Ubiquitous Cloud Native Service

Product Bulletin

Issue	01
Date	2024-03-06

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: https://www.huawei.com

Email: support@huawei.com

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Vulnerability Notices

1.1 Notice on HTTP/2 Rapid Reset Attack Vulnerability (CVE-2023-4487)

Details

This HTTP/2 vulnerability allows malicious actors to launch a DDoS attack targeting HTTP/2 servers. The attack sends a group of HTTP requests using HEADERS and RST_STREAM and repeating this pattern to generate a high volume of traffic on the targeted HTTP/2 servers. By packing multiple HEADERS and RST_STREAM frames in a single connection, attackers can cause a significant increase in the request per second and high CPU utilization on the servers that eventually can cause resource consumption. This results in service request rejection.

Table 1-1 Vulnerability information

Vulnerability Name	CVE-ID	Severity	Discovered
HTTP/2 Rapid Reset Attack Vulnerability	**CVE-2023-44487**	High	2023-10-10

Impact

This DDoS attack does not lead to the compromise of user data. However, malicious attackers may exploit this vulnerability to launch DDoS attacks targeting HTTP/2 servers, causing the servers to break down.

Solution

Harden security group protections in your VPC, so that interfaces are exposed only to trusted users.

Reference

HTTP/2 Rapid Reset Attack Vulnerability

Technical Details

The HTTP/2 protocol allows multiple requests or responses over a single connection. Each HTTP request or response uses a unique data stream. A data stream on a connection is called a data frame. Each data frame contains a fixed header, which specifies the data frame type and the ID of the data stream that the data frame belongs to. **Table 1-2** lists some important data frame types.

Table 1-2 Important data frame types

Type	Function
SETTINGS	Used to communicate configuration parameters for the HTTP2 connection.
HEADERS	Used to communicate header fields for a stream.
DATA	Used to transport HTTP message bodies.
RST_STREAM	Used to signal termination of a stream. The client can send an RST_STREAM frame to signal the server to cancel the stream. In this case, the stream is no longer active.

Assume that the maximum number of concurrent streams set for the current TCP connection is 1. After sending request 1, the client sends request 2 immediately. In this case, the server does not process request 2 but directly responds to the RST_STREAM frame. If the client sends the RST_STREAM frame immediately after sending a request, the client can continuously send requests to the server without waiting for any response. The server, however, is trapped in a loop of continuously receiving requests, processing requests, and directly ending requests. This process consumes some system resources.

Malicious attackers can exploit this vulnerability to continuously pack HEADERS and RST_STREAM frames to consume server resources, affecting the processing of normal requests by the server and causing DDoS attacks.

NOTE

- Maximum number of concurrent streams: HTTP/2 allows you to set the maximum number of concurrent streams on a TCP connection to limit the number of requests.
- DDoS attack: occurs when multiple machines are operating together to attack one target to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

1.2 Impact of the runC Vulnerability on UCS (CVE-2024-21626)

Details

runC is a lightweight tool for running containers. It implements the Open Container Initiative (OCI) specification. runC is the core and basic component of container software such as Docker, containerd, and Kubernetes. Recently, the runC community released the latest version to fix a high-risk container escape vulnerability (**CVE-2024-21626**). Due to an internal file descriptor leak, an attacker could control and set the working directory or the command path of a container process to the path under the parent directory of the file descriptor. This allows the container to read and write any files from and into the node, resulting in a container escape.

Table 1-3 Vulnerability information

Vulnerability Name	CVE-ID	Severity	Discovered
runC vulnerability	**CVE-2024-21626**	High	2024-02-01

Vulnerability Exploitation Conditions

UCS services in normal usage are not affected by this vulnerability. An attacker can exploit this vulnerability only when either of the following conditions is met:

- The attacker can create or update workloads in a cluster.
- The image source of a container that runs a workload is untrusted, which enables an attacker to modify the source image.

Impact

If either of the preceding exploitation conditions is met, a container process may escape to the node, resulting in node information leakage or malicious command execution.

The following shows the common ways in which exploitation can occur:

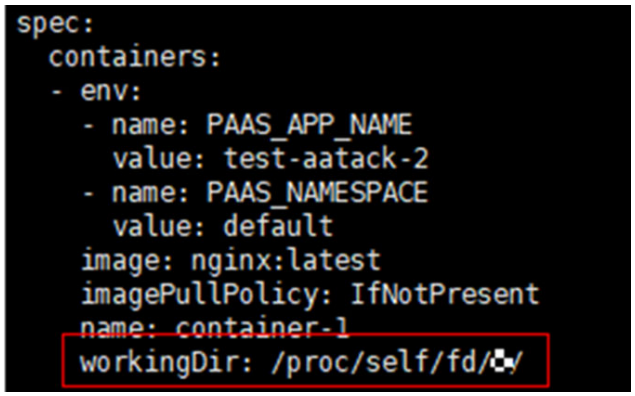
- An attacker, with permissions to create or update workloads in a cluster, sets **WORKDIR** of a container process to **/proc/self/fd/***<num>* during workload creation to access the node file system after the container runs.
- An attacker modifies an untrusted source container image of a workload and sets **WORKDIR** of the image to **/proc/self/fd/***<num>* to access the node file system after the container built from this image runs.

Identification Method

The risks may be present if workload configurations or container images in on-premises clusters on **Huawei Cloud (Chinese Mainland)** and multi-cloud clusters on **Huawei Cloud (International)** have either of the following characteristics:

- **WORKDIR** of a container process in a workload is set to **/proc/self/fd/***<num>*.

  Figure 1-1 Configurations of a workload with security risks

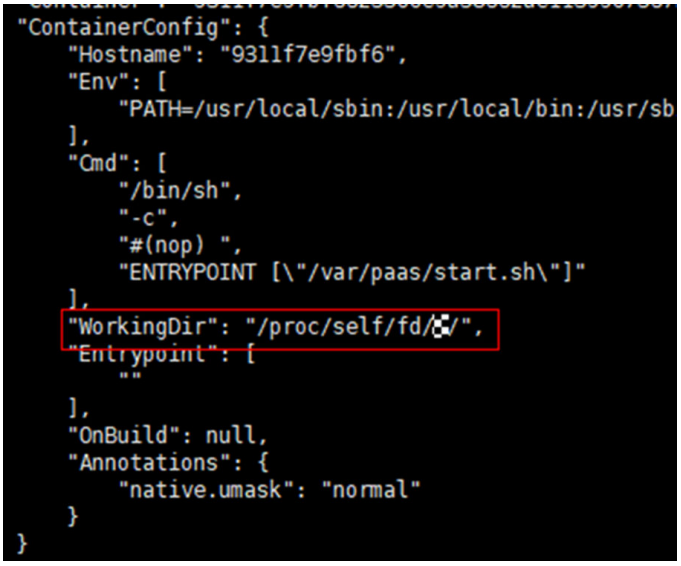
  

- The default value of **WORKDIR** or startup command of a container image in a workload contains **/proc/self/fd/***<num>*.

  View the container image metadata.

  - For a Docker container: **docker inspect** *<Image ID>*
  - For a containerd container: **crictl inspecti** *<Image ID>*

  Figure 1-2 Configurations of a workload with security risks

Solution

Preventive measures

- Set **WORKDIR** of a workload to a fixed directory.

- If **WORKDIR** is not set for a workload, ensure that the container images used by the workload are trusted.

NOTE

Before taking the preventive measures, evaluate the impact on services and perform tests.

Rectification method

This vulnerability has been fixed in UCS. Use the latest versions of on-premises clusters and multi-cloud clusters.

Reference

runC Container Escape Vulnerability (CVE-2024-21626)

