触发。因而，当执行新驱动程序5830的代码且新驱动程序尝试采取各种动作时，O/S下层安全代理5820可以被配置为监视新驱动程序5830的动作以便确保它在电子设备5801中不采取恶意动作。

[0712] 在另一实施例中，O/S下层安全代理5820可以被配置为通过当驱动程序驻留在存储器5803中时允许加载操作但然后实施驱动程序的附加安全验证来有条件地允许尝试加载驱动程序。O/S下层安全代理5820可以被配置为直到完成这样的检查之前停止驱动程序的执行。由O/S下层安全代理5820对存储器5803中的驱动程序的映像实施的安全验证可以类似于当驱动程序驻留在诸如盘5824之类的存储时对驱动程序的映像实施的那些。例如，O/S下层安全代理5820可以被配置为当新驱动程序5830的映像驻留在存储器5803中时扫描存储器中在地址范围(B)处的新驱动程序5830的映像的内容、确定映像的内容或映像的分部的散列或数字签名、把映像与白名单或黑名单进行比较以及和评估新驱动程序5830的映像的文件布局或文件名称。此外，O/S下层安全代理5820可以被配置为比较分析诸如盘5824之类的存储中的驱动程序的映像和存储器5803中的驱动程序的映像的结果。一些差异可以指示当加载驱动程序时该代码已经被注入到驱动程序中。

[0713] 在尝试卸载驱动程序时，O/S下层安全代理5820可以被配置为确定要卸载的驱动程序的身份，以便判断这样的驱动程序是否关键的。在一个实施例中，O/S下层安全代理5820可以被配置为查阅指示值(例如在存储器5803中的地址(C)和存储器范围(B)之间的那些值)与关联的驱动程序和实体的相关的存储器映射。在图58的示例中，这样的范围可以对应于由旧驱动程序5828使用的存储器空间。取决于要卸载的驱动程序的身份，O/S下层安全代理5820可以被配置为应用对要卸载的驱动程序的身份来说专用的安全规则5808。例如，为了最小化恶意卸载操作的假阳性标识，安全规则5808可以仅限定已经被标识为对电子设备5801的安全或操作来说是关键的、或者以另外方式倾向于是恶意软件的驱动程序的移除。

[0714] O/S下层安全代理5820可以被配置为捕获对驱动程序信息的尝试直接操纵。这样的直接操纵可以由恶意软件尝试，避免了使用系统函数来卸载驱动程序。在一个实施例中，O/S下层安全代理5820可以被配置为捕获对与在特定的地址范围内的加载或卸载相关的驱动程序的数据结构(例如用于存储器5803中在地址(C)和存储器范围(B)之间的旧驱动程序5828的那些)的已尝试访问。O/S下层安全代理5820可以被配置为判断是否通过经批准的函数调用来实施该尝试。如果不是，则O/S下层安全代理5820可以被配置为判断尝试是可疑的。

[0715] O/S下层安全代理5820可以被配置为确定任何已尝试加载或卸载驱动程序操作的源。O/S下层安全代理5820可以被配置为评估对用于加载或卸载的资源的已尝试访问，同时考虑该尝试源。例如，如果该尝试来自可信的源，其中做出操作的应用或驱动程序经过数字签署且其代码部分的散列是在白名单上，那么，O/S下层安全代理5820可以被配置为允许这样的源加载其恶意软件状态未知的驱动程序。在另一示例中，如果该尝试来自被判断为恶意的源，那么，O/S下层安全代理5820可以被配置为不允许它加载或卸载任何驱动程序。在又一示例中，如果已经访问用于加载或卸载的子函数，且调用进程、应用、驱动程序或函数不是被指派为调用子函数的系统提供的函数，那么，O/S下层安全代理5820可以被配置为不允许该访问。在再一个示例中，如果通过“JMP”或类似指令直接访问代码部分，则O/S下层安全代理5820可以被配置为确定从什么存储器位置做出该尝试指令，并且，如果该尝试不是来自用于加载或卸载驱动程序的经授权函数或子函数，则拒绝访问。在所有这样的示例中，O/S下层安全代理5820可以被配置为穿行调用栈区以确定引起对电子设备5801的资源的已尝试访问的调用链。在调用链的每一步骤，O/S下层安全代理5820可以被配置为评估做出调用的实体。

[0716] 此外，O/S下层安全代理5820可以被配置为基于管理员的设置判断已尝试加载或卸载操作是不是恶意的。例如，电子设备5801在其中操作的企业的管理员可以指定如果电子设备5801的用户没有登记为管理员则不可以加载或卸载驱动程序。

[0717] 另外，O/S下层安全代理5820可以被配置为基于先期对访问电子设备5801的资源的已捕获尝试判断已尝试加载或卸载操作是不是恶意的。可以使用确认对加载或卸载的尝试与电子设备5801上检测到的另一可疑动作相关的任何合适的准则。例如，如果应用5826先前在尝试卸载诸如旧驱动程序5828之类的关键驱动程序时被拒绝，则O/S下层安全代理5820可以被配置为拒绝应用5826的后续加载或卸载尝试，这是由于应用5826已经实施了可疑活动。

[0718] 在一个实施例中，O/S下层安全代理5820可以被配置为基于每一存储器页面捕获处理器5802对指令的执行和对存储器5803的访问。在这样的实施例中，操作系统5812可以被配置为把对处理器5802和存储器5803的访问虚拟化到实体，这些实体依赖于操作系统5812来在电子设备5801上执行。在另一实施例中，O/S下层安全代理5820可以被配置为在物理存储器地址的基础上捕获对处理器5802指令的执行和对存储器5803的访问。在这样的实施例中，尽管存储器5803的内容被示出为是连续的，但这样的内容可以是散布在物理存储器的不同部分当中。

[0719] 一旦已经检测到加载或卸载驱动程序的尝试，O/S下层安全代理5820可以被配置为采取任何合适的矫正动作。例如，可以隔离做出尝试的实体或将其从电子设备5801移除。在另一示例中，可以隔离所加载的驱动程序或将其从电子设备5801移除。

[0720] 在操作中，O/S下层安全代理5820可以在电子设备5801上运行以便保护驱动程序的加载和卸载。O/S下层安全代理5820可以查阅安全规则5808以判断对于加载和卸载驱动程序要保卫5801的什么资源。然后，O/S下层安全代理5820可以在一个或多个控制结构中设置标志以便捕获对这样的资源的已尝试访问。例如，O/S下层安全代理5820可以设置用于以下的标识：在存储器5803中的地址(D)处的SwLoadDriver()的执行、在地址(A)处的NtLoadDriver()的代码部分的访问、在地址(E)处的NtLoadDriver()的执行、对在存储器范围(B)中的为新驱动程序分派的空间的访问、对在(C)和(B)之间的范围中的旧驱动程序的空间的访问或盘5824上新驱动程序的映像的读取。

[0721] 应用5826可以访问电子设备5801的一个或多个资源以便尝试加载诸如新驱动程序5830之类的驱动程序和/或卸载诸如旧驱动程序5828之类的驱动程序。例如，应用5826可以调用ZwLoadDriver()5832的实例以便加载新驱动程序5830。O/S下层安全代理5820可以捕获对在地址(D)处的函数的已尝试执行并确定调用实体即应用5826的身份。O/S下层安全代理5820可以扫描应用5826，计算其内容的数字散列或签名，并检查白名单、黑名单、安全规则5808和/或保护服务器以确定应用5826的身份。O/S下层安全代理5820可以判断应用5826是否已知是安全的、已知是恶意的或未知的。基于所调用的函数、应用5826的身份和从O/S内部安全代理5819收集的任何上下文信息，O/S下层安全代理5820可以判断是否允许应用5826调用加载或卸载诸如ZwLoadDriver()5832之类的函数。

[0722] 在另一示例中，应用5826可以调用NtLoadDriver()5834的实例以便加载新驱动程序5830。可以通过直接调用NtLoadDriver()5834b直接实施对NtLoadDriver()5834的调用，或者通过调用例如ZwLoadDriver()5832，ZwLoadDriver()5832又调用NtLoadDriver()5834a的实例。O/S下层安全代理5820可以捕获对在地址(E)处的函数的已尝试执行并判断从电子设备5801中的什么实体做出调用。O/S下层安全代理5820可以重复地单步执行函数或执行栈区以便做出这样的判断。对于在执行链中找到的每一实体，O/S下层安全代理5820可以做出与在上面的示例中对ZwLoadDriver()做出的相似的判断。具体地，O/S下层安全代理5820可以判断是否出于这样的目的通过由操作系统5812提供的函数适当地访问子函数。因而，如果O/S下层安全代理5820判断是从ZwLoadDriver()做出该调用，那么，可以授权该调用。然而，如果直接地从应用5826做出该调用，且因而O/S下层安全代理5820判断不是通过经授权信道做出该调用，那么，可以判断该调用是恶意的。

[0723] 在又一示例中，O/S下层安全代理5820可以捕获到用于加载或卸载驱动程序的函

数或子函数(例如在地址(A)处的NtLoadDriver())的代码部分的意图的跳转、分支或其他执行。O/S下层安全代理5820可以判断是否在NtLoadDriver()或另一经授权实体内做出意图的跳转、分支或执行。如果不是,O/S下层安全代理5820可以判断意图的跳转、分支或执行是恶意的。这样的意图的跳转、分支或执行可以是恶意软件尝试规避调用函数或子函数并直接地加载或卸载驱动程序的结果。

[0724] 在再一个示例中,O/S下层安全代理5820可以捕获对把驱动程序的映像从存储加载到存储器的尝试,例如把新驱动程序5830的映像从盘5824加载到存储器5803的范围(B)中为新驱动程序5830分配的空间。在允许加载之前,O/S下层安全代理5820可以检查盘5824上的新驱动程序5830的映像的内容和/或表征调用实体。如果映像不指示恶意软件,或者如果实体是未知的或已知是安全的,那么,O/S下层安全代理5820可以允许把驱动程序加载到存储器中。O/S下层安全代理5820可以把附加的标志放置在对地址范围(B)的执行或访问上以提供针对新近加载的新驱动程序5830的恶意活动的附加防护。O/S下层安全代理5820可以把驻留在存储器5803中的新驱动程序5830的映像与在盘5824上观察到的映像进行比较,并判断任何改变是否表示在加载过程期间已经被注入的代码。任何这样的代码注入可以指示恶意软件。

[0725] 在进一步的示例中,O/S下层安全代理5820可以捕获对被加载到存储器中的驱动程序的存储器空间(例如在存储器5803中的位置(C)和存储器范围(B)之间为旧驱动程序5828分配的空间)的已尝试访问(例如写入命令)。O/S下层安全代理5820可以捕获这样的访问、确定调用实体,并根据安全规则5808判断这样的实体是否拥有做出这样的改变的权限。这样的已尝试改变可以是尝试手动地从电子设备5801移除驱动程序的恶意软件的部分。

[0726] 可以允许或拒绝在尝试访问用于加载和卸载驱动程序的资源的同时由O/S下层安全代理5820捕获的已尝试动作。O/S下层安全代理5820可以采取附加的矫正措施,例如记录尝试以供用于将来的评估、隔离或移除尝试实体、把该尝试报告给保护服务器或任何其他合适的动作。

[0727] 在操作系统5812的级别运行的反恶意软件软件不能够完全地解决驱动程序的恶意加载和卸载,这是因为实施这样的活动的恶意软件也可以在相同的优先级水平运行。

[0728] 图59A和59B阐释用于保护电子设备中的驱动程序的加载和卸载的方法5900的示例实施例。在步骤5905,可以访问安全规则以判断与驱动程序加载和卸载相关联的资源是受保护的。这样的安全规则可以标识资源以及捕获和评估对资源的已尝试访问的准则。

[0729] 在步骤5910,可以在控制结构中在低于电子设备内的操作系统的级别设置标志。例如,可以为捕获对加载和卸载函数的已尝试执行、加载子函数或卸载子函数的执行、写入被加载到存储器的驱动程序的映像、在加载执行时从存储中的驱动程序的映像读取和/或驱动程序加载和卸载函数和子函数的代码部分的跳转、分支或其他直接执行设置标志。可以为通过对应于以上所描述的尝试的存储器地址、通过存储器页面和/或通过物理存储器访问的虚拟存储器访问设置标志。

[0730] 在步骤5915,可以监视电子设备以便发现对访问与驱动程序的加载和卸载相关联的资源的已捕获尝试。在步骤5920,如果没有捕获尝试,那么,进程5900可以进行到步骤5915以继续监视以便发现已捕获尝试。如果已经捕获了尝试,那么,可以在步骤5925开始应对该尝试。可以在低于电子设备的操作系统的级别实施这样的应对。在步骤5925,可以收集

可用于分析该尝试是不是恶意的信息。例如，可以确定做出该尝试的进程、应用、驱动程序或例程。可以从O/S内部安全代理获得来自电子设备的操作系统中的上下文信息。如果做出了对加载驱动程序的尝试，那么，可以评估盘上驱动程序的映像。

[0731] 在步骤5935，可以判断是否直接访问用于加载或卸载驱动程序的驱动程序的内容而没有使用经授权函数。在一个实施例中，可以判断调用进程或例程是否未经授权访问驱动程序的这样的部分。如果直接访问该内容，那么，在步骤5960可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。可以采取任何合适的矫正动作。如果不是借助于未经授权的函数或从未经授权的存储器位置直接地访问该内容，那么，在步骤5940可以评估与已尝试加载操作相关联的存储中驱动程序映像的内容。可以扫描映像以便发现恶意内容，观察和记录驱动程序布局，计算驱动程序的散列，评估数字证书的创建者，或者可以采取任何其他合适的调查动作。在步骤5943，可以判断关于存储中的盘的映像的信息是否指示该内容是可疑的和/或恶意的。如果该内容是可疑的和/或恶意的，那么，在步骤5960可以拒绝该请求。如果不是，那么，在步骤5945可以判断是否直接地执行加载或卸载子函数而不使用被指派为用于这样的访问的函数。在一个实施例中，可以判断调用进程或例程是否未经授权做出这样的尝试。如果调用进程是未经授权的，那么，在步骤5960可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果不是，那么，在步骤5950可以判断是否由经授权实体调用加载函数或卸载函数，或者是否由经授权实体调用意图的分支、跳转或其他直接执行。如果不是由经授权函数调用函数，那么，在步骤5960可以判断该尝试是可疑的和/或恶意的，且可以拒绝该尝试。如果由经授权函数调用，在步骤5955可以允许该尝试。

[0732] 如果允许该尝试，且该尝试用于加载驱动程序，那么，在步骤5965可以标记为新近加载的驱动程序分配的存储器以供附加捕获。如果尝试加载驱动程序的实体是未知的，或如果所加载的驱动程序的恶意软件状态没有最后确定，则尤其可以采取这样的步骤。因而，可以捕获对驱动程序存储器的后续的读、写或执行。在步骤5970，可以在驱动程序的映像被加载到存储器中时评估在被加载到存储器中的驱动程序映像的内容和/或与在驱动程序的映像被加载之前驻留在存储中时评估驱动程序的映像的结果进行比较。在步骤5975，如果结果不同且指示恶意软件的注入，或者如果新近加载的驱动程序的评估指示恶意软件，那么，在步骤5980可以移除、隔离驱动程序或针对它采取其他合适的矫正动作。否则，在步骤5985可以允许驱动程序执行。

[0733] 在执行步骤5960、5980或5985之后，方法5900可以可选地返回到步骤5915以继续监视以便发现对用于驱动程序加载和卸载的电子设备的资源的已尝试访问。

[0734] 图60是用于操作系统下层捕获和保护把代码加载到存储器的系统6000的示例实施例。系统6000可以包括被配置为在电子设备6001上操作以便捕获对把代码加载到诸如存储器6003之类的存储器中的O/S下层安全代理6020。在一个实施例中，O/S下层安全代理6020可以被配置为捕获对把内核模式代码加载到存储器6003中的尝试。此外，O/S下层安全代理6020可以被配置为使用一个或多个安全规则6008来判断保护电子设备6001的什么资源以便捕获对把内核模式代码加载到存储器6003中的尝试，并基于该尝试和涉及实施尝试的实体判断是否授权这样的尝试。O/S下层安全代理6020可以被配置为允许或拒绝尝试或采取其他矫正动作。

[0735] 电子设备6001可以全部地或部分地由图1的电子设备103、图2的电子设备204、图4

的电子设备404、图7的电子设备701、图9的电子设备901、图12的电子设备1201和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备6001可以包括被耦合到诸如存储器6003之类的存储器的一个或多个处理器6002。处理器6002可以全部地或部分地由图2的处理器208、图4的处理器408、图7的处理器702、图9的处理器1202、图12的处理器1202和/或其任何组合实现，或者被配置为实现它们的功能性。存储器6003可以全部地或部分地由图2的存储器206、图4的存储器406、图7的存储器703、图9的存储器903、图12的物理存储器1203或虚拟存储器1204和/或其任何组合实现，或者被配置为实现它们的功能性。电子设备6001可以包括操作系统6012，操作系统6012可以包括被耦合到一个或多个安全规则6021的O/S内部安全代理6019，或通信上耦合到O/S内部安全代理6019。O/S内部安全代理6019可以通信上耦合到O/S下层安全代理6020。操作系统6012可以全部地或部分地由图1的操作系统112、图2的操作系统212、图4的操作系统412、图7的操作系统713、图9的操作系统913、图12的操作系统1213和/或其任何组合实现，或者被配置为实现它们的功能性。O/S内部安全代理6019可以全部地或部分地由图1的O/S内部安全代理218、图4的O/S内部安全代理418和/或图7的O/S内部安全代理719、图9的O/S内部安全代理919、图12的O/S内部安全代理1219或其任何合适的组合实现，或者被配置为实现它们的功能性。

[0736] O/S下层安全代理6020可以由图1的O/S下层捕获代理104或已触发事件应对程序108、SVMM 216或图2的SVMM安全代理217、图4的固件安全代理440、442、O/S下层代理450或PC固件安全代理444、图5的固件安全代理516或图7的微代码安全代理708或O/S下层代理712、图9的O/S下层捕获代理920或已触发事件应对程序922、图12的O/S下层安全代理1220和/或其任何组合实现，或者被配置为实现它们的功能性。

[0737] 安全规则6008可以由图1的安全规则114、图2的安全规则222、图4的安全规则434、436、438、图5的安全规则518、图7的安全规则707、723、图9的安全规则908、图12的安全规则1208和/或其任何组合实现，或者被配置为实现它们的功能性。安全规则6021可以由图2的安全规则220、图4的安全规则420、图7的安全规则721、图9的安全规则921、图12的安全规则1221和/或其任何组合实现，或者被配置为实现它们的功能性。

[0738] O/S下层安全代理6020和/或O/S内部安全代理6019可以通信上耦合到存储器映射6010。存储器映射6010可以包含存储器6003中操作系统6012的各种实体的页面位置或地址的映射。O/S下层安全代理6020和/或O/S内部安全代理6019可以被配置为访问存储器映射6010，以便判断对于存储器6003中给定的存储器位置或页面，什么进程、动态链接库(“DLL”)、应用、模块或电子设备6020的其他实体与该位置或页面关联。O/S下层安全代理6020和/或O/S内部安全代理6019可以被配置为使用这样的信息，例如以便确定捕获存储器6003的什么部分，或者对于存储器6003的已捕获访问或函数的已捕获执行，判断电子设备6001中的什么实体发起该尝试。此外，O/S下层安全代理6020可以使用存储器映射6010来把附加的已捕获尝试关联到存储器6003的恶意的区域，其中，先前对把代码加载到存储器6003中的尝试被判断为恶意的且与存储器6003的区域相关联。

[0739] 存储6044可以通信上耦合到电子设备6001或驻留在电子设备6001内。存储6044可以包括用于大容量存储的任何合适的介质，包括硬盘、闪速驱动器、随机存取存储器(“RAM”)盘、紧致盘、DVD介质驱动器或任何其他合适的存储器。存储6044可以通过诸如外围组件互连、串行高级技术附件、通用串行总线或火线之类的任何合适的接口通信上耦合到

电子设备6001。

[0740] 电子设备6001可以包括为了把代码加载到存储器6003中而尝试访问电子设备6001的资源的一个或多个应用、驱动程序或其他实体——例如，应用6026或驱动程序6028。应用6026或驱动程序6028可以包括任何进程、应用、程序或驱动程序。应用6026或驱动程序6028可以直接地或通过调用其他例程来尝试把代码加载到存储器6003中，例如通过借助于读、写或执行指令访问存储器6003或存储6044的部分。O/S下层安全代理6020可以被配置为截取这样的已尝试调用或访问、查阅安全规则6008和/或来自O/S内部安全代理6019的上下文信息，以便判断该尝试是否指示恶意软件，并采取任何适当的矫正动作。O/S下层安全代理6020可以被配置为通过捕获对存储器6003或存储6044的访问和/或对处理器6002的使用来做出这样的截取。O/S下层安全代理6020可以被配置为访问安全规则6008和/或存储器映射6010，并判断要捕获对存储器6003或存储6044的什么已尝试访问和/或对处理器6002的什么使用。O/S下层安全代理6020可以被配置为在控制结构中设置对应于要捕获的动作的标志。

[0741] 在一个实施例中，诸如应用6026或驱动程序6028之类的实体可以尝试访问与通过存储器页面把代码加载到存储器6003中相关联的存储器6003的部分，其中，存储器6003已经被操作系统6012虚拟化。在这样的实施例中，O/S下层安全代理6020可以被配置为基于存储器页面捕获对存储器6003的已尝试访问或执行。在另一实施例中，应用6026或驱动程序6028可以尝试访问与把代码加载到存储器中相关联的存储器6003的物理部分。在这样的实施例中，O/S下层安全代理6020可以被配置为基于存储器地址捕获对存储器6003的已尝试访问或执行。

[0742] 存储器6003可以包含与用于把代码加载到存储器6003中的动作相关联的一个或多个内容或位置。O/S下层安全代理6020可以被配置为捕获对任何这样的内容或位置的访问。作为说明性示例给出下文。存储器6003可以包含用于容纳页面表目录6030的页面或地址范围，页面表目录6030可以包含在存储器6003内的其他页面或地址范围的权限6032。例如，权限6032可以包含用于读、写和/或执行在存储器位置(A)、(B)、(C)、(D)和(E)处的内容的权限的组合的设置。存储器6003可以包含在位置(A)处的空空间6034，该空间可以例如由应用6026或驱动程序6028分配。存储器6003可以包含电子设备6001的实体或函数的代码部分，例如在存储器位置(E)处的驱动程序代码部分6036。存储器6003可以包含未分配给任何实体的存储器的部分或范围(只要涉及到操作系统6012)，例如在存储器位置(B)处的未经分配空间。存储器6003可以包含其恶意软件状态(无论是安全的还是恶意的)为未知的驱动程序存储器的部分或范围，例如在存储器位置(C)处的不可信的驱动程序6040的空间。存储器6003可以包含具有不存在的内容6042的存储器的部分或范围。存储器6003中的这样的不存在的内容6042可以包括，例如，其内容已被交换到盘且可以例如驻留在存储6044上的一个或多个已交换文件6046的已交换内容6048中的存储器页面。

[0743] 存储6044可以包含与用于把代码加载到存储器6003中的动作相关联的一个或多个内容或位置。O/S下层安全代理6020可以被配置为捕获对任何这样的内容或位置。作为说明性示例给出下文。存储6044可以包含已交换内容6048可以存储在其中的交换文件6046。已交换内容6048对电子设备6001的实体来说看上去是出现在存储器6003内，但实际上被存储在存储6044中。存储6044可以包含存储应用映像6050的部分或地址。应用映像6050可以

包括应用、驱动程序、DLL或诸如应用6026或驱动程序6028之类的在电子设备6001上执行其他实体的映像。

[0744] O/S下层安全代理6020和/或O/S内部安全代理6019可以被配置为扫描存储器6003和/或存储6044的各部分以便发现恶意软件。这样的扫描可以包括计算在存储器6003和/或存储6044内的内容的数字签名、校验和或散列，以便判断该内容是否与被确定是恶意的内容相同，如由安全规则6008、6021所定义的。然而，一些恶意软件可以通过把自身或其他恶意代码插入到要执行的存储器6003尝试攻击电子设备6001。通过捕获把代码加载到存储器6003中的尝试，O/S下层安全代理6020可以被配置为捕获、扫描或保护最初不放置在例如驱动程序文件、存储器中的映像或存储代码的其他规范方法内的代码。通过最初不把这样的代码放置在驱动程序文件、存储器中的映像或存储代码的其他规范方法内，恶意软件可以尝试避免O/S下层安全代理6020和/或O/S内部安全代理6019扫描存储器6003和/或存储6044的各部分以便发现恶意软件的先前所描述的努力。进一步，无论是否已知所注入的内容或尝试注入的实体是恶意的，把代码注入到存储器6003中的违规方法可以指示尝试这样的注入的实体都是恶意的。

[0745] O/S下层安全代理6020可以被配置为确定用于把代码加载到存储器6003中的安全的或规范的方法，例如使用操作系统加载器。可以测试或映射这样的安全的或规范的方法，以使得操作系统6012所采取的逻辑或步骤是已知的。在捕获对把代码加载到存储器6003中的尝试时，O/S下层安全代理6020可以判断这样的尝试是否匹配用于加载代码的已知方法。例如，如果该尝试涉及把代码加载到存储器的已分配部分，且通过回避操作系统加载器借助于直接写入存储器来尝试这样，则可以判断该尝试是恶意的。

[0746] 为了阻止恶意软件把代码注入到存储器6003中以供处理器6002执行，O/S下层安全代理6020可以被配置为捕获注入代码所要求的、避免把代码加载到存储器6003中以供执行的正规方法的一个或多个已尝试步骤。把代码加载到存储器6003中以供执行的这样的正规方法可以包括，例如，使用操作系统6012的操作系统加载器来读取例如要从存储6044执行的应用、驱动程序或其他实体的映像，把来自这样的映像的代码放置到存储器6003中，且然后，执行已经加载的代码。O/S下层安全代理6020可以被配置为捕获例如避免使用操作系统加载器和/或一违规方式实施这些步骤中的任何的、对加载代码以便执行的尝试。这样的违规方法可以由安全规则6008定义。这样的违规方法可以包括捕获对最初不是从经授权应用映像6050中的文件读取或被放置到存储器6003中的等效应用映像的代码的任何已尝试加载。

[0747] 在一个实施例中，O/S下层安全代理6020可以被配置为捕获尝试把代码写入到存储器6003中以及尝试随后执行该代码。例如，O/S下层安全代理6020可以是被配置为捕获地址(A)处的尝试写入，地址(A)是新近分配的空间6034的部分。这样的已尝试写入自身不指示恶意软件，因此可以记录该尝试并将其用于将来的引用。可以扫描、记录或以另外方式评估所写入的内容。O/S下层安全代理6020可以被配置为捕获对被写入到地址(A)的后续已尝试执行。O/S下层安全代理6020可以被配置为确定：因为对内容的已尝试执行跟随在先前已捕获的内容的加载之后，内容的加载包括把代码加载到存储器6003中。O/S下层安全代理6020可以被配置为从诸如尝试的源、尝试的目标尝试、所加载的内容、来自O/S内部安全代理6019的信息或任何其他合适的信息之类的上下文信息判断代码的加载是否指示恶意软

件。例如，在检测到已经把加载代码到存储器且尝试执行时，O/S下层安全代理6020可以被配置为扫描所加载的内容并判断是否已知该代码是恶意软件。在另一示例中，在检测到对代码的加载和已尝试执行时，O/S下层安全代理6020可以被配置为判断是否使用诸如操作系统加载器之类的操作系统6012的正常函数来加载代码。如果使用其他非标准方法例如来自未知的驱动程序的直接写入来加载代码，那么，O/S下层安全代理6020可以被配置为判断对代码的已尝试加载是可疑的并阻止代码的执行。下面描述O/S下层安全代理6020可以捕获对存储器6103中的代码的已尝试加载和执行以及O/S下层安全代理6020可以判断这样的尝试是可疑的或恶意的方法的其他示例。

[0748] 在一个实施例中，O/S下层安全代理6020可以被配置为捕获从存储6044对应用映像6050的已尝试读取，以及后续把内容写入到诸如部分6034之类的存储器6003的新的部分以及后续的执行。O/S下层安全代理6020可以被配置把驻留在盘上的映像与驻留在存储器中的映像进行比较。如果两个映像不同的，例如代码部分的签名或文件布局不同，那么，这样的差异可以指示代码已经被注入到驻留在存储器6003中的映像。这样的注入可以被判断为恶意的。然而，O/S下层安全代理6020可以不捕获从盘对应用映像6050的已尝试读取，且因而O/S下层安全代理6020可能无法把该映像与被加载到存储器6003中的映像进行比较。

[0749] 图61是应用如何收集这样的已注入代码并将其放置在存储器6003中以供执行的示例阐释。应用6102可以由图60的应用6026或驱动程序6028实现。O/S下层安全代理6120可以由图60的O/S下层安全代理6020实现。存储器6103可以由图60的存储器6003实现。存储器6103可以包括用于应用6130和驱动程序6132的存储器空间。每一存储器空间6130、6132可以包括它们各自的实体的代码部分6138、6140的空间。应用6102可以从恶意软件服务器6104下载代码。恶意软件服务器6104可以是网站、网络上的另一电子设备或通信上耦合到应用6102的任何其他实体。应用6102可以把这样的代码写入到应用6130自身的代码部分6138、6140，或者写入到诸如驱动程序6132之类的另一实体。O/S下层安全代理6120可以被配置为根据在此给出的各种教导截取和评估这样的尝试。例如，O/S下层安全代理6120可以被配置为捕获已尝试写入和执行、评估写入尝试的内容、评估应用6102的身份并评估目标存储器位置。

[0750] 图62A和图62B阐释应用如何收集已注入代码以便放置在存储器6003中的另一示例。应用1 6202和应用2 6204可以由图60的应用6026或驱动程序6028或其他类似实体实现。O/S下层安全代理6220可以由图60的O/S下层安全代理6020实现。存储器6203可以由图60的存储器6003实现。盘6244可以由图60的存储6044实现。

[0751] 图62A示出把诸如应用1之类的应用的映像6252从盘6244加载到存储器6203的示例阐释。操作系统加载器可以被配置为读取应用1 6250的盘映像6250，并把它写入到存储器6203中应用1的映像6234。应用1的盘映像6250可以包括经加密的内容6252，该内容可以被复制并作为经加密的内容6236保留在存储器6203中应用1的映像6234中。O/S下层安全代理6220可以捕获由操作系统加载器调用的已尝试读取和写入。然而，在一个实施例中，O/S下层安全代理6220可能没有理由怀疑任何恶意活动涉及映像的加载，这是由于两个映像将不会不同，且该动作是由操作系统的正常机制实施的。

[0752] 图62B示出在应用的映像被加载到存储器之后实施的可能动作的示例阐释。应用2 6204可以在电子设备上操作。应用1 6202可以在电子设备上操作且可以发送指令以便解密

经加密的内容6236。这样的命令可以引起经加密的内容6236被解密，且经解密内容6238a被写入到应用1的映像6234的存储器空间，和/或作为经解密内容6238b被写到应用2的映像6240的存储器空间。因而，应用1 6202可以尝试把代码注入到其自己的代码部分或另一实体的代码部分。应用1 6202可以把经解密内容6238a写到先前的经加密的内容6236的位置上，或者写到应用1的映像6234的另一存储器部分中。来自经解密内容6238的代码可以是恶意的。因而，O/S下层安全代理6220可以被配置为根据在此给出的各种教导捕获和评估对诸如经解密内容6238之类的代码的已尝试写入和执行。例如，O/S下层安全代理6220可以被配置为截取已尝试写入和执行、评估写入尝试的内容、评估应用6102的身份并评估目标存储器位置。

[0753] 返回到图60，O/S下层安全代理6020可以确定存储器6003的部分的布局，包括电子设备6001的已知实体的位置。在一个实施例中，这样的布局可以包括内核虚拟存储器的布局，该布局包含操作系统6012的内核映像及其组件，以及可信驱动程序(和它们的代码和数据部分)和其他驱动程序(和它们的代码和数据部分)。这样的布局可以被存储在存储器映射6010中。通过剖析操作系统6012的启动和操作、通过检验操作系统6012和可信驱动程序的组件、通过扫描或检验数字证书或任何其他合适的方法，O/S下层安全代理6020可以从安全规则6008或保护服务器确定这样的布局。O/S下层安全代理6020也可以确定存储器6003的哪些部分未被分配给电子设备6001的任何实体。内核存储器到物理存储器的映射可以被包含在存储器映射6010中，描述内核存储器中的给定页面的关联物理存储器地址。存储器映射6010也可以包含对存储器的什么部分已经被分配、未被分配、与不存在的内容或交换文件相关联的描述。操作系统6012可以具有诸如页帧号数据库(“Pfn”)之类的数据库中的信息，该数据库描述由操作系统6012管理的虚拟页面。可以使用对这样的数据库的遍历来确定已分配、已映射、包含不存在的内容等等的存储器页面。

[0754] 一旦已知存储器6003的布局，O/S下层安全代理6020可以被配置为监视未经分配区域中的存储器的分配和执行。O/S下层安全代理6020可以被配置为捕获这样的已尝试操作并扫描要写的内容。用于这样的操作的步骤可以包括实体分配具有启用的写入权限的存储器6003的部分、把新内容写到存储器6003的部分、改变存储器6003的部分的权限以便允许执行以及调用存储器6003的部分以供执行。O/S下层安全代理6020可以被配置为捕获这些步骤并在判断这些步骤是否指示恶意软件之前停止已尝试执行。例如，O/S下层安全代理6020可以被配置为捕获对在位置(A)处的存储器6003的未经分配部分6034的尝试分配、对把内容写入到部分6034的尝试、写入页面表目录6030的权限6032以便把位置(A)的权限从“写”改变成“写/执行”以及对位置(A)处的权限的已尝试执行。O/S下层安全代理6020可以被配置为捕获用于分配存储器的任何合适的函数，例如以下的Windows™函数：

[0755] • MmAllocateContiguousMemory

[0756] • MmAllocateContiguousMemorySpecifyCache

[0757] • MmAllocateSpecialPool

[0758] • MmAllocateContiguousMemorySpecifyCacheNode

[0759] • MmAllocateIndependentPages

[0760] • MmAllocateMappingAddress

[0761] • MmAllocateNonCachedMemory

[0762] • MmAllocatePagesForMdl，或

[0763] • MmAllocatePagesForMdlEx

[0764] O/S下层安全代理6020可以被配置为捕获对应于要捕获对存储器的已尝试分配的这些函数的存储器6003中的页面的执行或存储器6003中的物理地址的执行。O/S下层安全代理6020确定函数的调用者并扫描调用者的内容以便发现已知的恶意软件。例如，如果对分配的尝试是借助于已知的、受保护的分配函数的子函数做出的，那么，这样的尝试可以指示恶意软件尝试规避与函数相关联的保护。O/S下层安全代理6020可以被配置为把已分配存储器部分添加到存储器映射6010以供将来引用。O/S下层安全代理6020可以被配置为记录已尝试存储器分配，以及诸如调用者之类的上下文信息。

[0765] O/S下层安全代理6020可以被配置为捕获用于把代码写到存储器的任何合适的函数或方法。诸如应用6026或驱动程序6028之类的实体可以使用由操作系统6012提供的函数或方法来把代码输入到存储器6003。然而，这些实体可以尝试直接写到存储器以便避免检测。因而，如果O/S下层安全代理6020捕获对把代码输入到存储器6003的尝试，其中不使用这样的指定函数就做出该尝试，则O/S下层安全代理6020可以被配置为确定把代码加载到存储器6003中的尝试是可疑的。

[0766] O/S下层安全代理6020可以被配置为捕获用于改变存储器位置的权限的任何合适的函数或方法。例如，为了改变存储器页面权限，可以调用下列Windows™函数中的一个：

[0767] • MmProtectMdlSystemAddress

[0768] • NtProtectVirtualMemory

[0769] • MiGetPageProtection

[0770] • MiQueryAddressState

[0771] • MiSetProtectionOnSection

[0772] • MiGetpageProtection

[0773] • MiProtectPrivateMemory

[0774] 因而，O/S下层安全代理6020可以被配置为通过监视存储器中这些函数的位置的执行来捕获这些函数中的任何的执行。O/S下层安全代理6020可以被配置为捕获对页面表目录6030及其中包含的权限6032的任何尝试写入。如果捕获了对权限权限6032的尝试写入，同时没有捕获用于改变权限6032的经批准的函数(例如上面所列出的那些)的执行，那么，O/S下层安全代理6020可以被配置为确定对写入的尝试指示恶意软件。这样的尝试可以是以下的结果：恶意软件直接地编辑页面表目录中用于存储器位置的权限的条目的结果(例如“读”改为“写”或“执行”)，以使得可以用恶意软件写入相应的位置或者不引起与操作系统6012函数相关联的保护就执行相应的位置。

[0775] 如果捕获到后面是尝试执行的对存储器的尝试写入，且O/S下层安全代理6020判断已经对诸如位置(B)处的未经分配存储器部分6038之类的存储器的未经分配部分做出这样的尝试，那么，O/S下层安全代理6020可以被配置为判断这样的尝试是恶意的。O/S下层安全代理6020可以被配置为判断是否已经对诸如位置(B)处的未经分配部分6038之类的未经分配空间做出了写入和执行内容的尝试。下面-O/S安全代理6020可以被配置为通过访问存储器映射6010或引用先前已捕获的对分配函数的执行来做出这样的判断。在一个实施例中，如果在被判断为未经分配存储器6003中做出的意图的写入和执行，则恶意软件已经分

配存储器而不使用操作系统6012定义的例程。在这样的实施例中，O/S下层安全代理6020可以判断该尝试是对把代码加载到存储器6003中以供执行的恶意尝试。在另一实施例中，如果在已经被分配给电子设备6001的实体的存储器6003中做出已尝试写入和执行，如存储器映射6010中所示出的，那么，O/S下层安全代理6020可以被配置为检查指令的调用者是否拥有权限或经授权为把代码写入到另一实体。如果不是，那么，这样的尝试可以指示恶意软件尝试把代码注入到属于另一函数的存储器6003的部分，这可能是恶意的。

[0776] O/S下层安全代理6020可以被配置为捕获在诸如位置(B)处的部分6038之类的被指定为未经分配的存储器6003的部分中的任何意图的写、读或执行，或改变权限6032中的未经分配部分6038的权限的尝试。起源于不同于经授权分配函数(这些函数又被操作系统6012的经授权实体调用)的实体的任何这样的尝试可以被判断为是恶意的，而不考虑调用者的恶意软件状态是不是未知的。如果存储器位置经历了尝试写入，那么，可以判断该尝试指示代码正被加载到存储器中。如果存储器位置经历了已尝试执行，那么，可以判断该尝试指示新近加载的代码正被执行。

[0777] O/S下层安全代理6020可以被配置为捕获非可信驱动程序6040的任何已尝试写入或执行或与非可信驱动程序6040相关联的存储器位置(C)的权限6032的改变。非可信驱动程序6040可以包括其恶意软件状态未知的驱动程序。黑名单或白名单中可能不存在用于非可信驱动程序6040的签名或散列的条目。非可信驱动程序6040可以是系统6000或类似系统未遇到的安全的、新的驱动程序，或者非可信驱动程序6040可以是系统6000未遇到的恶意软件的置换。一旦捕获了对非可信驱动程序6040的已尝试写入或执行，O/S下层安全代理6020可以被配置为应用附加安全规则6008以便评估这样的已捕获操作。尤其，O/S下层安全代理6020可以捕获并仔细检查对写入且随后执行代码的尝试，以便判断是否对诸如操作系统6012的内核模块或已知的驱动程序之类的已知元素尝试这样的操作。如果是，则可以阻止这样的尝试，且判断非可信驱动程序6040是恶意软件。

[0778] O/S下层安全代理6020可以被配置为捕获对诸如在地址(D)处的部分6042之类的具有不存在的内容的存储器6003的部分的已尝试写入或执行。可以根据虚拟存储器分配来分配存储器6003的部分6042，但部分6042的内容实际上不出现在存储器6003的物理存储器中。相反，这样的部分6042的内容可以出现在其他地方，例如在存储6044中。不存在的内容可以驻留在交换文件6046中的存储6044中，交换文件6046包含作为页面交换操作的部分的已交换内容6048，其中虚拟存储器6003的内容被移动到盘，以便为存储器6003的物理存储器中的其他元素提供空间。操作系统6012可以被配置为实施这样的交换操作，且可以被配置为在需要已交换内容6048时把已交换内容6048重新加载到物理存储器中。因而，尽管部分6042包含不存在的内容，但O/S下层安全代理6020可以捕获对部分6042的尝试入或执行。如果这样的尝试写入起源于不同于实施页面交换的操作系统6012的实体，那么，该写入是恶意的。例如，可以拒绝不起源于操作系统6012的虚拟存储器管理器的对部分6042的尝试写入。进一步，如果在部分6042的内容是不存在的时做出对部分6042的已尝试执行，那么，这样的已尝试执行可能是恶意的。

[0779] 此外，O/S下层安全代理6020可以捕获诸如对存储6044上的已交换文件6046的已交换内容6048的写入之类的已尝试访问。如果这样的尝试访问起源于不同于经授权实体的实体，那么，写入是恶意的。例如，可以拒绝不起源于操作系统6012的虚拟存储器管理器的

对已交换内容6048的尝试写入，并判断其指示恶意软件。例如，通过捕获盘写入函数或通过捕获存储6044中的输入/输出命令，可以做出这样的捕获。例如，由在存储6044上运行的固件安全代理完成对存储6044中的输入/输出命令的这样的捕获。

[0780] 图63阐释对已交换内容恶意攻击以便注入代码的附加示例。内核虚拟存储器6304可以表示已经被虚拟化的存储器6003的部分。内核虚拟存储器6304的内容可以被映射到它们驻留在其中位置。例如，这样的内容可以被映射到物理存储器6302和/或盘6344。物理存储器6302可以把存储器6003的物理布局阐释为它物理上驻留在电子设备6001中。盘6344可以由图60的存储6044实现。内核虚拟存储器6304的一些部分6306、6310、6314、6318可以未经分配，且对虚拟存储器的用户来说是可用的。内核虚拟存储器6304的其他部分可以包括用于操作系统内核6308和诸如驱动程序1 6312、驱动程序2 6316和驱动程序36320之类的驱动程序的部分。内核虚拟存储器6304中已分配存储器的部分可以映射到物理存储器6302和/或盘6344中的各种非连续部分。示出驱动程序36320中所包括的示例页面，包括页面6322、6324和6326。对应于驱动程序3页面0的页面6326可以被映射到盘6344上的交换文件6350。对应于驱动程序3页面1的页面6324可以被映射到在地址(A)处并继续到地址(B)的物理存储器6302中的地址。对应于驱动程序3页面2的页面6322可以被映射到盘6344中的交换文件6348。

[0781] 在实施交换文件操作时，可以把驱动程序3页面0的内容写到交换文件6350，该内容可以是“XYZ”。驱动程序3页面0的内容因而可以是不存在的。在此期间，恶意软件6352可以把交换文件6350的内容重写为“PDQ”。因而，在反转交换文件操作并从盘6344和交换文件6350读取内容时，新的代码将被加载到内核虚拟存储器6304。恶意软件6352把值直接地写到诸如页面6324之类的其他页面的物理存储器6302的动作也是可能的。

[0782] 返回到图60，O/S下层安全代理6020可以被配置为防止把代码加载到与带有不存在的内容6042的页面相关联的交换文件6046中。O/S下层安全代理6020可以被配置为捕获来自存储器6003的对交换文件6046的尝试写操作或反之亦然，并确定内容的快照、签名、密码散列、校验和或其他指示。O/S下层安全代理6020可以被配置为通过设置对应于存储6044上的位置的标志或在用于交换存储器页面的函数或例程上的标志来捕获这样的尝试。O/S下层安全代理6020可以被配置为捕获所尝试的来自交换文件6046的读操作和/或写回到存储器部分以便发现不存在的内容6042。O/S下层安全代理6020可以被配置为捕获交换函数的已尝试执行以便完成这样的任务。这样的函数可以包括或调用，例如，IoPageRead()、IoAsynchronousPageWrite()或IoAsynchronousPageWrite()。O/S下层安全代理6020可以被配置为确定从交换文件6046读取的和/或被写到不存在的内容6042的内容的快照、签名、密码散列、校验和或其他指示，并判断该内容是否已经被改变。如果是，O/S下层安全代理6020可以被配置为确定已经用代码注入交换文件6046的内容。O/S下层安全代理6020可以被配置为阻止新的内容的执行或采取任何其他矫正措施。

[0783] O/S下层安全代理6020可以被配置为确定对存储器6003的特定写入是通过捕获对改变内容的存储器位置的权限6032以便给予执行权限的后续尝试来加载代码。此外，O/S下层安全代理6020可以被配置为确定写入存储器6003且随后尝试写入权限6032以便允许执行存储器的实体。这样的实体通常是已知的、可信的实体，例如操作系统加载器。因而，O/S下层安全代理6020可以被配置为判断是否由诸如操作系统加载器之类的已知的、可信的实

体产生把代码加载到存储器6003的尝试。如果不是，那么，O/S下层安全代理6020可以被配置为拒绝这样的尝试并判断该尝试和/或实体指示恶意软件。

[0784] O/S下层安全代理6020可以被配置为捕获对从存储器6003复制合法的函数代码、把函数代码复制到诸如新近分配的部分6034之类的新位置且然后执行所复制的代码的尝试。这样的尝试可以是恶意软件对运行系统函数而不从例如操作系统6012或持有函数的驱动程序获得授权的尝试。

[0785] 在捕获已尝试操作时，O/S下层安全代理6020可以被配置为基于来自O/S内部安全代理6019的上下文信息、驱动程序的调用栈区和/或存储器映射6010标识操作的调用者。使用包括行为规则、白名单或黑名单的安全规则6008，O/S下层安全代理6020可以被配置为判断是否已知调用者是恶意的。如果这样的恶意状态是未知的，那么，O/S下层安全代理6020可以被配置为允许写操作继续。然而，O/S下层安全代理6020可以被配置为停止后续的执行尝试。一些操作可以涉及多次尝试写入以便完全加载所注入的代码。O/S下层安全代理6020可以在停止执行之前允许这样的写入，以便完全地确定和表征所注入的代码。

[0786] 在由可信的或非可信的实体分配、释放或写入存储器6003时，O/S下层安全代理6020可以被配置为把这样的信息记录在存储器映射6010中。O/S下层安全代理6020可以标记易受没有执行权限的代码注入攻击感染的存储器6003的部分。存储器6003的这样的部分的已尝试执行可以引起已捕获操作。O/S下层安全代理6020可以被配置为定位存储器映射6010中与已尝试执行相关联的存储器的部分并确定所关联的实体。O/S下层安全代理6020可以扫描代码字节以便发现已知的恶意软件的指示或评估在写代码时观察到的可能恶意的行为。

[0787] 如果判断代码的已尝试加载和后续的已尝试执行是可疑的或以另外方式指示恶意软件，则O/S下层安全代理6020可以被配置为采取合适的矫正动作。在一个实施例中，O/S下层安全代理6020可以被配置为用诸如“NOOP”指令或其他模式之类的哑元信息填充所写的代码。另外，可以用哑元信息填充做出该尝试的实体的存储器位置。在另一实施例中，O/S下层安全代理6020可以被配置为把执行传送给矫正引擎以便清洁和隔离该尝试的源和目标。在又一实施例中，O/S下层安全代理6020可以被配置为用不允许读、写或执行的权限6032标记调用者和目标的存储器位置。类似地，O/S下层安全代理6020可以被配置为捕获对该尝试的调用者或目标的存储器位置的任何后续的尝试读、写或执行。在存储器映射6010中可以把与尝试相关联的存储器位置标记为是恶意的。

[0788] 图64是诸如在存储器的部分已经被确定为是恶意的之后的图60的存储器映射6010之类的存储器映射6400的示例实施例。存储器映射6400的各部分6402、6406、6410、6414、6418、6422可以被示出为未经分配。存储器映射可以示出用于操作系统内核6404、驱动程序1 6408、驱动程序2 6412和驱动程序3 6416的位置。被确定为与注入和执行代码的恶意尝试相关联存储器的部分可以被指定为恶意软件6420。可以设置拒绝对恶意软件部分6420的读、写和执行的权限。恶意软件部分6420可以反映，例如，代码被注入到其中的存储器的部分或从中做出注入尝试的存储器的部分。随后，可以由诸如图60的O/S下层安全代理6020之类的O/S下层安全代理捕获来自未知的部分6424的恶意软件部分6420的已尝试访问。这样的O/S下层安全代理可以被配置为设置用于捕获对对应于恶意软件部分6420的存储器6003的部分的访问的标志。例如，未知的部分6424可以做出对读取先前被写到部分

6420的代码的尝试。在另一示例中，可以由未知部分6424做出对执行先前被写到部分6420的代码的尝试。O/S下层安全代理可以被配置为判断对被指定为恶意软件的存储器的任何这样的尝试访问本身是恶意的。因而，O/S下层安全代理可以阻止尝试并把先前未知部分642重新指定为恶意软件部分6424。O/S下层安全代理可以被配置为对新近指派的恶意软件部分6424采取矫正动作，例如用哑元数据填充它、把它传送给用于清洁和隔离它的进程、用拒绝读、写或执行访问的权限标记恶意软件部分6424，和/或捕获对恶意软件部分6424的后续访问。

[0789] 图65是用于加载和执行电子设备中的存储器中的代码的操作系统下层捕获的方法6500的示例实施例。在步骤6505，可以访问安全规则以判断与加载和执行存储器中的代码相关联的资源。这样的安全规则可以标识资源，以及捕获和评估对资源的已尝试访问的准则。

[0790] 在步骤6510，可以在控制结构中在低于电子设备内的操作系统的级别设置标志。可以例如为捕获代码的已尝试注入和后续执行设置标志。可以为通过对应于以上所描述的尝试的存储器地址、通过存储器页面和/或通过物理存储器访问的虚拟存储器访问设置标志。

[0791] 在步骤6515，可以监视电子设备以便发现与把代码诸如到存储器中相关联的资源的已捕获尝试。在步骤6520，如果没有捕获尝试，那么，进程6500可以进行到步骤6515以继续监视以便发现已捕获尝试。如果已经捕获了尝试，那么，可以在步骤6525开始应对该尝试。可以在低于电子设备的操作系统的级别实施这样的应对。在步骤6525，可以收集可用于分析该尝试是不是恶意的信息。例如，可以确定做出该尝试的进程、应用、驱动程序或例程以及该尝试的目标。可以从O/S内部安全代理获得来自电子设备的操作系统中的上下文信息。如果做出了对注入代码的尝试，那么，可以扫描调用者的映像。

[0792] 在步骤6530，可以判断是否已知做出加载或注入尝试的实体未经授权做出这样的尝试。如果是，那么，在步骤6565，可以拒绝该尝试且可以采取任何合适的矫正动作。如果不是，这意味着该实体的恶意软件状态仍然未知，那么，在步骤6535可以允许加载尝试，且可以判断该加载尝试是否潜在可疑的，这取决于后续已尝试执行的环境。如果该加载尝试不是潜在可疑的，那么，方法可以进行到步骤6515以便继续监视电子设备。

[0793] 如果加载尝试是仍然潜在地可疑的，那么，在步骤6545可以监视电子设备以便发现对被写入的代码的后续执行尝试。如果没有捕获尝试，那么，进程6500可以重复步骤6545或并行进行到步骤6515以继续监视以便发现已捕获尝试。如果已经捕获执行尝试，那么，可以在步骤6550开始应对该尝试。可以在低于电子设备的操作系统的级别实施这样的应对。在步骤6550，可以收集可用于分析与加载尝试组合的执行尝试是不是恶意的信息。例如，可以确定做出该尝试的进程、应用、驱动程序或例程以及该尝试的目标。可以从O/S内部安全代理获得来自电子设备的操作系统中的上下文信息。可以扫描执行尝试的调用者的映像以便发现恶意软件的指示。

[0794] 在步骤6555，可以判断与已尝试加载组合的已尝试执行是否指示恶意软件。如果是，那么，在步骤6565可以拒绝该尝试，且可以采取任何合适的矫正动作。如果不是，那么，在步骤6560可以允许执行和加载尝试，且方法6500可以继续到步骤6515，以便可选地继续监视电子设备。

[0795] 使用图1-2、4-5、7、9-10、12-13、15-18、20-23、26-27、29、31、33、35、38-39、41、45、47-48、50-52、54-56、58或60-64的系统中的任何，或者可操作为实现方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900、6500的任何其他系统，可以实现方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900、6500。因而，方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900、6500的优选初始化点和它们各自的步骤的次序可以取决于所选择的实现。尽管图3、6、8、11、14、19、24-25、28、30、32、34、36-37、40、42-44、46、49、53、57、59和65揭示了关于示例方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900和6500要采取的特定数量的步骤，但可以以比各图中所叙述的那些步骤更多或更少的步骤执行各方法。另外，图3、6、8、11、14、19、24-25、28、30、32、34、36-37、40、42-44、46、49、53、57、59和65揭示了对于各方法要采取的步骤的特定次序，但可以以任何合适的次序完成包括这些方法的各步骤。此外，方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900和6500的一些或全部步骤可以与来自其他方法的步骤组合。在一些实施例中，可以可选地忽略、重复或组合一些步骤。在一些实施例中，方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900和6500中的一种或多种的一些步骤可以与彼此的其他步骤并行执行。在某些实施例中，可以部分地或完全地以在计算机可读介质中包含的软件来实现方法300、600、800、1100、1400、1900、2400、2500、2800、3000、3200、3400、3600、3700、4000、4200、4300、4400、4600、4900、5300、5700、5900和6500。

[0796] 出于本公开内容的目的，计算机可读介质可以包括可以在一段时间内保留数据和/或指令的任何媒介或媒介的集合。计算机可读介质可以包括但不限于诸如以下的存储介质：直接访问存储设备(例如，硬盘驱动器或软盘)、顺序访问存储设备(例如，磁带驱动器)、紧致盘、CD-ROM、DVD、随机存取存储器(RAM)、只读存储器(ROM)，电可擦除可编程只读存储器(EEPROM)和/或闪速存储器；以及非暂态通信介质；和/或前述的任何组合。

[0797] 图1-2、4-5、7、9-10、12-13、15-18、20-23、26-27、29、31、33、35、38-39、41、45、47-48、50-52、54-56、58或60-64中的系统中的一个或多个可以与相同的系统的其他部分组合。

[0798] 尽管已经详细描述了本公开内容，但应理解，可以在不偏离由所附权利要求界定的本公开内容的精神和范围的前提下做出各种改变、替换和变更。

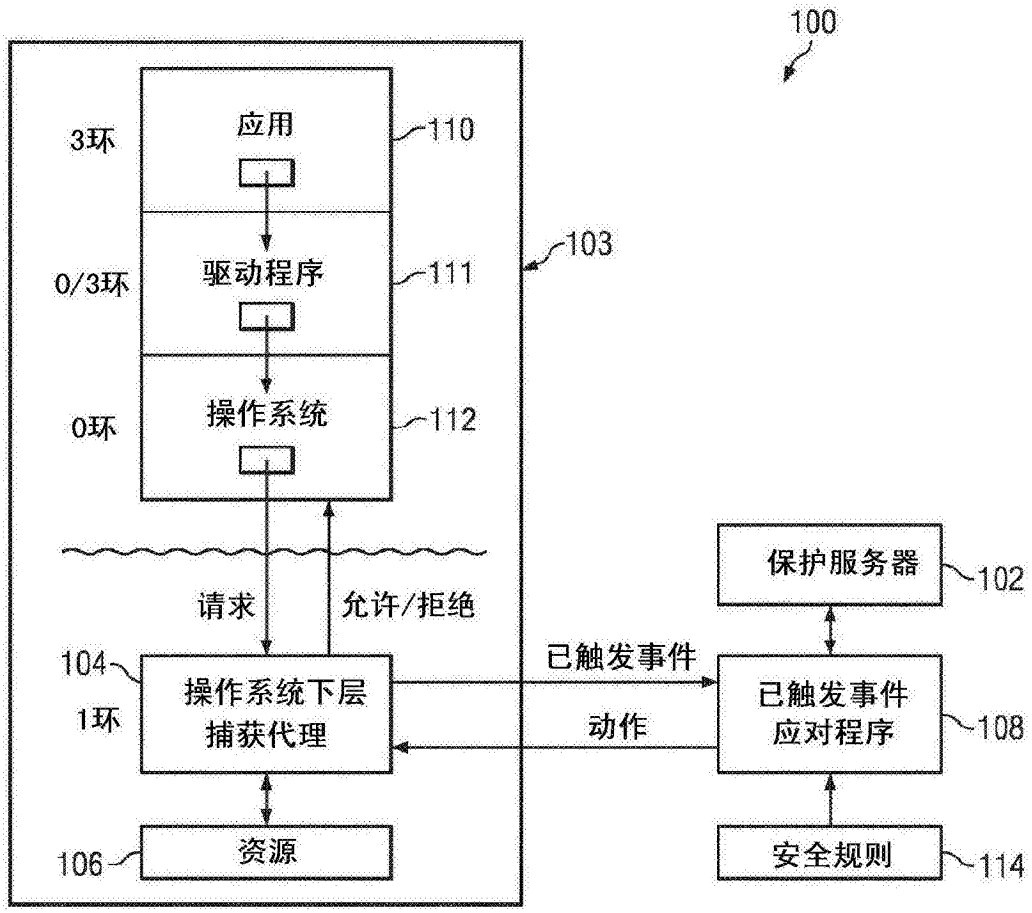

100
3环
应用
110
0/3环
驱动程序
111
103
0环
操作系统
112
请求
允许/拒绝
保护服务器
102
104
1环
操作系统下层
捕获代理
已触发事件
动作
已触发事件
应对程序
108
106
资源
安全规则
114

图1

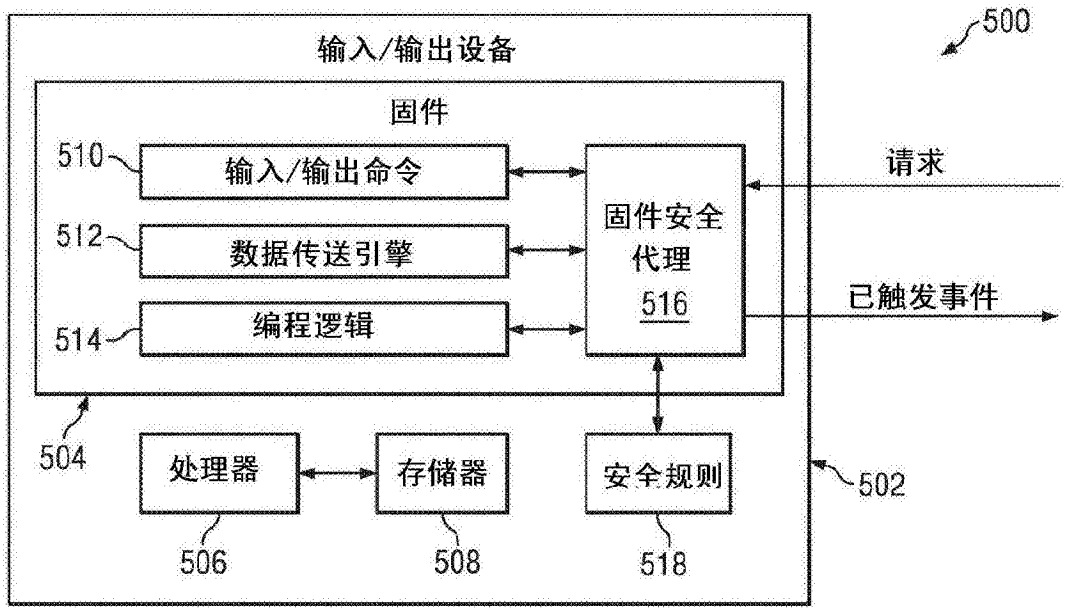

500
输入/输出设备
固件
510
输入/输出命令
512
数据传送引擎
514
编程逻辑
固件安全
代理
516
请求
已触发事件
504
处理器
存储器
安全规则
502
506
508
518

图5

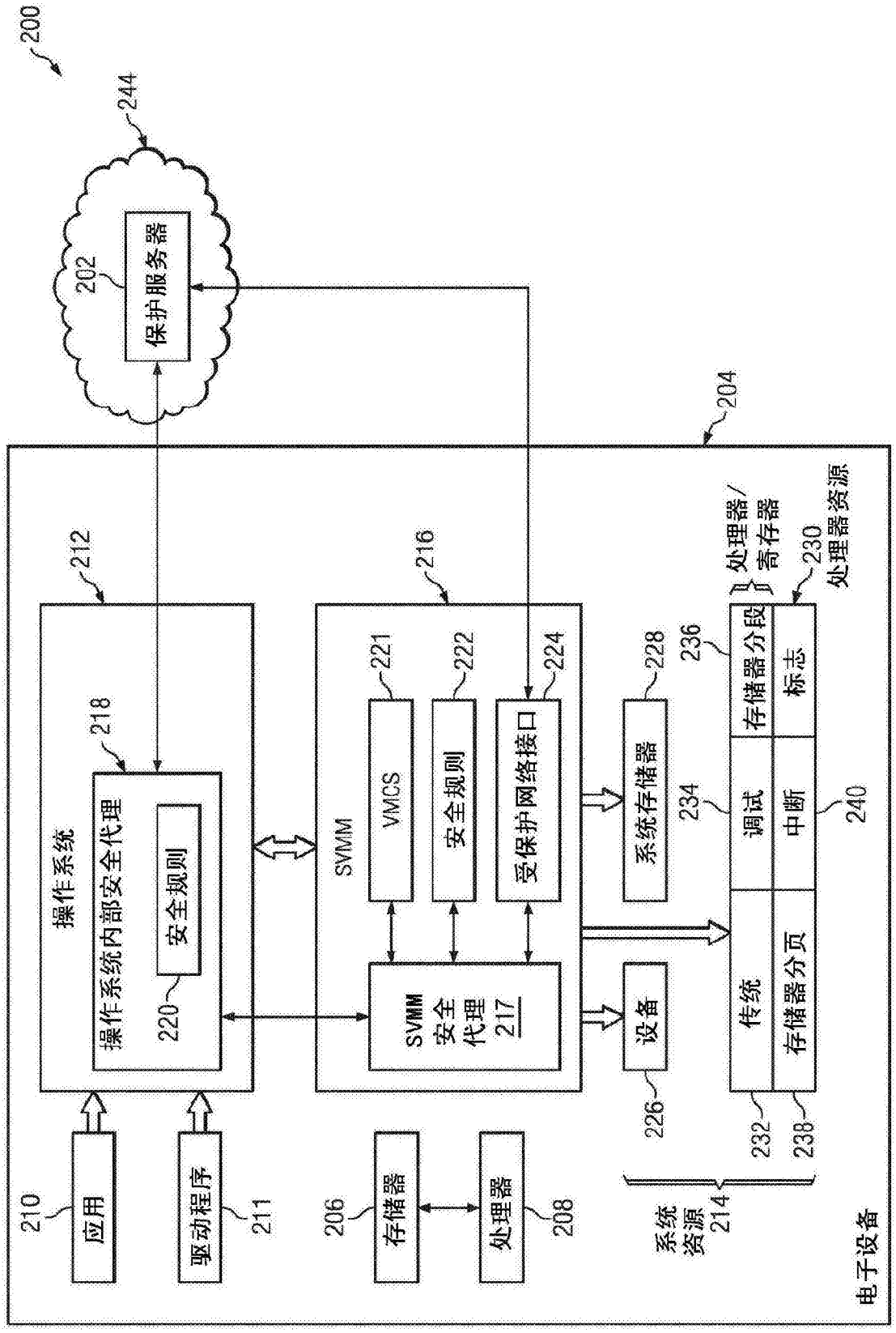

200
244
保护服务器
202
204
212
操作系统
操作系统内部安全代理
218
安全规则
220
216
SVMM
VMCS
221
安全规则
222
受保护网络接口
224
SVMM 安全代理 217
系统存储器
228
设备
226
处理器/寄存器
230
处理器资源
236
存储器分段
标志
调试
234
中断
240
传统
存储器分页
232
238
系统资源 214
应用
210
驱动程序
211
存储器
206
处理器
208
电子设备

图2

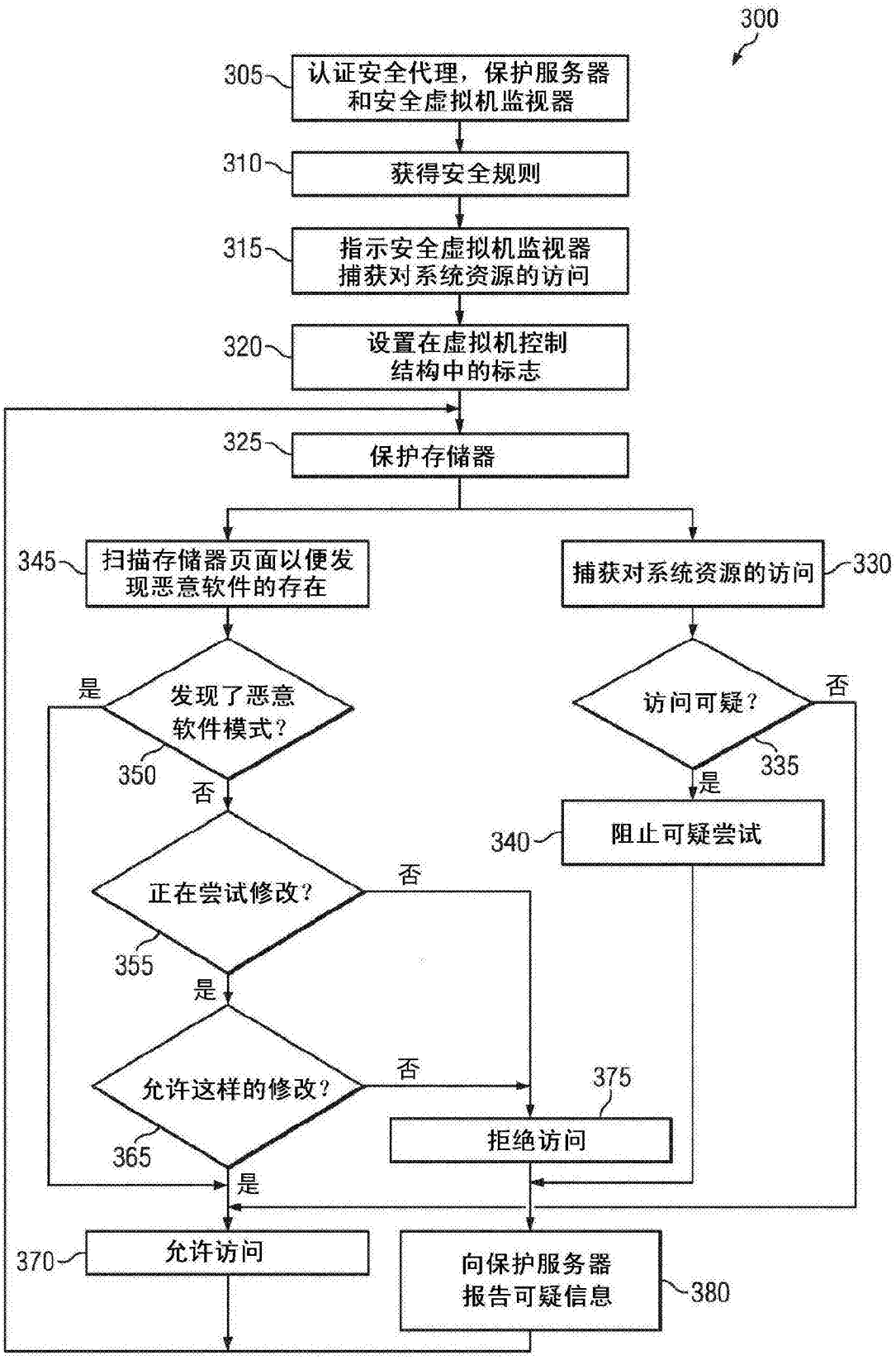

300
305
认证安全代理，保护服务器和安全虚拟机监视器
310
获得安全规则
315
指示安全虚拟机监视器捕获对系统资源的访问
320
设置在虚拟机控制结构中的标志
325
保护存储器
345
扫描存储器页面以便发现恶意软件的存在
捕获对系统资源的访问
330
是
发现了恶意软件模式？
350
否
访问可疑？
否
335
是
340
阻止可疑尝试
正在尝试修改？
否
355
是
允许这样的修改？
否
375
拒绝访问
365
是
370
允许访问
向保护服务器报告可疑信息
380

图3

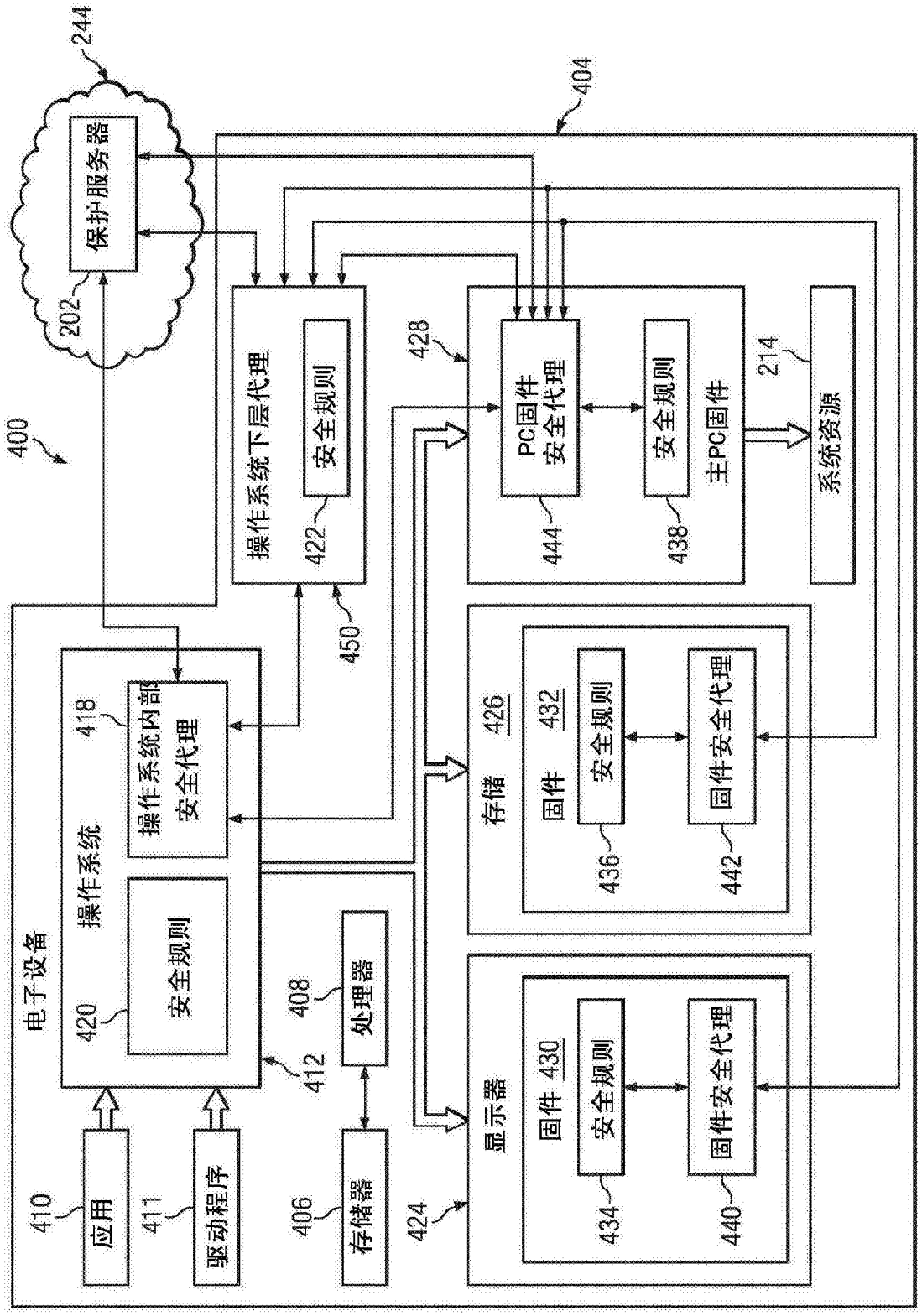

244
保护服务器
202
404
400
操作系统下层代理
安全规则
422
428
PC固件
安全代理
444
安全规则
438
主PC固件
214
系统资源
450
电子设备
操作系统
418
操作系统内部
安全代理
420
安全规则
存储
426
固件
432
安全规则
436
固件安全代理
442
408
处理器
412
显示器
固件
430
安全规则
434
固件安全代理
440
410
应用
411
驱动程序
406
存储器
424

图4

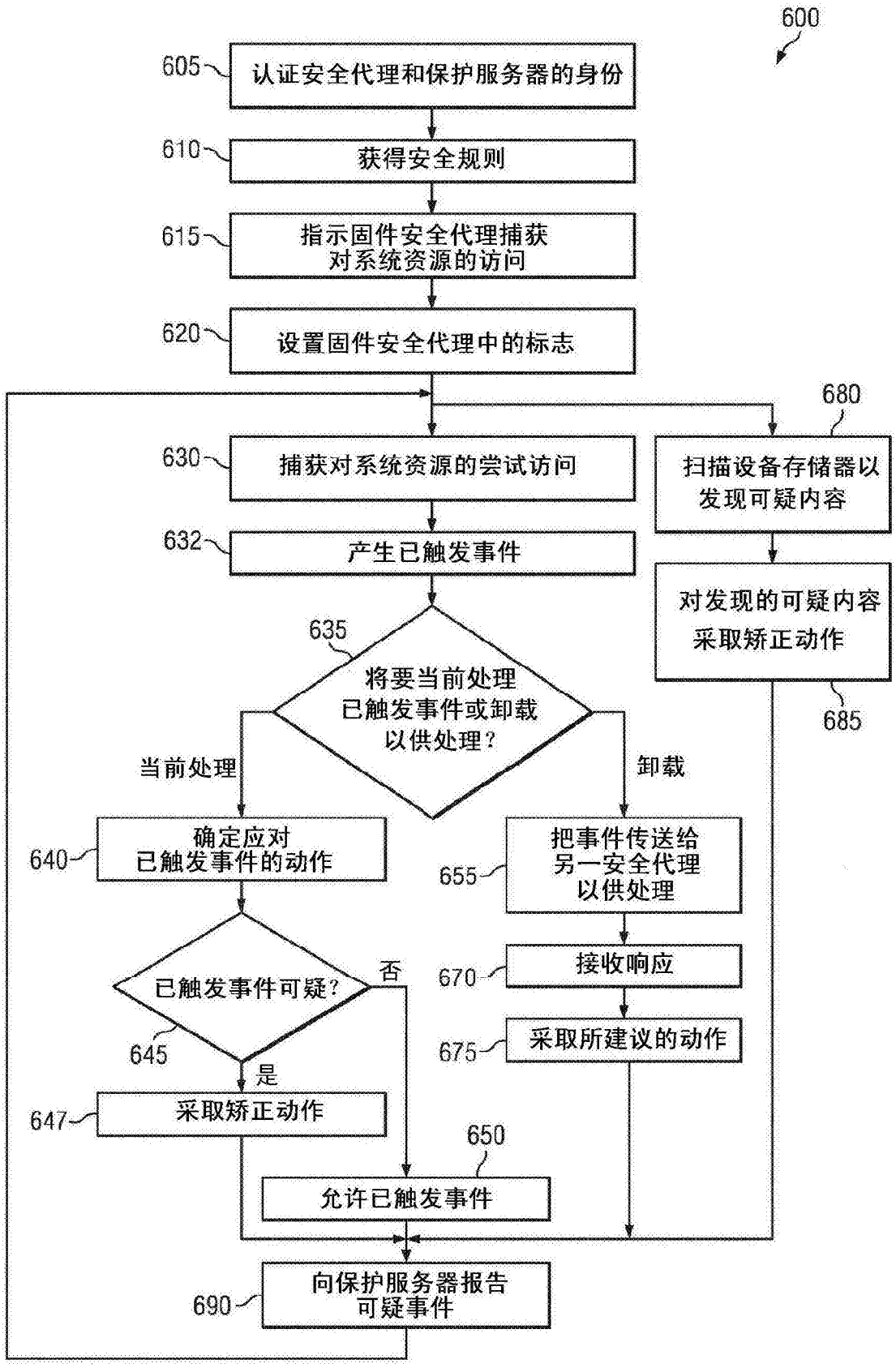

600
605
认证安全代理和保护服务器的身份
610
获得安全规则
615
指示固件安全代理捕获
对系统资源的访问
620
设置固件安全代理中的标志
680
630
捕获对系统资源的尝试访问
扫描设备存储器以
发现可疑内容
632
产生已触发事件
对发现的可疑内容
采取矫正动作
685
635
将要当前处理
已触发事件或卸载
以供处理？
当前处理
卸载
640
确定应对
已触发事件的动作
655
把事件传送给
另一安全代理
以供处理
已触发事件可疑？
否
670
接收响应
645
675
采取所建议的动作
是
647
采取矫正动作
650
允许已触发事件
690
向保护服务器报告
可疑事件

图6

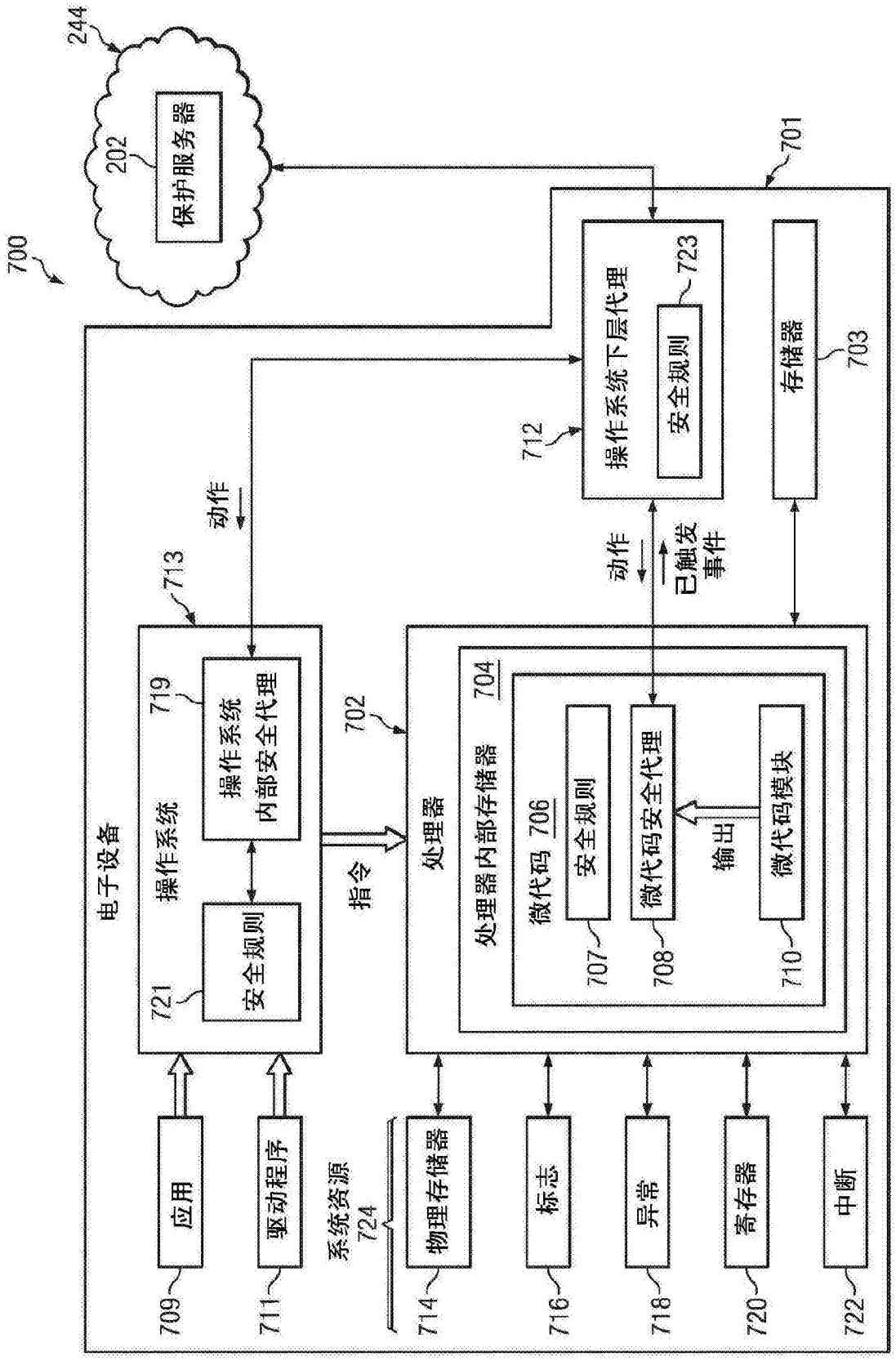

700
244
保护服务器
202
701
电子设备
操作系统
719
操作系统
内部安全代理
安全规则
721
713
动作
712
操作系统下层代理
安全规则
723
存储器
703
动作
已触发
事件
指令
702
处理器
处理器内部存储器
704
微代码 706
安全规则
707
微代码安全代理
708
输出
微代码模块
710
应用
709
驱动程序
711
系统资源
724
物理存储器
714
标志
716
异常
718
寄存器
720
中断
722

图7

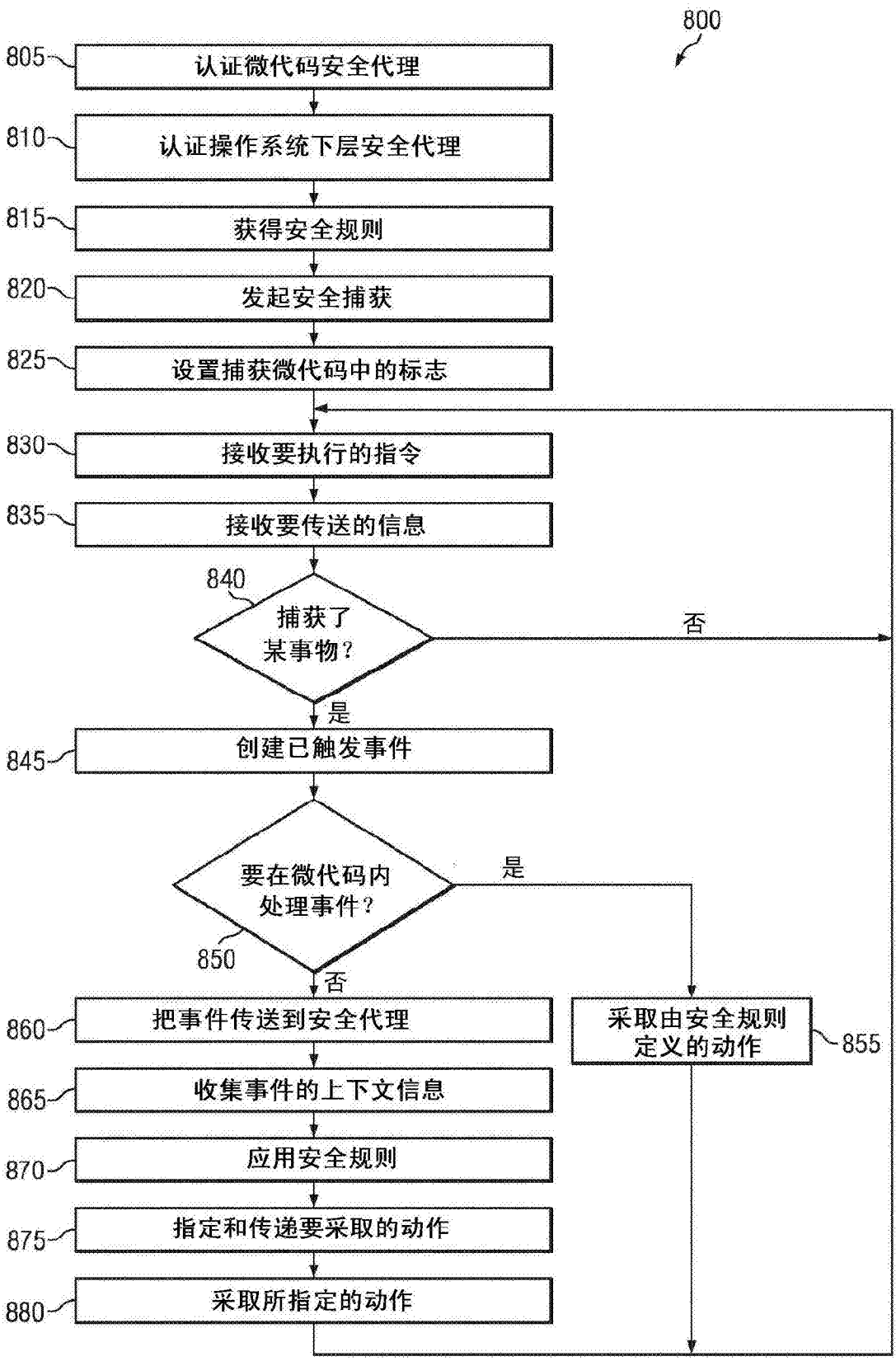

800
805
认证微代码安全代理
810
认证操作系统下层安全代理
815
获得安全规则
820
发起安全捕获
825
设置捕获微代码中的标志
830
接收要执行的指令
835
接收要传送的信息
840
捕获了某事物？
否
是
845
创建已触发事件
要在微代码内处理事件？
是
850
否
860
把事件传送到安全代理
865
收集事件的上下文信息
870
应用安全规则
875
指定和传递要采取的动作
880
采取所指定的动作
采取由安全规则定义的动作
855

图8

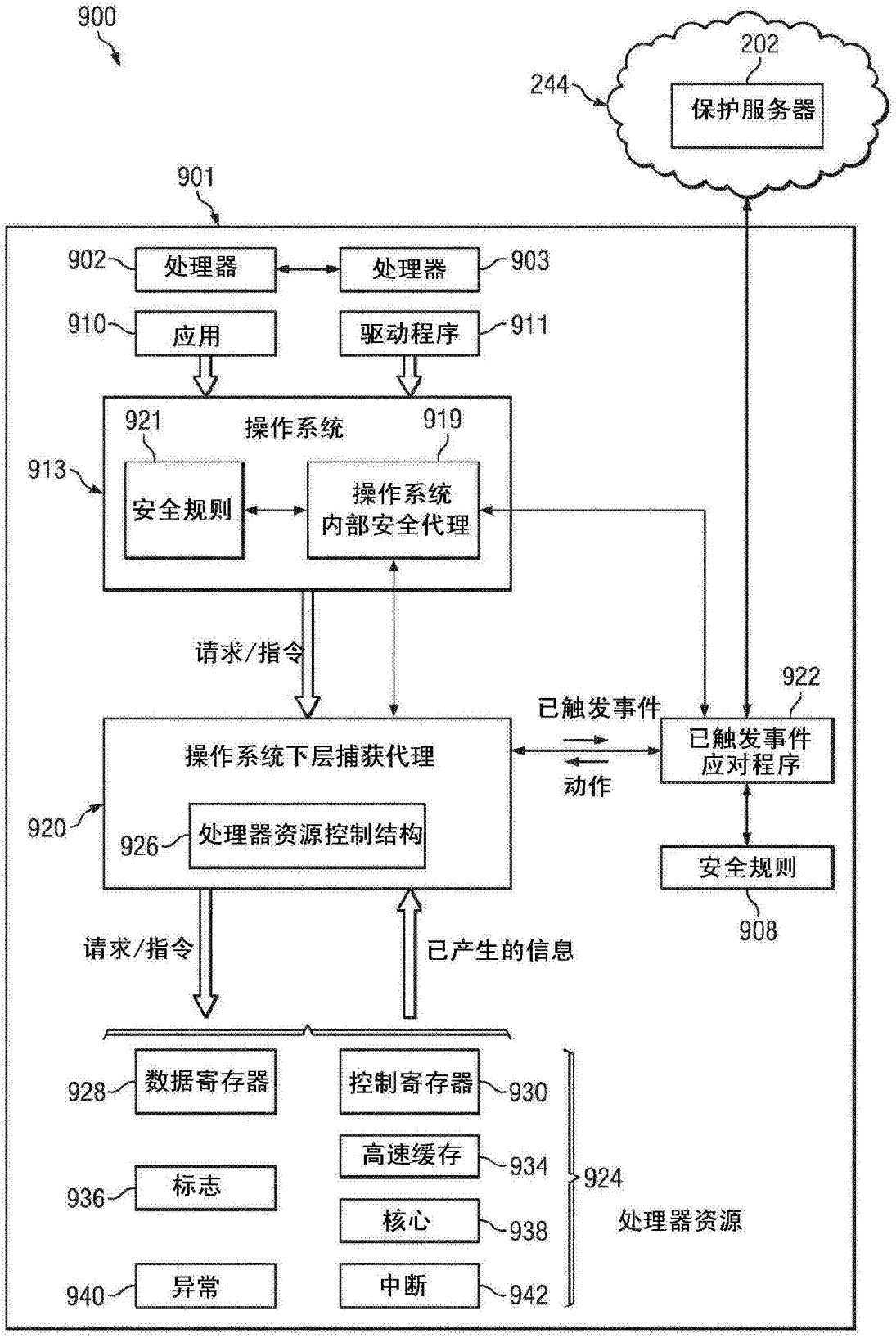

900
202
244
保护服务器
901
902
处理器
处理器
903
910
应用
驱动程序
911
921
操作系统
919
913
安全规则
操作系统
内部安全代理
请求/指令
922
已触发事件
操作系统下层捕获代理
已触发事件
应对程序
动作
920
926
处理器资源控制结构
安全规则
908
请求/指令
已产生的信息
928
数据寄存器
控制寄存器
930
高速缓存
934
924
936
标志
核心
938
处理器资源
940
异常
中断
942

图9


处理器资源控制结构
1000
1002
1004
1006
1008
1010
1012
触发器标志
资源
类型
触发器类型
什么时候触发
触发的执行阶段
开 | 关
数据寄存器
寄存器
同步 / 异步
什么时候写入
在指令提取之前
开 | 关
控制寄存器
寄存器
同步 / 异步
什么时候写入值XXX
在指令提取之后
开 | 关
跳转
指令
同步 / 异步
什么时候执行JMP
在执行之后
开 | 关
转译后备缓存
高速缓存
同步 / 异步
什么时候使得高速缓存失效
在存取存储器之后
开 | 关
逻辑核心N
处理器
同步 / 异步
核心什么时候空闲
不适用
开 | 关
时间戳记数器
计数器
同步 / 异步
什么时候读取
在写回之后
开 | 关
ZZZ存储器地址
物理存储器地址
同步 / 异步
什么时候读取
在执行之后
关 | 关
YYY存储器地址
虚拟存储器地址
同步 / 异步
什么时候执行
在执行之后
关 | 关
中断-睡眠
中断
异步
在发送之前
不适用
. . .
实体
1014

图10

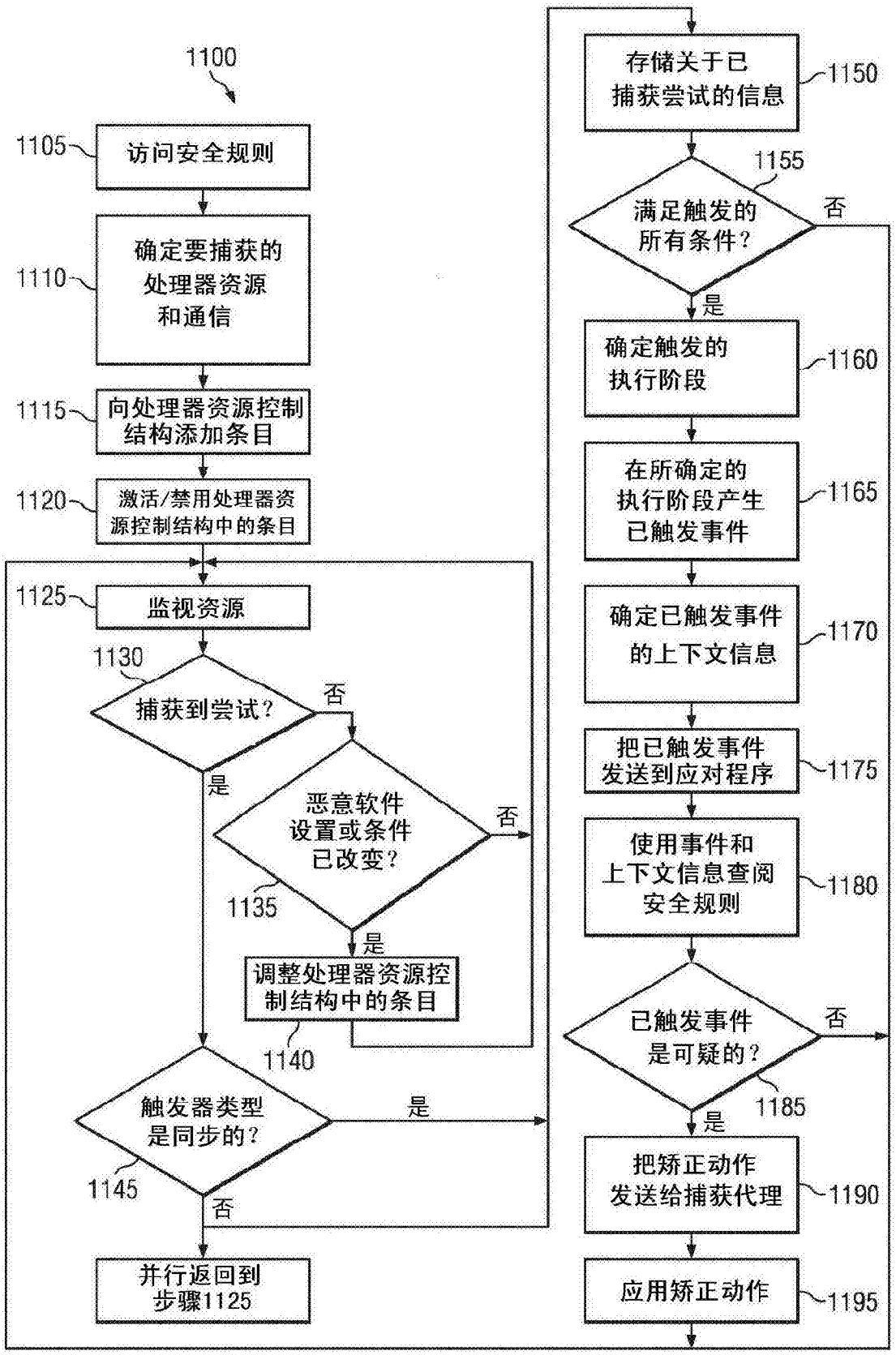

1100
1105
访问安全规则
1110
确定要捕获的处理器资源和通信
1115
向处理器资源控制结构添加条目
1120
激活/禁用处理器资源控制结构中的条目
1125
监视资源
1130
捕获到尝试？
否
是
恶意软件设置或条件已改变？
否
1135
是
调整处理器资源控制结构中的条目
1140
触发器类型是同步的？
是
1145
否
并行返回到步骤1125
存储关于已捕获尝试的信息
1150
1155
满足触发的所有条件？
否
是
确定触发的执行阶段
1160
在所确定的执行阶段产生已触发事件
1165
确定已触发事件的上下文信息
1170
把已触发事件发送到应对程序
1175
使用事件和上下文信息查阅安全规则
1180
已触发事件是可疑的？
否
1185
是
把矫正动作发送给捕获代理
1190
应用矫正动作
1195

图11

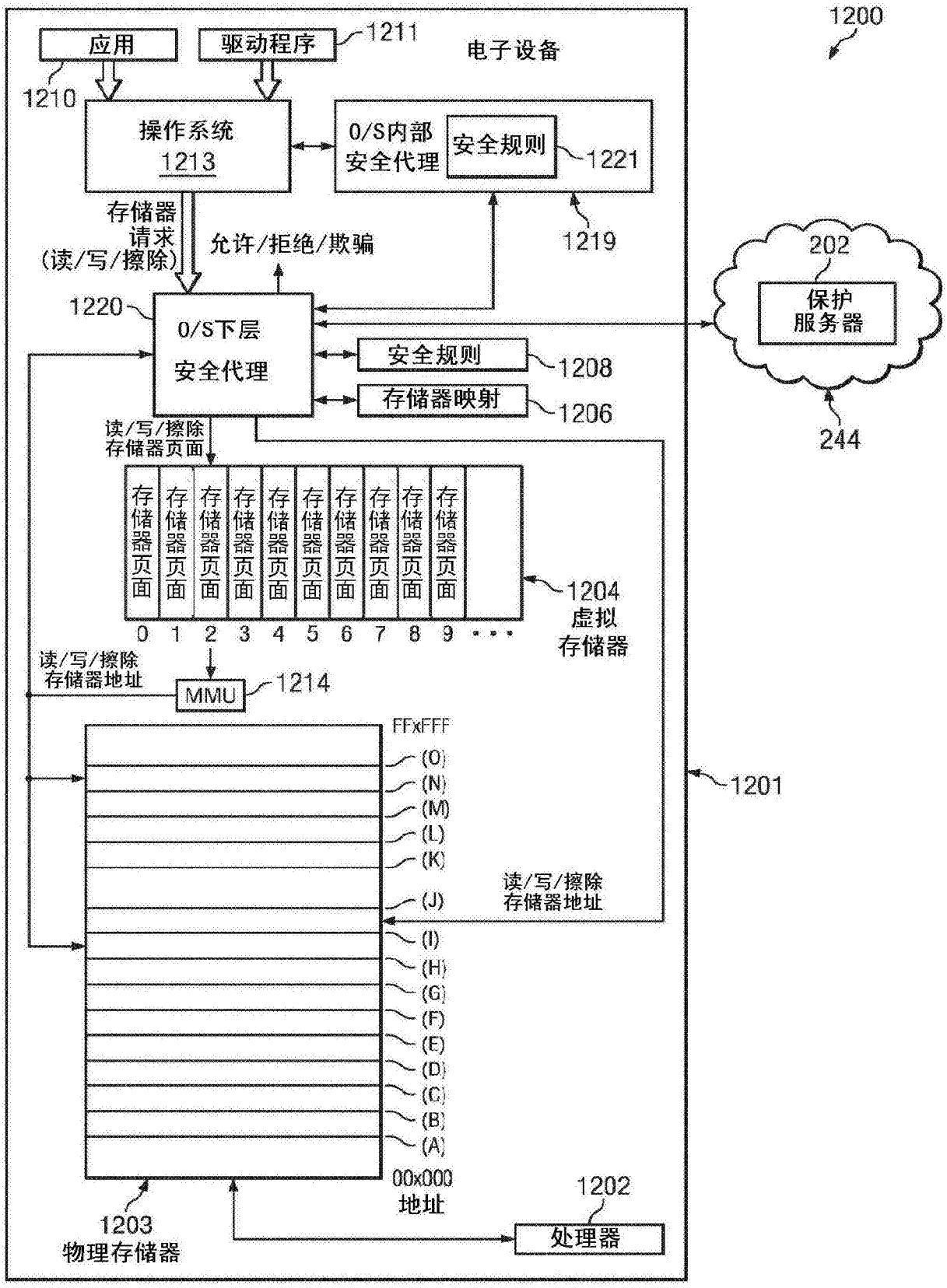

1200
应用
驱动程序
1211
电子设备
1210
操作系统
1213
O/S内部
安全代理
安全规则
1221
1219
存储器
请求
(读/写/擦除)
允许/拒绝/欺骗
202
保护
服务器
244
1220
O/S下层
安全代理
安全规则
1208
存储器映射
1206
读/写/擦除
存储器页面
存储器页面
0 1 2 3 4 5 6 7 8 9 • • •
1204
虚拟
存储器
读/写/擦除
存储器地址
MMU
1214
FFxFFF
(O)
(N)
(M)
(L)
(K)
(J)
(I)
(H)
(G)
(F)
(E)
(D)
(C)
(B)
(A)
00x000
地址
读/写/擦除
存储器地址
1201
1203
物理存储器
1202
处理器

图12

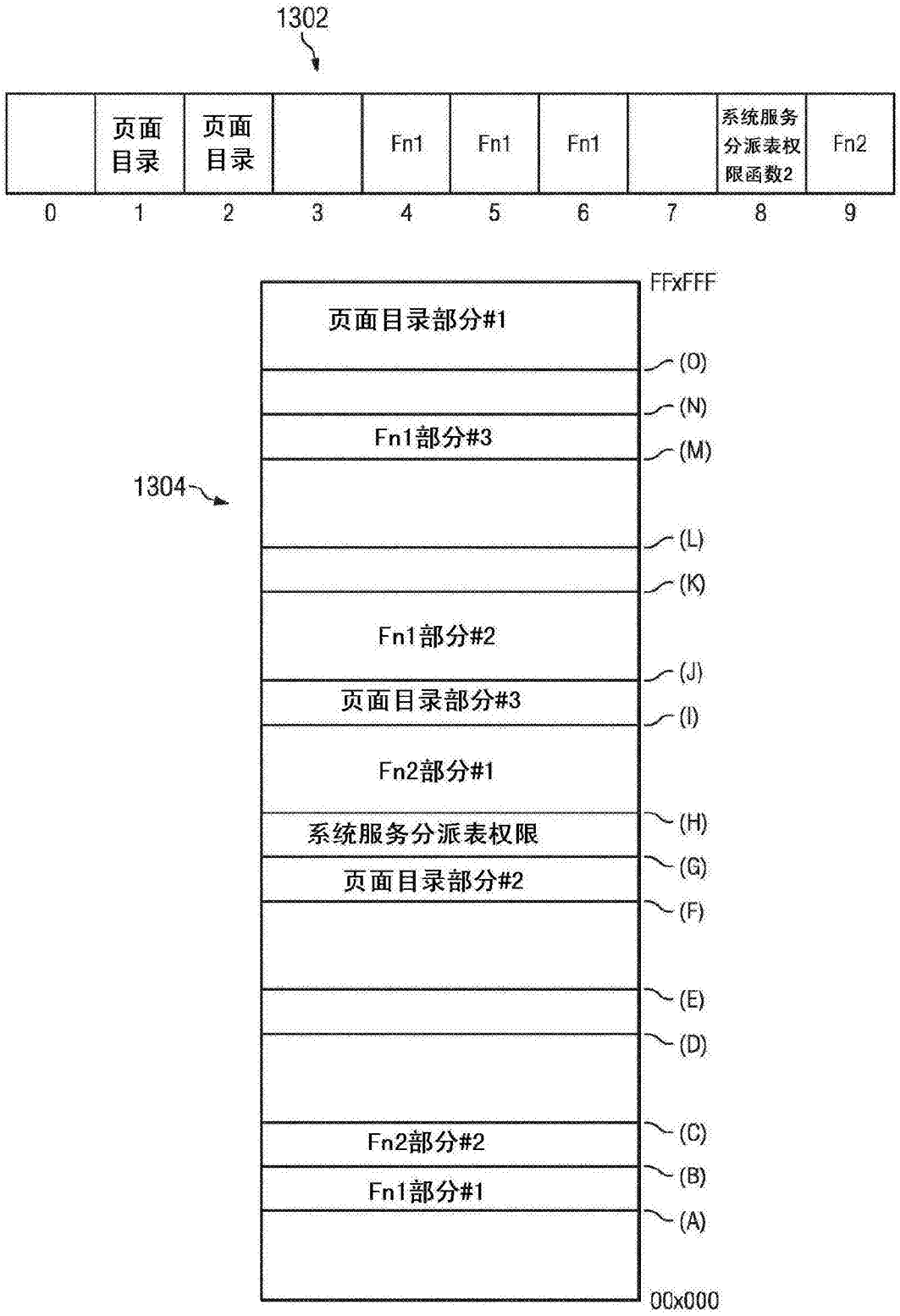

1302
页面目录
页面目录
Fn1
Fn1
Fn1
系统服务分派表权限函数2
Fn2
0
1
2
3
4
5
6
7
8
9
1304
FFxFFF
页面目录部分#1
(O)
(N)
Fn1部分#3
(M)
(L)
(K)
Fn1部分#2
(J)
页面目录部分#3
(I)
Fn2部分#1
(H)
系统服务分派表权限
(G)
页面目录部分#2
(F)
(E)
(D)
(C)
Fn2部分#2
(B)
Fn1部分#1
(A)
00x000

图13

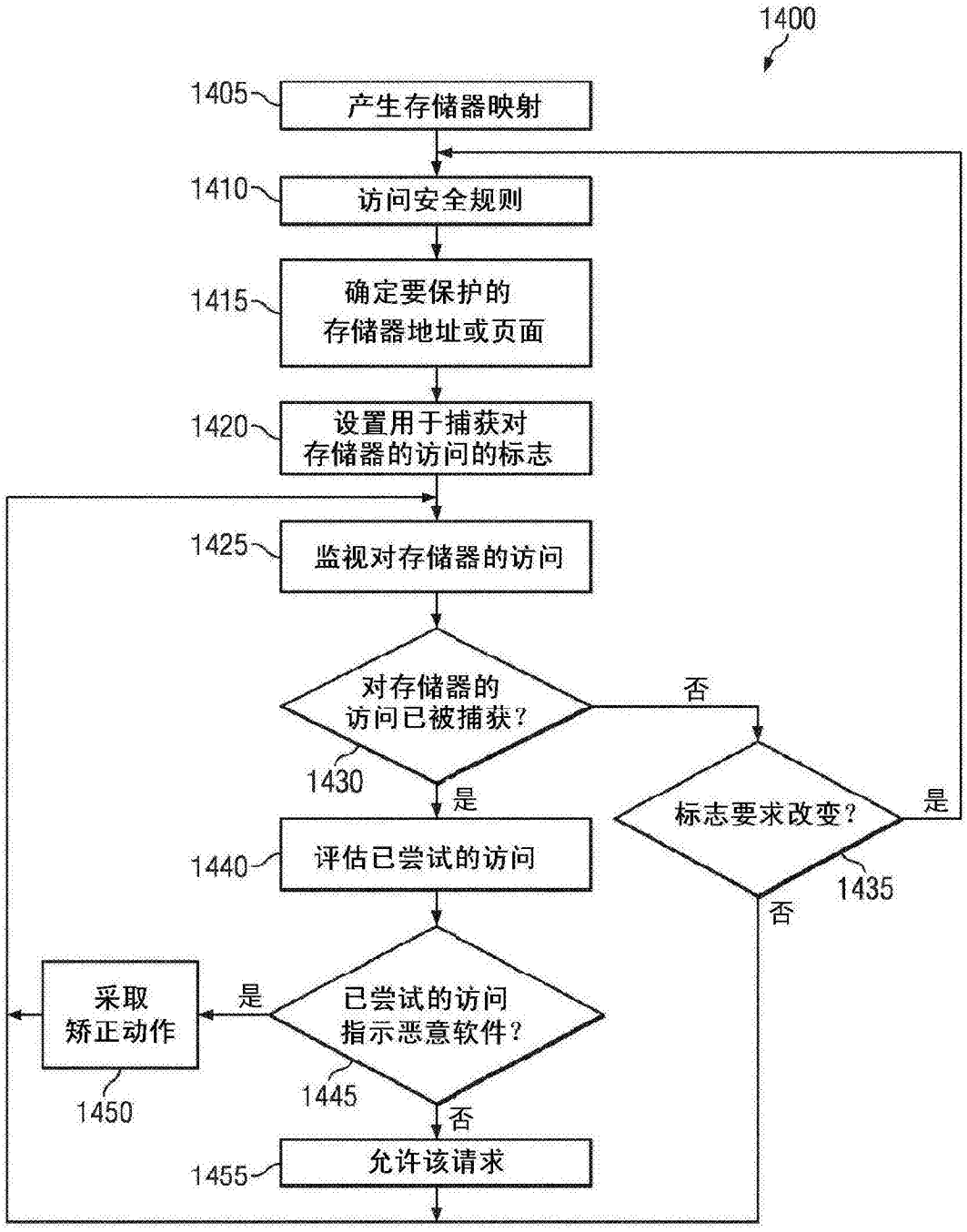

1400
1405
产生存储器映射
1410
访问安全规则
1415
确定要保护的
存储器地址或页面
1420
设置用于捕获对
存储器的访问的标志
1425
监视对存储器的访问
对存储器的
访问已被捕获？
1430
否
是
标志要求改变？
1435
是
否
1440
评估已尝试的访问
已尝试的访问
指示恶意软件？
1445
是
采取
矫正动作
1450
否
1455
允许该请求

图14

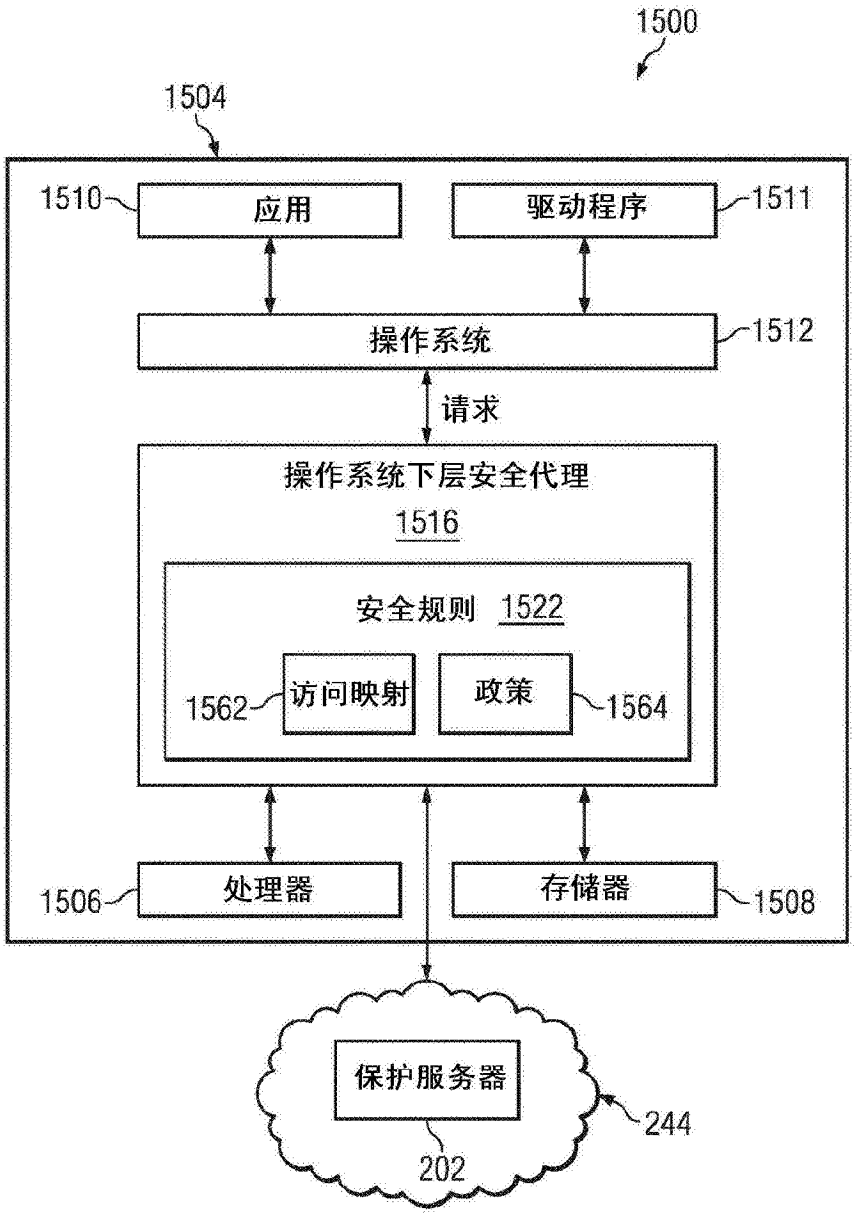

1500
1504
1510
应用
驱动程序
1511
操作系统
1512
请求
操作系统下层安全代理
1516
安全规则 1522
1562
访问映射
政策
1564
1506
处理器
存储器
1508
保护服务器
244
202

图15

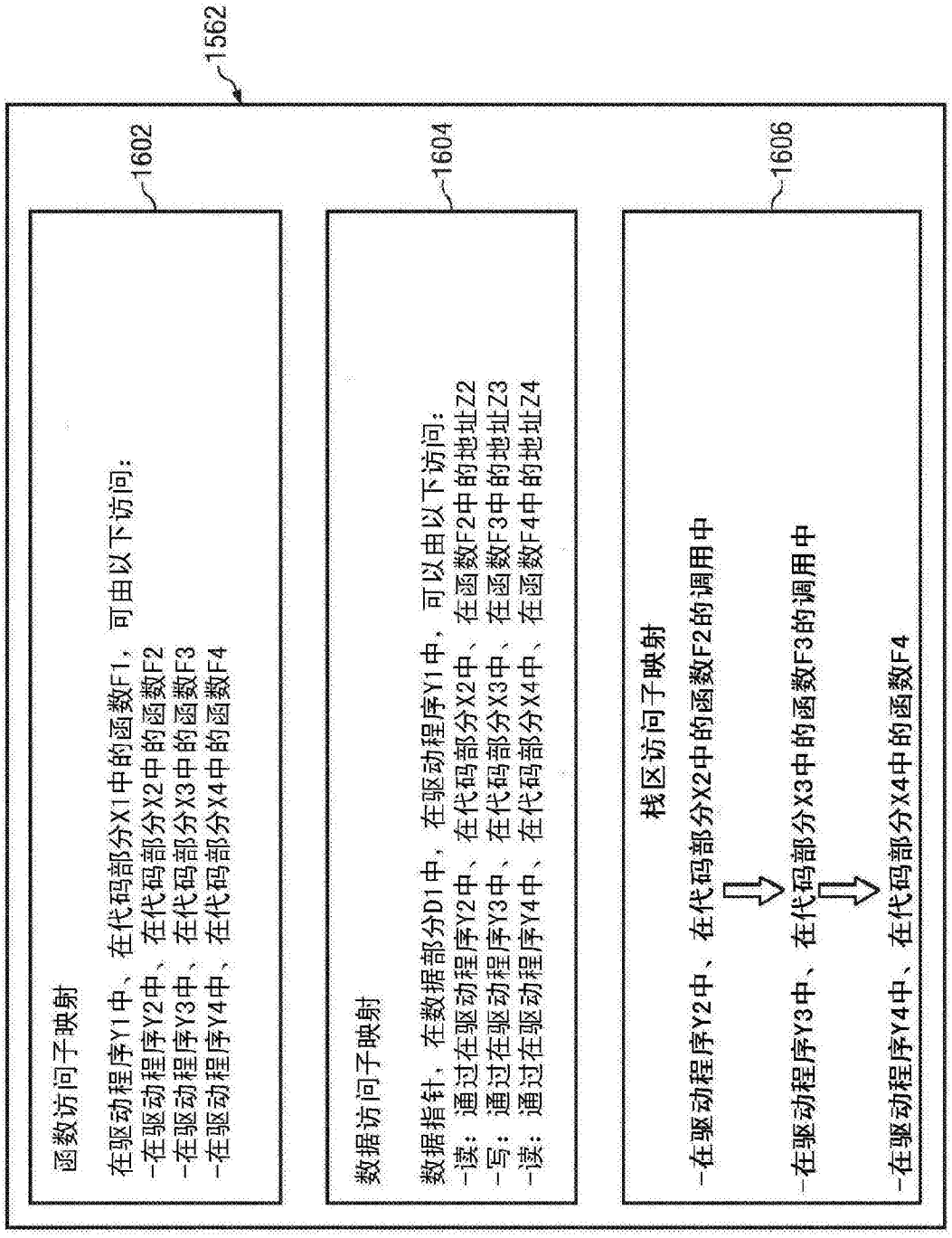

1562
1602
函数访问子映射
在驱动程序Y1中、在代码部分X1中的函数F1，可由以下访问：
-在驱动程序Y2中、在代码部分X2中的函数F2
-在驱动程序Y3中、在代码部分X3中的函数F3
-在驱动程序Y4中、在代码部分X4中的函数F4
1604
数据访问子映射
数据指针，在数据部分D1中，在驱动程序Y1中，可以由以下访问：
-读：通过在驱动程序Y2中、在代码部分X2中、在函数F2中的地址Z2
-写：通过在驱动程序Y3中、在代码部分X3中、在函数F3中的地址Z3
-读：通过在驱动程序Y4中、在代码部分X4中、在函数F4中的地址Z4
1606
栈区访问子映射
-在驱动程序Y2中、在代码部分X2中的函数F2的调用中
-在驱动程序Y3中、在代码部分X3中的函数F3的调用中
-在驱动程序Y4中、在代码部分X4中的函数F4

图16

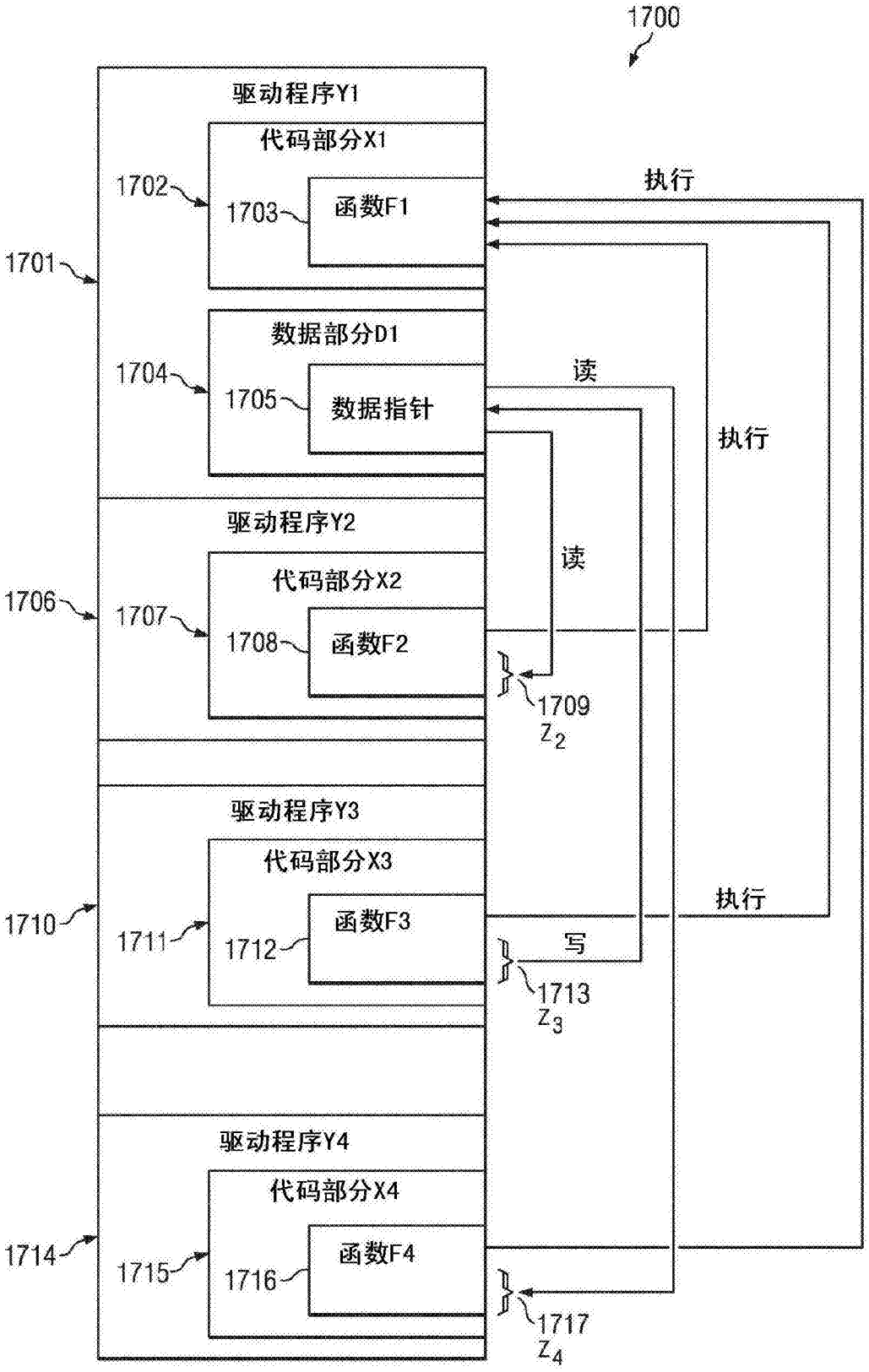

1700
驱动程序Y1
代码部分X1
1702
1703
函数F1
1701
执行
数据部分D1
1704
1705
数据指针
读
执行
驱动程序Y2
读
代码部分X2
1706
1707
1708
函数F2
1709
Z_2
驱动程序Y3
代码部分X3
1710
1711
1712
函数F3
执行
写
1713
Z_3
驱动程序Y4
代码部分X4
1714
1715
1716
函数F4
1717
Z_4

图17

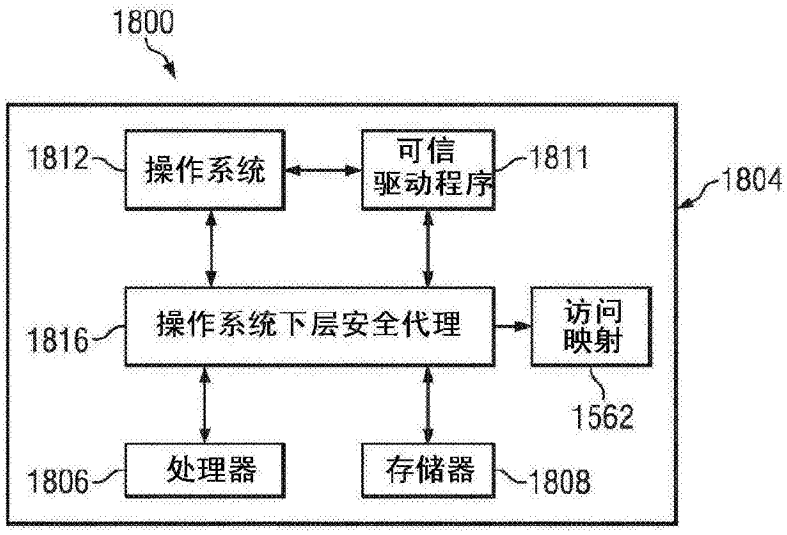

1800
1812
操作系统
可信
驱动程序
1811
1804
1816
操作系统下层安全代理
访问
映射
1562
1806
处理器
存储器
1808

图18

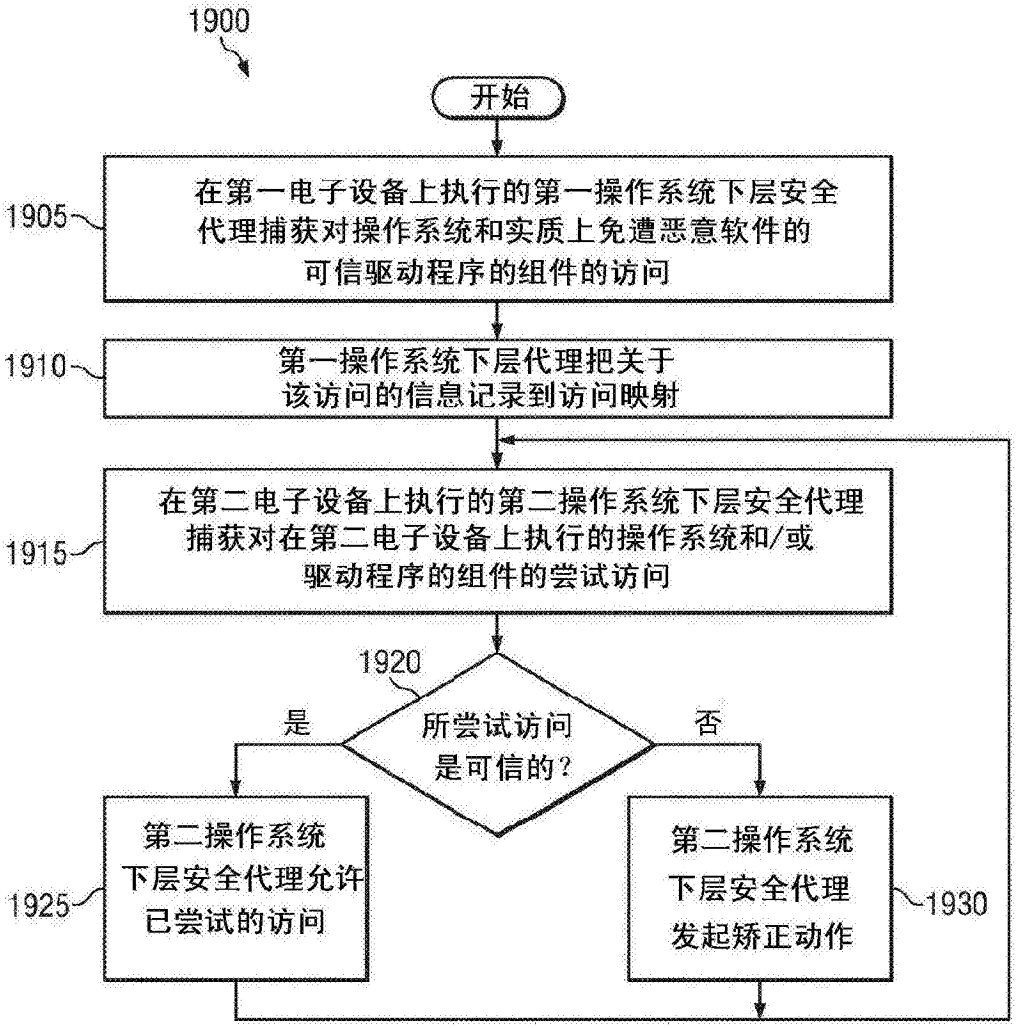

1900
开始
1905
在第一电子设备上执行的第一操作系统下层安全代理捕获对操作系统和实质上免遭恶意软件的可信驱动程序的组件的访问
1910
第一操作系统下层代理把关于该访问的信息记录到访问映射
1915
在第二电子设备上执行的第二操作系统下层安全代理捕获对在第二电子设备上执行的操作系统和/或驱动程序的组件的尝试访问
1920
所尝试访问是可信的?
是
否
1925
第二操作系统下层安全代理允许已尝试的访问
第二操作系统下层安全代理发起矫正动作
1930

图19

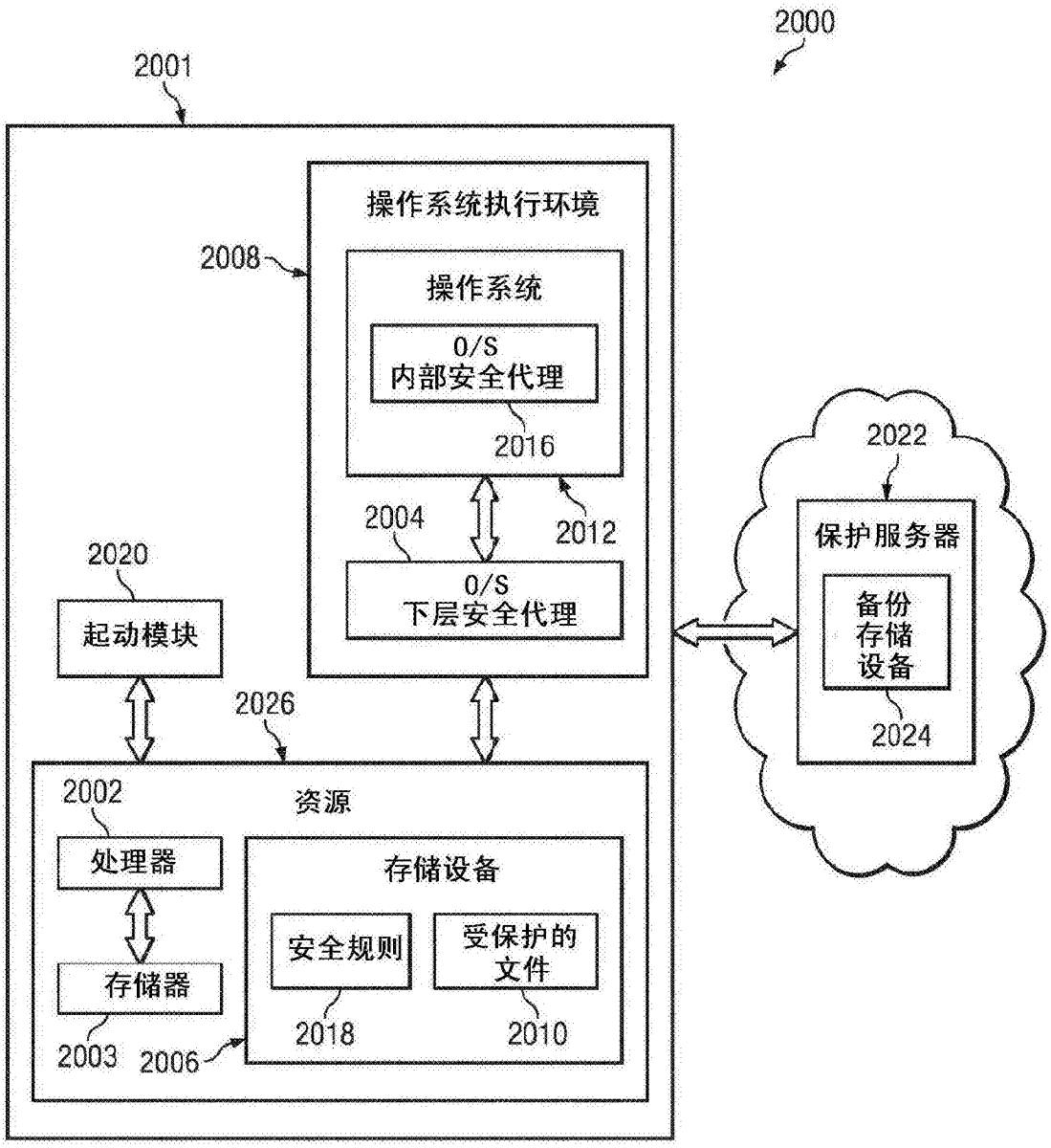

2000
2001
操作系统执行环境
2008
操作系统
0/S
内部安全代理
2016
2004
2012
0/S
下层安全代理
2020
起动模块
2026
2022
保护服务器
备份
存储
设备
2024
2002
资源
处理器
存储设备
安全规则
受保护的
文件
存储器
2003
2006
2018
2010

图20

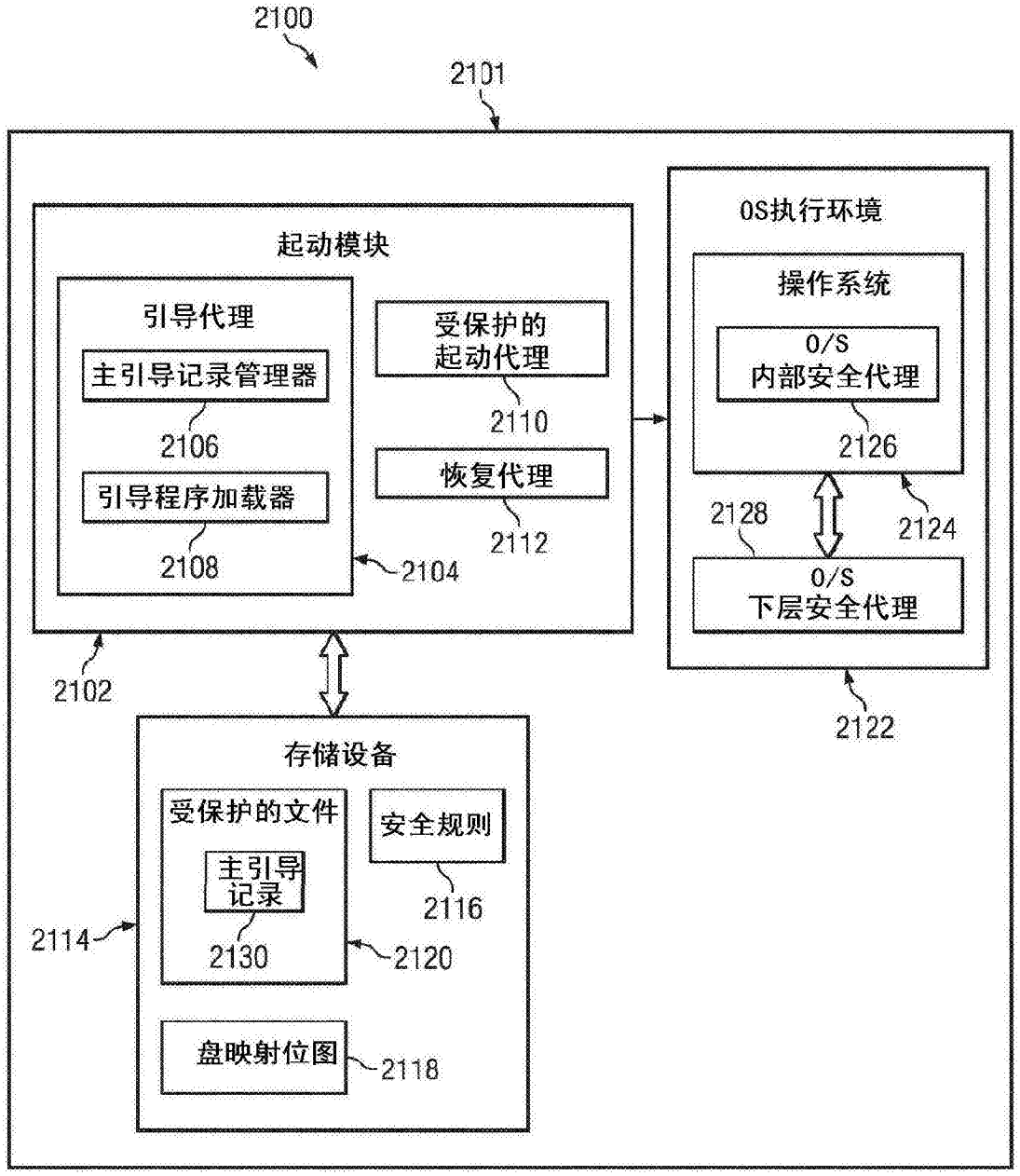

2100
2101
起动模块
引导代理
主引导记录管理器
2106
引导程序加载器
2108
受保护的
起动代理
2110
恢复代理
2112
2104
2102
OS执行环境
操作系统
O/S
内部安全代理
2126
2128
2124
O/S
下层安全代理
2122
存储设备
受保护的文件
主引导
记录
2130
安全规则
2116
2114
2120
盘映射位图
2118

图21

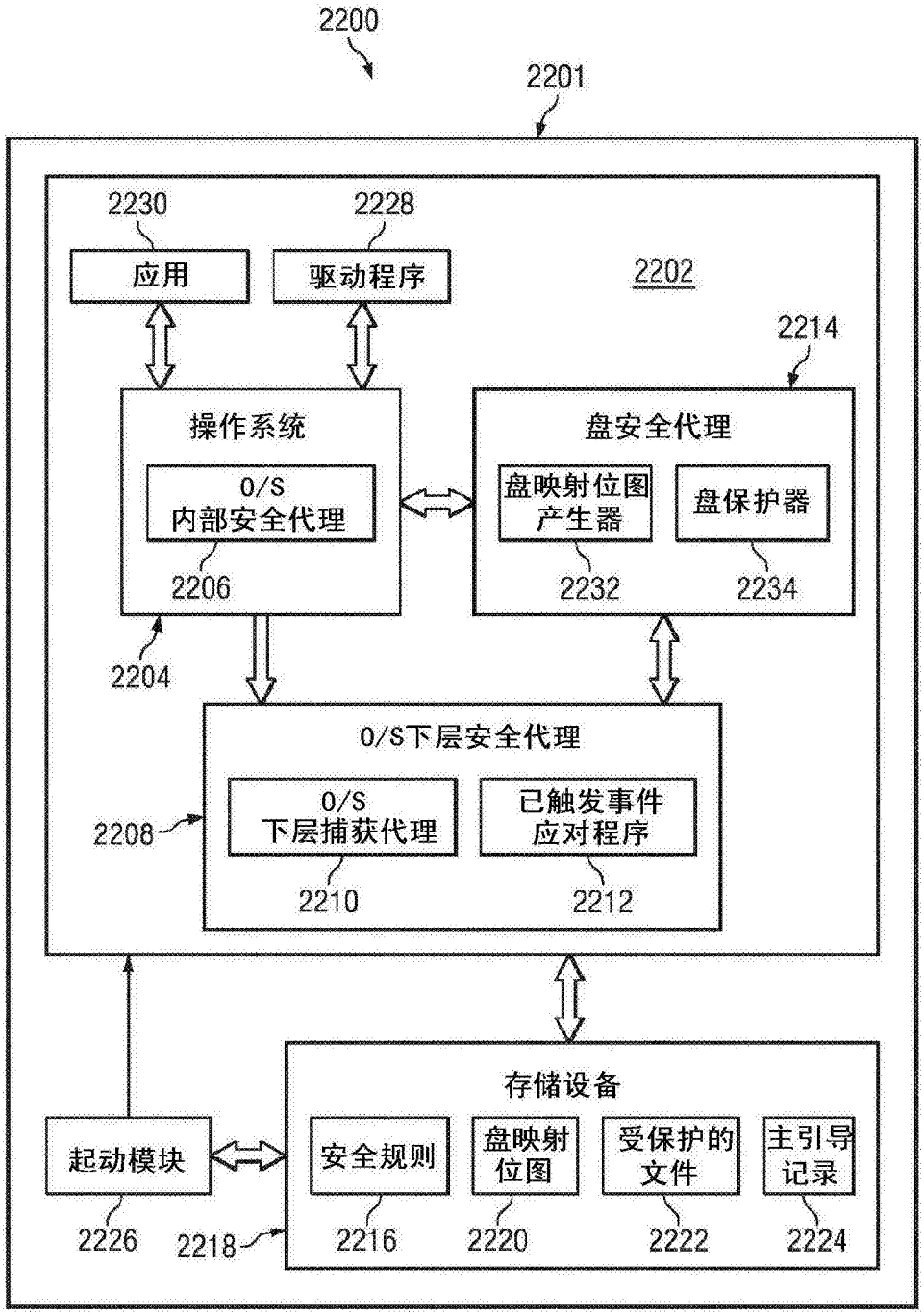
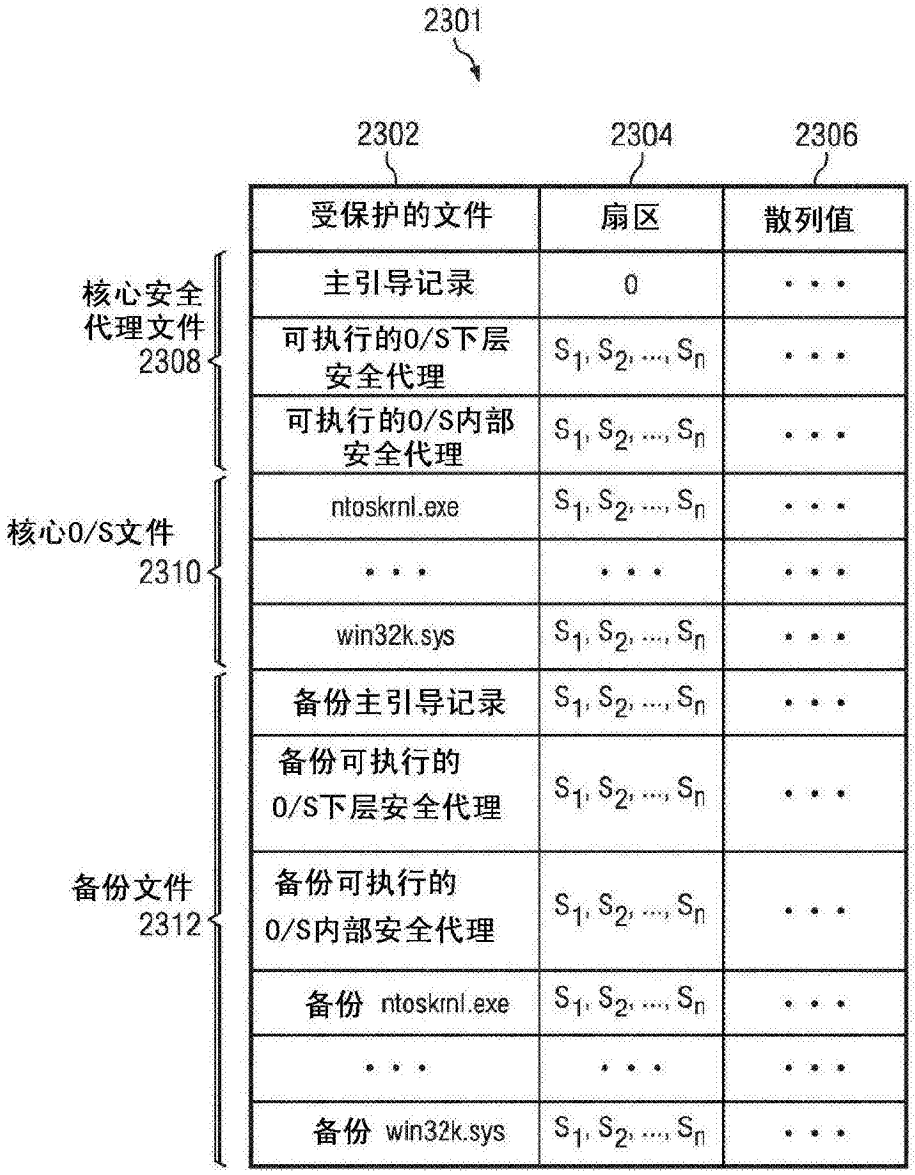

2200
2201
2230
2228
应用
驱动程序
2202
2214
操作系统
O/S
内部安全代理
2206
盘安全代理
盘映射位图
产生器
盘保护器
2232
2234
2204
O/S下层安全代理
2208
O/S
下层捕获代理
已触发事件
应对程序
2210
2212
存储设备
起动模块
安全规则
盘映射
位图
受保护的
文件
主引导
记录
2226
2218
2216
2220
2222
2224

图22



受保护的文件	扇区	散列值
主引导记录	0	• • •
可执行的O/S下层安全代理	$S_1, S_2, ..., S_n$	• • •
可执行的O/S内部安全代理	$S_1, S_2, ..., S_n$	• • •
ntoskrnl.exe	$S_1, S_2, ..., S_n$	• • •
• • •	• • •	• • •
win32k.sys	$S_1, S_2, ..., S_n$	• • •
备份主引导记录	$S_1, S_2, ..., S_n$	• • •
备份可执行的O/S下层安全代理	$S_1, S_2, ..., S_n$	• • •
备份可执行的O/S内部安全代理	$S_1, S_2, ..., S_n$	• • •
备份 ntoskrnl.exe	$S_1, S_2, ..., S_n$	• • •
• • •	• • •	• • •
备份 win32k.sys	$S_1, S_2, ..., S_n$	• • •

图23

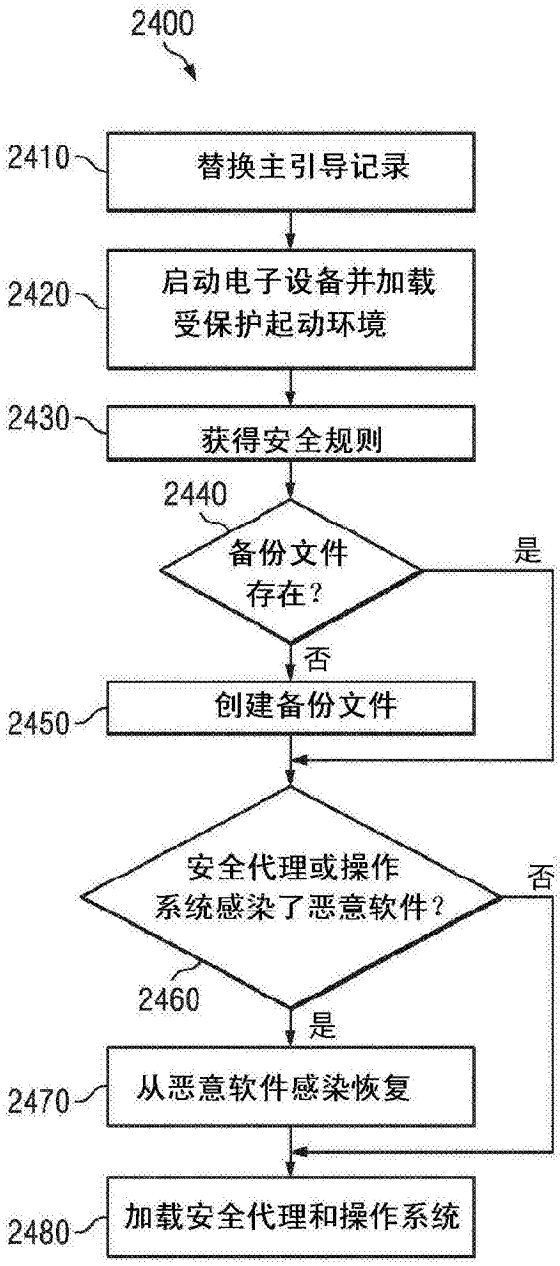

2400
2410
替换主引导记录
2420
启动电子设备并加载受保护起动环境
2430
获得安全规则
2440
备份文件存在？
是
否
2450
创建备份文件
安全代理或操作系统感染了恶意软件？
否
2460
是
2470
从恶意软件感染恢复
2480
加载安全代理和操作系统

图24

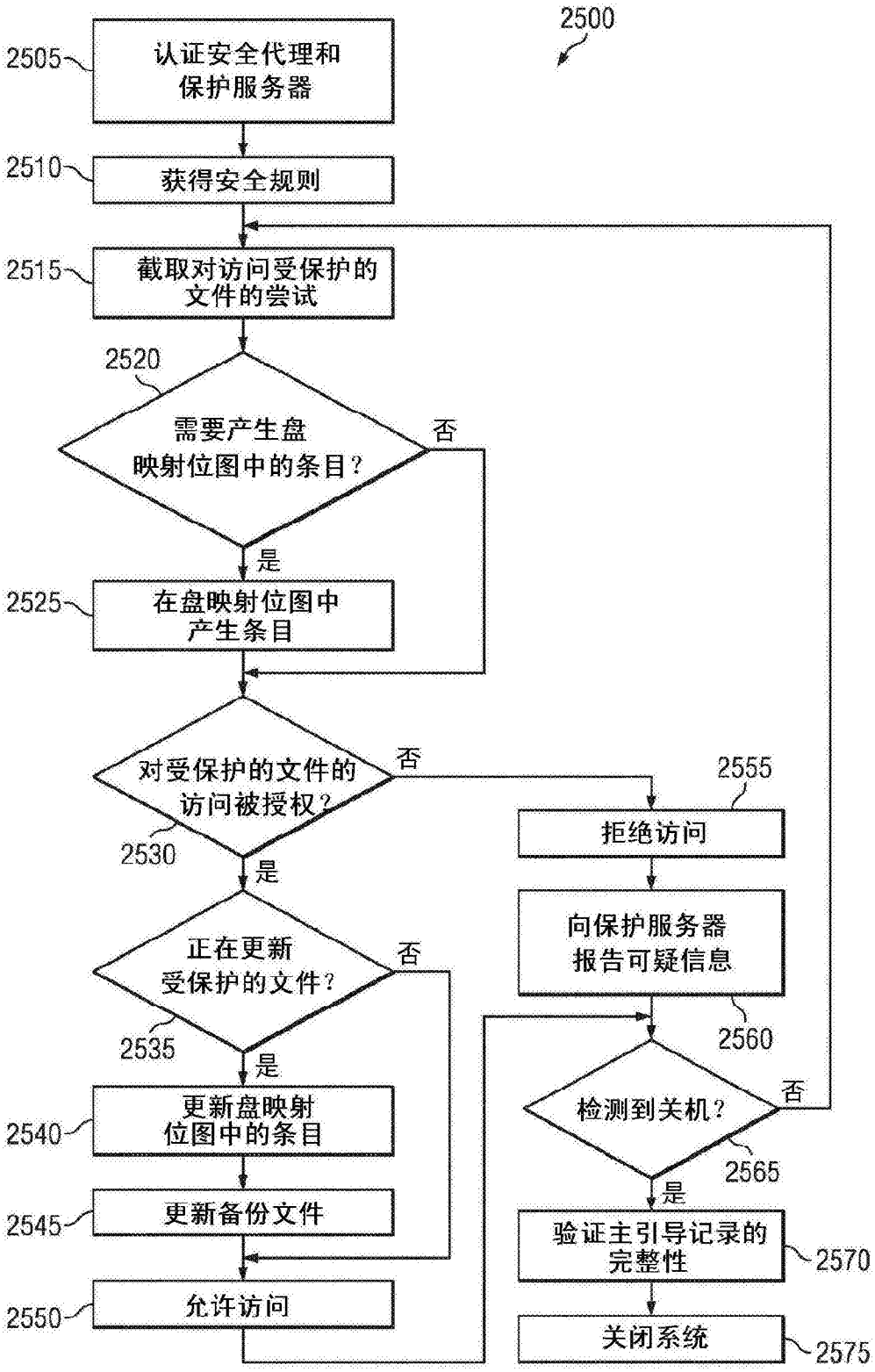

2500
2505
认证安全代理和保护服务器
2510
获得安全规则
2515
截取对访问受保护的文件的尝试
2520
需要产生盘映射位图中的条目？
否
是
2525
在盘映射位图中产生条目
对受保护的文件的访问被授权？
否
2530
是
2555
拒绝访问
向保护服务器报告可疑信息
2560
正在更新受保护的文件？
否
2535
是
更新盘映射位图中的条目
2540
更新备份文件
2545
允许访问
2550
检测到关机？
否
2565
是
验证主引导记录的完整性
2570
关闭系统
2575

图25

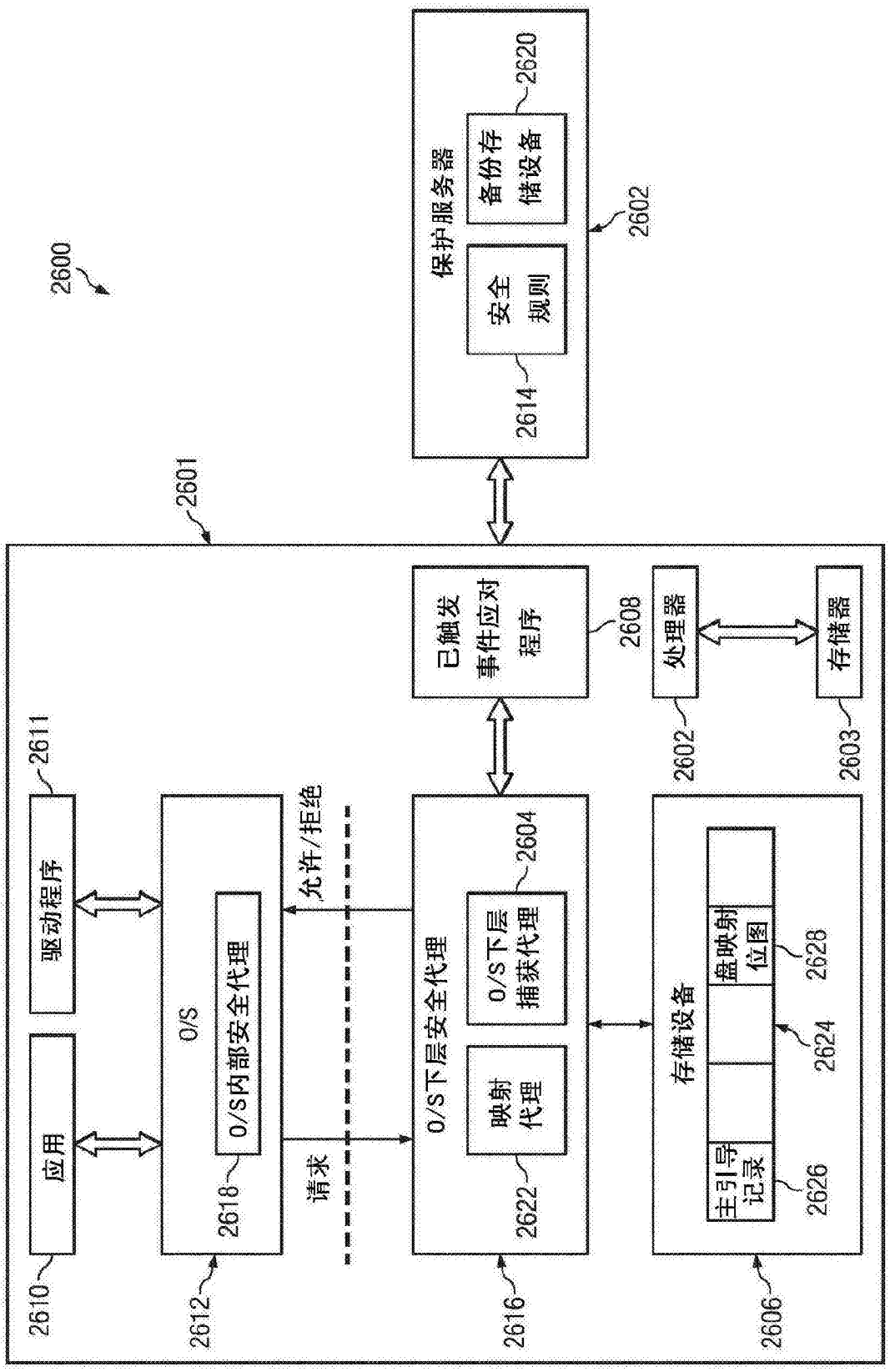

2600
保护服务器
备份存储设备
2620
安全规则
2614
2602
2601
已触发事件应对程序
2608
处理器
存储器
2602
2603
驱动程序
2611
应用
O/S
O/S内部安全代理
2618
允许/拒绝
请求
O/S下层安全代理
O/S下层捕获代理
2604
映射代理
2622
存储设备
盘映射位图
2628
2624
主引导记录
2626
2610
2612
2616
2606

图26

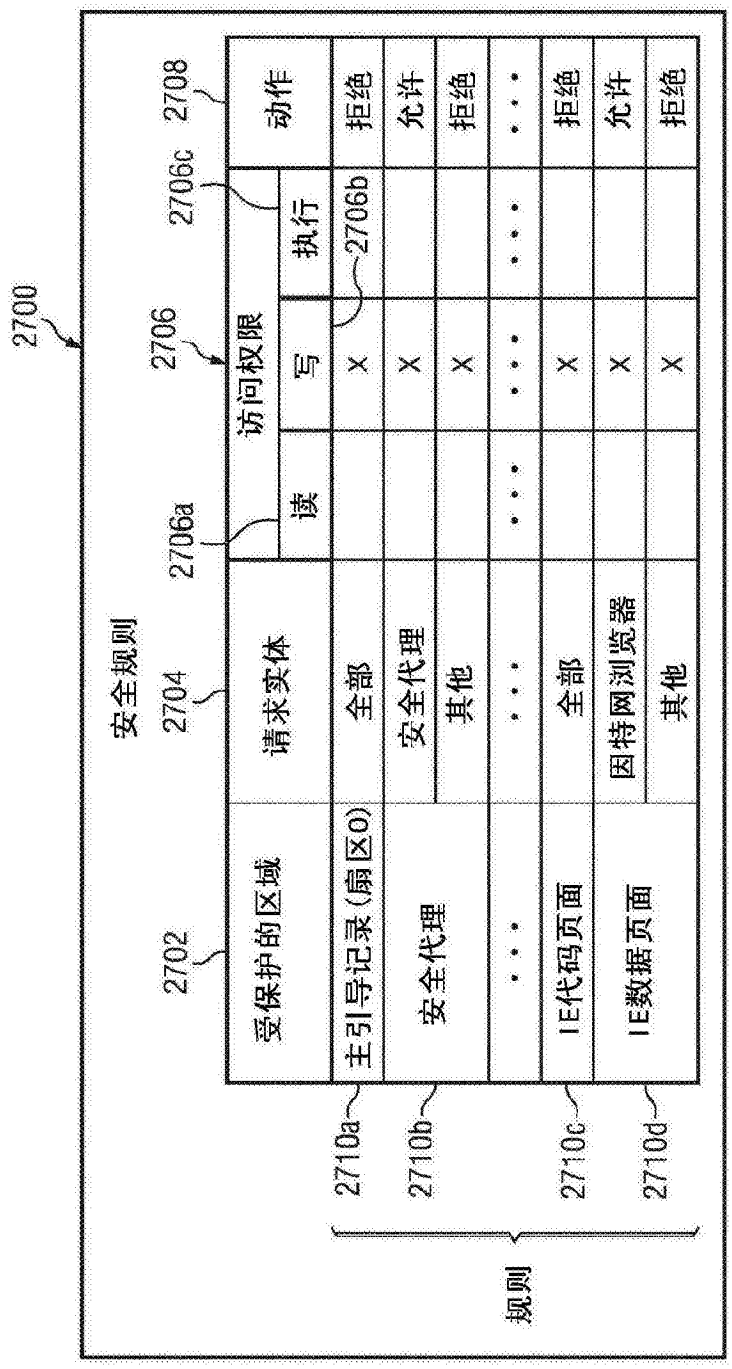

2700
安全规则
2702
受保护的区域
2704
请求实体
2706
访问权限
2706a
读
写
执行
2706b
2706c
2708
动作
主引导记录(扇区0)
全部
拒绝
安全代理
安全代理
其他
允许
拒绝
IE代码页面
全部
拒绝
IE数据页面
因特网浏览器
其他
允许
拒绝
2710a
2710b
2710c
2710d
规则

图27

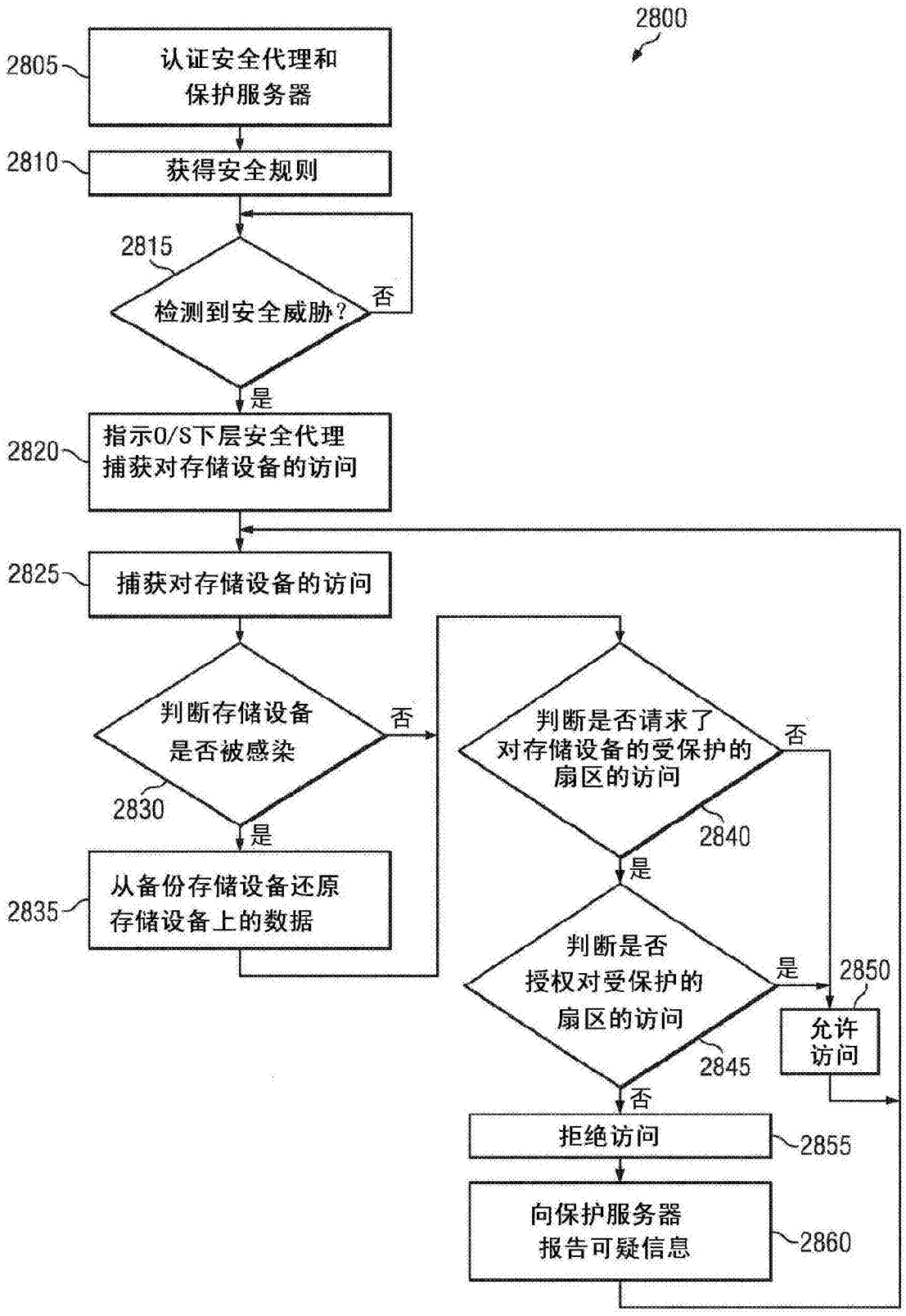

2800
2805
认证安全代理和
保护服务器
2810
获得安全规则
2815
检测到安全威胁?
否
是
2820
指示O/S下层安全代理
捕获对存储设备的访问
2825
捕获对存储设备的访问
判断存储设备
是否被感染
否
2830
是
2835
从备份存储设备还原
存储设备上的数据
判断是否请求了
对存储设备的受保护的
扇区的访问
否
2840
是
判断是否
授权对受保护的
扇区的访问
是
2850
允许
访问
2845
否
拒绝访问
2855
向保护服务器
报告可疑信息
2860

图28

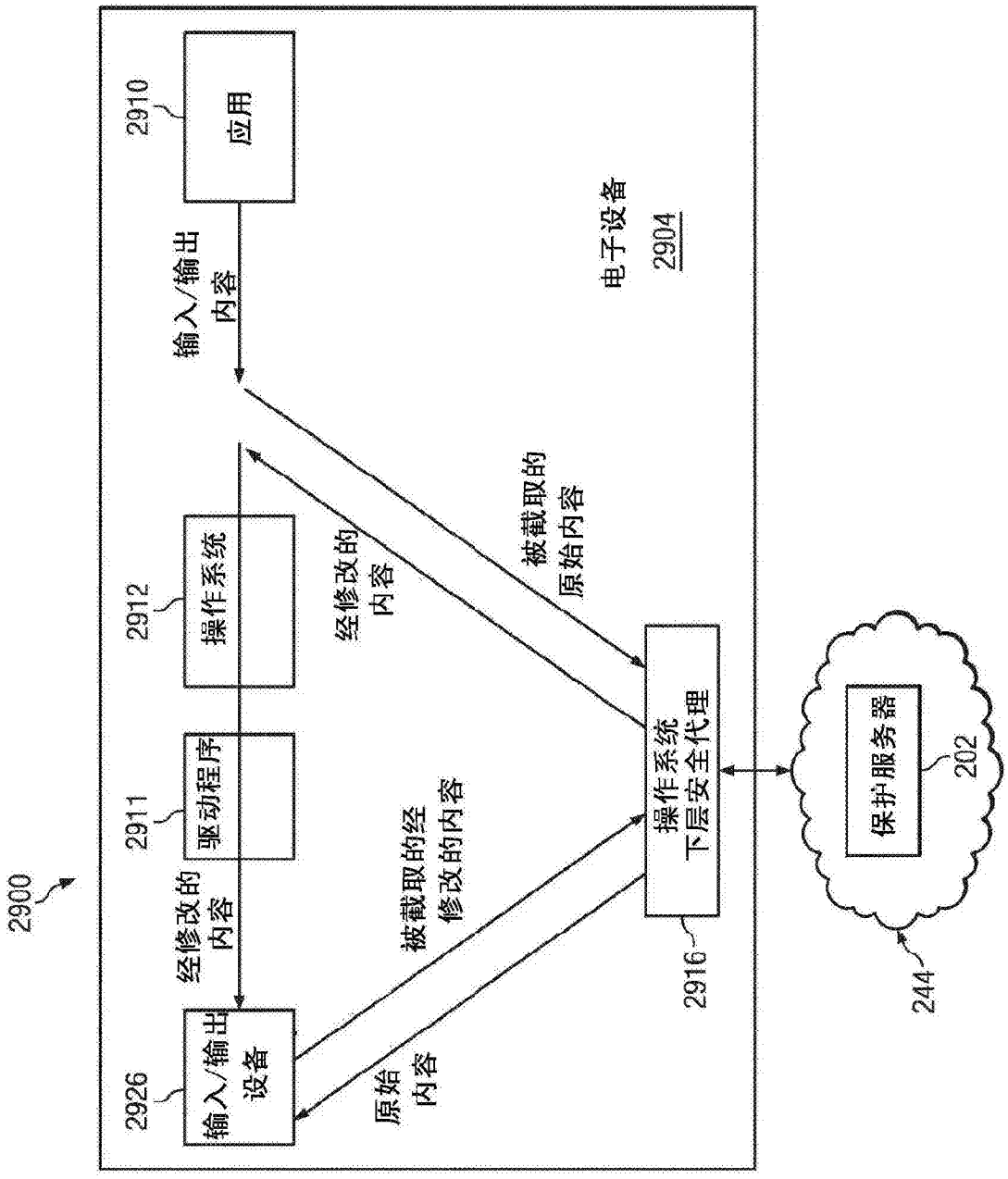

2900
电子设备
2904
2910
应用
输入/输出
内容
2912
操作系统
2911
驱动程序
经修改的
内容
2926
输入/输出
设备
原始
内容
被截取的经
修改的内容
经修改的
内容
被截取的
原始内容
操作系统
下层安全代理
2916
保护服务器
202
244

图29

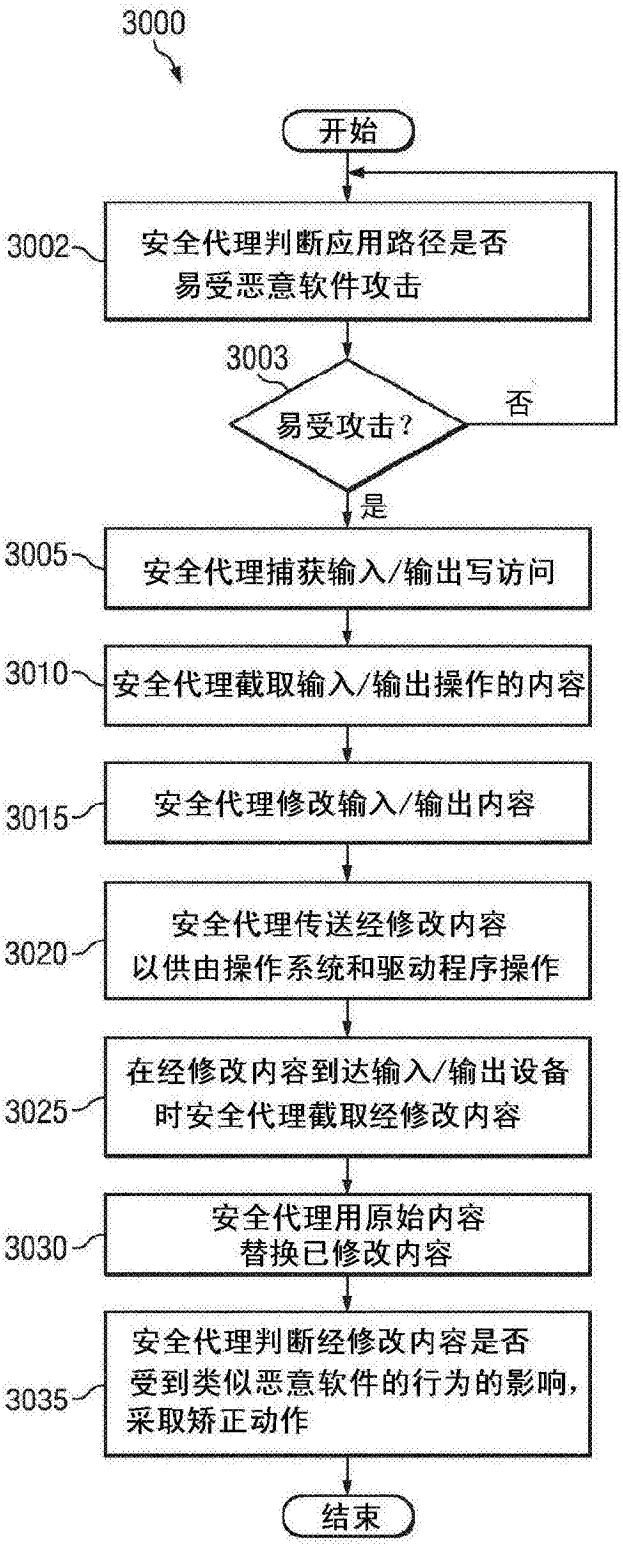

3000
开始
3002
安全代理判断应用路径是否易受恶意软件攻击
3003
易受攻击？
否
是
3005
安全代理捕获输入/输出写访问
3010
安全代理截取输入/输出操作的内容
3015
安全代理修改输入/输出内容
3020
安全代理传送经修改内容以供由操作系统和驱动程序操作
3025
在经修改内容到达输入/输出设备时安全代理截取经修改内容
3030
安全代理用原始内容替换已修改内容
3035
安全代理判断经修改内容是否受到类似恶意软件的行为的影响，采取矫正动作
结束

图30

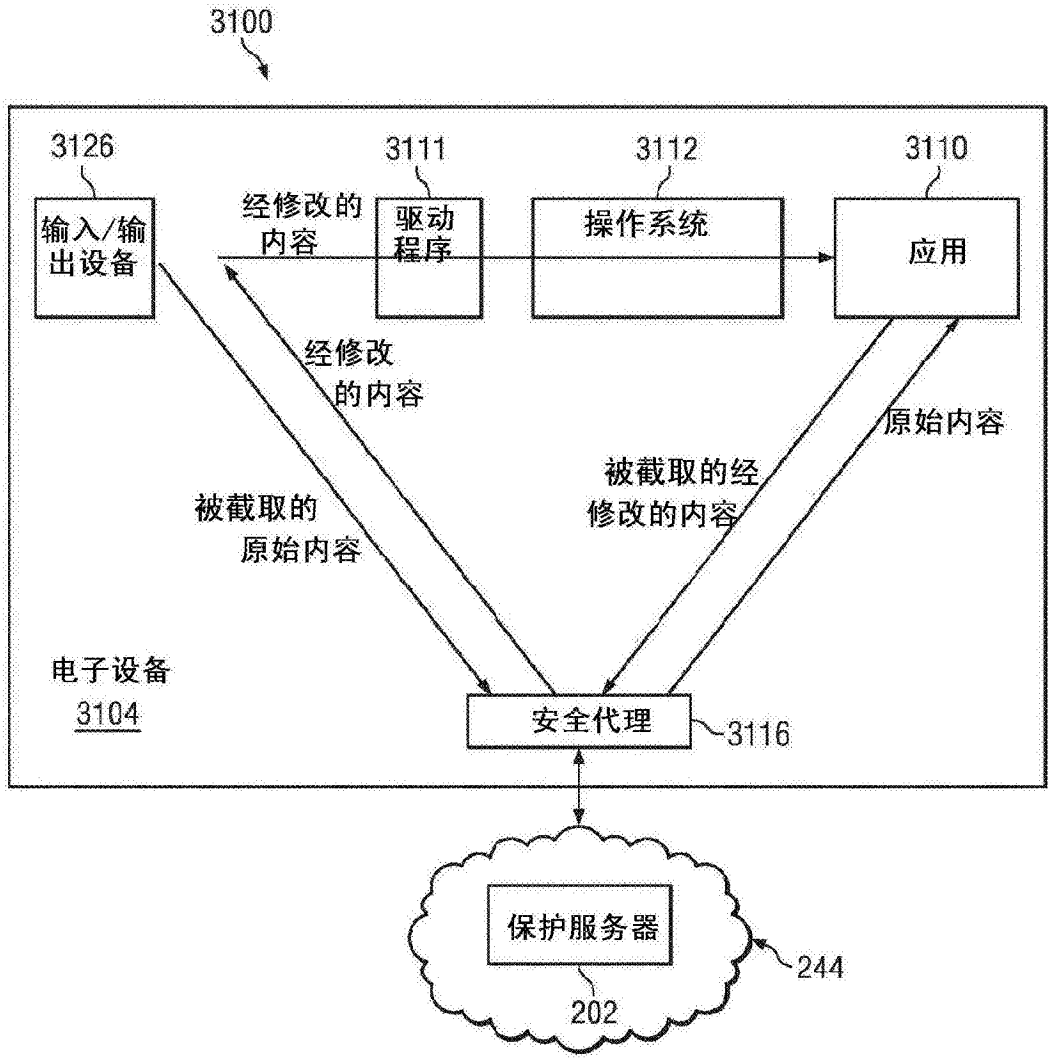

3100
3126
3111
3112
3110
输入/输出设备
经修改的内容
驱动程序
操作系统
应用
经修改的内容
原始内容
被截取的经修改的内容
被截取的原始内容
电子设备
3104
安全代理
3116
保护服务器
244
202

图31

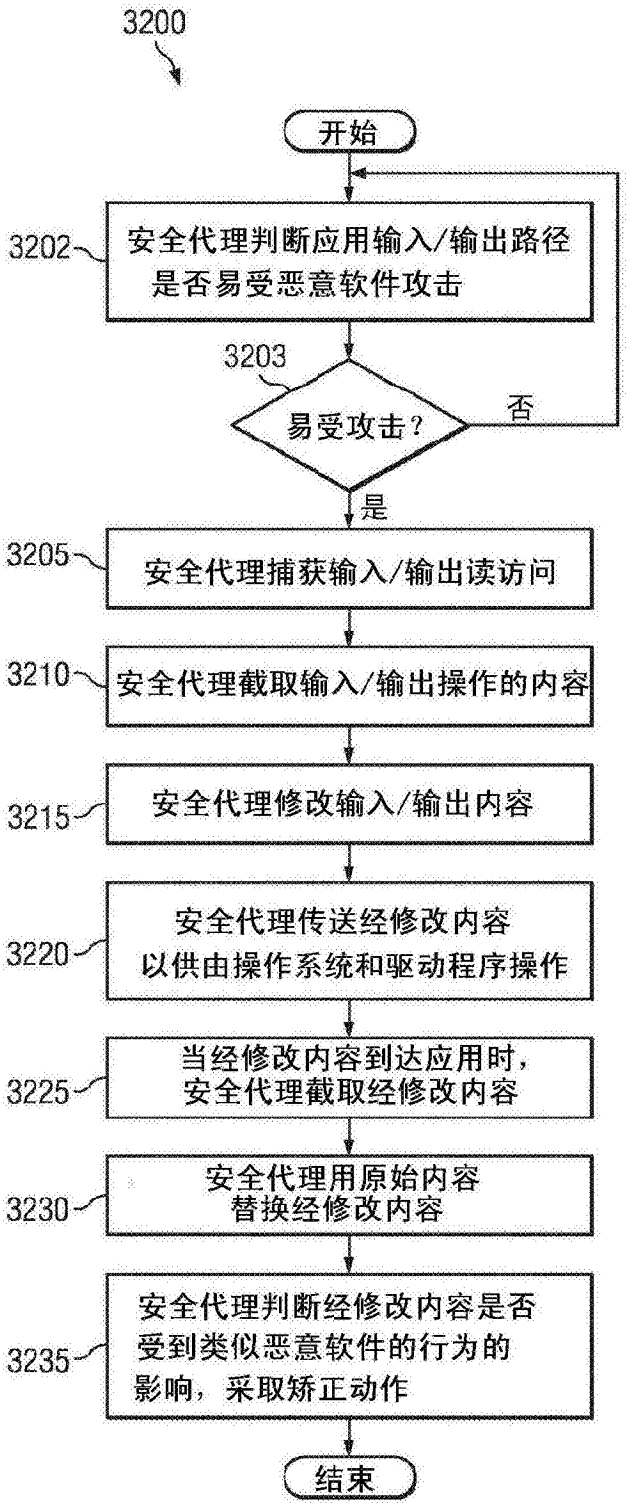

3200
开始
3202
安全代理判断应用输入/输出路径
是否易受恶意软件攻击
3203
易受攻击？
否
是
3205
安全代理捕获输入/输出读访问
3210
安全代理截取输入/输出操作的内容
3215
安全代理修改输入/输出内容
3220
安全代理传送经修改内容
以供由操作系统和驱动程序操作
3225
当经修改内容到达应用时，
安全代理截取经修改内容
3230
安全代理用原始内容
替换经修改内容
3235
安全代理判断经修改内容是否
受到类似恶意软件的行为的
影响，采取矫正动作
结束

图32

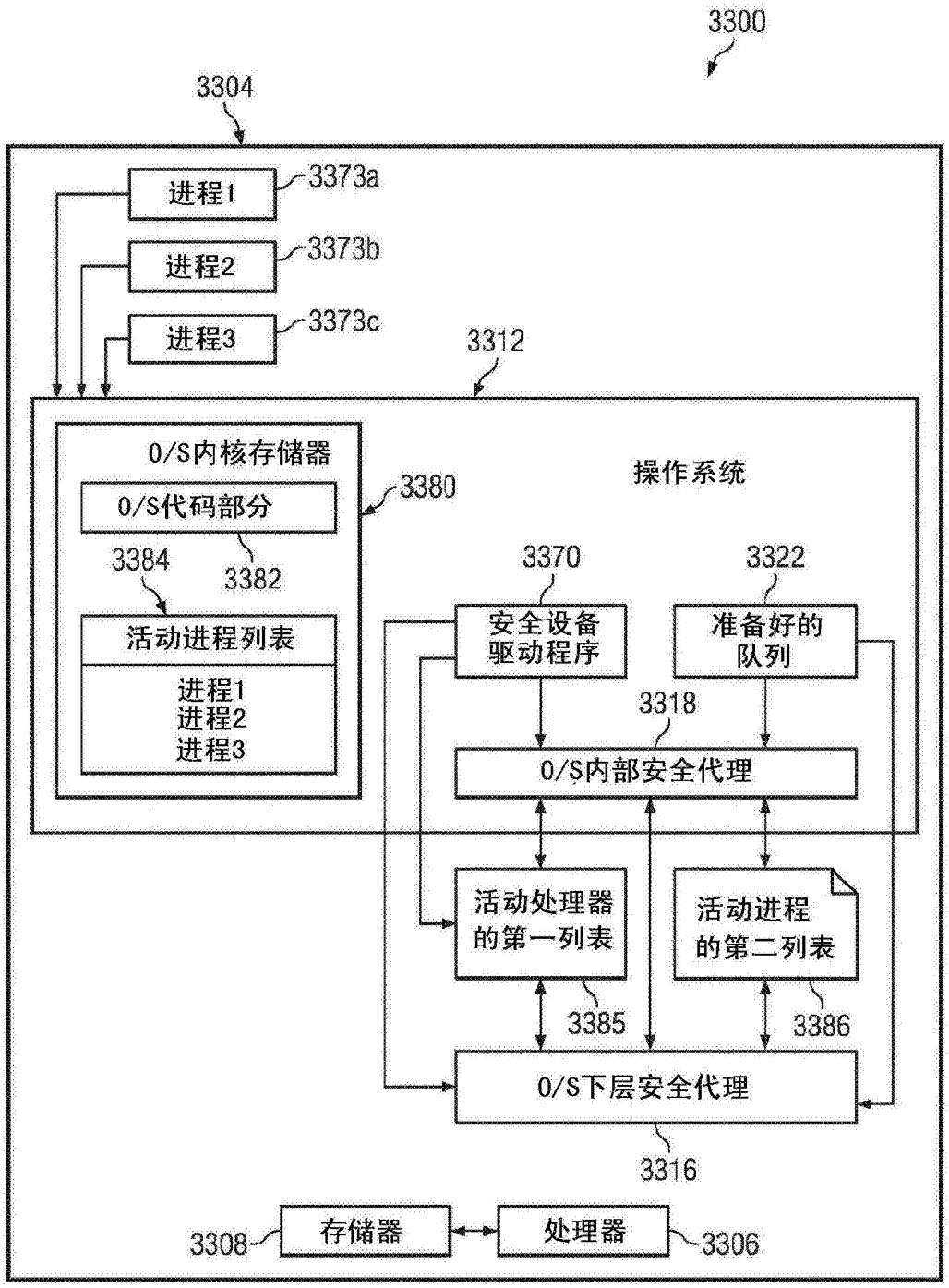

3300
3304
进程1
3373a
进程2
3373b
进程3
3373c
3312
0/S内核存储器
0/S代码部分
3380
操作系统
3384
3382
活动进程列表
进程1
进程2
进程3
3370
3322
安全设备
驱动程序
准备好的
队列
3318
0/S内部安全代理
活动处理器
的第一列表
活动进程
的第二列表
3385
3386
0/S下层安全代理
3316
3308
存储器
处理器
3306

图33

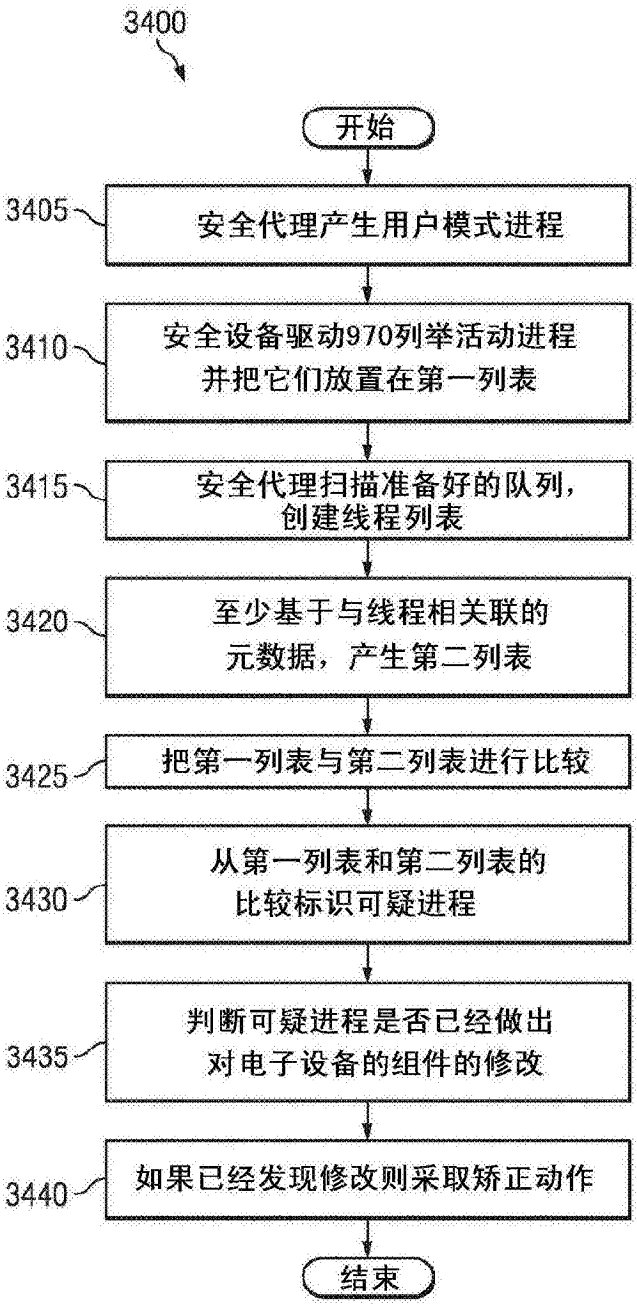

3400
开始
3405
安全代理产生用户模式进程
3410
安全设备驱动970列举活动进程
并把它们放置在第一列表
3415
安全代理扫描准备好的队列，
创建线程列表
3420
至少基于与线程相关联的
元数据，产生第二列表
3425
把第一列表与第二列表进行比较
3430
从第一列表和第二列表的
比较标识可疑进程
3435
判断可疑进程是否已经做出
对电子设备的组件的修改
3440
如果已经发现修改则采取矫正动作
结束

图34

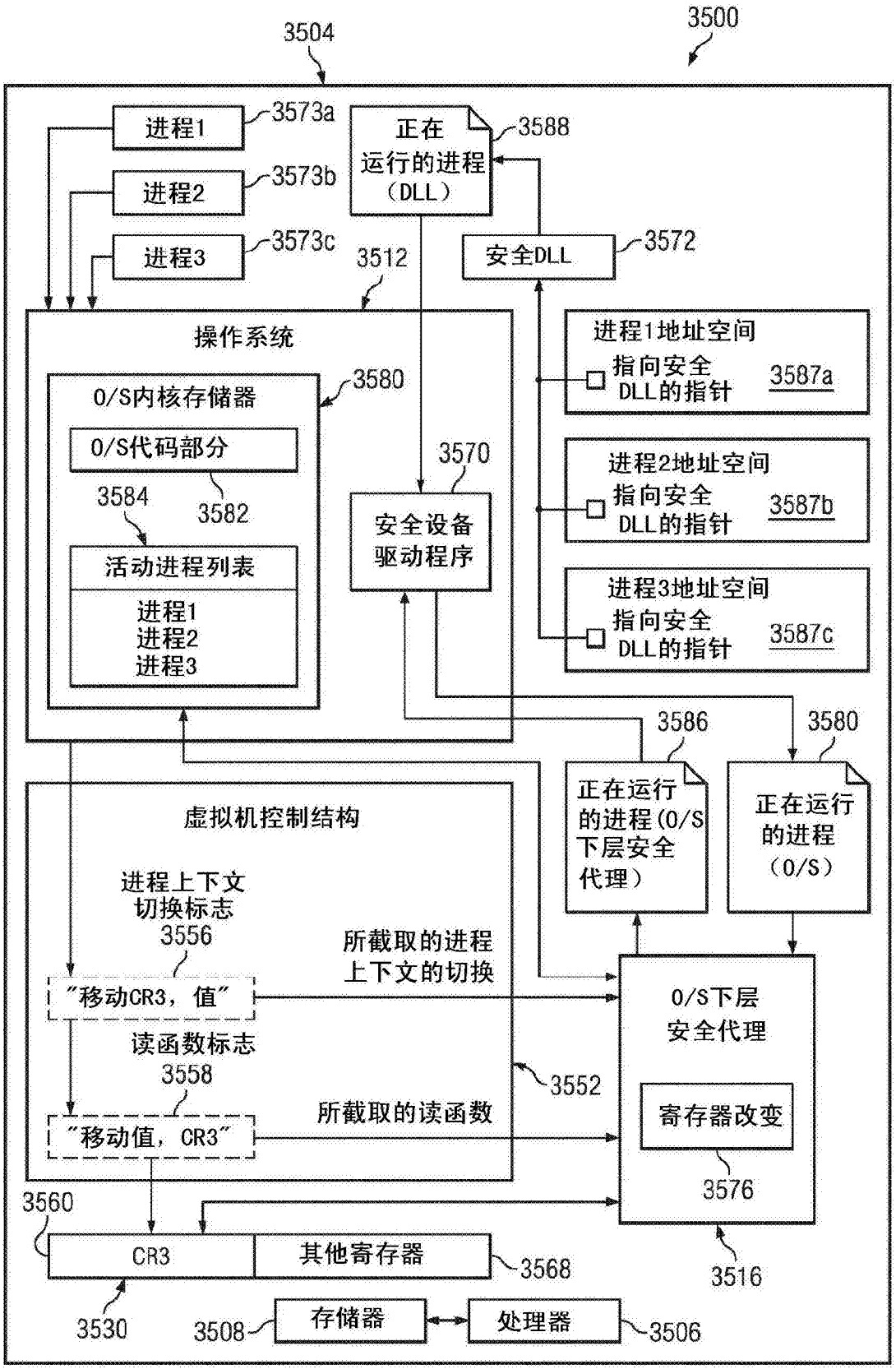

3504
3500
进程1
3573a
进程2
3573b
进程3
3573c
正在运行的进程（DLL）
3588
安全DLL
3572
3512
操作系统
O/S内核存储器
3580
O/S代码部分
3584
3582
活动进程列表
进程1
进程2
进程3
3570
安全设备驱动程序
进程1地址空间
指向安全DLL的指针
3587a
进程2地址空间
指向安全DLL的指针
3587b
进程3地址空间
指向安全DLL的指针
3587c
3586
3580
正在运行的进程(O/S下层安全代理）
正在运行的进程（O/S）
虚拟机控制结构
进程上下文切换标志
3556
"移动CR3，值"
所截取的进程上下文的切换
读函数标志
3558
"移动值，CR3"
所截取的读函数
3552
O/S下层安全代理
寄存器改变
3576
3560
CR3
其他寄存器
3568
3516
3530
3508
存储器
处理器
3506

图35

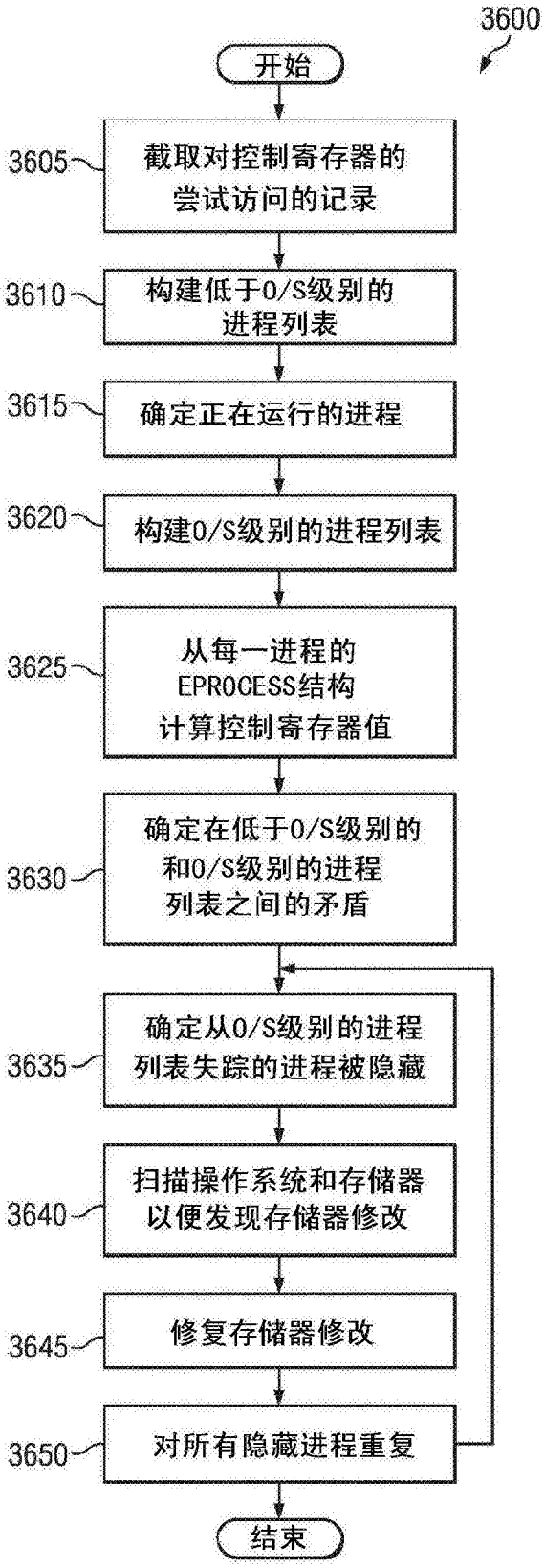

3600
开始
3605
截取对控制寄存器的尝试访问的记录
3610
构建低于O/S级别的进程列表
3615
确定正在运行的进程
3620
构建O/S级别的进程列表
3625
从每一进程的EPROCESS结构计算控制寄存器值
3630
确定在低于O/S级别的和O/S级别的进程列表之间的矛盾
3635
确定从O/S级别的进程列表失踪的进程被隐藏
3640
扫描操作系统和存储器以便发现存储器修改
3645
修复存储器修改
3650
对所有隐藏进程重复
结束

图36

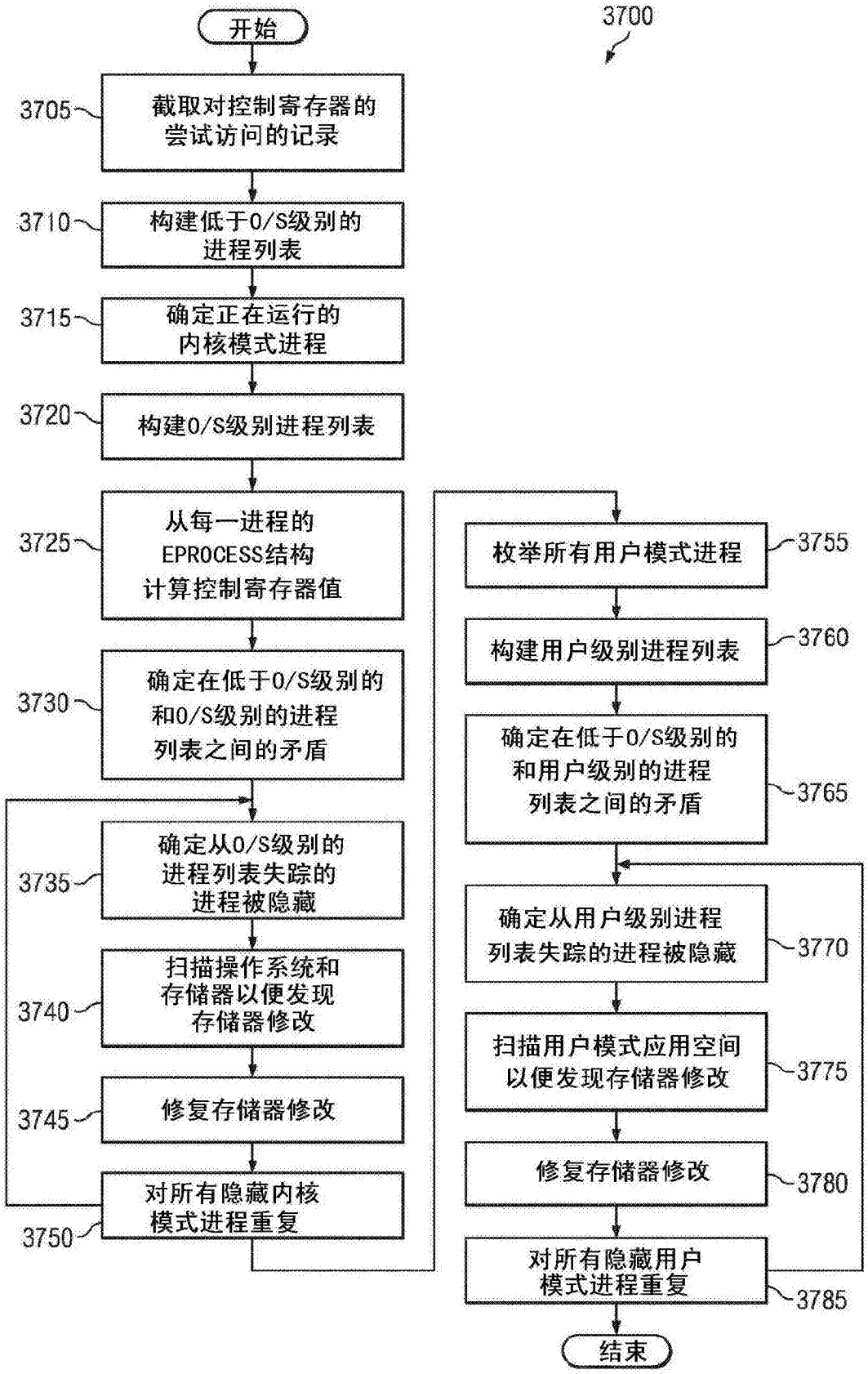

3700
开始
3705
截取对控制寄存器的尝试访问的记录
3710
构建低于O/S级别的进程列表
3715
确定正在运行的内核模式进程
3720
构建O/S级别进程列表
3725
从每一进程的EPROCESS结构计算控制寄存器值
3730
确定在低于O/S级别的和O/S级别的进程列表之间的矛盾
3735
确定从O/S级别的进程列表失踪的进程被隐藏
3740
扫描操作系统和存储器以便发现存储器修改
3745
修复存储器修改
3750
对所有隐藏内核模式进程重复
3755
枚举所有用户模式进程
3760
构建用户级别进程列表
3765
确定在低于O/S级别的和用户级别的进程列表之间的矛盾
3770
确定从用户级别进程列表失踪的进程被隐藏
3775
扫描用户模式应用空间以便发现存储器修改
3780
修复存储器修改
3785
对所有隐藏用户模式进程重复
结束

图37

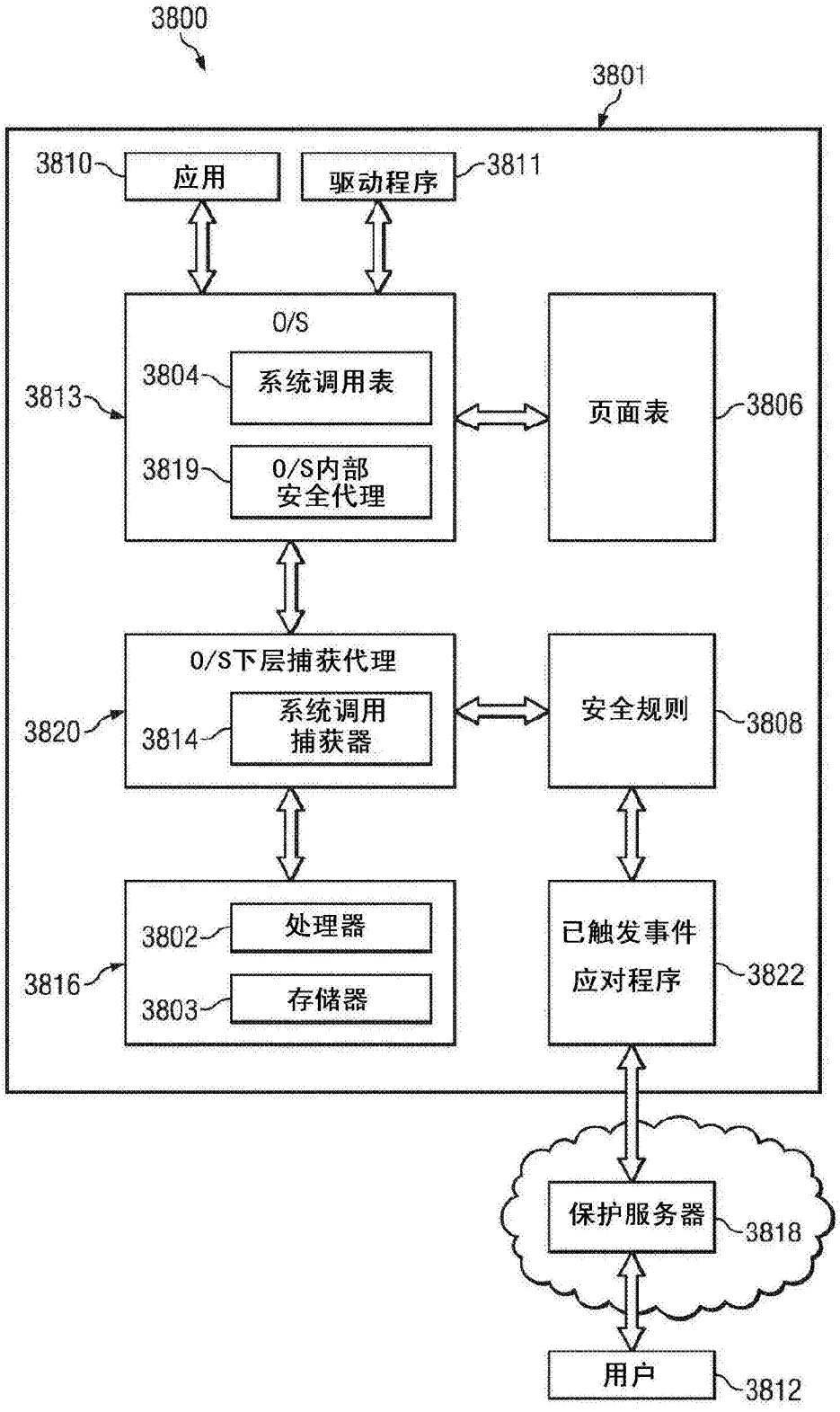

3800
3801
3810
应用
驱动程序
3811
O/S
3804
系统调用表
3813
3819
O/S内部
安全代理
页面表
3806
O/S下层捕获代理
3820
3814
系统调用
捕获器
安全规则
3808
3802
处理器
3816
3803
存储器
已触发事件
应对程序
3822
保护服务器
3818
用户
3812

图38

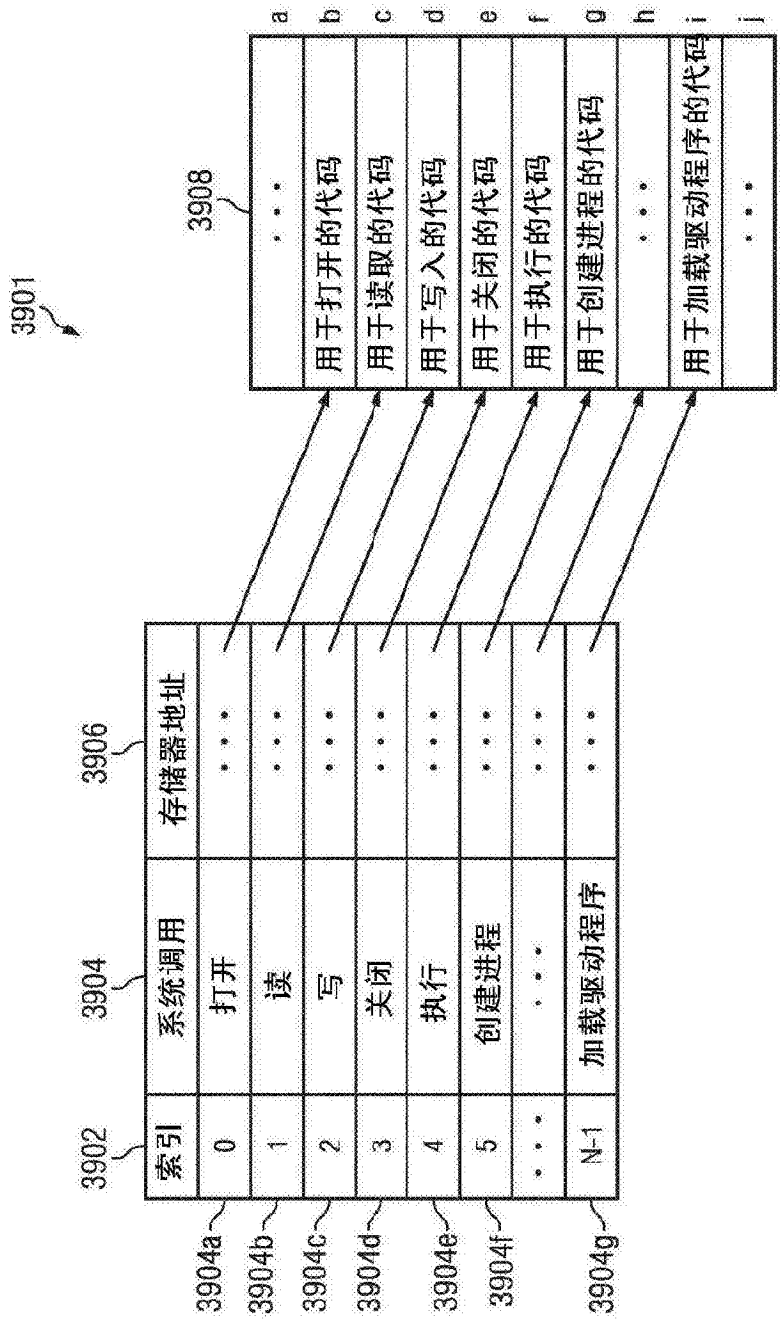

3901
3902
索引
0
1
2
3
4
5
• • •
N-1
3904
系统调用
打开
读
写
关闭
执行
创建进程
• • •
加载驱动程序
3906
存储器地址
3908
a
b
c
d
e
f
g
h
i
j
用于打开的代码
用于读取的代码
用于写入的代码
用于关闭的代码
用于执行的代码
用于创建进程的代码
用于加载驱动程序的代码
3904a
3904b
3904c
3904d
3904e
3904f
3904g

图39

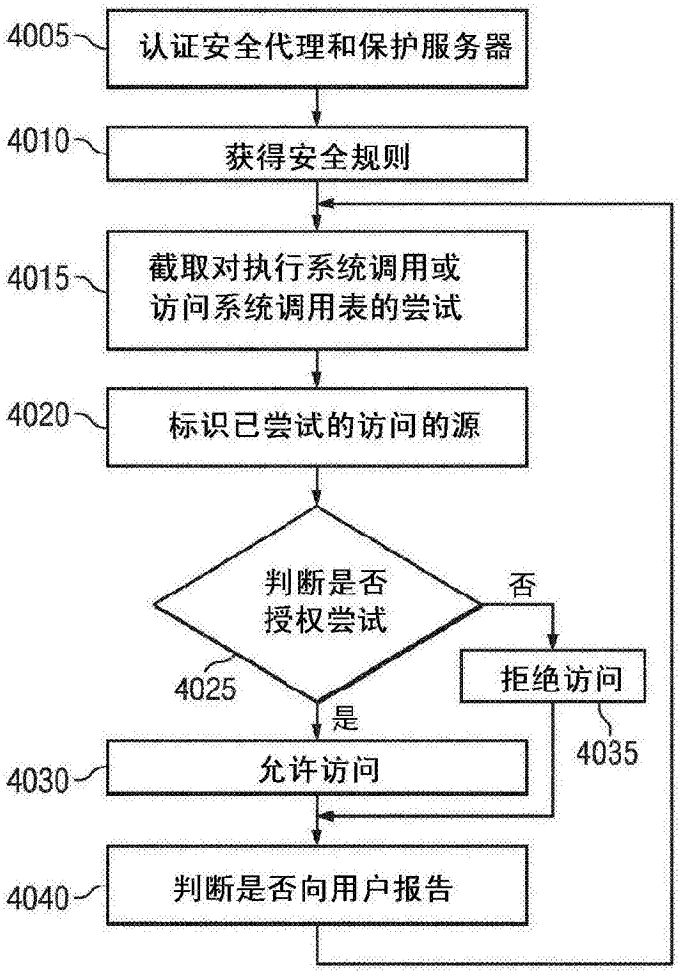

4005
认证安全代理和保护服务器
4010
获得安全规则
4015
截取对执行系统调用或
访问系统调用表的尝试
4020
标识已尝试的访问的源
判断是否
授权尝试
否
4025
拒绝访问
是
4035
4030
允许访问
4040
判断是否向用户报告

图40

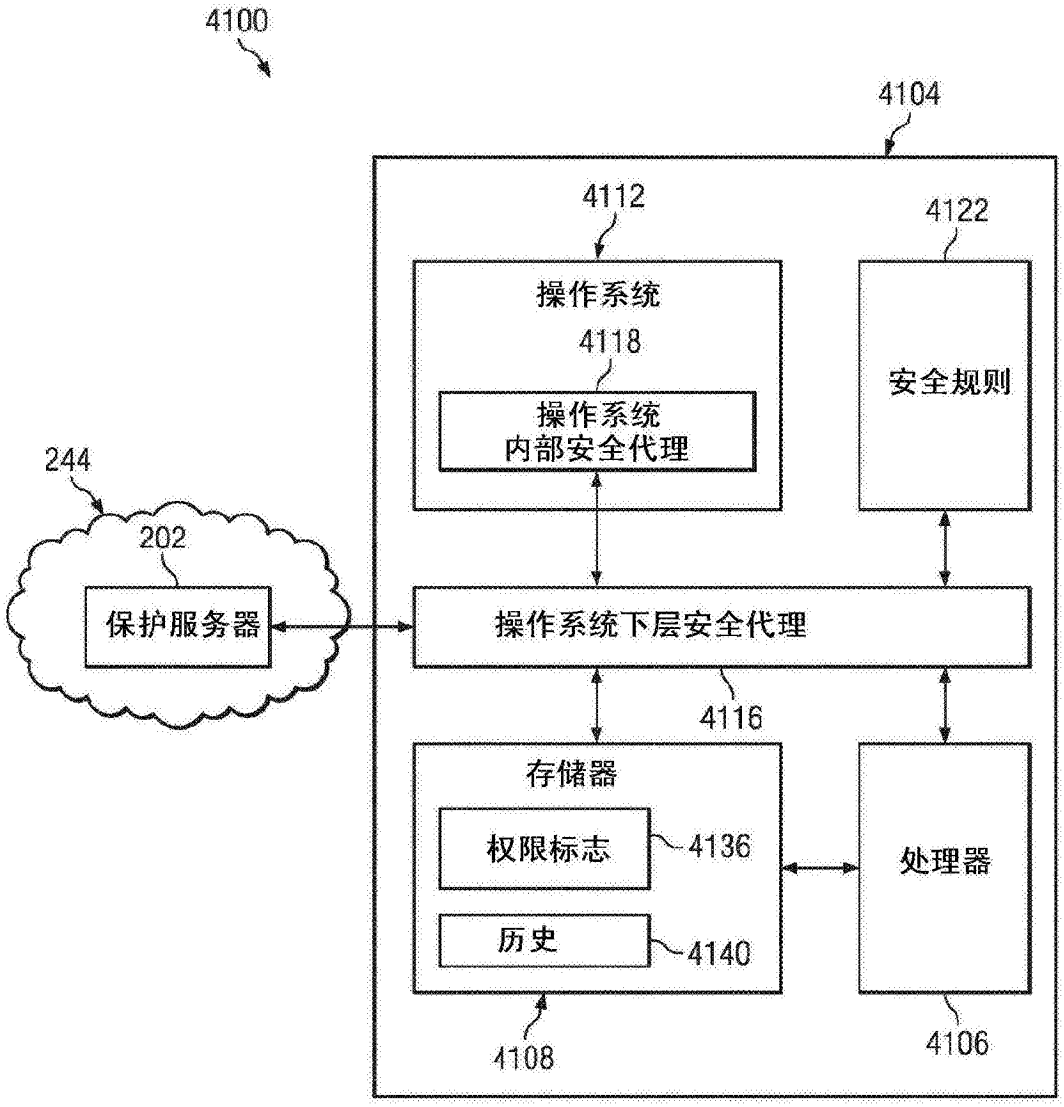

4100
4104
4112
4122
操作系统
4118
操作系统
内部安全代理
安全规则
244
202
保护服务器
操作系统下层安全代理
4116
存储器
权限标志
4136
历史
4140
处理器
4108
4106

图41

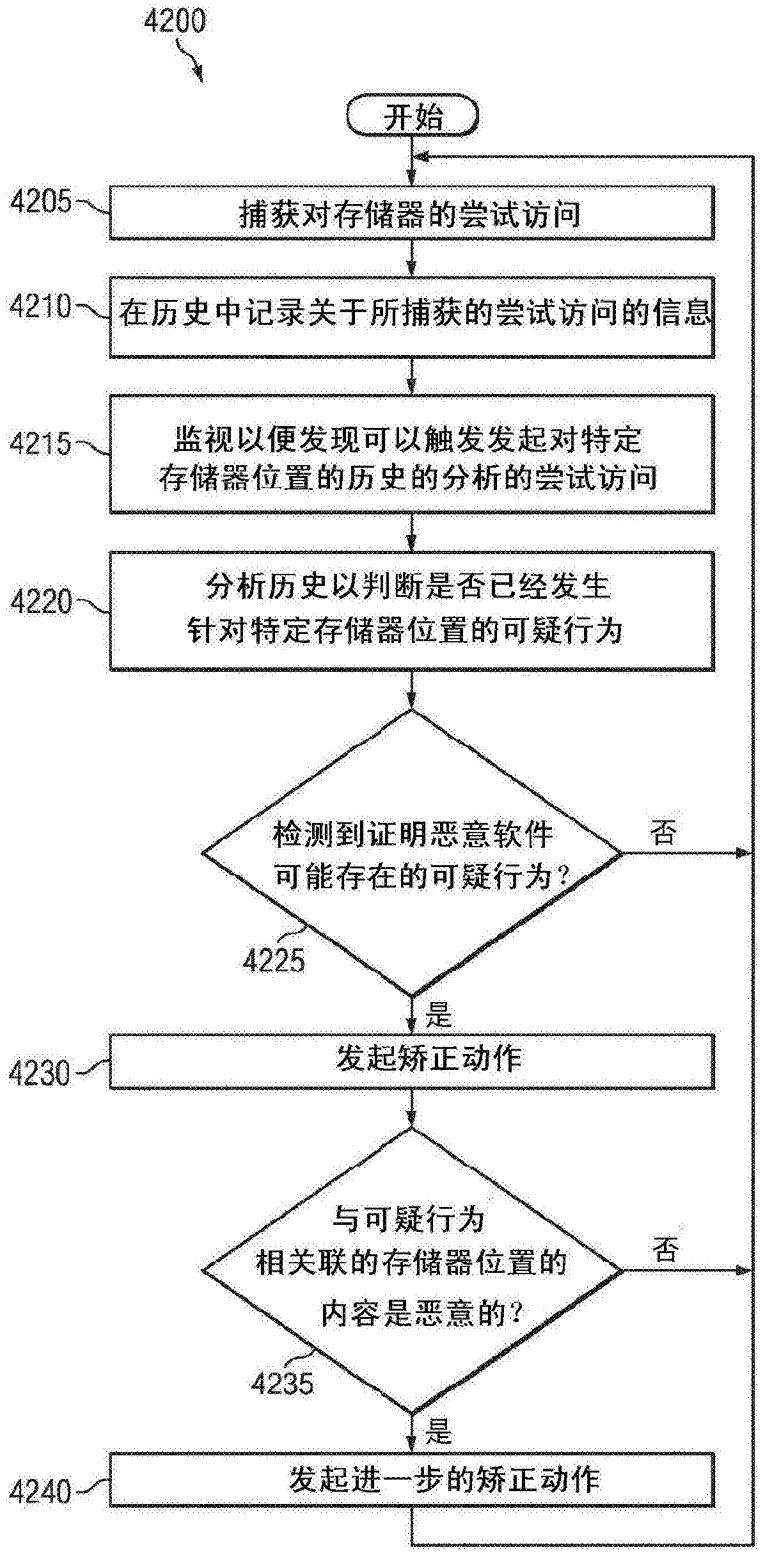

4200
开始
4205
捕获对存储器的尝试访问
4210
在历史中记录关于所捕获的尝试访问的信息
4215
监视以便发现可以触发发起对特定存储器位置的历史的分析的尝试访问
4220
分析历史以判断是否已经发生针对特定存储器位置的可疑行为
检测到证明恶意软件可能存在的可疑行为?
否
4225
是
4230
发起矫正动作
与可疑行为相关联的存储器位置的内容是恶意的?
否
4235
是
4240
发起进一步的矫正动作

图42

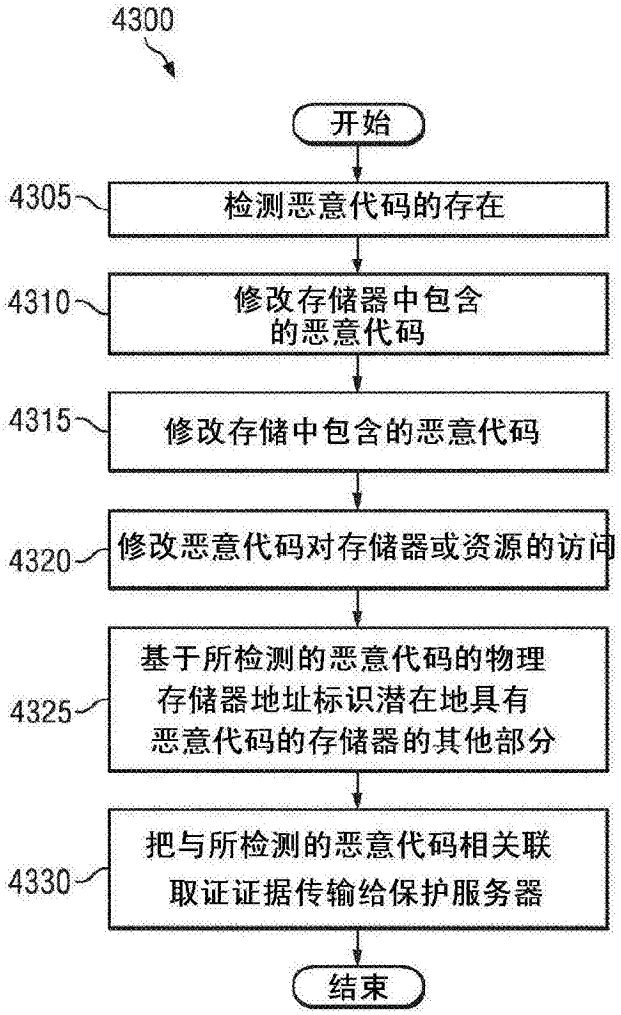

4300
开始
4305
检测恶意代码的存在
4310
修改存储器中包含的恶意代码
4315
修改存储中包含的恶意代码
4320
修改恶意代码对存储器或资源的访问
4325
基于所检测的恶意代码的物理存储器地址标识潜在地具有恶意代码的存储器的其他部分
4330
把与所检测的恶意代码相关联取证证据传输给保护服务器
结束

图43

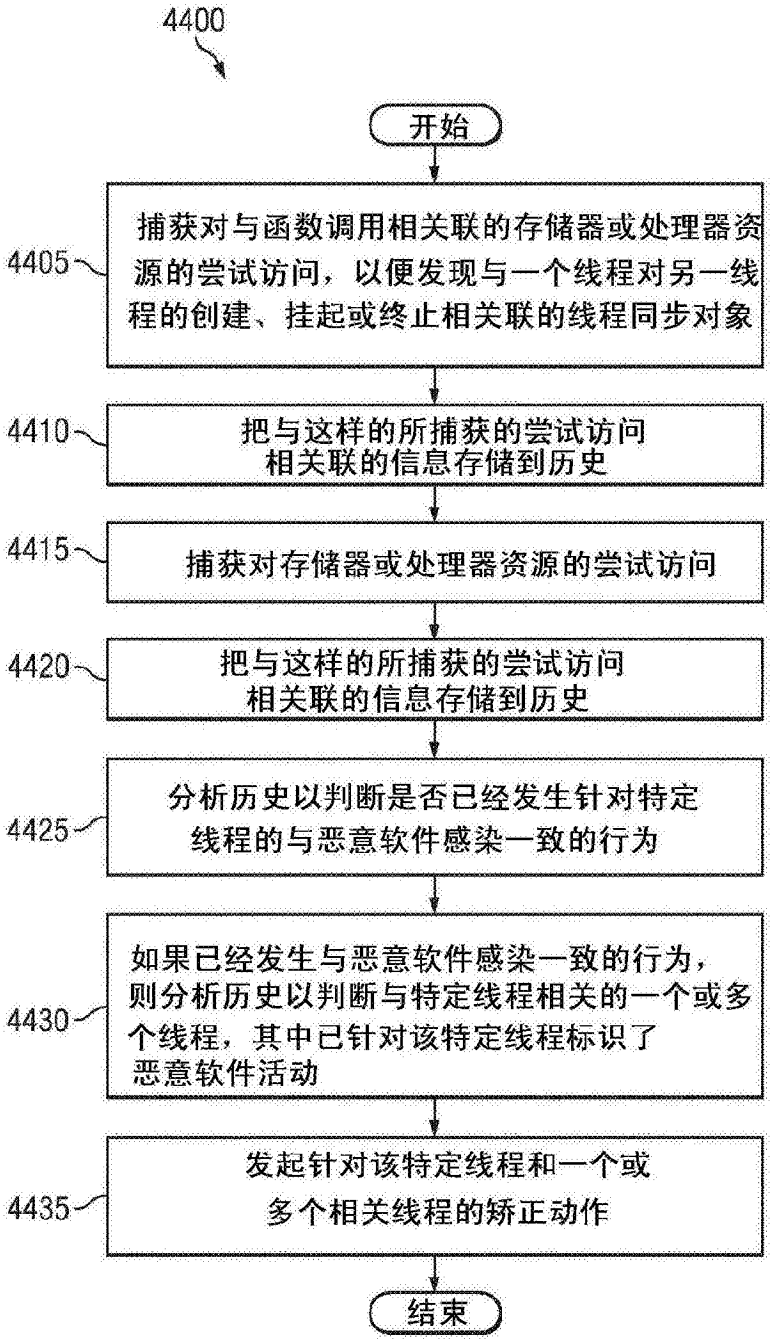

4400
开始
捕获对与函数调用相关联的存储器或处理器资源的尝试访问，以便发现与一个线程对另一线程的创建、挂起或终止相关联的线程同步对象
4405
把与这样的所捕获的尝试访问相关联的信息存储到历史
4410
捕获对存储器或处理器资源的尝试访问
4415
把与这样的所捕获的尝试访问相关联的信息存储到历史
4420
分析历史以判断是否已经发生针对特定线程的与恶意软件感染一致的行为
4425
如果已经发生与恶意软件感染一致的行为，则分析历史以判断与特定线程相关的一个或多个线程，其中已针对该特定线程标识了恶意软件活动
4430
发起针对该特定线程和一个或多个相关线程的矫正动作
4435
结束

图44

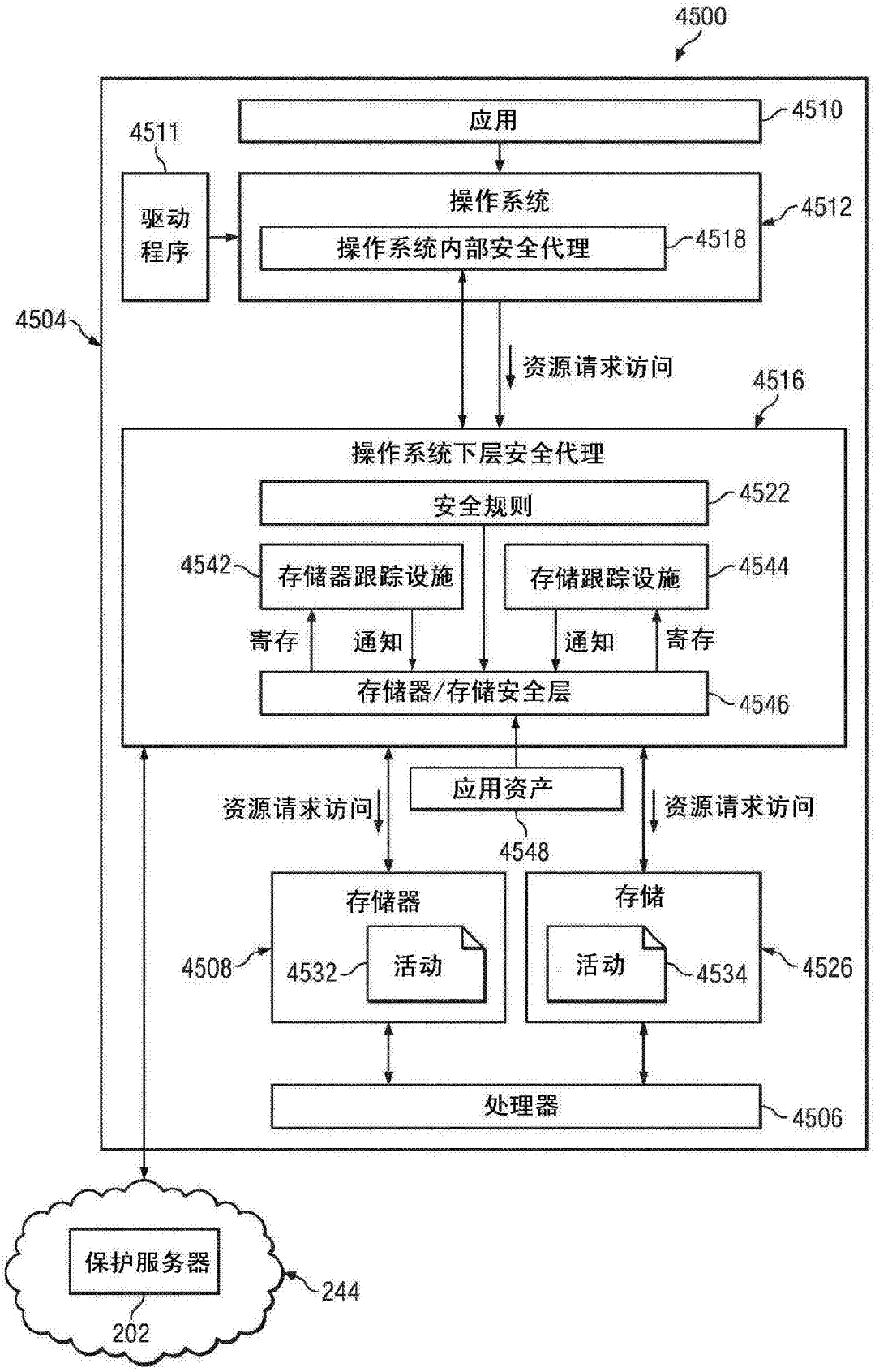

4500
应用
4510
4511
驱动程序
操作系统
4512
操作系统内部安全代理
4518
4504
资源请求访问
4516
操作系统下层安全代理
安全规则
4522
4542
存储器跟踪设施
存储跟踪设施
4544
寄存
通知
通知
寄存
存储器/存储安全层
4546
应用资产
4548
资源请求访问
资源请求访问
存储器
存储
4508
4532
活动
活动
4534
4526
处理器
4506
保护服务器
244
202

图45

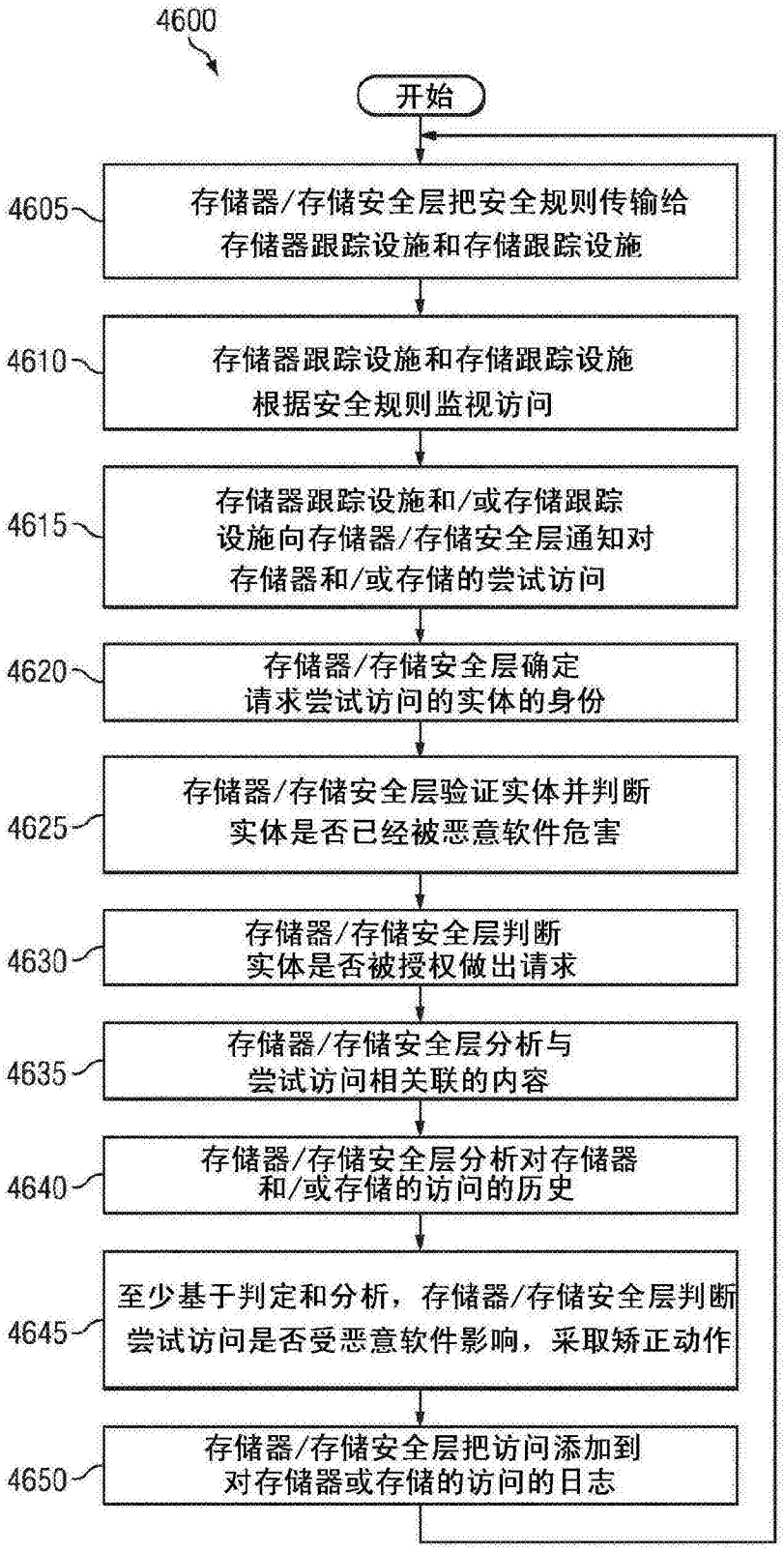

4600
开始
4605
存储器/存储安全层把安全规则传输给存储器跟踪设施和存储跟踪设施
4610
存储器跟踪设施和存储跟踪设施根据安全规则监视访问
4615
存储器跟踪设施和/或存储跟踪设施向存储器/存储安全层通知对存储器和/或存储的尝试访问
4620
存储器/存储安全层确定请求尝试访问的实体的身份
4625
存储器/存储安全层验证实体并判断实体是否已经被恶意软件危害
4630
存储器/存储安全层判断实体是否被授权做出请求
4635
存储器/存储安全层分析与尝试访问相关联的内容
4640
存储器/存储安全层分析对存储器和/或存储的访问的历史
4645
至少基于判定和分析，存储器/存储安全层判断尝试访问是否受恶意软件影响，采取矫正动作
4650
存储器/存储安全层把访问添加到对存储器或存储的访问的日志

图46

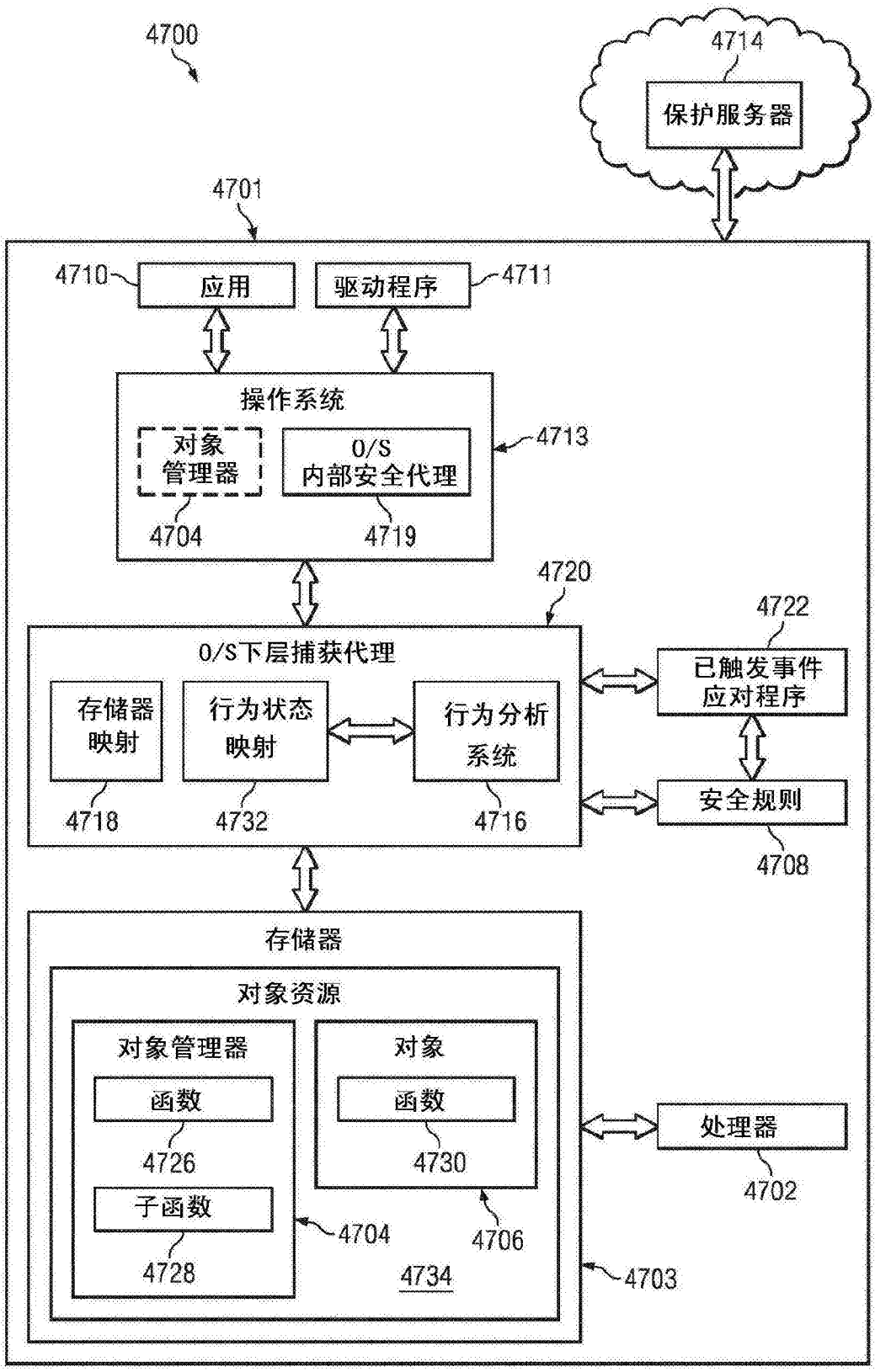

4700
4714
保护服务器
4701
4710
应用
驱动程序
4711
操作系统
对象
管理器
O/S
内部安全代理
4713
4704
4719
4720
4722
O/S下层捕获代理
存储器
映射
行为状态
映射
行为分析
系统
已触发事件
应对程序
安全规则
4718
4732
4716
4708
存储器
对象资源
对象管理器
函数
4726
子函数
4728
4704
对象
函数
4730
4706
4734
处理器
4702
4703

图47

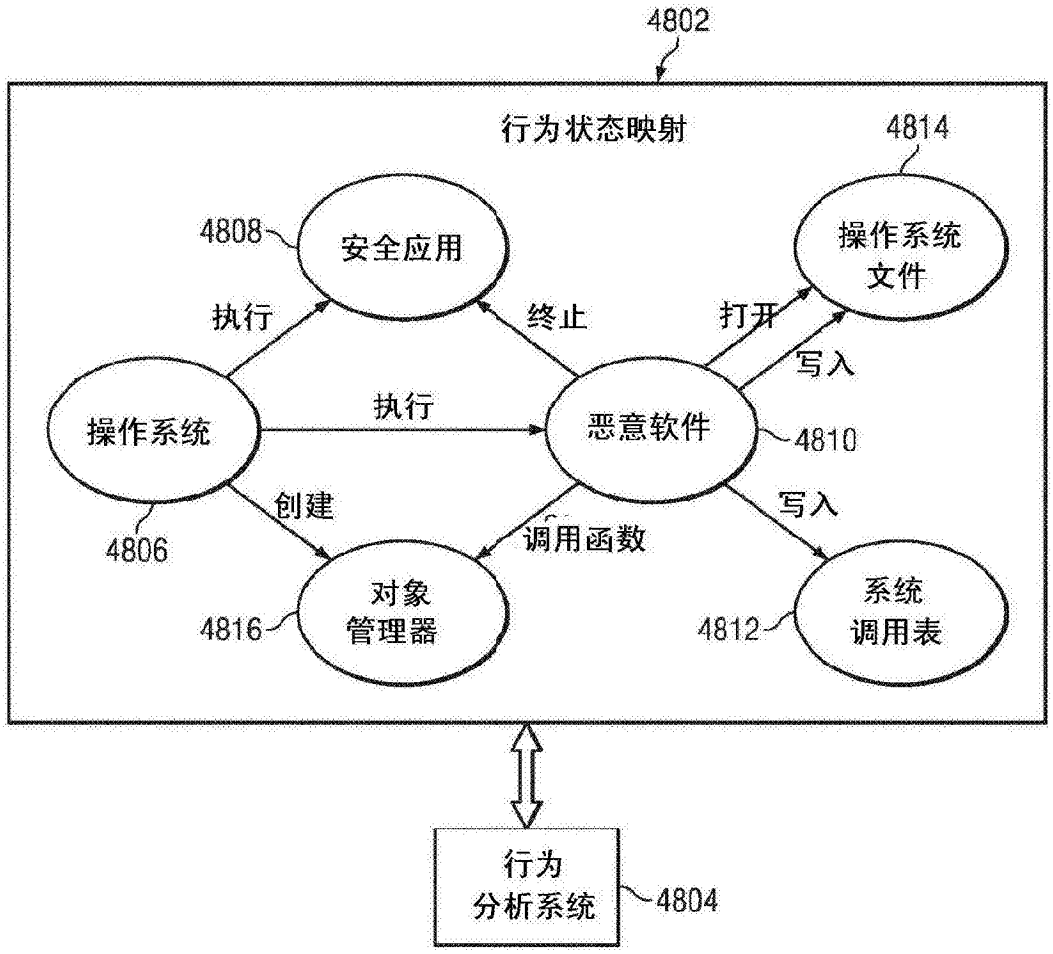

4802
行为状态映射
4814
4808
安全应用
操作系统
文件
执行
终止
打开
写入
操作系统
执行
恶意软件
4810
创建
写入
4806
调用函数
对象
管理器
系统
调用表
4816
4812
行为
分析系统
4804

图48

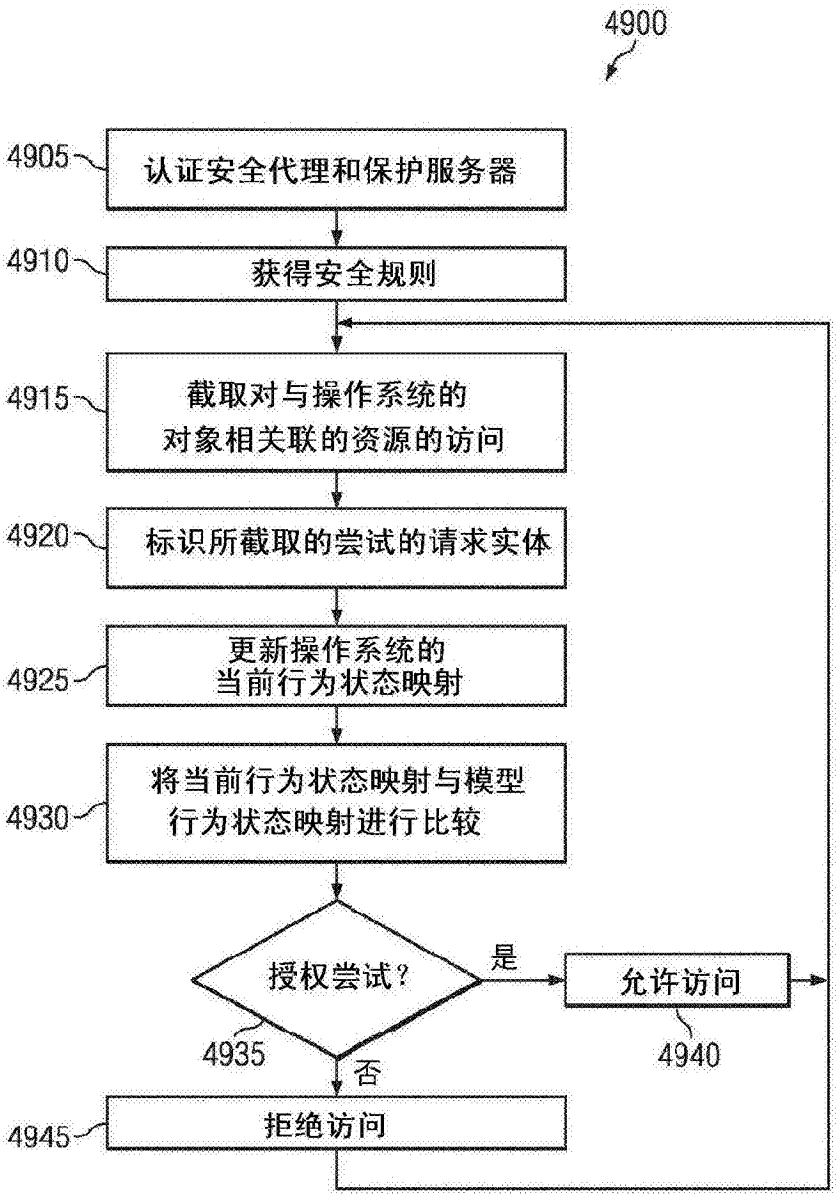

4900
4905
认证安全代理和保护服务器
4910
获得安全规则
4915
截取对与操作系统的
对象相关联的资源的访问
4920
标识所截取的尝试的请求实体
4925
更新操作系统的
当前行为状态映射
4930
将当前行为状态映射与模型
行为状态映射进行比较
授权尝试？
是
允许访问
4940
4935
否
4945
拒绝访问

图49

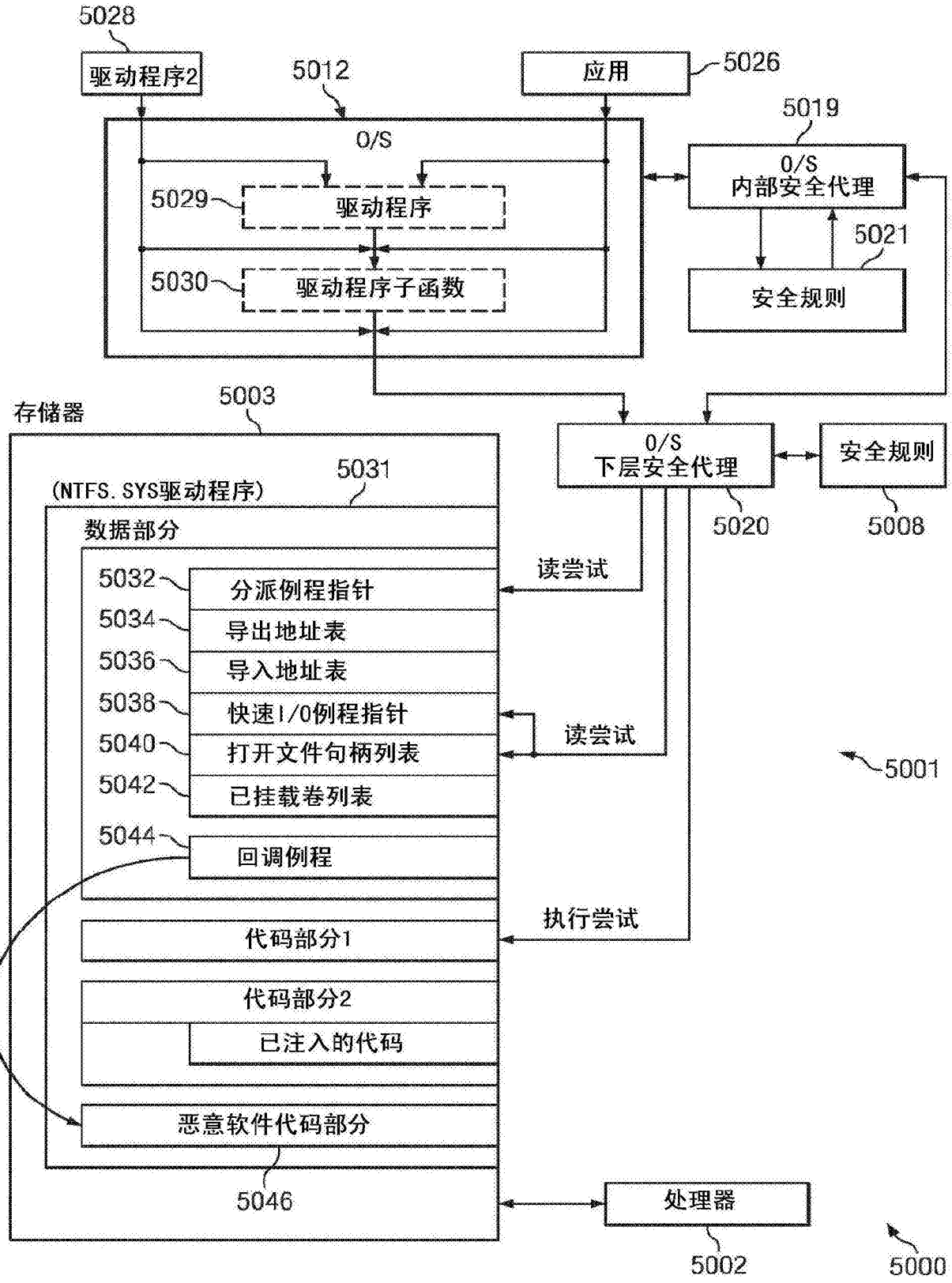

5028
驱动程序2
5012
应用
5026
5019
O/S
O/S
内部安全代理
5029
驱动程序
5021
5030
驱动程序子函数
安全规则
5003
存储器
O/S
下层安全代理
安全规则
5031
(NTFS.SYS驱动程序)
数据部分
5020
5008
读尝试
5032
分派例程指针
5034
导出地址表
5036
导入地址表
5038
快速I/O例程指针
读尝试
5040
打开文件句柄列表
5001
5042
已挂载卷列表
5044
回调例程
执行尝试
代码部分1
代码部分2
已注入的代码
恶意软件代码部分
5046
处理器
5002
5000

图50

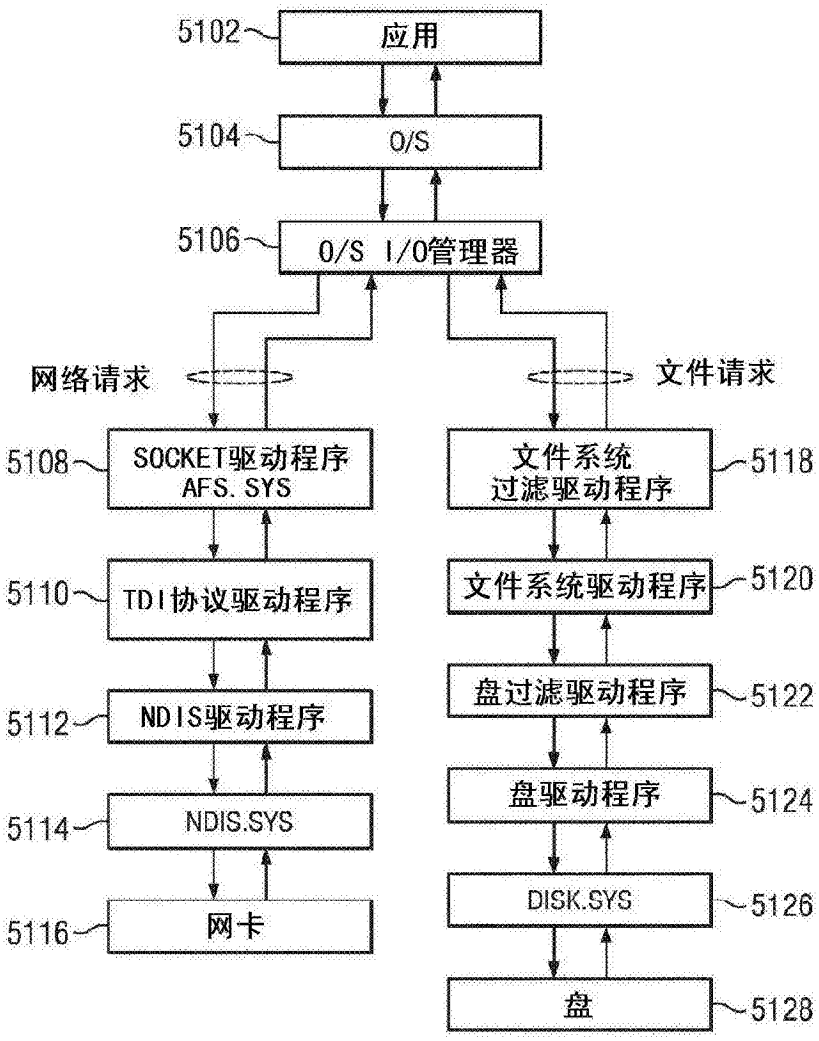

5102
应用
5104
O/S
5106
O/S I/O管理器
网络请求
文件请求
5108
SOCKET驱动程序
AFS. SYS
5110
TDI协议驱动程序
5112
NDIS驱动程序
5114
NDIS.SYS
5116
网卡
文件系统
过滤驱动程序
5118
文件系统驱动程序
5120
盘过滤驱动程序
5122
盘驱动程序
5124
DISK.SYS
5126
盘
5128

图51

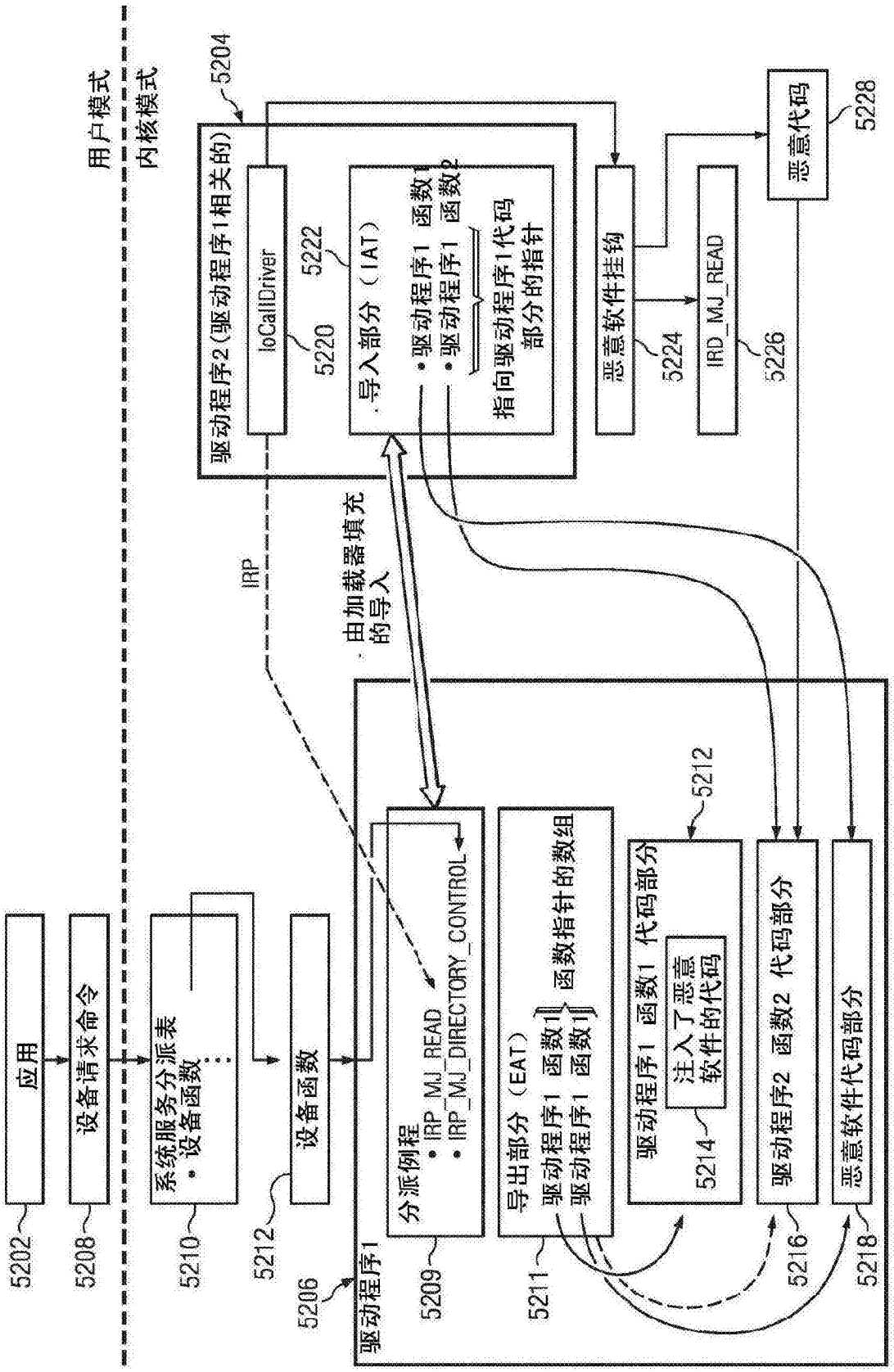

用户模式
内核模式
应用
设备请求命令
系统服务分派表
• 设备函数
设备函数
驱动程序1
分派例程
• IRP_MJ_READ
• IRP_MJ_DIRECTORY_CONTROL
导出部分（EAT）
驱动程序1 函数1
驱动程序1 函数1
函数指针的数组
驱动程序1 函数1 代码部分
注入了恶意软件的代码
驱动程序2 函数2 代码部分
恶意软件代码部分
IRP
由加载器填充的导入
驱动程序2（驱动程序1相关的）
IoCallDriver
导入部分（IAT）
•驱动程序1 函数1
•驱动程序1 函数2
指向驱动程序1代码部分的指针
恶意软件挂钩
IRD_MJ_READ
恶意代码
5202
5208
5210
5212
5206
5209
5211
5212
5214
5216
5218
5204
5220
5222
5224
5226
5228

图52

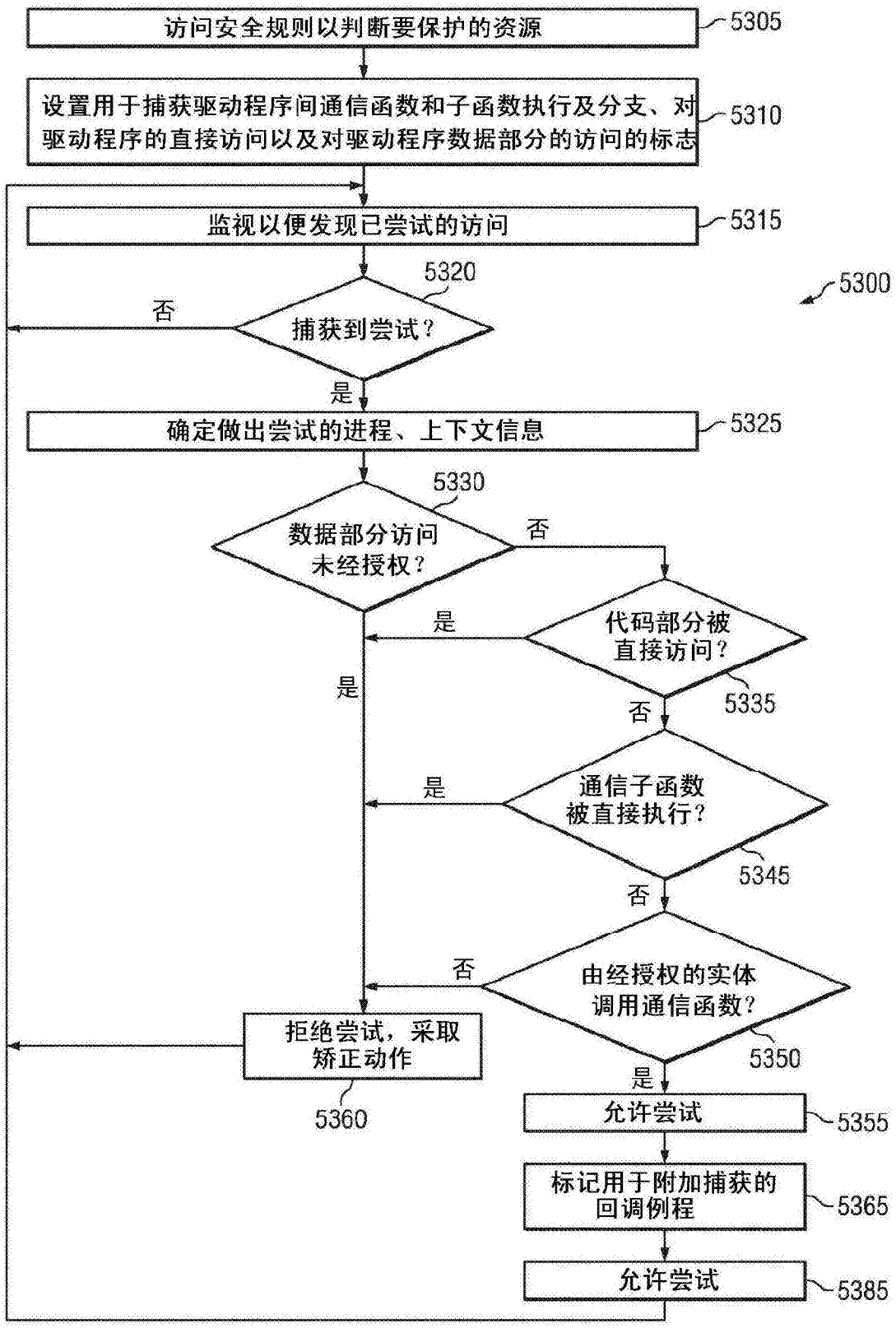

访问安全规则以判断要保护的资源
5305
设置用于捕获驱动程序间通信函数和子函数执行及分支、对驱动程序的直接访问以及对驱动程序数据部分的访问的标志
5310
监视以便发现已尝试的访问
5315
5320
5300
否
捕获到尝试？
是
确定做出尝试的进程、上下文信息
5325
5330
数据部分访问未经授权？
否
是
代码部分被直接访问？
5335
是
否
通信子函数被直接执行？
5345
是
否
由经授权的实体调用通信函数？
5350
否
拒绝尝试，采取矫正动作
5360
是
允许尝试
5355
标记用于附加捕获的回调例程
5365
允许尝试
5385

图53

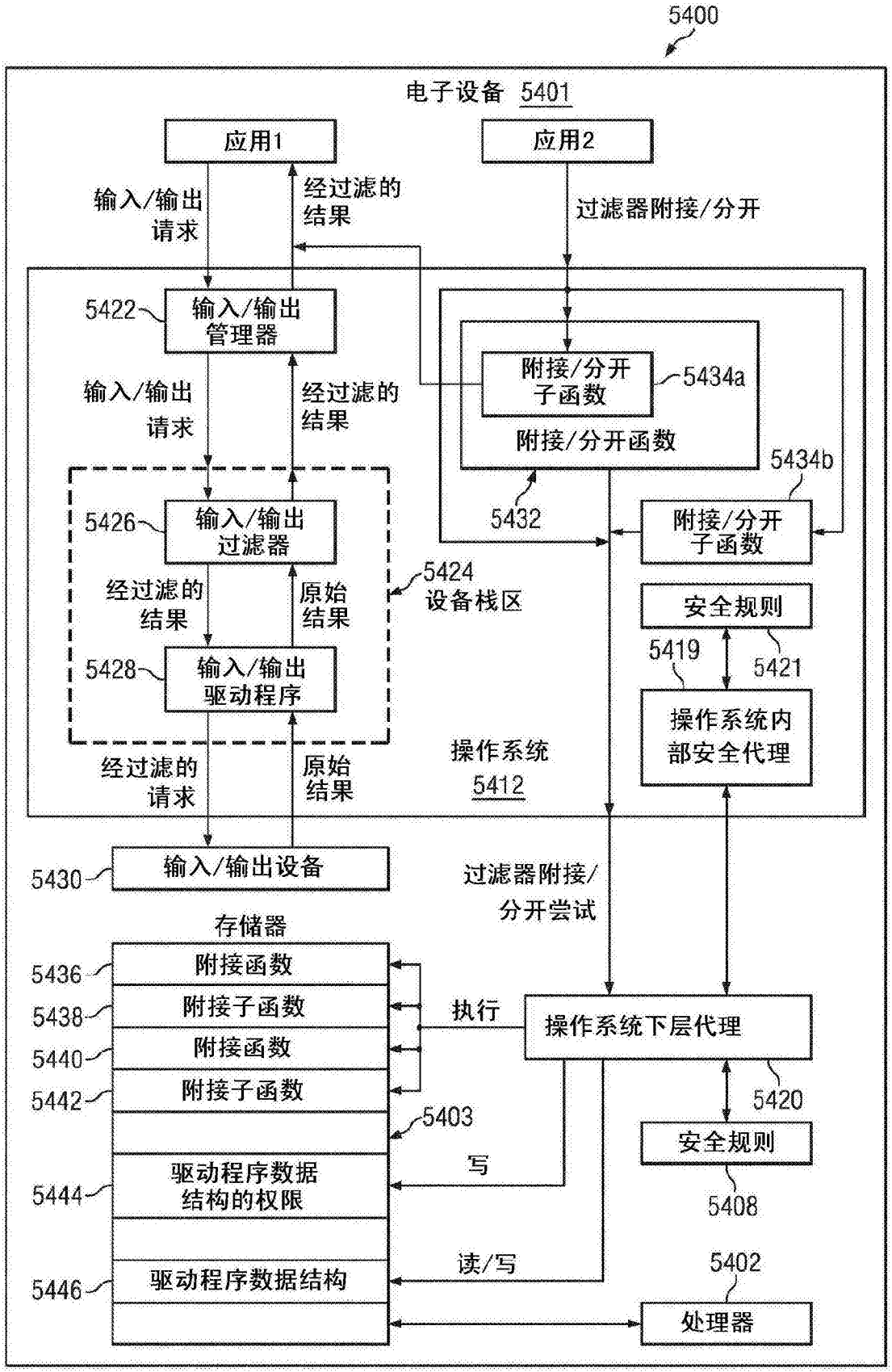

5400
电子设备 5401
应用1
应用2
输入/输出请求
经过滤的结果
过滤器附接/分开
5422
输入/输出管理器
附接/分开子函数
5434a
附接/分开函数
5434b
输入/输出请求
经过滤的结果
5426
输入/输出过滤器
5432
附接/分开子函数
5424
设备栈区
经过滤的结果
原始结果
安全规则
5419
5421
5428
输入/输出驱动程序
操作系统内部安全代理
操作系统
5412
经过滤的请求
原始结果
5430
输入/输出设备
过滤器附接/分开尝试
存储器
5436
附接函数
5438
附接子函数
执行
操作系统下层代理
5440
附接函数
5442
附接子函数
5420
5403
安全规则
写
5444
驱动程序数据结构的权限
5408
读/写
5402
5446
驱动程序数据结构
处理器

图54

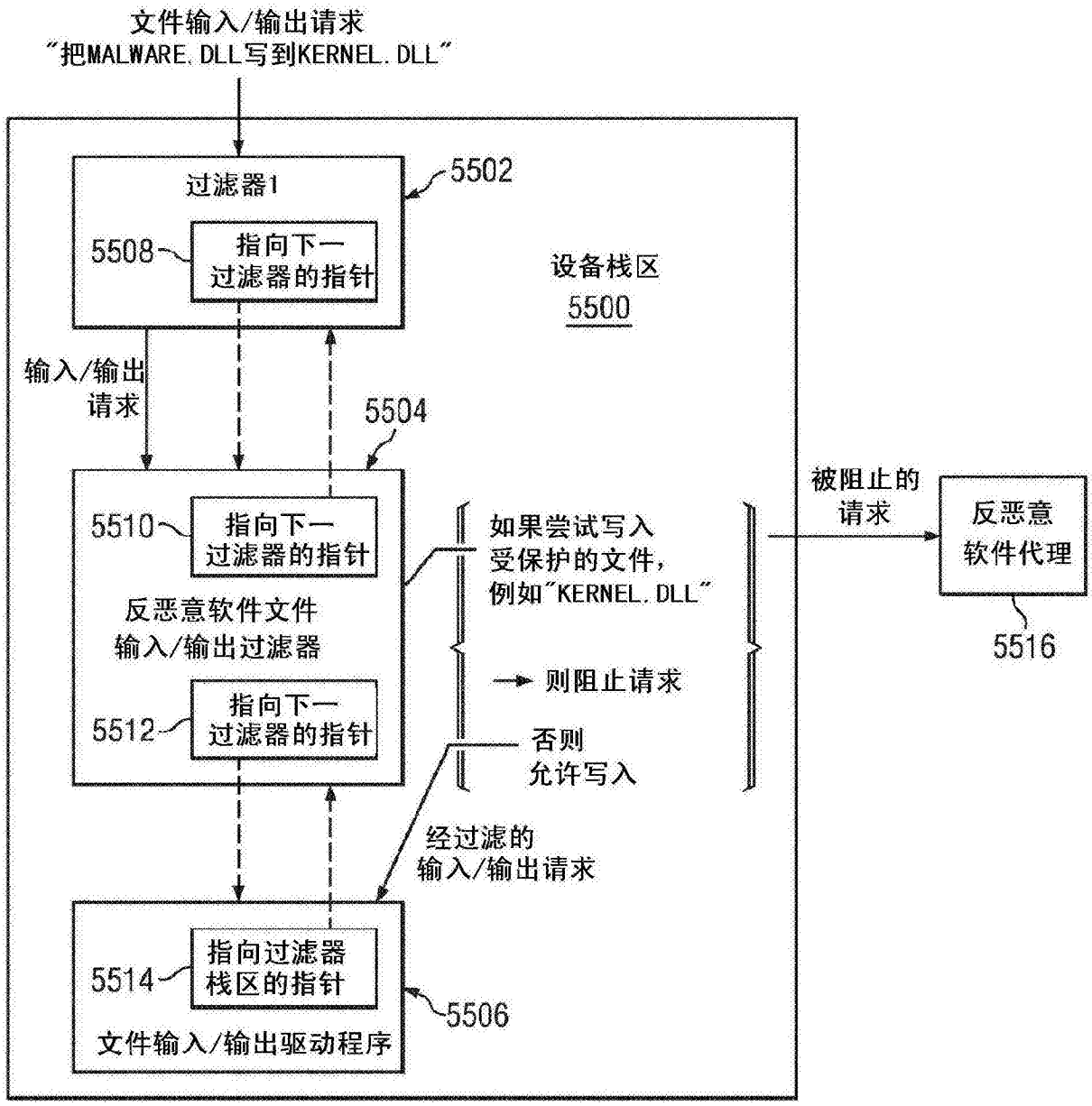

文件输入/输出请求
"把MALWARE.DLL写到KERNEL.DLL"
过滤器1
5502
5508
指向下一过滤器的指针
设备栈区
5500
输入/输出请求
5504
5510
指向下一过滤器的指针
反恶意软件文件输入/输出过滤器
5512
指向下一过滤器的指针
如果尝试写入受保护的文件，例如"KERNEL.DLL"
则阻止请求
否则允许写入
被阻止的请求
反恶意软件代理
5516
经过滤的输入/输出请求
5514
指向过滤器栈区的指针
5506
文件输入/输出驱动程序

图55

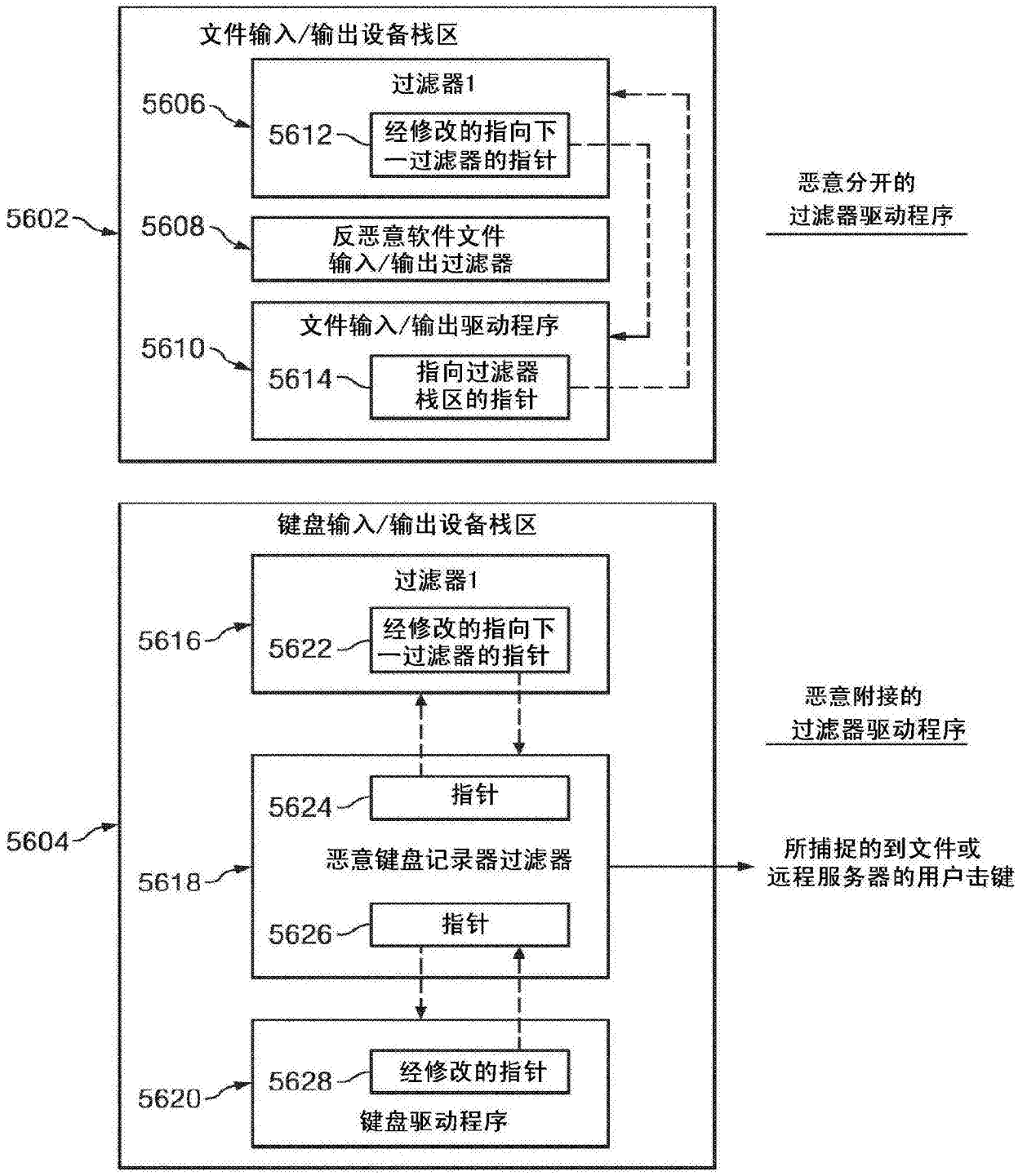

文件输入/输出设备栈区
过滤器1
5606
5612
经修改的指向下一过滤器的指针
5602
5608
反恶意软件文件输入/输出过滤器
文件输入/输出驱动程序
5610
5614
指向过滤器栈区的指针
恶意分开的过滤器驱动程序
键盘输入/输出设备栈区
过滤器1
5616
5622
经修改的指向下一过滤器的指针
恶意附接的过滤器驱动程序
5624
指针
5604
恶意键盘记录器过滤器
所捕捉的到文件或远程服务器的用户击键
5618
5626
指针
5628
经修改的指针
5620
键盘驱动程序

图56

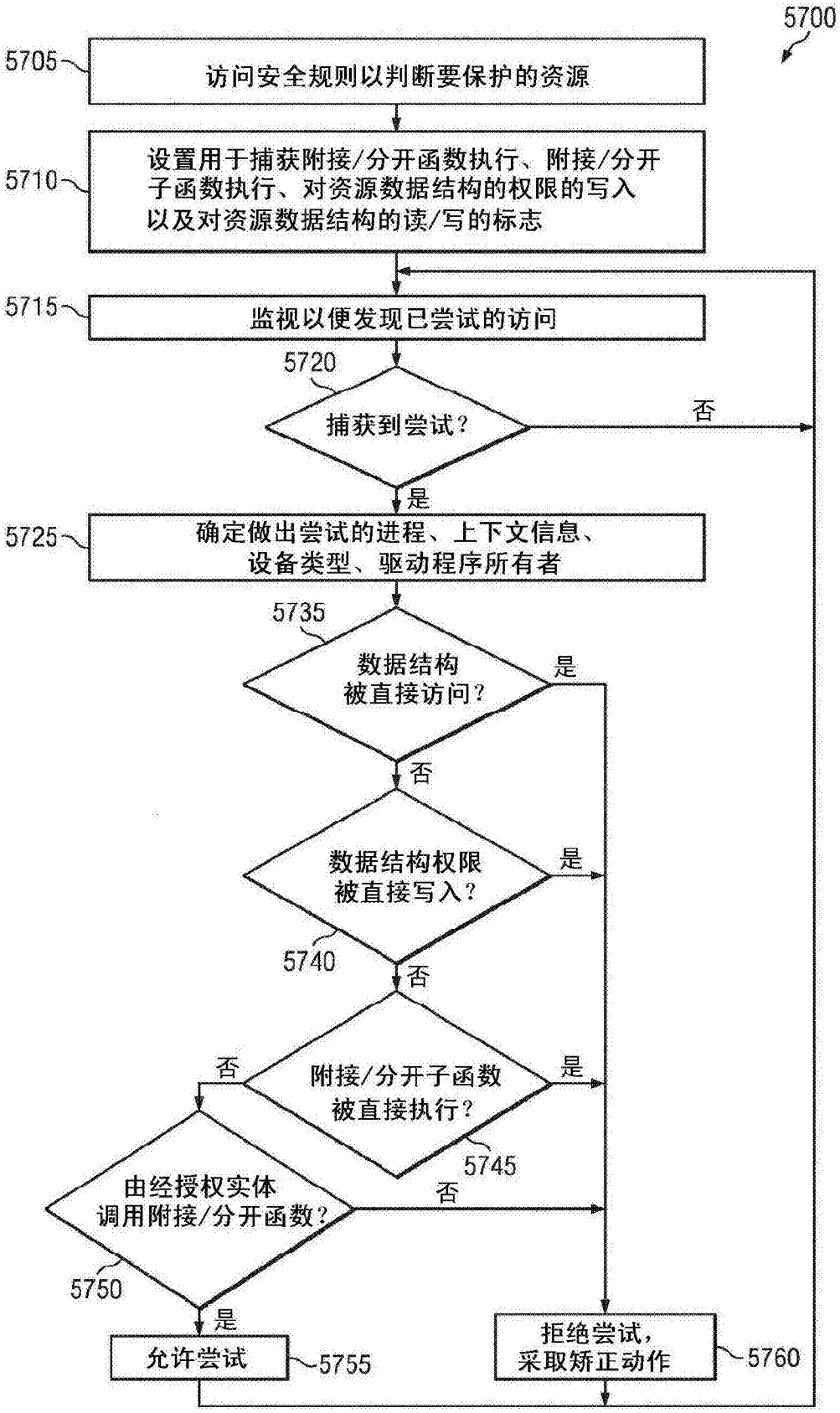

5700
5705
访问安全规则以判断要保护的资源
5710
设置用于捕获附接/分开函数执行、附接/分开子函数执行、对资源数据结构的权限的写入以及对资源数据结构的读/写的标志
5715
监视以便发现已尝试的访问
5720
捕获到尝试？
否
是
5725
确定做出尝试的进程、上下文信息、设备类型、驱动程序所有者
5735
数据结构被直接访问？
是
否
数据结构权限被直接写入？
是
5740
否
附接/分开子函数被直接执行？
否
是
5745
由经授权实体调用附接/分开函数？
否
5750
是
允许尝试
5755
拒绝尝试，采取矫正动作
5760

图57

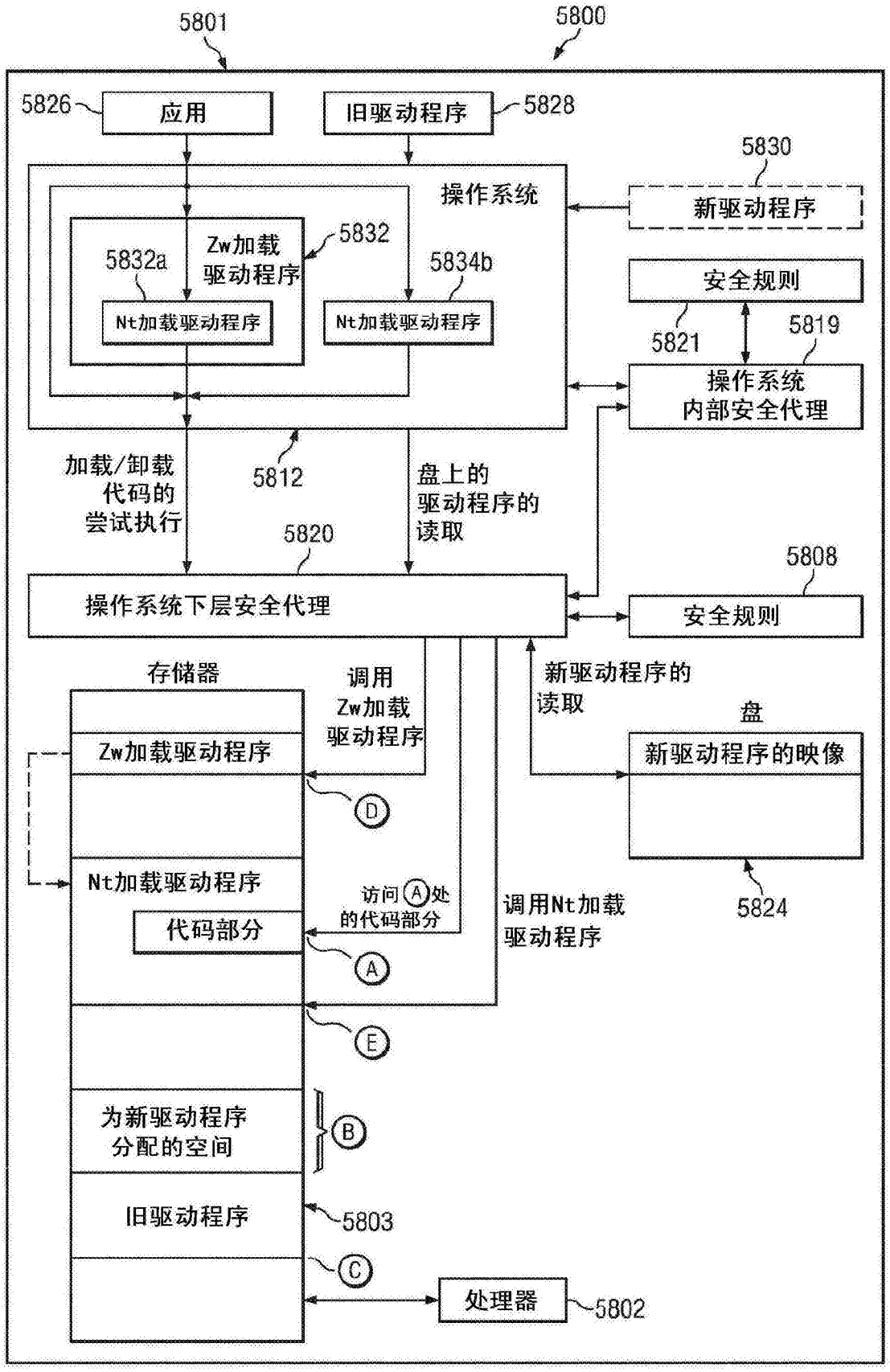

5801
5800
5826
应用
旧驱动程序
5828
5830
操作系统
新驱动程序
5832
Zw加载
驱动程序
5832a
5834b
安全规则
Nt加载驱动程序
Nt加载驱动程序
5821
5819
操作系统
内部安全代理
加载/卸载
代码的
尝试执行
5812
盘上的
驱动程序的
读取
5820
5808
操作系统下层安全代理
安全规则
存储器
调用
Zw加载
驱动程序
新驱动程序的
读取
盘
Zw加载驱动程序
新驱动程序的映像
D
Nt加载驱动程序
访问A处
的代码部分
调用Nt加载
驱动程序
代码部分
A
5824
E
为新驱动程序
分配的空间
B
旧驱动程序
5803
C
处理器
5802

图58

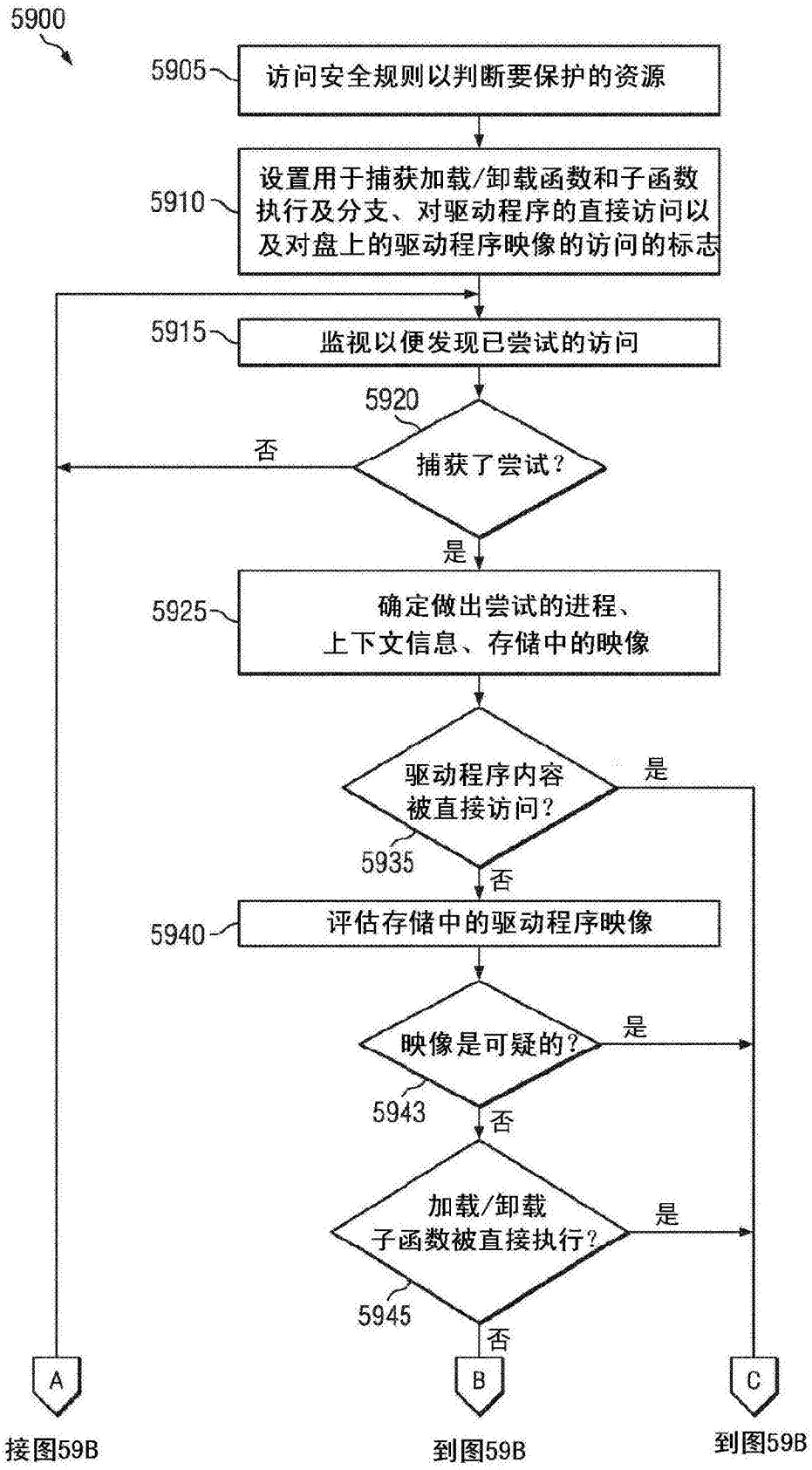

5900
5905
访问安全规则以判断要保护的资源
5910
设置用于捕获加载/卸载函数和子函数执行及分支、对驱动程序的直接访问以及对盘上的驱动程序映像的访问的标志
5915
监视以便发现已尝试的访问
5920
捕获了尝试?
否
是
5925
确定做出尝试的进程、上下文信息、存储中的映像
驱动程序内容被直接访问?
是
5935
否
5940
评估存储中的驱动程序映像
映像是可疑的?
是
5943
否
加载/卸载子函数被直接执行?
是
5945
否
A
B
C
接图59B
到图59B
到图59B

图59A

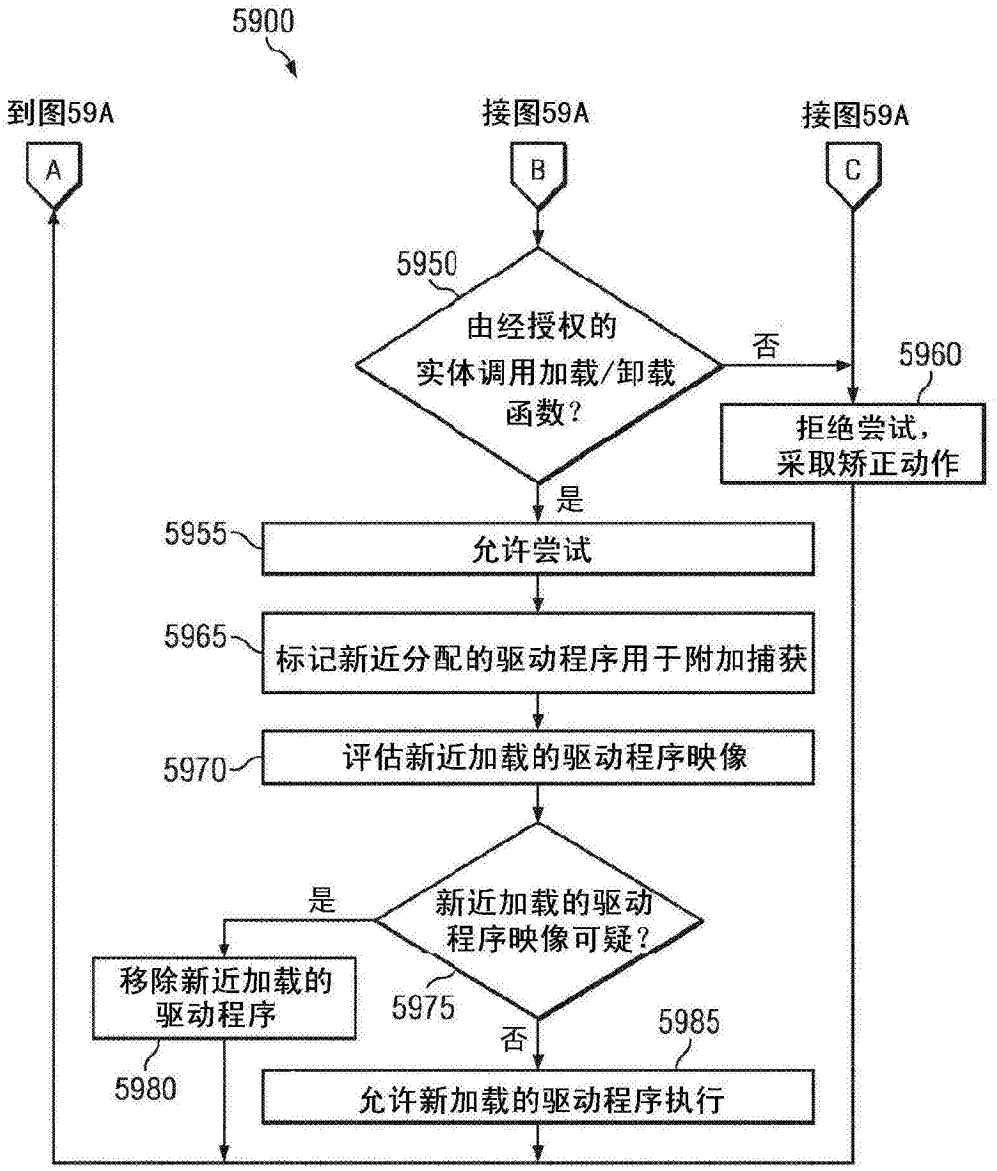

5900
到图59A
A
接图59A
B
接图59A
C
5950
由经授权的实体调用加载/卸载函数?
否
5960
拒绝尝试，采取矫正动作
是
5955
允许尝试
5965
标记新近分配的驱动程序用于附加捕获
5970
评估新近加载的驱动程序映像
是
新近加载的驱动程序映像可疑?
5975
移除新近加载的驱动程序
5980
否
5985
允许新加载的驱动程序执行

图59B

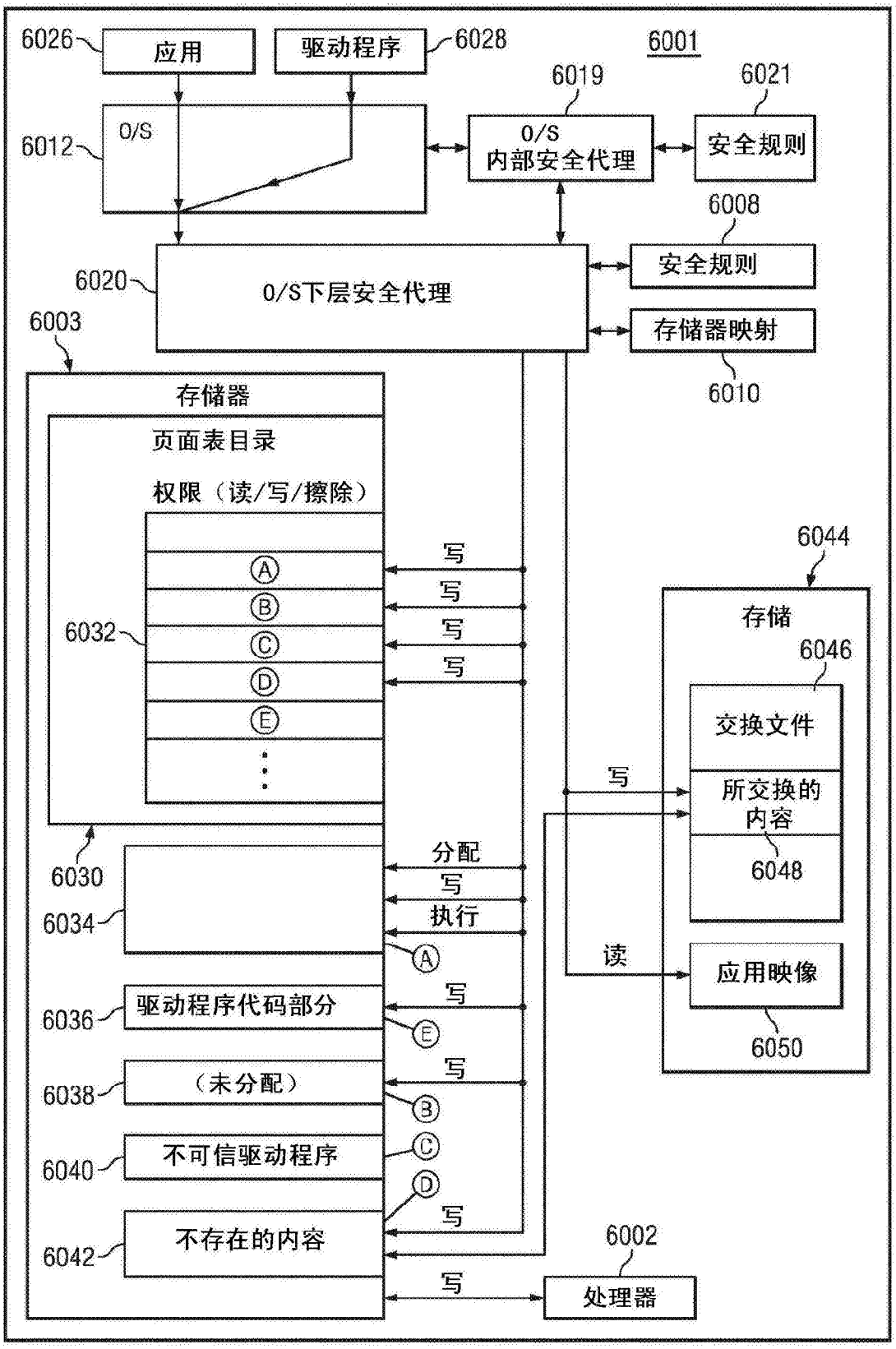

6026
应用
驱动程序
6028
6001
6019
6021
6012
O/S
O/S
内部安全代理
安全规则
6008
6020
O/S下层安全代理
安全规则
6003
存储器映射
6010
存储器
页面表目录
权限（读/写/擦除）
6044
写
A
B
C
D
E
6032
存储
6046
交换文件
写
所交换的
内容
6048
6030
分配
写
执行
6034
A
读
应用映像
写
6036
驱动程序代码部分
E
6050
写
6038
（未分配）
B
C
6040
不可信驱动程序
D
写
6042
不存在的内容
6002
写
处理器

图60

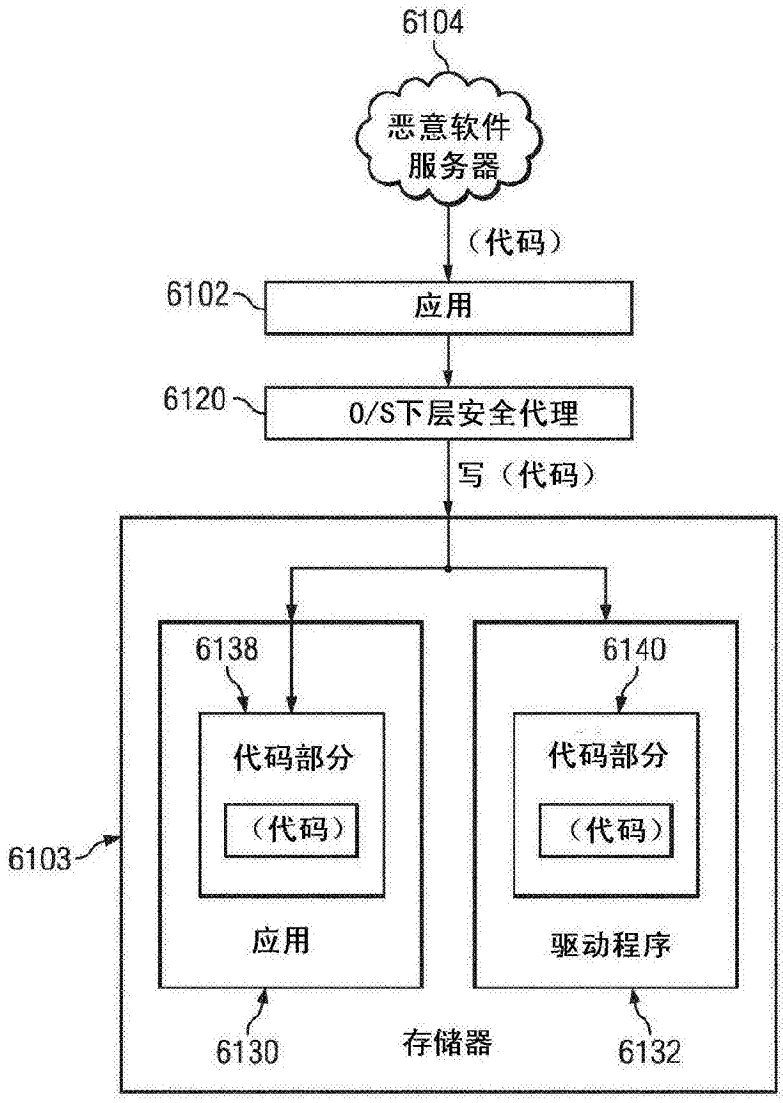

6104
恶意软件
服务器
（代码）
6102
应用
6120
0/S下层安全代理
写（代码）
6138
6140
代码部分
（代码）
代码部分
（代码）
6103
应用
驱动程序
6130
存储器
6132

图61

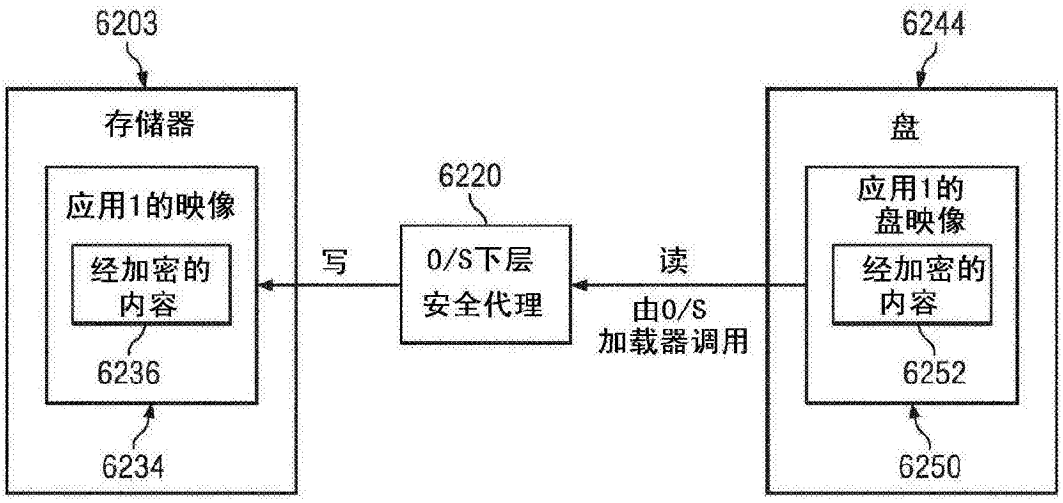

6203
6244
存储器
盘
6220
应用1的映像
应用1的
盘映像
写
0/S下层
安全代理
读
由0/S
加载器调用
经加密的
内容
经加密的
内容
6236
6252
6234
6250

图62A

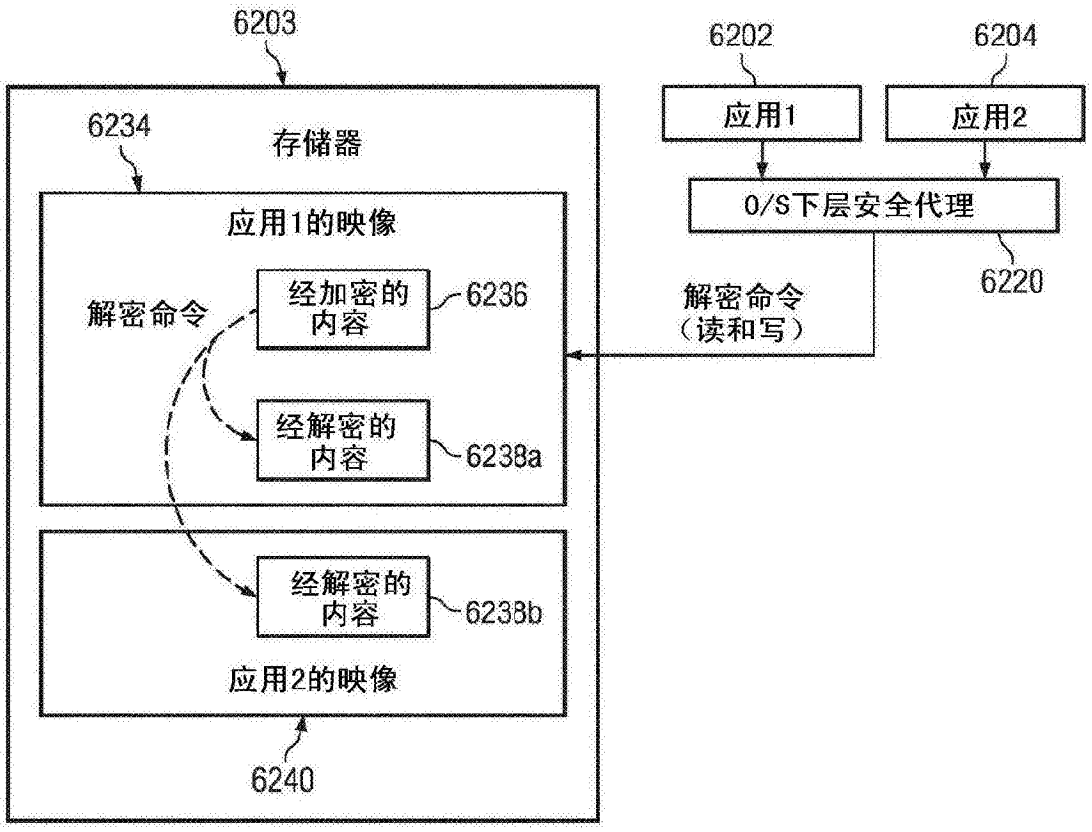

6203
6234
存储器
应用1的映像
解密命令
经加密的内容
6236
经解密的内容
6238a
经解密的内容
6238b
应用2的映像
6240
6202
应用1
6204
应用2
O/S下层安全代理
6220
解密命令（读和写）

图62B

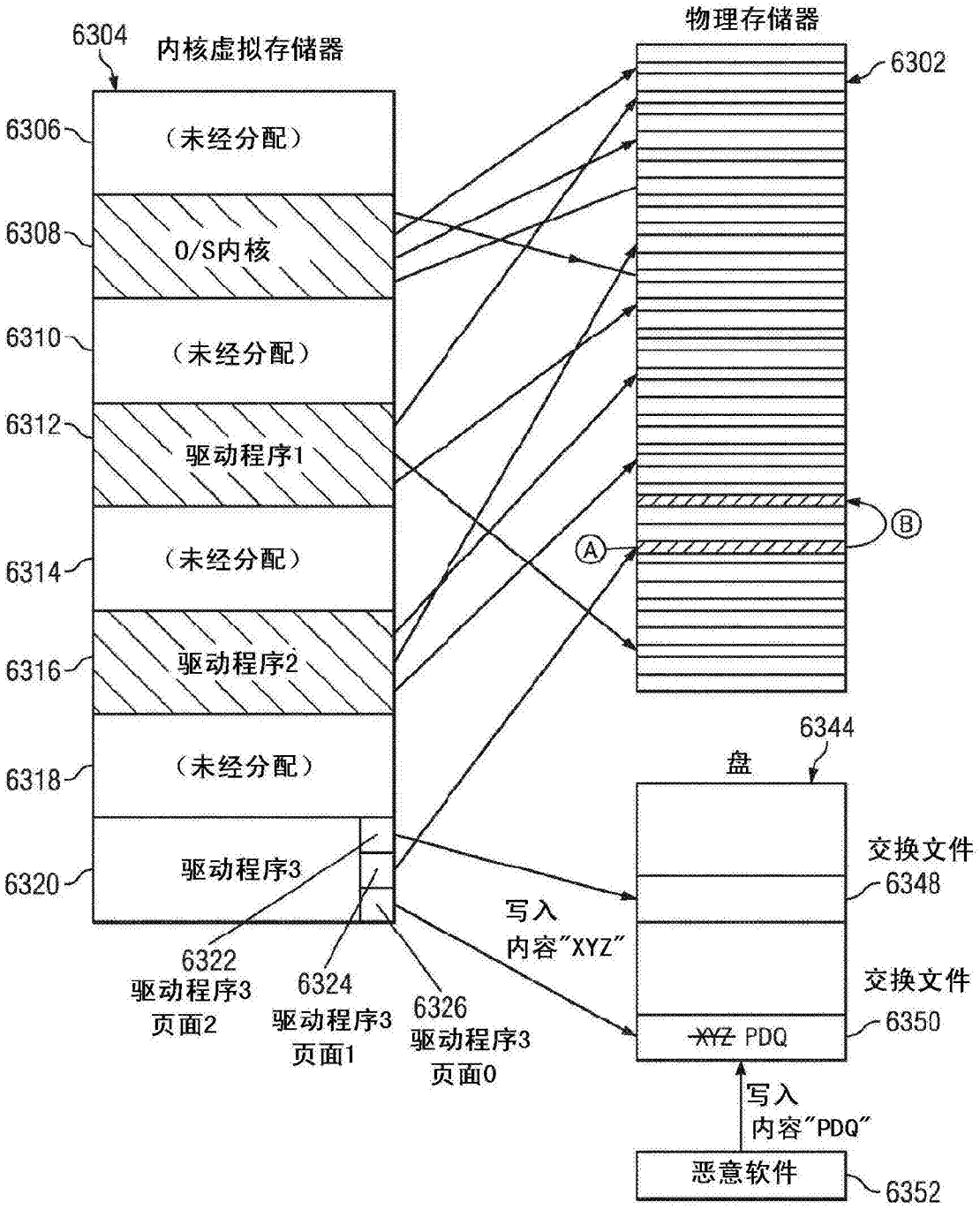

6304
内核虚拟存储器
物理存储器
6302
6306
（未经分配）
6308
O/S内核
6310
（未经分配）
6312
驱动程序1
6314
（未经分配）
6316
驱动程序2
6318
（未经分配）
6320
驱动程序3
A
B
6344
盘
交换文件
6348
写入
内容"XYZ"
交换文件
~~XYZ~~ PDQ
6350
写入
内容"PDQ"
恶意软件
6352
6322
驱动程序3
页面2
6324
驱动程序3
页面1
6326
驱动程序3
页面0

图63

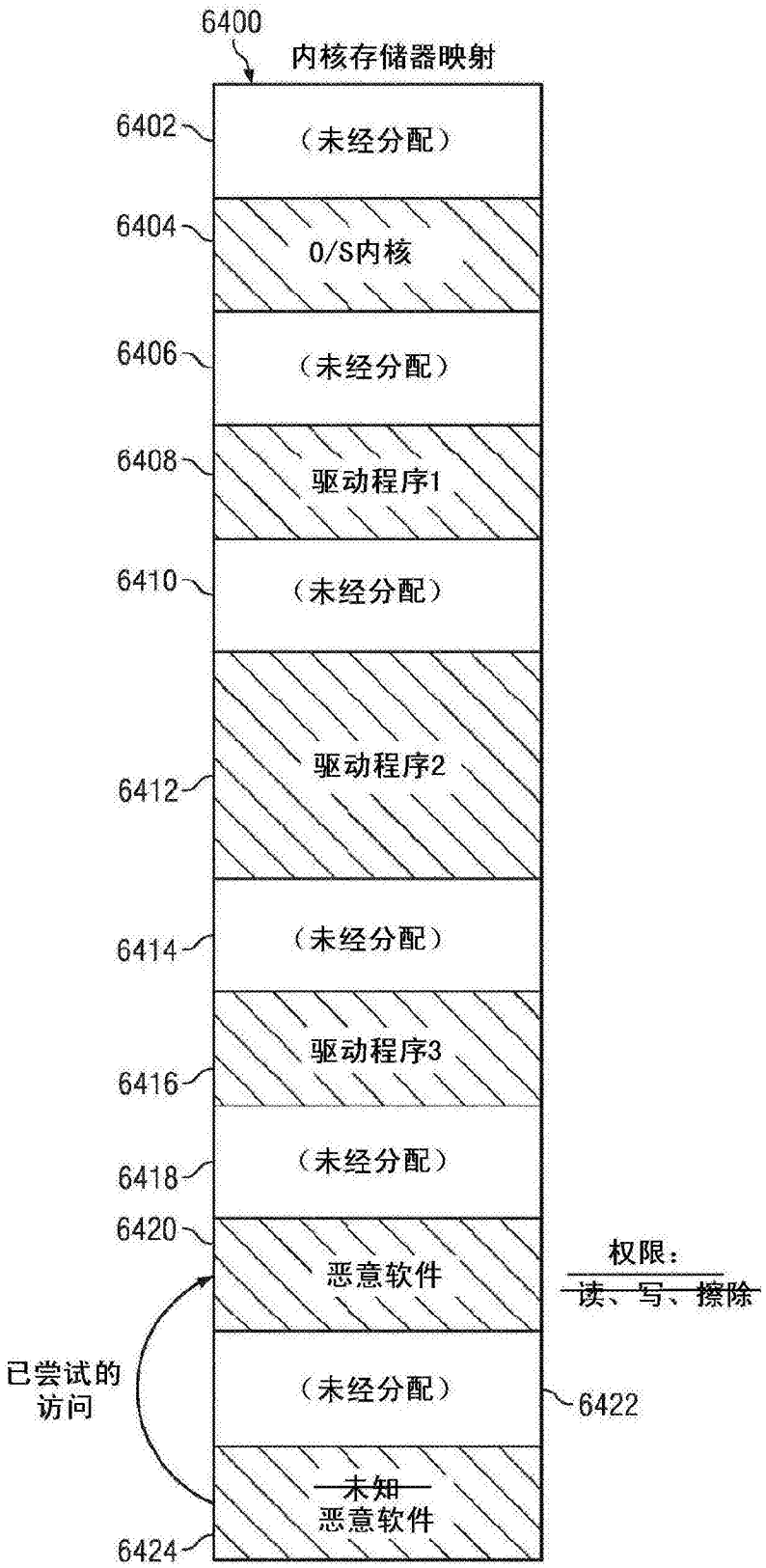

6400
内核存储器映射
6402
（未经分配）
6404
O/S内核
6406
（未经分配）
6408
驱动程序1
6410
（未经分配）
6412
驱动程序2
6414
（未经分配）
6416
驱动程序3
6418
（未经分配）
6420
恶意软件
权限：
~~读、写、擦除~~
已尝试的
访问
（未经分配）
6422
~~未知~~
恶意软件
6424

图64

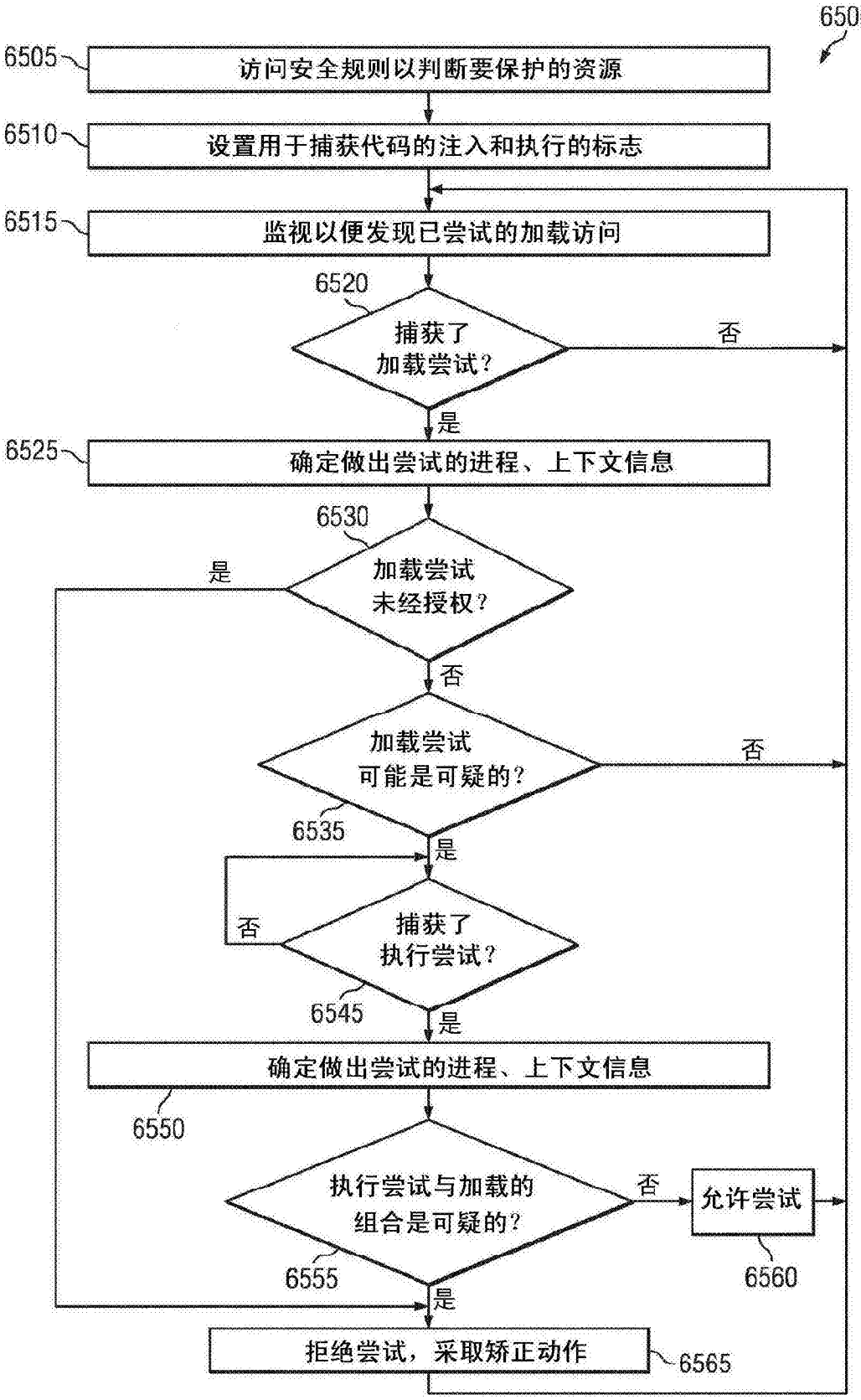

6500
6505
访问安全规则以判断要保护的资源
6510
设置用于捕获代码的注入和执行的标志
6515
监视以便发现已尝试的加载访问
6520
捕获了加载尝试？
否
是
6525
确定做出尝试的进程、上下文信息
6530
加载尝试未经授权？
是
否
加载尝试可能是可疑的？
否
6535
是
否
捕获了执行尝试？
6545
是
确定做出尝试的进程、上下文信息
6550
执行尝试与加载的组合是可疑的？
否
允许尝试
6560
6555
是
拒绝尝试，采取矫正动作
6565

图65