Basic Troubleshoot Guide for AMP for Endpoints Linux Connector

Contents

Introduction

This document describes a basic way to troubleshoot performance issues on the Cisco Advanced Malware Protection (AMP) for Endpoints Linux Connector.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- AMP for Endpoints
- Linux/Unix-based Operating Systems

Components Used

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

- Red Hat Enterprise Linux (RHEL) / Community Enterprise Operating System (CentOS) versions 6.10 and 7.7
- AMP For Endpoints Linux Connector version 1.11.1

For a full list of Compatible AMP versions with Linux Operating System, refer to this article.

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 active, 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; in the case of false positive, non-performance-related issues or performance issues that don't seem to be resolved via this guide, It is recommended to raise ticket support.

The flow of troubleshooting basic performance issues is as follows:

- Collect a Debug bundle while the issue is reproduced.
- Run the AMP support tool
- Review the pertinent files
- Add exclusions as needed

Troubleshoot

How to collect a debug bundle

A debug bundle is a zip file that contains detailed debug information (like scan logs) on the connector. This bundle is essential to troubleshoot most issues related to the AMP for Endpoints connector. To collect a debug bundle, follow the steps provided on Collection of Diagnostic Data from AMP for Endpoints Linux Connector.

What information does the amp support tool collect then a Debug bundle is run?

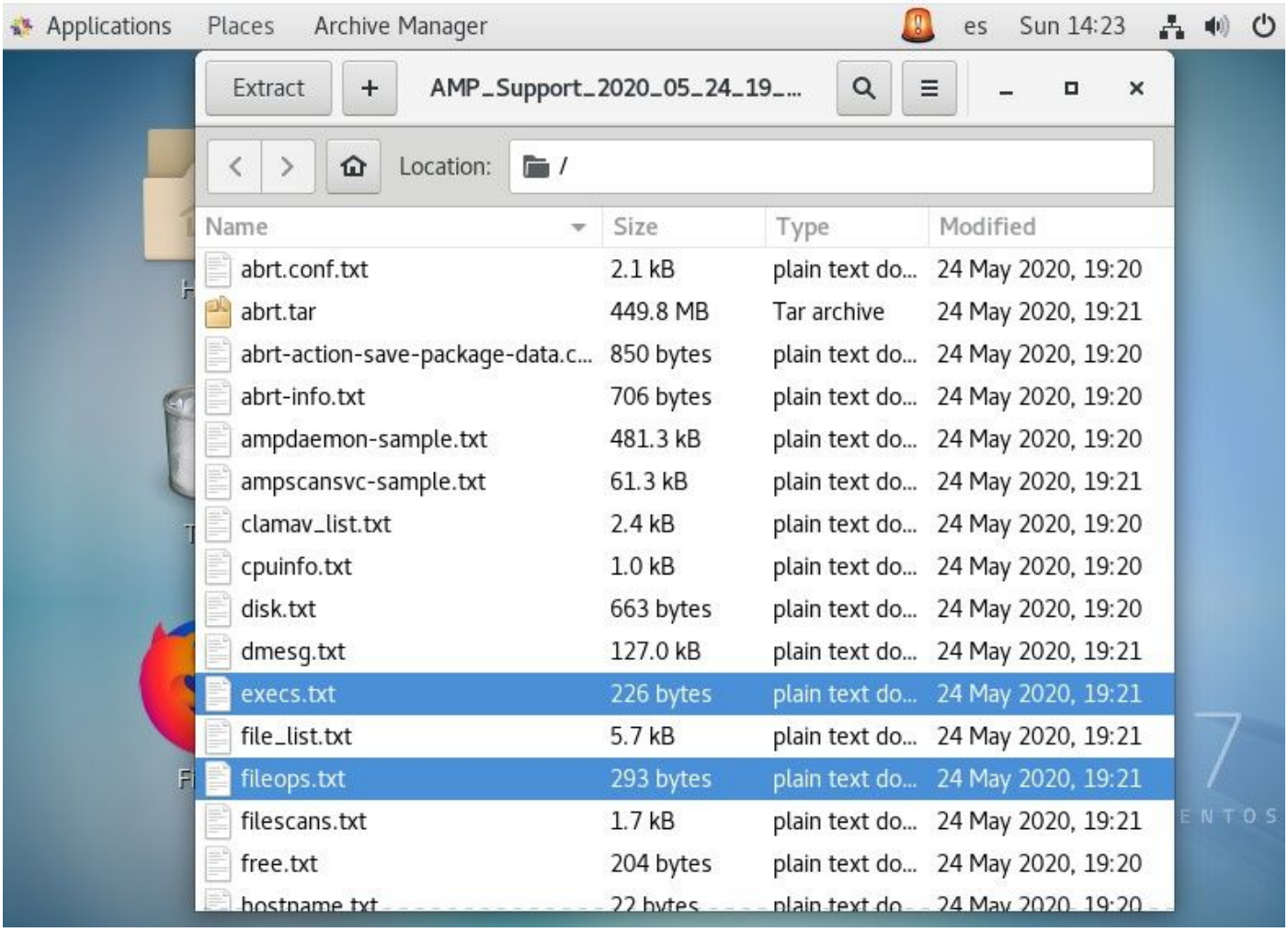
The debug bundle process input shows that the *ampsupport runs some log-collection commands, as shown in the image.*

```
top -b -n5 -d2 -H -p `pidof ampdaemon | tr ' ' ,` -p `pidof ampscansvc | tr ' ' ,`
[ -e 'abrt-cli' ] && abrt-cli list -d
[ -d '/var/spool/abrt' ] && for dir in $(find /var/spool/abrt/*/ -type d -maxdepth 1);
do echo -e "
Crash: ${dir}"; echo -e "
Kernel: $(cat "${dir}/kernel")"; echo -e "
Count: $(cat "${dir}/count")";echo -e "
Executable: $(cat "${dir}/executable")"; echo -e "
Uid: $(cat "${dir}/uid")";echo -e "
Reason: $(cat "${dir}/reason")"; echo -e "
Package: $(cat "${dir}/package")"; done
find: warning: you have specified the -maxdepth option after a non-option argument -typ
e, but options are not positional (-maxdepth affects tests specified before it as well
as those specified after it).  Please specify options before other arguments.

cat: /var/spool/abrt/oops-2020-05-18-18:21:09-10472-0//executable: No such file or dire
ctory
[ -e '/etc/abrt/abrt.conf' ] && cat '/etc/abrt/abrt.conf'
[ -e '/etc/abrt/abrt-action-save-package-data.conf' ] && cat '/etc/abrt/abrt-action-sav
e-package-data.conf'
cat /proc/slabinfo
```

How to read basic Linux bundle logs to identify the affected paths and processes

The Linux AMP for Endpoints Debug bundle carries a plethora of useful information, however, for basic performance troubleshooting purposes, there are only a few files to review, fileops.txt, fiescans.txt, and execs.txt, as shown in the image.



The File Operations (fileops) text file works as the main performance troubleshooting tool. it lists al current al currently active operations on your endpoint while the connector runs. These are the paths to add to the policy exclusion set if deemed necessary/safe.

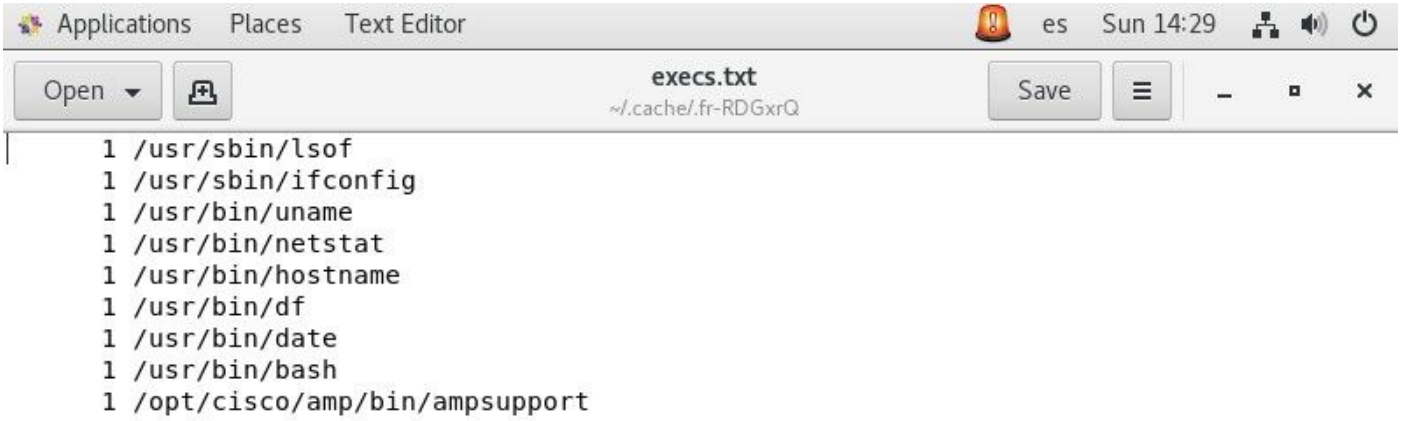
It is read as follows:

- <Number scans performed on the path performed while bundle collection process runs> /<Path scanned>

Scans example:

- 1 /homet/user/.mozila/Firefox/

The File Scans (filescan) Text file lists all processes that run while the connector collected debug information.

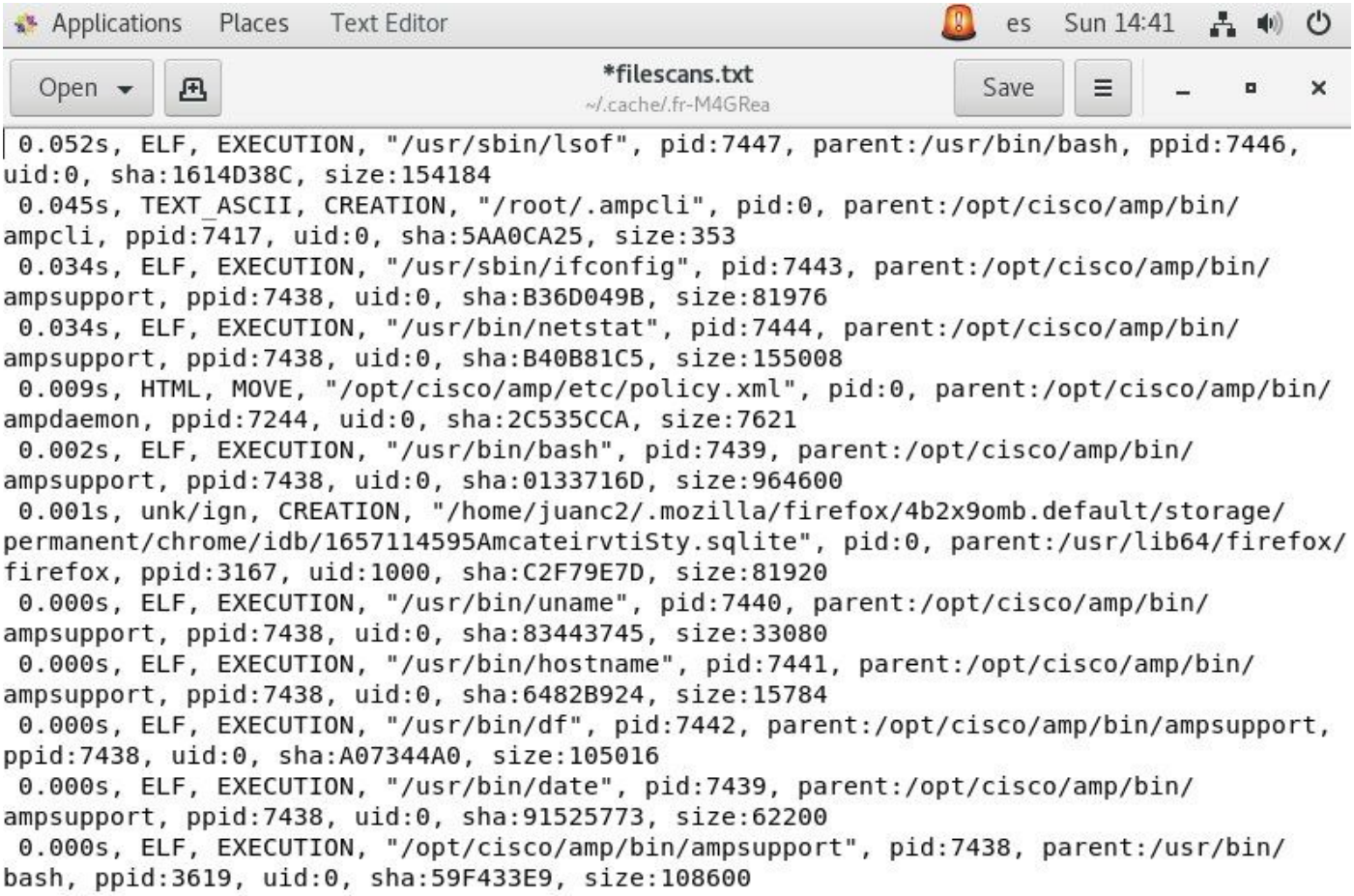


It reads as such:

- <Execution time> , <File Type>, <Operation type>, <Process path>, <Parent process path> , <Process ID>, <Parent PProcess ID> , <SHA signature (Not SHA256)> <File Sze>

The File Execution (execs) text file lists all Linux commands used by active processes on the connector while the connector collected the bundle.

> **Warning**: The paths listed here must not be excluded on the AMP policy, as these are binaries (/bin) and system binaries(/sbin) that all process utilizes, however, this list might come useful to trying to understand which actions are performed by the different processes that run on the target machine.



Once identified, Path is to be excluded via policy, please follow Best Practices for AMP for Endpoint Exclusions.

Process exclusions handled by the Mac and Linux connectors are similarly added via policy, however, the method differs slightly: Process Exclusions in macOS and Linux.

Once exclusions are added, test, and monitor if the problem persists. Contact AMP TAC Support.