Troubleshoot Fault ID 11 on SUSE Linux Secure Endpoint

Contents

Introduction

This document describes the process to resolve Fault ID 11 of Secure Endpoint on SUSE Linux Enterprise 15 SP2 .

Requirements

The command-line interface (CLI) is available for all users of a system, although the availability of some commands depends on policy configuration and/or root permissions. The commands dependent on this are disclosed throughout this article.

Cisco recommends that you have knowledge of these topics:

- Linux Command Line
- Secure Endpoint

Components Used

The information used in the document is based on these software versions:

- Secure Endpoint 1.20
- SUSE Linux Enterprise 15 SP2 kernel version 5.3.18-24.96-default

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

On SUSE Linux Enterprise 15 Service Pack (SP) 2 , with kernel versions greater than or equal to 5.3.18, connector uses eBPF modules for real time file system and network monitoring. The eBPF modules replaces the Linux Kernel Modules used when it runs on RHEL 6, RHEL 7, Oracle Linux 7 RHCK, Oracle Linux 7

UEK 5 and earlier, and Amazon Linux 2 kernel 4.14 or earlier. For Ubuntu 18.04 and later, as well as Debian 10 and later, eBPF modules are native.

For proper compatibility, the connector automatically compiles the eBPF modules used by the connector before it loads and runs them on the system. This compilation requires that kernel development header files that correspond to the current kernel-devel are installed. When real time filesystem and network monitoring is enabled, the connector compiles the eBPF modules each time the connector is started, or in real time when these features are enabled, as part of a policy update.

When the system misses the current kernel-devel package, the connector raises Fault ID 11: Realtime network and file monitoring is unavailable. Install the kernel-devel package for the currently-running kernel then restart the Connector. The problem with this Fault is that the Linux connector runs in a degraded state, which means that it does not work as expected until the fault is resolved.

Troubleshoot

If fault 11 is raised, then this error log appears:

- Look for log lines in the system log /var/log/messages that are similar to this:

init: cisco-amp pre-start: AMP kernel modules are not required on this kernel version '5.3.18-24.96-default'; skipping reinstalling kernel modules

The log states that the current kernel version on the computer does not use kernel modules for filesystem and network monitoring. On kernel versions greater than or equal to 4.18, the filesystem and network are monitored with the use of eBPF modules.

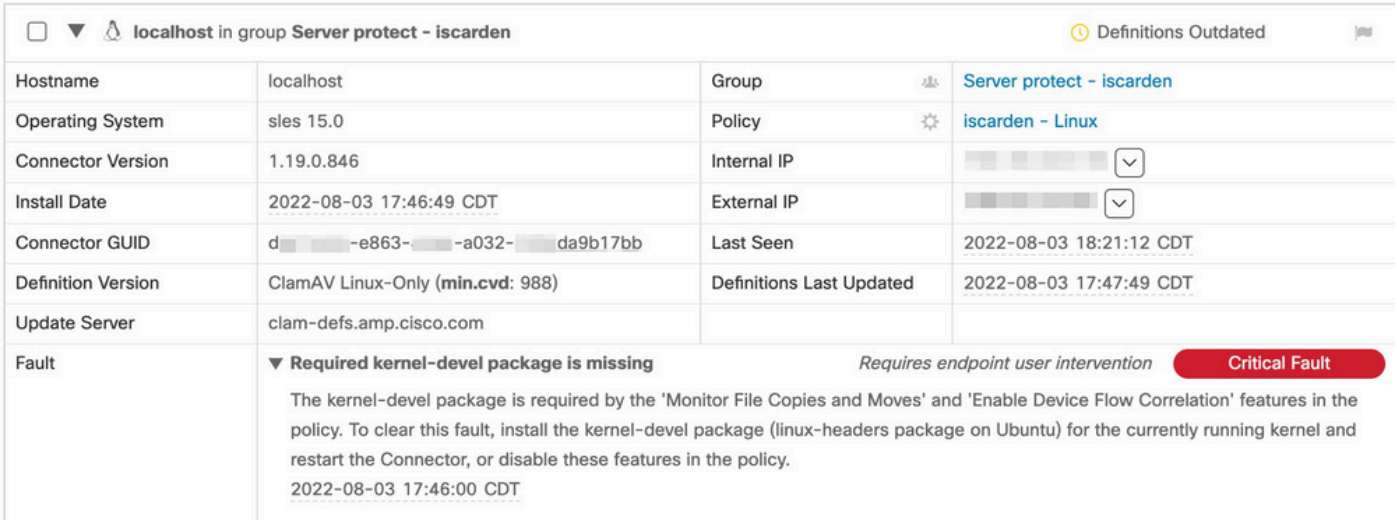
How to Identify Absent Kernel-Headers

When the connector runs on a computer without kernel headers, Fault ID 11 (Realtime network and file monitoring is unavailable), the connector runs in a degraded state without filesystem or network monitoring. These steps can be performed from a terminal window in order to identify whether the connector kernel-header is present or not.

Step 1. From the affected device, verify that the connector has Fault ID 11 :

/opt/cisco/amp/bin/ampcli # status [logger] Set minimum reported log level to notice Trying to connect... Connected. Status: Connected Mode: Degraded Scan: Ready for scan Last Scan: 2022-08-03 06:31:42 PM Policy: iscarden - Linux (#22192) Command-line: Enabled Orbital: Disabled Faults: 1 Critical Fault IDs: 11 ID 11 - Critical: Realtime network and file monitoring is unavailable. Install the kernel-devel package for the currently running kernel, then, restart the Connector.

From the Secure Endpoint console, find the affected device and expand the details to verify the Fault section.



Step 2. Check the current kernel with this command:

$ uname -r 5.3.18-150200.24.115-default

Step 3. In order to check whether the kernal headers are installed or not:

zypper se -s kernel-default-devel | grep $(uname -r | sed "s/-default//") # zypper se -s kernel-devel | grep $(uname -r | sed "s/-default//")

The output must be like this:

```
isaac@localhost:~> zypper se -s kernel-default-devel | grep $(uname -r | sed "s/-default//")
i+ | kernel-default-devel | package | 5.3.18-24.96.1   | x86_64 | SLE-Module-Basesystem15-SP2-Updates
```

Where the i+ signifies that the package is installed. If the left-hand column is v or is **blank**, the package must be installed.

The SUSE computer is suitable for the installation of kernel headers if all of these are true:

- The connector has Fault ID 11.
- The minimum kernel version is 5.3.18.
- The kernel headers are not installed.

Resolution

If the SUSE machine does not have the required kernel headers, then this procedure can be used to install the required kernel headers on the machine.

Step 1. Install the necessary kernel headers:

sudo zypper install --oldpackage kernel-default-devel=$(uname -r | sed 's/-default//') # sudo zypper install --oldpackage kernel-devel=$(uname -r | sed 's/-default//')

Step 2. Restart the connector:

sudo systemctl stop cisco-amp # sudo systemctl start cisco-amp

Step 3. Confirm the fault is cleared:

/opt/cisco/amp/bin/ampcli # status Trying to connect... Connected. ampcli> status Status: Connected Mode: Normal Scan: Ready for scan Last Scan: 2022-08-05 01:29:47 PM Policy: iscarden - Linux (#22201) Command-line: Enabled Orbital: Disabled Faults: None ampcli > quit

Verify

In order to verify if the kernel headers are now installed, run these commands:

zypper se -s kernel-default-devel | grep $(uname -r | sed "s/-default//") # zypper se -s kernel-devel | grep $(uname -r | sed "s/-default//")

Before you performed the workaround, you had an output similar to this:

```
isaac@localhost:~> zypper se -s kernel-default-devel | grep $(uname -r | sed 's/
-default//') $ zypper se -s kernel-devel | grep $(uname -r | sed 's/-default//')
isaac@localhost:~>
```

After you perform the workaround, the output must be similar to this:

```
isaac@localhost:~> zypper se -s kernel-default-devel | grep $(uname -r | sed "s/-default//")
i+ | kernel-default-devel | package | 5.3.18-24.96.1   | x86_64 | SLE-Module-Basesystem15-SP2-Updates
isaac@localhost:~> zypper se -s kernel-devel | grep $(uname -r | sed "s/-default//")
i | kernel-devel | package | 5.3.18-24.96.1   | noarch | SLE-Module-Basesystem15-SP2-Updates
isaac@localhost:~>
```

Related information

- Verify Secure Endpoint Linux Connector OS Compatibility
- Linux Kernel-Devel Fault
- Building Cisco Secure Endpoint Linux Connector Kernel Modules