Troubleshoot Secure Endpoint Linux Connector Fault 11

Contents

Introduction

Background Information

Determine BTF Support

BTF Not Supported

BTF Supported Distributions

Upgrade to a BTF Supported Kernel

- RPM-based Distributions
- Oracle Linux Unbreakable Enterprise Kernel (UEK)
- SUSE Linux Enterprise and openSUSE Leap
- Debian-based Distributions

Install Missing Kernel Header

- RPM-based Distributions
  - Verify
  - Resolution
- Oracle Linux Unbreakable Enterprise Kernel (UEK)
  - Verify
  - Resolution
- SUSE Linux Enterprise and openSUSE Leap
  - Verify
  - Resolution
- Debian-based Distributions
  - Verify
  - Resolution

Introduction

This document describes how to identify, resolve, and verify fault 11 in the Secure Endpoint Linux connector.

Background Information

The connector raises fault 11 whenever it is unable to load the eBPF module needed to monitor system activity and events.

Starting with Linux connector release 1.25, the connector prioritizes eBPF CO-RE (Compile Once – Run Everywhere) to monitor system events on kernels that support BTF, without requiring kernel headers to be installed. This method of system observability is more efficient and is the preferred option.

The connector falls back to the legacy eBPF loading method that requires kernel headers to be installed when using Linux connector releases older than 1.25 or running on a kernel that does not support BTF.

These headers are typically not included in standard distribution installations and need to be installed separately.

Fault 11 is raised if the connector fails to load the eBPF module using either of the approaches, indicating that the system is in an unprotected state. This document details the necessary steps required to resolve this problem.

Determine BTF Support

The connector requires one of these to be true in order to load the eBPF module necessary for monitoring filesystem and network activity:
1. The current kernel has `CONFIG_DEBUG_INFO_BTF` enabled. (This is the preferred option.)
2. The kernel headers for the currenty running kernel version are installed on the system.

To check if your currently running kernel has `CONFIG_DEBUG_INFO_BTF` enabled, run the command:

```
cat /boot/config-$(uname -r) | grep CONFIG_DEBUG_INFO_BTF
```

If you see `CONFIG_DEBUG_INFO_BTF=y` then your kernel supports BTF and the connector can monitor filesystem and network events using eBPF.

BTF Not Supported

If running connector version 1.25 or newer, consider upgrading the kernel to a version that supports BTF. A list of distributions that have BTF enabled by default can be found later in this document.

If your current kernel does not support BTF then the connector requires the kernel header files to be installed. Fault 11 is raised if the connector is unable to locate these required files. Use the steps described in Install the missing kernel header to resolve this issue.

BTF Supported Distributions

These distributions have `CONFIG_DEBUG_INFO_BTF` enabled by default in the latest kernel version:

- Centos/RHEL 8.2 and later
- Ubuntu 20.04 and later
- Oracle Linux 8.5 and later
- Debian 11 and later
- Alma Linux 8.3 and later
- Rocky Linux 8.3 and later
- SUSE Enterprise 15 SP4 / openSUSE Leap 15.4 and later

Upgrade to a BTF Supported Kernel

If you are on a distribution that supports BTF in a later kernel version, it is preferred that you update your kernel in order to resolve fault 11.

Before proceeding to update your kernel, first upgrade your Linux connector to version 1.25.0 or newer to ensure that your connector supports BTF.

RPM-based Distributions

This section is applicable to:

- Alma Linux
- Amazon Linux
- CentOS Linux
- Oracle Linux Red Hat Compatible Kernel (RHCK)
- Red Hat Enterprise Linux
- Rocky Linux

To upgrade to the latest kernel version:

1. Check if there are updates available for the kernel:

   ```
   yum check-update
   ```

   If you see `kernel.x86_64` in the output then there is a kernel upgrade available.
2. Update the kernel package:

   ```
   sudo yum update kernel
   ```

3. Reboot your system:

   ```
   sudo reboot
   ```

4. Verify the new kernel supports BTF and that fault 11 is cleared by the connector.

Oracle Linux Unbreakable Enterprise Kernel (UEK)

To upgrade to the latest kernel version:

1. Check if there are updates available for the kernel:

   ```
   yum check-update
   ```

   If you see `kernel-uek.x86_64` in the output then there is a kernel upgrade available.
2. Update the kernel package:

   ```
   sudo yum update kernel-uek
   ```

3. Reboot your system:

   ```
   sudo reboot
   ```

4. Verify the new kernel supports BTF and that fault 11 is cleared by the connector.

SUSE Linux Enterprise and openSUSE Leap

To upgrade to the latest kernel version:

1. Check your current kernel version:

```
uname -r
```

2. List the available `kernel-default` packages:

   ```
   zypper search -s --match-exact kernel-default
   ```

   The output appears as:

   ```
   <status> | kernel-default       | package | <version> | <arch> | <repository>
   ```

   Find a `version` that is greater than your current kernel version, determined in step 1.
3. Install the new versions of `kernel-default` with the command:

   ```
   sudo zypper install kernel-default=<new-version>
   ```

4. Reboot your system:

   ```
   sudo reboot
   ```

5. Verify the new kernel supports BTF and that fault 11 is cleared by the connector.

Debian-based Distributions

To upgrade to the latest kernel version:

1. Retrieve the latest package lists:

   ```
   sudo apt update
   ```

2. Install the latest packages:

   ```
   sudo apt upgrade
   ```

3. Reboot your system:

   ```
   sudo reboot
   ```

4. Verify the new kernel supports BTF and that fault 11 is cleared by the connector.

Install Missing Kernel Header

Before installing a missing kernel header, try upgrading your system to a BTF supported version. If this does not resolve fault 11, or if your distribution does not yet support BTF in the latest version, then continue to install the missing kernel header.

RPM-based Distributions

This section is applicable to:

- Alma Linux
- Amazon Linux

- CentOS Linux
- Oracle Linux Red Hat Compatible Kernel (RHCK)
- Red Hat Enterprise Linux
- Rocky Linux

Verify

RPM-based distributions require that a `kernel-devel` package is installed for the currently running kernel. To verify if a `kernel-devel` package is installed for the current running kernel, run:

```
rpm -qa | grep kernel-devel-$(uname -r)
```

If there are no results, then the required `kernel-devel` package is missing and needs to be installed.

If the correct `kernel-devel`package is installed and the connector is still raising fault 11, some of the required files could have been tampered with. This command checks if any files have been modified or are missing:

```
rpm --verify --nomtime --noghost kernel-default-devel
```

If any files are missing or modified then the`kernel-devel`package needs to be uninstalled and re-installed to ensure the required files are available. This command provides no output if all the package files are intact and unmodified.

Resolution

To install the required `kernel-devel` package, run the command:

```
sudo dnf install -y kernel-devel-$(uname -r)
```

Reverify. Fault 11 is cleared by the connector after about a minute.

Oracle Linux Unbreakable Enterprise Kernel (UEK)

Verify

Oracle Linux UEK requires that a `kernel-uek-devel` package is installed for the currently running kernel. To verify if a `kernel-uek-devel` package is installed for the current running kernel, run:

```
rpm -qa | grep kernel-uek-devel-$(uname -r)
```

If there are no results, then the required `kernel-uek-devel` package is missing and needs to be installed.

Resolution

To install the required `kernel-uek-devel` package, run the command:

```
sudo dnf install -y kernel-uek-devel-$(uname -r)
```

Reverify. Fault 11 is be cleared by the connector after about a minute.

SUSE Linux Enterprise and openSUSE Leap

Verify

SUSE and openSUSE require that the `kernel-default-devel` package is installed for the currently running kernel. To verify if the `kernel-default-devel` package is installed for the current running kernel, run:

```
zypper search -si kernel-default-devel | grep $(uname -r | sed "s/-default//")
```

If there are no results, then the required `kernel-default-devel` package is missing and needs to be installed.

Resolution

Install kernel headers

1. To determine if the correct `kernel-default-devel` package for your kernel version is available for install, run the command:

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

   The output includes the syntax:

   ```
   <status> | kernel-default-devel | <package_type> | <version> | <arch> | <repository>
   ```

   If you see a similar output, then take note of the `version` specified in the output and continue to step 2. Otherwise, skip this section and instead use the steps to update the kernel and install matching kernel headers.
2. Install the `kernel-default-devel` package using this command, replacing `<version>` with the available version identified in the previous step:

   ```
   sudo zypper install --oldpackage kernel-default-devel=<version>
   ```

3. Verify the `kernel-default-devel` package was installed. Fault 11 is cleared by the connector after about a minute.

Update Kernel and Install Matching Kernel Headers

If the required `kernel-default-devel` package is not available, update your kernel to a supported version and install the matching kernel headers.

1. List the available `kernel-default` and `kernel-default-devel` packages with the command:

   ```
   zypper search -s --match-exact kernel-default kernel-default-devel
   ```

   Look for matching versions of the `kernel-default` and the `kernel-default-devel` packages. For example:

   ```
   <status> | kernel-default       | package | <new-version> | <arch> | <repository>
   <status> | kernel-default-devel | package | <new-version> | <arch> | <repository>
   ```

   Take note of the `new-version` specified in the output.
   Note: if you cannot find a newer kernel to upgrade to, you must upgrade to a newer distribution release version.
2. Install the new versions of `kernel-default` and `kernel-default-devel` with the command:

   ```
   sudo zypper install kernel-default=<new-version> kernel-default-devel=<new-version>
   ```

3. Reboot your system:

   ```
   sudo reboot
   ```

4. Reverify. Fault 11 is cleared by the connector.

Debian-based Distributions

Verify

Debian-based distributions require that a `linux-headers` package is installed for the currently running kernel. To verify if a `linux-headers` package is installed for the current running kernel, run:

```
apt list linux-headers-$(uname -r)
```

If there are no results, then the required `linux-headers` package is missing and needs to be installed.

Resolution

To install the required `linux-headers` package, run the command:

```
sudo apt install linux-headers-$(uname -r)
```

Reverify. Fault 11 is cleared by the connector after about a minute.