(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 107203410 B
(45)授权公告日 2020.02.14

(21)申请号 201710248219.7

(22)申请日 2017.04.14

(65)同一申请的已公布的文献号
申请公布号 CN 107203410 A

(43)申请公布日 2017.09.26

(73)专利权人 华中科技大学
地址 430074 湖北省武汉市洪山区珞喻路1037号

(72)发明人 金海 羌卫中 徐公平 邹德清

(74)专利代理机构 华中科技大学专利中心 42201
代理人 李智 曹葆青

(51)Int.Cl.
G06F 9/455(2006.01)

(56)对比文件
CN 102521547 A,2012.06.27,
CN 102147843 A,2011.08.10,
CN 106055385 A,2016.10.26,
US 2011093847 A1,2011.04.21,
CN 104021063 A,2014.09.03,
CN 102207866 A,2011.10.05,
吴锐.基于系统调用重定向的虚拟机自省技术.《中国优秀硕士学位论文全文数据库 信息科技辑》.2016,

审查员 安飞

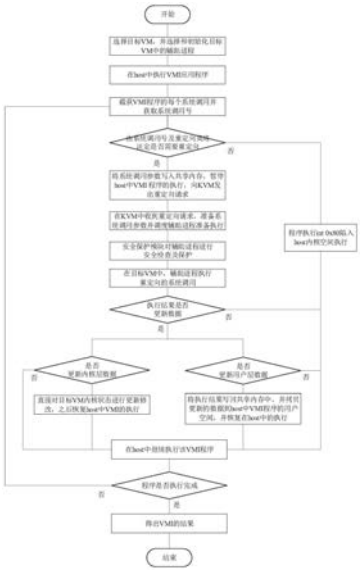
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种基于系统调用重定向的VMI方法及系统

(57)摘要

本发明公开了一种基于系统调用重定向的VMI方法及系统,通过目标VM选择为管理员提供运行中的VM的动态视角使得每个VM可以被监控;通过辅助进程生成为一个目标VM中的进程选择和初始化来执行重定向的系统调用;通过系统调用截获来截获VMI应用程序的每个系统调用,并决定该系统调用是否需要被重定向;通过重定向系统调用执行重定向一个系统调用到辅助进程中执行;通过安全保护确保辅助进程的安全执行,保证正确的自省结果。本发明能够用来监控云环境中的多个不同类型VM,实现了一种可写的VMI技术,能够从目标VM外面对该VM的内核状态进行修改,带来了一种高自动化的特点,能够被用于一种自动化的云管理中。



CN 107203410 B

1.一种基于系统调用重定向的VMI方法,其特征在于,包括:

(1)选择目标虚拟机VM,并将目标VM中的init进程选择为辅助进程;

(2)在host中运行虚拟机自省VMI应用程序,截获VMI应用程序的每个系统调用,获取系统调用号,依据系统调用号以及预设的系统调用重定向策略决定截获到的系统调用是否需要被重定向到目标VM中执行;

(3)若需要重定向,则将重定向的系统调用参数写入共享内存中,暂停host中VMI应用程序的执行,向内核虚拟机KVM发出重定向请求;

(4)KVM在接收到重定向请求后,从共享内存中读取系统调用参数,并对辅助进程进行安全检查及保护,然后调度辅助进程在目标VM中执行重定向的系统调用;

(5)若系统调用的执行结果对目标VM的用户层数据进行更新,则将执行结果重新写回到共享内存中,并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间;若系统调用的执行结果对目标VM的内核状态进行更新,则直接对目标VM的内存进行更新,然后恢复host中VMI应用程序的执行;

(6)若VMI应用程序执行结束,则获取目标VM中的监控结果;若VMI应用程序没有执行结束,则返回步骤(2);

在步骤(2)中,截获VMI应用程序的系统调用,获取系统调用号具体为:

利用共享库介入技术,修改glibc库文件中系统调用入口处的汇编代码,添加自定义的系统调用重定向决策函数,并重新编译glibc库文件,在VMI应用程序由用户空间进入到库空间开始执行系统调用时,自动执行系统调用重定向决策函数,截获VMI应用程序的系统调用,获取系统调用号。

2.根据权利要求1所述的方法,其特征在于,在步骤(2)中,依据系统调用号以及预设的系统调用重定向策略决定截获的系统调用是否需要重定向到目标VM中执行具体为:

将系统调用分为只读read-only和可写writable两类,并分别对每一个系统调用进行分类,分别对两类系统调用的重定向策略进行分析,其中与系统文件和socket读写相关的系统调用并且对此文件和socket后续的系统调用读写操作均需要重定向。

3.根据权利要求1所述的方法,其特征在于,在步骤(4)中,对辅助进程进行安全检查及保护具体为:

辅助进程的用户空间安全保护:利用KVM的EPT进程地址空间隔离将辅助进程所在的内存页的读写权限从EPT页表中移除,以使对辅助进程内存页的读写操作被拒绝,保证辅助进程的内存不会被恶意软件读写;

辅助进程的内核空间安全检查:在VM首次创建时,利用KVM获取内核静态函数的地址并保存,之后每次在辅助进程执行前,再一次的获取内核静态函数的地址,并与先前保存的地址做一致性对比,若二者不一致,则表明系统内核被rootkits破坏,利用KVM向VMI应用程序发出安全警告,并将不一致信息写入内核日志文件。

4.一种基于系统调用重定向的VMI系统,其特征在于,包括:

初始化模块,用于选择目标虚拟机VM,并将目标VM中的init进程选择为辅助进程;

系统调用截获及重定向决策模块,用于在host中截获VMI应用程序的每个系统调用,获取系统调用号,依据系统调用号以及预设的系统调用重定向策略决定截获到的系统调用是否需要重定向到目标VM中执行;系统调用截获及重定向决策模块截获VMI应用程序的系统

调用，获取系统调用号具体为：利用共享库介入技术，修改glibc库文件中系统调用入口处的汇编代码，添加自定义的系统调用重定向决策函数，并重新编译glibc库文件，在VMI应用程序由用户空间进入到库空间开始执行系统调用时，自动执行系统调用重定向决策函数，截获VMI应用程序的系统调用，获取系统调用号；

系统调用重定向模块，用于在需要重定向时，将重定向的系统调用参数写入共享内存中，暂停host中VMI应用程序的执行，向内核虚拟机KVM发出重定向请求；

重定向系统调用执行模块，用于在KVM接收到重定向请求后，从共享内存中读取系统调用参数，并调度辅助进程在目标VM中执行重定向的系统调用；

安全保护模块，用于在辅助进程执行系统调用前，检查保护系统调用的执行环境，保证可靠自省的结果；

重定向结果更新模块，用于在系统调用的执行结果对目标VM的用户层数据进行更新时，将执行结果重新写回到共享内存中，并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间，在系统调用的执行结果对目标VM的内核状态进行更新时，直接对目标VM的内存进行更新，然后恢复host中VMI应用程序的执行；

重定向结果获取模块，用于在VMI应用程序执行结束后，获取目标VM中的监控结果；若VMI应用程序没有执行结束，则返回所述系统调用截获及重定向决策模块。

一种基于系统调用重定向的VMI方法及系统

技术领域

[0001] 本发明属于云计算安全技术领域，更具体地，涉及一种基于系统调用重定向的VMI方法及系统。

背景技术

[0002] 随着云计算的快速发展，越来越多的人会开始租用虚拟机(Virtual Machine，VM)，VM背后的核心技术就是虚拟化。近年来，随着虚拟化的研究发展，导致虚拟机自省(Virtual Machine Introspection，VMI)技术及工具的演进。VMI指的是从VM外面监控VM内部运行的状态，实现入侵检测、恶意软件分析、完整性检查、日志审计等安全功能。

[0003] VMI的核心问题是语义鸿沟，即管理程序(hypervisor)能够看到的底层状态(二进制的byte或者bit)与它们在VM内部所表达的语义的分离(如进程PID)。到目前为止，已经有很多的VMI系统能够解决或避免语义鸿沟，如Xen Access库、内核数据重定向、进程及代码注入、系统调用重定向等。每种VMI系统解决语义鸿沟的方法都有自己的优劣，但是目前仍然没有一种VMI工具及技术能够直接应用到云计算环境，主要原因如下：

[0004] 首先，管理大量的VMs可能会导致不可避免的管理成本，这就需要一个自动的方式来管理这些VMs。最近一种可写的VMI技术已经被提出从VM外面修改VM的内核状态而不需要任何管理员的介入，此种方式极大的减少了管理成本。另外，可写的VMI更能够自发地响应目标VM，例如当检测到目标VM的一个隐藏进程后，可写VMI技术能够自发的从目标VM外面kill掉这个隐藏进程而不需任何的人工努力。因此，一个先进的VMI技术应该提供一个可写的能力来自动地管理云环境中的VMs和提高云安全。

[0005] 其次，云平台要同时为海量云用户提供服务，而各个云用户需要的系统环境可能是千差万别的，故针对云平台来说，VM的数量是庞大的，而且每个VM的客户操作系统的版本可能是不同的，这在一定程度上增加了监控的复杂性及难度，因此需要一种监控工具能够同时对云平台中大量VM同时监控，而且能够兼容每个VM的操作系统版本。

[0006] 最后，目前存在的大多数的VMI技术都带来了极大的性能开销及资源消耗。因此一个实用的VMI技术应该带来更低的性能开销和资源的消耗。

[0007] 综上所述，目前的VMI工具及技术难以全面适应到云计算环境的需求，一个适合云环境的VMI技术应该提供可写、低性能开销及监控多台不同操作系统的VMs的能力。

发明内容

[0008] 针对现有技术的以上缺陷或改进需求，本发明提供了一种基于系统调用重定向的VMI方法及系统，能够从目标VM外面对该VM的内核状态进行修改，而不需要任何的用户权限，极大的减少了VM的管理成本，而且能够用来主动地提高云环境安全，解决现有的VMI系统难以直接应用到云环境，无法同时满足云环境的一些现实需求：可写性、高效性、通用性、同时监控、可靠性等的技术问题。

[0009] 为实现上述目的，按照本发明的一个方面，提供了一种基于系统调用重定向的VMI

方法,包括:

[0010] (1)选择目标虚拟机VM,并将目标VM中的init进程选择为辅助进程;

[0011] (2)在host中运行虚拟机自省VMI应用程序,截获VMI应用程序的每个系统调用,获取系统调用号,依据系统调用号以及预设的系统调用重定向策略决定截获到的系统调用是否需要被重定向到目标VM中执行;

[0012] (3)若需要重定向,则将重定向的系统调用参数写入共享内存中,暂停host中VMI应用程序的执行,向内核虚拟机KVM发出重定向请求;

[0013] (4)KVM在接收到重定向请求后,从共享内存中读取系统调用参数,并对辅助进程进行安全检查及保护,然后调度辅助进程在目标VM中执行重定向的系统调用。

[0014] (5)若系统调用的执行结果对目标VM的用户层数据进行更新,则将执行结果重新写回到共享内存中,并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间;若系统调用的执行结果对目标VM的内核状态进行更新,则直接对目标VM的内存进行更新,然后恢复host中VMI应用程序的执行;

[0015] (6)若VMI应用程序执行结束,则获取目标VM中的监控结果;若VMI应用程序没有执行结束,则返回步骤(2)。

[0016] 优选地,在步骤(2)中,截获VMI应用程序的系统调用,获取系统调用号具体为:

[0017] 利用共享库介入技术,修改glibc库文件中系统调用入口处的汇编代码,添加自定义的系统调用重定向决策函数,并重新编译glibc库文件,在VMI应用程序由用户空间进入到库空间开始执行系统调用时,自动执行系统调用重定向决策函数,截获VMI应用程序的系统调用,获取系统调用号。

[0018] 优选地,在步骤(2)中,依据系统调用号以及预设的系统调用重定向策略决定截获的系统调用是否需要重定向到目标VM中执行具体为:

[0019] 将系统调用分为只读read-only和可写writable两类,并分别对每一个系统调用进行分类,分别对两类系统调用的重定向策略进行分析,其中与系统文件和socket读写相关的系统调用并且对此文件和socket后续的系统调用读写操作均需要重定向。

[0020] 优选地,在步骤(4)中,对辅助进程进行安全检查及保护具体为:

[0021] 辅助进程的用户空间安全保护:利用KVM的EPT进程地址空间隔离将辅助进程所在的内存页的读写权限从EPT页表中移除,以使对辅助进程内存页的读写操作被拒绝,保证辅助进程的内存不会被恶意软件读写;

[0022] 辅助进程的内核空间安全检查:在VM首次创建时,利用KVM获取内核静态函数的地址并保存,之后每次在辅助进程执行前,再一次的获取内核静态函数的地址,并与先前保存的地址做一致性对比,若二者不一致,则表明系统内核被rootkits破坏,利用KVM向VMI应用程序发出安全警告,并将不一致信息写入内核日志文件。

[0023] 按照本发明的另一方面,提供了一种基于系统调用重定向的VMI系统,包括:

[0024] 初始化模块,用于选择目标虚拟机VM,并将目标VM中的init进程选择为辅助进程;

[0025] 系统调用截获及重定向决策模块,用于在host中截获VMI应用程序的每个系统调用,获取系统调用号,依据系统调用号以及预设的系统调用重定向策略决定截获到的系统调用是否需要重定向到目标VM中执行;

[0026] 系统调用重定向模块,用于在需要重定向时,将重定向的系统调用参数写入共享

内存中，暂停host中VMI应用程序的执行，向内核虚拟机KVM发出重定向请求；

[0027] 重定向系统调用执行模块，用于在KVM接收到重定向请求后，从共享内存中读取系统调用参数，并调度辅助进程在目标VM中执行重定向的系统调用；

[0028] 安全保护模块，用于在辅助进程执行系统调用前，检查保护系统调用的执行环境，保证可靠自省的结果；

[0029] 重定向结果更新模块，用于在系统调用的执行结果对目标VM的用户层数据进行更新时，将执行结果重新写回到共享内存中，并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间，在系统调用的执行结果对目标VM的内核状态进行更新时，直接对目标VM的内存进行更新，然后恢复host中VMI应用程序的执行；

[0030] 重定向结果获取模块，用于在VMI应用程序执行结束后，获取目标VM中的监控结果；若VMI应用程序没有执行结束，则返回所述系统调用截获及重定向决策模块。

[0031] 总体而言，通过本发明所构思的以上技术方案与现有技术相比，主要有以下的技术优点：

[0032] (1)高效性。本发明基于系统调用重定向技术，且基于硬件虚拟化技术KVM实现，与现有的基于内核数据重定向及软件虚拟化技术QEMU相比，性能开销低，效率高，对客户VM的性能影响低。

[0033] (2)可写性。本发明详细设计了系统调用重定向策略，实现了一种可写VMI，不仅仅能够从VM外面获取目标VM的内存状态，而且能够从VM外面对目标VM的内核状态进行修改，不需要任何人工的操作，提高了VMI程序的自发性，能够用来主动地提高云环境安全。

[0034] (3)通用性。由于本发明基于系统调用重定向，系统调用在linux的不同发型版中除非存在系统调用接口随机化，否则几乎所有的系统调用的接口都是兼容的，这就保证了一个VMI程序能够兼容不同的guest OS。

[0035] (4)可靠性。本发明与现有的基于系统调用重定向的方法相比，针对辅助进程做了相关的安全保护策略，保护了重定向系统调用执行环境的安全，从而保证了自省结果的可靠性。

附图说明

[0036] 图1是本发明实施例公开的一种基于系统调用重定向的VMI方法的流程示意图；

[0037] 图2是本发明实施例公开的一种基于系统调用重定向的VMI的系统架构图；

[0038] 图3是本发明实施例公开的一种基于host截获系统调用的原理图。

具体实施方式

[0039] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。此外，下面所描述的本发明各个实施方式中所涉及到的技术特征只要彼此之间未构成冲突就可以相互组合。

[0040] 如图1所示为本发明实施例公开的一种基于系统调用重定向的VMI方法的流程示意图，在图1所示的方法中，包括以下步骤：

[0041] (1)选择目标虚拟机VM，并将目标VM中的init进程选择为辅助进程；

[0042] 其中,可以通过向目标VM中注入一个getpid系统调用来选择init进程作为执行重定向系统调用的辅助进程。

[0043] (2)在host中运行VMI应用程序,截获VMI应用程序的每个系统调用,获取系统调用号,依据系统调用号以及预设的系统调用重定向策略决定截获到的系统调用是否需要被重定向到目标VM中执行;

[0044] 其中,VMI应用程序可以包括ps,lsmod,iostat等现有的系统管理命令或者用户自定义的应用程序,可以利用动态库介入技术在host的库空间中截获VMI应用程序的每个系统调用,获取系统调用号。

[0045] (3)若需要重定向,则将重定向的系统调用参数写入共享内存中,暂停host中VMI应用程序的执行,向内核虚拟机(Kernel-based Virtual Machine,KVM)发出重定向请求;

[0046] 其中,若不需要重定向,则VMI应用程序继续执行系统调用入口指令陷入host内核空间中执行。

[0047] (4)KVM在接收到重定向请求后,从共享内存中读取系统调用参数,并对辅助进程进行安全检查及保护,然后调度辅助进程在目标VM中执行重定向的系统调用。

[0048] 其中,在调度辅助进程在目标VM中执行重定向的系统调用之前,需要对目标VM的控制流进行完整性检查,确保辅助进程的安全执行。

[0049] (5)若系统调用的执行结果对目标VM的用户层数据进行更新,则将执行结果重新写回到共享内存中,并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间,若系统调用的执行结果对目标VM的内核状态进行更新,则直接对目标VM的内存进行更新,然后恢复host中VMI应用程序的执行;

[0050] (6)若VMI应用程序执行结束,则获取目标VM中的监控结果;若VMI应用程序没有执行结束,则返回步骤(2)。

[0051] 作为一种可选的实施方式,在步骤(2)中,截获VMI应用程序的系统调用,获取系统调用号具体为:

[0052] 利用共享库介入技术,修改glibc库文件中系统调用入口处的汇编代码,添加自定义的系统调用重定向决策函数,并重新编译glibc库文件,在VMI应用程序由用户空间进入到库空间开始执行系统调用时,自动执行系统调用重定向决策函数,截获VMI应用程序的系统调用,获取系统调用号。

[0053] 作为一种可选的实施方式,在步骤(2)中,依据系统调用号以及预设的系统调用重定向策略决定截获的系统调用是否需要重定向到目标VM中执行具体为:

[0054] 将系统调用分为只读read-only和可写writable两类,并分别对每一个系统调用进行分类,如read、getdents64、getpid等属于read-only,而write、kill、nice等则属于writable类,分别对两类系统调用的重定向策略进行分析,其中与系统文件和socket读写相关的系统调用并且对此文件和socket后续的系统调用读写操作均需要重定向,例如open一个文件返回0后,之后对文件0操作的系统调用都要重定向,如read(0,,),write(0,,)等。

[0055] 作为一种可选的实施方式,在步骤(4)中,对辅助进程进行安全检查及保护具体包括:

[0056] 辅助进程的用户空间安全保护:辅助进程init在用户空间主要面临着恶意软件的威胁,恶意软件可能会破坏init进程的代码、干扰init进程读写的共享内存、恶意改变进程

的控制流。为了保证init进程在用户空间的代码、数据、控制流不会被恶意软件破坏，利用KVM的扩展页表(Extended Page Tables,EPT)进程地址空间隔离技术来保证init进程的内存不会被恶意软件读写。即使恶意软件具有客户VM系统的根权限，也不能访问KVM上的EPT。

[0057] EPT进程地址空间隔离主要是基于KVM，把init进程所在的内存页的写权限从EPT页表中移除。之后对init进程内存页写操作都会被拒绝，同样地，恶意软件也就不能对init进程做任何的操作，从而保证进程的代码及控制流的完整性。另外由于init进程读写的共享内存地址是随机生成的，恶意软件是不能够定位和修改的，故init进程的读写的数据也得到了完整性保护。

[0058] 辅助进程的内核空间安全检查:辅助进程主要用来在目标VM中执行重定向的系统调用，其主要依靠目标VM中的内核函数指针，例如中断描述符表(Interrupt Descriptor Table,IDT)、系统调用表及系统调用处理函数。通常地，绝大多数的内核rootkits通过hook这些IDT、系统调用表和函数指针来破坏内核的控制流完整性，从而实现一些恶意的目的，如隐藏恶意进程、隐藏恶意模块等。因此，这些内核rootkits将会直接影响辅助进程在目标VM中的正常执行，从而获得不正确的自省结果。

[0059] 然而，这些内核rootkits仍然会执行原始的控制流来隐藏自身的存在，因此内核rootkits不会使辅助进程中断执行。另外，IDT、系统调用表及函数指针都是静态的，在OS执行的过程中不会发生变化除非被恶意的改变。因此，我们提出了一种完整性检查机制来比对这些静态内核函数的地址是否一致。首先，在VM首次创建时，利用KVM获取这些内核静态函数的地址并保存。之后每次在辅助进程执行前，再一次的获取这些数据的地址，并与先前保存地址的做一致性对比。如果发现二者不一致，那么就能知道系统内核已经被rootkits破坏，利用KVM向VMI程序发出一个安全警告，并且将具体的不一致信息写入内核日志文件。当VMI程序完成执行后，如果收到一个安全警告信息时，管理员将会知道此次的自省结果是不正确的，另外KVM将会利用先前保存的数据自动地恢复内核控制流的完整性，从而保证了后续自省程序的正确执行。

[0060] 本发明还提供了一种基于系统调用重定向的VMI系统，包括：

[0061] 初始化模块，用于选择目标虚拟机VM，并将目标VM中的init进程确定为辅助进程；

[0062] 系统调用截获及重定向决策模块，用于在host中运行虚拟机自省VMI应用程序，截获VMI应用程序的每个系统调用，获取系统调用号，依据系统调用号以及预设的系统调用重定向策略决定截获的系统调用是否需要重定向到目标VM中执行；

[0063] 系统调用重定向模块，用于在需要重定向时，将重定向的系统调用参数写入共享内存中，暂停host中VMI应用程序的执行，向内核虚拟机KVM发出重定向请求；

[0064] 重定向系统调用执行模块，用于在KVM接收到重定向请求后，从共享内存中读取系统调用参数，并调度辅助进程在目标VM中执行重定向的系统调用。

[0065] 安全保护模块，在辅助进程执行系统调用前，检查保护系统调用的执行环境，保证可靠自省的结果。

[0066] 重定向结果更新模块，用于在系统调用的执行结果对目标VM的用户层数据进行更新时，将执行结果重新写回到共享内存中，并将共享内存中更新的数据拷贝到host中VMI应用程序的用户空间，在系统调用的执行结果对目标VM的内核状态进行更新时，直接对目标VM的内核状态进行更新修改，然后恢复host中VMI应用程序的执行；

[0067] 重定向结果获取模块，用于在VMI应用程序执行结束后，获取目标VM中的监控结果；若VMI应用程序没有执行结束，则返回所述系统调用截获及重定向决策模块。

[0068] 下面以具体的实例ps进程为例，说明本发明中的基于系统调用重定向的VMI方法的实现过程：

[0069] (1)系统管理员首先确定需要被监控的目标VM，之后在目标VM中选择并初始化init进程作为辅助进程；

[0070] (2)在host中执行ps命令，当程序需要执行系统调用时会陷入到用户库空间；

[0071] (3)在库空间中截获ps的所有系统调用，如open、read、close等，并且根据自定义的系统调用重定向策略决定该系统调用是否系统被重定向。如果需要被重定向则执行(4)，否则直接执行int 0x80指令陷入到host的内核中执行；

[0072] (4)将需要重定向的系统调用的参数写入到共享buffer中，暂停host中该程序的执行并通知KVM重定向该系统调用；

[0073] (5)KVM根据目标VM对应的信号量与host中的ps进程建立进程间通信，同时激活辅助进程；

[0074] (6)安全防护模块利用EPT保护辅助进程的用户层代码、数据；利用一致性检测机制对目标VM中的IDT表及系统调用表进行完整性检查和一致性恢复；

[0075] (7)辅助进程读取共享buffer中的数据，执行int 0x80或sysenter进入被监控目标VM的内核空间，执行重定向的系统调用；

[0076] (8)当重定向的系统调用执行完成后，根据其执行结果做出不同的处理。如果系统调用对系统的数据有更新，那么执行步骤(9)；否则，执行步骤(10)；

[0077] (9)如果系统调用执行结果对系统的用户层进行更新，那么将结果重新写回到共享内存中。然后将共享内存中更新的数据拷贝到host中ps进程的用户空间，并恢复在host中的执行。如果系统调用执行结果需要对目标VM的内核状态进行更新，那么这些更新直接修改目标VM的内核状态，执行完成后恢复在host中的ps执行；

[0078] (10)继续在host中执行ps程序，如果程序执行结束，那么得出目标VM中的进程列表；否则继续重复(3)-(9)的执行步骤。

[0079] 如图2所示为本发明基于系统调用重定向的高效可写VMI的系统架构图，主要包括系统调用截获及重定向决策、重定向系统调用执行、安全保护。系统调用截获及重定向决策：利用动态库介入技术完成基于host的系统调用截获，制定系统调用重定向的策略，依据截获的系统调用号和重定向策略决定该系统调用是否需要被重定向。重定向系统调用执行：基于KVM和VMI程序选择目标VM，并利用映射表与VMI程序建立通信，并将init进程由守护状态变为执行状态，读取共享buffer的数据来执行重定向来的系统调用。安全保护：在每个系统调用执行前，分别从用户空间和内核空间对init进程做安全检查及保护，从而保证系统调用执行结果的可靠性。

[0080] 如图3所示为本发明基于host截获系统调用并进行重定向选择的原理图。当VMI程序执行系统调用时，会首先进入用户的glibc库空间，通过系统调用处理的汇编代码ENTRY(syscall)找到进入系统内核入口的地址ENTER_KERNEL。传统的ENTER_KERNEL宏定义为int 0x80，因此只需要将ENTER_KERNEL宏的定义int 0x80改为自定义的系统调用重定向决策函数的地址，那么当VMI执行系统调用时，就会执行系统调用的决策函数，而不会直接执行int

0x80进入到host内核空间了，完成了系统调用的截获及重定向策略的执行。

[0081] 本领域的技术人员容易理解，以上所述仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

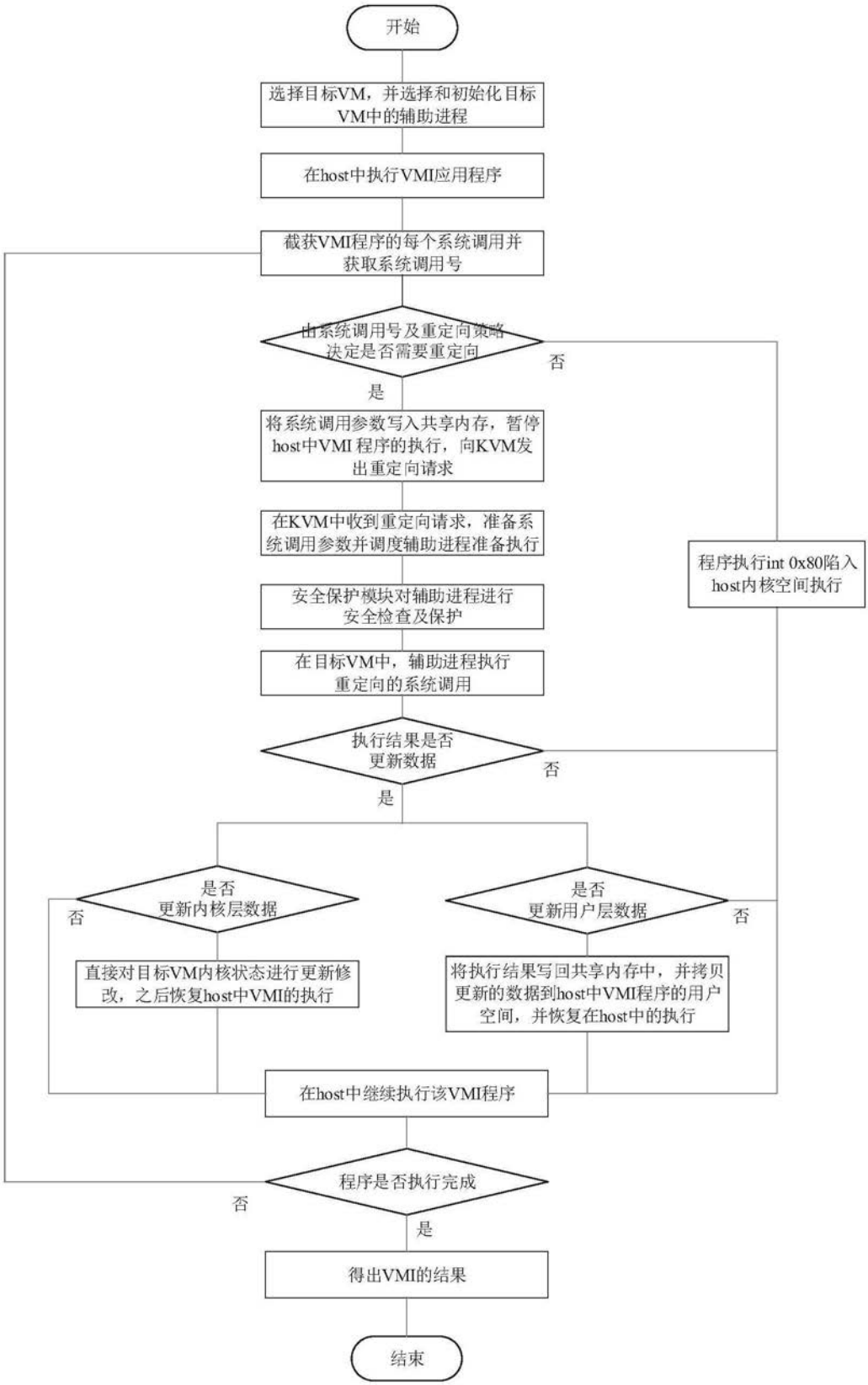

开始
选择目标VM，并选择和初始化目标VM中的辅助进程
在host中执行VMI应用程序
截获VMI程序的每个系统调用并获取系统调用号
由系统调用号及重定向策略决定是否需要重定向
否
是
将系统调用参数写入共享内存，暂停host中VMI 程序的执行，向KVM发出重定向请求
在KVM中收到重定向请求，准备系统调用参数并调度辅助进程准备执行
安全保护模块对辅助进程进行安全检查及保护
在目标VM中，辅助进程执行重定向的系统调用
执行结果是否更新数据
否
是
程序执行int 0x80陷入host内核空间执行
是否更新内核层数据
否
是否更新用户层数据
否
直接对目标VM内核状态进行更新修改，之后恢复host中VMI的执行
将执行结果写回共享内存中，并拷贝更新的数据到host中VMI程序的用户空间，并恢复在host中的执行
在host中继续执行该VMI程序
程序是否执行完成
否
是
得出VMI的结果
结束

图1

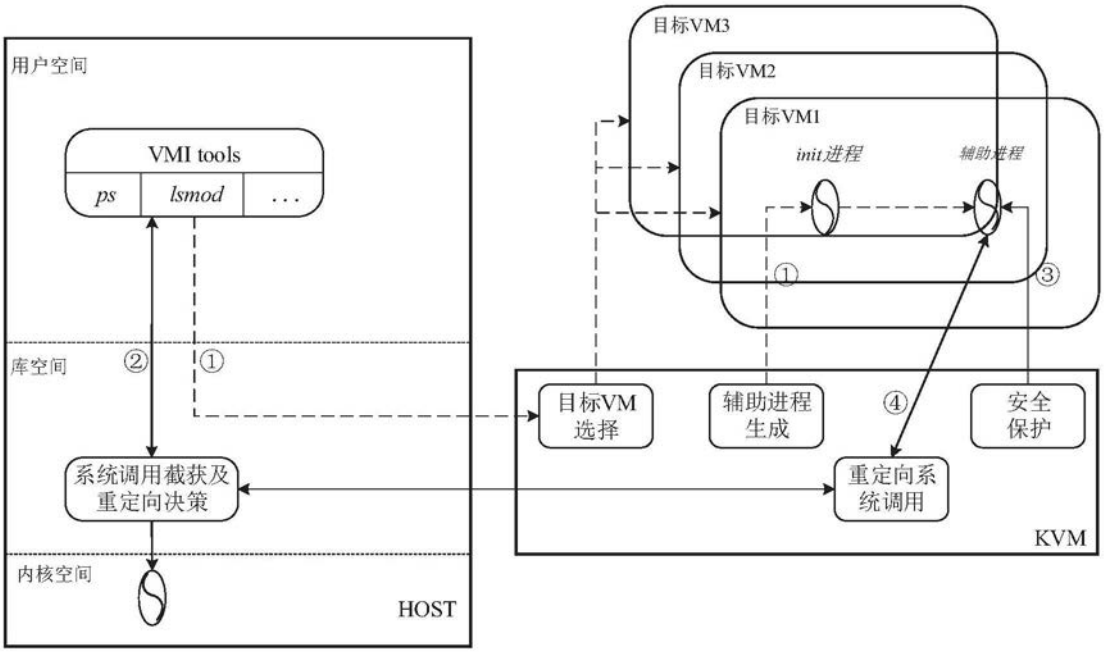

用户空间
VMI tools
ps
lsmod
...
库空间
②
①
系统调用截获及
重定向决策
内核空间
HOST
目标VM3
目标VM2
目标VM1
init进程
辅助进程
①
③
目标VM
选择
辅助进程
生成
④
安全
保护
重定向系
统调用
KVM

图2

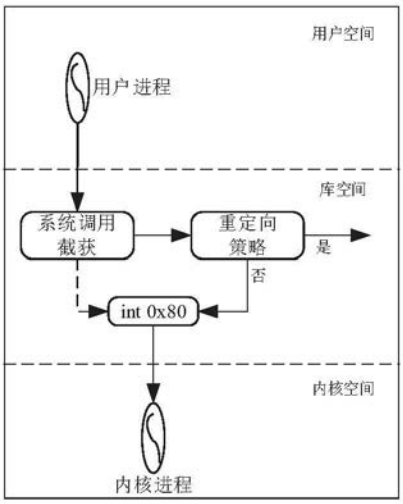

用户空间
用户进程
库空间
系统调用
截获
重定向
策略
是
否
int 0x80
内核空间
内核进程

图3