

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2019 年 9 月 19 日 (19.09.2019)



(10) 国际公布号
WO 2019/174193 A1

(51) 国际专利分类号:
G06F 21/53 (2013.01)

(21) 国际申请号: PCT/CN2018/102667

(22) 国际申请日: 2018 年 8 月 28 日 (28.08.2018)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
PCT/CN2018/079338
2018年3月16日 (16.03.2018) CN

(71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 张煜(ZHANG, Yu); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 匡大虎(KUANG, Dahu); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 俞岳(YU, Yue); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 陈宇(CHEN, Yu); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,

(54) **Title:** CONTAINER ESCAPE DETECTION METHOD, APPARATUS AND SYSTEM, AND STORAGE MEDIUM

(54) 发明名称: 容器逃逸检测方法、装置、系统及存储介质

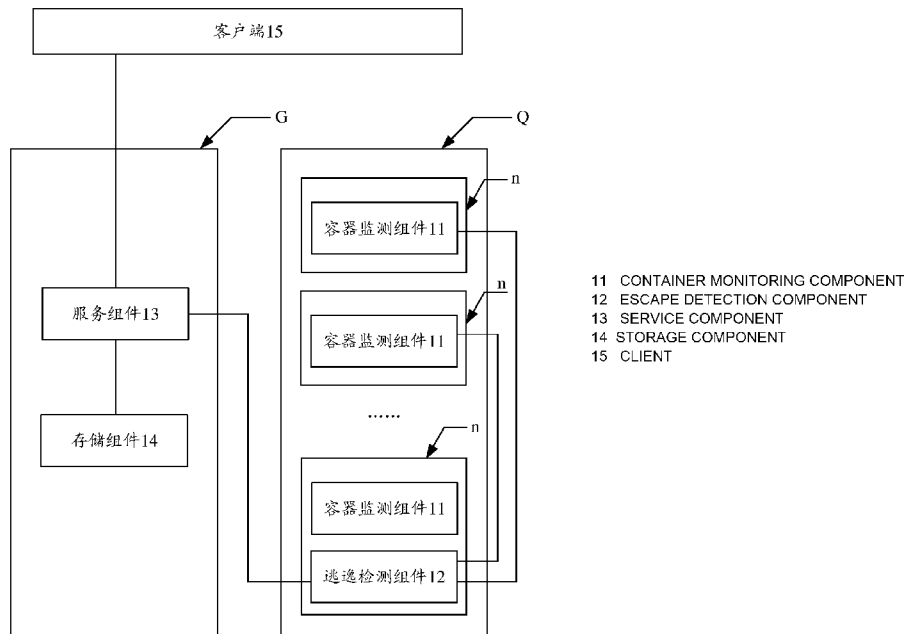


图 1B

(57) **Abstract:** A container escape detection method, apparatus and system, and a storage medium, which relate to the field of container security. The method comprises: an escape detection component (12) receiving information, reported by the container monitoring component (11), of multiple system calls triggered by a monitored container; and the escape detection component (12) matching the occurrence order of the multiple system calls with at least one set of pre-set system call orders in an escape detection rule, and determining, according to a matching result, whether the monitored container escapes, wherein each set of pre-set system call orders in the

PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG,
US, UZ, VC, VN, ZA, ZM, ZW。

- (84) 指定国 (除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM,
AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,
CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

- 包括国际检索报告 (条约第21条(3))。

escape detection rule corresponds to a predicted occurrence order of multiple system calls triggered by one container escape behavior. According to the present invention, the strength of preventing container escape can be improved.

(57) 摘要: 一种容器逃逸检测方法、装置、系统及存储介质, 涉及容器安全领域, 其中方法包括: 逃逸检测组件(12)接收所述容器监测组件(11)上报的被监测容器触发的多个系统调用的信息, 逃逸检测组件(12)将所述多个系统调用的发生顺序与逃逸检测规则中的至少一组预设系统调用顺序进行匹配, 并根据匹配结果确定所述被监测容器是否发生逃逸, 逃逸检测规则中的每组预设系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序。能够提高防范容器逃逸的力度。

容器逃逸检测方法、装置、系统及存储介质

技术领域

5 本申请涉及容器安全领域，特别涉及一种容器逃逸检测方法、装置、系统及存储介质。

背景技术

云计算（英文：Cloud Computing）是一种通过网络以服务的方式提供动态可伸缩的虚拟化资源的计算模式，其中，平台即服务（英文：Platform as a Service；简称：PaaS）是云计算的一种主要的模式。在 PaaS 模式中，云计算平台可以为每一个用户分配一个宿主机集群，该宿主机集群可以包括至少一个宿主机，其中，宿主机可以为物理机或虚拟机。用户可以在分配的宿主机集群的宿主机中基于容器镜像创建并运行容器，其中，每一个容器都拥有一个独立的进程运行空间，理想情况下，容器中的进程只能在该容器的进程运行空间中运行。然而，当容器中存在有恶意进程时，该恶意进程很可能会脱离容器的进程运行空间，继而对宿主机或其他容器展开攻击，这种现象被称为容器逃逸。

相关技术中，为了防范容器逃逸，在某一容器运行前，宿主机可以判断该容器是否拥有一些敏感的系统权限（例如，该敏感的系统权限可以为 `dac_override_capability` 系统权限），在宿主机确定该容器拥有敏感的系统权限时，宿主机可以拒绝该容器的运行，从而避免容器中的恶意进程基于该敏感的系统权限对宿主机或其他容器展开攻击。

20 在实现本申请的过程中，发明人发现现有技术至少存在以下问题：

相关技术仅能在容器运行前被动地对容器逃逸进行防范，而对容器运行后发生的容器逃逸事件无法检测防范，因此，对容器逃逸的防范力度较低。

发明内容

25 本申请提供了一种容器逃逸检测方法、装置、系统及存储介质，能够提高对容器逃逸的防范力度。

第一方面，提供了一种计算系统中容器逃逸检测方法，该计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则包括至少一组预设系统调用顺序，每组预设系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；该方法包括：所述逃逸检测组件接收所述容器监测组件上报的被监测容器触发的多个系统调用的信息；所述逃逸检测组件将所述多个系统调用的发生顺序与所述逃逸检测规则中的至少一组预设系统调用顺序进行匹配，并根据匹配结果确定所述被监测容器是否发生逃逸。

35 在容器逃逸发生的过程中，容器中的恶意进程会不断地通过暴力破解的方式尝试获取宿主机文件的句柄，在获取到宿主机文件的句柄后，该恶意进程可以篡改宿主机中的敏感文件，从而完成容器逃逸攻击，而在暴力破解宿主机文件句柄的过程中，恶意进程往往需

要按照一定的顺序使用宿主机中的系统调用（英文：system call）。因此，逃逸检测组件可以根据被监测容器中的进程使用宿主机中的多个系统调用的发生顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性，其中，预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的预测发生顺序，这使得逃逸检测组件可以对容器运行后发生的容器逃逸事件进行检测，从而提高到了对容器逃逸的防范力度。

具体地，逃逸检测组件上可以预先配置逃逸检测规则，该逃逸检测规则包含至少一组预设系统调用顺序，每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序。当逃逸检测组件获取目标顺序信息之后，可以将目标顺序信息与逃逸检测规则中的至少一组预设系统调用顺序进行匹配以确定第一容器逃逸的可能性，也就是可能性标识。

可选地，逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力；所述方法还包括：所述逃逸检测组件接收所述容器监测组件上报的被监测容器的启动参数，所述启动参数包含所述被监测容器启用的系统能力；所述逃逸检测组件根据预设的所述逃逸检测规则以及所述被监测容器的启用的系统能力，确定所述被监测容器是否发生逃逸。

Shocker 攻击运行需要使用一些特定的对系统调用(system call)的能力，例如使用 dac_override 的能力或者使用 dac_read_search 的能力。攻击者在启动(docker run)shocker 攻击的时候，docker run 的命令中会增加一些特定的系统能力参数。逃逸检测组件可以利用容器启动参数中容器启动的系统能力来检测容器的逃逸可能性，在逃逸检测规则中预先配置高危敏感的系统能力，每种高危系统能力对应一个逃逸可能性（可能性标识）。容器监测组件通过调用 docker.socket 接口可以获取容器启动参数，并上报到容器逃逸检测组件，容器逃逸检测组件根据预设的所述逃逸检测规则以及所述被监测容器的启用的系统能力，确定所述被监测容器是否发生逃逸。可选的，所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括：

获取目标状态机，所述目标状态机包括初始状态、最终状态以及由所述初始状态转移至所述最终状态所需满足的状态转移条件，所述状态转移条件为输入的顺序信息所指示的顺序与所述预设系统调用顺序相同的条件；

在所述目标状态机处于所述初始状态时，将所述目标顺序信息输入至所述目标状态机；当所述目标状态机由所述初始状态转移至所述最终状态时，确定所述第一容器发生了容器逃逸。

可选的，所述获取目标状态机，包括：

获取所述被监测容器的第一进程的进程标识；

根据所述第一进程的进程标识查询状态机数据库，所述状态机数据库中存储有至少一组状态机与进程标识的对应关系；

当所述状态机数据库中存储有与所述第一进程的进程标识对应的状态机时，将所述状态机获取为所述目标状态机。

可选的，所述获取目标状态机，还包括：

当所述状态机数据库中存储有与所述第一进程的进程标识对应的状态机时，根据所

述预设系统调用顺序创建状态机;

将创建的所述状态机获取为所述目标状态机。

可选的,所述逃逸检测规则采用目标决策树表达,所述目标决策树包括至少一个分支,每个分支包括系统能力节点和有序的多个系统调用节点,每个分支的系统能力节点对应于一种高危系统能力,每个分支的多个系统调用节点的预设顺序代表启用该分支的系统能力节点所描述的高危系统能力的容器发生逃逸时所触发的多个系统调用的预测发生顺序,所述目标决策树的每个分支的每个节点上都预先设置有相应的可能性标识;所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括:

所述逃逸检测组件在接收到所述被监测容器的启动参数后,根据获取到的所述被监测容器的启用的系统能力,与所述目标决策树中的各个系统能力节点进行匹配,获取匹配成功的系统能力节点对应的第一可能性标识,将所述第一可能性标识作为所述被监测容器当前的可能性标识;

所述逃逸检测组件在接收到所述被监测容器触发的多个系统调用的信息后,确定所述多个系统调用的实际发生顺序,将所述多个系统调用的实际发生顺序与匹配成功的系统能力节点所在的分支的预设顺序进行匹配,若在匹配成功的系统能力节点所在的分支的第 n 个系统调用节点匹配成功,获取所述第 n 个系统调用节点所对应的第二可能性标识,使用所述第二可能性标识更新所述被监测容器当前的可能性标识。

可选地,目标决策树中每个节点还可以进一步设置发生频率信息,每个节点所设置的发生频率表示对应该节点的事件发生的次数要求,当达到该次数要求,才可将该节点设置的可能性标识(逃逸系数)确定为被监测容器的当前的可能性标识。例如,节点 a1 为系统权限节点,节点 a1 设置的发生频率为 1,则表示只要收到被监测容器启用节点 a1 对应的系统权限 `dac_override_capability`,就可以确定被监测容器发生逃逸的可能性为可能性标识所指示的 4。节点 a2 为系统调用节点,节点 a2 设置的发生频率为 2,则表示收到被监测容器使用系统调用 `open_by_handle_at` 的频率为发生两次,才可以确定被监测容器发生逃逸的可能性为该节点的可能性标识所指示的 5。其他节点的发生频率以此类推,不再赘述。

在容器逃逸发生的过程中,容器中的恶意进程往往需要按照一定的顺序使用宿主机中的系统调用,而只有在容器拥有某些系统权限的前提下,容器中的进程才能使用宿主机中的系统调用。因此,在本申请实施例,逃逸检测组件可以根据被监测容器拥有的系统权限以及目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度,确定第一容器发生容器逃逸的可能性。

可选的,所述方法还包括:

接收所述容器监测组件发送的敏感文件篡改信息,所述敏感文件篡改信息是所述容器监测组件在监测到被监测容器中的第二进程对所述宿主机中预设的至少一个敏感文件中的任意一敏感文件进行写操作后发送的;基于所述敏感文件篡改信息,确定所述被监测容器发生了容器逃逸。

容器中的恶意进程在进行容器逃逸攻击时会篡改宿主机中的敏感文件,因此,当逃逸检测组件接收到容器监测组件发送的敏感文件篡改信息后,该逃逸检测组件即可确定被监测容器发生了容器逃逸。

可选的，在接收所述容器监测组件发送的敏感文件篡改信息之前，所述方法还包括：

将敏感文件配置信息发送至所述容器监测组件，所述敏感文件配置信息包括至少一个敏感文件的标识，所述敏感文件配置信息用于供所述容器监测组件根据所述敏感文件配置信息确定所述至少一个敏感文件。

5 可选的，所述方法还包括：

所述逃逸检测组件根据所述被监测容器是否发生逃逸，发送容器逃逸事件报告或逃逸告警。可选的，所述方法还包括：

接收所述容器监测组件发送的非法进程运行信息，所述非法进程运行信息是所述容器监测组件在监测到被监测容器中运行的第三进程的进程标识不属于进程白名单包括的进程标识后发送的，所述进程白名单包括多个合法进程的进程标识；基于所述非法进程运行信息，确定所述被监测容器可能发生了容器逃逸。

当目标容器监测组件监测到被监测容器中运行的第三进程的进程标识不属于进程白名单包括的进程标识后，该容器监测组件可以向逃逸检测组件发送非法进程运行信息。逃逸检测组件接收到该非法进程运行信息后，可以确定被监测容器中存在非法进程，由于该非法进程很可能为恶意进程，因此，逃逸检测组件在接收到该非法进程运行信息后，可以确定被监测容器可能发生了容器逃逸。

可选的，所述容器逃逸检测系统还包括客户端，所述在接收所述容器监测组件发送的非法进程运行信息之前，所述方法还包括：

接收所述容器监测组件发送的所述宿主机中的容器镜像与进程的对应关系；
20 将所述对应关系发送至所述客户端；
接收所述客户端发送的编辑后的所述对应关系；
根据编辑后的所述对应关系生成所述进程白名单；
将所述进程白名单发送至所述容器监测组件。

可选的，所述逃逸检测组件设置于所述宿主机集群中的第二宿主机中，所述第二宿主机的运算能力大于所述目标宿主机集群中除所述第二宿主机以外的至少一个宿主机。

可选的，所述逃逸检测组件设置于所述第二宿主机部署的一个容器中。

可选的，所述容器逃逸检测系统包括逃逸检测组件集群，所述逃逸检测组件集群包括多个所述逃逸检测组件，所述逃逸检测组件集群部署于防火墙内的可信区域。

第二方面，提供一种容器逃逸检测的方法，所述计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则包括至少一组预设系统调用顺序，每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；

所述方法包括：所述容器监测组件对被监测容器的运行情况进行监测；所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息，所述多个系统调用的发生顺序用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

可选地，所述容器监测组件向所述逃逸检测组件上报所述被监测容器的信息，所述被监测容器的信息包括所述被监测容器的 ID 或所述被监测容器的启动的进程 ID；所述容器监测组件接收所述逃逸检测组件根据所述被监测容器的信息下发的审计规则，所述审计规则指示所述容器监测组件对所述被监测容器的系统调用事件进行审计；所述容器监测组件向

所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息包括：所述容器监测组件在监测到所述被监测容器触发的系统调用事件后，向所述逃逸检测组件上报审计报告，所述审计报告携带所述被监测容器触发的多个系统调用的信息。

可选的，所述逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力；

5 所述方法还包括：

向所述逃逸检测组件发送所述被监测容器的启动参数，所述启动参数包含所述被监测容器启用的系统能力，所述被监测容器启用的系统能力用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。具体地，所述启动参数包括至少一个系统能力信息，每个所述系统能力信息用于指示所述被监测容器拥有的一个系统能力，所述启动参数用于

10 供所述逃逸检测组件根据所述启动参数确定所述被监测容器发生容器逃逸的可能性。

可选地，所述容器监测组件向所述逃逸检测组件上报被监测容器的启动参数包括：所述容器监测组件调用 docker socket 接口获取所述被监测容器的启动参数，向所述逃逸检测组件上报携带所述被监测容器的启动参数的 docker run 命令。可选的，所述方法还包括：

在监测到被监测容器中的第二进程对所述第一宿主机中预设的至少一个敏感文件中的

15 任一敏感文件进行写操作后，向所述逃逸检测组件发送敏感文件篡改信息，所述敏感文件篡改信息用于供所述逃逸检测组件基于所述敏感文件篡改信息确定所述被监测容器发生了容器逃逸，所述被监测容器是所述第一宿主机中部署的一个容器。

可选的，所述向所述逃逸检测组件发送敏感文件篡改信息之前，所述方法还包括：

接收所述逃逸检测组件发送的敏感文件配置信息，所述敏感文件配置信息包括至少一个敏感文件的标识；

根据所述敏感文件配置信息确定所述至少一个敏感文件。

可选的，所述方法还包括：

在监测到被监测容器中运行的第三进程的进程标识不属于进程白名单包括的进程标识后，向所述逃逸检测组件发送非法进程运行信息，所述非法进程运行信息用于供所述逃逸检测组件基于所述非法进程运行信息确定所述被监测容器可能发生了容器逃逸，所述进程白名单包括多个合法进程的进程标识，所述被监测容器是所述第一宿主机中部署的一个容器。

可选的，所述容器逃逸检测系统还包括客户端，所述向所述逃逸检测组件发送非法进程运行信息之前，所述方法还包括：

30 向所述逃逸检测组件发送所述第一宿主机中的容器镜像与进程的对应关系，以由所述逃逸检测组件将所述对应关系发送至所述客户端，并接收所述客户端返回的编辑后的所述对应关系；

接收所述进程白名单，所述进程白名单是所述逃逸检测组件根据编辑后的所述对应关系生成的。

35 可选的，所述逃逸检测组件设置于所述被监测容器所在的宿主机集群中的第二宿主机中，所述第二宿主机的运算能力大于所述被监测容器所在的宿主机集群中除所述第二宿主机以外的至少一个宿主机。

可选的，所述逃逸检测组件设置于所述第二宿主机部署的一个容器中。

可选的，所述容器逃逸检测系统包括逃逸检测组件集群，所述逃逸检测组件集群包括

多个所述逃逸检测组件，所述逃逸检测组件集群部署于防火墙内的可信区域。

可选的，所述容器监测组件设置于所述被监测容器所在宿主机部署的一个容器中。

第三方面，提供了一种容器逃逸检测装置，该装置包括至少一个模块，该至少一个模块用于实现上述第一方面或第一方面的任一可选方式所提供的容器逃逸检测方法。

5 第四方面，提供了一种容器逃逸检测装置，该装置包括至少一个模块，该至少一个模块用于实现上述第二方面或第二方面的任一可选方式所提供的容器逃逸检测方法。

第五方面，提供了一种容器逃逸检测装置，所述装置包括：处理器和存储器；

所述处理器被配置为执行所述存储器中存储的指令，所述处理器通过执行所述指令来实现上述第一方面或第一方面的任一可选方式所提供的容器逃逸检测方法。

10 第六方面，提供了一种容器逃逸检测装置，所述装置包括：处理器和存储器；

所述处理器被配置为执行所述存储器中存储的指令，所述处理器通过执行所述指令来实现上述第二方面或第二方面的任一可选方式所提供的容器逃逸检测方法。

第七方面，提供一种容器逃逸监测方法，所述计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则包括预定义的至少一种与容器逃逸相关的高危系统能力；所述方法包括：所述逃逸检测组件接收所述容器监测组件上报的被监测容器的启动参数，所述启动参数包含所述被监测容器启用的系统能力；所述逃逸检测组件根据预设的所述逃逸检测规则以及所述被监测容器的启用的系统能力，确定所述被监测容器是否发生逃逸。

20 可选地，所述逃逸检测规则还包括至少一组预设系统调用顺序，每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；所述方法还包括：所述逃逸检测组件接收所述容器监测组件上报的所述被监测容器触发的多个系统调用的信息；所述逃逸检测组件将所述多个系统调用的发生顺序与所述逃逸检测规则中的至少一组预设系统调用顺序进行匹配，并根据匹配结果确定所述被监测容器是否发生逃逸。

25 可选地，所述容器逃逸检测方法还执行如第一方面的可能的实现方式。

第八方面，提供一种容器逃逸监测方法，所述计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力；所述方法包括：所述容器监测组件对被监测容器的运行情况进行监测；
30 所述容器监测组件在监测到所述被监测容器启动后，向所述逃逸检测组件上报所述被监测容器的启动参数，所述启动参数包含所述被监测容器启用的系统能力，所述被监测容器启用的系统能力用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

可选地，所述容器监测组件向所述逃逸检测组件上报被监测容器的启动参数包括：所述容器监测组件调用 docker socket 接口获取所述被监测容器的启动参数，向所述逃逸检测组件上报携带所述被监测容器的启动参数的 docker run 命令。

35 可选地，所述逃逸检测规则还包括至少一组预设系统调用顺序，每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；所述方法还包括：所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息，所述多个系统调用的发生顺序用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生

逃逸。第九方面，提供了一种计算机可读存储介质或者计算机程序产品，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得所述计算机执行上述第一方面或第一方面的任一可选方式所提供的容器逃逸检测方法，或者，使得所述计算机执行上述第二方面或第二方面的任一可选方式所提供的容器逃逸检测方法，或者，使得所述计算机执行上述第七方面或第七方面的任一可选方式所提供的容器逃逸检测方法，或者，使得所述计算机执行上述第八方面或第八方面的任一可选方式所提供的容器逃逸检测方法。

第十方面，提供了一种容器逃逸检测系统，该容器逃逸检测系统包括上述第一方面任一所述的逃逸检测组件和上述第二方面任一所述的容器监测组件。

第十一方面，提供了一种容器逃逸检测系统，该容器逃逸检测系统包括上述第七方面任一所述的逃逸检测组件和上述第八方面任一所述的容器监测组件。本申请提供的技术方案的有益效果是：

通过在计算系统中设置逃逸检测组件和容器监测组件，容器监测组件对被监测容器的运行状况进行监测，上报相关的监测事件到逃逸检测组件，逃逸检测组件预先设置逃逸检测规则，逃逸检测组件根据上报的监测事件以及预设的逃逸检测规则确定被监测容器是否发生逃逸。当监测事件为被监测容器触发的多个系统调用的信息时，逃逸检测组件将多个系统调用的发生顺序与逃逸检测规则中的预设系统调用顺序进行匹配，并根据匹配程度确定被监测容器是否发生逃逸。当监测事件为被监测容器启用的系统能力时，逃逸检测组件将被监测容器启用的系统能力与逃逸检测规则中预设的高位系统能力进行匹配，并根据匹配程度确定被监测容器是否发生逃逸。对高危系统能力的使用检测以及被监测容器触发的系统调用的顺序的检测还可以结合起来。上述对容器的逃逸行为的监测与检测实现了容器执行过程中对容器逃逸攻击的预警和防范，保证了云计算系统的运行安全。。

附图说明

图 1A 是本申请实施例提供的一种容器逃逸攻击的路径的示意图。

图 1B 是本申请实施例提供的一种实施环境的示意图。

图 1C 是本申请实施例提供的另一种实施环境的示意图。

图 2A 是本申请实施例提供的一种容器逃逸检测方法的流程图。

图 2B 是本申请实施例提供的一种目标状态机的示意图。

图 2C 是本申请实施例提供的另一种目标状态机的示意图。

图 2D 是本申请实施例提供的一种目标决策树的示意图。

图 3 是本申请实施例提供的一种容器逃逸检测装置的框图。

图 4 是本申请实施例提供的一种容器逃逸检测装置的框图。

图 5 是本申请实施例提供的一种容器逃逸检测装置的框图。

图 6 是本申请实施例提供的一种容器逃逸检测装置的框图。

图 7 是本申请实施例提供的一种容器逃逸检测装置的框图。

图 8 是本申请实施例提供的一种容器逃逸检测系统的框图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作

进一步地详细描述。

云计算（英文：Cloud Computing）是一种通过网络以服务的方式提供动态可伸缩的虚拟化资源的计算模式，其中，平台即服务（英文：Platform as a Service；简称：PaaS）是云计算的一种主要的模式。

5 在PaaS模式中，云计算平台上可以注册有多个用户，对于每个用户，云计算平台都可以为其分配一个宿主机集群，该宿主机集群可以包括至少一个宿主机（英文：Host）。其中，宿主机是一个相对容器（英文：container）而言的概念，所谓宿主机指的就是部署容器的设备，实际应用中，宿主机可以是物理机（如服务器等），也可以是虚拟机。

10 PaaS模式的用户可以在云计算平台为其分配的宿主机集群中的宿主机上部署容器，以利用部署的容器进行相应的运算。在部署容器的过程中，用户可以从云计算平台的镜像软件仓库中下载容器镜像，并根据下载的容器镜像在宿主机中创建容器。其中，容器镜像可以包括至少一个可读不可写的文件层，在创建容器时，宿主机可以在该容器镜像上创建一个可读且可写的文件层，并创建一个独立的进程运行空间，容器中的进程可以在该进程运行空间中运行，运行中的进程可以读取该至少一个可读不可写的文件层中的文件，也可以
15 对该可读且可写的文件层中的文件进行写操作。

通常情况下，云计算平台可以允许普通用户向镜像软件仓库上传容器镜像，因此，该镜像软件仓库中容器镜像的安全性难以得到保证，这使得基于镜像软件仓库中的容器镜像创建的容器中很可能存在恶意进程，该恶意进程很可能会脱离容器的进程运行空间，继而
20 对容器的宿主机或其他容器展开攻击，这种现象通常被称为容器逃逸。

20 如图1A所示，实际应用中，容器逃逸一般可以有以下几种攻击路径：

第一种、容器中的恶意进程对容器守护进程（英文：docker daemon）进行攻击。

如图1A中的攻击路径1所示，宿主机1中部署的容器A内的恶意进程a可以对容器A进行攻击，在攻陷了容器A后，该恶意进程a可以脱离容器A的进程运行空间，并对容器守护进程S1进行攻击。

25 第二种、容器中的恶意进程通过容器守护进程对宿主机中部署的其他容器进行攻击。

如图1A中的攻击路径2所示，宿主机2中部署的容器C内的恶意进程c可以对容器C进行攻击，在攻陷了容器C后，该恶意进程c可以脱离容器C的进程运行空间，并对容器守护进程S2进行攻击，在攻陷了容器守护进程S2后，该恶意进程c可以对宿主机2中部署的容器D进行攻击。

30 第三种、容器中的恶意进程对宿主机的操作系统（英文：Operating System；简称：OS）进行攻击。

如图1A中的攻击路径3所示，宿主机2中部署的容器D内的恶意进程d可以对容器D进行攻击，在攻陷了容器D后，该恶意进程d可以脱离容器D的进程运行空间，并利用容器共享库对宿主机2的操作系统OS2进行攻击。

35 第四种、容器中的恶意进程通过宿主机的操作系统对其他容器进行攻击。

如图1A中的攻击路径4所示，宿主机1中部署的容器A内的恶意进程a可以对容器A进行攻击，在攻陷了容器A后，该恶意进程a可以脱离容器A的进程运行空间，并将病毒上传到宿主机1的操作系统OS1的共享存储卷上，宿主机1中部署的容器B内的进程b读取并执行该恶意进程a在共享存储卷中上传的病毒后即会中毒。

第五种、容器中的恶意进程对与该容器位于同一组（英文：group）中的其他容器进行攻击。

如图 1A 中的攻击路径 5 所示，宿主机 1 中部署的容器 B 内的恶意进程 b 可以对容器 B 进行攻击，在攻陷了容器 B 后，该恶意进程 b 可以脱离容器 B 的进程运行空间，并对与容器 B 位于同一组中的容器 C 进行攻击。

由于容器逃逸会对宿主机和容器带来较大的安全隐患，因此，防范容器逃逸已经成为保证云计算平台安全性的一个重要环节。

通常情况下，容器中的恶意进程往往需要基于容器拥有的一些敏感的系统权限对宿主机或其他容器展开攻击，例如，该敏感的系统权限可以为 `dac_override_capability` 或 `dac_read_search_capability` 等。因此，相关技术中，为了防范容器逃逸，在容器运行前，宿主机可以判断该容器是否拥有一些敏感的系统权限，在宿主机确定容器拥有敏感的系统权限时，宿主机即可确定该容器有可能会发生容器逃逸，因此，宿主机可以拒绝该容器的运行，从而达到防范容器逃逸的目的。

然而，上述相关技术仅能在容器运行前被动地对容器逃逸进行防范，而对容器运行后发生的容器逃逸事件无法检测防范，因此，对容器逃逸的防范力度较低。

本申请实施例提供了一种容器逃逸检测方法，该容器逃逸检测方法可以对容器运行后发生的容器逃逸事件进行检测，因此，可以提高防范容器逃逸的力度。

图 1B 为本申请实施例提供的容器逃逸检测方法所涉及的一种实施环境的示意图，如图 1B 所示，该实施环境可以包括至少一个容器监测组件 11（图 1B 中仅示出了 3 个容器监测组件 11）、逃逸检测组件 12、服务组件 13、存储组件 14 和客户端 15。

其中，如图 1B 所示，对于云计算平台为用户 A 分配的宿主机集群 Q 而言，该宿主机集群 Q 中的每个宿主机 n 上均可以设置有一个容器监测组件 11，可选的，某一宿主机中的容器监测组件 11 可以设置于该宿主机部署的一个容器中。其中，某一宿主机中设置的容器监测组件 11 可以对该宿主机中部署的容器进行监测，并将监测结果发送至逃逸检测组件 12 中。

逃逸检测组件 12 可以设置于宿主机集群 Q 中的某一个宿主机上，可选的，逃逸检测组件 12 可以设置于该宿主机部署的一个容器中。需要指出的是，逃逸检测组件 12 通常可以设置于宿主机集群 Q 中运算能力较强的一个宿主机中，实际实现时，该逃逸检测组件 12 可以设置于宿主机集群 Q 中运算能力最强的一个宿主机中。逃逸检测组件 12 可以接收宿主机集群 Q 中设置的各个容器监测组件 11 发送的监测结果，并可以根据接收到的监测结果确定宿主机集群 Q 中各宿主机中的容器发生容器逃逸的可能性，逃逸检测组件 12 还可以向服务组件 13 发送容器逃逸事件报告，该容器逃逸事件报告可以包括宿主机集群 Q 中某一宿主机中的容器发生容器逃逸的可能性。

服务组件 13 和存储组件 14 可以设置于防火墙内的可信区域 G 中，该服务组件 13 可以接收逃逸检测组件 12 发送的容器逃逸事件报告，服务组件 13 可以将接收到的容器逃逸事件报告存储于存储组件 14 中。

客户端 15 可以为网页客户端或者应用程序客户端，该客户端 15 可以在用户 A 的触发下向服务组件 13 发送查询请求，以查询宿主机集群 Q 的容器逃逸事件报告，服务组件 13

可以在接收到客户端 15 发送的查询请求后,从存储组件 14 中获取宿主机集群 Q 的容器逃逸事件报告,并将该容器逃逸事件报告发送至该客户端 15。

图 1C 为本申请实施例提供的容器逃逸检测方法所涉及到的另一种实施环境的示意图,如图 1C 所示,该实施环境可以包括至少一个容器监测组件 21、逃逸检测组件 22、服务组件 23、存储组件 24 和客户端 25。

其中,容器监测组件 21、服务组件 23、存储组件 24 和客户端 25 分别与图 1B 中所示的容器监测组件 11、服务组件 13、存储组件 14 和客户端 15 的部署方式相同,且,容器监测组件 21、服务组件 23、存储组件 24 和客户端 25 分别与图 1B 中所示的容器监测组件 11、服务组件 13、存储组件 14 和客户端 15 所执行的技术过程同理。

与图 1B 中的逃逸检测组件 12 不同的是,图 1C 中的逃逸检测组件 22 设置于防火墙内的可信区域 G 中,该逃逸检测组件 22 可以接收多个宿主机集群(图 1C 中仅示出了两个宿主机集群 U 和 V)中的容器监测组件 21 发送的监测结果,并可以根据接收到的监测结果确定多个宿主机集群中的宿主机发生容器逃逸的可能性。可选的,为了保证逃逸检测组件 22 的高可用性(英文:High Availability;简称:HA),在图 1C 所示的实施环境中,可以部署多个逃逸检测组件 22(图 1C 中仅示出了两个逃逸检测组件 22),这样就可以保证某一逃逸检测组件 22 发生故障不会影响到对容器逃逸的检测。

图 2A 为本申请实施例提供的一种容器逃逸检测方法的流程图,该容器逃逸检测方法可以应用于图 1B 或图 1C 所示的实施环境中,如图 2A 所示,该容器逃逸检测方法可以包括以下步骤:

步骤 201、在监测到第一容器中的第一进程使用第一宿主机中的系统调用后,目标容器监测组件向逃逸检测组件发送多个系统调用信息。

如上所述,本申请实施例提供的容器逃逸检测方法所涉及到的实施环境中可以包括至少一个容器监测组件。其中,每个容器监测组件均可以设置于宿主机集群中的一个宿主机上,用于对该宿主机中的容器进行监测,在进行容器逃逸检测时,每个容器监测组件所执行的技术过程都是同理的,因此,本申请实施例仅以该至少一个容器监测组件中的一个容器监测组件(也即是目标容器监测组件)为例对本申请实施例提供的容器逃逸检测方法进行说明。

其中,第一宿主机可以是待进行容器逃逸检测的宿主机集群中的一个宿主机,换句话说,该第一宿主机可以是云计算平台分配给某一用户的宿主机集群中的一个宿主机,目标容器监测组件是设置于该第一宿主机中的容器监测组件,第一容器为该第一宿主机中部署的一个容器(第一容器为被监测的多个容器中的任意一个容器),第一进程为该第一容器中运行的一个进程。

由于在容器逃逸发生的过程中,容器中的恶意进程会不断地通过暴力破解的方式尝试获取宿主机文件的句柄,在获取到宿主机文件的句柄后,该恶意进程可以篡改宿主机中的敏感文件,从而完成容器逃逸攻击,而在暴力破解宿主机文件句柄的过程中,恶意进程往往需要按照一定的顺序使用宿主机中的系统调用(英文:system call),例如,恶意进程可以按照如下顺序使用宿主机中的系统调用:open_by_handle_at、fstat、fcntl、getdents 和 open_by_handle_at。因此,在进行容器逃逸检测的过程中,容器监测组件需要对宿主机中部署的容器中的系统调用事件进行监控,并将监控结果发送至逃逸检测组件,以由逃逸检测

组件根据该监控结果确定宿主机中的容器发生容器逃逸的可能性。

在步骤 201 中，目标容器监测组件可以对第一宿主机中部署的容器中的系统调用事件进行监控，当监测到第一容器中的第一进程使用第一宿主机中的系统调用时，该目标容器监测组件可以生成多个系统调用信息，并将该多个系统调用信息发送至逃逸检测组件中，
5 以由逃逸检测组件根据该多个系统调用信息确定第一容器发生容器逃逸的可能性。其中，每个系统调用信息可以与第一进程使用的第一宿主机中的一个系统调用相对应，每个系统调用信息可以包括对应的系统调用的标识。

在 Linux 系统中，可以利用 audit 审计模块对第一宿主机中部署的容器中的系统调用事件进行监控，该 audit 审计模块中可以配置有多个审计规则，每个审计规则可以为容器中的
10 进程使用宿主机中的某一系统调用的规则。该 audit 审计模块可以在第一进程触发审计规则时，也即是，在第一进程使用第一宿主机中的系统调用时，生成审计报告，并将该审计报告发送至目标容器监测组件，目标容器监测组件接收到该审计报告后，可以基于该审计报告生成系统调用信息，并将该系统调用信息发送至逃逸检测组件。如下所示为一个示例性的审计报告：

15 arch=c00003e syscall=304 success=no exit=-116 a0=3 a1=7ffecb757ac0 a2=0 a3=4 item=0
ppid=24057 pid=24785 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0
tty=pts3 ses=4284967295 comm="shocker"exe="/app/shocker"subj=system_u:system_r:spc_t:s0
key=67fd5ccc277e19114d7c4516b71743180149b26bc745c94e23c60bb8c933543_gra_highrisk_s
yscall_24057。

20 在该审计报告中，字段：syscall=304 表示第一进程使用了第一宿主机中编号为 304 的系统调用，其中，该系统调用为 open_by_handle_at。

字段：success=no 表示第一进程对第一宿主机中编号为 304 的系统调用使用不成功。

25 字段：key=67fd5ccc277e19114d7c4516b71743180149b26bc745c94e23c60bb8c933543_gra_highrisk_syscall_24057 表示第一进程触发的审计规则为=67fd5ccc277e19114d7c4516b71743180149b26bc745c94e23c60bb8c933543_gra_highrisk_syscall_24057。

步骤 202、逃逸检测组件接收目标容器监测组件发送的多个系统调用信息，并根据该多个系统调用信息获取目标顺序信息。

30 其中，该目标顺序信息用于指示第一进程使用第一宿主机中的多个系统调用的发生顺序。本申请中，目标顺序信息为一种表达，该目标顺序信息为第一容器（被监测容器）触发的 n 个系统调用的发生顺序。

在一种可能的实现方式中，每个系统调用信息还可以携带有生成时间，该生成时间用于指示目标容器监测组件生成该系统调用信息的时间，逃逸检测组件可以根据每个系统调用信息携带的生成时间获取该多个系统调用信息对应的多个系统调用的发生顺序，而后，逃逸检测组件可以基于该多个系统调用信息对应的多个系统调用的顺序生成目标顺序信息，其中，该多个系统调用信息对应的多个系统调用的发生顺序即为该目标顺序信息所指示的顺序。本申请中，多个系统调用的发生顺序可以根据各个系统调用的生成时间确定，也可以根据各个系统调用的触发时间确定，或者可以根据各个系统调用的执行时间确定，或者也可以根据各个系统调用的接收时间确定。

例如，逃逸检测组件接收到目标容器监测组件发送的 3 个系统调用信息，该 3 个系统

调用信息分别为系统调用信息 a、系统调用信息 b 和系统调用信息 c。其中，系统调用信息 a 与第一进程使用的系统调用 p 相对应，系统调用信息 a 携带的生成时间为 14:00，系统调用信息 b 与第一进程使用的系统调用 q 相对应，系统调用信息 b 携带的生成时间为 14:01，系统调用信息 c 与第一进程使用的系统调用 k 相对应，系统调用信息 c 携带的生成时间为 14:02。则逃逸检测组件可以根据该 3 个系统调用信息携带的生成时间获取该 3 个系统调用信息对应的 3 个系统调用的发生顺序，该发生顺序为：系统调用 p、系统调用 q 和系统调用 k，而后，逃逸检测组件可以根据该顺序生成目标顺序信息，其中，目标顺序信息所指示的顺序即为：系统调用 p、系统调用 q 和系统调用 k。

在另一种可能的实现方式中，逃逸检测组件可以根据接收到该多个系统调用信息的顺序获取该多个系统调用信息对应的多个系统调用的顺序，而后，逃逸检测组件可以基于该多个系统调用信息对应的多个系统调用的顺序生成目标顺序信息，其中，该多个系统调用信息对应的多个系统调用的顺序即为该目标顺序信息所指示的顺序。

例如，逃逸检测组件接收到目标容器监测组件发送的 3 个系统调用信息，该 3 个系统调用信息分别为系统调用信息 a、系统调用信息 b 和系统调用信息 c。其中，系统调用信息 a 与第一进程使用的系统调用 p 相对应，逃逸检测组件在 14:00 接收到该系统调用信息 a，系统调用信息 b 与第一进程使用的系统调用 q 相对应，逃逸检测组件在 14:01 接收到该系统调用信息 b，系统调用信息 c 与第一进程使用的系统调用 k 相对应，逃逸检测组件在 14:02 接收到该系统调用信息 c。则逃逸检测组件可以根据接收到该 3 个系统调用信息的顺序获取该 3 个系统调用信息对应的 3 个系统调用的顺序，该顺序为：系统调用 p、系统调用 q 和系统调用 k，而后，逃逸检测组件可以根据该顺序生成目标顺序信息，其中，目标顺序信息所指示的顺序即为：系统调用 p、系统调用 q 和系统调用 k。

步骤 203、逃逸检测组件根据目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性。

其中，该预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的顺序。实际实现时，该预设系统调用顺序可以由技术人员根据已知的容器逃逸事件中容器的恶意进程使用宿主机中多个系统调用的顺序进行设定。例如，比较常见的一种容器逃逸攻击是 shocker 攻击，那么技术人员可以根据发生 shocker 攻击时，恶意进程使用系统调用的顺序设定该预设系统调用顺序。

如上所述，在容器逃逸发生的过程中，恶意进程往往需要按照一定的顺序使用宿主机中的系统调用，因此，逃逸检测组件可以根据第一进程使用第一宿主机中的多个系统调用的顺序（也即是目标顺序信息所指示的顺序）与预设系统调用顺序之间的相似程度来确定第一容器发生容器逃逸的可能性。

通常情况下，目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度越高，第一容器发生容器逃逸的可能性就越高，目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度越低，第一容器发生容器逃逸的可能性就越低。

本申请实施例提供了两种根据目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度确定第一容器发生容器逃逸的可能性的方式，下面，本申请实施例将对这两种方式进行一一说明：

第一种方式：逃逸检测组件基于状态机（英文：Finite-state machine）确定目标顺序信

息所指示的顺序与预设系统调用顺序之间的相似程度，并根据该相似程度确定第一容器发生容器逃逸的可能性。

其中，状态机是一种数学模型，该数学模型可以包括有限个状态以及这些状态之间相互转移所需满足的状态转移条件。其中，该有限个状态可以包括初始状态和最终状态，或者，该有限个状态可以包括初始状态、最终状态和至少一个中间状态，其中，状态机可以在转移到最终状态时进行相应的信息输出。

在本申请实施例中，逃逸检测组件可以根据第一进程的标识查询状态机数据库，当该状态机数据库中存储有第一进程的标识对应的状态机时，逃逸检测组件可以将该状态机获取为目标状态机，当该状态机数据库中存储有第一进程的标识对应的状态机时，逃逸检测组件可以根据该预设系统调用顺序创建状态机，并将创建的状态机获取为目标状态机。其中，该状态机数据库中存储有至少一组状态机与进程标识的对应关系，该目标状态机包括初始状态、最终状态以及由初始状态转移至最终状态所需满足的状态转移条件，该状态转移条件为输入的顺序信息所指示的顺序与预设系统调用顺序相同的条件。

实际实现时，该目标状态机除了可以包括初始状态和最终状态外，还可以包括至少一个中间状态，相应地，在这种情况下，状态转移条件可以包括多个子条件，其中，该多个子条件可以包括初始状态转移至与该初始状态相邻的中间状态所需满足的子条件、相邻的两个中间状态转移所需满足的子条件以及由与该最终状态相邻的中间状态转移至该最终状态所需满足的子条件。

例如，如图 2B 所示，该目标状态机可以包括初始状态 C、最终状态 Z 以及由初始状态 C 转移至最终状态 Z 所需满足的状态转移条件 T，该状态转移条件 T 为输入的顺序信息所指示的顺序为系统调用 a、b、c、d、e 和 f 的条件，且，预设系统调用顺序为系统调用 a、b、c、d、e 和 f 的顺序。

又例如，如图 2C 所示，该目标状态机可以包括初始状态 C、中间状态 J1、中间状态 J2 和最终状态 Z，该目标状态机还可以包括由初始状态 C 转移至中间状态 J1 所需满足的子条件 T1、由中间状态 J1 转移至中间状态 J2 所需满足的子条件 T2 和由中间状态 J2 转移至最终状态 Z 所需满足的子条件 T3，该子条件 T1、T2 和 T3 组成了状态转移条件，其中，该子条件 T1 为输入的顺序信息所指示的顺序中的前两个系统调用为 a 和 b 的条件，该子条件 T2 为输入的顺序信息所指示的中间两个系统调用为 c 和 d 的条件，该子条件 T3 为输入的顺序信息所指示的后两个系统调用为 e 和 f 的条件，且，预设系统调用顺序为系统调用 a、b、c、d、e 和 f 的顺序。

在获取了目标状态机后，逃逸检测组件可以基于该目标状态机确定目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，并根据该相似程度确定第一容器发生容器逃逸的可能性。

可选的，在目标状态机处于初始状态时，逃逸检测组件可以将目标顺序信息输入至该目标状态机，当目标状态机由初始状态转移至最终状态时，说明目标顺序信息所指示的顺序与预设系统调用顺序相同，因此，逃逸检测组件可以确定目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度为 100%，在这种情况下，逃逸检测组件可以确定第一容器发生了容器逃逸。

第二种方式、逃逸检测组件根据第一容器拥有的系统权限以及目标顺序信息所指示的

顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性。

如上所述，在容器逃逸发生的过程中，容器中的恶意进程往往需要按照一定的顺序使用宿主机中的系统调用，而只有在容器拥有某些系统权限的前提下，容器中的进程才能使用宿主机中的系统调用。因此，在本申请实施例，逃逸检测组件可以根据第一容器拥有的系统权限以及目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性。需要指出的是，在 Linux 系统中，系统权限也可以称为能力（英文：capability）。

可选的，目标容器监测组件可以通过 docker socket 接口获取第一容器的启动参数，该启动参数为第一宿主机创建该第一容器时采用的 docker run 命令中包括的参数，该启动参数可以包括至少一个权限信息，其中，每个权限信息用于指示第一容器拥有的一种系统权限，每个权限信息也指示第一容器启用的一种系统能力。目标容器监测组件可以将该第一容器的启动参数发送至逃逸检测组件中，以使该逃逸检测组件获取第一容器拥有的系统权限（系统能力），并根据第一容器拥有的系统权限以及目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性。

可选的，逃逸检测组件中可以设置有多个权限提取插件（英文：plugin），每个权限提取插件可以与一个目标权限信息相对应，每个权限提取插件用于从第一容器的启动参数包括的至少一个权限信息中提取与该权限提取插件对应的目标权限信息，其中，每个目标权限信息所指示的系统权限为发生容器逃逸的容器所拥有的系统权限。

在提取了至少一个目标权限信息后，逃逸检测组件可以根据该至少一个目标权限信息以及目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性。

可选的，逃逸检测组件可以获取目标决策树，该目标决策树可以包括至少一个分支，每个分支包括有序的多个节点。该多个节点中的每个节点与一个可能性标识对应，该可能性标识用于指示发生容器逃逸的可能性。该多个节点包括系统权限节点和与系统权限节点相连的有序的多个系统调用节点，其中，系统权限节点与一个系统权限对应，每个系统调用节点与一个系统调用对应，与多个系统调用节点所对应的多个系统调用的顺序与预设系统调用顺序相同。本申请中，系统权限节点也可称为系统能力节点。

例如，如图 2D 所示，目标决策树可以包括两个分支 1 和 2，其中，分支 1 包括根节点 g、节点 a1、节点 a2、节点 a3、节点 a4、节点 a5 和节点 a6，分支 2 包括根节点 g、节点 b1、节点 b2、节点 b3、节点 b4、节点 b5 和节点 b6。其中，节点 a1 为系统权限节点，其与系统权限 dac_override_capability 对应，其对应的可能性标识为 4，节点 a2 为系统调用节点，其与系统调用 open_by_handle_at 对应，其对应的可能性标识为 5，节点 a3 为系统调用节点，其与系统调用 fstat 对应，其对应的可能性标识为 5，节点 a4 为系统调用节点，其与系统调用 fcntl 对应，其对应的可能性标识为 6，节点 a5 为系统调用节点，其与系统调用 getdents 对应，其对应的可能性标识为 6，节点 a6 为系统调用节点，其与系统调用 open_by_handle_at 对应，其对应的可能性标识为 8。节点 b1 为系统权限节点，其与系统权限 dac_read_search_capability 对应，其对应的可能性标识为 4，节点 b2 为系统调用节点，其与系统调用 open_by_handle_at 对应，其对应的可能性标识为 5，节点 b3 为系统调用节点，其与系统调用 fstat 对应，其对应的可能性标识为 5，节点 b4 为系统调用节点，其与系统调用

fcntl 对应，其对应的可能性标识为 6，节点 b5 为系统调用节点，其与系统调用 getdents 对应，其对应的可能性标识为 6，节点 b6 为系统调用节点，其与系统调用 open_by_handle_at 对应，其对应的可能性标识为 8。其中，预设系统调用的顺序为：open_by_handle_at、fstat、fcntl、getdents 和 open_by_handle_at，可能性标识的取值范围可以为[0,10]，其中，可能性标识的数值越大说明发生容器逃逸的可能性越高。在获取了目标决策树之后，逃逸检测组件可以从该目标决策树包括的至少一个分支中确定至少一个目标分支，该至少一个目标分支与该逃逸检测组件从第一容器的启动参数包括的至少一个权限信息中提取的至少一个目标权限信息一一对应，其中，每个目标分支包括的系统权限节点所对应的系统权限与该目标分支对应的目标权限信息所指示的系统权限相同。

例如，当逃逸检测组件从第一容器的启动参数包括的至少一个权限信息中提取的目标权限信息所指示的系统权限为 dac_read_search_capability 时，逃逸检测组件可以将图 2D 中的分支 2 确定为目标分支。

在确定了至少一个目标分支后，逃逸检测组件可以基于该至少一个目标分支确定第一容器发生容器逃逸的可能性。

可选的，对于每个目标分支，当目标顺序信息所指示的多个系统调用中的前 n 个系统调用与该目标分支中包括的多个系统调用节点中的前 n 个系统调用节点所对应的 n 个系统调用的顺序一致时，逃逸检测组件可以获取目标分支中的第 n 个系统调用节点所对应的可能性标识，其中，n 为大于或等于 1 的正整数。而后，逃逸检测组件可以根据该可能性标识确定第一容器发生容器逃逸的可能性。

例如，目标顺序信息所指示的顺序可以为：open_by_handle_at、fstat、fcntl 和 open_by_handle_at，其前 3 个系统调用与分支 2 中的前 3 个系统调用节点所对应的 3 个系统调用的顺序（open_by_handle_at、fstat 和 fcntl）一致，则逃逸检测组件可以获取分支 2 中的第 3 个系统调用节点所对应的可能性标识 6，逃逸检测组件可以根据该可能性标识确定第一容器发生容器逃逸的可能性。

需要指出的是，本申请实施例中逃逸检测组件还可以根据目标容器监测组件上报的其他信息确定第一容器发生容器逃逸的可能性。

在一种可能的实现方式中，目标容器监测组件可以向逃逸检测组件上报敏感文件篡改信息，逃逸检测组件可以根据该敏感文件篡改信息确定第一容器发生容器逃逸的可能性。

可选的，逃逸检测组件可以将敏感文件配置信息发送至该目标容器监测组件中，其中，该敏感文件配置信息可以包括第一宿主机中预设的至少一个敏感文件的标识。目标容器监测组件可以根据该敏感文件配置信息确定第一宿主机中预设的至少一个敏感文件，并对该至少一个敏感文件进行写操作的监控。当监测到第一容器中的第二进程（该第二进程可以是第一进程，也可以是与第一进程不同的进程）对该至少一个敏感文件中的任一敏感文件进行写操作后，目标容器监测组件可以向逃逸检测组件发送敏感文件篡改信息。其中，在 Linux 系统中，目标容器监测组件可以基于 inotify 模块中的观测器 watcher 对第一宿主机中预设的至少一个敏感文件进行写操作的监控。

如上所述，容器中的恶意进程在进行容器逃逸攻击时会篡改宿主机中的敏感文件，因此，当逃逸检测组件接收到目标容器监测组件发送的敏感文件篡改信息后，该逃逸检测组件即可确定第一容器发生了容器逃逸。

实际实现时，该敏感文件配置信息还可以包括第一容器中预设的至少一个敏感文件的标识，目标容器监测组件可以根据该敏感文件配置信息确定第一容器中预设的至少一个敏感文件，并对该第一容器中至少一个敏感文件进行写操作的监控。当监测到第一容器中的第二进程对该第一容器中至少一个敏感文件中的任一敏感文件进行写操作后，目标容器监测组件也可以向逃逸检测组件发送敏感文件篡改信息。在这种情况下，逃逸检测组件接收到该敏感文件篡改信息后，可以确定第一容器中可能存在恶意进程，因此，此时逃逸检测组件可以确定第一容器可能会发生容器逃逸。

在另一种可能的实现方式中，目标容器监测组件可以向逃逸检测组件发送非法进程运行信息，逃逸检测组件可以根据该非法进程运行信息确定第一容器发生容器逃逸的可能性。

可选的，目标容器监测组件可以向逃逸检测组件发送第一宿主机的容器镜像与进程的对应关系，逃逸检测组件可以将该对应关系发送至客户端，以由客户端将该对应关系展示给用户。用户可以对对应关系进行编辑，例如，用户可以在该对应关系中增加某些进程，或者，删除某些进程。客户端可以将经过用户编辑的对应关系发送至逃逸检测组件，逃逸检测组件接收到该对应关系后，可以基于该对应关系中包括的进程生成进程白名单，该进程白名单可以包括多个合法进程的进程标识。逃逸检测组件可以将该进程白名单发送至目标容器监测组件。

目标容器监测组件可以对第一容器中运行的进程进行监控，当目标容器监测组件监测到第一容器中运行的第三进程（该第三进程可以是第一进程，也可以是与第一进程不同的进程）的进程标识不属于进程白名单包括的进程标识后，该目标容器监测组件可以向逃逸检测组件发送非法进程运行信息。逃逸检测组件接收到该非法进程运行信息后，可以确定第一容器中存在非法进程，由于该非法进程很可能为恶意进程，因此，逃逸检测组件在接收到该非法进程运行信息后，可以确定第一容器可能发生了容器逃逸。

需要指出的是，在 Linux 系统中，目标容器监测组件可以利用 cAdvisor 模块对第一容器中运行的进程进行监控。

步骤 204、逃逸检测组件在确定第一容器发生容器逃逸的可能性后，生成并向服务组件发送容器逃逸事件报告。

该容器逃逸事件报告可以包括第一宿主机中的第一容器发生容器逃逸的可能性。

服务组件在接收到该容器逃逸事件报告后，可以将接收到的容器逃逸事件报告存储于存储组件中。

客户端可以在用户的触发下向服务组件发送查询请求，以查询第一容器的容器逃逸事件报告，服务组件可以在接收到客户端发送的查询请求后，从存储组件中获取第一容器的容器逃逸事件报告，并将该容器逃逸事件报告发送至该客户端。

综上所述，本申请实施例提供的容器逃逸检测方法，通过逃逸检测组件根据第一容器中的第一进程使用第一宿主机中的多个系统调用的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性，其中，预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的顺序，使得逃逸检测组件可以对容器运行后发生的容器逃逸事件进行检测，从而提高了对容器逃逸的防范力度。

下述为本申请的装置实施例，可以用于执行本申请的方法实施例。对于本申请装置实

施例中未披露的细节，请参照本申请方法实施例。

请参考图 3，其示出了本申请实施例提供的一种容器逃逸检测装置 300 的框图，该容器逃逸检测装置 300 可以应用于图 1B 或图 1C 所示的逃逸检测组件中。参见图 3，该容器逃逸检测装置 300 可以包括第一接收模块 301、获取模块 302 和第一确定模块 303。

5 第一接收模块 301，用于接收目标容器监测组件发送的多个系统调用信息，每个系统调用信息是该目标容器监测组件在监测到第一容器中的第一进程使用第一宿主机中的系统调用后发送的，该系统调用信息携带该第一进程所使用的系统调用的标识，该目标容器监测组件是该至少一个容器监测组件中的任一容器监测组件，该第一宿主机是该目标容器监测组件监测的宿主机，该第一容器是该第一宿主机中部署的一个容器。

10 获取模块 302，用于根据该多个系统调用信息获取目标顺序信息，该目标顺序信息用于指示该第一进程使用该第一宿主机中的多个系统调用的顺序。

第一确定模块 303，用于根据该目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定该第一容器发生容器逃逸的可能性，该预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的顺序。

15 在本申请的一个实施例中，该第一确定模块 303，包括：

获取子模块，用于获取目标状态机，该目标状态机包括初始状态、最终状态以及由该初始状态转移至该最终状态所需满足的状态转移条件，该状态转移条件为输入的顺序信息所指示的顺序与该预设系统调用顺序相同的条件；

20 输入子模块，用于在该目标状态机处于该初始状态时，将该目标顺序信息输入至该目标状态机；

第一确定子模块，用于在该目标状态机由该初始状态转移至该最终状态时，确定该第一容器发生了容器逃逸。

在本申请的一个实施例中，该第一确定模块 303，包括：

25 接收子模块，用于接收该目标容器监测装置发送的该第一容器的启动参数，该启动参数包括至少一个权限信息，每个权限信息用于指示该第一容器拥有的一种系统权限；

第二确定子模块，用于根据该启动参数以及该目标顺序信息所指示的顺序与该预设系统调用顺序之间的相似程度，确定该第一容器发生容器逃逸的可能性。

在本申请的一个实施例中，该第二确定子模块，用于：

30 从该启动参数包括的至少一个权限信息中获取至少一个目标权限信息，每个目标权限信息所指示的系统权限为发生容器逃逸的容器所拥有的系统权限；

根据该至少一个目标权限信息以及该目标顺序信息所指示的顺序与该预设系统调用顺序之间的相似程度，确定该第一容器发生容器逃逸的可能性。

在本申请的一个实施例中，该第二确定子模块，用于：

35 获取目标决策树，该目标决策树包括至少一个分支，每个分支包括有序的多个节点，该多个节点中的每个节点与一个可能性标识对应，该可能性标识用于指示发生容器逃逸的可能性，该多个节点包括系统权限节点和与该系统权限节点相连的有序的多个系统调用节点，该系统权限节点与一个系统权限对应，每个系统调用节点与一个系统调用对应，与该多个系统调用节点所对应的多个系统调用的顺序与该预设系统调用顺序相同；

从该至少一个分支中确定至少一个目标分支，该至少一个目标分支与该至少一个目标

权限信息一一对应，每个目标分支包括的系统权限节点所对应的系统权限与该目标分支对应的目标权限信息所指示的系统权限相同；

对于每个目标分支，当该目标顺序信息所指示的多个系统调用中的前 n 个系统调用与该目标分支中包括的多个系统调用节点中的前 n 个系统调用节点所对应的 n 个系统调用的顺序一致时，获取该目标分支中的第 n 个系统调用节点所对应的可能性标识， n 为大于或等于 1 的正整数；

根据该可能性标识确定该第一容器发生容器逃逸的可能性。

本申请实施例还提供了另一种容器逃逸检测装置 400，如图 4 所示，该容器逃逸检测装置 400 除了可以包括容器逃逸检测装置 300 包括的各个模块外，还可以包括第二接收模块 304、第二确定模块 305、第三接收模块 306 和第三确定模块 307。

第二接收模块 304，用于接收该目标容器监测组件发送的敏感文件篡改信息，该敏感文件篡改信息是该目标容器监测组件在监测到该第一容器中的第二进程对该第一宿主机中预设的至少一个敏感文件中的任一敏感文件进行写操作后发送的。

第二确定模块 305，用于基于该敏感文件篡改信息，确定该第一容器发生了容器逃逸。

第三接收模块 306，用于接收该目标容器监测组件发送的非法进程运行信息，该非法进程运行信息是该目标容器监测组件在监测到该第一容器中运行的第三进程的进程标识不属于进程白名单包括的进程标识后发送的，该进程白名单包括多个合法进程的进程标识。

第三确定模块 307，用于基于该非法进程运行信息，确定该第一容器可能发生了容器逃逸。

综上所述，本申请实施例提供的容器逃逸检测装置，通过根据第一容器中的第一进程使用第一宿主机中的多个系统调用的顺序与预设系统调用顺序之间的相似程度，确定第一容器发生容器逃逸的可能性，其中，预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的顺序，从而可以对容器运行后发生的容器逃逸事件进行检测，提高到了对容器逃逸的防范力度。

请参考图 5，其示出了本申请实施例提供的一种容器逃逸检测装置 500 的框图，该容器逃逸检测装置 500 可以应用于图 1B 或图 1C 所示的容器监测组件中。参见图 5，该容器逃逸检测装置 500 可以包括生成模块 501 和第一发送模块 502。

生成模块 501，用于在监测到第一容器中的第一进程使用第一宿主机中的系统调用后，生成多个系统调用信息，每个系统调用信息携带该第一进程所使用的系统调用的标识，该第一宿主机是该目标容器监测组件监测的宿主机，该第一容器是该第一宿主机中部署的一个容器。

第一发送模块 502，用于向该逃逸检测组件发送该多个系统调用信息，该多个系统调用信息用于供该逃逸检测组件根据该多个系统调用信息获取目标顺序信息，并根据该目标顺序信息所指示的顺序与预设系统调用顺序之间的相似程度，确定该第一容器发生容器逃逸的可能性，该目标顺序信息用于指示该第一进程使用该第一宿主机中的多个系统调用的顺序，该预设系统调用顺序为发生容器逃逸时，容器中的恶意进程使用宿主机中多个系统调用的顺序。

本申请实施例还提供了另一种容器逃逸检测装置 600，如图 6 所示，该容器逃逸检测装

置 600 除了可以包括容器逃逸检测装置 500 包括的各个模块外, 还可以包括第二发送模块 503、第三发送模块 504 和第四发送模块 505。

第二发送模块 503, 用于向该逃逸检测组件发送该第一容器的启动参数, 该启动参数包括至少一个权限信息, 每个权限信息用于指示该第一容器拥有的一个系统权限, 该启动参数用于供该逃逸检测组件根据该启动参数以及该目标顺序信息所指示的顺序与该预设系统调用顺序之间的相似程度, 确定该第一容器发生容器逃逸的可能性。

第三发送模块 504, 用于在监测到该第一容器中的第二进程对该第一宿主机中预设的至少一个敏感文件中的任一敏感文件进行写操作后, 向该逃逸检测组件发送敏感文件篡改信息, 该敏感文件篡改信息用于供该逃逸检测组件基于该敏感文件篡改信息确定该第一容器发生了容器逃逸。

第四发送模块 505, 用于在监测到该第一容器中运行的第三进程的进程标识不属于进程白名单包括的进程标识后, 向该逃逸检测组件发送非法进程运行信息, 该非法进程运行信息用于供该逃逸检测组件基于该非法进程运行信息确定该第一容器可能发生了容器逃逸, 该进程白名单包括多个合法进程的进程标识。

综上所述, 本申请实施例提供的容器逃逸检测装置, 通过在第一容器中的第一进程使用第一宿主机中的系统调用后向逃逸检测组件发送系统调用信息, 使得逃逸检测组件根据第一容器中的第一进程使用第一宿主机中的多个系统调用的顺序与预设系统调用顺序之间的相似程度, 确定第一容器发生容器逃逸的可能性, 其中, 预设系统调用顺序为发生容器逃逸时, 容器中的恶意进程使用宿主机中多个系统调用的顺序, 从而可以对容器运行后发生的容器逃逸事件进行检测, 提高到了对容器逃逸的防范力度。

本申请实施例提供了一种容器逃逸检测装置 700, 该容器逃逸检测装置 700 可以为服务器, 或者其它可以部署容器的计算设备。该容器逃逸检测装置 700 可以用于执行上述方法实施例中提供的容器逃逸检测方法。

参见图 7, 该容器逃逸检测装置 700 包括处理器, 例如中央处理单元(central processing unit, CPU) 701。该容器逃逸检测装置 700 包括存储器, 例如随机存取存储器(random-access memory, RAM) 702 和只读存储器(read-only memory, ROM) 703 的系统存储器 704。该容器逃逸检测装置 700 包括连接系统存储器 704 和中央处理单元 701 的系统总线 705。该容器逃逸检测装置 700 还包括帮助计算机内的各个器件之间传输信息的基本输入/输出系统(input/output, I/O) 系统 706, 和用于存储操作系统 713、应用程序 714 和其他程序模块 715 的大容量存储设备 707。

该基本输入/输出系统 706 包括有用于显示信息的显示器 708 和用于用户输入信息的诸如鼠标、键盘之类的输入设备 709。其中该显示器 708 和输入设备 709 都通过连接到系统总线 705 的输入输出控制器 710 连接到中央处理单元 701。该基本输入/输出系统 706 还可以包括输入输出控制器 710 以用于接收和处理来自键盘、鼠标、或电子触控笔等多个其他设备的输入。类似地, 输入输出控制器 710 还提供输出到显示屏、打印机或其他类型的输出设备。

该大容量存储设备 707 通过连接到系统总线 705 的大容量存储控制器(未示出)连接到中央处理单元 701。该大容量存储设备 707 及其相关联的计算机可读介质为容器逃逸检测

装置 700 提供非易失性存储。也就是说, 该大容量存储设备 707 可以包括诸如硬盘或者 CD-ROM 驱动器之类的计算机可读介质 (未示出)。

不失一般性, 该计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。计算机存储介质包括 RAM、ROM、EPROM、EEPROM、闪存或其他固态存储其技术, CD-ROM、DVD 或其他光学存储、磁带盒、磁带、磁盘存储或其他磁性存储设备。当然, 本领域技术人员可知该计算机存储介质不局限于上述几种。上述的系统存储器 704 和大容量存储设备 707 可以统称为存储器。

根据本申请的各种实施例, 该容器逃逸检测装置 700 还可以通过诸如因特网等网络连接到网络上的远程计算机运行。也即容器逃逸检测装置 700 可以通过连接在该系统总线 705 上的网络接口单元 711 连接到网络 712, 或者说, 也可以使用网络接口单元 711 来连接到其他类型的网络或远程计算机系统 (未示出)。

该存储器还包括一个或者一个以上的程序, 该一个或者一个以上程序存储于存储器中, 中央处理器 701 通过执行该一个或一个以上程序来实现上述方法实施例提供的容器逃逸检测方法。

请参考图 8, 其示出了本申请实施例提供的一种容器逃逸检测系统 800 的框图, 该容器逃逸检测系统 800 可以包括逃逸检测组件 801 和至少一个容器监测组件 802 (图 8 中仅示出了一个容器监测组件 802)。

其中, 逃逸检测组件 801 用于执行上述方法实施例中逃逸检测组件所执行的容器逃逸检测方法。

容器监测组件 802 用于执行上述方法实施例中目标容器监测组件所执行的容器逃逸检测方法。

本申请实施例还提供了一种包括指令的非临时性计算机可读存储介质, 例如包括指令的存储器, 上述指令可由处理器执行以完成本申请方法实施例所示的容器逃逸检测方法。例如, 该非临时性计算机可读存储介质可以是 ROM、随机存取存储器 (RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

本申请实施例还提供了一种包含指令的计算机程序产品, 当该计算机程序产品在计算机上运行时, 使得计算机执行本申请方法实施例所示的容器逃逸检测方法。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成, 也可以通过程序来指令相关的硬件完成, 所述的程序可以存储于一种计算机可读存储介质中, 上述提到的存储介质可以是只读存储器, 磁盘或光盘等。

以上所述仅为本申请的可选实施例, 并不用以限制本申请, 凡在本申请的精神和原则之内, 所作的任何修改、等同替换、改进等, 均应包含在本申请的保护范围之内。

权 利 要 求 书

1、一种计算系统中容器逃逸检测方法，其特征在于，所述计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则包括至少一组预设系统调用顺序，每
5 组预设系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；

所述方法包括：

所述逃逸检测组件接收所述容器监测组件上报的被监测容器触发的多个系统调用的信息；

所述逃逸检测组件将所述多个系统调用的发生顺序与所述逃逸检测规则中的至少一组预
10 设系统调用顺序进行匹配，并根据匹配结果确定所述被监测容器是否发生逃逸。

2、根据权利要求1所述的方法，其特征在于，所述逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力；

所述方法还包括：

所述逃逸检测组件接收所述容器监测组件上报的被监测容器的启动参数，所述启动参数
15 包含所述被监测容器启用的系统能力；

所述逃逸检测组件根据预设的所述逃逸检测规则以及所述被监测容器的启用的系统能力，确定所述被监测容器是否发生逃逸。

3、根据权利要求1或2所述的方法，其特征在于，还包括：

所述逃逸检测组件接收所述容器监测组件上报的敏感文件篡改信息，所述敏感文件篡改
20 信息是所述容器监测组件在监测到所述被监测容器中的进程对所述被监测容器的宿主机中预设的敏感文件进行写操作后发送的；

所述逃逸检测组件基于所述敏感文件篡改信息，确定所述被监测容器发生了容器逃逸。

4、根据权利要求1-3任一项所述的方法，其特征在于，还包括：

所述逃逸检测组件根据所述被监测容器是否发生逃逸，发送容器逃逸事件报告或逃逸告
25 警。

5、根据权利要求1-4任一项所述的方法，其特征在于，所述逃逸检测规则采用目标决策树表达，所述目标决策树包括至少一个分支，每个分支包括系统能力节点和有序的多个系统调用节点，每个分支的系统能力节点对应于一种高危系统能力，每个分支的多个系统调用节点的预设顺序代表启用该分支的系统能力节点所描述的高危系统能力的容器发生逃逸时所触发的多个系统调用的预测发生顺序，所述目标决策树的每个分支的每个节点上都预先设置有可能性标识，所述可能性标识的大小指示容器逃逸的可能性大小；

所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括：

所述逃逸检测组件在接收到所述被监测容器的启动参数后，根据获取到的所述被监测容器的启用的系统能力，与所述目标决策树中的各个系统能力节点进行匹配，获取匹配成功的
35 系统能力节点对应的第一可能性标识，将所述第一可能性标识作为所述被监测容器当前的可能性标识；

所述逃逸检测组件在接收到所述被监测容器触发的多个系统调用的信息后，确定所述多个系统调用的实际发生顺序，将所述多个系统调用的实际发生顺序与匹配成功的系统能力节点所在的分支的预设顺序进行匹配，若在匹配成功的系统能力节点所在的分支的第n个系统

调用节点匹配成功, 获取所述第 n 个系统调用节点所对应的第二可能性标识, 使用所述第二可能性标识更新所述被监测容器当前的可能性标识。

6、根据权利要求 1-4 任一项所述的方法, 其特征在于, 所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括:

- 5 获取目标状态机, 所述目标状态机包括初始状态、最终状态以及由所述初始状态转移至所述最终状态所需满足的状态转移条件, 所述状态转移条件为输入的顺序信息所指示的顺序与所述预设系统调用顺序相同的条件;

在所述目标状态机处于所述初始状态时, 将所述多个系统调用的发生顺序输入至所述目标状态机;

- 10 当所述目标状态机由所述初始状态转移至所述最终状态时, 确定所述被监测容器发生了容器逃逸。

- 7、一种计算系统中容器逃逸检测方法, 其特征在于, 所述计算系统包括至少一台宿主机, 所述宿主机上运行容器监测组件和至少一个容器, 所述计算系统还包括逃逸检测组件, 所述逃逸检测组件设置有逃逸检测规则, 所述逃逸检测规则包括至少一组预设系统调用顺序, 每
15 组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序;

所述方法包括:

所述容器监测组件对被监测容器的运行情况进行监测;

- 20 所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息, 所述多个系统调用的发生顺序用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

8、根据权利要求 7 所述的方法, 其特征在于, 还包括:

所述容器监测组件向所述逃逸检测组件上报所述被监测容器的信息, 所述被监测容器的信息包括所述被监测容器的 ID 或所述被监测容器的启动的进程 ID;

- 25 所述容器监测组件接收所述逃逸检测组件根据所述被监测容器的信息下发的审计规则, 所述审计规则指示所述容器监测组件对所述被监测容器的系统调用事件进行审计;

所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息包括:

所述容器监测组件在监测到所述被监测容器触发的系统调用事件后, 向所述逃逸检测组件上报审计报告, 所述审计报告携带所述被监测容器触发的多个系统调用的信息。

- 30 9、根据权利要求 7 或 8 所述的方法, 其特征在于, 所述逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力;

所述方法还包括:

- 35 所述容器监测组件在监测到所述被监测容器启动后, 向所述逃逸检测组件上报所述被监测容器的启动参数, 所述启动参数包含所述被监测容器启用的系统能力, 所述被监测容器启用的系统能力用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

10、根据权利要求 9 所述的方法, 其特征在于, 所述容器监测组件向所述逃逸检测组件上报被监测容器的启动参数包括:

所述容器监测组件调用 docker socket 接口获取所述被监测容器的启动参数, 向所述逃逸检测组件上报携带所述被监测容器的启动参数的 docker run 命令。

11、根据权利要求 7-10 任一项所述的方法，其特征在于，还包括：

所述容器监测组件向所述逃逸检测组件上报敏感文件篡改信息，所述敏感文件篡改信息是所述容器监测组件在监测到所述被监测容器中的进程对所述被监测容器的宿主机中预设的敏感文件进行写操作后发送的。

5

12、一种计算系统中容器逃逸检测方法，其特征在于，所述计算系统包括至少一台宿主机，所述宿主机上运行容器监测组件和至少一个容器，所述计算系统还包括逃逸检测组件，所述逃逸检测组件设置有逃逸检测规则，所述逃逸检测规则包括预定义的至少一种与容器逃逸相关的高危系统能力；

10 所述方法包括：

所述逃逸检测组件接收所述容器监测组件上报的被监测容器的启动参数，所述启动参数包含所述被监测容器启用的系统能力；

所述逃逸检测组件根据预设的所述逃逸检测规则以及所述被监测容器的启用的系统能力，确定所述被监测容器是否发生逃逸。

15 13、根据权利要求 12 所述的方法，其特征在于，所述逃逸检测规则还包括至少一组预设系统调用顺序，每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序；

所述方法还包括：

20 所述逃逸检测组件接收所述容器监测组件上报的所述被监测容器触发的多个系统调用的信息；

所述逃逸检测组件将所述多个系统调用的发生顺序与所述逃逸检测规则中的至少一组预设系统调用顺序进行匹配，并根据匹配结果确定所述被监测容器是否发生逃逸。

14、根据权利要求 12 或 13 所述的方法，其特征在于，还包括：

25 所述逃逸检测组件接收所述容器监测组件上报的敏感文件篡改信息，所述敏感文件篡改信息是所述容器监测组件在监测到所述被监测容器中的进程对所述被监测容器的宿主机中预设的敏感文件进行写操作后发送的；

所述逃逸检测组件基于所述敏感文件篡改信息，确定所述被监测容器发生了容器逃逸。

15、根据权利要求 12-14 任一项所述的方法，其特征在于，还包括：

30 所述逃逸检测组件根据所述被监测容器是否发生逃逸，发送容器逃逸事件报告或逃逸告警。

16、根据权利要求 12-15 任一项所述的方法，其特征在于，所述逃逸检测规则采用目标决策树表达，所述目标决策树包括至少一个分支，每个分支包括系统能力节点和有序的多个系统调用节点，每个分支的系统能力节点对应于一种高危系统能力，每个分支的多个系统调用节点的预设顺序代表启用该分支的系统能力节点所描述的高危系统能力的容器发生逃逸时所触发的多个系统调用的预测发生顺序，所述目标决策树的每个分支的每个节点上都预先设置有相应的可能性标识，所述可能性标识的大小指示容器逃逸的可能性大小；

所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括：

所述逃逸检测组件在接收到所述被监测容器的启动参数后，根据获取到的所述被监测容器的启用的系统能力，与所述目标决策树中的各个系统能力节点进行匹配，获取匹配成功的

系统能力节点对应的第一可能性标识, 将所述第一可能性标识作为所述被监测容器当前的可能性标识;

所述逃逸检测组件在接收到所述被监测容器触发的多个系统调用的信息后, 确定所述多个系统调用的实际发生顺序, 将所述多个系统调用的实际发生顺序与匹配成功的系统能力节点所在的分支的预设顺序进行匹配, 若在匹配成功的系统能力节点所在的分支的第 n 个系统调用节点匹配成功, 获取所述第 n 个系统调用节点所对应的第二可能性标识, 使用所述第二可能性标识更新所述被监测容器当前的可能性标识。

17、根据权利要求 12-15 任一项所述的方法, 其特征在于, 所述逃逸检测组件确定所述被监测容器是否发生逃逸具体包括:

获取目标状态机, 所述目标状态机包括初始状态、最终状态以及由所述初始状态转移至所述最终状态所需满足的状态转移条件, 所述状态转移条件为输入的顺序信息所指示的顺序与所述预设系统调用顺序相同的条件;

在所述目标状态机处于所述初始状态时, 将所述多个系统调用的发生顺序输入至所述目标状态机;

当所述目标状态机由所述初始状态转移至所述最终状态时, 确定所述被监测容器发生了容器逃逸。

18、一种计算系统中容器逃逸检测方法, 其特征在于, 所述计算系统包括至少一台宿主机, 所述宿主机上运行容器监测组件和至少一个容器, 所述计算系统还包括逃逸检测组件, 所述逃逸检测组件设置有逃逸检测规则, 所述逃逸检测规则还包括预定义的至少一种与容器逃逸相关的高危系统能力;

所述方法包括:

所述容器监测组件对被监测容器的运行情况进行监测;

所述容器监测组件在监测到所述被监测容器启动后, 向所述逃逸检测组件上报所述被监测容器的启动参数, 所述启动参数包含所述被监测容器启用的系统能力, 所述被监测容器启用的系统能力用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

19、根据权利要求 18 所述的方法, 其特征在于, 所述容器监测组件向所述逃逸检测组件上报被监测容器的启动参数包括:

所述容器监测组件调用 docker socket 接口获取所述被监测容器的启动参数, 向所述逃逸检测组件上报携带所述被监测容器的启动参数的 docker run 命令。

20、根据权利要求 18 或 19 所述的方法, 其特征在于, 所述逃逸检测规则还包括至少一组预设系统调用顺序, 每组系统调用顺序对应于一种容器逃逸行为所触发的多个系统调用的预测发生顺序;

所述方法还包括:

所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息, 所述多个系统调用的发生顺序用于与所述逃逸检测规则进行匹配以确定所述被监测容器是否发生逃逸。

21、根据权利要求 20 所述的方法, 其特征在于, 还包括:

所述容器监测组件向所述逃逸检测组件上报所述被监测容器的信息, 所述被监测容器的信息包括所述被监测容器的 ID 或所述被监测容器的启动的进程 ID;

所述容器监测组件接收所述逃逸检测组件根据所述被监测容器的信息下发的审计规则，所述审计规则指示所述容器监测组件对所述被监测容器的系统调用事件进行审计；

所述容器监测组件向所述逃逸检测组件上报所述被监测容器触发的多个系统调用的信息包括：

5 所述容器监测组件在监测到所述被监测容器触发的系统调用事件后，向所述逃逸检测组件上报审计报告，所述审计报告携带所述被监测容器触发的多个系统调用的信息。

22、根据权利要求 18-21 任一项所述的方法，其特征在于，还包括：

10 所述容器监测组件向所述逃逸检测组件上报敏感文件篡改信息，所述敏感文件篡改信息是所述容器监测组件在监测到所述被监测容器中的进程对所述被监测容器的宿主机中预设的敏感文件进行写操作后发送的。

23、一种检测装置，其特征在于，所述装置用于检测计算系统中的宿主机上运行的容器的逃逸行为，所述装置包括处理器、第一存储器和第二存储器；

所述第二存储器中设置有逃逸检测规则；

15 所述处理器被配置为执行所述第一存储器中存储的指令，所述处理器通过执行所述指令来实现权利要求 1 至 6 任一所述的检测方法。

24、一种检测装置，其特征在于，所述装置用于检测计算系统中的宿主机上运行的容器的逃逸行为，所述装置包括处理器、第一存储器和第二存储器；

所述第二存储器中设置有逃逸检测规则；

20 所述处理器被配置为执行所述第一存储器中存储的指令，所述处理器通过执行所述指令来实现权利要求 12 至 17 任一所述的检测方法。

25、一种监测装置，其特征在于，所述装置用于监测计算系统中的宿主机上运行的容器的运行，所述装置包括处理器和存储器；

所述处理器被配置为执行所述存储器中存储的指令，所述处理器通过执行所述指令来实现权利要求 7 至 11 任一所述的检测方法。

25 26、一种监测装置，其特征在于，所述装置用于监测计算系统中的宿主机上运行的容器的运行，所述装置包括处理器和存储器；

所述处理器被配置为执行所述存储器中存储的指令，所述处理器通过执行所述指令来实现权利要求 18 至 22 任一所述的检测方法。

30 27、一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中存储有指令，当所述指令在计算机上运行时，使得所述计算机执行权利要求 1 至 22 中任一所述的方法。

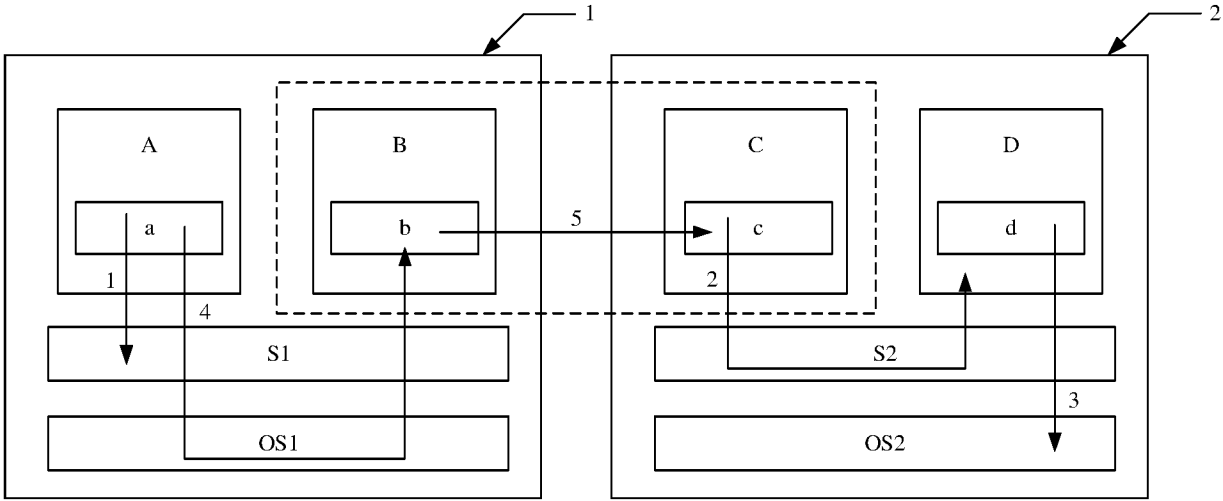


图 1A

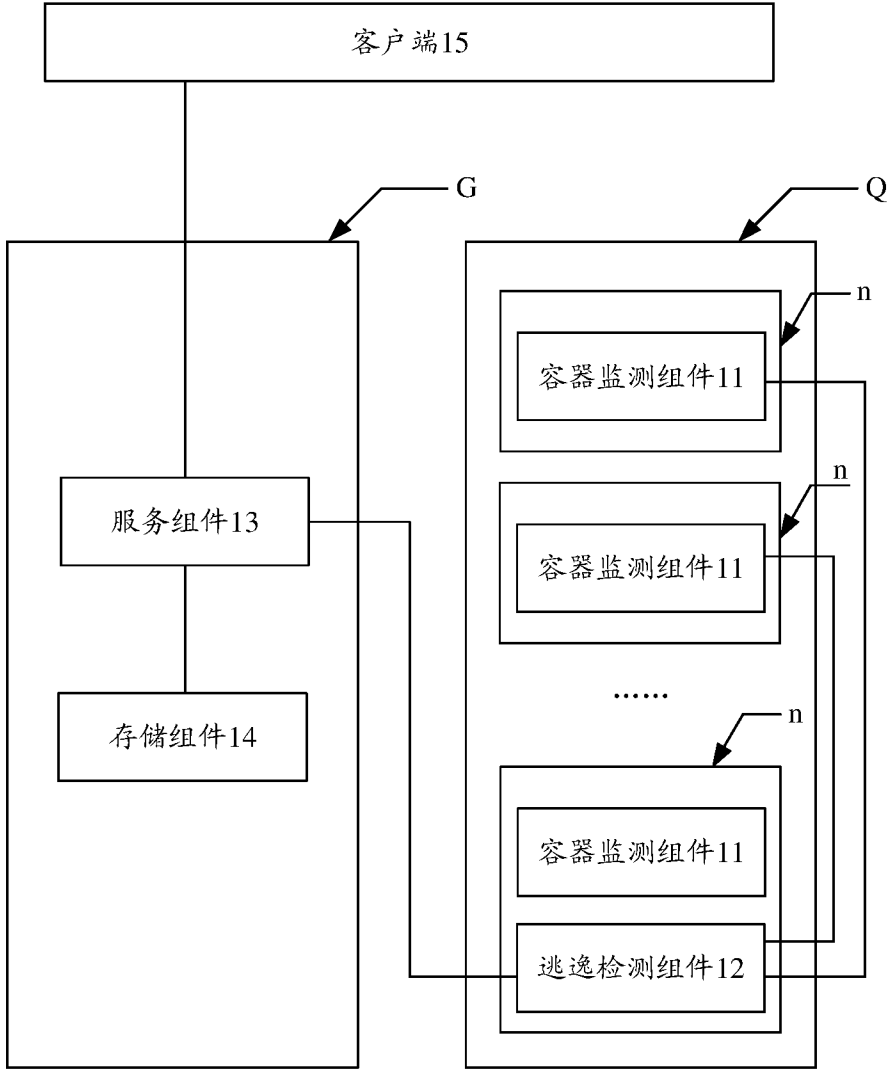


图 1B

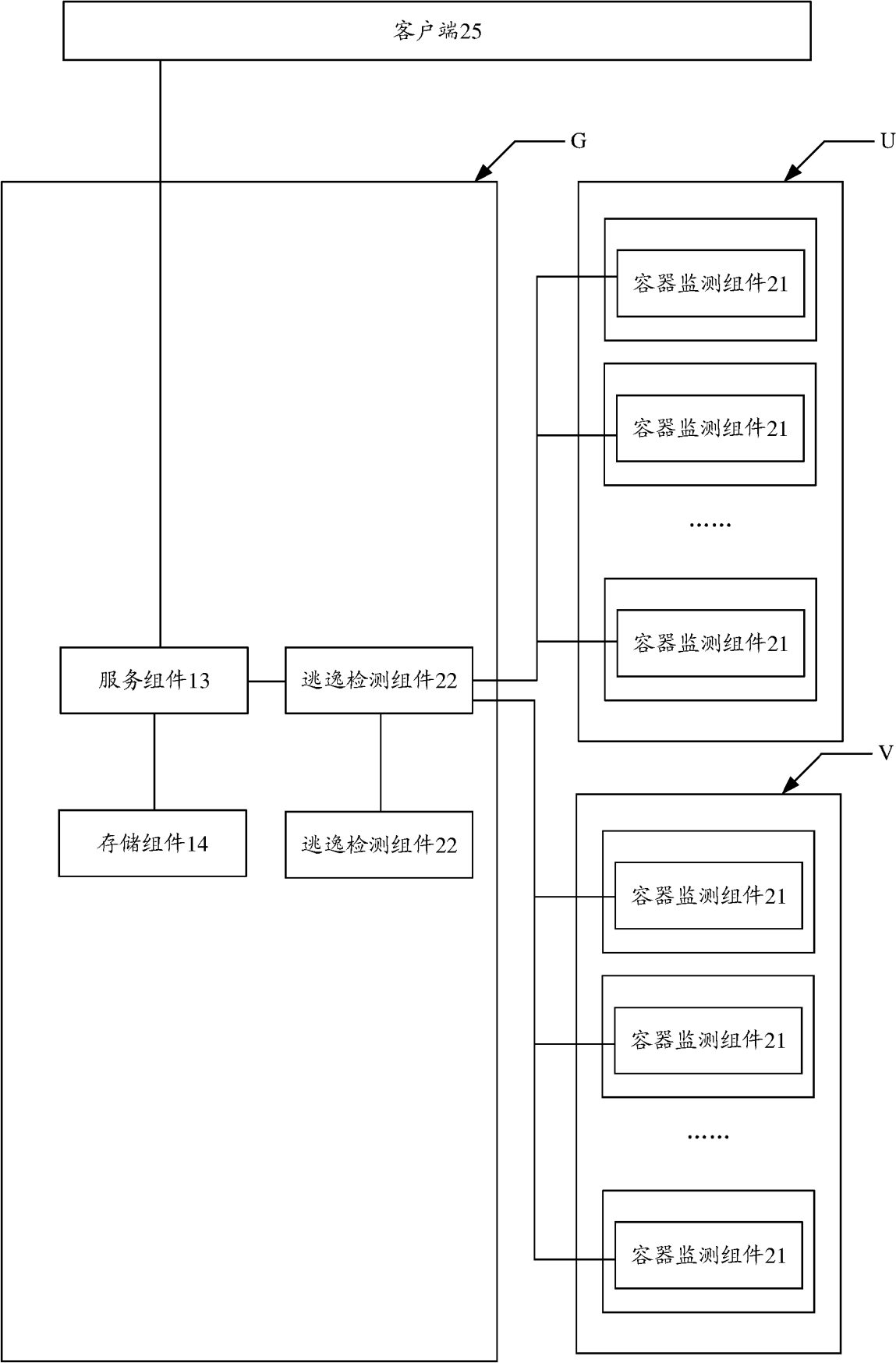


图 1C

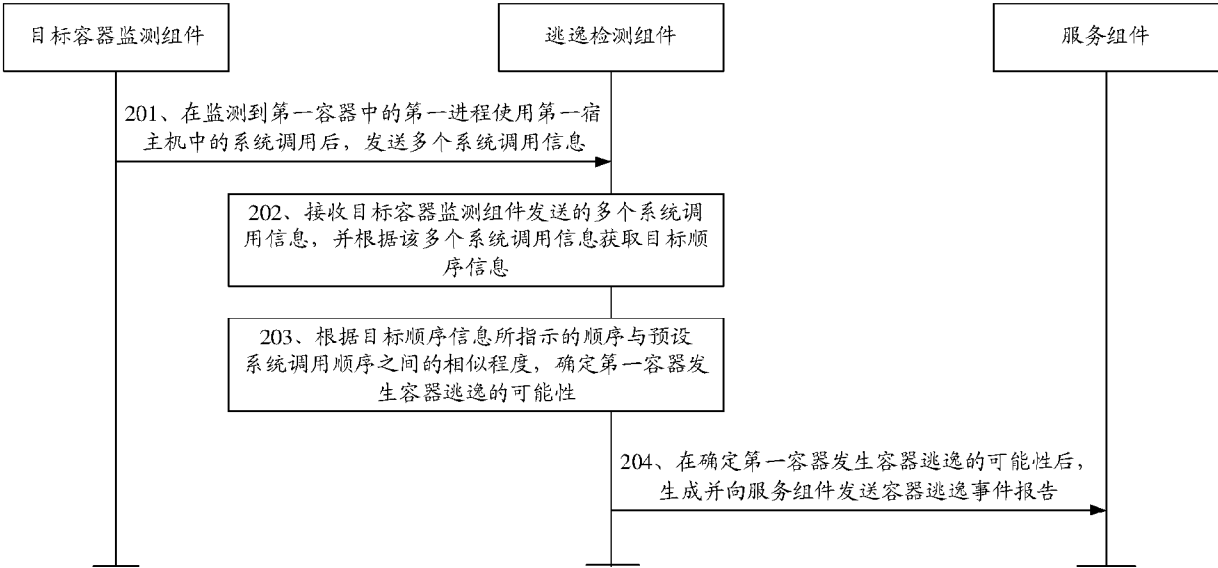


图 2A

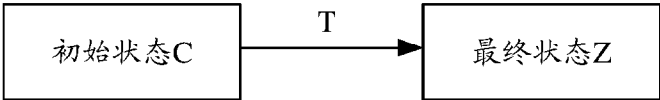


图 2B

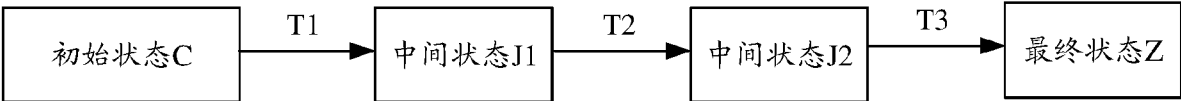


图 2C

4/7

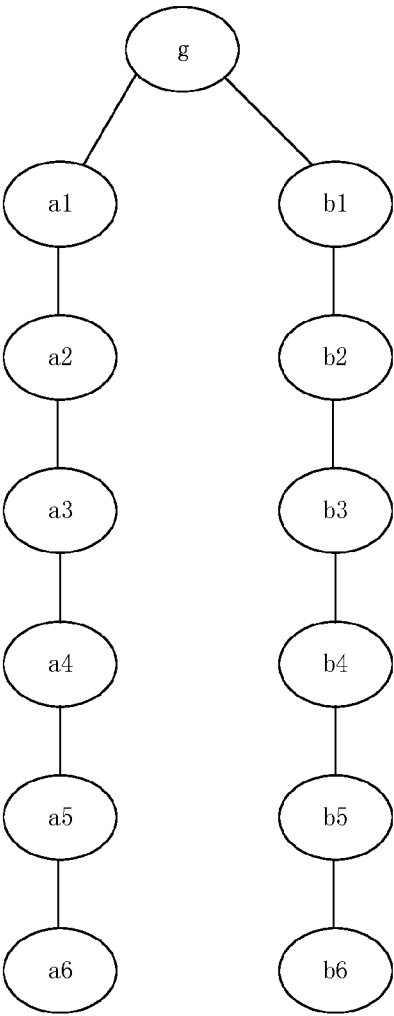


图 2D

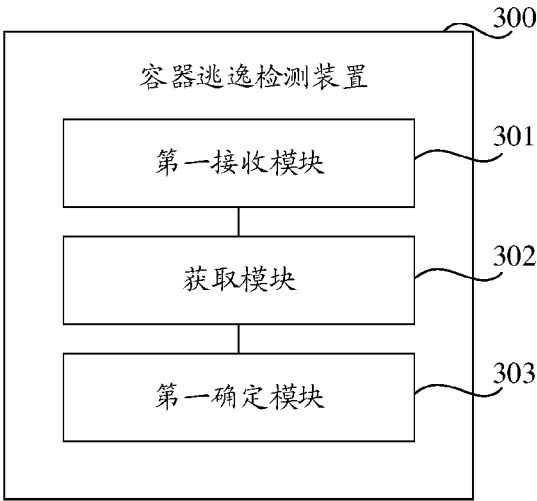


图 3

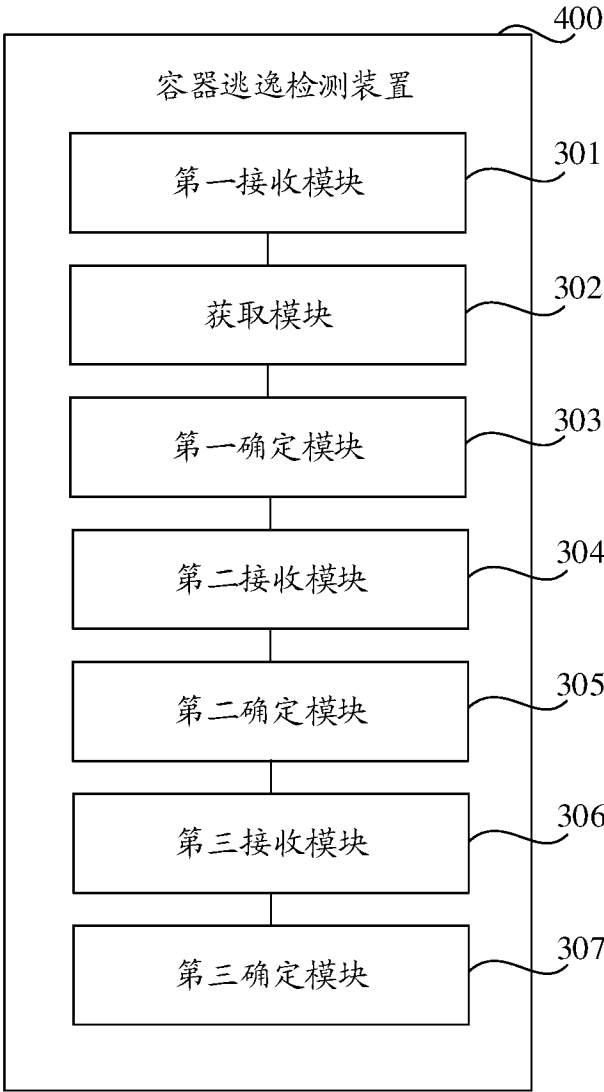


图 4

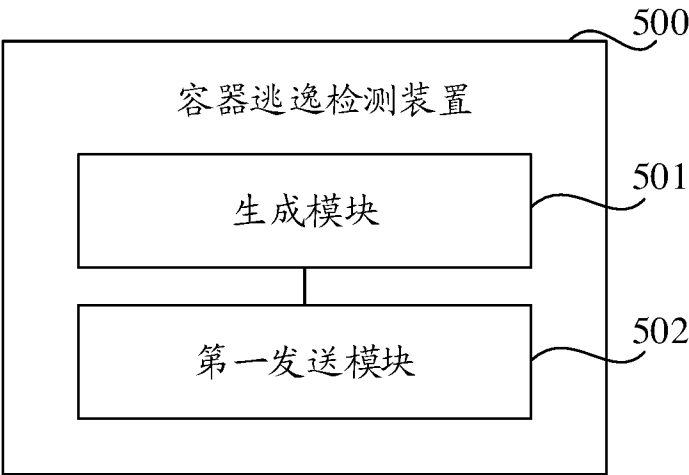


图 5

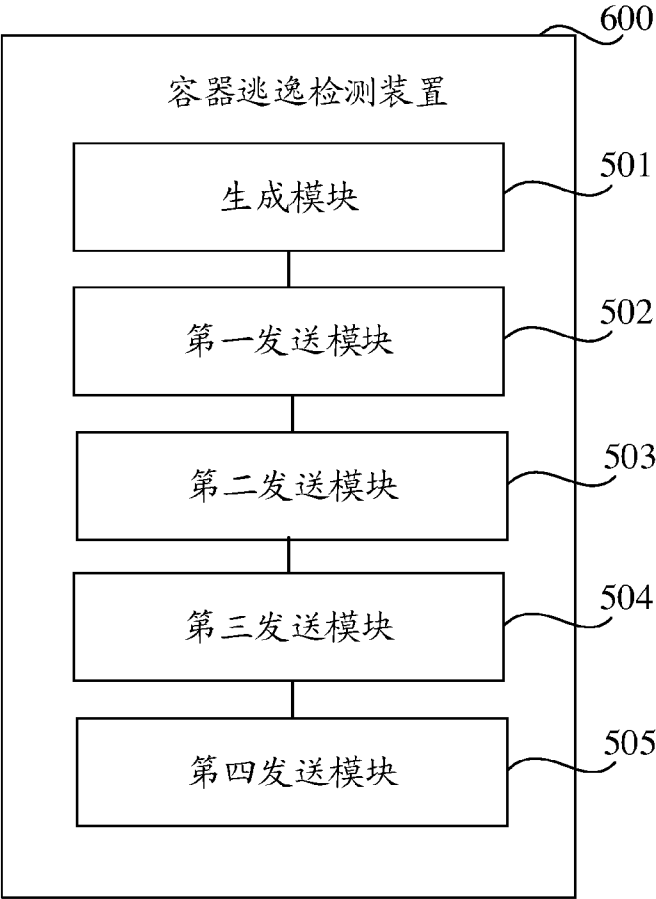


图 6

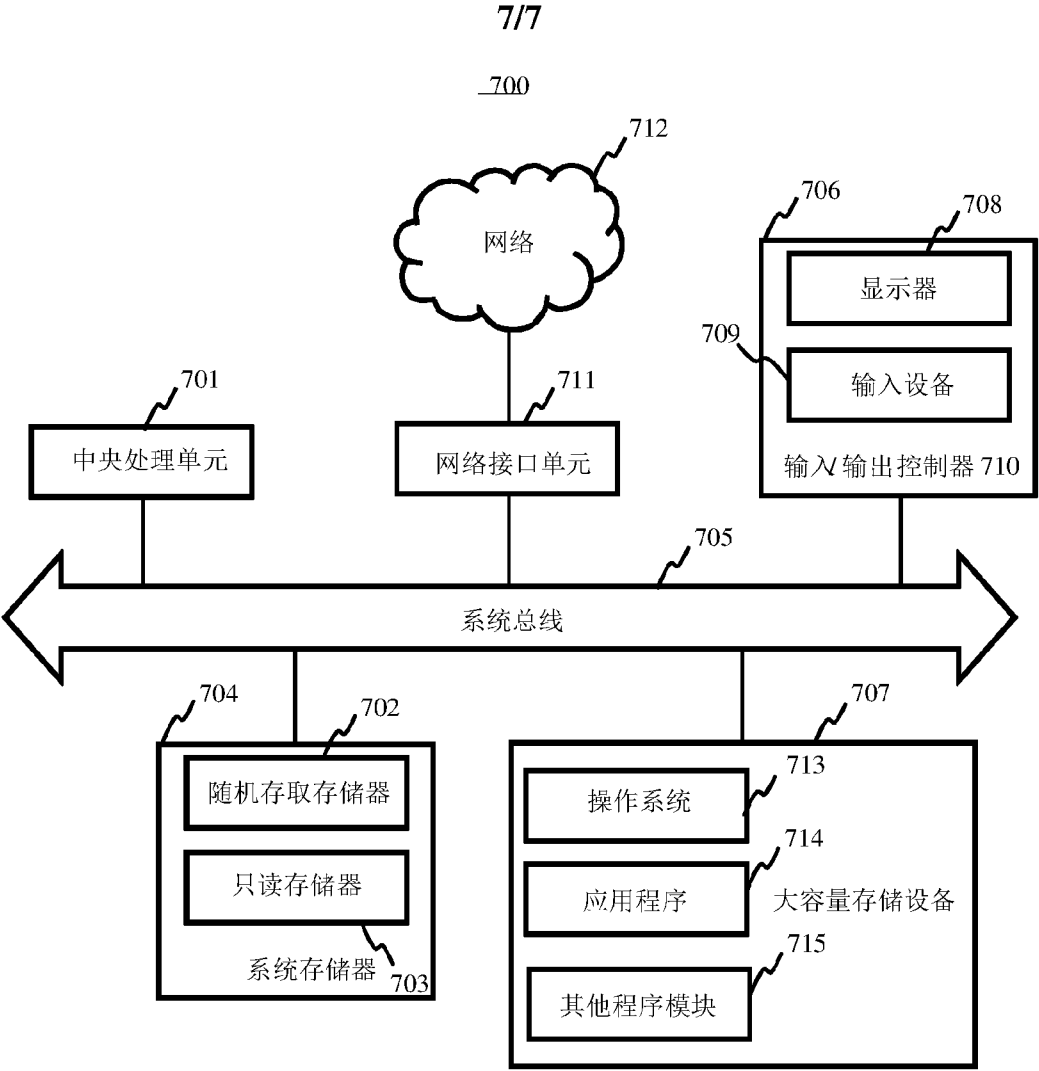


图 7

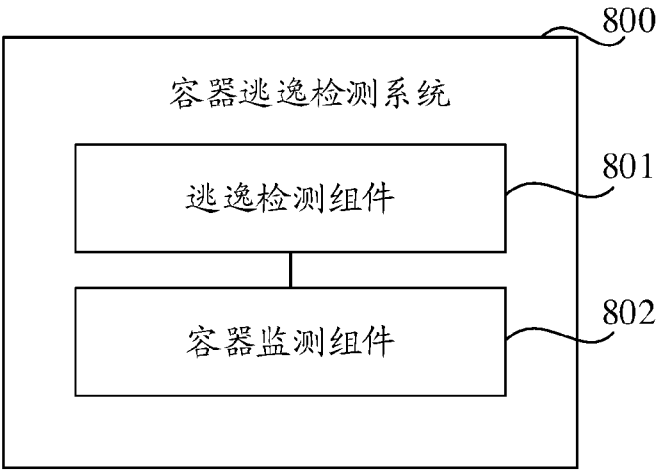


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2018/102667

A. CLASSIFICATION OF SUBJECT MATTER

G06F 21/53(2013.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, EPODOC, WPI, CNKI, IEEE: 容器, 逃逸, 检测, 系统, 权限, 调用, 顺序, 恶意, 攻击, 拦截, 主机, container, escape, detection, system, right, call, sequence, malice, attack, halting, host

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 107679399 A (ZHENGZHOU YUNHAI INFORMATION TECHNOLOGY CO., LTD.) 09 February 2018 (2018-02-09) description, paragraphs [0006]-[0028]	1, 2, 4, 7, 9, 10, 12, 13, 15, 18-20, 23-27
Y	CN 105608374 A (BEIJING QIHOO TECHNOLOGY CO., LTD. ET AL.) 25 May 2016 (2016-05-25) description, paragraphs [0004]-[0017] and [0153]-[0155]	1, 2, 4, 7, 9, 10, 12, 13, 15, 18-20, 23-27
A	CN 106778257 A (BEIJING GUODIANTONG NETWORK TECHNOLOGY CO., LTD. ET AL.) 31 May 2017 (2017-05-31) entire document	1-27
A	CN 104915285 A (BEIJING QIHOO TECHNOLOGY CO., LTD. ET AL.) 16 September 2015 (2015-09-16) entire document	1-27
A	WO 2014116740 A2 (AMAZON TECHNOLOGIES, INC.) 31 July 2014 (2014-07-31) entire document	1-27

☐ Further documents are listed in the continuation of Box C.☒ See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

19 November 2018

Date of mailing of the international search report

28 November 2018

Name and mailing address of the ISA/CN

State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2018/102667

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)		Publication date (day/month/year)
CN	107679399	A	09 February 2018	None		
CN	105608374	A	25 May 2016	None		
CN	106778257	A	31 May 2017	None		
CN	104915285	A	16 September 2015	None		
WO	2014116740	A2	31 July 2014	EP	2949074	A2 02 December 2015
				SG	11201506603P	A 28 August 2015
				KR	20150106935	A 22 September 2015
				CN	104982005	A 14 October 2015
				JP	2016511872	A 21 April 2016
				US	2014208123	A1 24 July 2014
				CA	2898756	A1 31 July 2014
				AU	2014209467	A1 27 August 2015

A. 主题的分类 G06F 21/53 (2013.01) i 按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类		
B. 检索领域 检索的最低限度文献 (标明分类系统和分类号) G06F 包含在检索领域中的除最低限度文献以外的检索文献 在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用)) CNPAT, EPDOC, WPI, CNKI, IEEE: 容器, 逃逸, 检测, 系统, 权限, 调用, 顺序, 恶意, 攻击, 拦截, 主机, container, escape, detection, system, right, call, sequence, malice, attack, halting, host		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
Y	CN 107679399 A (郑州云海信息技术有限公司) 2018年 2月 9日 (2018 - 02 - 09) 说明书第[0006]-[0028]段	1、2、4、7、 9、10、12、13、 15、18-20、23-27
Y	CN 105608374 A (北京奇虎科技有限公司 等) 2016年 5月 25日 (2016 - 05 - 25) 说明书第[0004]-[0017], [0153]-[0155]段	1、2、4、7、 9、10、12、13、 15、18-20、23-27
A	CN 106778257 A (北京国电通网络技术有限公司 等) 2017年 5月 31日 (2017 - 05 - 31) 全文	1-27
A	CN 104915285 A (北京奇虎科技有限公司 等) 2015年 9月 16日 (2015 - 09 - 16) 全文	1-27
A	WO 2014116740 A2 (AMAZON TECHNOLOGIES, INC.) 2014年 7月 31日 (2014 - 07 - 31) 全文	1-27
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件		
国际检索实际完成的日期		国际检索报告邮寄日期
2018年 11月 19日		2018年 11月 28日
ISA/CN的名称和邮寄地址		受权官员
中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451		胡贝贝 电话号码 86-(10)-53961415

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2018/102667

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	107679399	A	2018年 2月 9日	无			
CN	105608374	A	2016年 5月 25日	无			
CN	106778257	A	2017年 5月 31日	无			
CN	104915285	A	2015年 9月 16日	无			
WO	2014116740	A2	2014年 7月 31日	EP	2949074	A2	2015年 12月 2日
				SG	11201505653P	A	2015年 8月 28日
				KR	20150106935	A	2015年 9月 22日
				CN	104982005	A	2015年 10月 14日
				JP	2016511872	A	2016年 4月 21日
				US	2014208123	A1	2014年 7月 24日
				CA	2898756	A1	2014年 7月 31日
				AU	2014209467	A1	2015年 8月 27日