容器隔离方法、装置、系统、电子设备及存储介质

本公开要求申请号为 202410346511.2 的中国专利申请的优先权，该中国专利申请于2024 年 03 月 25 日提交中国专利局，申请名称为“容器隔离方法、装置、系统、电子设备及存储介质”，其全部内容通过引用结合在本公开中。

技术领域

本公开涉及网络安全技术领域，尤其涉及一种容器隔离方法、装置、系统、电子设备及存储介质。

背景技术

在容器集群网络中，整个集群所有容器的网络是互通的，入侵一旦发生，入侵者利用容器网络连通性，会在整个容器集群上横向移动，寻找可进一步利用的入侵点，造成更大范围的入侵。因此，需要对容器按需求进行隔离，来防止恶意流量在容器集群中任意流动。

容器集群网络有如下特点：

1、同一应用的容器的互联网协议（Internet Protocol，IP）地址和数量是不固定的，容器会在集群中的不同宿主机之间漂移。

2、容器集群网络是基于宿主机信息基础设施网络实现的网络抽象，无法在物理网络层进行常规处理。

因此，传统网络防火墙技术并不适用于容器网络环境。目前，在开源容器管理平台（例如，Kubernetes，K8S）通过网络策略（network policy）来进行容器隔离，这种隔离方式依赖网络插件的支持，对于无法使用合适的网络插件的容器集群网络无法使用，不具有通用性。而且，在使用中存在诸多的限制和不便，对运维人员的能力和业务水平有很高的要求。

发明内容

本公开实施例提供了一种容器隔离方法、装置、系统、电子设备及存储介质，以解决通过网络插件实现容器隔离不具有通用性的问题。

第一方面，本公开实施例提供了一种容器隔离方法，应用于发送端，该方法包括：获取握手报文和握手报文所属的源容器的属性标签组；将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中；根据响应报文，确定目标容器的属性标签组和目标端口；将

源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果；根据匹配结果，执行容器隔离动作。

第二方面，本公开实施例提供了一种容器隔离方法，应用于接收端，该方法包括：接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中；将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

第三方面，本公开实施例提供了一种容器隔离装置，应用于发送端，该装置包括：获取模块，用于获取握手报文和握手报文所属的源容器的属性标签组；发送模块，用于将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中；确定模块，用于根据响应报文，确定目标容器的属性标签组和目标端口；匹配模块，用于将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果；执行模块，用于根据匹配结果，执行容器隔离动作。

第四方面，本公开实施例提供了一种容器隔离装置，应用于接收端，该装置包括：获取模块，用于接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中；发送模块，用于将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

第五方面，本公开实施例提供了一种容器隔离系统，包括发送端和接收端；发送端执行上述任一项的方法；所述接收端执行上述任一项的方法。

第六方面，本公开实施例提供了一种电子设备，包括存储器、处理器及存储在存储器上的计算机程序，处理器在执行计算机程序时实现上述任一项的方法。

第七方面，本公开实施例提供了一种计算机可读存储介质，计算机可读存储介质内存储有计算机程序，计算机程序被处理器执行时实现上述任一项的方法。

第八方面，本公开实施例提供了一种计算机程序产品，计算机程序产品包括计算机程序，计算机程序被处理器执行时实现上述任一项的方法。

与现有技术相比，本公开具有如下优点：

本公开提供了一种容器隔离方法、装置、系统、电子设备及存储介质，首先，发送端获取握手报文和握手报文所属的源容器的属性标签组；将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中；然后，发送端根据响应报文，确定目标容器的属性标签组和目标端口；将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器

网络规则集进行匹配，得到匹配结果；根据匹配结果，执行容器隔离动作。本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

上述说明仅是本公开技术方案的概述，为了能够更清楚了解本公开的技术手段，可依照说明书的内容予以实施，并且为了让本公开的上述和其他目的、特征和优点能够更明显易懂，以下特举本公开的具体实施方式。

附图说明

在附图中，除非另外规定，否则贯穿多个附图相同的附图标记表示相同或相似的部件或元素。这些附图不一定是按照比例绘制的。应该理解，这些附图仅描绘了根据本公开的一些实施方式，而不应将其视为是对本公开范围的限制。

图 1 为本公开提供的容器隔离方法的一个应用场景的示意图；

图 2 为本公开一实施例的容器隔离方法的流程图；

图 3 为本公开一实施例的规则标签树的示意图；

图 4 为本公开一实施例的容器隔离方法的流程图；

图 5 为本公开一实施例的容器隔离装置的结构框图；

图 6 为本公开一实施例的容器隔离装置的结构框图；

图 7 为用来实现本公开实施例的电子设备的框图。

具体实施方式

在下文中，仅简单地描述了某些示例性实施例。正如本领域技术人员可认识到的那样，在不脱离本公开的构思或范围的情况下，可通过各种不同方式修改所描述的实施例。因此，附图和描述被认为本质上是示例性的，而非限制性的。

为便于理解本公开实施例的技术方案，以下对本公开实施例的相关技术进行说明。以下相关技术作为可选方案与本公开实施例的技术方案可以进行任意结合，其均属于本公开实施例的保护范围。

需要说明的是，本公开所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于用于分析的数据、存储的数据、展示的数据等），均为经用户授权或者经过各方充分授权的信息和数据，并且相关数据的收集、使用和处理需要遵守相关地区的相关法律法规和标准，并提供有相应的操作入口，供用户选择授权或者拒绝。

图 1 为本公开提供的容器隔离方法的一个应用场景的流程图。如图 1 所示，宿主机 A 中的源容器和宿主机 B 中的目标容器是容器集群网络的组成部分，通过发送握手

报文来确定隔离动作。需要说明的是，源容器和目标容器也可以在同一宿主机中，隔离实现过程与在不同的宿主机中的实现过程相同。另外，源容器和目标容器也可以各自分别是一个容器组，从而实现容器组之间的隔离。本实施例中，容器隔离方法包括如下步骤：

1、发送端应用层组件的采集模块，实时采集源容器的容器信息，容器信息包括以下至少一项：容器标识（Identity，ID）、容器网络命名空间（Network namespace，NS）、容器名称、容器镜像、容器所属容器管理平台（例如，k8s（Kubernetes））命名空间、容器应用信息。

具体的，在容器的宿主机上部署代理探针程序，通过接口或监测机制（例如，docker unix api（Application Programming Interface，应用程序编程接口）、cri（Container Runtime Interface，容器运行时接口）、k8s watch 机制）等手段实时获取其所在宿主机上容器的启动、停止事件、每个容器的容器信息。代理探针程序在容器启动时将每个容器的容器信息转化为 NS+属性标签组的格式，唯一的描述 NS 内的一个容器或一个容器组的属性，并将该信息发送到位于内核的网络驱动组件。

2、发送端通过应用层组件接收用户定义的容器网络规则集（如图 1 中所示的“规则”），将容器网络规则集下发到网络驱动组件。其中，容器网络规则集包括：规则标识、多个源容器的属性标签组，多个目标容器的属性标签组和目标端口，以及多个容器隔离动作等。

在容器集群中的容器 IP 和实例中是易变的，但其身份属性，如镜像、命名空间等是固定的，每个属性标签可以用 key:value 形式定义，一组多个标签（即属性标签组）即可用来固定的描述指定的一类容器。如：

image:centos:latest,namespace:defalut,app:clientapp 表示容器应用标签为 app:clientapp，网络命名空间标签为 default，镜像标签为 image:centos:latest 的容器。在同一个属性标签组中的标签是唯一的。

利用源容器的属性标签组、目标容器的属性标签组、目标端口定义两类容器之间容器网络规则，可以描述两个容器之间网络拓扑关系。举例如下：

（image:centos:latest,namespace:defalut,app:clientapp）-
（image:nginx:latest,namespace:default
,app:pop）:8080 表示在 default 命名空间中，具有 app:clientapp 标签，image:centos:latest 镜像的容器可以和在 default 命名空间中，具有 app:pop 标签，image:nginx:latest 镜像的容器的 8080 端口进行网络连接。多组上述容器网络规则，构成容器网络规则集，可描述整个容器集群的网络拓扑关系，而不在容器网络规则集中的网络连接行为即可根据生产实际需求进行审计、告警、拦截，从而实现容器隔离。

3、发送端通过网络驱动组件接收容器网络规则集，构建规则标签树。网络驱动组件接收容器信息，构建源容器的属性标签组和网络命名空间的映射关系。

接收端按照同样方式采集目标容器的容器信息执行上述预配置过程，源容器和目标容器之间通过网口发送传输控制协议（Transmission Control Protocol，TCP）握手报文（如图 1 中所示的“网络包处理”）。

4、发送端网络驱动组件使用数据包过滤模块（例如，netfilter）拦截出方向 TCP 握手报文（如图 1 中所示的“tcp syn”）。具体的，通过 netfilter 的接口注册位于三层网络输入\输出方向两个钩子（hook）点的回调函数，可以拦截内核网络栈上出入两个方向的 IP 层报文，识别其中的 TCP 握手报文，改造符合预设条件的 TCP 握手报文，根据当前源容器的网络命名空间，在 TCP 握手报文中增加预设标识（预设表示可以是任意形式，例如，fast open option 标识），表明这是一个经过改造过的 TCP 握手报文。

5、接收端网络驱动组件使用 netfilter 拦截出方向握手报文的响应报文（如图 1 中所示的“tcp syn ack”），根据目标容器的网络命名空间，获取目标容器的属性标签组及当前 TCP 连接的目标端口，生成有效载荷数据 payload，插入到响应报文的头部之后，重新计算报文的校验和，发送改造后的响应报文。

6、发送端网络驱动层使用 netfilter 拦截入方向的 TCP 握手报文的响应报文，解析响应报文中的 payload，获取目标容器的属性标签组和目标端口。同时根据当前网络接口的网络命名空间，在本地缓存中获取源容器的属性标签组。

7、发送端将源容器的属性标签组和目标容器的属性标签组在规则标签树中进行匹配，将目标端口与容器网络规则集中的目标端口集合进行匹配，得到最终的匹配结果。根据匹配结果，确定相应的容器隔离动作并执行。其中，容器隔离动作包括以下任一项：拦截、放行、审计、丢弃。可以为每个容器网络规则配置相应的容器隔离动作，例如，若规则 1（图中未示出）匹配成功，且相应的容器隔离动作为放行，则向接收端发送确认报文（如图 1 中所示的“tcp ack”），完成三次握手。若规则 1 匹配不成功，则丢弃响应报文，源容器和目标容器无法建立连接。

8、将执行结果发送到应用层组件，进行结果展示。

本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

本公开实施例提供了一种容器隔离方法，本实施例中的方法可以应用于网络请求的发送端。如图 2 所示为本公开一实施例的容器隔离方法的流程图，包括：

步骤 S201，获取握手报文和握手报文所属的源容器的属性标签组。

其中，属性标签组中包括至少一个属性标签，属性标签是根据容器的容器信息确定的。容器信息包括以下至少一项：容器标识（Identity，ID）、容器网络命名空间（Network namespace，NS）、容器名称、容器镜像、容器所属容器管理平台（例如，

k8s）命名空间、容器应用信息。

步骤 S202，将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中。

其中，预设标识可以包括任意形式的标识，用于指示该网络报文需要获取目标容器的属性标签组和目标端口。

步骤 S203，根据响应报文，确定目标容器的属性标签组和目标端口。

具体的，对响应报文进行解析，得到目标容器的属性标签组和目标端口。

步骤 S204，将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果。

其中，容器网络规则集包括多个容器网络规则，将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则进行匹配，得到匹配结果。

步骤 S205，根据匹配结果，执行容器隔离动作。

如果源容器的属性标签组、目标容器的属性标签组和目标端口，与一个容器网络规则匹配成功，则执行该容器网络规则对应的容器隔离动作。

示例性的，容器隔离动作包括以下任一项：拦截、放行、审计、丢弃。

本公开实施例提供的容器隔离方法，首先，发送端获取握手报文和握手报文所属的源容器的属性标签组；将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中；然后，发送端根据响应报文，确定目标容器的属性标签组和目标端口；将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果；根据匹配结果，执行容器隔离动作。本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

下面通过多种实现方式对上述各步骤的具体实现过程进行介绍。

在一种实现方式中，方法还包括：获取多个源容器的属性标签组，多个目标容器的属性标签组和目标端口；基于多个源容器的属性标签组，多个目标容器的属性标签组和目标端口，以及多个容器隔离动作，配置容器网络规则集。

其中，容器网络规则集中的每个规则由如下信息构成：规则 ID、源标签列表（即源容器的属性标签组）、目标标签列表（目标容器的属性标签组）、目标端口。

在一示例中，容器网络规则集如表 1 所示：

规则 ID	源标签列表	目标标签列表	目标端口集口
1	标签 1, 标签 2, 标签 3...	标签 1, 标签 4, 标签 5...	[22, 80, 8080...]

2	标签 1, 标签 5, 标签 6. . .	标签 2, 标签 3, 标签 4. . .	[22, 80, 8080. . .]
3	. . . .	. . . .	. . . .

表 1

在规则匹配过程中，如果一个网络连接的发起源容器的标签列表匹配规则中的源标签列表，目标容器的标签列表匹配规则中的目标标签列表，且端口属于规则中的目标端口集合，则命中该规则，即可执行该规则定义的容器隔离动作。具体的，通过构建规则标签树和简化规则集，在容器网络规则中的属性标签和容器的属性标签匹配时，通过反向索引，快速获取每个属性标签对应的容器网络规则，根据命中次数，快速匹配出规则结果。

匹配过程具体见如下实施例：

在一种实现方式中，匹配结果包括标签匹配结果和端口匹配结果；将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，包括：根据容器网络规则集，生成规则标签树；将源容器的属性标签组、目标容器的属性标签组，与规则标签树进行匹配，得到标签匹配结果；将目标端口与容器网络规则集中的目标端口集合进行匹配，得到端口匹配结果；其中，规则标签树中的节点为属性标签描述信息，属性标签描述信息包括：属性标签文本、当前属性标签出现在源容器的属性标签组中的规则标识列表、当前属性标签出现在目标容器的属性标签组中的规则标识列表，规则标识列表中包括至少一个容器网络规则的标识。

在实际应用中，应用层接收处理用户定义的容器网络规则集，在容器网络规则集发生变化时，将最新容器网络规则集更新到位于内核的网络驱动组件。网络驱动组件接收应用层下发的容器网络规则集，为在后续规则匹配过程中优化性能，将其解析为规则标签树。规则标签树是一种数据结构，可以是通过红黑算法计算得到的红黑树，规则标签树上每个节点为一个属性标签描述信息，包括：属性标签文本、当前属性标签出现在源容器的属性标签组中的规则标识列表、当前属性标签出现在目标容器的属性标签组中的规则标识列表。

在一示例中，根据表 1 的容器网络规则集得到的规则标签树如图 3 所示。标签 1、标签 2...标签 6 表示标签文本，源容器对应的规则标识表示标签出现在源容器的属性标签组中的规则标识列表，目标容器对应的规则标识表示标签出现在目标容器的属性标签组中的规则标识列表。如表 1 所示，标签 1 出现在规则 1 和规则 2 的源容器的属性标签组中，出现在规则 1 的目标容器的属性标签组中。标签 2 出现在规则 1 的源容器的属性标签组中，出现在规则 2 的目标容器的属性标签组中。标签 3 出现在规则 1 的源容器的属性标签组中，出现在规则 2 的目标容器的属性标签组中。标签 4 没有出现在表 1 中任一规则的源容器的属性标签组中，出现在规则 1 和规则 2 的目标容器的属性标签组中。标签 5 出现在规则 2 的源容器的属性标签组中，出现在规则 1 的目标

容器的属性标签组中。标签 6 出现在规则 2 的源容器的属性标签组中，没有出现在表 1 中任一规则的目标容器的属性标签组中。

在一种实现方式中，将源容器的属性标签组、目标容器的属性标签组，与规则标签树进行匹配，得到标签匹配结果，包括：在规则标签树中，查询当前源容器的属性标签组匹配的多个源容器的属性标签组对应的第一规则标识列表，查询当前目标容器的属性标签组匹配的多个目标容器的属性标签组对应的第二规则标识列表；根据第一规则标识列表和第二规则标识列表，得到标签匹配结果。

在实际应用中，在规则标签树中，查询当前源容器的属性标签组匹配的多个源容器的属性标签组对应的容器网络规则，得到容器网络规则的标识列表，即第一规则标识列表，查询当前目标容器的属性标签组匹配的多个目标容器的属性标签组对应的容器网络规则，得到容器网络规则的标识列表，即第二规则标识列表；根据第一规则标识列表和第二规则标识列表，计算标签匹配结果。

在一种实现方式中，根据第一规则标识列表和第二规则标识列表，得到标签匹配结果，包括：根据第一规则标识列表和第二规则标识列表的交集，得到匹配过程中多个规则标识被命中的数量列表；根据规则标签树，生成简化规则集；简化规则集包括容器网络规则集中的每个规则中的源容器的属性标签数量和目标容器的属性标签数量；将多个规则标识被命中的数量列表和简化规则集进行比对，得到标签匹配结果。

对于任一个标签文本，可在规则标签树快速检索到它在容器网络规则集中源容器的属性标签组中出现的规则标识集合、目标容器的属性标签组中出现的规则标识集合。同时生成简化规则集，简化规则集中的每个规则均记录它拥有的源容器的属性标签组中标签数量，目标容器的属性标签组中的标签数量。

在一示例中，根据如图 3 所示的规则标签树得到的简化规则集如表 2 所示，源标签列表标签数量表示源容器的属性标签组中标签数量，目标标签列表标签数量表示目标容器的属性标签组中的标签数量。

规则 ID	源标签列表标签数量	目标标签列表标签数量	目标端口集口
1	3	3	[22,80,8080...]
2	3	3	[22,80,8080...]
3	....	....	....

表 2

在规则匹配时，首先遍历源容器的属性标签组中的每个标签，用该标签在规则标签树中快速查找到其标签描述信息，获取标签描述信息中源容器的属性标签组中的规则标识列表。例如，从一个属性标签组为（label1,label2,label3）的容器向另一个属性标签组为（label1,label4,label5）容器发起网络连接，在匹配时得到待匹配源容器的属

性标签和标签树中源容器对应的规则标识列表，如表 3 所示：

待匹配源容器的属性标签	标签树中源容器对应的规则标识
label1	1,2
label2	1
lable3	1

表 3

同样，对目标容器的属性标签组进行相同操作，得到待匹配目标容器的属性标签和标签树中目标容器对应的规则标识列表，如表 4 所示：

待匹配目标容器的属性标签	标签树中目标容器对应的规则标识
label1	1
label4	1，2
lable5	1

表 4

根据获取到两个匹配过程表表 3 和表 4 的标签树中源容器对应的规则标识和目标容器对应的规则标识进行交集计数，得到每个规则 ID 在匹配中被命中的数量列表，如表 5 所示：

规则 ID	本次匹配中，命中本规则源容器的属性标签的次数	本次匹配中，命中本规则目标容器的属性标签的次数
1	3	3
2	1	1

表 5

将规则匹配命中数量表（表 5）和简化规则集（表 2）进行对比匹配，表 5 中的源容器的属性标签和目标容器的属性标签命中次数均分别等于简化规则集中，该容器网络规则中源容器的属性标签组中标签的数量、目标容器的属性标签组中标签的数量，并且目标端口属于目标端口集合，即认为命中该容器网络规则，即可执行该容器网络规则定义的容器隔离动作。

在本示例中规则 1 的源容器的属性标签命中 3 次，目标容器的属性标签命中 3 次，匹配规则 1 成功，即可执行规则 1 定义的容器隔离动作。

可选的，在容器防火墙中采用白名单模式，即与容器网络规则匹配的网络连接予以放行，其它的网络连接不在容器网络规则集中，即执行规则集预定义的默认动作，

进行拦截、审计或告警。

在一种实现方式中，获取源容器的属性标签组，包括：获取握手报文所属的源容器的网络命名空间标识，根据网络命名空间标识获取源容器的属性标签组。

在实际应用中，通过宿主机操作系统给每个容器或容器组分配的内核网络命名空间来实现同一个宿主机上不同的独立网络空间隔离，可以用每个容器的网络命名空间标识，在内核网络驱动层作为描述一个容器或容器组的唯一标识，预先建立网络命名空间标识和容器的属性标签组的映射关系，根据网络命名空间标识，可以在内存中获取容器的属性标签组。

本公开实施例提供了一种容器隔离方法，本实施例中的方法可以应用于网络请求的接收端。如图 4 所示为本公开一实施例的容器隔离方法的流程图，包括：

步骤 S401，接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中。

其中，属性标签组中包括至少一个属性标签，属性标签是根据容器的容器信息确定的。容器信息包括以下至少一项：容器标识（Identity，ID）、容器网络命名空间（Network namespace，NS）、容器名称、容器镜像、容器所属容器管理平台（例如，k8s）命名空间、容器应用信息。

其中，预设标识可以包括任意形式的标识，用于指示该网络报文需要获取目标容器的属性标签组和目标端口。

步骤 S402，将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

如果源容器的属性标签组、目标容器的属性标签组和目标端口，与一个容器网络规则匹配成功，则执行该容器网络规则对应的容器隔离动作。

示例性的，容器隔离动作包括以下任一项：拦截、放行、审计、丢弃。

本公开实施例提供的容器隔离方法，首先，接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中；然后，将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

在一种实现方式中，获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中，包括：获取目标容器的属性标签组和目标端口，生成有效载荷数据；将有

效载荷数据添加到响应报文的包头中，并计算添加有效载荷数据后的响应报文的校验和。

具体的，接收端网络驱动组件使用 netfilter 拦截出方向握手报文的响应报文，根据目标容器的网络命名空间，获取目标容器的属性标签组及当前 TCP 连接的目标端口，生成有效载荷数据 payload，插入到响应报文的头部之后，重新计算响应报文的校验和，发送改造后的响应报文。

与本公开实施例提供的方法的应用场景以及方法相对应地，本公开实施例还提供一种容器隔离装置。如图 5 所示为本公开一实施例的容器隔离装置的结构框图，该装置包括：

获取模块 501，用于获取握手报文和握手报文所属的源容器的属性标签组。

发送模块 502，用于将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中。

确定模块 503，用于根据响应报文，确定目标容器的属性标签组和目标端口。

匹配模块 504，用于将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果。

执行模块 505，用于根据匹配结果，执行容器隔离动作。

本公开实施例提供的容器隔离装置，首先，发送端获取握手报文和握手报文所属的源容器的属性标签组；将添加预设标识后的握手报文发送至接收端，预设标识用于接收端获取目标容器的属性标签组和目标端口并添加到握手报文的响应报文中；然后，发送端根据响应报文，确定目标容器的属性标签组和目标端口；将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果；根据匹配结果，执行容器隔离动作。本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

在一种实现方式中，容器隔离装置还用于：获取多个源容器的属性标签组，多个目标容器的属性标签组和目标端口；基于多个源容器的属性标签组，多个目标容器的属性标签组和目标端口，以及多个容器隔离动作，配置容器网络规则集。

在一种实现方式中，匹配结果包括标签匹配结果和端口匹配结果；匹配模块 504，用于：根据容器网络规则集，生成规则标签树；将源容器的属性标签组、目标容器的属性标签组，与规则标签树进行匹配，得到标签匹配结果；将目标端口与容器网络规则集中的目标端口集合进行匹配，得到端口匹配结果；其中，规则标签树中的节点为属性标签描述信息，属性标签描述信息包括：属性标签文本、当前属性标签出现在源容器的属性标签组中的规则标识列表、当前属性标签出现在目标容器的属性标签组中

的规则标识列表，规则标识列表中包括至少一个容器网络规则的标识。

在一种实现方式中，匹配模块 504 在将源容器的属性标签组、目标容器的属性标签组，与规则标签树进行匹配，得到标签匹配结果时，用于：在规则标签树中，查询当前源容器的属性标签组匹配的多个源容器的属性标签组对应的第一规则标识列表，查询当前目标容器的属性标签组匹配的多个目标容器的属性标签组对应的第二规则标识列表；根据第一规则标识列表和第二规则标识列表，得到标签匹配结果。

在一种实现方式中，匹配模块 504 在根据第一规则标识列表和第二规则标识列表，得到标签匹配结果时，用于：根据第一规则标识列表和第二规则标识列表的交集，得到匹配过程中多个规则标识被命中的数量列表；根据规则标签树，生成简化规则集；简化规则集包括容器网络规则集中的每个规则中的源容器的属性标签数量和目标容器的属性标签数量；将多个规则标识被命中的数量列表和简化规则集进行比对，得到标签匹配结果。

在一种实现方式中，获取模块 501 在获取源容器的属性标签组时，用于：获取握手报文所属的源容器的网络命名空间标识，根据网络命名空间标识获取源容器的属性标签组。

在一种实现方式中，容器隔离动作包括以下任一项：拦截、放行、审计、丢弃。

本公开实施例中的各模块的功能可以参见上述方法中的对应描述，并具备相应的有益效果，在此不再赘述。

与本公开实施例提供的方法的应用场景以及方法相对应地，本公开实施例还提供一种容器隔离装置。如图 6 所示为本公开一实施例的容器隔离装置的结构框图，该装置包括：

获取模块 601，用于接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中。

发送模块 602，用于将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

本公开实施例提供的容器隔离装置，首先，接收发送端发送的握手报文，若握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中；然后，将响应报文返回给发送端，响应报文用于发送端确定目标容器的属性标签组和目标端口，并将源容器的属性标签组、目标容器的属性标签组和目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。本公开实施例中，通过改造握手报文，添加预设标识，在握手报文的响应报文中携带网络对端的目标容器的属性标签组和目标端口，从而实现网络连接双方的容器身份确认。在对握手性能无影响的情况下，通过容器网络规则集进行规则匹配，执行容器隔离动

作。不依赖容器网络的网络插件，可以在网络插件不支持的环境下实现容器隔离。

在一种实现方式中，获取模块 601 在获取目标容器的属性标签组和目标端口添加到握手报文的响应报文中时，用于：获取目标容器的属性标签组和目标端口，生成有效载荷数据；将有效载荷数据添加到响应报文的包头中，并计算添加有效载荷数据后的响应报文的校验和。

本公开实施例中的各模块的功能可以参见上述方法中的对应描述，并具备相应的有益效果，在此不再赘述。

图 7 为用来实现本公开实施例的电子设备的框图。如图 7 所示，该电子设备包括：存储器 710 和处理器 720，存储器 710 内存储有可在处理器 720 上运行的计算机程序。处理器 720 执行该计算机程序时实现上述实施例中的方法。存储器 710 和处理器 720 的数量可以为一个或多个。

该电子设备还包括：

通信接口 730，用于与外界设备进行通信，进行数据交互传输。

如果存储器 710、处理器 720 和通信接口 730 独立实现，则存储器 710、处理器 720 和通信接口 730 可以通过总线相互连接并完成相互间的通信。该总线可以是工业标准体系结构（Industry Standard Architecture，ISA）总线、外部设备互连（Peripheral Component Interconnect，PCI）总线或扩展工业标准体系结构（Extended Industry Standard Architecture，EISA）总线等。该总线可以分为地址总线、数据总线、控制总线等。为便于表示，图 7 中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

可选的，在具体实现上，如果存储器 710、处理器 720 及通信接口 730 集成在一块芯片上，则存储器 710、处理器 720 及通信接口 730 可以通过内部接口完成相互间的通信。

本公开实施例提供了一种容器隔离系统，包括发送端和接收端；发送端执行本公开实施例提供的方法；接收端执行本公开实施例提供的方法。

本公开实施例提供了一种计算机可读存储介质，其存储有计算机程序，该程序被处理器执行时实现本公开实施例中提供的方法。

本公开实施例提供了一种计算机程序产品，计算机程序产品包括计算机程序，计算机程序被处理器执行时实现本公开实施例中提供的方法。

本公开实施例还提供了一种芯片，该芯片包括处理器，用于从存储器中调用并运行存储器中存储的指令，使得安装有芯片的通信设备执行本公开实施例提供的方法。

本公开实施例还提供了一种芯片，包括：输入接口、输出接口、处理器和存储器，输入接口、输出接口、处理器以及存储器之间通过内部连接通路相连，处理器用于执行存储器中的代码，当代码被执行时，处理器用于执行申请实施例提供的方法。

应理解的是，上述处理器可以是中央处理器（Central Processing Unit，CPU），还

可以是其他通用处理器、数字信号处理器（Digital Signal Processor，DSP）、专用集成电路（Application Specific Integrated Circuit，ASIC）、现场可编程门阵列（Field Programmable Gate Array，FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者是任何常规的处理器等。值得说明的是，处理器可以是支持进阶精简指令集机器（Advanced RISC Machines，ARM）架构的处理器。

进一步地，可选的，上述存储器可以包括只读存储器和随机访问存储器。该存储器可以是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其中，非易失性存储器可以包括只读存储器（Read-Only Memory，ROM）、可编程只读存储器（Programmable ROM，PROM）、可擦除可编程只读存储器（Erasable PROM，EPROM）、电可擦除可编程只读存储器（Electrically EPROM，EEPROM）或闪存。易失性存储器可以包括随机访问存储器（Random Access Memory，RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的 RAM 均可用。例如，静态随机访问存储器（Static RAM，SRAM）、动态随机访问存储器（Dynamic Random Access Memory，DRAM）、同步动态随机访问存储器（Synchronous DRAM，SDRAM）、双倍数据速率同步动态随机访问存储器（Double Data Rate SDRAM，DDR SDRAM）、增强型同步动态随机访问存储器（Enhanced SDRAM，ESDRAM）、同步链接动态随机访问存储器（Sync link DRAM，SLDRAM）和直接内存总线随机访问存储器（Direct Rambus RAM，DR RAM）。

在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行计算机程序指令时，全部或部分地产生依照本公开的流程或功能。计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。计算机指令可以存储在计算机可读存储介质中，或者从一个计算机可读存储介质向另一个计算机可读存储介质传输。

在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包括于本公开的至少一个实施例或示例中。而且，描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外，在不相互矛盾的情况下，本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

此外，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或隐含地包括至少一个该特征。在本公开的描述中，“多个”的含义是两个或两个以上，除非另有明确具体的限定。

流程图中描述的或在此以其他方式描述的任何过程或方法可以被理解为，表示包括一个或更多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分。并且本公开的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能。

在流程图中描述的或在此以其他方式描述的逻辑和/或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供指令执行系统、装置或设备（如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统）使用，或结合这些指令执行系统、装置或设备而使用。

应理解的是，本公开的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中，多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。上述实施例方法的全部或部分步骤是可以通过程序来指令相关的硬件完成，该程序可以存储于一种计算机可读存储介质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

此外，在本公开各个实施例中的各功能单元可以集成在一个处理模块中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个模块中。上述集成的模块既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。上述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时，也可以存储在一个计算机可读存储介质中。该存储介质可以是只读存储器，磁盘或光盘等。

以上所述，仅为本公开的示例性实施方式，但本公开的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本公开记载的技术范围内，可轻易想到其各种变化或替换，这些都应涵盖在本公开的保护范围之内。因此，本公开的保护范围应以权利要求的保护范围为准。

权 利 要 求 书

1. 一种容器隔离方法，其中，所述方法应用于发送端，所述方法包括：

获取握手报文和所述握手报文所属的源容器的属性标签组；

将添加预设标识后的握手报文发送至接收端，所述预设标识用于所述接收端获取目标容器的属性标签组和目标端口并添加到所述握手报文的响应报文中；

根据所述响应报文，确定所述目标容器的属性标签组和所述目标端口；

将所述源容器的属性标签组、所述目标容器的属性标签组和所述目标端口，与容器网络规则集进行匹配，得到匹配结果；

根据所述匹配结果，执行容器隔离动作。

2. 根据权利要求 1 所述的方法，其中，所述方法还包括：

获取多个源容器的属性标签组，多个目标容器的属性标签组和目标端口；

基于所述多个源容器的属性标签组，所述多个目标容器的属性标签组和目标端口，配置容器网络规则集。

3. 根据权利要求 1 或 2 所述的方法，其中，所述匹配结果包括标签匹配结果和端口匹配结果；所述将所述源容器的属性标签组、所述目标容器的属性标签组和所述目标端口，与容器网络规则集进行匹配，得到匹配结果，包括：

根据所述容器网络规则集，生成规则标签树；

将所述源容器的属性标签组、所述目标容器的属性标签组，与所述规则标签树进行匹配，得到所述标签匹配结果；

将所述目标端口与所述容器网络规则集中的目标端口集合进行匹配，得到所述端口匹配结果；

其中，所述规则标签树中的节点为属性标签描述信息，所述属性标签描述信息包括：属性标签文本、当前属性标签出现在所述源容器的属性标签组中的规则标识列表、所述当前属性标签出现在所述目标容器的属性标签组中的规则标识列表，所述规则标识列表中包括至少一个容器网络规则的标识。

4.根据权利要求 3 所述的方法，其中，所述将所述源容器的属性标签组、所述目标容器的属性标签组，与所述规则标签树进行匹配，得到标签匹配结果，包括：

在所述规则标签树中，查询所述源容器的属性标签组匹配的多个源容器的属性标签组对应的第一规则标识列表，查询所述目标容器的属性标签组匹配的多个目标容器的属性标签组对应的第二规则标识列表；

根据所述第一规则标识列表和所述第二规则标识列表，得到所述标签匹配结果。

5.根据权利要求 4 所述的方法，其中，所述根据所述第一规则标识列表和所述第二规则标识列表，得到所述标签匹配结果，包括：

根据所述第一规则标识列表和所述第二规则标识列表的交集，得到匹配过程中多个规则标识被命中的数量列表；

根据所述规则标签树，生成简化规则集；所述简化规则集包括所述容器网络规则集中的每个规则中的源容器的属性标签数量和目标容器的属性标签数量；

将所述多个规则标识被命中的数量列表和所述简化规则集进行比对，得到所述标签匹配结果。

6.根据权利要求 1-5 任一项所述的方法，其中，获取源容器的属性标签组，包括：

获取所述握手报文所属的源容器的网络命名空间标识，根据所述网络命名空间标识获取所述源容器的属性标签组。

7. 根据权利要求 1-5 任一项所述的方法，其中，所述容器隔离动作包括以下任一项：拦截、放行、审计、丢弃。

8. 一种容器隔离方法，其中，所述方法应用于接收端，所述方法包括：

接收发送端发送的握手报文，若所述握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到所述握手报文的响应报文中；

将所述响应报文返回给所述发送端，所述响应报文用于所述发送端确定所述目标容器的属性标签组和所述目标端口，并将源容器的属性标签组、所述目标容器的属性标签组和所述目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

9. 根据权利要求 8 所述的方法，其中，所述获取目标容器的属性标签组和目标端口添加到所述握手报文的响应报文中，包括：

获取目标容器的属性标签组和目标端口，生成有效载荷数据；

将所述有效载荷数据添加到所述响应报文的包头中，并计算添加所述有效载荷数据后的响应报文的校验和。

10. 一种容器隔离装置，其中，所述装置应用于发送端，所述装置包括：

获取模块，用于获取握手报文和所述握手报文所属的源容器的属性标签组；

发送模块，用于将添加预设标识后的握手报文发送至接收端，所述预设标识用于所述接收端获取目标容器的属性标签组和目标端口并添加到所述握手报文的响应报文中；

确定模块，用于根据所述响应报文，确定所述目标容器的属性标签组和所述目标端口；

匹配模块，用于将所述源容器的属性标签组、所述目标容器的属性标签组和所述目标端口，与容器网络规则集进行匹配，得到匹配结果；

执行模块，用于根据所述匹配结果，执行容器隔离动作。

11. 一种容器隔离装置，其中，所述装置应用于接收端，所述装置包括：

获取模块，用于接收发送端发送的握手报文，若所述握手报文中包含预设标识，则获取目标容器的属性标签组和目标端口添加到所述握手报文的响应报文中；

发送模块，用于将所述响应报文返回给所述发送端，所述响应报文用于所述发送

端确定所述目标容器的属性标签组和所述目标端口，并将源容器的属性标签组、所述目标容器的属性标签组和所述目标端口，与容器网络规则集进行匹配，得到匹配结果，根据匹配结果，执行容器隔离动作。

12. 一种容器隔离系统，其中，所述容器隔离系统包括发送端和接收端；所述发送端执行权利要求 1-7 任一项所述的方法；所述接收端执行权利要求 8-9 任一项所述的方法。

13. 一种电子设备，其中，所述电子设备包括存储器、处理器及存储在存储器上的计算机程序，所述处理器在执行所述计算机程序时实现权利要求 1-9 中任一项所述的方法。

14. 一种计算机可读存储介质，其中，所述计算机可读存储介质内存储有计算机程序，所述计算机程序被处理器执行时实现权利要求 1-9 中任一项所述的方法。

15. 一种计算机程序产品，其中，所述计算机程序产品包括计算机程序，所述计算机程序被处理器执行时实现权利要求 1-9 中任一项所述的方法。