GeCAD srl

The Software Company

User's Guide

RAV AntiVirus for Mail Servers

Release date for this User's Guide: May 30th, 2002
Version: 8.3.2

GeCAD Software a - Sos. Mihai Bravu, nr. 223, sector 3, Bucharest, Romania
Phone/fax +40 1 321 78 03, hotline: +40 1 321 78 59

Copyright © since 2001 GeCAD Software® s.r.l.

Terms and conditions of the License Agreement

RAV Reliable AntiVirus is a registered trademark of GeCAD Software s.r.l. (hereafter referred as "GeCAD"). ***RAV AntiVirus for Mail Servers*** (hereafter referred as "The Software") and the products from this family are offered to our clients under the terms and conditions of the License Agreement accompanying all the products of GeCAD.

Before installing or using The Software, please read carefully this License Agreement, because it represents a legal agreement between you and GeCAD for the software product you are installing, which includes the software itself and the related documentation. By installing or otherwise using the software, you accept all the terms and conditions of this agreement. If you don't accept the terms of this agreement, you don't have the rights to install or otherwise use The Software.

The License Agreement is available after installing the product in the following path: `\usr\local\rav8\docs\license`.

> ***! On the distribution CD-ROM you might find other programs, in addition to the one you have bought. These programs are offered for evaluation only and are making the object of separate terms of license. These terms are included in the Evaluation section of the License Agreement.***

Introduction

Congratulations! You have just acquired a ***RAV AntiVirus*** product. RAV AntiVirus is one of the best antivirus programs, ranked among the first ten in the world.

The RAV Antivirus products and the documentation associated to these products are the exclusive property of GeCAD Software. The products are licensed to the users under the terms of the License Agreement accompanying each product, so please carefully this License Agreement.

We suggest to take a moment to fill the *Registration Form* attached to the *License Certificate* or register to the manufacturer's site. The personal data you fill in this form is strictly confidential and will only be recorded in our database of registered customers to keep you informed on new developments and updates and to activate your technical support account. Any suggestion you make will be taken into consideration for future versions.

We value very much your opinion and try to fulfill our customers' requests as soon as possible. Remember that you, the user of RAV AntiVirus, can make the product better and more suitable to your needs.

The License Certificate is your guarantee that GeCAD Software legally licenses the program you have acquired, under the law 8/1996. The License Certificate grants you the right to use the product according to the terms of The License Agreement, to upgrade it to new versions at preferential prices, and to receive free revisions of the current version.

> ***! Because of the attention we pay to delivering highly efficient antivirus software, the rapid evolution of viruses and implementation of features requested by our customers, the present documentation may not be up to date. The best source of information on RAV is our website, which you can find at http://www.ravantivirus.com.***

Structure of this document

This document is structured in five major components.

The first part, *Introduction*, includes information on the *Scope* of this document, its *Intended audience* and the *Related documentation*. You are also briefed on the RAV AntiVirus' manufacturer, GeCAD Software, and informed about modalities you can use to obtain the *Technical support* you might need for the product you have acquired.

The second part, *RAV AntiVirus Product Family*, is an overview of the products included in the RAV AntiVirus Product Family (*RAV AntiVirus Desktop*, *RAV AntiVirus for Mail Servers, RAV AntiVirus Mail Filter* and *RAV AntiVirus for File Serves*). All these products are based on RAV Engine, a revolutionary antivirus engine. The cutting-edge technologies included in RAV Engine and the most important updates included in RAV Engine version 8.7 are also included in this *RAV AntiVirus Product Family* part.

The third part of the document, *RAV AntiVirus for Mail Servers*, includes some sections (*Short description, What can it do, How does it work, Who should use it*) briefly presenting RAV AntiVirus for Mail Servers, its functionality and the potential users. Sections describing the *Currently supported operating systems and MTAs* and the *Hardware and software requirements* are also included, as well as a section presenting in more detail the *Features* of RAV AntiVirus for Mail Servers. An *Awards* section containing international acknowledgements gained by this product is also included in this part of the document.

The fourth part of the document, *The license policy of GeCAD*, explains the main principles fundamental for the license policy of our company, including a detailed description of the main differences between the three different phases in which you can use a RAV AntiVirus product (evaluation, registration and activation).

The fifth part of the document, *Configuration files and other resources*, contains the manuals for configuring the RAV AntiVirus for Mail Servers, RAV mail scanning daemon and RAV AntiVirus command line version, as well as the configuration files for the external filters for the MTAs supported by RAV AntiVirus for Mail Servers. Three "How to…" files are also included, and a path to other "How to" files on our website is provided.

An *Index* section is included at the end of the document.

Scope

This document describes the features and functionality of ***RAV AntiVirus for Mail Servers***. Additional valuable documentation is also separately available, as specified in the ***Related documentation*** section below.

To make sure you will be efficiently using ***RAV AntiVirus for Mail Servers*** from the very beginning, we strongly recommend you to read carefully this User's Guide, even if you have been using a previous version of RAV AntiVirus.

Intended audience

This User Guide is intended for administrators responsible with the installation of ***RAV AntiVirus for Mail Servers***. These persons should have a strong and comprehensive knowledge and an extensive working experience in the operating systems the product is designed for.

Related documentation

Here is a list of documents that should be used in connection with this *User's Guide* for ***RAV AntiVirus for Mail Servers***:

- RAV AntiVirus for Mail Servers – *Product Sheet*;
- RAV Update – *User's Guide*;
- White Paper: Viruses – What they are and how to defend us against them.

These documents and other up-to-date documentation concerning GeCAD's products, as well as white papers on security policies and latest info about viruses are available on our web site at www.ravantivirus.com.

About GeCAD

GeCAD Software is a leading Romanian IT company specialized in providing top anti-virus solutions for all categories of users. After releasing its first antivirus program (RAV 3.2 PRO) in 1994, GeCAD Software has grown ever since to be represented, by Distributors, Resellers and OEM Partners, on all the continents around the world. Our strong commitment towards quality has secured us a privileged position in a fast-evolving market, the key advantage being a state of art product based on cutting edge technologies.

Founded in 1992, GeCAD Software is headquartered in Bucharest, Romania. Our activity is focused on the following major activity axes:

- Producing, developing and internationally distributing high-quality anti-virus products; and
- Providing reliable services in fields like software distribution, consultancy and Technical Support.

Technical support

For any details regarding the installation and the functionality of this product, please contact the local dealer you have bought the product from. If he does not offer you the adequate technical support, please contact us.

For any suggestions or problems regarding the copyright, the guarantee and other aspects related to ***RAV AntiVirus*** or data recovery from a destructive viral attack, please contact us at the following addresses:

GeCAD Software srl

Address:	223, Mihai Bravu Blvd, 3rd district, Bucharest, ROMANIA
Phone:	+40-1-3217803, 3217859
Fax:	+40-1-3217803
Hotline:	+40-1-3217859
E-mail:	support@ ravantivirus.com
Internet:	**http://www.ravantivirus.com**

To keep in contact with other users of RAV AntiVirus products, join the *discussion lists* available for our products or subscribe to the free weekly *newsletter* edited by GeCAD Software.

RAV Discussion Lists

Interesting ideas and insights, installing and configuration scenarios, troubleshooting solutions and other information are also available *via* specialized *discussion lists* for RAV Antivirus products:

- rav-sendmail - RAV AntiVirus for Sendmail
- rav-qmail - RAV AntiVirus for qmail
- rav-postfix - RAV AntiVirus for Postfix
- rav-cgate - RAV AntiVirus for CommunigatePro
- rav-exim - RAV AntiVirus for Exim MTA
- rav-desktop-windows - RAV AntiVirus Desktop for Windows
- rav-desktop-unices - RAV AntiVirus Desktop for Unices
- rav-mailrelay - RAV AntiVirus Relay for POP3,IMAP,SMTP
- rav-exchange - RAV AntiVirus for MS Exchange Server
- rav-nms - RAV AntiVirus for Netscape Messaging Server
- rav-enterprise - RAV AntiVirus Enterprise

You can subscribe to these discussion lists by visiting our website www.ravantivirus.com (*RAV Discussion Lists* section) or by sending an empty email message to: listname-subscribe@lists.ravantivirus.com (replacing "listname" with the adequate discussion list you want to subscribe to).

RAV Newsletter

A weekly *newsletter*, containing virus alerts, advisories and advices ways to avoid virus disasters, as well as information regarding updates, tips and tricks and insights on RAV AntiVirus products, is also available from GeCAD Software. You can subscribe to this newsletter from our website www.ravantivirus.com (*RAV Newsletter* section).

RAV AntiVirus Product Family

RAV AntiVirus for Mail Servers and all the other products included in the RAV AntiVirus family are based on the *RAV Engine*, now at version 8.7. The RAV Engine combines the operational strength, the extensibility, the scalability, the scanning speed and the robustness needed in the fight against viruses and other malicious software. The RAV Engine version 8.7 has some unique characteristics that made it one of the most appreciated antivirus engines in the world.

The ***RAV AntiVirus*** product family is currently having the following members:

- RAV AntiVirus Desktop (for Windows and Unices)
- RAV AntiVirus for Mail Servers
- RAV AntiVirus Mail Filter
- RAV AntiVirus for File Servers.

RAV Enterprise Server is now a distinct product, based on client-server technologies. You can find more details on *RAV Enterprise Server* at ***www.ravantivirus.com*** and by subscribing to the RAV Enterprise discussion list.

Cutting-edge technologies included in the RAV engine version 8.7

The most important distinctive characteristics of the RAV Engine are described below.

The TPI (Total Platform Independent) technology

The same engine is used for detection and cleaning for each operating system RAV AntiVirus is installed on. Thus, the intelligibility, the unity and the easy update of our programs are logically ensured for all products in the RAV AntiVirus Family (RAV AntiVirus Desktop, RAV AntiVirus for Mail Servers, and RAV AntiVirus for File Servers and RAV AntiVirus MailFilter).

The IC (Integrity Checker) technology

When the files are scanned for the first time, the detection engine creates a database with the information it has gathered during the scanning process. When doing a second scan, only the new or changed files are scanned, therefore increasing the detection speed with more than 50%.

The MLES (Multi Layer Embedded Scanning) technology

The engine is promptly responding to any threat, scanning embedded objects on multiple layers, without affecting the speed of detection or slowing down the machine it is installed on.

The HMETH (heuristic method) technology

Using this technology for all electronic threats, the RAV engine can study the behavior of eventual malwares (malicious software, like worms, Trojans and hoaxes) and propose different methods of handling it. Therefore, a virus can be detected even if its signature doesn't exist in the database.

The CUP (Cumulative Update Plug-ins) technology

The cumulative update is another advantage of the RAV engine, being used to add to the main signatures database only the latest available signatures. This procedure results in extremely small files used for the update (10-15Kb), very little time for the download process (5-10 seconds at a 28,8 kb/s connection speed) and better management of the virus signatures.

RAV engine 8.7 vs 8.5

RAV Engine version 8.7 was released on May 7th, 2002. In comparison with the precedent version of RAV engine (8.5), the current version is enjoying the following improvements:

- scanning inside the following types of archives was added: HA, HAP;
- the support for RTF/RTFHtml files was improved - the scanning speed in these files should be faster and the support for strange/damaged files should be greatly improved;
- fixed several bugs in ZIP/ARJ/CAB/RAR damaged archives handling;
- the HTML handler can now detect and report hidden IFRAMEs, used by a large number of viruses to auto-execute attachments - such IFRAMes are reported as HTML/IFrame_Exploit*. This way we'll be able to detect new viruses that will use this exploit to spread;
- scanning inside BinHex encoded files was added;
- the MIME handler was improved to handle various specially crafted MIMEs;
- the PST handler has been optimized for large PST files;
- scanning inside the following packed objects was added/improved: PkLite32, VgShrink, PeNightmare, PeProtector, FSGPE, SecuPack, PEMangle, PackLite, XPeOR, NoodleCrypt, tElock, Neolite, PEtite, AsPack, PEDiminisher, PcShrink,PeX, Bjfnt, PECompact, CodeCrypt, PCPec, EZip, YodaCrypt, PeNinja, PeProtect,Pklite, DbgScr, Merlin, OleData and SimBoot;
- scanning inside CHM [Compiled HTML files] was added as a response to the new VBS/Chick family;
- an improved code emulator was added to speed up polymorphic/metamorphic virus detection for both Windows and DOS viruses;
- the virus detection patterns for a large number of DOS viruses were changed to avoid false alarms and improve the scanning speed;
- support for LE/LX executable scanning was improved;
- added a new heuristic engine for VB5/6 compiled internet worms/viruses;
- improved the heuristic engine for PE executable files;
- removed several false alarms.

RAV AntiVirus for Mail Servers

Short description

RAV AntiVirus for Mail Servers is exactly what its name is telling…and much more: an award-winning antivirus program for mail servers using different operating systems on different platforms. It is designed to protect the e-mail messages flowing directly from/to your mail server, regardless of the format for these e-mail messages.

What can it do

RAV AntiVirus for Mail Servers scans and cleans e-mail messages and all types of attachments including archives, exe files, embeded files, etc. It helps you avoid Internet malwares (viruses, worms, Trojans, hoaxes, etc.), bulk mail and information leaks. ***RAV AntiVirus for Mail Servers*** is scanning, detecting and removing any electronic threats from the messages flowing to/from your mail server, therefore protecting important data for your company and preventing your computers to be infected by viruses, worms, Trojans and other malwares. For instance, the latest daily update to the engine of AntiVirus (available at the date this *User's Guide* was written) added another 601 records, so that ***RAV AntiVirus for Mail Servers*** is now protecting you of 68.678 malwares (as of June 4, 2002).

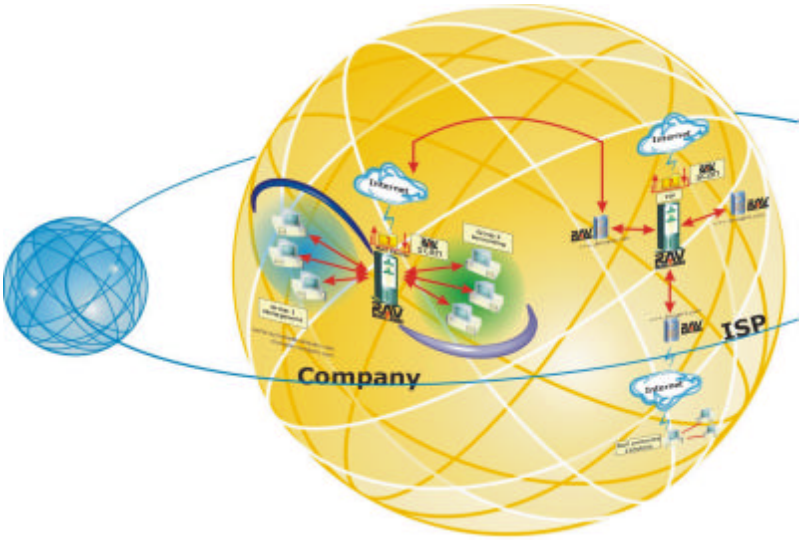
How does it work

The product contains a demon based on the *RAV engine* and a filter module that interfaces the demon with the mail server. When an e-mail message is reaching a mail server protected by RAV AntiVirus, it is intercepted by the filter and sent to the demon for scanning (the scan process is done using the RAV engine). If the message is clean, it is sent back to the filter, which will decode and redirect it to the mail server, to be sent - scanned and cleaned – to the initial destination. If the message is not clean, customizable actions are taken, according to the options set by the user.

Who should use it

RAV AntiVirus for Mail Servers is targeting three categories of users:

- **large companies**, with heavy traffic on their mail servers and demanding security policies;
- **Internet Solution Providers** (ISPs – over 60% of them are using Linux-based mail servers). RAV AntiVirus is the perfect solution for the mail servers of the ISPs, which are dealing with heavy traffic and large amount of clients. RAV AntiVirus can improve the services ISPs are providing to their customers by scanning the e-mail flow and adding protection against viruses for the hosted domains; and
- **small companies** willing to protect their internet traffic at low cost.



No matter if you're a big or small company or an ISP, the protection offered by ***RAV AntiVirus for Mail Serves*** is always covering different levels:

- ✓ You are protected of viruses and other malwares that might try to infect your machines coming from the Internet, as mail attachments or e-mail messages.
- ✓ The outgoing flow is scanned for viruses and other malwares.

You can also customize the outgoing mail flow for not allowing sensible information to leave your company.

Awards

Starting with March 1st, 2002, ***RAV AntiVirus for Mail Servers*** (Linux) is awarded the West Coast Labs'[1] Checkmark Certificate *Level 1* and *Level 2*, the two standards to be achieved by antivirus products. *RAV AntiVirus for Mail Servers* is the first (and so far the only) anti-virus product winning both certificates for Linux-based mail servers.
The Checkmark Certificate system establishes standards for computer security products, giving end users a clear idea on reliable anti-virus products.
For a product to be certified to Anti-Virus Checkmark, *Level One*, the product must be able to detect all "in the wild" viruses. According to Checkmark, "In the Wild" viruses are currently defined as those appearing on The Wildlist Organization's "In the Wild" list and reported as such by more than one person.
For a product to be certified to Anti-Virus Checkmark, *Level Two*, the product must be able to comply with Checkmark Level One and, in addition, disinfect all "in the wild" viruses capable of being disinfected. The product must also detect all viruses on the wild list more than one month old.
After rigorous tests, conducted in the West Coast Labs, RAV AntiVirus for Mail Servers (Linux) proved to successfully detect and disinfect 100% of all in-the-wild types of viruses[2]. Anti-Virus Checkmark Level 2 being granted to RAV AntiVirus for Mail Servers (Linux) indemnify our product's users to have complete confidence in its abilities to prevent infection and insure disinfection by viruses on the "In the Wild" list.

Currently supported operating systems and MTAs

The following categories of operating systems are currently supported by ***RAV AntiVirus for Mail Servers***:

- Linux variants: Slackware, Mandrake, SuSE, RedHat, e-Smith (SME) and Debian;
- Unices: BSD (FreeBSD, Open BSD, NetBSD); Solaris (i386 and SPARC); Unixware;
- Win32 (Windows NT and Windows 2000).

The following MTAs are currently supported by ***RAV AntiVirus for Mail Servers***:

- Communigate Pro
- Courrier
- DMail
- Exim
- MS Exchange
- Postfix
- Qmail
- Sendmail
- Sendmail (with libmilter feature).

Here is a cross-table presenting the currently supported operating systems and MTAs:

Mail Server / OS	Linux	Free BSD	Open BSD 2.8	Open BSD 2.9	Open BSD 3.0	Net BSD	Solaris I386	Solaris SPARC	Unixware 7.1.1	Win 32
Sendmail	✓	✓	✓	✓	✓	✓	✓	✓		
Sendmail Libmilter	✓	✓	✓	✓	✓		✓	✓		
Qmail	✓	✓	✓	✓	✓	✓	✓	✓		
Postfix	✓	✓	✓	✓	✓	✓	✓	✓		
Exim	✓	✓	✓	✓	✓	✓	✓	✓		
CommuniGate Pro	✓	✓	✓	✓	✓		✓	✓	✓	✓
MS Exchange 5.5										✓
MS Exchange 2000										✓
Courier	✓						✓	✓		
DMail	✓						✓	✓		✓

Further developments

GeCAD Software is currently working on developing antivirus solutions for different operating systems and platforms. For details on the developing process and other info about the characteristics of our products, please visit our website: *http://www.ravantivirus.com*.

[1] West Coast Labs is an independent organization testing information security products. It is owned by West Coast Publishing Limited which also owns *SC Magazine*, the largest circulation information security magazine in the world.
[2] For an overview of different types of viruses and other malwares and methods for protecting against them please consult the white papers available on GeCAD Software's website.

Features of RAV AntiVirus for Mail Servers

Some of the distinctive features of RAV AntiVirus for Mail Servers are presented below:

- *Simple installation process*: The installation is simple and can be made using an interactive install script. To install RAV AntiVirus for Mail Servers, *unzip* the corresponding file (`gunzip filname`) and *untar* it (`tar -xvf filename`), change the directory to RAV and type `run ./install.sh`.
- *Easy to configure and use:* RAV AntiVirus for Linux Mail Servers is extremely easy to configure: options are available to order the actions to be taken by RAV AntiVirus when dealing with an infected file (Clean, Move/Copy to Quarantine, Delete, Rename, Ignore, Reject) or with a suspicious file (Move/Copy to Quarantine, Delete, Rename, Ignore, Reject).
- *Multi platform virus removal*: RAV Engine detects and removes all known Windows, Linux, Unix and DOS viruses, regardless of the operating system they're stored on or designed for, using heuristic methods to extend the protection it is offering to its users.
- *Integration:* RAV AntiVirus for Mail Servers is an **integrated** suite, containing all the necessary components in one single installation.
- *Enhanced virus scanning:* Mail attachments with multiple recipients are scanned only once, and not for all the recipients, therefore enhancing the scanning speed. Different technologies are also applied, in order to improve the speed and accuracy of the scanning process.
- *Content filtering:* All incoming/outgoing e-mail messages are scanned by *Subject*, *Attachment* and *Body*. This way, you can deny incoming e-mail messages containing suspicious objects and outgoing e-mail messages containing confidential information, reducing all types of threats to the very minimum. The available options for email files matching the content filter are: Move/Copy to Quarantine, Delete, Ignore, Reject. For more details, read the RAVMD daemon's configuration file.
- *Group configuration:* The system's administrator can define different groups of users and RAV AntiVirus for Mail Servers will scan these differently configured groups, according to their specified scanning needs.
- *Bulk mail prevention:* Using the *Content filtering* and the *Group configuration* features, RAV AntiVirus for Mail Servers can act as a spam filter, blocking incoming e-mail messages from certain addresses, according to the settings made by the server's administrator. This is more than protecting against viruses, it is about protecting privacy: you can specify that you do not want to receive bulk mail from certain addresses or mail messages with certain file formats (for example .scr attachments, e-mails with "I Love You" subject, or XXX mail).
- *Preventing information leaks:* RAV AntiVirus for Mail Servers restricts the e-mail flow according to defined parameters/patterns. You can configure specific groups (for example one group named Accounting, including all the computers from your company's accounting department) and set RAV AntiVirus to deny all outgoing e-mail messages defined as containing confidential informantion, according to the patterns set by the server's administrator.
- *Warning notifications:* You can configure RAV AntiVirus for Mail Servers to send instantly warning notifications when a virus allert occurs to the sender of the message, to the receiver and/or to a third party (for example RAV Research Team or the server's administrator).
- *Intelligent Update:* The Update process may be performed on demand or on a scheduled basis, according to the administrator's settings. The latest versions of RAV AntiVirus for Mail Servers (including engine updates and documentation) can be found on the following mirror sites:

  ftp.ravantivirus.com/pub/rav/mailservers (site located in Romania)
  download.ravantivirus.com/pub/rav/mailservers (site located in USA).

  New mirror sites are expected to be available for update. Please check regularly our website to find out the addresses for these new mirror sites.

  The updates for ***RAV AntiVirus for Mail Servers*** are run automatically via a **cron** script that is automatically creating during the installation on Unix variants. You may modify this script to change e-mail notification parameters for updates.

Other features of RAV AntiVirus for Mail Servers

- Modular structure, allowing easy customization for every work environment;
- Extended protection (for new mail boxes);
- Scanning archives inside archives;
- Scanning for packed executables;
- Multiple MIME-type encodings;
- Free technical support available by e-mail and phone for registered users.

Hardware and software requirements

Software requirements:

RAV AntiVirus for Mail Servers has been tested on the following flavors of Linux: Slackware 8.0, Mandrake 7.2, SuSE 6.4, RedHat 6.2, eSmith (SME) and Debian Linux. The Linux kernel version (for the Linux flavors) should be 2.2 or later. Fully functional installation kits for some MTAs are also provided for the following operating systems**:** Free BSD (version 4.1 or later), Open BSD (version 2.8, 2.9 and 3.0), Net BSD (version 1.5 or later), Solaris 8 (i386 and SPARC), Unixware 7.1.1 and Win32.

The requirements for the Unices OS's libraries are the following:

- glibc version: 2.1.3 or later;
- rpm version: 3.0 or later;
- zlib version: 1.1.3 or later.

For Win32 OSs, the requirements are as following: Windows 2000 with Service Pack 2 or Windows NT 4 with Service Pack 4.

RAV AntiVirus for Mail Servers has been comprehensively tested on the following MTAs:

- CommuniGate Pro version 3.4b1 or later.
- Courier version 0.38.0 or later.
- DMail version 3.0 or later.
- Exim version 3.33 or later.
- Postfix version 20000531 or later.
- Qmail version 1.03 or later.
- Sendmail version 8.11 or later (for LIBMilter version only).
- Sendmail (any version).

Hardware requirements:

RAV AntiVirus for Mail Servers will run on a computer meeting the minimal requirements for the installed Mail Server program. For Intel-based systems, this means:

Processor: Compatible Intel CPU, 150 Mhz or better;

Memory: 32 MB of RAM or more.

For computers with SPARC processors, the hardware requirements are:

Processor: SPARC V8 or later

Memory: 32 MB of RAM or more.

Note: In order to send warning messages, according to the settings made in the scanning configuration, you must have a MTA running on the local machine and listening on port 25.

Registration procedure

Due to an extremely flexible licensing system, you can acquire different packages, depending on your needs of protection. ***RAV AntiVirus for Mail Servers*** is licensed *per domain*. The more domains you support, the less you pay per domain.

All the products included in the RAV AntiVirus family are available under a license scheme with three different stages: *evaluation*, *registration* and *activation*. Each different stage has its own characteristics, described below.

Evaluation

The products included in the RAV AntiVirus family are fully functional and their users benefit of complete update services and free technical support for an *evaluation period* of 30 or 60 days (depending on the product). During this evaluation period, our potential customers should be able to evaluate all the functionalities and services provided by our products, in order to make a proper purchasing decision. For details regarding your rights and obligations pertaining to the usage of GeCAD's products during the evaluation version, please read the *Evaluation license* section from the License Agreement.

During the evaluation period, each e-mail message scanned with a RAV AntiVirus product contains info about the number of days remaining for evaluation. After this evaluation period, the product expires. If you do not introduce a valid *Registration Code* during the Evaluation period, you will not be able to use the product anymore. Introducing (anytime) a valid *Registration Code* extends *ad infinitum* the lifetime of the product.

Evaluation for RAV AntiVirus for Mail Servers

RAV AntiVirus for Mail Servers can be used for two domains for an evaluation period of 60 days. If you do not register and activate the product within 60 days, you will no longer be able to use it to scan the e-mail traffic.

Registration

In order to license the ***RAV AntiVirus for Mail Servers***, you have to purchase it from an authorized RAV reseller/distributor. When you purchase the ***RAV AntiVirus for Mail Servers*** from an authorized RAV reseller/distributor, you receive a *Registration Code*. If the purchasing involves physical delivery, the *License Certificate* will also be available.

After registering your product, you have 30 days to *activate* it by installing an *Activation key* (please refer to the next section to see how you can do it). Activating your registered product is very important, because it offers you *free* updates and technical support for *one year*.

Activation

The product's activation is done free of charge, *upon request*, by sending your End User information to GeCAD Software (https://register.ravantivirus.com; alternatively, you can fax this info to the number specified on the License Certificate). Subsequently, you will receive by e-mail an *Activation Key* and instructions for its installation. Installing this Activation Key enables you to use the ***RAV AntiVirus for Mail Servers*** for the number of domains you have purchased and offers you the following integrated services:

- Free engine extensions for one year;
- Free daily updates for one year;
- Full technical support for one year;
- Virus alerts and security advisories.

You can extend the rights resulted from activating your product by purchasing annual update extensions available at special discounts.

Updates

The latest versions of RAV AntiVirus products (including engine updates and documentation) can be found on the following mirror sites:

ftp.ravantivirus.com/pub/rav/ (site located in Romania)
download.ravantivirus.com/pub/rav/ (site located in USA).

New mirror sites are expected to be available for update. Please check regularly our website to find out the addresses for these new mirror sites.

Configuration files

In the following section you will find:

- The configuration file for RAV AntiVirus for Mail Servers (ravmd.conf)
- The configuration file for RAV mail scanning daemon (ravmd)
- The configuration file for RAV AntiVirus command line version (ravav)
- The configuration files for the external filters for the MTAs supported by RAV AntiVirus for Mail Servers:
    - ✓ ravcgate - external filter for CommuniGate Pro
    - ✓ ravcourier - external filter for Courier
    - ✓ ravdmail - external filter for DMail
    - ✓ ravexim - external filter for Exim
    - ✓ ravpostfix - external filter for Postfix
    - ✓ ravqmail - external filter for Qmail
    - ✓ ravsendmail - external filter for Sendmail
    - ✓ ravmilter - external filter for Sendmail (with Milter features)

These files are not meant to represent an exclusive documentation for RAV AntiVirus for File Servers, as additional documentation might also be found:

- In the *tar.gz* files containing the installation kit for the product you are using. Here you can find the *ReadMe, Install* and *Uninstall* files providing valuable information for the distinct operating system and mail server you are using.
- On the manufacturer's website (www.ravantivirus.com). Here you can find additional "How To" files and "Frequently Asked Questions", the latest product updates, as well as documentation and information about our new products.

Introduction

This is the configuration file for **ravmd** daemon, **ravmd.conf**, and it contains info on the run-time configuration for the RAV AntiVirus mail scanning daemon. After installing the product, **ravmd.conf** resides in the `/usr/local/rav8/etc/` directory.

Structure

This configuration file consists of:

- description of three declaration sections (the *Regular Expression Declaration* section, the *Action Definitions* section and the *Warning Mails Message Declarations* section),
- group definitions, and
- explanation of possible parameters.

The information included in this part of the documentation is for reference purposes. Other valuable sources of information are also available. Information pertaining to the installing and un-installing processes, for instance, is included in the *Install* and *Uninstall* files provided in the *tar.gz* files containing the installation kit for the product you are using. Additional information about RAV AntiVirus for Mail Servers (hardware and software requirements, list of installed files, instructions for configuration and updating) is provided in the *ReadMe* file, included in the installation kit.

When presenting the possible parameters in **ravmd**, besides the default values and examples for these parameters *Frequently Asked Questions* are also provided, to help you better understand the functions of **ravmd** and know all the options available in it.

Definitions

Some concepts frequently used in the programming world might have different understandings for the purpose of this manual. Here you can find these terms and the meanings they are given in this document:

- a **variable** is a place where we can store data;
- a **string** is a sequence of characters ending with a new line or delimited by quotes;
- a **boolean** is one of the keywords: **yes/no**;
- an **enumeration** is a sequence of words separated by commas;
- a **regexp** is a POSIX regular expression;
- a **group** is a category of users (senders or receivers) that have different email addresses and/or domains, but share the same action parameters for **ravmd**;
- a **macro** is a stored template of instructions to be replaced with actual values by **ravmd**;
- a **commented** line is a line beginning with a hash (#) character. All commented lines are ignored. All the lines containing only white spaces are also ignored;
- a **default** is a predefined value for one parameter. If that **parameter** is missing from `ravmd.conf` then it is considered to have that **default** value.

The values following the '=' sign in *parameters* may be: a **string**, a **boolean**, a **regexp** or an **enumeration**.

Note: In **ravmd**, the section and parameter names are **not** case sensitive.

Section Descriptions

Three different sections are included in the `ravmd.conf` man page: the *Regular Expression Declaration* section, the *Action Definitions* section and the *Warning Mails Message Declarations* section. These sections are presented below in the following format:

- Short description;
- Keyword representing the beginning of the section;
- Syntax;
- Example.

Some Frequently Asked Questions (FAQs) are also provided in connection with the relevant aspects of these sections. The Frequently Asked Questions are meant to help you with the practical aspects of using RAV AntiVirus for Mail Servers.

The Frequently Asked Questions were selected from the questions asked by users of RAV AntiVirus for Mail Servers on the mailing lists available for our products. For details on how to subscribe to these mailing lists, please read the *RAV Discussion Lists* section in the *User's Guide for RAV AntiVirus for Mail Servers*.

Regular Expression Declaration section

In the *Regular Expression Declaration* section you can define all the regular expressions you will use for the content filtering feature. This section can appear anywhere in the configuration file, as long as it is placed before the group definitions.

This section begins with the keyword:
`_define_regular_expressions`.

Syntax:
variable = string
or
variable = regexp

Example 1:

```
for_subject_filter = I love you
```

This regular expression defines a filter for all the e-mails including the string "I love you" in the **Subject**.

Example 2:

```
file_regexp = .*\.((vbs)|(vbe)|(exe)|(com)|(pif)|(lnk)|(scr)|(bat)|(shs)|(sh))
```

This regular expression defines a filter for all the mail messages containing attachments with the specified extensions.

Action Definitions section

In the *Action Definitions* section you can define the actions you want to be used by **ravmd**. The *Action Definitions* section can appear anywhere in the configuration file as long as it is placed before group definitions.

This section starts with the keyword:
`_define_actions.`

Syntax:
variable = enumeration

Depending on the scanning status of the e-mail file (infected, suspicious, subject/attachment/content filter match), the variable will be associated with some different actions (*clean, move, copy, delete, rename, ignore, reject*).

The enumeration contains one or multiple actions separated by a comma. The enumerated actions are executed by **ravmd** in the specified order. After the first successful action, **ravmd** goes on to scan the next MIME part from the current mail message.

The following table includes a description of all the actions available in **ravmd** (first column), a description of these actions (second column) and a listing of circumstances when each specific action is available (third column), depending on the scanning status:

Action	Description	Scanning status
Clean	Asks RAV AntiVirus to clean the infected file.	Infect
Move	Asks RAV AntiVirus to move the file to quarantine (equivalent to `Copy` + `Delete` actions).	Infect, suspicious, attach match, content match
Copy	RAV AntiVirus is copying the file to quarantine.	Infect, suspicious, subject match, attach match, content match
Delete	RAV AntiVirus is deleting the file and replacing it with a new file automatically generated. The file's name is `warn.txt` and it contains the following text: *"RAV AntiVirus has deleted this file because it contained dangerous code."* This text is customizable (please see FAQ no 1 for details). Note that RAV doesn't change the mail file size because of some protocols (like IMAP) may request the mail size first and then the mail body. So, the `warn.txt` file will be filled with spaces to fit the original file length.	Infect, suspicious, attach match, content match
Rename	The file will be renamed using the `rename_ext` extension specified in the configuration file.	Infect, suspicious, attach match
Ignore	The file is ignored, no action is taken and the e-mail is delivered.	Infect, suspicious, subject match, attach match, content match
Reject	The e-mail is rejected; it will not be delivered to any of its recipients.	Infect, suspicious, subject filter, attach match, content match

Depending on the scanning status, the following valid actions are supported:

a. *for infected files*: clean, move, copy, delete, rename, ignore, reject.

b. *for suspicious files*: move, copy, delete, rename, ignore, reject.

c. *for e-mail files matching the subject filter*: copy, ignore, reject.

d. *for files matching the attachment filter*: move, copy, delete, rename, ignore, reject.

e. *for e-mail files matching the content filter*: move, copy, delete, ignore, reject.

For more information pertaining to the definition of filters, please refer to the section *Advanced Content Filtering feature* below.

The following Frequently Asked Questions will help you better understand the characteristics of the actions available in **ravmd**.

FAQ 1: Scanning existing mails

Question: I have just installed RAV AntiVirus for Mail Servers. How can I scan e-mail messages existing prior to this installation?

Answer: If you wish to scan email messages existing prior to the installation of RAV for Mail Servers, do the following:

```
/usr/local/rav8/bin/ravav -AM --smart --report=/tmp/ravreport.txt [path to mail accounts]
```

This setting results in scanning a mailbox (with *ignore* as default action) and delivering a `ravreport.txt` in the `tmp` directory.

FAQ 2: Separate domain file

Question: Is it possible to put all scannable domains in a separate hash or normal file?

Answer: First include the domains you want to be scanned in `/usr/local/rav8/etc/domains.list`:

```
###domains to scan (separate them with ',')
domain1.com, domain2.com, domain3.net,
domain4.com,
domain5.org
###
```

Then include this `domain.list` in the `ravmd.conf` for the [`global`] group:

```
domain = _include /usr/local/rav8/domains.list
```

FAQ 3: Changing the contents of the warn.txt file

Question:

Recipients who get a virus-infected email have their attachments replaced with a `warn.txt` file. How can I change the contents of the `warn.txt` file?

Answer: Please edit the file

`/usr/local/rav8/etc/language/english`

and customize the value of

`warn_txt_msg_english = "Your message"`.

In `/usr/local/rav8/etc/languages/english.equiv` specify

```
warn_txt_msg  = warn_txt_msg_english
```

As is the case for each change in the RAV config files (`ravmd.conf, global, english, english.equiv`), you must restart **ravmd** with:

```
kill -HUP `cat /usr/local/rav8/private/ravmd.pid`
```

This way, your customized message will appear after the string: "RAV has deleted this file because it contained dangerous code."

> **Note:** `warn.txt` has exactly the size of the deleted attachment. Sometimes, because of the small size of the attachment you will not be able to view your complete customized message. For instance, in case of sending an e-mail containing the eicar.com virus (with only 64 octets), the message “RAV has deleted…” itself will not appear completely.

Warning Mails Message Declarations section

In the *Warning Mails Message Declarations* section you can define the subjects and the messages for the warning mails that will be sent to those who are interested. The *Warning Mails Message Declarations* section can be declared anywhere in the configuration file as long as it is placed before the group definitions.

This section starts with the keyword:
`_define_warning_mail_messages`

Syntax:
variable = string

The string defining one warning mail should give the user some basic information such as: what file has caused the warning, who sent that file and whom was it intended for, what is the warning about, what action was taken by **ravmd** and so on. One example of good warning string would be:

> "*That* file coming from *that* sender and addressed to *this* user is infected with *this* virus. The action taken by ravmd was *this*."

Of course, not all the warning mails are about virus-infected files. There are some other situations when you might as well want to be alerted by **ravmd**. For instance, you might want **ravmd** to alert you when an e-mail containing one specific type of attachment has arrived on your mail server or when one of your users is trying to transmit a confidential document. Of course, the contents of the warning mail should change in these circumstances according to that specific situation. However, some information (the name of the file causing the alert, the sender, the action, etc.) should *always* be included in the warning mail.

This information being always included in a warning mail is based on *macros*. One macro is (for the purpose of this document) a stored template of instructions containing one piece on information in the string representing the warning mail. `FILE_NAME`, for instance, is an example of macro containing the name of the file causing the alert you're receiving. **ravmd** will replace this macro with the actual name of the trouble-making file.

The string that the warning message will be created from can contain the following macros:

`FILE_NAME`	The full path to the scanned file and its name,
`ATTACH_NAME`	the name of the scanned attachment,
`VIRUS_NAME`	virus name found,
`FROM_USER`	mail sender address,
`TO_USER(S)`	mail addresses of receiver(s),
`SUBJECT`	mail subject,
`QUARANTINE_NAME`	the name of the file saved to quarantine by the *Move* or *Copy* action.
`SAVED_FILE_NAME`	The name of the mail file saved to quarantine when `save_infected=yes` and/or `save_suspicious=yes`.
`HEADER_RECEIVED`	This macro is replaced with all the **Received:** lines from the e-mail header (if this information exists). This way the infected machine can be found more easily.
`HEADER`	This macro is replaced with the entire e-mail header.

These macros are combined in one warning string that might look like this:
`"The file ATTACH_NAME attached to mail (with subject:SUBJECT) sent by FROM_USER to TO_USER(S)is infected with virus: VIRUS_NAME".`
In this case, one attachment is virus-infected. Information is provided about the attachment's name, the subject of the e-mail containing the infected attachment (this is an optional info), the sender's name, the addressees' names and the virus name.

To eliminate any uncertainty, here are some lines about the differences between two of the most confusing pairs of macros:

1. `QUARANTINE_NAME` *vs* `SAVED_FILE_NAME`

   `QUARANTINE_NAME` represents the name of the file saved to Quarantine as a result of using the *Move* or *Copy* action. The file is saved by **ravmd** in the Quarantine directory with the .qto extension. The files are encrypted, but **ravav** version 8.3.3 can decrypt them.

   `SAVED_FILE_NAME` represents the name of the file saved to Quarantine as a result of using the parameters `save_infected=yes` and `save_suspicious=yes`. The file is saved by **ravmd** in the Quarantine directory in the UNIXTIME-RAV {PID_OF_RAVMD_PROCESS} format.

2. `HEADER_RECEIVED` *vs* `HEADER`
   The `HEADER` macro is replaced by the complete e-mail header, while `HEADER_RECEIVED` is replaced only by the lines beginning with "**Received:**". Therefore, the lines *Message-Id*, *X-Sender* and *X-Mailer*, for instance, are only included when the HEADER macro is used, but not when the `HEADER_RECEIVED` macro is used.

The following Frequently Asked Questions will hopefully help you better understanding this very interesting feature or **ravmd**.

FAQ 4: Warning messages not sent

Question:
Why don't I get warning messages when viruses are found?

Answer:
You probably did not correctly configure the **ravms** mail account. Here are the excerpts from documentation that should help you:

"Default values:
`ravms_name = ravms`
`on_host` = the official name returned by `gethostbyname()` function
`smtp_server` = same as host (IP address)
`smtp_port = 25`
Default values are specified and they will probably work. Define these fields only if warning mails are NOT sent when a virus is found or if you want to use a different account instead of **ravms**. Specify **smtp_server** IP address only if that machine is behind a firewall and **ravmd** can't get its host name".

FAQ 5: Omitting the SUBJECT from the warning mail

Question:
Is it possible to define that SUBJECT should be omitted from the warning for specific viruses? Some of the viruses disclose potentially sensitive information from the victim's hard drive, and puts it in the subject of the email.

Answer:
In the `/usr/local/rav8/etc/languages/english` file, edit the line:
`infected_m_english = "The file ATTACH_NAME attached to mail (with subject:SUBJECT) sent by FROM_USER to TO_USER(S)is infected with virus: VIRUS_NAME."`
and remove `"with subject: SUBJECT"`.
The subject will not be displayed in the warning mails anymore. Please note that this solution will be applied to all warning messages for that specific group, and not just for "infected" messages.

FAQ 6: Changing the warning mail message

Question:
When a mail with .exe or .com attachment is sent, a warning message is also sent, which is OK, but it says the file is infected with the virus: `UNAUTHORIZED_MAIL_ATTACHMENT`, which seems to scare the average user. Can I modify this message?

Answer:
Yes, the message can be modified.
In `/usr/local/rav8/etc/languages/your_language` file
define one variable, for instance:

```
infected_attach = "The file ATTACH_NAME attached to mail (with subject:SUBJECT) sent by
FROM_USER to TO_USER(S) has an unauthorized file extension"
```

Then in the file `/usr/local/rav8/etc/languages/your_language.equiv`, in the
`_attachment_filter_warning_messages` section use:
`infected_msg = infected_attach`
instead of
`infected_msg = infected_m_english`

Restart **ravmd** with:
`kill -HUP `cat /usr/local/rav8/private/ravmd.pid``
and it's done.

FAQ 7: Stopping all update notifications

Question:
How do I stop all the notifications concerning the update process?

Answer:
To stop all the notifications pertaining to the update process, in `/usr/local/rav8/bin/ravupdate.sh` change this:

```
#Specify when should the administrator be notified by the update process
#VERBOSE="silent"
VERBOSE="noisy"
#VERBOSE="errors"
```

to:

```
#Specify when should the administrator be notified by the update process
#VERBOSE="silent"
#VERBOSE="noisy"
VERBOSE="errors"
```

FAQ 8: Stopping all warning mails

Question:
How do I stop all warning mails?

Answer:
To stop all the warning mails, use in the group' configuration file:

```
warn_sender=never
warn_receivers=never
warn_admin=never
```

Group Declarations

What is a group

A group is a category of users (senders or receivers) that have different email addresses and/or domains, but share the same action parameters for **ravmd**. Using the group feature of **ravmd**, you can use the same characteristics (for the scanning engine or the update process, for instance) and the same actions (filters, warning mails and so on) for users having different email addresses and mail domains. This group structure is very useful in case you use **ravmd** on a large network having hundreds of mailboxes.

In **ravmd** there is one default group, called [global], that initially contains all the users and mail domains protected by RAV AntiVirus for Mail Servers. When you define a new group, their members are taken from the [global] group and included in the new group, for which you should define the new options.

The GLOBAL group

This is the default group, which contains all the users and mail domains that are not defined in the other groups. The [global] group MUST NOT contain any member declaration.

Defining other groups

A new group definition begins with the group name written between "[]". The group definition must be followed by the member declarations (it is mandatory that the members are declared before any other parameters) and the group options.

How do I configure groups

The configuration file must include the [global] group, containing the default options for all mail scanning processes. Besides the [global] group, you can customize **ravmd** by creating additional groups, with different configurations.

When an option is not specified in the [global] group, then the default one is used. For all other groups the undefined options are inherited from the [global] group with the following exceptions:
filter_subject, filter_attachment, filter_content, warn_sender, warn_receivers, warn_admin, do_not_warn, do_not_show, admin_addr, do_not_scan, advertising_msg.

The _include directive

In order to keep the configuration file more readable you can use the **_include** directive to insert other files in the main one. This can be very useful if you have a large number of groups. Defining all the groups in a single configuration file could make very difficult the maintenance of this file. Instead of using a single large configuration file, you can split it in more small files and use this **_include** directive.

For instance, if you want to add a new group called [mygroup] in the main configuration file all you have to do is append it with this line:

```
_include/usr/local/rav8/etc/groups/mygroup_file
```

and define that group in the mygroup_file:

```
[mygroup]
sender = user_1@domain.com
...
```

The following Frequently Asked Questions are inserted in this document in order to clarify some of the most common confusions the existing users of RAV AntiVirus for Mail Servers have been confronted with in aspects pertaining to group configuration.

FAQ 9: Creating different rules for a domain

Question: How do I create different rules for a domain?

Answer: It is possible to define different rules for a domain by creating a group in which you specify `from_host` and/or `to_host`. After creating the group, you have to specify the actions to be performed for that specific group. Here are the steps to be followed for creating different rules for a domain:

- In `ravmd.conf`, in the section `_define_actions` use:

  ```
  act_for_my_group = clean, delete, ..
  ```

- At the end of `ravmd.conf` define the group:
  ```
  [my_domain]
  _include /usr/local/rav8/etc/groups/my_domain
  ```

- Create the file `/usr/local/rav8/etc/groups/my_domain` and edit it:
  ```
  from_host=a.com
  to_host=a.com
  infected_actions=act_for_my_group
  ```

Note: This rule is applying to infected messages, and not to the messages matching the content filtering rules.

FAQ 10: Example on how to set the domains and the IP addresses

Question: I need an example on how to set the domains and the IP addresses.

Answer: In `/usr/local/rav8/etc/ravmd.conf`, in the [`global`] section use:

```
domain = localhost, your.host.com
```

In order to receive the warning mails, in `/usr/local/rav8/etc/groups/global` use:

```
ravms_name = ravms
on_host = your.host.com
smtp_server = ip.address.of.host
```

FAQ 11: Excluding one particular account

Question: I need to exclude 1 account in particular as I receive a daily e-mail of over 60 MB in size and I don't want RAV to process that file. How do I get it done?

Answer: Here is one *temporary* solution for not scanning one particular message received from one sender. Create a group including that specific sender and select not to scan the mail messages:

In `/usr/local/rav8/etc/ravmd.conf`, add at the end of the file:

```
[group1]
_include /usr/local/rav8/etc/groups/group1
```

Then create the file `/usr/local/rav8/etc/groups/group1` and edit it:

```
sender=sender@host.com
do_not_scan=yes
```

Then restart **ravmd** with:

```
kill -HUP `cat /usr/local/rav8/private/ravmd.pid`
```

After receiving the tricky mail message, you have to de-activate the group, commenting in `ravmd.conf` the following lines:

```
 #[group_name]
 #_include …
```

Then restart **ravmd.**

The Advanced Content Filtering feature

What is this

The Advanced Content Filtering is a very powerful feature of **ravmd** that can help you configure the product to better answer your specific needs. Using this feature, you can define filters for different e-mail components (subject, message body or attachment), so that specific actions are taken in case one condition you're specifying is fulfilled. For instance, you can define filters for mail messages containing one specific string in their subject (like "I love you"), having one specific type of attachment or containing in their body one specific string. You can afterwards define specific actions for the mail messages matching these rules.

How does it work

The Content Filtering module of **ravmd** uses POSIX regular expressions (for more info about the regular expression, please refer to the *Regular Expression Declaration* section above) in order to find a pattern in the following e-mail components:

- Subject,
- Message body,
- Attached file names.

The rules you have defined are processed in the following order:

- Subject,
- File names,
- Body.

When defining more rules for one single component, the rules are processed in the order you defined them.

Here are some Frequently Asked Questions the users of RAV AntiVirus for Mail Servers have asked about the Advanced Content Filtering feature of **ravmd**. You can even find an example for configuring a content filter for mail messages containing one specific string.

FAQ 12: Deleting or rejecting double extension files

Question: Has anyone setup the configuration file to delete or reject any attachment with a double extension?
Answer: Please use in `ravmd.conf`:

```
var_regexp = .*\..*\..*
var_action = reject
```

In [`global`]:

```
filter_attachment var_regexp var_action
```

Please note that this restrictive action will also filter the `tar.gz` files, for instance.

FAQ 13: Denying certain types of attachments

Question: Has can I deny certain types of attachments?
Answer: RAV Antivirus for Mail Servers content filtering feature can help you if you want to deny a certain type of attachment from entering your company via email.

In the `_define_regular_expressions` paragraph from `/usr/local/rav8/etc/ravmd.conf` define:

```
file_regex = .*\.exe
```

In the `_define_actions` paragraph from `/usr/local/rav8/etc/ravmd.conf` define:

```
file_action = delete, reject
```

In the `/usr/local/rav8/etc/groups/global` file define (wherever inside the file):

```
filter_attachment file_regexp file_action
```

To add other types of attachments separate them with a | symbol on the `regex` definition.
Example:

```
file_regexp = .*\.((vbs)|(vbe)|(js)|(exe)|(com)|(pif)|(lnk)|(scr)|(bat)|(shs)|(sh))
```

As is the case for each change in the RAV configuration files (`ravmd.conf, global, english, english.equiv`), you must restart **ravmd**:

```
kill -HUP `cat /usr/local/rav8/private/ravmd.pid`
```

FAQ 14: Example of content filtering for Subject

Question: Can I have an example of content filtering for mail messages containing one defined expression?
Answer: Below you can find how to create a content filter for e-mails containing in the Subject field the expression "xxx". In this scenario, the email will be allowed to pass RAV AntiVirus for Mail Servers. The sender and the receiver will not receive any warning mails, but instead a warning mail will be sent to the administrator. Of course, you can customize **ravmd** to act according to your specific needs for all defined groups.

1. Define the expression `xxx` in the `_define_regular_expressions` from `ravmd.conf`:
```
subjxxx_regexp = xxx
```

2. Define the action for `xxx` in the `_define_actions` section from `ravmd.conf`:
```
define xxxsubj_action = ignore
```
This will allow the `xxx` mail to pass the RAV content filter.

To add other words, separate them with a | symbol in the **regex** definition.
Example:
```
subjxxx_regexp = (word1)|(word2)|(word3)
```

3. Save the configuration file.

4. Edit the file `/usr/local/rav8/etc/groups/global`.

5. Specify the administrator's e-mail address for alerting him that an e-mail matching an expression from content filter has been entering in the company:

```
admin_addr=administrator@gecadsoftware.com
```

Please check that the following options is included in the group file (`global`, in this case):
```
warn_admin=always
```

6. Activate the content filter by adding the following line:
```
filter_subject subjxxx_regexp xxxsubj_act
```

7. Save the configuration file.

8. Restart **ravmd** with:
```
kill -HUP `cat /usr/local/rav8/private/ravmd.pid`
```

Explaining the parameters

The parameters included in configuration file of ravmd have different functions. Some are used for specifying the domain parameters, other for specifying the group members, the actions for the scanning engine, the warning messages and the sender of warning messages and so one. All the parameters have been grouped below depending on their function.
Some of these parameters are group-specific, meaning that they are different for each defined group. Other parameters are common for different groups and/or are inherited from the [global] group. This classification is also important for understanding the way **ravmd** is working.

Domain Specification parameters

domain = domains enumeration
Description: This parameter specifies the domains scanned by RAV AntiVirus.

> **Note:** During the evaluation period (60 days) you can set only two domains. Mails for/from other domains are not scanned. They are normally delivered.

This parameter is a global one. It is sufficient to define it in the [global] group.
Important:

> You **must** specify at least one domain name, or else **ravmd** will not start. No default group is provided by RAV.

Example:

```
domain = mail.domain.com, domain.com
domain = second.domain.net
```

IMPORTANT:
For RAV, a 'domain' is the string which follows the '@' symbol in an e-mail address.

> **Note:** In the next version of ravmd (8.3.3) the machine name and the name of the local domain will be automatically added upon installation.

do_not_scan = boolean
Description: This parameter is used to specify if the e-mail files for the current group are scanned or not. This way it is possible to exclude some e-mail addresses and/or domains from the scanning process.
Default value: **No**.

Example:

```
do_not_scan = yes
```

Specifying the group members

sender = user e-mail addresses enumeration
Description: This parameter is used to specify the email addresses of the senders who will be members in the actual group.

Example:

```
sender = user1@domain1.com, user2@domain1.net
sender = user3@domain2.org
```

receiver = user e-mail addresses enumeration
Description: This parameter is used to specify the e-mail addresses of the receivers who will be members in the actual group.

Example:

```
receiver = user4@domain4.com, user5@domain4.org
receiver = user5@domain2.org
```

from_host = host names enumeration
Description: This parameter is used to specify the sending host names who will be members in the actual group.

Example:
```
from_host = mail.domain1.com, domain1.net
from_host = domain2.org
```

to_host = host names enumeration
Description: This parameter is used to specify the receiving host name(s) who will be members in the actual group.

Example:
```
to_host = domain3.com, domain3.net
to_host = domain3.com
```

Defining the actions for the scanning engine

The following parameters are used for defining the actions for the scanning engine: **infected_actions, suspicious_actions.** The `infected_actions` parameter helps you define the actions to be performed when an infected object is found, and the `suspicious_actions` parameter - the actions to be performed when a suspicious object is found.

infected_actions = variable
Description: This parameter is used to specify a variable name defined in the `define_actions` section, which contains the actions to be performed when an infected object is found. If this parameter is not defined then the *reject* action is used for the infected e-mails.

Example:
```
infected_actions = act_for_infected_files
```
Where:
```
act_for_infected_files = clean, move, delete, reject
```
In this example, when an infected file is detected, **ravmd** tries first to *clean* that file. If the cleaning action is completed successfully, **ravmd** moves to the next file. If the cleaning action fails, **ravmd** tries to *move* (*copy* to quarantine and *delete* that file from e-mail) the file. If the moving action is completed successfully, **ravmd** moves to the next file. If the moving action fails, **ravmd** tries to *delete* the file, and so on. The *ignore* and *reject* actions *always* return success.
Note: RAV AntiVirus appends a *reject* action to the actions enumeration by default.

suspicious_actions = variable
Description: This parameter is used to specify a variable name defined in the `_define_actions` section, which contains the actions to be performed when a suspicious object is found. If this parameter is not defined then the *reject* action is used for the suspicious e-mails.

Example:
```
suspicious_actions = act_for_suspicious_files
```
Where:
```
act_for_suspicious_files = move, rename, delete, ignore
```
As in the previous example, when **ravmd** detects a suspicious file, it tries successively to *move, rename* and *delete* that file. If the *move* action fails, **ravmd** tries to *rename* the file (see bellow the description for `rename_ext`). If the *rename* action fails, **ravmd** tries to *delete* the file. If one of the previous actions is successfully completed, **ravmd** moves to the next file. Eventually, if all of the *moving, renaming, deleting* actions failed **ravmd** will *ignore* that file.

Specifying the warning messages

The following parameters help you define the messages used to create warning notifications in different circumstances (virus found, subject filter match, attachment filter match, content filter match).
If some warning messages are not specified for the [`global`] group, then **ravmd** uses a default string (`<<not defined>>`). For all the other groups, the warning messages are inherited from the [`global`] group.

_virus_warning_messages
Description: This parameter is used to specify the messages used to create the warning notifications when a virus is found in the e-mail.

_subject_filter_warning_messages
Description: This parameter is used to specify the messages used to create the warning notifications when the content filtering is enabled and the e-mail subject matches one of the defined rules.

_attachment_filter_warning_messages
Description: This parameter is used to specify the messages used to create the warning notifications when the content filtering is enabled and an attached file name matches one of the defined rules.

_content_filter_warning_messages
Description: This parameter is used to specify the messages used to create the warning notifications when the content filtering is enabled and the e-mail body or one of the attachments contains a string matching one of the defined rules.

warning_mail_subj = variable
Description: This parameter is used to specify the subject used in the notification e-mail.

Example:

```
warning_mail_subj = wm_sbj
```

Note: In the *Warning Mails Message Declarations Section* you must be specify:

```
wm_sbj = "RAV AntiVirus scan results."
```

infected_msg = variable
Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when an *infected* file is detected.

Example:

```
infected_msg = wm_inf_msg
```

Where:

```
wm_inf_msg =  "The file ATTACH_NAME attached to mail (with subject:SUBJECT)
               sent by FROM_USER to TO_USER(S) is infected with virus: VIRUS_NAME."
```

suspicious_msg = variable
Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when a *suspicious* file is detected.

Example:

```
suspicious_msg = wm_sus_msg
```

Where:

```
wm_sus_msg =  "The file ATTACH_NAME attached to mail (with subject:SUBJECT)
               sent by FROM_USER to TO_USER(S) contains suspicious code."
```

ignored_msg = variable
Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when an *infected/suspicious* mail file is ignored.

Example:

```
ignored_msg = wm_ign_msg
```

Where:

```
wm_ign_msg =  "The file was ignored because all previous actions failed.
              It is highly recommended not to use this file."
```

rejected_msg = variable
Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when an *infected/suspicious* mail file is rejected.

Example:
```
rejected_msg = wm_rej_msg
```
Where:
```
wm_rej_msg = "The e-mail was rejected because it contains dangerous code."
```

cleaned_msg = variable
Description: This parameter is used to specify the *body* of the info e-mail sent by **ravmd** when a file is *cleaned*.

Example:
```
cleaned_msg = clean_ok
```
Where:
```
clean_ok = "The file was successfully cured by RAV AntiVirus."
```

moved_msg = variable
Description: This parameter is used to specify the *body* of the info e-mail sent by **ravmd** when a file is *moved*.

Example:
```
moved_msg = move_ok
```
Where:
```
move_ok = "The file was successfully moved to quarantine with name: QUARANTINE_NAME."
```

copied_msg = variable
Description: This parameter is used to specify the *body* of the info e-mail sent by **ravmd** when a file is *copied*.

Example:
```
copied_msg = copy_ok
```
Where:
```
copy_ok = "The file was successfully copied to quarantine with name: QUARANTINE_NAME."
```

deleted_msg = variable
Description: This parameter is used to specify the *body* of the info e-mail sent by **ravmd** when a file is *deleted*.

Example:
```
deleted_msg = delete_ok
```
Where:
```
delete_ok = "The file was successfully deleted by RAV AntiVirus."
```

renamed_msg = variable
Description: This parameter is used to specify the *body* of the info e-mail sent by **ravmd** when a file is *renamed*.

Example:
```
renamed_msg = rename_ok
```
Where:
```
rename_ok = "The file was successfully renamed by RAV AntiVirus."
```

saved_inf_msg = variable
saved_sus_msg = variable

Description: These parameters are used to specify the string used in the notification e-mail when the infected/suspicious file is saved to quarantine.

Example:

```
saved_inf_msg = save_ok
saved_sus_msg = save_ok
```

Where:

```
save_ok = "The e-mail file SAVED_FILE_NAME was saved to quarantine."
```

cannot_clean_msg = variable

Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when a file *cannot be cleaned*.

Example:

```
cannot_clean_msg = not_cleaned
```

Where:

```
not_cleaned = "Cannot clean this file."
```

cannot_move_msg = variable

Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when a file *cannot be moved*.

Example:

```
cannot_move_msg = not_moved
```

Where:

```
not_moved = "Cannot move this file."
```

cannot_copy_msg = variable

Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when a file *cannot be copied*.

Example:

```
cannot_copy_msg = not_copied
```

Where:

```
not_copied = "Cannot copy this file."
```

cannot_delete_msg = variable

Description: This parameter is used to specify *body* of the warning mail sent by **ravmd** when a file *cannot be deleted*.

Example:

```
cannot_delete_msg = not_deleted
```

Where:

```
not_deleted = "Cannot delete this file."
```

cannot_rename_msg = variable

Description: This parameter is used to specify the *body* of the warning mail sent by **ravmd** when a file *cannot be renamed*.

Example:

```
cannot_rename_msg = not_renamed
```

Where:

```
not_renamed = "Cannot rename this file."
```

cannot_save_inf_msg = variable
cannot_save_sus_msg = variable
Description: These parameters are used to specify the *string* used in the notification e-mail when the infected/suspicious file cannot be saved to quarantine.

Example:
```
cannot_save_inf_msg = not_saved
cannot_save_sus_msg = not_saved
```
Where:
```
not_saved= "The infected e-mail file cannot be saved to quarantine."
```

Miscellaneous parameters

These parameters are inherited by additional groups from the [global] group.

warn_header_msg = variable
Description: This parameter is used to specify the text used as a header in the notification e-mail.

Example:
```
warn_header_msg = notification_header
```
Where:
```
notification_header = "This message is automatically generated by RAV AntiVirus."
```

warn_footer_msg = variable
Description: This parameter is used to specify the text used as a footer in the notification e-mail.

Example:
```
warn_footer_msg = notification_footer
```
Where:
```
notification_footer = "The e-mail scanned was received from: HEADER_RECEIVED."
```

warn_txt_msg = variable
Description: This parameter is used to specify the text appended after the default one in the warn.txt file.

Example:
```
warn_txt_msg = append_to_warn_txt
```
Where:
```
append_to_warn_txt = "Please contact your system administrator for more information. "
```

For more information about the warn.txt, please read the FAQ 3.

charset = string
Description: This parameter is used to specify the value for the **charset** field used in the MIME header of the virus notification mails. If the warning mails contain strings with a character encoding system different from ASCII you should specify the respective encoding using the **charset** parameter.
Default value is US-ASCII.

Example:
```
charset = iso-2022-jp
```

custom_msg = number
The notification e-mails are created using the strings defined by the user in the _define_warning_mail_messages section and some RAV-related information (always added during the evaluation period). A notification message looks as follows:

RAV AntiVirus for OSTYPE version: x.x.x (snapshot-yyyymmdd)
Copyright since 1995 GeCAD The Software Company. All rights reserved.
X more days to evaluate. (or: Registered version for N domain(s).)
Running on host: HOSTNAME

The file ATTACHED_NAME attached to mail (with subject:SUBJECT) sent by FROM_USER to TO_USER(S) is infected with virus: VIRUS_NAME. The file was successfully deleted by RAV AntiVirus.

Scan engine 8.7 () for i386.
Last update: Sat, 25 May 2002 19:14:44 +0300
Scanning for 67940 malwares (viruses, trojans and worms).

To get a free 60-days evaluation version of RAV AntiVirus v8 (yet fully functional) please visit:
http://www.ravantivirus.com

The macros are replaced with their corresponding values. In the registered version of RAV AntiVirus for Mail Servers, all the RAV-related information can be omitted, except for the first header line.

In the registered version the warning mails can be customized. Default value is 127 (use all RAV information).
= 0 - the notification e-mail will consist of the first header line on notification e-mail
+ 1 - add "Registered version ..."
+ 2 - add "Running on host ..."
+ 4 - add "Scan engine ..."
+ 8 - add "Last update ..."
+16 - add "Scanning for ..."
+32 - add "Scanning for ..."
+64 - add "Copyright ..."

Example:

custom_msg = 8

```
RAV AntiVirus for OSTYPE version: x.x.x (snapshot-yyyymmdd)

The file ATTACHED_NAME attached to mail (with subject: SUBJECT) sent by FROM_USER to
TO_USER(S) is infected with virus: VIRUS_NAME.
The file was successfully deleted by RAV AntiVirus.

Last update: Thu Oct 25 07:55:57 2001
```

max_processes = number
Description: This parameter is used to specify the maximum number of **ravmd** scanning processes running at the same time. The value must be in the [**1**-**128**] interval. Default value is **24**.

Example:

```
max_processes = 80
```

timeout_per_file = number
timeout_per_mega = number
Description: These parameters are used to specify the maximum time in seconds that a scanning process can spend on a file. The total timeout is computed using the following formula:

timeout_per_file + timeout_per_mega * filesize/1Mb

The values must fall between **10**-**600** (for **timeout_per_file**), respectively **5**-**600** (for **timeout_per_mega**).
Default values: **60** / **30**.

Example:

```
timeout_per_file = 120
timeout_per_mega = 25
```

save_infected = boolean
save_suspicious = boolean
Description: These parameters are used to specify if infected/suspicious email files are saved to the local disk before executing any action. You can set these options to **Yes** or **No**. The default value is **Yes**. Please note that the infected/suspicious messages will be placed in the quarantine regardless of the defined `infected_actions` and `suspicious_actions`.

Example:
```
save_infected = no
save_suspicious = yes
```

quarantine = string
Description: This parameter is used to specify the directory where the e-mails are saved.
Default value is: **/var/spool/rav/quarantine**

Example:
```
quarantine = /tmp/rav/quarantine
```

Scan engine configuration

use_heuristics = boolean
Description: This parameter is used to control the heuristic methods for detecting new viruses.
Default value is **Yes**.

Example:
```
use_heuristics = no
```

scan_packed_executables = boolean
Description: This parameter is used to control the scanning process for packed executables like VVPACK, UCEXE, PEPACK, etc.
Default value is **Yes**.

Example:
```
scan_packed_executables = no
```

scan_archives = boolean
Description: This parameter is used to control scanning in archive files, like **ZIP**, **ARJ**, **RAR**, **LHA**, **LHZ**, **ACE**, **CAB**, etc. If the Boolean value is set to **Yes**, **ravmd** will scan inside archives. If the Boolean value is set to **No**, **ravmd** will not scan inside archives.
Default value is **Yes**.

Example:
```
scan_archives = yes
```

rename_ext = string
Description: This parameter is used to specify the *extension* used to *rename* the infected/suspicious files. This function is necessary for preventing inexperienced users from accessing by accident infected/suspicious files moved to Quarantine.
Default extension is: _??

Example:
```
rename_ext = _??
```

scan_ext = string
smart_scan = boolean
Description: These parameters are used to specify the scanning modes for **ravmd**. The options are:

- *scan all files*,
- *let RAV decide* what files to scan,
- scan only files with the listed extensions.

Note: If **scan_ext** and **smart_scan** *are not defined* then **ravmd** will scan all files. If **scan_ext** is defined then this option is selected regardless of the **smart_scan** value. Please be careful with **scan_ext** declaration syntax.
By default, the smart scanning is enabled.

Example:

```
scan_ext = .EXE\".COM\".OVL\".PRG\".SCR\".VXD\".BIN\".BOO\".TD0\".XL?\".DO?\"
.MDB\".PPT\".VBS\".BAT\".SAM\".JS\".HT*\".DLL\".POT\".DRV\".IMG\".OVR\".386\".PIF\".PDF
```

Specifying the sender of the warning mails

ravms_name = string
on_host = string
smtp_server = string
smtp_port = number

Using these parameters you can define the e-mail address for the sender of the warning mails. Default values are specified and they will probably work. Define these fields only if warning mails are NOT sent when a virus is found or if you want to use a different account instead of **ravms**. Specify **smtp_server** IP address only if that machine is behind a firewall and **ravmd** can't get its host name.
Default values:

ravms_name = ravms
on_host = official host name
smtp_server = official host IP address
smtp_port = 25

Example:

```
ravms_name = ravms
on host = host_name_where_ravms_is_defined
smtp_server = 127.0.0.1
smtp_port = 25
```

no_subject = string
Description: This parameter is used to specify the string replacing the `SUBJECT` macro in the warning mail if **ravmd** does not find a valid subject in the infected e-mail.
Default value: **--no subject found--**.

Example:

```
no_subject = "original e-mail didn't contain any subject field"
```

mailer_daemon = string
Description: This parameter is used to specify the name that will replace the `FROM_USER` macro when the e-mail sender is **<>**.
Default value is **--unknown--**.

Example:

```
mailer_daemon= "MAILER-DAEMON"
```

RAV logging system

When **ravmd** starts it will log some information in the system mail info file (using `syslog`) then it switches to the internal log. By default the log files are created in: `/var/spool/rav/log/`.

It is possible to use a different log file (with different options) for every group. To do this you must declare one of the following log options in that group. If a group doesn't contain any log options, then the global log data are used.
This feature is available starting with 8.3.3 version of **ravmd**.

Here are the parameters used for RAV logging system:

log_file_name = string
Description: This parameter is used to specify the full path and the name of the log file. Any other values prevent RAV from starting.
The default value for the log file name is:
`log_file_name = /var/spool/rav/log/group_name` (for the `group_name` group)

Example:
```
log_file_name = /var/spool/rav/log/global
log_file_name = /var/spool/rav/log/my_group
```

log_max_length = {number}(Kb|Mb)
Description: This parameter is used to specify the maximum size for the log file.
Default value is 500Kb.
Accepted values are: 10-1000Kb and 1-5Mb.

Example:
```
log_max_length = 2Mb
```

log_rotate_after = {number}(m|h|d)
Description: This parameter is used to specify the period of time elapsing before creating a new log file.
Default value is 6h (hours).
Accepted values are: 10-60m - minutes (23m is rounded at 20m, 25m at 30m), 24h - hours or 1-30d - days.

Example:
```
log_rotate_after = 1d
```

log_delete_after = {number}(h|d|m)
Description: This parameter is used to specify the period of time elapsing before deleting log files older than the specified period of time.
Default value is 7d days.
The accepted values are: 1-24h - hours, 1-30d - days or 1-12m - months

Example:
```
log_delete_after = 15d
```

log_use_zip = boolean
Description: This parameter is used to specify if the log files should be arhived (using the zlib library) or not.
Default value is **Yes**.
The accepted values are: **Yes** or **No**

Example:

```
log_use_zip = yes
```

log_level = number
Controls the logging level. Default value is 511 (use all RAV information).
= 0 - No log information
+ 1 - add error messages (i.e. can't fork, error reading from socket, etc.)
+ 2 - add mail file name
+ 4 - add mime part scanned
+ 8 - add final scan result
+ 16 - add actions taken during scanning
+ 32 - add the e-mail addresses of the sender and the first receiver
+ 64 - add the group name matched
+128 - add information generated by the external triggered update
+256 - add LICENSE LIMIT warnings

Example:

```
log_level = 255
```

Group specific parameters

The following parameters must have **different** values for every defined group. If these parameters are not defined for each group, their values are **NOT** copied from the [`global`] group. The default values are specified for each of these parameters in the following section.

filter_subject variable_1 variable_2
Description: This parameter is used to filter the e-mail subject.
variable_1 is a regular expression defined in the *Regular Expression Declarations Section*.
variable_2 is a variable defined in the *Action Declarations Section*. If this parameter is not defined then the subject filter is disabled.

Example:

```
filter_subject subj_regexp subj_actions
```

Where:

```
subj_regexp = I love you
subj_actions = reject
```

Using this rule, e-mails with string "I love you" contained in the subject will be rejected. Note that you can use a regular expression here instead of a simple string.

filter_attachment variable_1 variable_2
Description: This parameter is used to filter the names of the e-mail attachments.
variable_1 is a regular expression defined in the *Regular Expression Declarations Section*.
variable_2 is a variable defined in the *Action Declarations Section*. If this parameter is not defined then the filter for attachment names is disabled.

Example:

```
filter_attachment file_regexp file_actions
```

Where:

```
file_regexp = .*\.((vbs)|(exe)|(com))
file_actions = delete, reject
```

This filtering rule deletes all the attached files with extension ".vbs", ".exe" or ".com" from an e-mail.
If a file cannot be deleted then the whole e-mail will be rejected.

filter_content variable_1 variable_2
Description: This parameter is used to filter the e-mail body and the contents of the attachment.

variable_1 is a regular expression defined in the *Regular Expression Declarations Section*.
variable_2 is a variable defined in the *Action Declarations Section*. If this parameter is not defined then the message content filter is disabled.

Example:

```
filter_content body_regexp_1 body_actions_1
filter_content body_regexp_2 body_actions_2
```

Where:

```
body_regexp_1 = confidential.*document
body_regexp_2 = .*salaries.*
body_actions_1 = delete, reject
body_actions_2 = copy, ignore
```

In this case the filter has to search for both the regular expressions specified. Every rule has a priority associated. The rule with the highest priority is the first specified. This rule has a priority of 1. The next rules receive a lower priority (2...n). In our example, RAV AntiVirus will start searching for both regular expressions in the same time. If the first match is found with the expression of higher priority (`body_regexp_1`) the search stops and the actions from `body_actions_1` are executed. If the first match is found with the expression of lower priority (`body_regexp_2`) the search will continue for the rest of the e-mail in order to verify matches with the higher priority expression. If the rest of the e-mail contains a match with `body_regexp_1` RAV AntiVirus will execute `body_actions_1` and ignore the first match. If the rest of the e-mail does not contain a match with `body_regexp_1` then RAV will execute `body_actions_2`.

`warn_sender = keyword enumeration`
Description: This parameter is used to send notifications to the e-mail sender.

`warn_receiver = keyword enumeration`
Description: This parameter is used to send notifications to the e-mail receivers.

`warn_admin = keyword enumeration`
Description: This parameter is used to send notifications to the administrators.

You have to specify *who* is warned by RAV and *when*. If one of these parameters is not defined then the respective user category will **not** receive notifications. The valid keywords are:

Keyword	Meaning
`found_virus`	Send alert to the defined recipient(s) when a virus is found.
`found_subject`	Send alert to the defined recipient(s) when the subject matches a content filtering rule
`found_attach`	Send alert to the defined recipient(s) when an attached file name matches a content filtering rule.
`found_content`	Send alert to the defined recipient(s) when the mail body contains a string matched by a rule.
`always`	Send alert to the defined recipient(s) in all the above-mentioned situations.
`never`	Do not send any warning mail.

Please note that these values are not inherited from the [`global`] group. This way you can specify different notification policies for different groups.
Default value: no warning mails are sent.

Example:

```
warn_sender = found_virus, found_subject, found_attach, found_content
warn_receivers = found_virus
warn_admin = always
```

do_not_warn = user e-mail address enumeration
Description: This parameter is used to specify the e-mail address that will not be notified.

do_not_show = user e-mail address enumeration

Why is this parameter required?
Sometimes you might want a user not to be notified or his e-mail address not be displayed in the warning mails. If this is the case, these options will help you solve the problem.

Note that only the receiver's e-mail address is compared against the specified e-mail address.
These parameters have no default values (no comparisons will be made).

Example:

```
do_not_warn = user3@domain2.org
do_not_show = user1@domain1.com, user2@domain1.com
```

admin_addr = e-mail addresses enumeration
Description: This parameter is used to specify the *e-mail addresses* of the administrators that will be notified when an infected or suspicious file has been detected. This warning mail contains messages created using the strings specified for each situation. This parameter has no default value.

Example:

```
admin_addr = postmaster@domain1.com, postmaster@domain2.net, user1@domain1.com
admin_addr = ravmails@ravantivirus.com
```

Note: The option of forwarding the e-mail to ravmails@ravantivirus.com is highly recommended. This will help the RAV Development Team to determine the level of spreading of this virus and new viruses found in the world or potential detection problems. Also the RAV Technical Support Team may diagnose potential problems for the end-user, such as old updates of the virus signatures database or old product kits. In this case, the end-user will be informed about the best solution for solving his problem.
GeCAD Software treats each mail in strict confidence.

advertising_msg= variable
Description: Using this parameter you can append a text message to every e-mail scanned. If this option is used, **ravmd** will add your personal message to the following text (that is included by default and cannot be changed or removed): "This e-mail was scanned by RAV AntiVirus!". Please note that the e-mail length will grow if you will use this feature.
This parameter must be declared in the group file or even in the [`global`] group. The value for **variable** must be declared in the language file and the `language.equiv` file must be included in the group file.
Please note that RAV cannot append the text to any e-mail due to the MIME format of the mail.
`advertising_msg` is not supported by RAV Antivirus for CommuniGate PRO.

Example:
In the group file:

```
_include /usr/local/rav8/etc/languages/english.equiv
advertising_msg = my_advertising
```

Where:

```
my_advertising = "COMPANY NAME maintains e-mail messages virus free"
```

is declared in `/usr/local/rav8/etc/languages/english`

update_executable= string
Description: This parameter is used to specify the name of an executable file used by **ravmd** to start the update process. You must specify the full path to the executable file. The update process is started only if **ravmd** is receiving an update e-mail sent by RAV Team. If you want to receive such e-mails, please send a request to:
<mailto:updates-l-subscribe@lists.ravantivirus.com>

Default value: **/usr/local/rav8/bin/ravupdate.sh**

Example:

```
update_executable = null
```

This line is disabling the **update executable** feature:

```
update_executable = /home/ravms/ravupdate.sh
```

This line executes the specified script file every time an update e-mail is processed by **ravmd**.

Index of Group Parameters

save_suspicious
quarantine

Scan engine

use_heuristics
scan_archives
scan_packed_executables
rename_ext
scan_ext
smart_scan

Specifying the sender of the warning mails

ravms_name
on_host
smtp_port
smtp_server

no_subject
mailer_daemon

RAV logging system

log_file_name
log_max_length
log_rotate_after
log_delete_after
log_use_zip
log_level

Specify actions

infected_actions
suspicious_actions

Groups specific parameters

filter_subject
filter_attachment
filter_content

warn_sender
warn_receivers
warn_admin
do_not_warn
do_not_show

admin_addr

update_executable

NAME

ravmd - **rav m**ail scanning **d**aemon

SYNOPSIS

/usr/local/rav8/bin/ravmd **[-cdfhrtvug] [--config=config_file] [--dump_conf=config_file] [–foreground] [--help] [--rav8path=rav8basedir] [--testconf=config_file] [--version] [--user=user_name] [–group=group_name]**

DESCRIPTION

DEFINITIONS

`filter client'

It is a program that resides into **/usr/local/rav8/bin/** directory and its name depends on your MTA. Its function is to hook the MTA's e-mail flux and pass each e-mail to ravmd for scanning. Depending by response from ravmd the `filter client' will discard or transmit that e-mail.

The RAV AntiVirus mail scanning daemon is powered by the platform independent RAV engine, so it can detect and clean all viruses detected by this (linux, win, macro, dos, trojan, hoax, etc.). The program can scan e-mail files in MIME format containing attachments encoded with: base64, quoted-printable, uuencode, 7bit, 8bit. RAV also supports e-mail content filtering for the e-mail subject, attachment file names and message body.

When **ravmd** starts it gets the configuration from **/usr/local/rav8/etc/ravmd.conf** file. If there are some errors (e.g. missing or bad format of these files) ravmd program exits with non zero status. Else it starts in background and binds an UNIX socket (/usr/local/rav8/private/_ravmdcom) and listen it for **`filter clients**' queries. When a **`filter client**' connects to this socket ravmd forks and the child will process **`filter client**' commands.

The daemon uses the system mail info log file for logging. The following command should display that file:

```
cat /etc/syslog.conf |grep -e ^[^#].*mail\.[^acdenw] | \
        awk '{print $2}'
```

If you would like to perform periodic updates to the RAV AntiVirus engine and virus signature files, you should use a scheduling daemon (**cron, fcron, ucron**...) to execute the **/usr/local/rav8/bin/ravupdate.sh** script. Please modify that script file to fit your configuration. We recommend configuring the scheduling process to take place once or twice an hour.

Example for fcron:
su
fcrontab -e
Insert the following line to run ravupdate.sh every 30 minutes:
@ 30 /usr/local/rav8/bin/ravupdate.sh

Example for cron:
su
crontab -e
Insert the following line to run ravupdate.sh every 30 minutes:
***/30 * * * * /usr/local/rav8/bin/ravupdate.sh**

OPTIONS

Arguments are mandatory for both long and sort options.

-c, --config=config_file
Use the config_file instead of **/usr/local/rav8/etc/ravmd.conf**

-d, --dumpconf=config_file
Print the configuration from config_file to stdout.
-f, –foreground
Run the daemon in fo reground instead of background.
-h, --help
Display the help screen.
-r, –rav8path=rav8basedir
Full path to RAV directory.
-t, –testconf=config_file
Test the configuration from config_file.
-v, –version
Display ravmd version.
-u, --user=user_name
-g, --group=group_name

Use **user_name** and **group_name** as real user and real group for ravmd processes. By default the current user uid and gid are used. For security reasons, when executing ravmd as root, it is highly recommended to use these options in order to drop the superuser privileges to an unprivileged user.

FILES

/etc/rc.d/init.d/rav /etc/rc.d/init.d/ravmail
Depending your Linux distribution these script files *may* exist or *not*. If exists then they manage starting, stopping and report status of ravmd processes.
/usr/local/rav8/etc/ravmd.key
This file contains an initialization key for RAV engine. During the evaluation period ravmd will scans two domains for sixty days beginning the day you've installed ravmd. For registered users this file contains a unique RAV key string.
/usr/local/rav8/etc/ravmd.conf
With this file you manage the **ravmd** behavior. See **ravmd.conf(5)** manual for more information.

EXIT STATUS

On error returns non-zero, else returns zero.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd.conf(5), ravav(8), sysklogd(8)

NAME

ravav - RAV AntiVirus command line version

SYNOPSIS

ravav [OPTION ...] *TARGET [TARGET ...]*

DESCRIPTION

ravav is a command line antivirus. It is able to detect and remove known and unknown computer viruses, trojans and worms. It uses the same engine as all the RAV products, with daily updates available on our web site: **http://www.ravantivirus.com**

OPTIONS

The following options are recognized by **ravav**:

-h, --help
Print the help screen to the console.
-v, –version
Display ravav version.
-V, --virlist
Print viruses list.
--license=AuthorizationCode
Licenses ravav using the Authorization Code provided by your supplier, as a string.

-u, --update=engine|full
Start RAV AntiVirus updating process.
--host=host_name
Download files from host_name. Current ftp sites are:
ftp.us.ravantivirus.com (Default value)
ftp://ftp.ro.ravantivirus.com
ftp.download.ravantivirus.com

--ravpath=ravbasedir
full path to RAV directory. Default value for **ravbasedir** is **/usr/local/rav8**
--hostpath=dirname
RAV path on ftp host. Default value for **dirname** is **/pub/GeCAD/rav8**
--ftpuser=username
The user name used on the ftp connection. Default value for **ftpuser** is **ftp**
--ftppass=password
The password used on the ftp connection. Default value for **ftppass** is **rave@**

--all
Scan all files (default).
--smart
Use smart scan mode.
--ask
Ask for user choice when scanning.
--clean
Clean viruses from infected objects.
--delete
Delete infected and suspicious files
--copy
Copy infected/suspicious objects to quarantine.
--move
Move infected/suspicious objects to quarantine.
--rename
Rename infected/suspicious using the extension defined with: -R option.

-A, --archive
Scan inside archives.
-M, --mail
Scan mail files.
-H, --heuristics=on|off
Control the heuristic scanning.
-I, –integrity_check=on|off
Enable/disable integrity checker.
-Q, --quarantine=dir_name
Specify quarantine path. Default dir_name=**~/.rav8/quarantine**.
-R, --rename_ext=extension
Specify the extension used for rename action. Default extension=_**??**.

-l, –listall
List all scanned files.
--report=filename
Report names of viruses found in the filename file.
--rptall
Include all scanned files in the filename file.
--append
append information to the filename file.

EXIT STATUS

ravav returns the status of the last action it does:

1	The file is clean.
2	Infected file.
3	Suspicious file.
4	The file was cleaned.
5	Clean failed.
6	The file was deleted.
7	Delete failed.
8	The file was successfully copied to quarantine.
9	Copy failed.
10	The file was successfully moved to quarantine.
11	Move failed.
12	The file was renamed.
13	Rename failed.
20	No TARGET is defined.
30	Engine error.
31	Syntax error.
32	Help message.
33	Viruses list.
34	The updating process was successfully completed.
35	The updating process failed.
36	Already updated.
37	The licensing process was successfully completed.
38	The licensing process failed.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8)

NAME

ravcgate - external filter for CommuniGate Pro

SYNOPSIS

ravcgate

DESCRIPTION

This is the external filter program executed by CommuniGate Pro server in order to scan all messages for virus protection and/or content filtering. The program runs as a `filter client' for RAV AntiVirus mail scanning daemon (ravmd).

USAGE

Connect to CommuniGate Pro administration web interface.

Open **SETTINGS ->General->Helpers**. In Program Path text entry from Content Filtering section write **ravcgate** (in the CommuniGate Pro base directory must be a symbolic link named **ravcgate** to **/usr/local/rav8/cgatepro/ravcgate file**).

Check the Content Filtering checkbox. Click the Update button.

Read instructions from:
http://www.stalker.com/CommuniGatePro/VirusScan.html#Scanning
to learn how to make a rule for an ExternalFilter.

RAV AntiVirus can scan multiple files in parallel. To activate this facility you must select more processors in **SETTINGS ->Queue->Message Enqueuer**.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8), ravmd.conf(5), ravav(8)

NAME

ravcourier - external filter for the Courier MTA

SYNOPSIS

ravcourier

DESCRIPTION

This is the external filter program executed by the Courier MTA in order to scan all messages for virus protection and/or content filtering. The program runs as a `filter client' for RAV AntiVirus mail scanning daemon (ravmd).

USAGE

Install the RAV filter:
/usr/lib/courier/sbin/filterctl start ravcourier

Uninstall the RAV filter:
/usr/lib/courier/sbin/filterctl stop ravcourier

Start global mail filtering for Courier:
/usr/lib/courier/sbin/courierfilter start

Filter configuration
When **ravcourier** is started it reads the runtime parameters from the file:
/usr/local/rav8/etc/ravcourier.conf
If the configuration file doesn't exist then the default values will be used.

nthreads - working threads started in parallel. Accepted valued: between **1** and **128**.
Default: **20**.

max_connections - maximum accepted connections. Accepted value: between **1** and **256**.
Default: **100**.

use_allfilters - install the filter in the '**allfilters**' directory. If you specify no here then the filter will be installed in the '**filters**' directory.
Default: **yes**.

filters_dir - the full path to the filters root directory.
Default: **/usr/lib/courier/var**.

queue_dir - the full path to the queue directory.
Default: "". This parameter must be used only if Courier is configured to use relative paths to its queue files.

If you want to modify some of these parameters then you have to edit the **/usr/local/rav8/etc/ravcourier.conf** file and reinstall the filter:
/usr/lib/courier/sbin/filterctl stop ravcourier
/usr/lib/courier/sbin/filterctl start ravcourier

FILES

/usr/local/rav8/etc/ravcourier.conf

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8), ravmd.conf(5), ravav(8)

NAME

ravdmail - external filter for **DMail MTA**

SYNOPSIS

ravdmail

DESCRIPTION

This is the external filter program executed by **DMail** server in order to scan all messages for virus protection and/or content filtering. The program runs as a `**filter client**' for RAV AntiVirus mail scanning daemon (**ravmd**).

USAGE

In the DMail configuration file the following line must by added: **virus_robot /usr/local/rav8/bin/ravdmail** and then DMail has to be restarted.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8), ravmd.conf(5)

NAME

ravexim - RAV AntiVirus `filter client' for **exim MTA**

SYNOPSIS

/usr/local/rav8/bin/ravexim $sender_address $recipients

DESCRIPTION

The program will be executed by the exim system filter file each time a new message is processed. It is used to scan all incoming/outgoing e-mails for virus protection and/or content filtering. It runs as a '**filter client**' for RAV AntiVirus mail scanning daemon (**ravmd**).

USAGE

The following parameters are **exim** internal variables:

$sender_address
: The e-mail sender address.

$recipients
: A list with the recipient addresses of a message.

EXIT CODES

The program returns some of the exit codes defined in the **sysexits.h** file:

0 - EX_OK
: Successful termination.

64 - EX_USAGE
: The command line parameters are not specified correctly.

66 - EX_NOINPUT
: Cannot open a temporary file.

69 - EX_UNAVAILABLE
: The sendmail executable is not found. You have to create a symbolic link to the exim executable: ln -s /usr/exim/bin/exim /usr/sbin/sendmail

71 - EX_OSERR
: Cannot create a pipe or a new process.

73 - EX_CANTCREAT
: Cannot create a temporary file.

74 - EX_IOERR
: An error has occurred on a read/write operation.

75 - EX_TEMPFAIL
: Temporary error: **ravmd** is not running or it doesn't temporarily accept connections.

77 - EX_NOPERM
: The e-mail file is infected and its delivery is denied.

All other exit codes are those returned by the command:
"sendmail -oMr mail-ok -i -f $sender $recipients"

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd.conf(5), ravmd(8)

NAME

ravpostfix - RAV AntiVirus '**filter client**' for postfix

SYNOPSIS

/usr/local/rav8/bin/ravpostfix **[-hvSsCctflug] [--help] [–version] [--server_ip=IP_ADDRESS] [--server_port=PORT] [--client_ip=IP_ADDRESS] [--client_port=PORT] [--timeout=SECONDS] [–foreground] [--log_level=NUMBER] [--user=user_name] [--group=group_name]**

DESCRIPTION

This is an external filter program integrated with **postfix** MTA and used for scanning all incoming/outgoing messages for virus protection and/or content filtering. It implements the protocol described in the **FILTER_README** file from **postfix** snapshot >= 20000531. The program runs as **filter client** for RAV AntiVirus mail scanning daemon (**ravmd**).

USAGE

-h, --help
Displays the help screen.
-v, –version
Displays **ravpostfix** version.
-S, --server_ip=IP_ADDRESS
The computer IP address where **ravpostfix** is running. Default: **127.0.0.1**
-s, --server_port=PORT
Inet port on server_ip where ravpostfix is listening. Default: **10025**
-C, --client_ip=IP_ADDRESS
The computer ip address where second postfix SMTP server is running. Default: 127.0.0.1
-c, –client_port=PORT
Inet port where second **postfix** SMTP server is listening. Default: **10026.**
-t, –timeout=SECONDS
The timeout in seconds used by ravpostfix for SMTP communication. Default: **120s.**
-f, –foreground
Runs the daemon in foreground instead of background (background is default value).
-l, –log_level=0|1|2
Control the logging details. Default: **1**.
-u, --user=user_name
-g, --group=group_name
Use **user_name** and **group_name** as real user and real group for ravpostfix processes. By default the current user uid and gid are used. For security reasons, when executing ravpostfix as root, it is highly recommended to use these options in order to drop the superuser privileges to an unprivileged user.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8), ravmd.conf(5), ravav(8)

NAME

ravqmail - RAV AntiVirus `**filter client**' for qmail

INSTALL

In order to install ravqmail you must have installed **qmail**-1.03 or newer in directory **/var/qmail**. You must stop **qmail-send** from sending it the **SIGTERM** signal. After this you must rename the original **qmail-queue** to **qmail-queue.orig** and make a symbolic link named **qmail-queue** to **ravqmail**:

mv /var/qmail/bin/qmail-queue /var/qmail/bin/qmail-queue.orig
ln -s /usr/local/rav8/bin/**ravqmail** /var/qmail/bin/**qmail-queue**

USAGE

After installing **ravqmail** you must start **ravmd** daemon. Then you can start **qmail** system.

UNINSTALL

In order to uninstall **ravqmail** you must stop qmail-send from sending it the **SIGTERM** signal. After this you must remove the **link** file **qmail-queue** and copy the original **qmail-queue.orig** in **qmail-queue**:

rm -f /var/qmail/bin/**qmail-queue**
mv /var/qmail/bin/**qmail-queue.orig** /var/qmail/bin/**qmail-queue**

Please read section **NOTE** bellow.

FILES

After the installation process /usr/local/rav8/bin/**ravqmail** must have the same owner, group and permissions as **/var/qmail/bin/qmail-queue.orig.** Directories **/usr/local/rav8/private/** and **/var/spool/rav/qmail/** must have the same owner and group as **/var/qmail/bin/qmail-queue.orig** and **0750** permissions set.

EXIT CODES

The same as **qmail-queue(8).**

NOTES

There are some programs or scripts automating the starting and stopping processes. If you are using one of them it might be wrong to send **SIGTERM** signal to **qmail-send** instead of using them to stop **qmail-send**, since these programs might respawn the **qmail-send** during the installation process therefore your **qmail** won't be able to placed e-mails in its queue.

When an e-mail is handled by **qmail** first it is put in queue by using the **qmail-queue** program. Because in the installation process the original **qmail-queue** is replaced with **ravqmail** the e-mail is first scanned and after that ravqmail pass the e-mail to the **qmail-queue.orig** and replies with **qmail-queue.orig** exit code.

If **ravmd** is not started or **ravqmail** can't communicate with **ravmd**, it doesn't pass the e-mail to the original **qmail-queue.orig** and will exit with error code **71** which means that the mail server temporarily refuses to send the messages to any of the recipients.

Also if **ravmd** is configured to reject the e-mails that cannot be cleaned **ravqmail** doesn't pass the e-mail to **qmail-queue.orig** and exits with error code 31 meaning that the mail server permanently refuses to send the message to any recipients.

If there is a license limit **ravmd** will not scan the e-mails not belonging to the domains specified in the configuration file. These e-mails will be passed **unchanged** to **qmail-queue.orig**.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd.conf(5), ravmd(8), qmail-queue(8)

NAME

ravsendmail - RAV AntiVirus '**filter client**' for sendmail

SYNOPSIS

ravsendmail $h $f $u $i

DESCRIPTION

The program will be executed by sendmail MTA as a local mailer. It is used to scan all incomming/outgoing e-mails for virus protection and/or content filtering. The program runs as a '**filter client**' for RAV AntiVirus mail scanning daemon (**ravmd**).

USAGE

The following parameters are sendmail internal macros.

$h The recipient host.
$f The sender e-mail address.
$u The recipient e-mail address.
$I The queue id. It is used to identify an e-mail file in the queue.

FILES

/etc/mail/sendmail.cf

NOTES

This product uses two sendmail executables running at the same time. It can introduce a semnificant delay in the e-mail delivery process depending on how often the queues are processed. This is why it is highly recommended to use the RAVMilter product instead of ravsendmail.

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravmd(8), ravmd.conf(5), ravav(8), sendmail(8)

NAME

RAVMilter - RAV AntiVirus '**filter client**' for sendmail

SYNOPSIS

RAVMilter [-dV] [-C sendmail.cf] [-g 0|1|2|3] [-X[on|off]] [--configuration=sendmail.cf] [--daemon] [--debug=0|1|2|3] [--addheader[=on|off]] [–help] [--usage] [–version]

DESCRIPTION

This is an external filter program used by sendmail (compiled with libmilter feature) MTA in order to scan all the incoming/outgoing messages for virus protection and/or content filtering. It uses the **libmilter feature** included in sendmail versions >= 8.11. The program runs as a client for RAV AntiVirus mail scanning daemon (**ravmd**).

USAGE

-d, --daemon
Run as daemon.

-g, --debug=0|1|2|3
Enable debug features (0->silent, 3->verbose).

-C, --configuration=sendmail.cf
Read connection information from this file. See **NOTES** and **FILES** bellow.

-X, --addheader[=on|off]
Add or don't add X-Version header to each message scanned.

-V, --version
Print to stdout RAVMilter version and exits.

--help
--usage
Display information about RAVMilter usage.

FILES

/etc/mail/sendmail.cf
/etc/mail/sendmail.mc

NOTES

Sendmail version must be 8.11 or later in order to compile **libmilter feature** of **sendmail**. Please read **libmilter/README** file from **sendmail** package in order to get instructions for how to compile **sendmail** with this feature. After **sendmail** was compiled and installed with **libmilter feature** you can proceed with RAV AntiVirus instalation.

To **configure RAVMilter** you must add the following two lines into **/etc/mail/sendmail.mc**:
define(`_FFR_MILTER', `true')
INPUT_MAIL_FILTER(`RAVMilter', `S=local:/usr/local/rav8/private/ravmilter.sock, F=R, T=S:10s;R:5m;E:5m')

After that, you must generate **/etc/mail/sendmail.cf** with the following command:
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

When RAVMilter starts it will parse **/etc/mail/sendmail.cf** to find out what socket to bind (in our example it will bind the UNIX socket **/usr/local/rav8/private/ravmilter.sock**).

BUGS

Please mail bug reports and suggestions to: <**mailto:ravteam@ravantivirus.com**>

SEE ALSO

ravm d(8), ravmd.conf(5), ravav(8), sendmail(8)

Index