

Report Problems of Memory Utilization via EEM

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Symptoms of IOS XE Routers Running out of Memory](#)

[Information TAC Needs for Initial Triage](#)

[Understanding High Memory Usage](#)

[EEM to Monitor Memory Utilization](#)

[Triggers](#)

[Core File](#)

Introduction

This document describes general troubleshooting tips for collecting additional information for a memory leak problem.

Prerequisites

Requirements

Cisco recommends that you have basic knowledge of these topics:

- Basic knowledge of Cisco IOS® XE
- Basic knowledge in Embedded Event Manager (EEM)

Components Used

This document is not restricted to specific software and hardware versions. It applies for any routing Cisco IOS XE platforms like ASR1000, ISR4000, ISR1000, Cat8000 or Cat8000v.

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

In this document, you can find common logs that the device generates in case of a high memory utilization.

Also, you can see how you can benefit from the Embedded Event Manager feature to help TAC to monitor and get data on situations where the IOS XE router is frequently running out of memory.

The purpose of this document is not to explain any troubleshooting procedures extensively, if available, but

only references to more in depth troubleshooting guides are provided.

Symptoms of IOS XE Routers Running out of Memory

When dealing with high memory usage problems, typically, you see a log message indicating that the warning limit of 85% has been reached. This value vary depending the version. Different logs are generated depending on where the system found the problem:

TCAM problems:

CPP_FM-3-CPP_FM_TCAM_WARNING

IOSd (Control plane):

SYS-2-MALLOCFAIL

SYS-2-CHUNKEXPANDFAIL

SYS-4-CHUNKSIBLINGSEXCEED

QFP (Data plane):

QFPOOR-4-LOWRSRC_PERCENT_WARN

QFPOOR-4-TOP_EXMEM_USER

CPPEXMEM-3-NOMEM

CPPEXMEM-3-TOPUSER

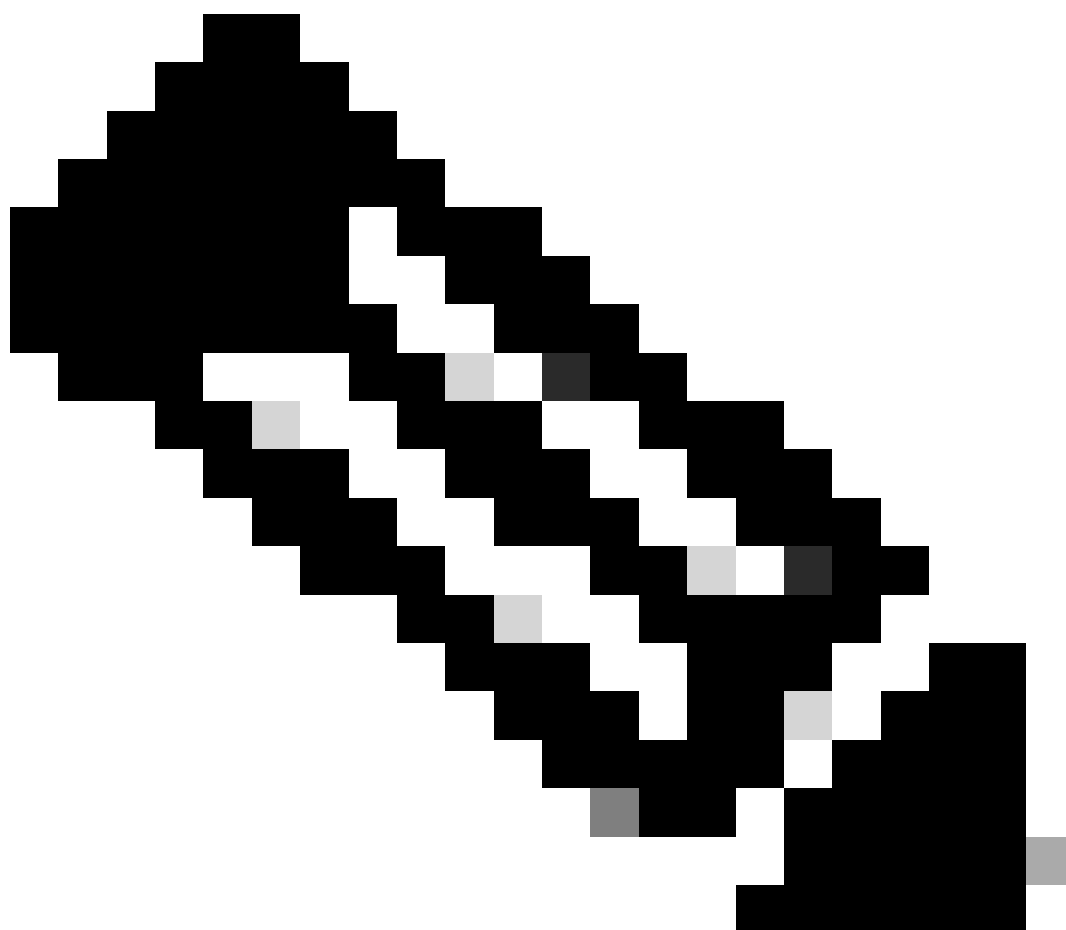
Temporal file system (TMPFS):

PLATFORM-3- ELEMENT_TMPFS_WARNING

General system log (need isolation):

PLATFORM-4-ELEMENT_WARNING

PLATFORM-3-ELEMENT_CRITICAL



Note: Log improvements are available from version 16.12 and later.

Information TAC Needs for Initial Triage

- Commands output:

show clock

show version

show platform resources

show platform software status control-processor brief

show process memory sorted

show memory statistics

show memory allocating-process totals

show process memory platform sorted

show logging

- In case of an unexpected reload due to a low memory condition:

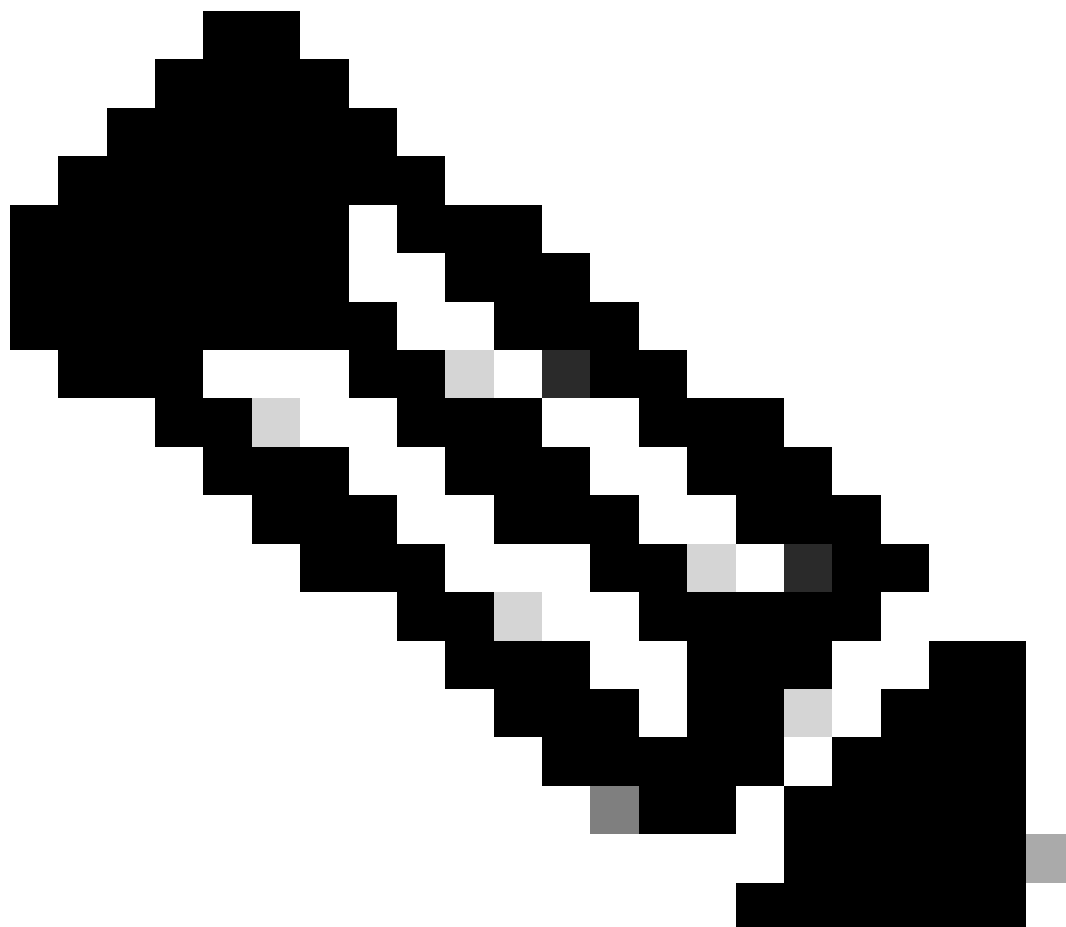
core file/system report

- Graph of memory utilization over time.

Attaching a show tech is desirable, it is helpful for TAC, and you can benefit from the automation TAC has developed to help you find issues faster.

Conditions leading to a high memory utilization are always software related. However, not all instances of high memory usage are unexpected. It is important to consider the available DRAM and the mix of features running on the device.

Troubleshooting high memory utilization is smoother, more effective, and with a better TAC interaction if you use Radkit. This tool, developed by Cisco, provides TAC a highly secure and easy way to access to the devices you select in your network. For more information, visit: [Cisco RADKit](#)



Note: Make sure you are running a supported version. Look for the End-of-Sale and End-of-Life document for the release. If needed, move to a version that is currently under Software Maintenance Releases. Otherwise, TAC can be limited on the troubleshooting and resolution options.

For a complete document around memory troubleshooting you can refer to these guides:

On ISR4K: [Memory Troubleshooting Guide for Cisco 4000 Series ISRs.](#)

On ASR1K: [ASR 1000 Series Router Memory Troubleshoot Guide.](#)

Understanding High Memory Usage

In Cisco IOS XE routers, DRAM is one of the most important resources that supports core functionality. DRAM is employed to store different data types and processes/features information that are essential for both the control plane and data plane operations.

Main uses of DRAM in IOS XE routers include:

IOSd Memory (Control Plane Structures): Stores processes related information related to control plane of the device, such as: routing information/protocols, network management structures, system configurations and feature information.

QFP Memory (Data Plane Structures): Stores everything around QFP operations handled by the microcode, such as key structures of features stored in the QFP, microcode instructions and forwarding instructions.

Temporary File System (TMPFS): Mounted in DRAM and managed by IOSd, TMPFS serves as a quick-access storage area for files needed by the processes. In case those files are persistent, they are moved to a harddisk/bootflash. It enhances system performance by reducing read/write times for temporary data.

General Processes Running on the Linux Kernel: Since IOS XE operates on a Linux-based kernel, DRAM also supports various system processes that run on top of this kernel.

High memory utilization of greater than 85% typically indicates significant DRAM consumption, which can impact router performance. This elevated usage can be a result from legitimate demands, such as storing extensive routing tables, or enabling resource-intensive features. However, it can also signal issues like inefficient memory management by certain features or memory leaks, where memory is not properly released back to the system after use.

By monitoring memory utilization across IOSd memory, QFP memory, TMPFS, and general Linux processes, you and TAC can identify potential problems early.

EEM to Monitor Memory Utilization

For memory troubleshooting, TAC needs to collect a set of commands over a period of time to identify the offending process. Sometimes, after the culprit process is identified, additional specific commands are needed, making memory troubleshooting one of the most time-consuming types of troubleshooting.

In order to make this troubleshooting easier, you can use the EEM feature to monitor and automatically collect information. There are two main considerations for writing the EEM script: trigger and commands to be collected.

Triggers

Pattern. You can use the pattern of section Symptoms of Cisco IOS XE routers running out of memory. The format looks like this:

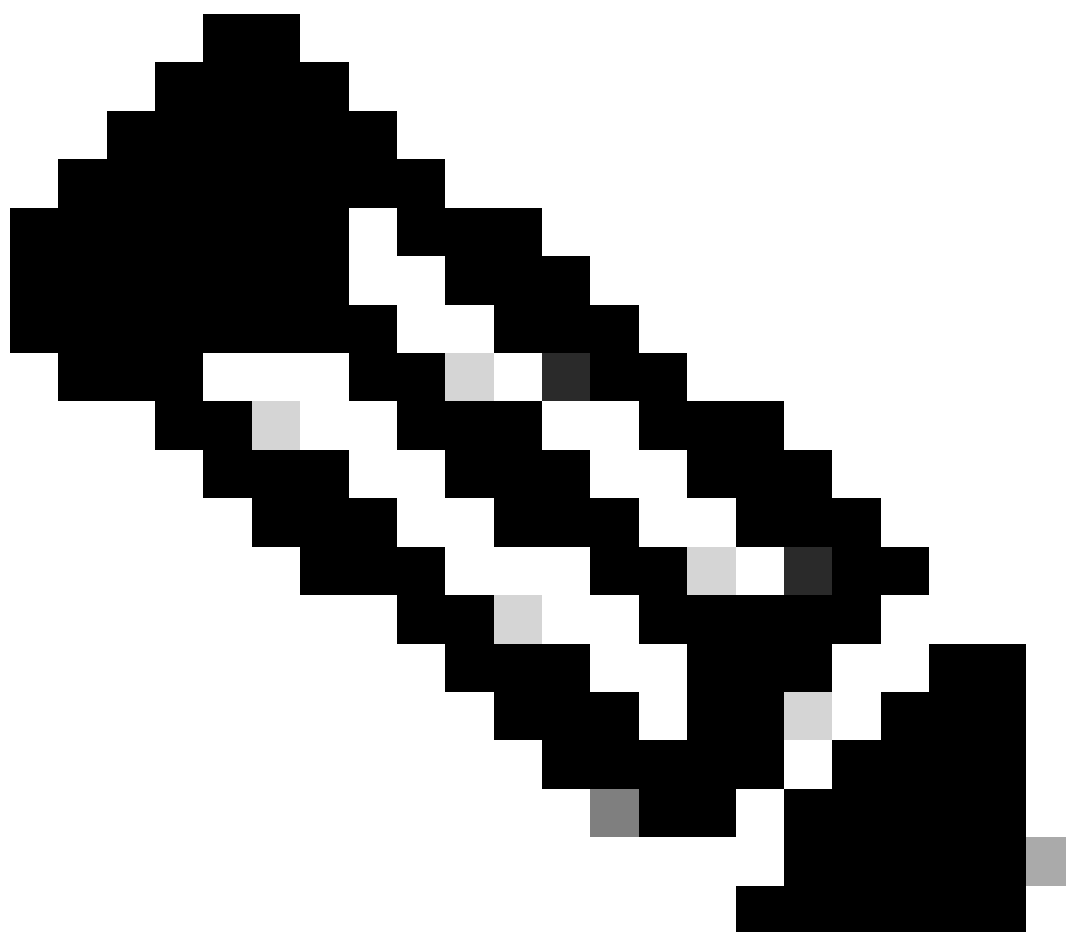
```
event syslog pattern <pattern> ratelimit 300 maxrun 180
```

One of the considerations while using a pattern as a trigger, is that the log is generated once the warning threshold is reached, depending on the memory consumption rate, trying to do it manually, you or TAC do not have enough time for a more detailed troubleshooting.

Cron timer. Example of a cron timer to be activated every 30 minutes:

```
event timer cron name HalfHour cron-entry "*\30 * * * *"
```

One of the advantages of a cron timer over a pattern is that you do not need to wait until the device almost run out of memory resources to collect information. Depending on the memory consumption rate, with proper monitoring and information, TAC can identify the offended process before reaching warning threshold.



Note: Ratelimit and maxrun options are used to guarantee that the entire set of outputs are collected. They also help to avoid additional noise or EEM activation in situations where multiple logs appear in a short period of time.

EEM examples with general commands for initial triage:

```
configure terminal
event manager applet TAC_EEM authorization bypass
event syslog pattern " PLATFORM-4-ELEMENT_WARNING" ratelimit 300 maxrun 180
action 0.1 cli command "enable"
action 0.2 cli command "term exec prompt timestamp"
action 0.3 cli command "term length 0"
action 0.4 cli command "show process memory platform sorted | append bootflash:TAC_EEM.txt"
action 0.5 cli command "show processes memory platform sorted location chassis 1 R0 | append bootflash:TAC_EEM.txt"
action 0.9 cli command "show platform resources | append bootflash:TAC_EEM.txt"
action 1.0 cli command "show platform software status control-processor brief | append bootflash:TAC_EEM.txt"
action 1.1 cli command "show clock | append bootflash:TAC_EEM.txt"
action 1.3 cli command "show platform software process memory chassis active r0 all sorted | append bootflash:TAC_EEM.txt"
action 1.5 cli command "show process memory platform accounting | append bootflash:TAC_EEM.txt"
```

Monitor daily with a cron timer:

```
configure terminal
event manager applet TAC_EEM2 authorization bypass
event timer cron name DAYLY cron-entry "0 0 * * *"
action 0.1 cli command "enable"
action 0.2 cli command "term exec prompt timestamp"
action 0.3 cli command "term length 0"
action 0.4 cli command "show process memory platform sorted | append bootflash:TAC_EEM2.txt"
action 0.5 cli command "show processes memory platform sorted location chassis 1 R0 | append bootflash:TAC_EEM2.txt"
action 0.6 cli command "show processes memory platform sorted location chassis 2 R0 | append bootflash:TAC_EEM2.txt"
action 0.9 cli command "show platform resources | append bootflash:TAC_EEM2.txt"
action 1.0 cli command "show platform software status control-processor brief | append bootflash:TAC_EEM2.txt"
action 1.1 cli command "show log | append bootflash:TAC_EEM2.txt"
action 1.2 cli command "show clock | append bootflash:TAC_EEM2.txt"
action 1.3 cli command "show platform software process memory chassis active r0 all sorted | append bootflash:TAC_EEM2.txt"
action 1.5 cli command "show process memory platform accounting | append bootflash:TAC_EEM2.txt"
```

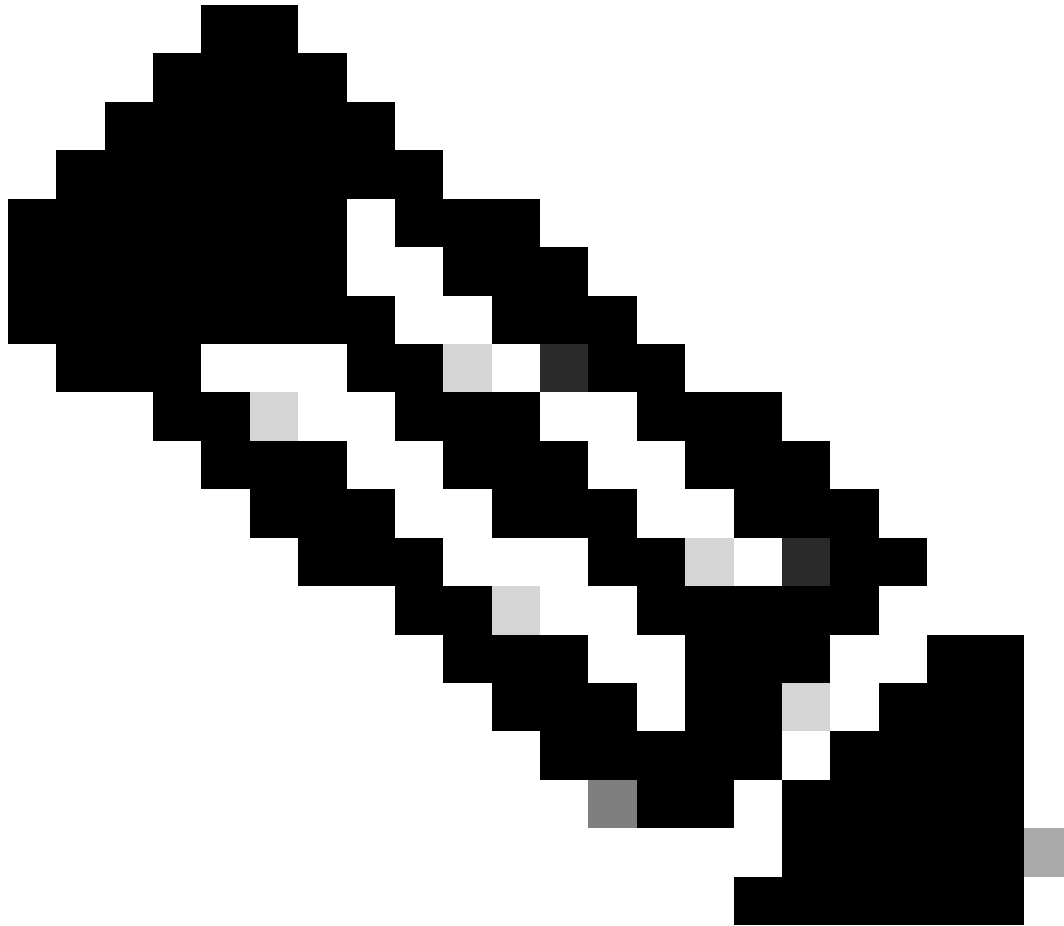
For a more comprehensive list of commands, please refer to the guides from the section for Information TAC needs for initial triage.

Core File

When the memory utilization reaches a critical level, chances are that the operating system forces a crash in order to recover from this condition, generating a system report which that contains a core file.

The core file is the full dump of memory for a particular process that crashed at certain point in time. This core file is critical for TAC to inspect memory and analyze source code to understand the conditions and potential reasons of the unexpected reload/crash of the process.

The core file helps TAC and developers to find the root cause of the problem, debug, and fix the issue.



Note: Even though TAC and developers strive for getting a root cause, there are times where the crash was a consequence of a network event, or a timing issue which makes it virtually impossible to reproduce it in the lab.

For more information about unexpected reloads and how to retrieve a core file refer to [Troubleshoot Unexpected Reloads in Cisco IOS® Platforms with TAC](#).