Analyze macOS AMP Diagnostic Bundle for High CPU

Contents

Introduction

This document describes the steps to analyze a diagnostic bundle from Advanced Malware Protection (AMP) for Endpoints Public Cloud on macOS devices to troubleshoot high CPU usage.

Contributed by Uriel Torres and Edited by Yeraldin Sanchez, Cisco TAC Engineers.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic navigation in the AMP Console
- Navigation of the MAC Terminal

Components Used

The information in this document is based on these software and hardware versions:

- AMP for Endpoints Console 5.4.20200512

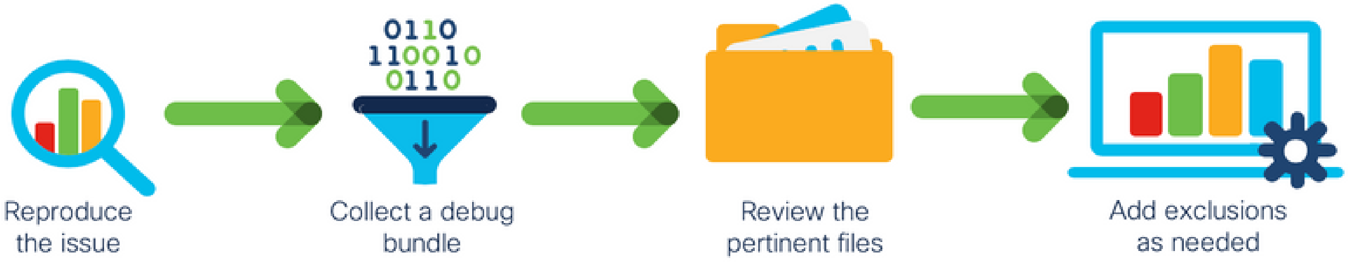
- macOS Catalina version 10.15.4
- AMP Connector 1.12.3.738

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

The AMP Connector scans all active files (those which move, copy and/or modify themselves) on a machine unless explicitly told not to, that inevitably brings performances issues if too many processes and operations run while the Connector is running, which leads to high CPU utilization, slowdowns and in some cases software that will not run or run slowly. In addition, the AMP Connector may block files based on their cloud reputation, which can some times be erroneous (false positive). The solution to both issues is to exclude these paths and processes.

The flow of troubleshooting performance issues is shown in the image.



Troubleshoot

This section provides the information you can use to troubleshoot your configuration.

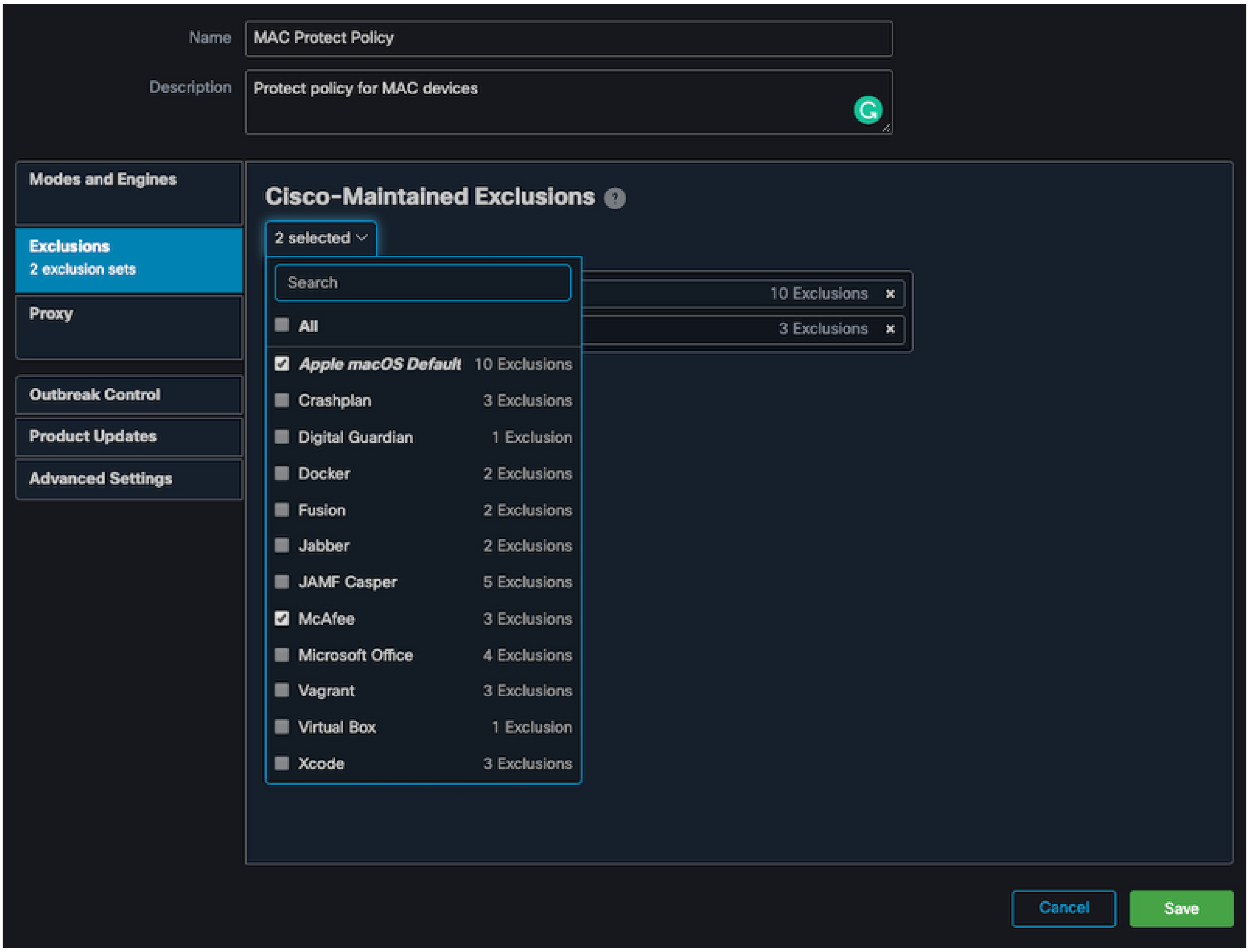
Verify if another antivirus is installed on the machine

> **Tip**: Use the Cisco-Maintained exclusions if the software that is used is included on the list, remember that these exclusions can be added to new versions of an application.

In order to see the lists available in Cisco maintained exclusions section on the AMP console:

- Navigate to **Management > Policies**.
- Find the policy and click on **Edit**.
- On the policy, settings window click on **Exclusions**.

Select the ones your endpoint would need according to the software currently installed on the machine, then, save the policy, as shown in the image.



Name: MAC Protect Policy

Description: Protect policy for MAC devices

Cisco-Maintained Exclusions

2 selected

□	All	
☑	*Apple macOS Default*	10 Exclusions
□	Crashplan	3 Exclusions
□	Digital Guardian	1 Exclusion
□	Docker	2 Exclusions
□	Fusion	2 Exclusions
□	Jabber	2 Exclusions
□	JAMF Casper	5 Exclusions
☑	McAfee	3 Exclusions
□	Microsoft Office	4 Exclusions
□	Vagrant	3 Exclusions
□	Virtual Box	1 Exclusion
□	Xcode	3 Exclusions

Identify the high CPU when a specific application is in use

Identify if the issue happens while one application or a few of them are executed if you are able to replicate the issue helps in the process to identify potential exclusions.

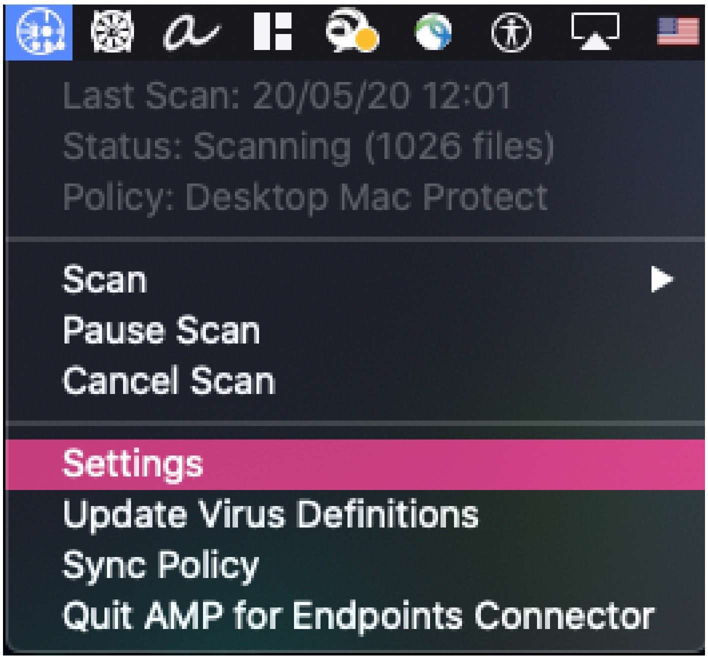
Gatter a diagnostic bundle for analysis

In order to gather a useful diagnostic bundle, the debug log level must be enabled.

Debug Level in the Endpoint

If you can replicate the issue and have access to the endpoint, below is the best procedure to capture the diagnostic bundle.

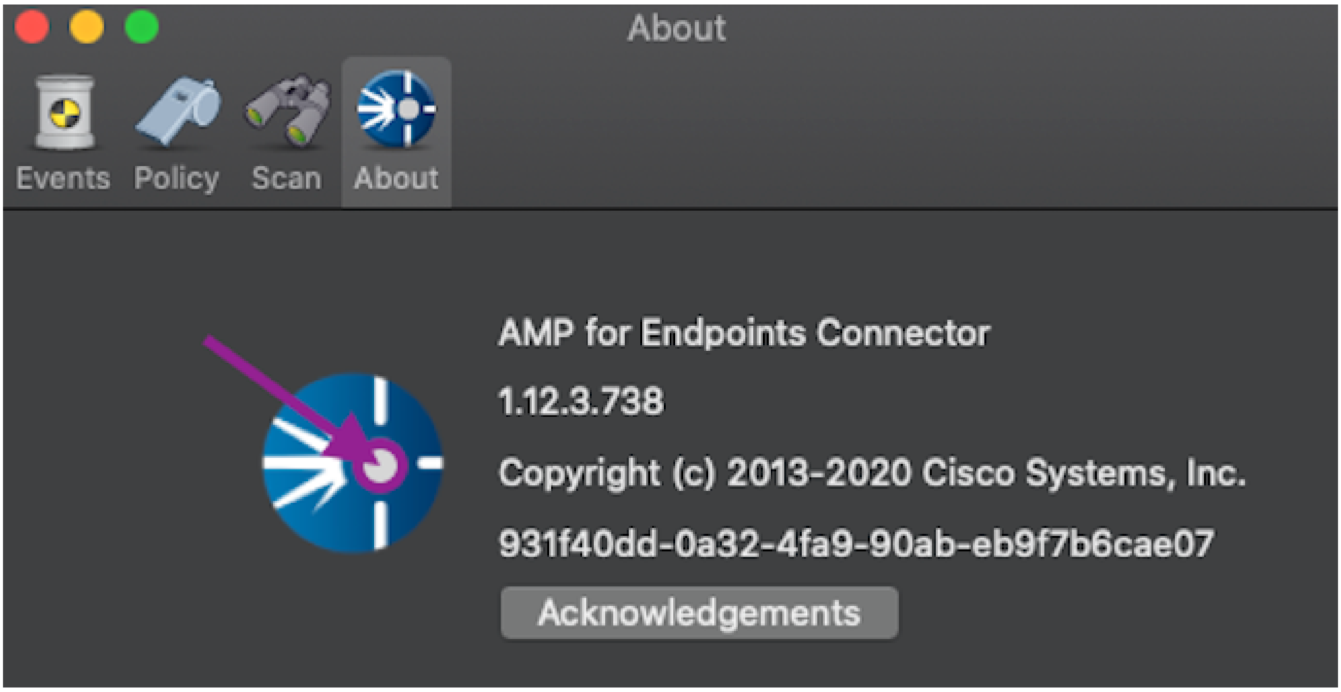
- On the MAC Menu bar click on the AMP icon.
- Navigate to **Settings** section, as shown in the image.



Last Scan: 20/05/20 12:01
Status: Scanning (1026 files)
Policy: Desktop Mac Protect

- On the settings windows, navigate to **About**.

- In order to enable debug mode click inside the AMP logo, as shown in the image.



AMP for Endpoints Connector

1.12.3.738

Copyright (c) 2013-2020 Cisco Systems, Inc.

931f40dd-0a32-4fa9-90ab-eb9f7b6cae07

A popup indicates the AMP Connector is on Debug Mode

This procedure enables debug log level until the next policy heartbeat interval.

Debug Level in the AMP Command Line Interface (CLI)

- Open a Terminal
- Navigate to **/opt/cisco/amp/bin/**
- Run ampcli:
```
./ampcli
```
- On the AMP CLI enable debug mode:
```
ampcli>debuglevel 1
```

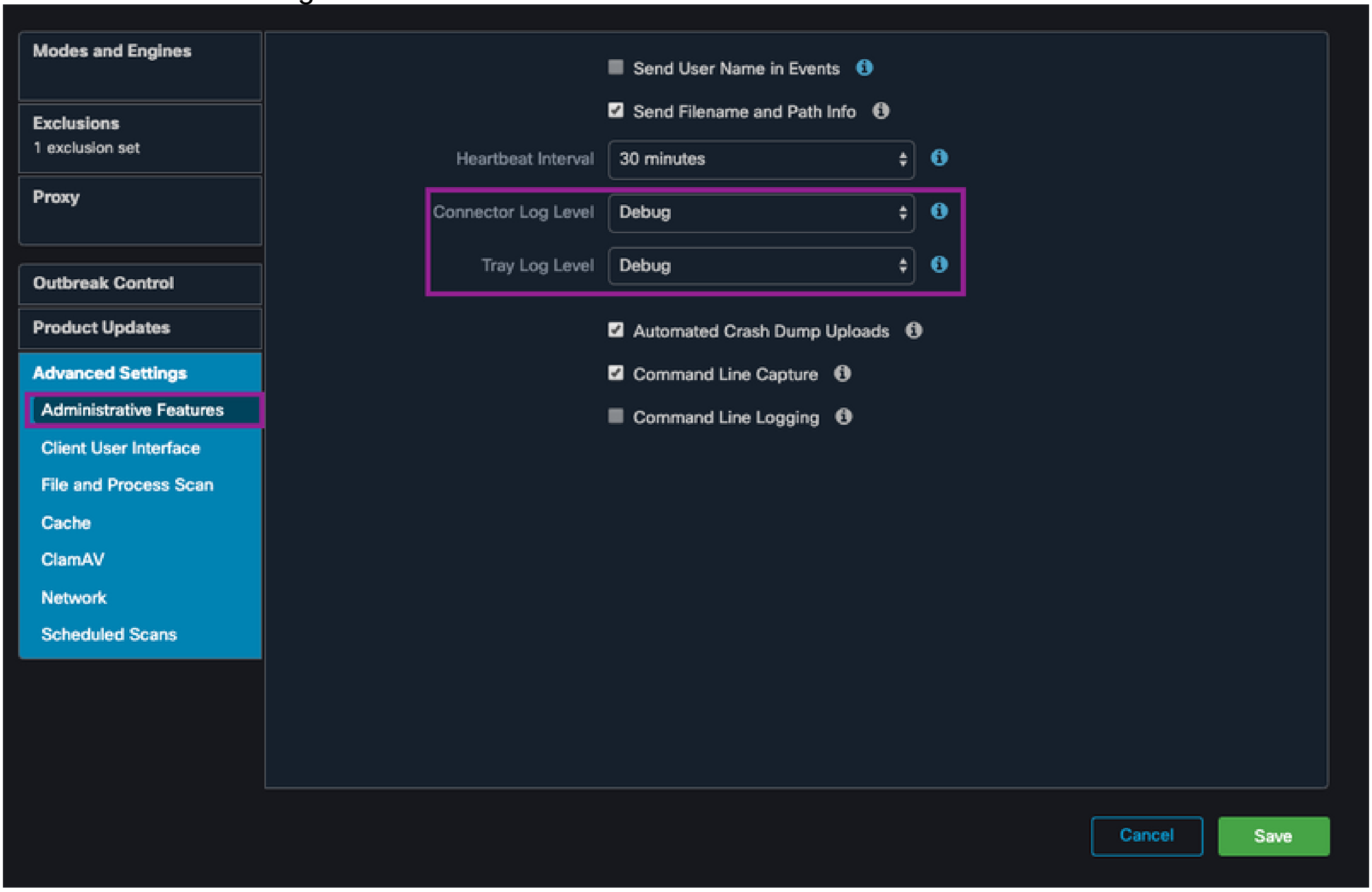
This process enables debug log level until the next policy heartbeat interval.

Debug Level in the Policy

If you don´t have access to the endpoint or the issue can´t be reproduced consistently, the debug log level must be enabled in the policy.

In order to enable the debug log level by the policy:

- Navigate to **Management > Policies**
- Find the policy and click on **Edit**
- Navigate to **Advanced Settings > Administrative Features**
- Configure **Connector Log Level** and T**ray Log Level** to Debug and save the policy, as shown in the image



□ Send User Name in Events
☑ Send Filename and Path Info
Heartbeat Interval: 30 minutes
Connector Log Level: Debug
Tray Log Level: Debug
☑ Automated Crash Dump Uploads
☑ Command Line Capture
□ Command Line Logging

Caution: If debug mode is enabled from the policy, all endpoints receive this configuration.

Note: Sync the policy of the endpoint to ensure the debug mode.

Exclude AMP from other antivirus solutions

According to the user guide, antivirus products must exclude the next directories and any files, directories and executable files within them to be compatible with the AMP Connector for MAC, the directories to exclude are the followings:

- **/Library/Application Support/Cisco/AMP for Endpoints Connector**

- **/opt/cisco/amp**

Reproduce the issue and gather a diagnostic bundle

When the debug level is configured, wait until the state of High CPU happens on the system or manually reproduce the conditions previously identified and then gather the diagnostic bundle.

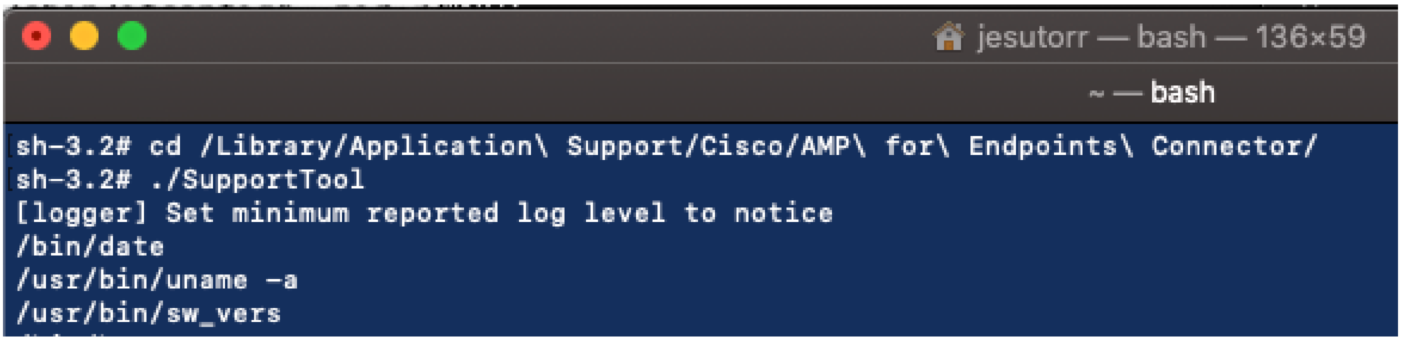
In order to collect the debug bundle:

- Open a Terminal.
- Access to superuser level, then navigate to **/Library/Application Support/Cisco/AMP for Endpoints Connector**:

```
cd /Library/Application\ Support/Cisco/AMP\ for\ Endpoints\ Connector/
```

- In order to run the Support Tool use the next command:

```
./SupportTool
```



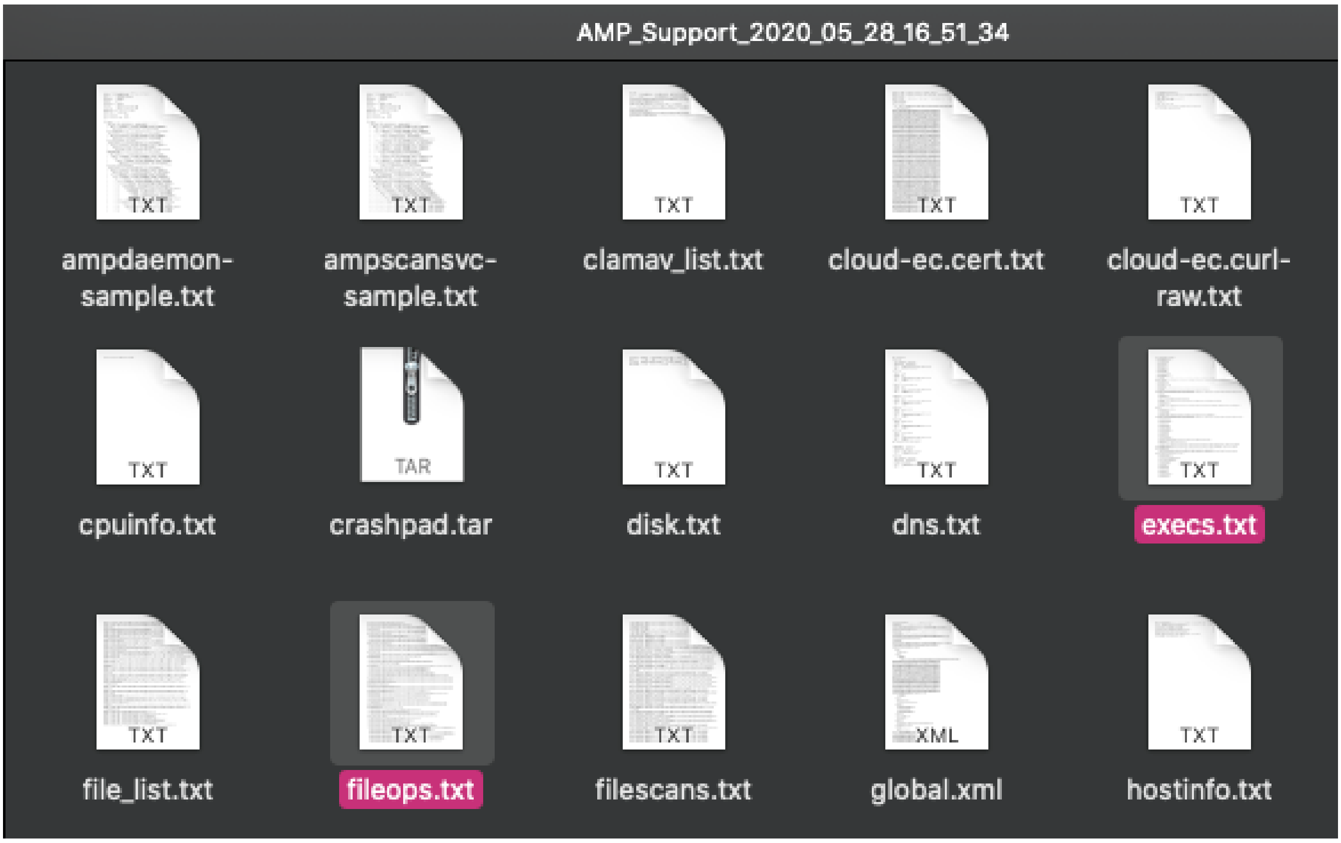
```
sh-3.2# cd /Library/Application\ Support/Cisco/AMP\ for\ Endpoints\ Connector/
sh-3.2# ./SupportTool
[logger] Set minimum reported log level to notice
/bin/date
/usr/bin/uname -a
/usr/bin/sw_vers
```

The debug bundle is saved in the Desktop folder as a .zip file extension.

Analysis of high CPU performance

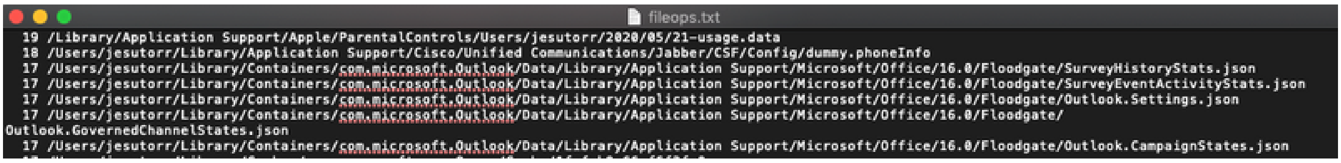
The debug diagnostic bundle is storage in the Desktop, to start the analysis:

- Decompress the Diagnostic Bundle
- There are 2 files to review File Operations: fileops.txtFile Executions: execs.txt



- The fileops.txt works as the main performance tool to troubleshoot. It lists al current al currently active operations on your endpoint while the Connector runs, It is read as follows:

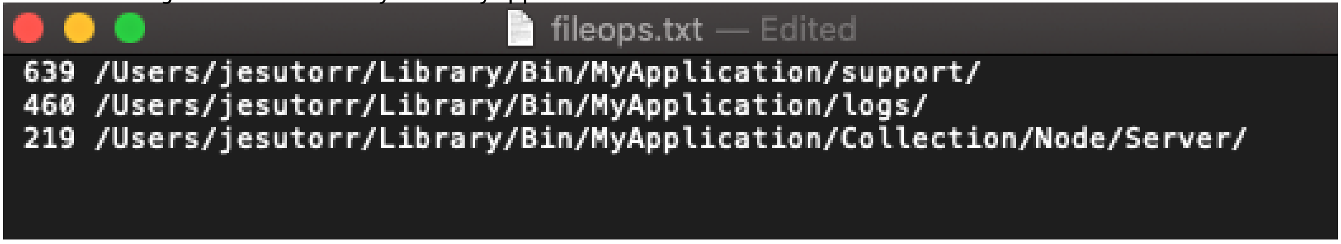
<Number scans performed on the path when the bundle is collected> / **<Path scanned>**



```
  19 /Library/Application Support/Apple/ParentalControls/Users/jesutorr/2020/05/21-usage.data
  18 /Users/jesutorr/Library/Application Support/Cisco/Unified Communications/Jabber/CSF/Config/dummy.phoneInfo
  17 /Users/jesutorr/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Floodgate/SurveyHistoryStats.json
  17 /Users/jesutorr/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Floodgate/SurveyEventActivityStats.json
  17 /Users/jesutorr/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Floodgate/Outlook.Settings.json
  17 /Users/jesutorr/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Floodgate/
Outlook.GovernedChannelStates.json
  17 /Users/jesutorr/Library/Containers/com.microsoft.Outlook/Data/Library/Application Support/Microsoft/Office/16.0/Floodgate/Outlook.CampaignStates.json
```

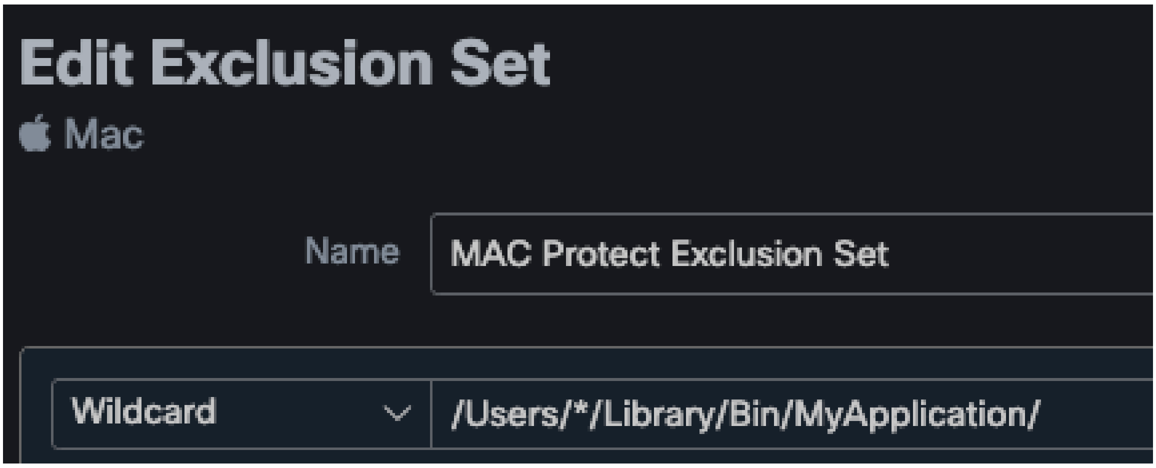
For example, If you have a homebrew application, fileops.txt shows the next active operations:

```
639 /Users/jesutorr/Library/Bin/MyApplication/support/

460 /Users/jesutorr/Library/Bin/MyApplication/logs/

219 /Users/jesutorr/Library/Bin/MyApplication/Collection/Node/Server/
```



```
639 /Users/jesutorr/Library/Bin/MyApplication/support/
460 /Users/jesutorr/Library/Bin/MyApplication/logs/
219 /Users/jesutorr/Library/Bin/MyApplication/Collection/Node/Server/
```

- Once the process has been identified an exclusion can be created
- In order to create the exclusion
- On the AMP Console, navigate to **Management > Exclusions**
- Select the exclusion set and click on **Edit**
- The exclusion can be added as shown in the image



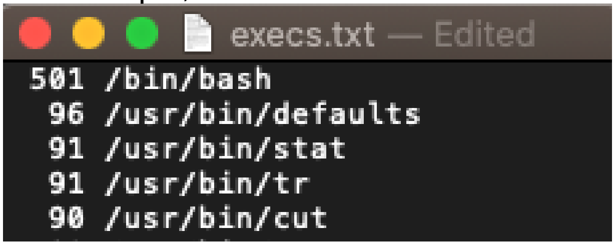
Edit Exclusion Set

Mac

Name: MAC Protect Exclusion Set

Wildcard: /Users/*/Library/Bin/MyApplication/

- The Execs.txt file contains all commands used by processes that run while the Connector collects bundles. The paths listed here must not be excluded on the AMP policy, as these are binaries (/bin) and system binaries(/sbin) that all processes utilize, however, on the Execs.text can provide the main process that is running.
  For example, if the Execs.txt file shows the next logs.

  ```
   501 /bin/bash
    96 /usr/bin/defaults
    91 /usr/bin/stat
    91 /usr/bin/tr
    90 /usr/bin/cut
  ```

  Since the homebrew application uses bash you can confirm that the application is the cause of the high CPU.

Related Information

- [AMP for Endpoints: Process Exclusions in macOS and Linux]
- [Best Practices for AMP for Endpoints Exclusions]
- [Technical Support & Documentation - Cisco Systems]