CONTAINER ISOLATION METHOD, APPARATUS AND SYSTEM, ELECTRONIC DEVICE, AND STORAGE MEDIUM

The present disclosure claims priority to Chinese Patent Application No. 202410346511.2, filed with the China National Intellectual Property Administration on March 25, 2024, entitled "CONTAINER ISOLATION METHOD, APPARATUS, SYSTEM, ELECTRONIC DEVICE, AND STORAGE MEDIUM," the entire content of which is incorporated herein by reference.

TECHNICAL FIELD

The present disclosure relates to the field of network security technology, and in particular to a container isolation method, apparatus, system, electronic device, and storage medium.

BACKGROUND

In a container cluster network, the networks of all containers in an entire cluster are interconnected. Once an intrusion occurs, an intruder exploits container network connectivity to move laterally across the entire container cluster, seeking further exploitable intrusion points, resulting in a wider range of intrusion. Therefore, it is necessary to isolate containers on demand to prevent malicious traffic from flowing arbitrarily in the container cluster.

The container cluster network has the following characteristics:

1. Internet Protocol (Internet Protocol, IP) addresses and the number of containers of the same application are not fixed, and the containers migrate among different hosts in the cluster.

2. The container cluster network is a network abstraction implemented based on a host information infrastructure network, and conventional processing cannot be performed at a physical network layer.

Therefore, traditional network firewall technology is not applicable to a container network environment. Currently, in an open-source container management platform (for example, Kubernetes, K8S), container isolation is performed through a network policy (network policy). This isolation manner relies on support of a network plugin, and is not usable for a container cluster network where a suitable network plugin cannot be used, thus lacking universality. Moreover, there are numerous limitations and inconveniences in use, and high requirements

are imposed on capabilities and professional proficiency of operation and maintenance personnel.

SUMMARY

Embodiments of the present disclosure provide a container isolation method, apparatus, system, electronic device, and storage medium, to solve the problem that implementing container isolation through a network plugin lacks universality.

In a first aspect, an embodiment of the present disclosure provides a container isolation method applied to a sending end, the method including: obtaining a handshake packet and an attribute label group of a source container to which the handshake packet belongs; sending the handshake packet with a preset identifier added thereto to a receiving end, where the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet; determining the attribute label group and the target port of the target container according to the response packet; matching the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result; and performing a container isolation action according to the matching result.

In a second aspect, an embodiment of the present disclosure provides a container isolation method applied to a receiving end, the method including: receiving a handshake packet sent by a sending end, and if the handshake packet includes a preset identifier, obtaining an attribute label group and a target port of a target container and adding them to a response packet of the handshake packet; and returning the response packet to the sending end, where the response packet is used for the sending end to determine the attribute label group and the target port of the target container, and match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and perform a container isolation action according to the matching result.

In a third aspect, an embodiment of the present disclosure provides a container isolation apparatus applied to a sending end, the apparatus including: an obtaining module, configured to obtain a handshake packet and an attribute label group of a source container to which the handshake packet belongs; a sending module, configured to send the handshake packet with a preset identifier added thereto to a receiving end, where the preset identifier is used for the

receiving end to obtain an attribute label group and a target port of a target container and add them to a response packet of the handshake packet; a determining module, configured to determine the attribute label group and the target port of the target container according to the response packet; a matching module, configured to match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result; and a performing module, configured to perform a container isolation action according to the matching result.

In a fourth aspect, an embodiment of the present disclosure provides a container isolation apparatus applied to a receiving end, the apparatus including: an obtaining module, configured to receive a handshake packet sent by a sending end, and if the handshake packet includes a preset identifier, obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet; and a sending module, configured to return the response packet to the sending end, where the response packet is used for the sending end to determine the attribute label group and the target port of the target container, match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and perform a container isolation action according to the matching result.

In a fifth aspect, an embodiment of the present disclosure provides a container isolation system, including a sending end and a receiving end, where the sending end performs the method according to any one of the above; and the receiving end performs the method according to any one of the above.

In a sixth aspect, an embodiment of the present disclosure provides an electronic device, including a memory, a processor, and a computer program stored on the memory, where the processor implements the method according to any one of the above when executing the computer program.

In a seventh aspect, an embodiment of the present disclosure provides a computer-readable storage medium, where a computer program is stored in the computer-readable storage medium, and when the computer program is executed by a processor, the method according to any one of the above is implemented.

In an eighth aspect, an embodiment of the present disclosure provides a computer program product, where the computer program product includes a computer program, and when the computer program is executed by a processor, the method according to any one of the above is implemented.

Compared with the prior art, the present disclosure has the following advantages:

The present disclosure provides a container isolation method, apparatus, system, electronic device, and storage medium. First, a sending end obtains a handshake packet and an attribute label group of a source container to which the handshake packet belongs; sends the handshake packet with a preset identifier added thereto to a receiving end, where the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet. Then, the sending end determines the attribute label group and the target port of the target container according to the response packet, matches the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and performs a container isolation action according to the matching result. In embodiments of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby achieving container identity confirmation of both parties of a network connection. Without affecting handshake performance, rule matching is performed through the container network rule set, and the container isolation action is performed. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

The above description is merely an overview of the technical solutions of the present disclosure. In order to provide a clearer understanding of the technical means of the present disclosure, implementation may be made in accordance with the content of the specification, and in order to make the above and other objects, features, and advantages of the present disclosure more apparent and comprehensible, specific implementations of the present disclosure are given below.

BRIEF DESCRIPTION OF DRAWINGS

In the drawings, unless otherwise specified, the same reference numerals throughout the multiple drawings denote the same or similar components or elements. The drawings are not necessarily drawn to scale. It should be understood that the drawings only depict some embodiments according to the present disclosure and should not be construed as limiting the scope of the present disclosure.

FIG. 1 is a schematic diagram of an application scenario of a container isolation method provided by the present disclosure;

FIG. 2 is a flowchart of a container isolation method according to an embodiment of the present disclosure;

FIG. 3 is a schematic diagram of a rule label tree according to an embodiment of the present disclosure;

FIG. 4 is a flowchart of a container isolation method according to an embodiment of the present disclosure;

FIG. 5 is a structural block diagram of a container isolation apparatus according to an embodiment of the present disclosure;

FIG. 6 is a structural block diagram of a container isolation apparatus according to an embodiment of the present disclosure;

FIG. 7 is a block diagram of an electronic device used to implement an embodiment of the present disclosure.

DETAILED DESCRIPTION

In the following, only certain exemplary embodiments are briefly described. As those skilled in the art would recognize, the described embodiments may be modified in various different ways without departing from the spirit or scope of the present disclosure. Accordingly, the drawings and description are to be regarded as illustrative in nature and not restrictive.

To facilitate understanding of the technical solutions of the embodiments of the present disclosure, the related technologies of the embodiments of the present disclosure are described below. The following related technologies, as optional solutions, may be combined in any manner with the technical solutions of the embodiments of the present disclosure, all of which fall within the protection scope of the embodiments of the present disclosure.

It should be noted that the user information (including but not limited to user device information, user personal information, etc.) and data (including but not limited to data used for analysis, stored data, displayed data, etc.) involved in the present disclosure are all information and data authorized by users or fully authorized by all relevant parties, and the collection, use, and processing of the relevant data need to comply with the relevant laws, regulations, and standards of the relevant regions, and corresponding operation entries are provided for users to select authorization or refusal.

FIG. 1 is a flowchart of an application scenario of a container isolation method provided by the present disclosure. As shown in FIG. 1, a source container in host A and a target container in host B are components of a container cluster network, and an isolation action is determined by sending a handshake packet. It should be noted that the source container and the target container may also be in the same host, and the isolation implementation process is the same as the implementation process in different hosts. In addition, the source container and the target container may each be a container group, thereby achieving isolation between container groups. In this embodiment, the container isolation method includes the following steps:

1. A collection module of an application layer component at a sending end collects container information of a source container in real time, where the container information includes at least one of the following: container identity (Identity, ID), container network namespace (Network namespace, NS), container name, container image, namespace of a container management platform to which the container belongs (e.g., k8s (Kubernetes)), and container application information.

Specifically, a proxy probe program is deployed on a host of a container, and startup and stop events of containers on the host and container information of each container are obtained in real time through an interface or a monitoring mechanism (e.g., docker unix api (Application Programming Interface, application programming interface), cri (Container Runtime Interface, container runtime interface), k8s watch mechanism) or other means. When a container starts, the proxy probe program converts the container information of each container into a format of NS+attribute label group, which uniquely describes an attribute of one container or one container group in the NS, and sends the information to a network driver component located in a kernel.

2. The sending end receives, through an application layer component, a user-defined container network rule set (e.g., "Rule" as shown in FIG. 1), and delivers the container network rule set to a network driver component. The container network rule set includes: a rule identifier, attribute label groups of a plurality of source containers, attribute label groups and target ports of a plurality of target containers, and a plurality of container isolation actions, etc.

Container IP addresses and instances in a container cluster are volatile, but identity attributes such as image and namespace are fixed. Each attribute label may be defined in a key:value format, and a group of a plurality of labels (i.e., an attribute label group) may be used to fixedly describe a specified type of container. For example: image:centos:latest,namespace:default,app:clientapp represents a container with a container

application label of app:clientapp, a network namespace label of default, and an image label of image:centos:latest. Labels in the same attribute label group are unique.

The attribute label group of the source container, the attribute label group of the target container, and the target port can be utilized to define a container network rule between two types of containers, which can describe a network topology relationship between two containers. An example is as follows:

(image:centos:latest,namespace:default,app:clientapp)-
(image:nginx:latest,namespace:default

,app:pop):8080 indicates that in the default namespace, a container with the app:clientapp label and the image:centos:latest image is allowed to establish a network connection with port 8080 of a container with the app:pop label and the image:nginx:latest image in the default namespace. A plurality of groups of the above container network rule constitute a container network rule set, which may describe the network topology relationship of the entire container cluster, and network connection behaviors not in the container network rule set may be audited, alerted, or blocked according to actual production requirements, thereby achieving container isolation.

3. The sending end receives the container network rule set through the network driver component, and constructs a rule label tree. The network driver component receives container information, and constructs a mapping relationship between the attribute label group and the network namespace of the source container.

A receiving end collects container information of a target container in the same manner to perform the above pre-configuration process, and a transmission control protocol (Transmission Control Protocol, TCP) handshake packet is sent between the source container and the target container through a network interface (e.g., "Network packet processing" as shown in FIG. 1).

4. A network driver component at the sending end uses a data packet filtering module (e.g., netfilter) to intercept an outbound TCP handshake packet (e.g., "tcp syn" as shown in FIG. 1). Specifically, by registering callback functions at two hook (hook) points in an inbound/outbound direction of a layer-three network through an interface of netfilter, IP layer packets in both inbound and outbound directions on a kernel network stack may be intercepted, the TCP handshake packet among them may be identified, the TCP handshake packet that meets a preset condition may be modified, and a preset identifier may be added to the TCP handshake packet according to the network namespace of the current source container (the

preset representation may be in any form, e.g., a fast open option identifier), indicating that this is a modified TCP handshake packet.

5. A network driver component at the receiving end uses netfilter to intercept a response packet of an outbound handshake packet (e.g., "tcp syn ack" as shown in FIG. 1), obtains the attribute label group of the target container and a target port of a current TCP connection according to a network namespace of the target container, generates payload data payload, inserts the payload data after a header of the response packet, recalculates a checksum of the packet, and sends the modified response packet.

6. A network driver layer at the sending end uses netfilter to intercept an inbound response packet of the TCP handshake packet, parses payload in the response packet, and obtains the attribute label group and the target port of the target container. Meanwhile, according to a network namespace of a current network interface, the attribute label group of the source container is obtained from a local cache.

7. The sending end matches the attribute label group of the source container and the attribute label group of the target container in the rule label tree, and matches the target port against a target port set in the container network rule set, to obtain a final matching result. According to the matching result, a corresponding container isolation action is determined and performed. The container isolation action includes any one of the following: blocking, allowing, auditing, and discarding. A corresponding container isolation action may be configured for each container network rule. For example, if rule 1 (not shown in the figure) is matched successfully, and the corresponding container isolation action is allowing, a confirmation packet (e.g., "tcp ack" as shown in FIG. 1) is sent to the receiving end to complete a three-way handshake. If rule 1 is not matched successfully, the response packet is discarded, and the source container and the target container are unable to establish a connection.

8. An execution result is sent to the application layer component for result display.

In the embodiment of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby implementing container identity confirmation of both parties of a network connection. Without affecting handshake performance, rule matching is performed through the container network rule set to perform a container isolation action. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

An embodiment of the present disclosure provides a container isolation method. The method in this embodiment is applied to a sending end of a network request. FIG. 2 is a flowchart of a container isolation method according to an embodiment of the present disclosure, including:

Step S201, obtain a handshake packet and an attribute label group of a source container to which the handshake packet belongs.

The attribute label group includes at least one attribute label, and the attribute label is determined according to container information of the container. The container information includes at least one of the following: container identity (Identity, ID), container network namespace (Network namespace, NS), container name, container image, namespace of a container management platform to which the container belongs (e.g., k8s), and container application information.

Step S202, send the handshake packet with a preset identifier added thereto to a receiving end, where the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet.

The preset identifier may include an identifier in any form, used to indicate that the network packet requires obtaining the attribute label group and the target port of the target container.

Step S203, determine the attribute label group and the target port of the target container according to the response packet.

Specifically, the response packet is parsed to obtain the attribute label group and the target port of the target container.

Step S204, match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set, to obtain a matching result.

The container network rule set includes a plurality of container network rules, and the attribute label group of the source container, and the attribute label group and the target port of the target container are matched against the container network rules to obtain the matching result.

Step S205, perform a container isolation action according to the matching result.

If the attribute label group of the source container, and the attribute label group and the target port of the target container are successfully matched with a container network rule, the container isolation action corresponding to the container network rule is performed.

Exemplarily, the container isolation action includes any one of the following: blocking, allowing, auditing, and discarding.

According to the container isolation method provided by the embodiment of the present disclosure, first, the sending end obtains the handshake packet and the attribute label group of the source container to which the handshake packet belongs; sends the handshake packet with the preset identifier added thereto to the receiving end, where the preset identifier is used for the receiving end to obtain the attribute label group and the target port of the target container and add the attribute label group and the target port of the target container to the response packet of the handshake packet; then, the sending end determines the attribute label group and the target port of the target container according to the response packet; matches the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain the matching result; and performs the container isolation action according to the matching result. In the embodiment of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby implementing container identity confirmation of both parties of the network connection. Without affecting handshake performance, rule matching is performed through the container network rule set, and the container isolation action is performed. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

The specific implementation processes of the above steps are introduced below through various implementations.

In an implementation, the method further includes: obtaining attribute label groups of a plurality of source containers, attribute label groups and target ports of a plurality of target containers; and configuring the container network rule set based on the attribute label groups of the plurality of source containers, the attribute label groups and target ports of the plurality of target containers, and a plurality of container isolation actions.

Each rule in the container network rule set is constituted by the following information: a rule ID, a source label list (i.e., the attribute label group of the source container), a target label list (the attribute label group of the target container), and a target port.

In an example, the container network rule set is as shown in Table 1:

Rule ID	Source label list	Target label list	Target port set

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

Table 1

During rule matching, if a label list of a source container that initiates a network connection matches a source label list in a rule, a label list of a target container matches a target label list in the rule, and a port belongs to a target port set in the rule, the rule is hit, and the container isolation action defined by the rule may be performed. Specifically, by constructing a rule label tree and a simplified rule set, when attribute labels in the container network rule match attribute labels of a container, container network rules corresponding to each attribute label are quickly obtained through a reverse index, and a rule result is quickly matched according to hit counts.

The matching process is specifically described in the following embodiment:

In an implementation, the matching result includes a label matching result and a port matching result; matching the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain the matching result includes: generating a rule label tree according to the container network rule set; matching the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result; and matching the target port against a target port set in the container network rule set to obtain the port matching result; where nodes in the rule label tree are attribute label description information, and the attribute label description information includes: attribute label text, a rule identifier list in which a current attribute label appears in the attribute label group of a source container, and a rule identifier list in which the current attribute label appears in the attribute label group of a target container, where the rule identifier list includes an identifier of at least one container network rule.

In practical application, the application layer receives and processes a user-defined container network rule set, and when the container network rule set changes, updates the latest container network rule set to the network driver component located in the kernel. The network driver component receives the container network rule set delivered by the application layer, and parses it into a rule label tree for optimizing performance in a subsequent rule matching process. The rule label tree is a data structure, which may be a red-black tree obtained through

a red-black algorithm. Each node on the rule label tree is a piece of attribute label description information, including: attribute label text, a rule identifier list in which a current attribute label appears in the attribute label group of the source container, and a rule identifier list in which the current attribute label appears in the attribute label group of the target container.

In an example, the rule label tree obtained according to the container network rule set in Table 1 is as shown in FIG. 3. Label 1, label 2... label 6 represent label text, the rule identifier corresponding to the source container represents the rule identifier list in which the label appears in the attribute label group of the source container, and the rule identifier corresponding to the target container represents the rule identifier list in which the label appears in the attribute label group of the target container. As shown in Table 1, label 1 appears in the attribute label groups of the source containers of rule 1 and rule 2, and appears in the attribute label group of the target container of rule 1. Label 2 appears in the attribute label group of the source container of rule 1, and appears in the attribute label group of the target container of rule 2. Label 3 appears in the attribute label group of the source container of rule 1, and appears in the attribute label group of the target container of rule 2. Label 4 does not appear in the attribute label group of the source container of any rule in Table 1, and appears in the attribute label groups of the target containers of rule 1 and rule 2. Label 5 appears in the attribute label group of the source container of rule 2, and appears in the attribute label group of the target container of rule 1. Label 6 appears in the attribute label group of the source container of rule 2, and does not appear in the attribute label group of the target container of any rule in Table 1.

In an implementation, matching the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result includes: in the rule label tree, querying a first rule identifier list corresponding to attribute label groups of a plurality of source containers that match the attribute label group of the current source container, and querying a second rule identifier list corresponding to attribute label groups of a plurality of target containers that match the attribute label group of the current target container; and obtaining the label matching result according to the first rule identifier list and the second rule identifier list.

In practical application, in the rule label tree, container network rules corresponding to attribute label groups of a plurality of source containers that match the attribute label group of the current source container are queried to obtain an identifier list of the container network rules, i.e., the first rule identifier list, and container network rules corresponding to attribute label groups of a plurality of target containers that match the attribute label group of the current target container are queried to obtain an identifier list of the container network rules, i.e., the

second rule identifier list; and the label matching result is calculated according to the first rule identifier list and the second rule identifier list.

In an implementation, obtaining the label matching result according to the first rule identifier list and the second rule identifier list includes: obtaining, according to an intersection of the first rule identifier list and the second rule identifier list, a list of counts of a plurality of rule identifiers being hit during the matching process; generating a simplified rule set according to the rule label tree, where the simplified rule set includes the number of attribute labels of the source container and the number of attribute labels of the target container in each rule in the container network rule set; and comparing the list of counts of the plurality of rule identifiers being hit with the simplified rule set to obtain the label matching result.

For any label text, a set of identifiers of rules in which it appears in the attribute label groups of source containers and a set of identifiers of rules in which it appears in the attribute label groups of target containers in the container network rule set may be quickly retrieved in the rule label tree. Meanwhile, a simplified rule set is generated, and each rule in the simplified rule set records the number of labels in the attribute label group of the source container and the number of labels in the attribute label group of the target container that it has.

In an example, the simplified rule set obtained according to the rule label tree shown in FIG. 3 is as shown in Table 2, where the number of labels in the source label list represents the number of labels in the attribute label group of the source container, and the number of labels in the target label list represents the number of labels in the attribute label group of the target container.

Rule ID	Number of labels in source label list	Number of labels in target label list	Target port set
1	3	3	[22,80,8080...]
2	3	3	[22,80,8080...]
3	....	....	....

Table 2

During rule matching, first, each label in the attribute label group of the source container is traversed, and the label is used to quickly find its label description information in the rule label tree, and the list of rule identifiers in the attribute label group of the source container in the label description information is obtained. For example, a network connection is initiated from a container with an attribute label group of (label1, label2, label3) to another container

with an attribute label group of (label1, label4, label5), and during matching, the attribute labels of the source container to be matched and the list of rule identifiers corresponding to the source container in the label tree are obtained, as shown in Table 3:

Attribute label of source container to be matched	Rule identifier corresponding to source container in label tree
label1	1, 2
label2	1
label3	1

Table 3

Similarly, the same operation is performed on the attribute label group of the target container to obtain the attribute labels of the target container to be matched and the list of rule identifiers corresponding to the target container in the label tree, as shown in Table 4:

Attribute label of target container to be matched	Rule identifier corresponding to target container in label tree
label1	1
label4	1, 2
label5	1

Table 4

Intersection counting is performed on the rule identifiers corresponding to the source container and the rule identifiers corresponding to the target container in the label tree in the two obtained matching process tables, Table 3 and Table 4, to obtain a list of counts of each rule ID being hit in the matching, as shown in Table 5:

Rule ID	Count of hits on attribute labels of source container of this rule in this matching	Count of hits on attribute labels of target container of this rule in this matching
1	3	3
2	1	1

Table 5

The rule matching hit count table (Table 5) and the simplified rule set (Table 2) are compared for matching, and if the hit counts of the attribute labels of the source container and the attribute labels of the target container in Table 5 are respectively equal to the number of

labels in the attribute label group of the source container and the number of labels in the attribute label group of the target container in the container network rule in the simplified rule set, and the target port belongs to the target port set, the container network rule is considered to be hit, and the container isolation action defined by the container network rule may be performed.

In this example, the attribute labels of the source container of rule 1 are hit 3 times, the attribute labels of the target container are hit 3 times, rule 1 is matched successfully, and the container isolation action defined by rule 1 may be performed.

Optionally, a whitelist mode is adopted in a container firewall, that is, network connections matching a container network rule are allowed, and other network connections not in the container network rule set are subjected to a default action predefined by the rule set, such as blocking, auditing, or alerting.

In an implementation, obtaining the attribute label group of the source container includes: obtaining a network namespace identifier of the source container to which the handshake packet belongs, and obtaining the attribute label group of the source container according to the network namespace identifier.

In practical application, isolation of different independent network spaces on the same host is implemented by kernel network namespaces allocated by a host operating system to each container or container group. The network namespace identifier of each container may be used as a unique identifier describing one container or one container group at a kernel network driver layer, a mapping relationship between network namespace identifiers and attribute label groups of containers is pre-established, and the attribute label group of a container may be obtained in memory according to the network namespace identifier.

An embodiment of the present disclosure provides a container isolation method. The method in this embodiment may be applied to a receiving end of a network request. FIG. 4 shows a flowchart of a container isolation method according to an embodiment of the present disclosure, including:

Step S401, receive a handshake packet sent by a sending end, and if the handshake packet contains a preset identifier, obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet.

The attribute label group includes at least one attribute label, and the attribute label is determined according to container information of the container. The container information includes at least one of the following: container identity (Identity, ID), container network

namespace (Network namespace, NS), container name, container image, namespace of a container management platform to which the container belongs (e.g., k8s), and container application information.

The preset identifier may include an identifier in any form, used to indicate that the network packet requires obtaining the attribute label group and the target port of the target container.

Step S402, return the response packet to the sending end, where the response packet is used for the sending end to determine the attribute label group and the target port of the target container, and to match an attribute label group of a source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and to perform a container isolation action according to the matching result.

If the attribute label group of the source container, and the attribute label group and the target port of the target container are successfully matched with a container network rule, a container isolation action corresponding to the container network rule is performed.

Exemplarily, the container isolation action includes any one of the following: blocking, allowing, auditing, and discarding.

According to the container isolation method provided by the embodiment of the present disclosure: first, the handshake packet sent by the sending end is received, and if the handshake packet contains the preset identifier, the attribute label group and the target port of the target container are obtained and added to the response packet of the handshake packet; then, the response packet is returned to the sending end, where the response packet is used for the sending end to determine the attribute label group and the target port of the target container, and to match the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain a matching result, and to perform the container isolation action according to the matching result. In the embodiment of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby implementing container identity confirmation of both parties of a network connection. Without affecting handshake performance, rule matching is performed through the container network rule set, and the container isolation action is performed. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

In an implementation, obtaining the attribute label group and the target port of the target container and adding the attribute label group and the target port of the target container to the response packet of the handshake packet includes: obtaining the attribute label group and the target port of the target container, and generating payload data; and adding the payload data to a packet header of the response packet, and calculating a checksum of the response packet after adding the payload data.

Specifically, a network driver component at the receiving end uses netfilter to intercept a response packet of an outbound handshake packet, obtains the attribute label group of the target container and a target port of a current TCP connection according to a network namespace of the target container, generates payload data payload, inserts the payload data after the header of the response packet, and then recalculates the checksum of the response packet, and sends the modified response packet.

Corresponding to the application scenario and the method provided by the embodiments of the present disclosure, an embodiment of the present disclosure further provides a container isolation apparatus. As shown in FIG. 5, which is a structural block diagram of a container isolation apparatus according to an embodiment of the present disclosure, the apparatus includes:

an obtaining module 501, configured to obtain a handshake packet and an attribute label group of a source container to which the handshake packet belongs;

a sending module 502, configured to send the handshake packet with a preset identifier added thereto to a receiving end, where the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet;

a determining module 503, configured to determine the attribute label group and the target port of the target container according to the response packet;

a matching module 504, configured to match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set, to obtain a matching result; and

a performing module 505, configured to perform a container isolation action according to the matching result.

According to the container isolation apparatus provided by the embodiment of the present disclosure: first, the sending end obtains the handshake packet and the attribute label group of the source container to which the handshake packet belongs; sends the handshake packet with the preset identifier added thereto to the receiving end, where the preset identifier is used for

the receiving end to obtain the attribute label group and the target port of the target container and add the attribute label group and the target port of the target container to the response packet of the handshake packet; then, the sending end determines the attribute label group and the target port of the target container according to the response packet; matches the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain the matching result; and performs the container isolation action according to the matching result. In the embodiment of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby implementing container identity confirmation of both parties of the network connection. Without affecting handshake performance, rule matching is performed through the container network rule set, and the container isolation action is performed. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

In an implementation, the container isolation apparatus is further configured to: obtain attribute label groups of a plurality of source containers, and attribute label groups and target ports of a plurality of target containers; and configure the container network rule set based on the attribute label groups of the plurality of source containers, the attribute label groups and target ports of the plurality of target containers, and a plurality of container isolation actions.

In an implementation, the matching result includes a label matching result and a port matching result; the matching module 504 is configured to: generate a rule label tree according to the container network rule set; match the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result; match the target port against a target port set in the container network rule set to obtain the port matching result; where nodes in the rule label tree are attribute label description information, and the attribute label description information includes: attribute label text, a rule identifier list in which a current attribute label appears in the attribute label group of the source container, and a rule identifier list in which the current attribute label appears in the attribute label group of the target container, where the rule identifier list includes an identifier of at least one container network rule.

In an implementation, when matching the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result, the matching module 504 is configured to: query, in the rule label tree, a first

rule identifier list corresponding to attribute label groups of a plurality of source containers that match the attribute label group of the current source container, and query a second rule identifier list corresponding to attribute label groups of a plurality of target containers that match the attribute label group of the current target container; and obtain the label matching result according to the first rule identifier list and the second rule identifier list.

In an implementation, when obtaining the label matching result according to the first rule identifier list and the second rule identifier list, the matching module 504 is configured to: obtain, according to an intersection of the first rule identifier list and the second rule identifier list, a list of counts of a plurality of rule identifiers being hit during the matching process; generate a simplified rule set according to the rule label tree, where the simplified rule set includes the number of attribute labels of the source container and the number of attribute labels of the target container in each rule in the container network rule set; and compare the list of counts of the plurality of rule identifiers being hit with the simplified rule set to obtain the label matching result.

In an implementation, when obtaining the attribute label group of the source container, the obtaining module 501 is configured to: obtain a network namespace identifier of the source container to which the handshake packet belongs, and obtain the attribute label group of the source container according to the network namespace identifier.

In an implementation, the container isolation action includes any one of the following: blocking, allowing, auditing, and discarding.

For the functions of each module in the embodiments of the present disclosure, reference may be made to the corresponding description in the above method, and the corresponding beneficial effects are achieved, which are not repeated herein.

Corresponding to the application scenario and the method provided by the embodiments of the present disclosure, an embodiment of the present disclosure further provides a container isolation apparatus. As shown in FIG. 6, which is a structural block diagram of a container isolation apparatus according to an embodiment of the present disclosure, the apparatus includes:

an obtaining module 601, configured to receive a handshake packet sent by a sending end, and if the handshake packet contains a preset identifier, obtain an attribute label group and a target port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet;

a sending module 602, configured to return the response packet to the sending end, where the response packet is used for the sending end to determine the attribute label group and the

target port of the target container, and to match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and to perform a container isolation action according to the matching result.

According to the container isolation apparatus provided by the embodiment of the present disclosure: first, the handshake packet sent by the sending end is received, and if the handshake packet includes the preset identifier, the attribute label group and the target port of the target container are obtained and added to the response packet of the handshake packet; then, the response packet is returned to the sending end, where the response packet is used for the sending end to determine the attribute label group and the target port of the target container, and to match the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain the matching result, and to perform the container isolation action according to the matching result. In the embodiment of the present disclosure, the handshake packet is modified by adding the preset identifier, and the attribute label group and the target port of the target container of a network peer are carried in the response packet of the handshake packet, thereby implementing container identity confirmation of both parties of a network connection. Without affecting handshake performance, rule matching is performed through the container network rule set, and the container isolation action is performed. Without depending on a network plugin of a container network, container isolation can be implemented in an environment where the network plugin is not supported.

In an implementation, when obtaining the attribute label group and the target port of the target container and adding the attribute label group and the target port of the target container to the response packet of the handshake packet, the obtaining module 601 is configured to: obtain the attribute label group and the target port of the target container, and generate payload data; and add the payload data to a packet header of the response packet, and calculate a checksum of the response packet after adding the payload data.

For the functions of each module in the embodiment of the present disclosure, reference may be made to the corresponding description in the above method, and the corresponding beneficial effects are achieved, which are not repeated herein.

FIG. 7 is a block diagram of an electronic device used to implement the embodiments of the present disclosure. As shown in FIG. 7, the electronic device includes: a memory 710 and a processor 720, where the memory 710 stores a computer program executable on the processor 720. When the processor 720 executes the computer program, the method in the above

embodiments is implemented. The number of the memory 710 and the number of the processor 720 may be one or more.

The electronic device further includes:

a communication interface 730, configured to communicate with an external device and perform data interaction transmission.

If the memory 710, the processor 720, and the communication interface 730 are implemented independently, the memory 710, the processor 720, and the communication interface 730 may be connected to each other through a bus and perform communication with each other. The bus may be an industry standard architecture (Industry Standard Architecture, ISA) bus, a peripheral component interconnect (Peripheral Component Interconnect, PCI) bus, an extended industry standard architecture (Extended Industry Standard Architecture, EISA) bus, or the like. The bus may be divided into an address bus, a data bus, a control bus, etc. For ease of representation, only one thick line is used in FIG. 7, but this does not mean that there is only one bus or only one type of bus.

Optionally, in a specific implementation, if the memory 710, the processor 720, and the communication interface 730 are integrated on one chip, the memory 710, the processor 720, and the communication interface 730 may perform communication with each other through an internal interface.

An embodiment of the present disclosure provides a container isolation system, including a sending end and a receiving end; the sending end executes the method provided by the embodiments of the present disclosure; the receiving end executes the method provided by the embodiments of the present disclosure.

An embodiment of the present disclosure provides a computer-readable storage medium storing a computer program, where when the program is executed by a processor, the method provided in the embodiments of the present disclosure is implemented.

An embodiment of the present disclosure provides a computer program product, where the computer program product includes a computer program, and when the computer program is executed by a processor, the method provided in the embodiments of the present disclosure is implemented.

An embodiment of the present disclosure further provides a chip, where the chip includes a processor, configured to call instructions stored in a memory from the memory and run the instructions, so that a communication device installed with the chip executes the method provided by the embodiments of the present disclosure.

An embodiment of the present disclosure further provides a chip, including: an input interface, an output interface, a processor, and a memory, where the input interface, the output interface, the processor, and the memory are connected to each other through an internal connection path, and the processor is configured to execute code in the memory, and when the code is executed, the processor is configured to execute the method provided by the embodiments of the present disclosure.

It should be understood that the above processor may be a central processing unit (Central Processing Unit, CPU), or may be another general-purpose processor, a digital signal processor (Digital Signal Processor, DSP), an application specific integrated circuit (Application Specific Integrated Circuit, ASIC), a field programmable gate array (Field Programmable Gate Array, FPGA) or another programmable logic device, a discrete gate or transistor logic device, a discrete hardware component, etc. A general-purpose processor may be a microprocessor or any conventional processor, etc. It is worth noting that the processor may be a processor supporting an advanced reduced instruction set computer machines (Advanced RISC Machines, ARM) architecture.

Further, optionally, the above memory may include a read-only memory and a random access memory. The memory may be a volatile memory or a non-volatile memory, or may include both volatile and non-volatile memory. The non-volatile memory may include a read-only memory (Read-Only Memory, ROM), a programmable ROM (Programmable ROM, PROM), an erasable PROM (Erasable PROM, EPROM), an electrically EPROM (Electrically EPROM, EEPROM), or a flash memory. The volatile memory may include a random access memory (Random Access Memory, RAM), which is used as an external cache. By way of example and not limitation, many forms of RAM are available, for example, a static RAM (Static RAM, SRAM), a dynamic random access memory (Dynamic Random Access Memory, DRAM), a synchronous DRAM (Synchronous DRAM, SDRAM), a double data rate SDRAM (Double Data Rate SDRAM, DDR SDRAM), an enhanced SDRAM (Enhanced SDRAM, ESDRAM), a sync link DRAM (Sync link DRAM, SLDRAM), and a direct rambus RAM (Direct Rambus RAM, DR RAM).

In the above embodiments, the implementation may be achieved wholly or partially through software, hardware, firmware, or any combination thereof. When implemented using software, the implementation may be achieved wholly or partially in the form of a computer program product. The computer program product includes one or more computer instructions. When the computer program instructions are loaded and executed on a computer, the processes or functions according to the present disclosure are wholly or partially produced. The computer

may be a general-purpose computer, a special-purpose computer, a computer network, or other programmable apparatus. The computer instructions may be stored in a computer-readable storage medium, or transmitted from a computer-readable storage medium to another computer-readable storage medium.

In the description of this specification, descriptions with reference to the terms "one embodiment", "some embodiments", "example", "specific example", or "some examples" etc. mean that specific features, structures, materials, or characteristics described in connection with the embodiment or example are included in at least one embodiment or example of the present disclosure. Moreover, the described features, structures, materials, or characteristics may be combined in any suitable manner in any one or more embodiments or examples. Furthermore, those skilled in the art can integrate and combine different embodiments or examples described in this specification and features of different embodiments or examples, as long as they are not contradictory.

Furthermore, the terms "first" and "second" are used for descriptive purposes only, and are not to be understood as indicating or implying relative importance or implicitly specifying the number of indicated technical features. Thus, features defined with "first" and "second" may explicitly or implicitly include at least one such feature. In the description of the present disclosure, "a plurality of" means two or more, unless otherwise explicitly and specifically defined.

Any process or method described in the flowcharts or otherwise described herein may be understood as representing inclusion of one or more modules, segments, or portions of code of executable instructions for implementing specific logical functions or steps of the process. Furthermore, the scope of preferred embodiments of the present disclosure includes alternative implementations in which functions may be performed not in the order shown or discussed, including in a substantially concurrent manner or in reverse order depending on the functions involved.

The logic and/or steps described in the flowcharts or otherwise described herein, for example, may be considered as an ordered listing of executable instructions for implementing logical functions, and may be embodied in any computer-readable medium for use by or in connection with an instruction execution system, apparatus, or device (such as a computer-based system, a system including a processor, or other system that can fetch instructions from and executes instructions of the instruction execution system, apparatus, or device).

It should be understood that the various parts of the present disclosure may be implemented using hardware, software, firmware, or a combination thereof. In the above

embodiments, a plurality of steps or methods may be implemented using software or firmware stored in a memory and executed by a suitable instruction execution system. All or some of the steps of the method in the above embodiments may be completed by instructing relevant hardware through a program, where the program may be stored in a computer-readable storage medium, and when executed, the program includes one of the steps of the method embodiments or a combination thereof.

Furthermore, functional units in various embodiments of the present disclosure may be integrated in one processing module, or each unit physically may exist individually, or two or more units may be integrated in one module. The integrated module may be implemented in the form of hardware, or may be implemented in the form of a software functional module. If the integrated module is implemented in the form of a software functional module and sold or used as an independent product, it may also be stored in a computer-readable storage medium. The storage medium may be a read-only memory, a magnetic disk, an optical disk, or the like.

The above descriptions are merely exemplary embodiments of the present disclosure, but the protection scope of the present disclosure is not limited thereto. Any person skilled in the art who is familiar with the present technical field may readily conceive of various changes or replacements within the technical scope disclosed in the present disclosure, all of which should fall within the protection scope of the present disclosure. Therefore, the protection scope of the present disclosure is subject to the protection scope of the claims.

CLAIMS

1. A container isolation method, wherein the method is applied to a sending end, the method comprising:

obtaining a handshake packet and an attribute label group of a source container to which the handshake packet belongs;

sending the handshake packet with a preset identifier added thereto to a receiving end, wherein the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and to add the attribute label group and the target port of the target container to a response packet of the handshake packet;

determining, according to the response packet, the attribute label group and the target port of the target container;

matching the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result; and

performing a container isolation action according to the matching result.

2. The method according to claim 1, wherein the method further comprises:

obtaining attribute label groups of a plurality of source containers, and attribute label groups and target ports of a plurality of target containers; and

configuring the container network rule set based on the attribute label groups of the plurality of source containers and the attribute label groups and the target ports of the plurality of target containers.

3. The method according to claim 1 or 2, wherein the matching result comprises a label matching result and a port matching result; and the matching the attribute label group of the source container, and the attribute label group and the target port of the target container against the container network rule set to obtain the matching result comprises:

generating a rule label tree according to the container network rule set;

matching the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result;

matching the target port against a target port set in the container network rule set to obtain the port matching result;

wherein nodes in the rule label tree are attribute label description information, and the attribute label description information comprises: attribute label text, a rule identifier list in which a current attribute label appears in the attribute label group of the source container, and

a rule identifier list in which the current attribute label appears in the attribute label group of the target container, wherein the rule identifier list comprises an identifier of at least one container network rule.

4. The method according to claim 3, wherein the matching the attribute label group of the source container and the attribute label group of the target container against the rule label tree to obtain the label matching result comprises:

querying, in the rule label tree, a first rule identifier list corresponding to a plurality of attribute label groups of source containers that match the attribute label group of the source container, and querying a second rule identifier list corresponding to a plurality of attribute label groups of target containers that match the attribute label group of the target container; and

obtaining the label matching result according to the first rule identifier list and the second rule identifier list.

5. The method according to claim 4, wherein the obtaining the label matching result according to the first rule identifier list and the second rule identifier list comprises:

obtaining, according to an intersection of the first rule identifier list and the second rule identifier list, a list of counts of a plurality of rule identifiers being hit during a matching process;

generating a simplified rule set according to the rule label tree, wherein the simplified rule set comprises a number of attribute labels of a source container and a number of attribute labels of a target container in each rule of the container network rule set; and

comparing the list of counts of the plurality of rule identifiers being hit with the simplified rule set to obtain the label matching result.

6. The method according to any one of claims 1 to 5, wherein the obtaining the attribute label group of the source container comprises:

obtaining a network namespace identifier of the source container to which the handshake packet belongs, and obtaining the attribute label group of the source container according to the network namespace identifier.

7. The method according to any one of claims 1 to 5, wherein the container isolation action comprises any one of: blocking, allowing, auditing, and discarding.

8. A container isolation method, wherein the method is applied to a receiving end, the method comprising:

receiving a handshake packet sent by a sending end, and if the handshake packet contains a preset identifier, obtaining an attribute label group and a target port of a target container and

adding the attribute label group and the target port of the target container to a response packet of the handshake packet; and

returning the response packet to the sending end, wherein the response packet is used for the sending end to determine the attribute label group and the target port of the target container, to match an attribute label group of a source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and to perform a container isolation action according to the matching result.

9. The method according to claim 8, wherein the obtaining the attribute label group and the target port of the target container and adding the attribute label group and the target port of the target container to the response packet of the handshake packet comprises:

obtaining the attribute label group and the target port of the target container and generating payload data; and

adding the payload data to a packet header of the response packet, and calculating a checksum of the response packet after adding the payload data.

10. A container isolation apparatus, wherein the apparatus is applied to a sending end, the apparatus comprising:

an obtaining module, configured to obtain a handshake packet and an attribute label group of a source container to which the handshake packet belongs;

a sending module, configured to send the handshake packet with a preset identifier added thereto to a receiving end, wherein the preset identifier is used for the receiving end to obtain an attribute label group and a target port of a target container and to add the attribute label group and the target port of the target container to a response packet of the handshake packet;

a determining module, configured to determine, according to the response packet, the attribute label group and the target port of the target container;

a matching module, configured to match the attribute label group of the source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result; and

a performing module, configured to perform a container isolation action according to the matching result.

11. A container isolation apparatus, wherein the apparatus is applied to a receiving end, the apparatus comprising:

an obtaining module, configured to receive a handshake packet sent by a sending end, and if the handshake packet contains a preset identifier, obtain an attribute label group and a target

port of a target container and add the attribute label group and the target port of the target container to a response packet of the handshake packet; and

a sending module, configured to return the response packet to the sending end, wherein the response packet is used for the sending end to determine the attribute label group and the target port of the target container, to match an attribute label group of a source container, and the attribute label group and the target port of the target container against a container network rule set to obtain a matching result, and to perform a container isolation action according to the matching result.

12. A container isolation system, wherein the container isolation system comprises a sending end and a receiving end; the sending end is configured to execute the method according to any one of claims 1 to 7; and the receiving end is configured to execute the method according to any one of claims 8 to 9.

13. An electronic device, wherein the electronic device comprises a memory, a processor, and a computer program stored on the memory, and the processor, when executing the computer program, implements the method according to any one of claims 1 to 9.

14. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed by a processor, the method according to any one of claims 1 to 9 is implemented.

15. A computer program product, wherein the computer program product comprises a computer program, and when the computer program is executed by a processor, the method according to any one of claims 1 to 9 is implemented.