Camav Documentation

Release 0.1

September 11, 2012

Contents

1 Introduction

This doc is meant to explain how to the CAMAV (multi antivirus engine scanner) is working. This is a draft document. This is a living document and will be updated with each stage of the project.

1.1 Requirements

CAMAV makes use of the following components:

- Apache+PHP for the web interface.
- MySQL to store information about: samples, users, queue, submissions, access logs, analysis results…
- Python.
- VirtualBox to execute and isolate the malware.
- Bash script for managing the different antivirus engines.

1.2 Overview of the project

The system functions as follows. The system has two main interfaces.
A web interface where the user can interact with the system and a web service were the system could be integrated with other products. At this stage the first (web page) interface is developed only for test purposes.

The second interface - web service; provides a requester with the result of a submitted file analyzed by the internal antiviruses.
On both cases the file is submitted through a POST request to the web server.

The file is uploaded on the system and then renamed as md5_actual_timestamp and moved to under the update folder in CAMAV apache home folder (/var/www/camav/upload)

Different information about the file is extracted and inserter into the database.

At this time apache is inserting the file into the queue (mysql queue).

A python script is waiting for the elements in the queue. When an element is new in the queue is stars the main script (upanddown.sh script) that manages the scanning of the file.

The upanddown script, launch each machine, run the antivirus program that scans the submitted file, collect the scanning logs files, and outputs a xml file with each antivirus engine and its results.
After the results xml file is done, the script launches another script that inserts all the results into the database.

The system has another component (future daemon) used to update the machines.

1.3 Use Case

CAMAV is used to scan files with multiple antivirus engines. It can be used as a web service. The user or the system uploads a file and receive back a JSON answer with the result of the scanning.

1.4 Architecture

CAMAV consists of central management software which handles the file scanning. Each scanning (for each antivirus) is launched in a fresh and isolated virtual machine (Virtual Box). Each antivirus has its own virtual machine.

The Host runs:

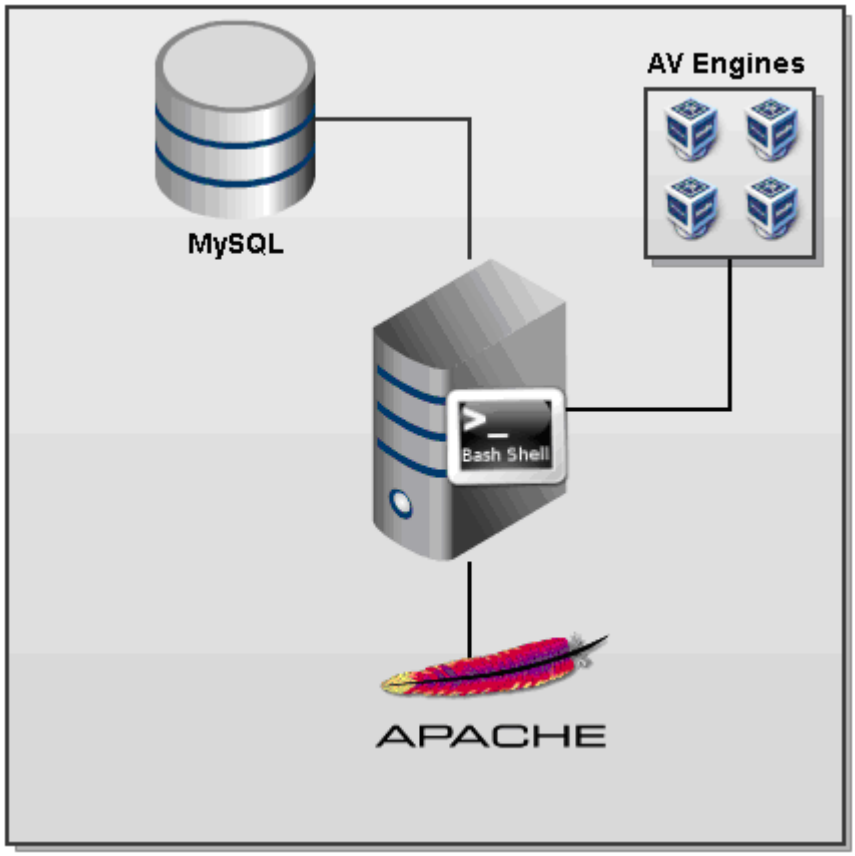
- Apache+PHP, for the web interface.
- MySQL for the database.
- Bash script for managing the scanning process

The Guests runs:

- Windows XP SP3
- An antivirus (different for each virtual machine) with its updating mechanism.

The Guest is an isolated environment where the malware is safely executed.

The following diagram resumes the architecture:



2 Installation

2.1 Preparing the Host

We are assuming that you're setting up CAVAM works on Ubuntu.

2.1.1 Python

Python version 2.7 preferred, but it can work with other versions (not tested).

2.1.1.1 Checking requisites

We need some extra modules:

sudo apt-get install libmagic-dev
sudo apt-get install python-magic
sudo apt-get install python-mysqldb
sudo apt-get install python-dev

2.1.2 Apache

Install apache2 using the Ubuntu package:

sudo apt-get install apache2

2.1.3 PHP

Install php using the Ubuntu packages:

sudo apt-get install php5

2.1.3.1 Requirements

These modules are needed for the Web interface:

sudo apt-get install libapache2-mod-php5
sudo apt-get install php5-curl

We also need the PHP Extension and Application Repository, PEAR:

sudo apt-get install php-pear

2.1.4 MySQL

Install MySQL and MySQL module for PHP using the Ubuntu package:

sudo apt-get install mysql-server
sudo apt-get install php5-mysql

2.1.5 VirtualBox

This system makes heavily use of VirtualBox and its SDK API. It has been tested with VirtualBox 4.0.x and 4.1.x and with both version works good. There are differences between the version 4.0 and version 4.1. The running versions runs on 4.1 and I recommend the usage of at least 4.1.6 or 4.1.8

NOTE: If you want to run the 4.0 you need to run a script version made for this versions

2.1.5.1 Install

Download VirtualBox and the Extension Pack (*Please install the extension pack with the same version as your installed version of VirtualBox)*.

Install the VirtualBox package and then the Extension Pack. VirtualBox package will install the vboxapi, so nothing else to do.

Add your user to the vboxusers group, in order to have access to USB devices:

sudo *adduser USERNAME vboxusers*

2.2 Setup the host machine

Create a the user camav

sudo adduser camav

As root you have to create the folder
/usr/local/camav

Copy the camav files (those meant for the usr/local/camav/) to the folder.

Change the owner of the folder to camav
chown camav:camav -R /usr/local/camav/

Copy the web files to /var/www/camav/

Create a database sandbox
Import the sandbox database structure (sandbox.sql)

2.3 The Host

The host system is made up of four scripts (as of today).
Two python scripts queue_put.py and queue_get.py used to manage the elements in the queue. The queue_get.py is running in the background as a daemon.
Two bash scripts used for manipulating the machines upanddown.sh and update_maV31.sh used to update machines. The last script has a configuration file camav.conf . This configuration file is not yet used in the upanddown.sh script, but in the next minor version will be integrated.
.

2.3.1 CAMAV.conf

```
#=====================================================================
#
#          FILE: camav.conf
#
#         USAGE: This is camav configuration file
#
#   DESCRIPTION: CAMAV - CART Multi Antivirus Engine
#                Manages Virtual Machines with Antiviruses for VirtualBox guest machine.
#                CAMAV configuration file
#
#       OPTIONS: ---
```

```
#                BUGS: ---
#               NOTES: ---
#              AUTHOR: Catalin Anton, catalin.anton@ec.europa.eu
#             COMPANY: European Commission, Luxembourg
#             VERSION: 1.0
#             CREATED: 06.10.2011 - 17:01:50
#          REVISION: ---
#             TODO: * clean up
#       CHANGE LOG: ---
#=====================================================================
#
# This is the main CAMAV server configuration file.  It contains the
#

### Section 1: Global Environment
#
# The directives in this section affect the overall operation of CAMAV,
# such as the number of concurrent requests it can handle or where it
# can find its configuration files.
#

#
# VBOXMANAGE: This is the path to the VirtualBox executable file
#
# NOTE!  This path should be set to "/usr/bin/VBoxManage"
#
#VBOXMANAGE="/usr/bin/VBoxManage"
VBOXMANAGE="/usr/bin/VBoxManage"

#
# INPUTFOLDER: This is the path to the folder where CAMAV is storing
# the reports files together with sample files. The sample files are
# also stored in the /var/www/camav/upload/
#
#INPUTFOLDER="/root/inputcamav/"
INPUTFOLDER="/root/inputcamav/"

#
# LOGFILE: This is the path to the file where the log of CAMAV is
# maintained
#
#LOGFILE="/root/inputcamav/updateslog.txt"
LOGFILE="/root/inputcamav/updateslog.txt"

#
# WAITING_TIME: This is the parameter to tell to the machine how much
# time to stay online (to update)
# maintained
#
#WAITING_TIME=5 #1800
WAITING_TIME=1200 #1800


#
# SLEEPING_TIME: This is the parameter to pause the script for number
# of seconds, to allow some tasks to be finish. The default values
# is 15 seconds. A bigger value could lead to long waiting time,
# therefore the updating time will be longer. A shorther value
# could lead to have some unfinished tasks
#
#SLEEPING_TIME=15
SLEEPING_TIME=15 #15



#
# MACHINENAMES: This parameters contains the machines that are used
# to scan the files.
# When a new machine is added here should be the added
#
#-------------------------------------------------------------------------------------------------------
#       avso - Sophos avmc -Mcafee avfs - F-Secure avcl - Clamwin avav - Avira avvb - VirusBuster avavg - AVG avka
-Karspersky avno - Eset Nod32 avava- AVAST
#-------------------------------------------------------------------------------------------------------
declare -a MACHINENAMES=('avso' 'avmc' 'avfs' 'avcl' 'avav' 'avvb' 'avavg' 'avka' 'avno' 'avava');
```

```
#
# UPDATE_CHUNKS: This is the parameter to indicate how many machines
# should be allow to update in paralele. To allow them to update from
# internet. The recomanded value is 3 or 4. Smaller value will raise
# the overall update time, bigger value will increase the resources
# used in paralel (ram, internet band, ip addresses, etc).
# The update time could be calculated as number of chunks (this variable)
# multiplied by 30 minutes.
#
#UPDATE_CHUNKS=3
UPDATE_CHUNKS=4
```

2.4 The Guest

There are already exported VirtualBox hosts with the actual antiviruses installed and the script on the machine configured. There is also a template VirtualBox exported machine that can be used when is needed to add a new antivirus.

The virtual machine is a standard Virtual Machine with Windows XP SP3, and antivirus software installed.

2.4.1 The Antivirus Machines naming

Each machine has its own name used to uniquely indentify a particular machine while scanning a file. The name of the machines is used:

- in the VirtualBox to indentify the machine is used;
- in the configuration file to chose with which machine to scan a file;
- its used as a internal name of the machine;

The name of the machine is composed from “av” plus the first two or three letter of the antivirus name. Therefore, Sophos machine is avso, the McAfee machine name is avmc, F-Secure avfs, etc. When the machine name already exist, three letter could be added. E.g AVAST antivirus avava.

2.4.2 Windows XP

You need to perform some changes on your Windows XP guest environment:

- Disable the screen server;
- Disable the Desktop cleaner;
- Rename the internal machine name as av+antivirus name (see naming);
- The system has a username “test” with the password “123456”;
- The Windows Firewall and Automatic Updates should be disabled;
- Install VirtualBox Guest Additions;

2.4.3 Shared Folders

CAMAV uses VirtualBox's Shared Folders to exchange data between the Host and the Guest. The Virtual Machine should be provided with one shared folders:

- A "inputcamav" folder used to map the host folder containing the file to be scanned The shared folder needs to be setup up with the following privileges: full access, auto-mount and permanent. For the moment the shared folder should point to /root/inputcamav/

2.4.4 Mapped folder

A mapped folder with the letter Z should be made out of the previous shared folder. Check the box “reconnect at logon”.

2.4.5 Network Configuration

Select bridge mode. This is used in order to allow the system to update. This will be probably changed in the future version of the system where all the host will be set as NAT to avoid the request of two many routable IP addresses.

2.4.6 Internal Scripts

Each system has his own folder Scripts “c:\script”. In this folder are stored the scripts that needs to run on the system (E.g avso.bat –the script for Sophos scan). These scripts need to be customized based on the antivirus specification running on that system.

E.g. F-Secure internal scanning script

```
echo off
REM This is the script for scanning a particular file passed as argument. The format of the command is avmc.bat c:\filename.exe
echo on

cd "C:\Program Files\F-Secure\Anti-Virus\"
set outputfilename=%~dp1avfs.txt
echo off
rem this command is actualy made from  set outputfilename=  %~dp1avmc.txt. This parameter is the path of the file argument passed to the script %~dp1
echo on
fsav.exe /report=%outputfilename% /archive  %~1
cd "c:\script\"
shutdown –s –f –t 3
```

2.5 Web Interface

The web interface is normally located on the system home folder (/var/www/camav/) This folder is made up from the following subfolders

upload – used to store uploaded files
tmp – used for temporary movements of the files while processing
style – containing css for the page
processed – used for internal processing.

The main home folder /var/www/camav/ contains among other image files , etc the following files:

is_file_done_new_version.php –internal processing
upload_file_new_version.php – Interface for uploading a file
search.php – interface for searching a file
new_search_integration.php – used as a web service

NOTE: The name of these files will be changed

2.6 CAMAV webservice

We can send files to CAMAV using a web service. We can POST samples to CAMAV and receive the results as a JSON.

2.6.1 Submit a file

To analyze files with CAMAV, we have to POST the file to the URL:
http://localhost/camav/upload_file.php using these POST parameters:

file = Path to the file
perId = "36875"
viewDate = """
loadFlag = "false"
btnSubmitFile = "Upload"

2.6.2 Get the results

To get the results of the analysis, we have to GET the status from this URL:
http://localhost/camav/new_search_integration?hash=MD5&nsub=1

hash - represents the MD5 of the file
nsub - is used to force the system to wait until the new analysis is finish, if nsub=0 the system will return the last analysis.

The result of the GET query can be one of these:

A) A string such this: "235 0 in the queue" where:

235 represents the estimated time before the analysis ends
0 is the number of samples in the queue.

B) A JSON data structure with the results. For example such this:

```
{
	"avres":[
			["positive","Sophos","9.5","2011-11-21","Exploit.PDF-JS.Gen"],
			["","Mcfee","8.7.0i","2011-11-21","-"],
			["positive","F-Secure","9.01","2011-12-19","Mal/JSShell-B"],
			["","Clamav","0.97","2011-12-19","-"],
			["","Avira","10.2.0","2011-1-19","-"],
			["","VirusBuster","7.1.74","0000-00-00","-"],
			["positive","AVG","10.0.390","2011-11-22","Exploit.SWF"],
			["","Karspersky","6.0","2011-12-19","-"],
			["postive","Eset","4.2.71.2","2011-11-22","JS\/Exploit.Pdfka.ONN
trojan"],
			["positive,"AVAST","4.8","2011-11-22"," JS:Pdfka-gen [Expl"]
		],
	"detected":5,
	"total":"50.00"
}
```