(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 116302850 B
(45)授权公告日 2023.09.12

(21)申请号 202310571158.3

(22)申请日 2023.05.20

(65)同一申请的已公布的文献号
申请公布号 CN 116302850 A

(43)申请公布日 2023.06.23

(73)专利权人 北京长亭科技有限公司
地址 100083 北京市海淀区学院路甲5号1幢三层1#厂房西区2-007

(72)发明人 罗浩然 朱文雷 李昌志 刘超 刘金钊 赖博阳 主洪尚

(74)专利代理机构 深圳睿臻知识产权代理事务所(普通合伙) 44684
专利代理师 李磊

(51)Int.Cl.
G06F 11/30(2006.01)

(56)对比文件
US 2008126828 A1,2008.05.29
US 2023052452 A1,2023.02.16
CN 104268019 A,2015.01.07
CN 115776451 A,2023.03.10
CN 108563493 A,2018.09.21
CN 116107846 A,2023.05.12
陶海鹏;王勇;俸皓.云环境下的Linux进程监控的设计与实现.微电子学与计算机.2017,(第07期),第143-146页.

审查员 魏丁雯

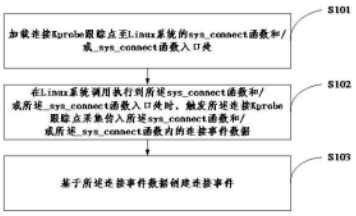
权利要求书2页 说明书8页 附图2页

(54)发明名称

一种Linux套接字连接事件监控方法及装置

(57)摘要

本申请实施例提供了一种Linux套接字连接事件监控方法及装置,通过加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;以在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;然后基于所述连接事件数据创建连接事件,能够解决现有套接字连接事件监控存在的占用运行资源多、无法监控容器内进程产生的套接字连接事件的问题,实现了能够监控容器内运行的进程产生的套接字连接事件、并且占用运行资源少,提高了套接字连接事件监控的可靠性。



CN 116302850 B

1.一种Linux套接字连接事件监控方法,其特征在于,包括:

加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;

在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;

基于所述连接事件数据创建连接事件;

加载协议Kprobe跟踪点至Linux系统的inet_stream_connect函数和/或inet_dgram_connect函数入口处;

在Linux系统调用执行到所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处时,触发所述协议Kprobe跟踪点采集传入所述inet_stream_connect函数和/或所述inet_dgram_connect函数内的套接字协议类型;

将所述套接字协议类型填充至所述连接事件内,得到协议连接事件;

加载状态Kprobe跟踪点至Linux系统的所述sys_connect函数和/或所述_sys_connect函数出口处;

在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数出口处时,触发所述状态Kprobe跟踪点采集执行所述sys_connect函数和/或所述_sys_connect函数后的返回值;

将所述返回值填充至所述协议连接事件内,得到状态连接事件。

2.根据权利要求1所述的Linux套接字连接事件监控方法,其特征在于,所述连接事件数据包括IP地址、端口号、文件描述符、进程描述符、事件时间戳及IP地址族中的至少一项。

3.根据权利要求2所述的Linux套接字连接事件监控方法,其特征在于,所述基于所述连接事件数据创建连接事件,包括:

设置地址族过滤条件和地址长度过滤条件;

基于所述地址族过滤条件及所述地址长度过滤条件从所述连接事件数据中过滤出包含ipv4地址和/或ipv6地址的地址事件数据;

基于所述地址事件数据创建所述连接事件。

4.根据权利要求3所述的Linux套接字连接事件监控方法,其特征在于,还包括:

将所述事件时间戳设置为所述状态连接事件的时间戳,得到套接字连接事件。

5.根据权利要求1-4中任一项所述的Linux套接字连接事件监控方法,其特征在于,所述基于所述连接事件数据创建连接事件,包括:

将所述连接事件数据写入循环缓冲,并将所述连接事件依据时间依次排序;

通过监控进程读取写入所述循环缓冲的所述连接事件数据;

在所述监控进程内,基于所述连接事件数据创建所述连接事件。

6.根据权利要求5所述的Linux套接字连接事件监控方法,其特征在于,还包括:

通过所述监控进程将所述套接字连接事件通知响应进程。

7.一种Linux套接字连接事件监控装置,其特征在于,包括:

跟踪加载模块,用于加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;

信息获取模块,用于在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;以及

事件创建模块,用于基于所述连接事件数据创建连接事件;

跟踪加载模块,还用于加载协议Kprobe跟踪点至Linux系统的inet_stream_connect函数和/或inet_dgram_connect函数入口处;

信息获取模块,还用于在Linux系统调用执行到所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处时,触发所述协议Kprobe跟踪点采集传入所述inet_stream_connect函数和/或所述inet_dgram_connect函数内的套接字协议类型;

事件创建模块,还用于将所述套接字协议类型填充至所述连接事件内,得到协议连接事件;

跟踪加载模块,还用于加载状态Kprobe跟踪点至Linux系统的所述sys_connect函数和/或所述_sys_connect函数出口处;

信息获取模块,还用于在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数出口处时,触发所述状态Kprobe跟踪点采集执行所述sys_connect函数和/或所述_sys_connect函数后的返回值;

事件创建模块,还用于将所述返回值填充至所述协议连接事件内,得到状态连接事件。

8.一种电子设备,其特征在于,包括:

处理器和存储器;

所述处理器通过调用所述存储器存储的程序或指令,用于执行如权利要求1至6中任一项所述方法的步骤。

一种Linux套接字连接事件监控方法及装置

技术领域

[0001] 本申请各实施例属计算机技术领域,尤其涉及一种Linux套接字连接事件监控方法及装置。

背景技术

[0002] 套接字(Socket),就是对网络中不同主机上的应用进程之间进行双向通信的端点的抽象。一个套接字就是网络上进程通信的一端,提供了应用层进程利用网络协议交换数据的机制。套接字上联应用进程,下联网络协议栈,是应用程序通过网络协议进行通信的接口,是应用程序与网络协议栈进行交互的接口,基于套接字的上述特性,在工作场景中出于安全因素考虑,通常需要对套接字的连接进行监控。

[0003] 目前对套接字连接事件的监控通常采用在套接字的API(Application Programming Interface,应用程序编程接口)上挂载 hook,采集进程的套接字事件数据,并通过跨进程通信的方式传到监控进程,其存在占用运行资源多、无法监控容器内进程产生的套接字连接事件等问题。

发明内容

[0004] 本实施例提供了一种Linux套接字连接事件监控方法及装置,能够解决现有套接字连接事件监控存在的占用运行资源多、无法监控容器内进程产生的套接字连接事件的问题。

[0005] 第一方面,本实施例提供了一种Linux套接字连接事件监控方法,包括:

[0006] 加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;

[0007] 在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;

[0008] 基于所述连接事件数据创建连接事件。

[0009] 在一些实施例中,所述连接事件数据包括IP地址、端口号、文件描述符、进程描述符、事件时间戳及IP地址族中的至少一项。

[0010] 在一些实施例中,所述将所述安装升级资源及所述安装升级描述文件打包成离线安装包,具体为:将所述应用描述文件、所述安装升级描述文件及所述应用docker 镜像打包成所述离线安装包。

[0011] 在一些实施例中,所述基于所述连接事件数据创建连接事件,包括:设置地址族过滤条件和地址长度过滤条件;基于所述地址族过滤条件及所述地址长度过滤条件从所述连接事件数据中过滤出包含ipv4地址和/或ipv6地址的地址事件数据;基于所述地址事件数据创建所述连接事件。

[0012] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:加载协议Kprobe

跟踪点至Linux系统的inet_stream_connect函数和/或inet_dgram_connect函数入口处；在Linux系统调用执行到所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处时，触发所述协议Kprobe跟踪点采集传入所述inet_stream_connect函数和/或所述inet_dgram_connect函数内的套接字协议类型；将所述套接字协议类型填充至所述连接事件内，得到协议连接事件。

[0013] 在一些实施例中，所述Linux套接字连接事件监控方法，还包括：加载状态Kprobe跟踪点至Linux系统的所述sys_connect函数和/或所述_sys_connect函数出口处；在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数出口处时，触发所述状态Kprobe跟踪点采集执行所述sys_connect函数和/或所述_sys_connect函数后的返回值；将所述返回值填充至所述协议连接事件内，得到状态连接事件。

[0014] 在一些实施例中，所述Linux套接字连接事件监控方法，还包括：将所述事件时间戳设置为所述状态连接事件的时间戳，得到套接字连接事件。

[0015] 在一些实施例中，所述基于所述连接事件数据创建连接事件，包括：将所述连接事件数据写入循环缓冲，并将所述连接事件依据时间依次排序；通过监控进程读取写入所述循环缓冲的所述连接事件数据；在所述监控进程内，基于所述连接事件数据创建所述连接事件。

[0016] 在一些实施例中，所述Linux套接字连接事件监控方法，还包括：通过所述监控进程将所述套接字连接事件通知响应进程。

[0017] 第二方面，本实施例提供了一种Linux套接字连接事件监控装置，包括：

[0018] 跟踪加载模块，用于加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处；

[0019] 信息获取模块，用于在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时，触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据；以及

[0020] 事件创建模块，用于基于所述连接事件数据创建连接事件。

[0021] 第三方面，本实施例提供了一种电子设备，包括处理器和存储器；

[0022] 所述处理器通过调用所述存储器存储的程序或指令，用于执行如第一方面中任一实施例所述方法的步骤。

[0023] 本申请提供了一种Linux套接字连接事件监控方法及装置，通过加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处；在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时，触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据；基于所述连接事件数据创建连接事件，能够解决现有套接字连接事件监控存在的占用运行资源多、无法监控容器内进程产生的套接字连接事件的问题，实现了能够监控容器内运行的进程产生的套接字连接事件、并且占用运行资源少，提高了套接字连接事件监控的可靠性。

附图说明

[0024] 此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。后文将参照

附图以示例性而非限制性的方式详细描述本申请的一些具体实施例。附图中相同的附图标记标示了相同或类似的部件或部分，本领域技术人员应该理解的是，这些附图未必是按比例绘制的，在附图中：

[0025] 图1为本说明书实施例提供的一种Linux套接字连接事件监控方法的流程图；

[0026] 图2为本说明书实施例提供的一种Linux套接字连接事件监控装置的示意图；

[0027] 图3为本说明书实施例提供的一种电子设备示意图。

具体实施方式

[0028] 为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。显然，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的范围。

[0029] Socket(套接字)可以看成是两个网络应用程序进行通信时，各自通信连接中的端点。它是网络环境中进程间通信的API(应用程序编程接口)，也是可以被命名和寻址的通信端点，使用中的每一个套接字都有其类型和一个与之相连进程。通信时其中一个网络应用程序将要传输的一段信息写入它所在主机的 Socket中，该 Socket通过与网络接口卡(NIC)相连的传输介质将这段信息送到另外一台主机的 Socket中，使对方能够接收到这段信息。

[0030] 要通过互联网进行通信，至少需要一对套接字，其中一个运行于客户端，称之为 Client Socket，另一个运行于服务器端，称之为 Server Socket。根据连接启动的方式以及本地套接字要连接的目标，套接字之间的连接过程可以分为三个步骤：1)服务器监听；2)客户端请求；3)连接确认。

[0031] 目前 Linux操作系统上的套接字连接事件监控方法及问题为：

[0032] 方法一、基于 LD_PRELOAD 进行 DLL注入，在libc的connect、bind、listen等套接字 API上挂载 hook，采集进程的套接字事件数据，并通过跨进程通信的方式传回监控进程；方法一存在的问题：基于 LD_PRELOAD进行 DLL注入，需要修改Linux系统的 LD_PRELOAD 配置，并且要求待监控程序不能以静态链接的方式连接libc，而且不能注入已经启动的进程， Linux容器(如 Docker)内的进程往往使用容器内的LD_PRELOAD配置，因此也不能被监控，且配置难度大；

[0033] 方法二、基于ptrace函数挂载进程，在套接字的connect、bind、listen 等相关系统调用上设置断点，采集进程的系统调用参数并转化为套接字事件数据；方法二存在的问题：基于ptrace函数挂载进程，每次执行到套接字相关系统调用都需要切换到监控进程以执行数据采集代码，并在不同进程地址空间内拷贝数据，耗费计算资源，会引起严重的性能问题；

[0034] 方法三、编写跟踪套接字活动的eBPF(Extended Berkeley Packet Filter，扩展的伯克利包过滤器)程序，在内核挂载跟踪点并采集套接字相关数据，监控进程通过eBPF Map 读取内核记录的数据；方法三存在的问题：编写跟踪套接字事件的eBPF(Extended Berkeley Packet Filter)程序，但只能在较高内核版本(>=4.1)的 Linux系统中使用，而

目前大量生产环境中运行的 Linux系统,内核版本均不符合该要求。

[0035] 针对上述技术问题,第一方面,如图1所示,本实施例提供了一种Linux套接字连接事件监控方法,包括:

[0036] S101:加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;

[0037] 需要说明的是,Linux基于 Unix的“一切皆文件”的原则,将一些进程可访问的对象抽象为文件,进程通过指定文件描述符(FileDescriptor,简称为FD)发起系统调用的方式对操作这些对象。而套接字属于 Linux下被抽象为文件的对象,进程可以通过socket系统调用来创建一个套接字,并通过 connect系统调用来创建套接字连接。

[0038] 需要说明的是,在远程/用户进程创建一个套接字连接时,需要调用Linux系统的connect系统调用,在进行connect系统调用时,会执行到sys_connect函数和/或_sys_connect函数,故可在sys_connect函数和/或_sys_connect函数的入口处设置连接Kprobe跟踪点,以采集传入所述sys_connect函数和/或所述_sys_connect函数的数据(sys_connect函数或_sys_connect函数的表示方式因Linux内核版本的不同而略有不同)。

[0039] S102:在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;

[0040] 需要说明的是,在将所述连接Kprobe跟踪点加载到所述sys_connect函数和/或所述_sys_connect函数入口处时,通常需在所述sys_connect函数和/或所述_sys_connect函数入口处设置中断指令,以便执行所述连接Kprobe跟踪点进行采集数据,并在采集数据完成后继续向后执行。

[0041] 在一些实施例中,所述连接事件数据包括IP地址、端口号、文件描述符、进程描述符、事件时间戳及IP地址族中的至少一项。

[0042] 需要说明的是,为了获得完整的连接事件的数据,通常需获取尽可能充分、全面的数据,以便对套接字连接事件有全面的了解,通常采集ipv4的连接事件数据的方式与采集ipv6的连接事件数据的方式不同,ipv4为:通过addr->sa_family采集连接事件的地址族、通过((struct sockaddr_in*)addr)->sin_port采集连接事件的端口号、通过((struct sockaddr_in*)addr)->sin_addr采集连接事件的IP地址;ipv6为:通过addr->sa_family采集连接事件的地址族、通过((struct sockaddr_in6*)addr)->sin6_port采集连接事件的端口号、通过((struct sockaddr_in6*)addr)->sin6_flowinfo、((struct sockaddr_in6*)addr)->sin6_addr、((struct sockaddr_in6*)addr)->sin6_scope_id采集连接事件的IP地址,所述IP地址及所述端口号即可构成连接事件的套接字。

[0043] 需要说明的是,针对不同类型的套接字,其套接字地址的构成方式可能不同,故可针对不同类型的套接字设置不同的函数/公式进行采集。

[0044] 在一些实施例中,所述基于所述连接事件数据创建连接事件,包括:设置地址族过滤条件和地址长度过滤条件;基于所述地址族过滤条件及所述地址长度过滤条件从所述连接事件数据中过滤出包含ipv4地址和/或ipv6地址的地址事件数据;基于所述地址事件数据创建所述连接事件。

[0045] 需要说明的是,根据IP地址族的不同其地址族过滤条件也不同,通常ipv4的地址

族过滤条件是addr->sa_family == AF_INET、地址长度过滤条件是size == sizeof(struct sockaddr_in);通常ipv6的地址族过滤条件是addr->sa_family == AF_INET6、地址长度过滤条件是size == sizeof(struct sockaddr_in6)。

[0046] 需要说明的是,针对需要采集信息的地址族的不同,可分别设置不同的连接kprobe跟踪点,即ipv4连接kprobe跟踪点、ipv6连接kprobe跟踪点,所述ipv4连接kprobe跟踪点用于采集、过滤ipv4的连接事件数据,所述ipv6连接kprobe跟踪点用于采集、过滤ipv6的连接事件数据。

[0047] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:加载协议Kprobe跟踪点至Linux系统的inet_stream_connect函数和/或inet_dgram_connect函数入口处;在Linux系统调用执行到所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处时,触发所述协议Kprobe跟踪点采集传入所述inet_stream_connect函数和/或所述inet_dgram_connect函数内的套接字协议类型;将所述套接字协议类型填充至所述连接事件内,得到协议连接事件。

[0048] 需要说明的是,通常传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据并不包含套接字的协议类型,sys_connect或 __sys_connect函数会依据文件描述符取出对应的文件和套接字对象,并依据套接字对象的proto_ops->connect 执行对应传输层协议的连接处理函数,对于TCP套接字,该函数为inet_stream_connect,对于UDP 和 raw套接字,该函数为inet_dgram_connect。故为了获取套接字的协议类型,可在所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处插入协议Kprobe跟踪点,以采集套接字的协议类型,通常是采集socket(套接字)参数中的type字段,当传输层协议类型为TCP 时,type字段的值为SOCK_STREAM;当传输层协议的类型为 UDP时,type字段的值为SOCK_DGRAM;当传输层协议的类型为raw 时,type字段的值为SOCK_RAW。并将采集到的type字段的值填充所述连接事件的传输层协议信息。

[0049] 需要说明的是,可将所述连接事件的地址分为网络层地址和传输层地址,所述网络层地址包括所述连接事件数据,所述传输层地址包括传输层协议(如SOCK_STREAM、SOCK_DGRAM、SOCK_RAW)。

[0050] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:加载状态Kprobe跟踪点至Linux系统的所述sys_connect函数和/或所述_sys_connect函数出口处;在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数出口处时,触发所述状态Kprobe跟踪点采集执行所述sys_connect函数和/或所述_sys_connect函数后的返回值;将所述返回值填充至所述协议连接事件内,得到状态连接事件。

[0051] 需要说明的是,为了对套接字连接进行完整的监控,还需获取远程进程进行套接字连接的结果,即连接成功、连接失败等信息,因套接字的连接结果是在执行所述sys_connect函数和/或所述_sys_connect函数后才会生成,故需在所述sys_connect函数和/或所述_sys_connect函数出口处设置状态Kprobe跟踪点,以采集远程进程进行套接字连接的状态,通常返回值(errno)有多种情况,如errno: 0表示Success;errno: 1表示Operation not permittederrno: 2 表示No such file or directoryerrno: 3 表示No such process;errno: 4 表示Interrupted system call;errno: 5 表示Input/output error等。

[0052] 需要说明的是,通过采集所述sys_connect函数和/或所述_sys_connect函数的返回值(errno),并将所述返回值填充至所述协议连接事件内,能够了解连接操作是否成功完成,以及失败的原因,以便进行后续的分析处理。

[0053] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:将所述事件时间戳设置为所述状态连接事件的时间戳,得到套接字连接事件。

[0054] 需要说明的是,需要对每个套接字连接事件打上时间戳,以便后续对多个套接字连接事件进行整体分析、汇总处理。

[0055] S103:基于所述连接事件数据创建连接事件。

[0056] 在一些实施例中,所述基于所述连接事件数据创建连接事件,包括:将所述连接事件数据写入循环缓冲,并将所述连接事件依据时间依次排序;通过监控进程读取写入所述循环缓冲的所述连接事件数据;在所述监控进程内,基于所述连接事件数据创建所述连接事件。

[0057] 需要说明的是,在获取到所述连接事件数据后,可直接创建所述连接事件;获取到所述传输层协议的值后将其填充到所述连接事件的传输层地址中,形成所述协议连接事件;获取到所述返回值后,将所述返回值填充至所述协议连接事件内,得到所述状态连接事件;也可分别将所述连接事件数据、所述传输层协议的值及所述返回值发送至监控进程,并在监控进程内创建/形成连接事件、协议连接事件、状态连接事件以及套接字连接事件。

[0058] 需要说明的是,可单独编写程序对所述连接事件进行排序,也可通过Linuxtracing子系统对写入所述循环缓冲的数据进行读取,并对所述连接事件进行排序,然后通过监控进程循环读取所述连接事件,此时监控进程读取的事件是已按时间顺序排序完成的事件。

[0059] 需要说明的是,监控进程可通过循环读取tracefs文件系统的trace_pipe特殊文件获取当前已经采集到的数据或等待时间可读数据,监控进程从循环缓冲中读取到的事件/数据,可派发给对应的事件处理函数进行处理。

[0060] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:通过所述监控进程将所述套接字连接事件通知响应进程。

[0061] 需要说明的是,监控进行可对采集/生成的连接事件(连接事件、协议连接事件、状态连接事件、套接字连接事件)进行分析,将所述连接事件发送至响应进程,以使所述响应进程采取相应的处理措施。

[0062] 综上所述,本实施例提供了一种Linux套接字连接事件监控方法,通过加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;基于所述连接事件数据创建连接事件,能够解决现有套接字连接事件监控存在的占用运行资源多、无法监控容器内进程产生的套接字连接事件的问题,实现了能够监控容器内运行的进程产生的套接字连接事件、并且占用运行资源少,提高了套接字连接事件监控的可靠性。

[0063] 需要说明的是,针对现有技术中存在的问题,本实施例的技术方案实现了:1)、能够应用于最低 Linux内核版本(2.6.32的 Linux发行版上),普遍适用于目前国内企业的

IT基础设施的网络安全能力建设需求;2)、对网络性能影响较小(通过wrk等网络性能测试工具对使用了本技术的环境进行压力测试和性能测试,测得使用本技术后对网络吞吐量的影响在2% 以内);3)、启用和停用本实施例的方法无需重启系统/修改系统配置,并且可以随时开始和停止监测;4)、能应用于 Linux容器场景,监控容器内的套接字连接事件。

[0064] 第二方面,如图2所示,本实施例提供了一种Linux套接字连接事件监控方法装置,包括:

[0065] 跟踪加载模块210,用于加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处;

[0066] 信息获取模块220,用于在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时,触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据;以及

[0067] 事件创建模块230,用于基于所述连接事件数据创建连接事件。

[0068] 在一些实施例中,所述连接事件数据包括IP地址、端口号、文件描述符、进程描述符、事件时间戳及IP地址族中的至少一项。

[0069] 在一些实施例中,所述将所述安装升级资源及所述安装升级描述文件打包成离线安装包,具体为:将所述应用描述文件、所述安装升级描述文件及所述应用docker 镜像打包成所述离线安装包。

[0070] 在一些实施例中,所述基于所述连接事件数据创建连接事件,包括:设置地址族过滤条件和地址长度过滤条件;基于所述地址族过滤条件及所述地址长度过滤条件从所述连接事件数据中过滤出包含ipv4地址和/或ipv6地址的地址事件数据;基于所述地址事件数据创建所述连接事件。

[0071] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:加载协议Kprobe跟踪点至Linux系统的inet_stream_connect函数和/或inet_dgram_connect函数入口处;在Linux系统调用执行到所述inet_stream_connect函数和/或所述inet_dgram_connect函数入口处时,触发所述协议Kprobe跟踪点采集传入所述inet_stream_connect函数和/或所述inet_dgram_connect函数内的套接字协议类型;将所述套接字协议类型填充至所述连接事件内,得到协议连接事件。

[0072] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:加载状态Kprobe跟踪点至Linux系统的所述sys_connect函数和/或所述_sys_connect函数出口处;在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数出口处时,触发所述状态Kprobe跟踪点采集执行所述sys_connect函数和/或所述_sys_connect函数后的返回值;将所述返回值填充至所述协议连接事件内,得到状态连接事件。

[0073] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:将所述事件时间戳设置为所述状态连接事件的时间戳,得到套接字连接事件。

[0074] 在一些实施例中,所述基于所述连接事件数据创建连接事件,包括:将所述连接事件数据写入循环缓冲,并将所述连接事件依据时间依次排序;通过监控进程读取写入所述循环缓冲的所述连接事件数据;在所述监控进程内,基于所述连接事件数据创建所述连接事件。

[0075] 在一些实施例中,所述Linux套接字连接事件监控方法,还包括:通过所述监控进

程将所述套接字连接事件通知响应进程。

[0076] 第三方面，如图3所示，本实施例提供了一种电子设备300，包括处理器320和存储器310；

[0077] 所述处理器320通过调用所述存储器310存储的程序或指令，用于执行如第一方面中任一实施例所述方法的步骤。

[0078] 最后应说明的是：以上各实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述各实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分或者全部技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

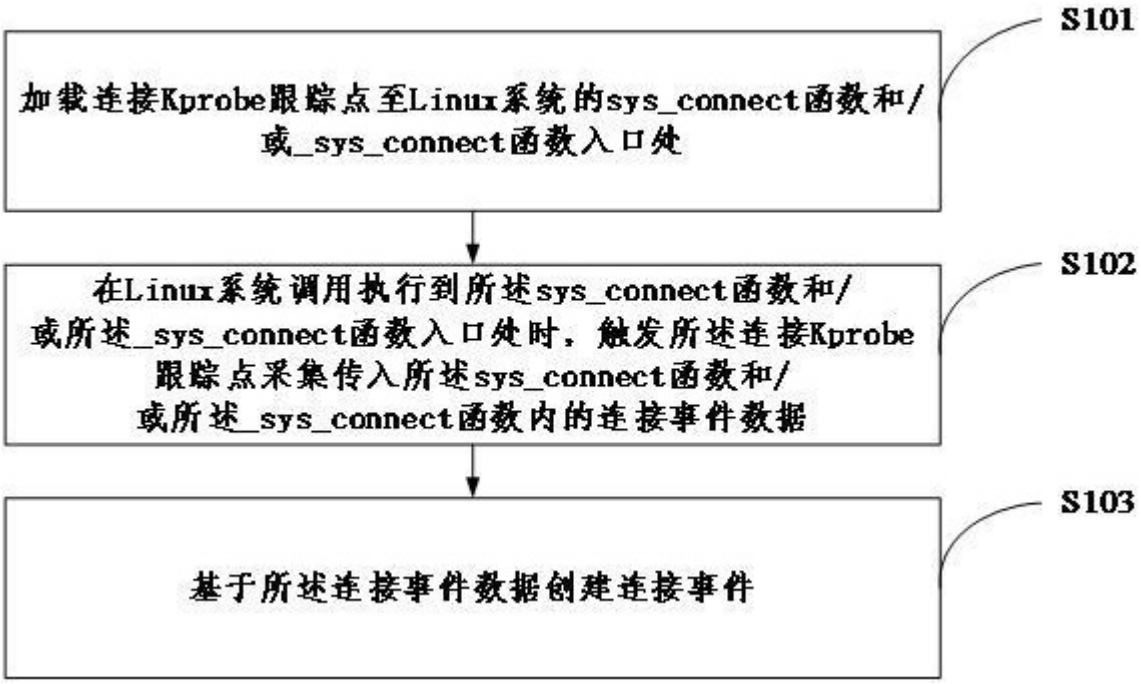

S101
加载连接Kprobe跟踪点至Linux系统的sys_connect函数和/或_sys_connect函数入口处
S102
在Linux系统调用执行到所述sys_connect函数和/或所述_sys_connect函数入口处时，触发所述连接Kprobe跟踪点采集传入所述sys_connect函数和/或所述_sys_connect函数内的连接事件数据
S103
基于所述连接事件数据创建连接事件

图 1

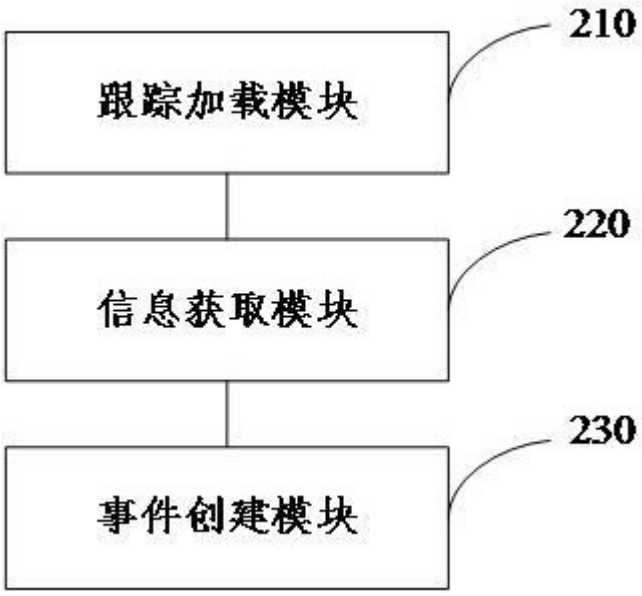

210
跟踪加载模块
220
信息获取模块
230
事件创建模块

图 2

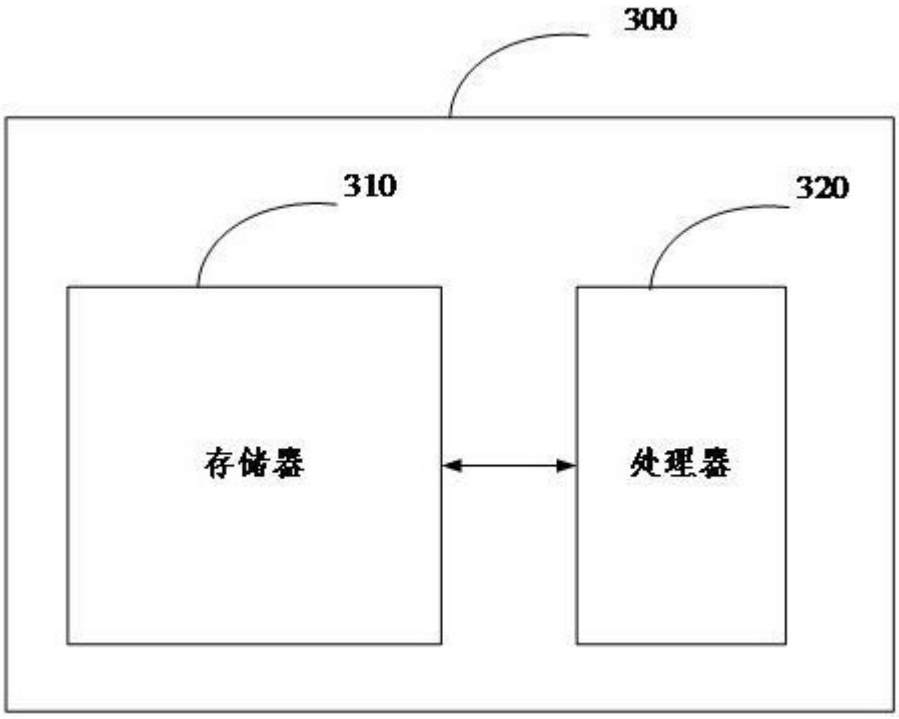


图 3