

User Guide

Vexira Antivirus for Mail Server + i For Linux/FreeBSD/OpenBSD/Solaris/AIX





TABLE OF CONTENTS

VEXIRA ANTIVIRUS FOR MAIL SERVER + I	3
System requirements	4
General information	5
Package naming.....	5
Installation.....	5
Assign to mail sever.....	7
Uninstallation	8
Binary files	8
Statistics screen explanation	9
Registration.....	10
Operation	11
First-level modules /filters/ (Level1)	11
Second-level modules /filters/ (Level2).....	11
Database update	12
The configuration file	13
Structure of the configuration file	13
Filter definitions.....	14
Action specification	16
Action settings	16
VAMSI daemon configuration file (vamsi.conf).....	18
General settings.....	18
Mailserv-module settings.....	20
Log settings	22
General settings of the virus scan engine.....	23
General settings of the spam scan engine	24
Global settings	25
Rule definition	26
Global module settings (level1)	27
Virus filter module settings (level2).....	28
File filter module settings (level2)	31
Spam filter module settings (level2).....	32
Address filter /White/Black list/ (level1).....	35
Result filter module settings (level1).....	37
RBL filter module settings (level1)	39
Module commands	40
Commands belong to level1 modules.....	40
Commands belong to level2 modules.....	41
VAMLOG daemon configuration file (vamlog.conf)	42
General settings.....	42
Log output settings	42
Output rules	43
Tokens.....	44
END USER AGREEMENT	46
CONTACT	48



VEXIRA ANTIVIRUS FOR MAIL SERVER + I

The Vexira Antivirus for Mail Server + i package (hereinafter called VAMSI protection) provides Sendmail, Qmail, Courier and GroupWise /this one only for Linux/ mail servers with virus and spam protection. The solution integrating to the mail server scans all the mails to be delivered and their attachments and ensures comprehensive protection against viruses, malicious codes and unsolicited mails.

Main features:

- Filter modules
- * Virus filter: effective virus recognition based on the outstanding virus scan engine
- * File filter: actions could be assigned to specified file formats
- * Spam filter: statistical spam filtering with many evaluation methods
- * Language/script filtering based on the e-mail's content ('language_filter' option of the spam filter)
- * Address filter - It is possible to set Black/White lists
- * Result filter - summarized result, all results of other filters could be used in the result filter
- * RBL filter
- Flexible rule system: ability to use parameters for the actions to be performed
- Heuristic virus analysis to recognize unknown viruses
- Advanced WormBuster function - for blocking I-Worms instantly
- Filtering encrypted archives
- Comprehensive statistical information about mail traffic and events
- Automatic, incremental virus database update
- Log daemon - logging to several output types: file, syslog, standard output

Spam filter:

The anti-virus system's spam filter operates based on statistical scan methods and has numerous leading evaluation techniques to provide effective protection against unsolicited mails.

- Recognizing based on statistical scan methods completed with other evaluation techniques
- Heuristics filtering
- HTML filtering
- UNICODE text handling
- Low false positives and high spam recognition rate
- Filter sensitivity: 3 level of spam filtering
- Frequently updated spam database



System requirements

Supported operating systems

Linux

FreeBSD 4.9, 5.4, 6.0

OpenBSD 3.4, 3.6

Solaris 9 (5.9)

AIX 4.3

Minimal requirements

Requirements for all the supported platforms:

- 128 MB free memory (256 MB for AIX)
- 64 MB free hard disk space
- wget (for update)
- perl5 (for update)

Requirements by platforms:

Linux, FreeBSD, OpenBSD:

- Intel Pentium (or compatible) processor at 300 MHz
- Minimal required for Linux: GLIBC 2.2.5, kernel 2.2.1
- Minimal required Linux distributions: SuSE 8.0, RedHat 7.3, Debian 3.0 (woody), Mandrake 9.0, Slackware 8.1

Solaris:

- Ultra Sparc IIe processor at 500 MHz

AIX:

- Power3 - II (G3) processor
- Maintenance level 10

Minimum required mail server versions:

Sendmail: 8.12

Qmail: 1.03

GroupWise: 7



General information

Package naming

The 'Vexira Antivirus for Mail Server + i' package is named according to the following parameters:

```
vamsi-<version>-<os>-<architecture>-<minimal libc version>.tgz
```

<version>: The package's version number. For example: 1.0.1

<os>: The package is working on the displayed system. For example: Linux

<architecture>: The processor type. For example: i386

<minimal libc version>: The required minimal version of the libc library. For example: libc6

Installation

You can start the installation by executing the 'vamsi-install.pl' program. After executing the following questions should be answered for the successful installation.

Value displayed between square brackets is default answers for the current question, you can simply use the <enter> button to accept it. It is recommended to use these default values.

One of the first steps is to specify the mail server you want to be protected!

```
#Please select the mail server you want to be protected!
```

```
 #(S)endmail, (Q)mail, (C)ourier or (G)roupWise
```

```
#[s]
```

Set the 'run-as-group' option's value. Please see the configuration specification for more information!

```
#In which group do you want to run Vexira Antivirus for Mail Server + i?
```

```
#[vexira]
```

Set the 'run-as-user' option's value. Please see the configuration specification for more information!

```
#With which user permission do you want to run Vexira Antivirus for Mail Server + i?
```

```
#[vexira]
```

You should specify the location of the binary file in the system:

```
#In which directory do you want to install the binary files?
```

```
#[/usr/sbin]
```

Define the path of the library files needed for the program:

```
#In which directory do you want to install the library files?
```

```
#[/usr/lib]
```

Target path of the database files:

```
#In which directory do you want to install the database (virus, spam) files?
```

```
#[/usr/lib/vexira]
```

Specify the location of the text type documentation files:

```
#In which directory do you want to install the documentation files?
```

```
#[/usr/share/doc]
```

Location for the documentation in man page format:

```
#In which directory do you want to install the manual files?
```

```
#[/usr/share/man]
```



Vexira Antivirus for Mail Server + i

The program creates communication and other files needed during its operation:

```
#Which directory do you want to be the run directory?
#[/var/run/vexira]
```

```
Name the directory of the log file:
Specify the log directory name
[/var/log]
```

```
Set the directory storing the initialization scripts:
#What is the directory that contains the init scripts?
#[/etc/init.d]
```

```
Define path for the initialization directories:
#What is the directory that contains the init directories (rc0.d -
#rc6.d)?
#[/etc]
```

If the program detects that a previous version of its configuration file is available in the system, it will offer the following selection:

```
#Found an existing config file (/etc/vexira/vamsi.conf).
#(K)eeep the existing file or (C)reate a new one?
#[k]
```

```
Specify name for the log file:
#Specify the log file name
#[vamsi.log]
```

```
Mails sent from the IP address you enter will be filtered:
#Which IP address do you want to filter?
#You should use the standard address/length format (example:
#194.222.242.0/24)
#[127.0.0.0/8]
```

```
Please enter your user name:
#Enter your registration user name
#[ ]
```

```
Please enter your registration key:
#Enter your registration key (example: WESAE-WCRVC-CSNEH)
#[ ]
```

If you are about to install the GroupWise protection module, you have to answer additional questions:

```
#Please specify the GroupWise's SMTP Service Queues directory!
```

```
#Please specify the GroupWise's SMTP Queues directory!
```

Consult the Gropupise module settings chapter to get more information about these settings.

The following lines are shown in case of successful installation:

```
Installing files... Done.
Installing config file...
Installing init scripts... Done.
```



Assign to mail sever

To activate anti-virus system you need to perform the following steps beside the configuration settings:

----- Using Sendmail -----

The VAMSI protection must be assigned to the Sendmail so that the mail server and the filter program can communicate to each other. You have to edit the Sendmail's configuration macro file then rebuild it to get the new configuration file.

Please insert one of the following versions into the sendmail.mc file (the name of the Sendmail's macro file may be different on different systems)!

Version A:

This entry consists of 2 lines!

First:

```
MAIL_FILTER(`vamsi', `S=inet:3333@localhost,F=T, T=S:4m;R:4m')dnl
```

Second:

```
define(`confINPUT_MAIL_FILTERS',`vamsi')dnl
```

Version B:

This entry consists of 1 line:

```
INPUT_MAIL_FILTER(`vamsi', `S=inet:3333@localhost,F=T,T=S:4m;R:4m')dnl
```

Please take care of the exact copy!

----- Using Qmail -----

1. Rename the original "qmail-queue" to "qmail-queue2" (the "original_qmail_queue" option found in the configuration file must have the same value)

2. Copy the "qmail-queue" found in the package's "qmail" directory to the Qmail's binary directory (default path: /var/qmail/bin)

3. Reset the owner of the "qmail-queue" which had been copied in the previous step to qmailq and its group to qmail with the following commands:

```
chown qmailq /var/qmail/bin/qmail-queue
```

```
chgrp qmail /var/qmail/bin/qmail-queue
```

----- Using Courier -----

1. Rename the original "submit" to "submit2" (the "original_courier_submit" option found in the configuration file must have the same value)

2. Copy the "submit" found in the package's "qcourier" directory to the Courier's binary directory (default path: /usr/lib/courier/courier)

Qmail and Courier interface module (VARAW client) operation -----

The Qmail and Courier applications consist of modules connected with each other. Vexira Antivirus integrates its own module between the source module (e.g. SMTP daemon) and the 'submit' (Courier) or the 'qmail-queue' (Qmail) module as a transparent proxy. Then it will forward the processed/modified mails to the original 'submit' or 'qmail-queue'.



The integrated VAMSI module will read the access information from the VAMSI's configuration file ([General]/address option) so as to be able to connect to the main VAMSI program. It tries to get the path of the configuration file from the VAMSI_CONFIG environment variable first time, if it fails then from the default path (/etc/vexira/vamsi.conf).

Using GroupWise

If the startup script (vamsictl) detects the GroupWise protection module (vagwia) in the binary folder of the product (/user/sbin by default), the module is ready to use, there is no need to do anything else if the module settings are correct in the configuration file.

Uninstallation

Please run the following program file to uninstall the package:
vamsi-uninstall.pl

Binary files

The following executable files and their parameters are found in the package. These files are placed in the /usr/sbin directory by default:

vamsi [options]

MailScan main daemon program.

Options:

-n, --nodaemon execute in no daemon mode
-v, --version displays the version of vamsi and exits
-c, --config=FILE reads configuration from FILE (path needed)
-l --license returns registration data

vamlog [options]

Log that responsible for controlling log messages.

Options:

-n, --nodaemon execute in no daemon mode
-v, --version displays the version of vamlog and exits
-c, --config=FILE reads configuration from FILE (path needed)

vamsictl start|stop|restart|cfigreload|dbreload|logrotate|statistic

Control file, you can realize the following functions by using the available parameters:

start: Starts the vamlog and vamsi files.

stop: Stops the vamlog and vamsi files.

restart: Stops and starts the vamlog and vamsi files.

cfigreload: Reloads the [Milter] section's settings of the vamsi configuration file and vamlog's configuration file and applies the new settings.

dbreload: Reloads the virus and spam database.

logrotate: Locks the current log file then opens a new one. This function is useful for archiver programs.

statistic: Displays the statistics and exits.

vamstat [options]

Statistics screen about anti-virus system's operation.

Options:

-d, --debug RAW output
-v, --version Display the version number.
-q, --quit Run statistics module only once, then quit.
-a ADDRESS, --address=ADDRESS Statserver's address (e.g. ip:host:port or



unix:path).

-p N, --processnumber=N If multi-process operation enabled, specify the process number in the N parameter. In such a case, the module displays global statistics about the programs run in different processes.

Statistics screen explanation

Date displayed on the right upper corner: system date of the computer running the VAMSI.

Started at:
Startup date of the VAMSI.

Uptime is:
Time has elapsed since the last startup.

Program version:
VAMSI's program version.

Virus database version:
Version of the used virus database.

Virus scan engine version:
Version of the virus scan engine.

Spam database version:
Version of the used spam database.

Spam scan engine version:
Version of the used spam scan engine.

System load average:
System load index number.

Current connections:
Number of connections being currently processed.
(If Sendmail is used: number of clients connected to the server.
If Qmail, Courier or GroupWise is used: number of mails being currently processed.)

Total connections:
If Sendmail is used: total number of connections.
If Qmail, Courier or GroupWise is used: total number of received mails.

Processed mail(s):
If Sendmail is used: total number of processed mails.
If Qmail, Courier or GroupWise is used: the same value as Total connections'.

Blocked mail(s):
Number of refused (dropped or rejected) mails.

Processing error:
Number of errors occurred during the process.

Scanned file for viruses:
Number of attachments scanned by the virus filter.

Virus found:
Number of found viruses.

I-Worms:
Number of found i-worms.



Virus killed:
Number of killed viruses.

Modified attachment(s):
Number of modified attachments.

Deleted attachment(s):
Number of deleted attachments.

Scanned mails for spam:
Number of mails checked by the spam filter.

Spam found:
Number of mails marked as spam by the spam filter.

Registration

The product can't be used without a valid registration key. The program warns the user by sending a message into the log filer once a day when the ending of the registration period is coming. After registration key had expired, the product works as before (without any restriction) until a program update (virus database updating is possible). After program updating, you need a new license (registration key) to use the program.

The registration key must be placed into the anti-virus system's configuration file (serialno option) together with the user name (username option). See the description of the configuration settings for more.



mail-part to the filters modules. The level2 filters also return a string value after processing the MIME-part, you can assign command(s) to the returned value.

Level2 modules (filters):

- virus filter (libflt2_virus.so)
- file filter (libflt2_fileflt.so)
- spam filter (libflt2_bayes.so)

Database update

The product uses incremental virus database update mechanism to keep the virus database up-to-date. The advantage of this method is that the program doesn't need to download the whole virus database file every time (its size is several MBs) but usually only a small additional database package including the virus signatures processed and released recently. Using this mechanism, the download time is decreased to a minimal level so we can release additional virus database packages several times a day to improve the defense. Users can obtain protection against new malware without spending long time and generating considerable network load for the update. The protection is available almost immediately after the signatures of the newly discovered viruses have been processed in our virus lab.

To automate the update processes, use the available scripts attached to the package, they are placed to the /usr/sbin/ directory (vam_dbupdate.sh and vam_dbupdate_http.sh).

Execute one of them, it is going to download the virus- and/or spam database, copies it/them into the correct directory and activates it/them. The download and update processes will only be performed, if the database available in the server is newer than one on your computer. Otherwise the database will be left unchanged.

To execute the scripts, you should enter the vam_dbupdate.sh (through HTTP use the vam_dbupdate_http.sh) command. It is possible to use parameters, too:

nosdb - the spam database will not be updated
verbose - display progress bar

Example:

```
vam_dbupdate.sh nosdb verbose
```

The spam database will not be updated, the progress bar will be displayed.

To run these scripts, you need wget program! By the help of cron, you can schedule the script executing to be performed by half an hour. Register into /etc/crontab:

```
0,30 * * * * root /usr/sbin/vam_dbupdate.sh
```

The database files can be downloaded manually from the following FTP server:

The virus database consists of several files, you need to download all the files from the following folder:

upd.vexira.com/pub10/vexira/vdb/

Spam database file, you need to uncompress before use:

upd.vexira.com/pub10/vexira/tgzs/vexira.tgz



The configuration file

Structure of the configuration file

The configuration file stores the program settings in hierarchical structure. The storing mechanism based on the encapsulation concept which means that user has to specify the storing path (section) for each coherent setting group step by step.

The path (section) must be specified between square brackets in the configuration file:

```
[Milter/Global]
```

Enter comments by using semicolon (;) before the comment text. The characters entered after semicolon will not be interpreted by the parser. You can also use this function to disable a selected option quickly.

```
;command2="copy_mail ('/tmp')"
```

If you have to specify network addresses, enter the following address forms usually in the whole configuration file:

```
unix:/path/to/file
```

```
inet:port@{hostname|ip-address}
```

Example:

```
unix:/var/run/vexira/vamsi
```

```
inet:9009@192.168.2.42
```

```
inet:9427@somebody.com
```



Filter definitions

```
[Milter/FilterRules/Rule/Filter]
; level1 filter settings (1)...
[Milter/FilterRules/Rule/Filter/Action]
; level1 action settings (1)...

[Milter/FilterRules/Rule/Filter]
; level2 general filter settings (2)...
[Milter/Filterrules/Rule/Filter/Level2]
; level2 filter settings (2)...
[Milter/FilterRules/Rule/Filter/Level2/Action]
; level2 action settings (2)...

...
[Milter/FilterRules/Rule/Filter]
; filter settings (n)...
[Milter/FilterRules/Rule/Filter/Action]
; action settings (n)...
```

Filter settings belonging to a rule must be defined after the rule definition line in the [Milter/Filterrules/Rule/Filter] section. Each new filter definition must be placed into a new [Milter/Filterrules/Rule/Filter] section. After the mail has been scanned by the filter, the program returns the filter's result. Based on these values, different actions may be performed on the checked mail. You can specify the required actions and result types in the [Milter/Filterrules/Rule/Filter/Action] section for the level1 filters.

The level2 filters are controlled by a special level1 filter, called: libflt_level2. If you would like to configure a level2 filter, first you have to specify the control filter (libflt_level2) and set its general settings in the [Milter/FilterRules/Rule/Filter] section. After defining general settings, you have to select and set the selected level2 filter in the [Milter/Filterrules/Rule/Filter/Level2] section. The actions (based on the filter result) may be specified in the [Milter/FilterRules/Rule/Filter/Level2/Action] section.

GENERAL FILTER OPTIONS

disable = <0|1>
 Disable or enable filter. Possible values: 0 or 1.
 1: Filter is disabled.
 0: Filter is active, enabled.

filter = <filter type>
 Set filter type.
 The available values (module level):
 libflt_addr - address filter (level1)
 libflt_result - result filter (level1)
 libflt_rbl - RBL filter
 libflt_level2 - level2 module manager (level1)
 libflt2_virus - virus filter module (level2)
 libflt2_fileflt - file filter module (level2)
 libflt2_bayes - spam filter module (level2)

In case of level2 filter:

filter2path = <path>
 Directory specification of the Level2 filter modules. Location where the level2 modules can be found.



max_mime_depth = <number>

The program is going to scan the embedded e-mail-type mimes down to the specified depth. Setting the 0 (zero) value for the option, none of the embedded e-mail-type mimes will be scanned.

Important!

E-mail-type mimes embedded deeper into the mail than the value of this option will not be scanned, so harmful materials may have found in that deeper levels will get into your system.



Action specification

```
[Milter/FilterRules/Rule/Filter]
filter_option_1
filter_option_2
...
filter_option_n

    [Milter/FilterRules/Rule/Filter/Action] ; action (1)
    result=
    count=
    command=

    [Milter/FilterRules/Rule/Filter/Action] ; action (2)
    result=
    count=
    command=
    command2=
    command3=
    ...
```

The filter module returns the result of the mail or attachment scan. The program compares this returned value with the 'result' property of the specified actions and where the values are the same, the action will be performed. You can use regular expression in the 'result' option, in such a case you must insert it between quotes (""). It is also possible to use tokens in the 'command' option.

Note!

Specify level1 actions in the [Milter/FilterRules/Rule/Filter/Action], and level2 actions in the [Milter/FilterRules/Rule/Filter/Level2/Action] section.

Action settings

result = <result value>

You can specify a result value for the filter module. If the filter module returns the same value as you specified, the 'command' options of the section will be performed. Result values may be different by filter modules, these are described in the chapter of the module descriptions. Regular values can be used between quotes ("").

count = <number>

This value specifies that how many times should the commands be performed during the mail processing (0 means unlimited). If 'count' option is not specified to the action, the command will be always performed.

command = <command>

Define actions. The possible commands and their functions is detailed in the "Module commands" chapter. You must insert the commands' value between quotes (""), the parameters between apostrophes (''), separated by comma (,). If you would like to use the apostrophe (') character in the parameters, then use the backslash (\) character right before it.

For example:

```
command="header_modify ( 'Subject', '%subject% \'Scanned e-mail\'' )"
```

Specify several parameters of the same name:

If you would like to specify several parameters of the same name, then you have to number them from the second one (where the number: 2..N).

For example:

```
[Milter/Filterrules/Rule/Filter/Action]
command=
```



Vexira Antivirus for Mail Server + i

command2=
command3=

The daemons of the package have different configuration files which will be detailed below.



VAMSI daemon configuration file (vamsi.conf)

General settings

```
[General]
logaddress=unix:/var/run/vexira/vamlog
address=inet:3333@localhost
socket_permission=0660
run-as-user=user
run-as-group=group
pid_file=/var/run/vexira/vamsi.pid
tempdir=/tmp
process_num=0
module=
```

The settings:

logaddress=<netcmd address>

You have to specify the communication address of the log component.
Default: logaddress=unix:/var/run/vexira/vamlog

address=<socket>

Specify address through which the MTA and the anti-virus application will communicate.
The same setting must be specified in the MTA's configuration Default:
address=inet:3333@localhost

socket_permission=<octal number>

Set unix socket permission with an octal number. Default: 0660 (in case of Qmail),
0600 (in case of Sendmail).

run-as-user=user

run-as-group=group

VAMLOG daemon starts with root permission as all the daemon programs usually at the computer startup. However, it is more secure to run with unprivileged user permission. If the VAMLOG is started with root permission, it is able to change to the user and group specified in these options.
If the VAMLOG is not started with root permission, it will not be able to change to other user permissions.

pid_file=/var/run/vexira/vamsi.pid

Pid file with path of the anti-virus application.

tempdir=/tmp

Set the temporary directory used by the application.

Default: /tmp

process_num = <processor number>

It is recommended to use this option if the following system components are available: FreeBSD 4.x, multi-processor system, SMP kernel
The kernel is only able to assign the processes to the CPUs existed on FreeBSD 4.x in multi-process environment (the anti-virus protection is thread-based so it will always be run by only one processor). In this option you can specify the number of the anti-virus protection instances to be run, these will be processed by the processors separately. A built-in load-balancing system is responsible for the equal load, it will assign the Sedmail connects to the instances of the anti-virus protection.

Operation:

It creates sockets according to the 'process_num' value:

```
unix:/var/run/vexira/vamsi ->
```

```
unix:/var/run/vexira/vamsi.0
```



Vexira Antivirus for Mail Server + i

```
unix:/var/run/vexira/vamsi.1  
...  
unix:/var/run/vexira/vamsi.n
```

or

```
inet:3333@localhost ->  
inet:3334@localhost  
inet:3335@localhost  
inet:3336@localhost  
inet:3337@localhost
```

module=

Specify the mail server interface module that makes the connection possible between the selected mail server and the anti-virus system.

libvaraw.so - using Qmail, Courier or GroupWise MTA

libvamilter.so - using Sendmail



Mailserv-module settings

Sendmail module setting

```
[General/Milter]
Milter_timeout = 300
smtpserver =
modify_body=
```

Set the following option if you are using Sendmail:

milter_timeout = <second>

Sets the number of seconds until libmilter is waiting for an MTA connection before timing out a socket.

smtpserver = <server address>

If the processed e-mail is refused by the reject_mail command, you can set an MTA needed to deliver the mail when the send_copy command is used (the set MTA must be different to the one to which the VAMSI is integrated).

Use the following address form:

```
inet:port@{hostname|ip-address}
```

Example:

```
inet:9442@somebody.com
```

modify_body = <0 or 1>

Certain mail server programs (e.g. Postfix earlier than version 2.3) will not work properly if the mail-body is modified through the milter interface during the virus scan process. Using this option you can enable (1) or disable (0) the modification of mail body if necessary.

Default setting: 1

Qmail module settings

```
[General/Qmail]
original_qmail_queue=/var/qmail/bin/qmail_queue2
accept_mail_retval=0
drop_mail_retval=0
reject_mail_retval=31
```

Set the following options if you are using Qmail:

original_qmail_queue=<path>

Path of the original qmail-queue which will be called by the VAMSI's own qmail-queue to deliver the mail finally.

Default: /var/qmail/bin/qmail_queue2

accept_mail_retval=<number>

drop_mail_retval=<number>

reject_mail_retval=<number>

Set the return value of the anti-virus system returned to the Qmail module if one of the above incidents (accept mail, drop mail, reject mail) have been detected. Based on these results you can control the Qmail's return value returned to the mailer client. For more information read the Qmail's own manual (man) ('qmail-queue').

Courier module settings

```
[General/Courier]
original_courier_submit=/usr/lib/courier/courier/submit2
```



Set the following options if you are using Qmail:

original_courier_submit=<path>

Path of the original 'submit' which will be called by the VAMSI's own 'submit' to deliver the mail finally.

Default: /usr/lib/courier/courier/submit2

GroupWise module settings

[General/Groupwise]

daemon_pid_file=/var/run/vexira/vagwia.pid

service_queue_dir=

queue_dir=

check_incoming=yes

check_outgoing=yes

checking_period=10

Set the following options if you are using GroupWise:

daemon_pid_file=<path>

Path of the vagwia pid file.

Default: /var/run/vexira/vagwia.pid

service_queue_dir=<path>

Set the SMTP Service Queue of the GroupWise Internet Agent to this option.

Find the requested path in the ConsoleOne program:

1. In the selected domain, click with the right mouse button on the GWIA you want to protect then select the 'Properties' menu item.
2. Select the 'Server Directories' panel.
3. Click the 'Advanced' button.

Enter the path found in the 'SMTP Service Queues Directory' setting to the 'service_queue_dir' option.

queue_dir=<path>

Enter the Groupwise Internet Agent's directory here. You can also find this path in the ConsoleOne program:

Do the 1st and 2nd steps mentioned above, find the 'SMTP Queues Directory' option on the 'Server Directories' panel and set it's value to the 'queue_dir' options.

check_incoming=<yes/no>

Yes: Incoming mails will be scanned (this is the default setting).

check_outgoing=<yes/no>

Yes: Outgoing mails will be scanned (this is the default setting).

checking_period=<period>

Set a time period (second) after the product scan the specified folders.

Default: 10



Log settings

[Logging]

logscreen=0

The setting:

logscreen=<0|1>

Log to screen.

1: The log messages came from the filter modules will be displayed on the screen.
This function could be used in non-daemon mode.

0: Inactive.



General settings of the virus scan engine

```
[Engine]
max_decompress_size=0
max_decompress_ratio=0
max_decompress_depth=5
vdb_file=/usr/lib/vexira/vdb9.xml
-----
```

Specify the general setting of the scan engine in the [Engine] section.

max_decompress_size=0

If this file size limit is exceeded while uncompress an archive, the program stops the uncompression and scanning of the file and returns the 'archive_exploit' result. (Option's value is in MByte).

Specifying the 0 value means using the virus scan engine's default value for this option.

max_decompress_ratio=0

If the size of the decompressed file is 50 times (or more) greater than the compressed file's, the program will return the 'archive_exploit' result.

Specifying the 0 value means using the virus scan engine's default value for this option.

Other explanation (option's value in percent): $1/n*100$, where n is the value.

For example the value is 50. $1/50*100 = 2\%$ so if the compression ratio is better than 2% the program will return the 'archive_exploit' result.

max_decompress_depth=5

The program will scan the multi-level archives down to the specified depth. If the program finds more depth levels, it will return the 'archive_depth_limit' result and files that are deeper than the specified level will not be scanned.

vdb_file = <file name with path>

Location of the XMS descriptor file for the virus database.



General settings of the spam scan engine

[Bayes]

```
bayes_sdb=/usr/lib/vexira/vexira.sdb
```

Specify the general setting of the spam scan engine in the [Bayes] section.

bayes_sdb = <File name with path>

The spam database file's name and location in the system.



Global settings

```
[Milter]
[Milter/Global]
username=user_name
serialno=xxxxx-xxxxx-xxxxx
filters=/usr/lib/vamsi/
acceptnomatch=1
cfg-watch-timer=120
stataddr=unix:/var/run/vexira/vamstat
max-connections=100
-----
```

You can find the MAILFILTER daemon settings in the [Milter] section. Inside this section the general settings are in the [Milter/Global] section.

username = <user name>

Specifying the user name based on your license.

serialno = <registration key>

Specifying the registration key in the following form: XXXXX-XXXXX-XXXXX

Note!

You are not allowed to use the program without (valid) registration data!

filters = <path>

Level1 filter modules' directory specification. The location where the level1 modules could be found.

acceptnomatch = <number>

How the program handles the mails which not matching the rules?

0: Refuses them.

1: Accepts them, but it does not perform filtering them, mails will be forwarded without checking.

cfg-watch-timer = <second>

The cfg-watch-timer field sets the intervals at which the program should check if the configuration file has been modified. If so the file will be reloaded.

stataddr=unix:/var/run/vexira/vamstat

The statistical server communicates through the specified address.

Default: stataddr=unix:/var/run/vexira/vamstat

max-connections=100

Client connection limit. Maximum number of the clients that will be allowed to connect to the anti-virus system. If this limit is reached 4xx error message (temporary unavailable) will be returned to the MTA in case of every further attempt.



Rule definition

```
[Milter/Filterrules]
[Milter/Filterrules/Rule]
sourcemask=194.222.242.0/24
```

Specify a rule, the filter modules defined for this rule will be applied to the mails matching this rule. Define the rule in the [Milter/Filterrules/Rule] section inside the [Milter/Filterrules] section.

sourcemask = <domain>

Filter modules defined after the sourcemask option will be applied to the mails sent from the specified domain. These filter modules belong to this rule. If you insert a new sourcemask option (with the required section specifications) the filter modules defined after the new sourcemask option will be applied the mails matching the new rule (sourcemask).



Global module settings (level1)

```
[Milter/Filterrules/Rule/Filter]
```

```
disable=0  
filter=libflt_global
```

```
[Milter/Filterrules/Rule/Filter/Action]
```

```
result=true  
command="add_header('X-VAMSI', 'Scanned by VAMSI')"
```

```
-----
```

In this module you can set actions which will be performed on each mail processed. Set the global filter module for the 'filter' option (libflt_global).

Actions

This module returns 'true' value without exceptions. Commands could be used in the 'command' option are detailed in the "Module commands" chapter.

In this instance, the selected 'command' will mark the processed mail. The 'add_header' option adds a new field and its content to the mail header.



Virus filter module settings (level2)

```

; ---- level2 filter module initialization
[Milter/Filterrules/Rule/Filter]
disable=0
filter=libflt_level2
filter2path= /usr/lib/vexira/filters/
; ---- end of level2 filter module initialization

[Milter/Filterrules/Rule/Filter/Level2]
disable=0
filter2=libflt2_virus

filemask=%DEFAULTMASKLIST%, *.jpg
exclude-filemask=

search_method=strict
heuristic_level=normal
macro_delete=no
containers=yes

[Milter/Filterrules/Rule/Filter/Level2/Action]
result="infe.*"
command="modify_header ('Subject','%virusname%')"
command2= \
"replace('*.txt','iso-8859-2',
'*****
** Attachment %filename% was infected with %virusname% virus,
** attachment part was removed.
*****')"

[Milter/Filterrules/Rule/Filter/Level2/Action]
result=cleaned
command="continue"

[Milter/Filterrules/Rule/Filter/Level2/Action]
result=i-worm
command="drop_mail"

```

The above configuration part is a possible example of the virus filter module setting. Because the virus filter module is level2 module, first the level2 manager module (libflt_level2) and the location of the level2 modules ('filter2path') must be defined. This is the initialization method of the level2 modules.

You have to specify and set the requested level2 module in the [Milter/Filterrules/Rule/Filter/Level2] section.

filter2 = <level2 filter module type>

Set the requested filter module, in the present case this is the virus filter module (libflt2_virus).

filemask =

Please specify attachment names and mask that you would like to be scanned by the filter module. These must be separated by commas (,). If you specify the star (*) character, then all the files will be scanned. Use the following mask to cover attachments without any extensions: "*."

The %DEFAULTMASKLIST% token is also available to use, it represents the default mask list of the engine (if the virus filter is active, this mask list is displayed in the log file (in case of INFO or greater log level)).

Remark!



In case of appearance of a new, critical threat, extra file masks may be added to the existed list automatically, supplied by the virus database. This file mask will also be scanned by default while the threat spells real danger.

exclude-filemask=

Attachment matching one of the filemasks specified in this option will not be virus scanned, even if its extension is registered in the 'filemask' option. Specify this option in the same way as 'filemask'.

search_method = <fast/strict/full>

Specify the search method. The virus scanning engine is able to scan for and detect viruses according to the set methods/levels. It is possible to choose the needed scanning method in the components in the software. The following levels are available:

fast:

Only scans those parts of the file, which are most likely to contain a virus and does not detect viruses, which can only be detected by using a major amount of system resources (e.g. Excel FORMULA viruses).

extensive:

Optimized scanning method, which detects all viruses registered in the virus database and scans those parts of the file, which are most likely to contain a virus.

full:

Detects all viruses registered in the virus database and scans the whole file, even those parts, where viruses are not likely to be found.

heuristic_level = <off/normal/high>

During the heuristic analysis, the software tries to detect codes and programs, which have virus-like characteristics but are not registered in the virus database. If such a suspicious file is found, the user is notified. The following levels of heuristic analysis are available:

off:

No heuristic analysis.

normal:

The depth of the analysis is limited, the possibility of false positives is low, but the chance of detecting unknown viruses is not too high.

high:

The chance of detecting unknown viruses is higher, but there is a higher possibility of false positives.

macro_delete = <yes/no>

yes: all the macros will be deleted.

no: inactive.

containers = <yes/no>

Scanning in compressed files.

yes: scanning in container files (archives, compressed files). The anti-virus system recognizes the compressed, archived files automatically.

Returned results, actions

Available return values of the virus filter module:

none: there was no virus found

i-worm: infected file, I-Worm type incident

cleaned: infected file, virus successfully killed

infected: infected file, fail to kill virus

encrypted_archive: compressed file protected by password

archive_exploit: too big archive (exploit)

archive_depth_limit: if the limit of the 'max_decompress_depth' option exceeded

error: error occurred during processing



These values are available to use as the value of the 'result' option. Commands could be used in the 'command' option are detailed in the "Module commands" chapter.

In the example

1.
result="(infected|i-worm)"
The result value is specified as a regular expression. If the filter module returns a string either 'infected' or 'i-worm' then the specified action will be performed. In this example the command is header modification: the program inserts the name of the virus into the subject field. Because this result has a secondary command (command2) so it also will be performed: The infected attachment will be replaced to the warning text file.
2.
result=cleaned
If the attachment was infected, but the virus was killed successfully, then the command to be performed is the 'continue', the mail will be forwarded to the MTA to deliver.
3.
result=i-worm
If the virus filter recognizes the mail as Internet worm, then the drop_mail action will be allied to the mail, the mail will not be delivered.



File filter module settings (level2)

```
[Milter/Filterrules/Rule/Filter/Level2]
disable=0
filter2=libflt2_fileflt

filemask=*.pif, *.scr, *.vbs
use-regex=no

[Milter/Filterrules/Rule/Filter/Level2/Action]
result=true
command="delete"
-----
```

Initialization of level2 modules has been described in the virus filter section. Initialization must be done only once so if it has been initialized in the virus filter (or rather in the first module specification in the configuration file) you don't need to do it again.

filter2 = <level2 filter module>

Set the requested filter module, in the present case this is the file filter module (libflt2_fileflt).

filemask =

You can specify as filemask as you wish by using the * and ? characters. If one of the filemask matches the attachment files, the specified command will be applied to the file. The file mask values must be separated by commas (,).

Returned results, actions

Return values of the file filter module:

true: the name of the file (attachment) matched one of the values of the 'filemask' option
false: the name of the attachment) didn't match the values of the 'filemask' option
error: error occurred during processing

In the example

result=true

If one of the values of the 'filemask' option matches the name of the file, the program will delete the attachment according to the command option's value.



Spam filter module settings (level2)

```
[Milter/Filterrules/Rule/Filter/Level2]
disable=0
filter2=libflt2_bayes

;language/script filter activation
language_filter=yes

;spam filter level setting
filter_level=high

;actions for each spam level
[Milter/Filterrules/Rule/Filter/Level2/Action]
result=low_level_spam
command="drop_mail"

[Milter/Filterrules/Rule/Filter/Level2/Action]
result=normal_level_spam
command="reject_mail('550','Recognized as SPAM')"
command2="add_rcpt_to ('admin@domain.com')"

[Milter/Filterrules/Rule/Filter/Level2/Action]
result=high_level_spam
command="modify_header ('Subject','***SPAM*** %subject%')"

;common spam filter action for spams
;[Milter/Filterrules/Rule/Filter/Level2/Action]
;result=true
;command="modify_header ('Subject','***SPAM*** %subject%')"

;for language/script filter
[Milter/Filterrules/Rule/Filter/Level2/Action]
result=language_chinese
command="drop_mail"
```

Initialization of level2 modules has been described in the virus filter section. Initialization must be done only once so if it has been initialized in the virus filter (or rather in the first module specification in the configuration file) you don't need to do it again.

Spam filtering:

The spam filter returns the 'true' or 'false' result to indicate the mail is spam ('true') or not ('false'). In case of spam it also returns the level of the spam. Set your security spam level in the 'filter_level' option and...
...use the 'true' result in the rule if you don't want to set different actions for the spam according to its spam level (common action).
...use the name of the security levels in the rules to assign different actions for the spam according to its spam level.

Language/script filtering (language_filter):

The language filter function provides great ability to filter e-mails according to the language and script-type of their text parts. The language/script-type database that needs for the recognition is built in the vexira.sdb (spam database) file so you need to have the spam database file downloaded and available to activate the language filter.



The 'language' means the natural language of the mail-text, for example: English, Hungarian, Russian, Chinese, etc.
The 'script-type' means the character-set used in the mail, for example: Latin, Cyrillic, Greek letters, Far-Eastern letters, etc.
The language filter works based on heuristics detection so its result will be the most likely script-type/language used in the mail.

Options:

filter2 = <level2 filter module>

Set the requested filter module, in the present case this is the spam filter module (libflt2_bayes).

filter_level = <low/normal/high>

Filter level setting. The filter marks the mail as spam which is found on the specified spam level or below.

low:

Insignificant false positives, the spam detection rate is normal. This means that the spam filter only marks that mails which are real spam by the spam database, normal mails are not affected (low false positives).

normal:

The false positive index increases a bit compared to 'low' level. This level provides effective spam recognition. This is the optimal level.

high:

The number of false positives increases but the filter filters out almost all the spam mails on this level. This setting is recommended if mails marked as spam can be reviewed because of the relatively high number of false positives.

If the mail is marked as spam on the selected level, the specified action will be performed. Different levels should have another actions. The following actions are recommended for the levels:

low: drop_mail

normal: reject_mail, add_rcpt_to

high: modify_header

language_filter = <yes/no>

Enable/disable language/script filter.

Returned results, actions

Return values of spam filtering:

true: the spam filter marked the mail as spam based on the specified setting

false: the mail is not spam according to the spam filter

If the result is 'true', the spam levels also be returned (explanation read above):

low_level_spam

normal_level_spam

high_level_spam

Return values of language filtering:

If the language/script filter is enabled, the program also returns the language/script type of the language filter.

When the program is started, it copies the available values to the log file if the language/script filter is active (language_filter=yes). The language/script database is improved continuously, that's why you can see the supported languages and scripts in the log file (in case of INFO or greater log level).



The supported language/script values can be set to the 'result' option.

Example:

```
result=language_english  
or  
result=script_latin
```

It is possible to invert the result of the comparison if you use a '!' (exclamation) mark at the beginning of the value. In such a case, the action(s) will be applied all the mails having different language/script to the specified one(s).

Example:

```
result=!language_hungarian|language_english
```

In such a case, the action(s) will not be applied to the mails using hungarian or english but to all the others.

Other possible result:

```
error: error occurred during processing
```

In the example

Assign different actions for spam mails:

result=low_level_spam

The mail is surely spam, the program simply does not forward the mail.

result=normal_level_spam

The mails is spam most likely, so the program rejects the mail 'command1' and sends a copy to the address specified in the 'command2' command.

result=high_level_spam

Because of the increase number of false positives, the program only modifies the subject field of the mail and forwards the mail back to the MTA.

Common action for spams:

result=true

If the mail is marked as spam on the high level, the program modifies the subject field of the mail and forwards it back to the MTA.

Language/script filtering:

result=language_chinese

All the mails written in Chinese language will be dropped (command="drop_mail").



Address filter /White/Black list/ (level1)

```
[Milter/Filterrules/Rule/Filter]
```

```
disable=0  
filter=libflt_addr
```

```
[Milter/Filterrules/Rule/Filter/Address]
```

```
sender=1  
entry=*@domain.com  
external_file=/etc/vexira/wlistaddr.txt
```

```
[Milter/Filterrules/Rule/Filter/Action]
```

```
result=all_rcptto_listed  
command="accept_mail"
```

This module filters the sender or recipient(s) of the mail based on the specified address(es).

Functioning:

- if all the recipients or the sender of the mail (according to the setting) are/is included in the address list then the action specified will be applied on the mail (in case of mailfrom_listed or all_rcptto_listed results)
- if there is at least one of all the recipients or the sender of the mail (according to the setting) who are/is not included in the address list and the mail would be blocked for this, the mail will be delivered without modification for those recipient(s) who are included in the address list. In this case even those actions will not be applied which would not modify the mail. (eg. copy mail)
- if there is at least one of all the recipients who is not included in the address list but the mail would not be blocked for this, the mail will be delivered to all the recipients with possible modifications which were set in the 'command' options.

sender=[0|1]

- 0: the module will filter the recipient addresses
- 1: it will filter the sender address

entry=[address(es)]

Enter address(es) to be filtered. Use comma (,) character to enumerate a number of addresses. You can use the * joker character in the localpar of the addresses. Example: *@domain.com, aaa@bbb.ccc

external_file=[file name with path]

Addresses to be filtered could be stored in an external file, too. You can specify the filename with its path in this option. The addresses will be read by lines from the file. If a semicolon (;) is placed at the beginning of the line, that line will be considered as comment.

Example for external file content:

```
*@domain.com  
a@b.com  
;d@e.com
```

Returned results, actions

Available return values of the address filter module:

mailfrom_listed: the sender is included in the list

all_rcptto_listed: all the recipients/sender are included in the list

rcptto_listed: there is at least one recipient/sender who is not included in the list

These values are available to use as the value of the 'result' option.



Commands could be used in the 'command' option are detailed in the "Module commands" chapter.

In the example

According to the result of the filter (resultt=all_rcptto_listed) all the recipients are included in the address list so the mail will be accepted (command="accept_mail").



Result filter module settings (level1)

```
[Milter/Filterrules/Rule/Filter]
disable=0
filter=libflt_result
```

```
[Milter/Filterrules/Rule/Filter/Action]
result=l2bayes_true.*l2virus_infected
command="drop_mail"
```

```
[Milter/Filterrules/Rule/Filter/Action]
result=addr_mailfrom_listed.*l2bayes_true
command="add_rcpt_to ('admin@domain.com')"
```

All results of other filters (virus, spam, ...) specified in the configuration file before the Result filter are available in this special filter. You can connect filters by assigning actions based on their result combinations specified in the Result filter.

So the Result filter provides in a string (result string) all the filters' results which have been performed before the Result filter. With the help of regular expressions you can compare various conditions with the result string and assign actions to the mail if there is a correspondence.

Actions as reject_mail, drop_mail or accept_mail specified before the Result filter can block the activation of the Result filter because these ones break the mail process so the Result filter can not be activated. Keep this in mind when composing the configuration file and the actions.

Other possibility is not to assign actions to the filters specified before the Result filter but set them in the Result filter getting their results from the result string.

Because the results can be the same even if they are resulted by two different filters (e.g. true), these values must be distinguished from each other. Use the following prefixes at the beginning of the result separated by an underline:

```
address filter: addr
spam filter: l2bayes
file filter: l2fileflt
virus filter: l2virus
global filter: global
rbl filter: rbl
```

In the example

1.
Reject infected and simultaneously spam mails:
virus filter result: infected
spam filter result: true

Use this mask if the spam filter is placed before the virus filter:
result=l2bayes_true.*l2virus_infected
command="drop_mail"

2.
Forward spam mails to the administrator that come from a specified sender:
spam filter result: true
address filter result: mailfrom_listed

Use this mask if the address filter is placed before the spam filter:



```
result=addr_mailfrom_listed.*l2bayes_true  
command="add_rcpt_to ('admin@domain.com')"
```

Important!

For correct operation, place the filters' results into the result mask in the same order as the filters are specified in the configuration file. Also keep the order of values inside a filter: first use the incident level (e.g. l2bayes_high_level_spam) then the incident flag (l2bayes_true).



RBL filter module settings (level1)

```
[Milter/Filterrules/Rule/Filter]
disable=0
filter=libflt_rbl

[Milter/Filterrules/Rule/Filter/DNSRBL]
host=relays.ordb.org
[Milter/Filterrules/Rule/Filter/DNSRBL]
host=sbl.spamhaus.org

[Milter/Filterrules/Rule/Filter/Action]
result=true
command="drop_mail"
```

In the RBL filter, you can specify web sites providing realtime database of IP addresses of verified spam sources, supplied as a free service to help email administrators better manage incoming email streams. Before receiving a mail, the program checks if the sender's IP address can be found in the specified database(s). If so, the selected command(s) will be performed.

Filter settings:

disable = <number>

Enable/disable the filter. Values: 0 or 1
1: The filter is disabled
0: The filter is enabled

filter = <filter type>

Specifying filter type. Set the libflt_rbl (level1) to the RBL filter.

Set RBL server in the 'host' option of the [Milter/Filterrules/Rule/Filter/DNSRBL] section (you are allowed to set it several times).

Returned results, actions

Available return values of RBL filter module:

true: IP address of the sender is found on the list(s)
false: IP address of the sender is NOT found on the list(s)

These values are available to use as the value of the 'result' option. Commands could be used in the 'command' option are detailed in the "Module commands" chapter.

In the example

```
result=true
If the RBL servers' IP list includes the client's IP (true), the program execute the drop_mail command ('command').
```



Module commands

Commands belong to level1 modules

continue

The mail processing may continue.

accept_mail

The mail will not be scanned, it will be accepted without checking.

reject_mail

The processing (filtering) may not continue. The mail will be rejected without scanning and the error codes and messages will be returned to the mailing client.

Parameters: error code, error message (only 5xx type error codes accepted)

Example: `command="reject_mail('550','Mail recognized as SPAM')"`

Important!

Using Q-mail mailing system, the given parameters will not be returned because the Q-mail will overwrite them with its own error code and message!

drop_mail

The mail processing may not continue, the mail will be accepted but will not be forwarded.

copy_mail

If one of the modules break the mail processing then the program will copy the whole mail named as 'mailXXXXXXX' where the XXXXXX is a random generated number. Parameter: target directory.

Example: `command="copy_mail ('/tmp')"`

add_rcpt_to

A copy of the mail will also be sent to the recipient specified in the parameter.

Parameter: e-mail address

Example: `command="add_rcpt_to ('admin@domain.com')"`

send_copy

The e-mail will be forced to forward to the address specified in this command, even if the mail is refused with the 'reject_mail' or 'drop_mail' command. If the returned value is 'continue' this command will work the same as 'add_rcpt_to'.

Parameters: mail state, e-mail address

The mail state parameter is not considered yet but you need to specify.

Example: `command="send_copy ('original', 'user@domain.com')"`

override_rcpt_to

The original and the added (send_copy, add_rcpt_to) recipients of the mail will be overwritten by the address specified in this command if the mail should be delivered.

Parameter: e-mail address

Example: `command="override_rcpt_to ('user@domain.com')"`

Important!

Use multiple add_rcpt_to and send_copy commands to set more than one recipients. It is pointless to use the override_rcpt_to command repeatedly because the mail will only be forwarded to the last-set one.

modify_header

Modify the mail's header. If the specified field could not be found in the header, then it will be inserted. Parameters: field name, value.

Example: `command="modify_header ('Subject','%virusname%')"`

add_header

Insert the specified field to the mail's header. Parameters: field name, value.



Example: `command="add_header('X-VAMSI', 'Scanned!')"`

execute_command

Execute external command. Parameters: name of the program to be executed (with path), possible command line switches. The anti-virus systems' tokens can be used in the command line.

Example: `execute_command('touch /tmp/command-executed')`

Commands belong to level2 modules

All the commands belonging to the level1 modules are available completed with the following:

delete

Delete attachment.

replace

Replace attachment to text file. Parameters: file extension, char set, text.

Example: `command="replace('*.txt','iso-8859-2','*The %filename% attachment is infected by the %virusname% virus*')`

modify

Modification is allowed. (For example the virus filter module is killed the virus)

copy

Makes a copy of the original file. The file will be named as 'mailXXXXXX' where the XXXXXX is a random generated number. Parameter: target directory.

Example: `copy('/var/lib/quar')`



VAMLOG daemon configuration file (vamlog.conf)

The LOG component is responsible for storing and handling the log messages came from other modules of anti-virus system. The modules could send messages to the LOG daemon with the help of netcmd.

General settings

```
[General]
netcmdaddr=unix:/var/run/vexira/vamlog
pid_file=/var/run/vexira/vamlog.pid
run-as-user=user
run-as-group=group
```

The settings:

netcmdaddr = <netcmd address>

You have to specify the communication address of VAMLOG component.
Default: netcmdaddr=unix:/var/run/vexira/vamlog

pid_file= <pid file>

Pid file with path of vamlog.

run-as-user=user

run-as-group=group

VAMLOG daemon starts with root permission as all the daemon programs usually at the computer startup. However, it is more secure to run with unprivileged user permission. If the VAMLOG is started with root permission, it is able to change to the user and group specified in these options.

If the VAMLOG is not started with root permission, it will not be able to change to other user permissions.

Log output settings

```
[OutputSetting]
[OutputSetting/Output]
type=file
filename=/var/log/vamsi.log ;if type=file
perpid=0 ;if type=file
facility= ;if type=syslog
ident= ;if type=syslog
format "[%d/%m/%Y %H:%M:%S %z] $k $l $A $C PID:$P TID:$T \"$M\""
```

There can be only one [OutputSetting] section specified in the vamlog.conf file. The output sub-sections could be defined inside this section.

You can set the log's general setting in the [OutputSetting/Output] section. The number of [OutputSetting/Output] sections is not restricted. Each of these sections have different rules. Rules determine which messages occurred in the system should be logged. The VAMLOG daemon processes the [OutputSetting/Output] sections and if the new log message's type matches one of the rule, it registers the log according to the settings of the specific output section.

type = <file|syslog|stdout>

Specify the type of the log file of the specific log section.

file: the log messages will be registered into a simple text file

syslog: the entries will be registered into syslog

stdout: the log will be written to the standard output

**filename = <log file name>**

If you would like the program to make log file (type=file), you can set the name of the file. Default: /var/log/vexira.log

perpid = <0|1>

This function available if type=file is set:

1: in case of making log file the program will insert the PID into the name of the log file
0: inactive

facility =

This function available if type=syslog is set:

In this option you can define which type of log messages will be considered. A type belongs to all the entries, it makes the search easier in the log file. You can specify several types separated by comma, or the ALL keyword.

For example: MAIL, USER

The following types are available:

KERN, USER, MAIL, DAEMON, AUTH, SYSLOG, LPR, NEWS, UUPC, CRON, AUTHPRIV, FTP

ident = <identifier>

This function available if type=syslog is set:

Identifier which will be placed before the log record. Default: vexira

format =

Specification of the log file's structure. The following tokens are available:

Building the date:

%d - day
%m - month
%Y - year
%H - hour
%M - minute
%S - second
%z - time zone

Other:

\$a/\$A - computer name (hostname)
\$c/\$C - component name (which created the record)
\$k/\$K - facility (\$k returns counter, \$K returns name)
\$l/\$L - log priority (Level) (\$l returns counter, \$L returns name)
\$m/\$M - log message
\$n/\$N - new line character
\$p/\$P - PID
\$t/\$T - TID
\$\$ - insert \$ character

Default: [%d/%m/%Y %H:%M:%S %z] \$k \$l \$A \$C PID:\$P TID:\$T \"\$M\"

Output rules

```
[OutputSetting/Output/RuleSetting]
[OutputSetting/Output/RuleSetting/Rule]
components=ALL
priority=DEBUG3
facility=ALL
```

The rules should be inserted in a new section inside the [OutputSetting/Output] section. There is a main rule section [OutputSetting/Output/RuleSetting] and inside this section you can create several rules in the [OutputSetting/Output/RuleSetting/Rule] section.

components = <component names separated by comma or ALL>



This section will log those messages which created by the specified component(s). You can enumerate different components, these must be separated by comma. The ALL keyword means all the components.

Available component(s): vamsi

priority = <keywords or 0..10>

Log level. Only those messages will be registered which have the equal or lower level to the specified level.

Available values:

EMERG	0	system is unusable
ALERT	1	action must be taken immediately
CRITICAL	2	critical conditions
ERROR	3	error conditions
WARNING	4	warning conditions
NOTICE	5	normal, but significant, condition
INFO	6	informational message
DEBUG0	7	debug-level message
DEBUG1	8	debug-level message
DEBUG2	9	debug-level message
DEBUG3	10	debug-level message

facility = <0..4 or keywords or ALL>

In this option you can define which type of log messages will be considered. A type belongs to all the entries, it makes the search easier in the log file. You can specify several types separated by comma, or the ALL keyword.

For example: VIRUS, SPAM

The following types are available:

ALL	0	All kind of message types
SYSTEM	1	System log message
VIRUS	2	Virus found log message
SPAM	3	Spam found log message
DEBUG	4	Debug log message

In the example

According the rule the program will register the logs messages come from one of the components and the log type is not important either. The DEBUG3 or higher level log messages will be registered.

Example 2:

```
[OutputSetting/Output/RuleSetting/Rule]
components=vamsi
priority=INFO
facility=SPAM VIRUS
```

In this case only those log messages will be logged which come from the vamsi component in case of spam- or virus found (facility=SPAM VIRUS) being on INFO or higher log level (priority=INFO).

Tokens

Tokens available in the system:

%productversion%	program's version number
%from%	from field of the mail
%to%	to field of the mail
%mailid%	value of the mail's 'message-id' field if it exists.
%vdbversion%	virus database version
%sdbversion%	spam database version



%subject% content of 'subject' field
%virusname% name of found virus
%filename% current attachment's name
%sender% e-mail address of the sender
%realip% address of the e-mail client which connected to the MTA
%recipient% e-mail address of the recipient
%mailfilename% file name and path of the copy of the original e-mail created by the
antivirus system



END USER AGREEMENT

IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS DO NOT INSTALL OR USE THE VEXIRA ANTIVIRUS SOFTWARE (referred to hereafter as the "Software"). BY CLICKING "YES", "I ACCEPT", "I AGREE", "OK", "CONTINUE", "NEXT" OR BY INSTALLING OR USING THE SOFTWARE IN ANY WAY, YOU ARE INDICATING YOUR COMPLETE UNDERSTANDING AND ACCEPTANCE OF THE TERMS AND CONDITIONS OF THIS END USER SOFTWARE LICENSE AGREEMENT (referred to hereafter as the "License"). IF YOU DO NOT ACCEPT ALL OF THE TERMS AND CONDITIONS OF THIS LICENSE, THEN CENTRAL COMMAND, INC. IS UNWILLING TO LICENSE THE SOFTWARE TO YOU. YOU MAY, WITHIN THIRTY (30) DAYS OF YOUR INITIAL PURCHASE OF A COPY OF THE SOFTWARE, RETURN THE ENTIRE COPY OF THE SOFTWARE (INCLUDING ALL COMPUTER MEDIA, PACKAGING AND DOCUMENTATION) WITH PROOF OF PURCHASE EITHER TO CENTRAL COMMAND, INC. DIRECTLY AT ITS CUSTOMER SERVICE DEPARTMENT OR TO THE RETAILER FROM WHICH YOU PURCHASED THE SOFTWARE, FOR A FULL REFUND OF THE AMOUNT INDICATED BY YOUR SALES RECEIPT OR PROOF OF PURCHASE FOR THE SOFTWARE.

IF YOU ARE INSTALLING THE SOFTWARE ON A COMPUTER THAT IS NOT OWNED BY YOU, YOU ARE BOUND TO THE TERMS AND CONDITIONS OF THIS LICENSE BOTH IN YOUR INDIVIDUAL CAPACITY AND AS AN AGENT OF THE OWNER OF THE COMPUTER, AND YOUR ACTIONS WILL BIND THE OWNER OF THE COMPUTER. YOU REPRESENT AND WARRANT TO CENTRAL COMMAND, INC. THAT YOU HAVE BOTH THE CAPACITY AND AUTHORITY TO ENTER INTO THIS LICENSE ON YOUR OWN BEHALF AS WELL AS ON BEHALF OF THE OWNER OF THE COMPUTER ON WHICH YOU ARE INSTALLING THE SOFTWARE. FOR PURPOSES OF THIS LICENSE, THE "OWNER" OF A COMPUTER IS THE INDIVIDUAL OR ENTITY THAT HAS LEGAL TITLE TO THE COMPUTER OR THAT HAS THE POSSESSORY INTEREST IN THE COMPUTER IF IT IS LEASED OR LOANED BY THE ACTUAL TITLE OWNER.

This End User License Agreement ("License") is a legal agreement between you (either an individual, agent of the owner, or a single entity end user) and Central Command, Inc. for use of the Central Command, Inc. software product identified above (i.e. Vexira Antivirus), which includes computer software and may include associated media, printed materials, and "online" or electronic documentation (collectively referred to as the "Software"), all of which are protected by U. S. copyright laws and international treaty protection. By installing, copying, or otherwise using the Software, you agree to be bound by the terms of this License. If you do not agree to the terms of this License, do not install or use the Software.

The Software and the name "Vexira Antivirus" is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The Software is licensed, not sold. If you agree to be bound by all of the terms of this License, you will only own the media on which the Software has been provided and not the Software itself.

THIRTY DAY MONEY BACK GUARANTEE: If you are the original licensee of this copy of the Software and are dissatisfied for any reason with it within the first thirty (30) days after your purchase or delivery date, you may return the complete product, together with your original proof of purchase to Central Command, Inc. or the retailer from which you purchased the Software, for a refund of the amount indicated by your original proof of purchase. If this purchase was completed using electronic delivery you are required to complete a Letter of Destruction (referred to hereafter as an "LOD") and return it within thirty (30) days after your purchase date to receive a refund. Central Command, Inc. uses the postmark or fax date of the completed and returned LOD to determine compliance. You can receive a LOD by contacting your electronic retailer from which you purchased the software or directly from Central Command, Inc. via e-mail at service@centralcommand.com, postal mail at P.O. Box 468, Medina, Ohio, 44256, or fax at +1 330-722-6517. For assistance you may also contact Central Command, Inc. by calling +1 330-723-2062 and requesting Customer Service.

GRANT OF LICENSE: Central Command, Inc. hereby grants you and only you a non-exclusive license to use the Software subject to and upon all of the terms and conditions set forth in this License.

APPLICATION SOFTWARE: You may install and use only one copy of the Software, and only on a single computer terminal.

NETWORK USE: You may also store or install a copy of the Software on a storage device, such as a network server, which is used only to install or run the Software on your other computers over an internal network; however, you must purchase and dedicate a separate license for each separate computer terminal on which the Software is installed or run from the storage device. A license for the Software may not be shared or used concurrently on different computers or computer terminals. You are required to purchase a license pack or multi-use license if you require multiple licenses for use on multiple computers or computer terminals.

If you purchase a License Pack and you have acquired this License for multiple licenses of the Software, you may make the number of additional copies of the computer software portion of the Software specified above as "Licensed copies." You are also entitled to make a corresponding number of secondary copies for use on a single home computer as specified above in the section entitled "Application Software".

If you purchase a License for the Software to be used to virus scan electronic messages or you install the Software in such a way to virus scan electronic messages you are required to purchase a license for each domain name and each sub domain name that is virus scanned. If your total electronic mail addresses exceed 6000 you are required to purchase a special Internet



Vexira Antivirus for Mail Server + i

Service Provider (ISP) License for use of the Software.

TERM OF LICENSE: The License granted hereunder shall commence on the date that you install, copy or otherwise first use the Software. You may terminate this License at any time. This License shall terminate automatically (and you shall have no right to use the Software) upon your breach of any term of this License. Upon termination, you must destroy the Software and all copies, if any, you made pursuant to this License.

UPGRADES: If the Software is labeled as an upgrade, you must be properly licensed to use a product identified by Central Command, Inc. as being eligible for the upgrade in order to use the Software. A copy of the Software labeled as an upgrade replaces and/or supplements the product that formed the basis for your eligibility for the upgrade. You may use the resulting upgraded product only in accordance with the terms of this License. If the Software is an upgrade of a component of a package of software programs that you licensed as a single product, the Software may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

COPYRIGHT: All right, title and interest in and to the Software and the name "Vexira Antivirus" and "Vexira" and all copyright rights in and to the Software (including but not limited to any images, photographs, animations, video, audio, music, text, and "applets" incorporated into the Software), the accompanying printed materials, and any copies of the Software are owned by Central Command, Inc. and/or its suppliers. The Software is protected by copyright laws and international treaty provisions. Therefore, you must treat the Software and the term "Vexira Antivirus" or "Vexira" like any other copyrighted material except that you may install the Software on a single computer provided you keep the original solely for backup or archival purposes. You may not copy the printed materials accompanying the Software. You may not use the name "Vexira Antivirus" or "Vexira" or any similar name except when referring to the Software. You must produce and include all copyright notices in their original form for all copies created irrespective of the media or form in which the Software exists. You may not sub-license, rent, sell, or lease the Software. You may not reverse engineer, recompile, disassemble, create derivative works, modify, translate, or make any attempt to discover the source code for the Software or any part thereof. Except as expressly permitted by applicable law, you may not remove from the Software, or alter, any of the trademarks, trade names, logos, patent or copyright notices or other proprietary rights notices or markings, or add any other notices or markings to the Software.

LIMITED WARRANTY: Central Command, Inc. warrants that the media on which the Software is distributed is free from defects for a period of thirty (30) days from your date of receipt or purchase date of the Software. Your sole remedy for a breach of this warranty will be that Central Command, Inc. at its option, may replace the defective media upon receipt of the damaged media, or refund the money you paid for the Software. Central Command, Inc. does not warrant that the Software will be uninterrupted or error free or that the errors will be corrected. Central Command, Inc. does not warrant that the Software will meet your requirements. CENTRAL COMMAND, INC. HEREBY DISCLAIMS ALL OTHER WARRANTIES FOR THE SOFTWARE, WHETHER EXPRESSED OR IMPLIED. THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESSED OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

DISCLAIMER OF DAMAGES: Anyone installing, using, testing, or evaluating the Software bears all risk to the quality and performance of the Software. In no event shall Central Command, Inc. be liable for any damages of any kind, including, without limitation, direct, indirect, exemplary, special, consequential or incidental damages of any kind (including without limitation lost profits or damage to other systems) arising out of the use, performance, or delivery of the Software, even if Central Command, Inc. has been advised of the existence or possibility of such damages. SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU. IN NO CASE SHALL CENTRAL COMMAND, INC.'S LIABILITY EXCEED THE PURCHASE PRICE PAID BY YOU FOR THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether you accept or use, evaluate, or test the Software.

IMPORTANT NOTICE TO USERS: THE SOFTWARE IS NOT FAULT-TOLERANT AND IS NOT DESIGNED OR INTENDED FOR USE IN ANY HAZARDOUS ENVIRONMENT REQUIRING FAIL-SAFE PERFORMANCE OR OPERATION. THIS SOFTWARE IS NOT FOR USE IN THE OPERATION OF AIRCRAFT NAVIGATION, NUCLEAR FACILITIES, OR COMMUNICATION SYSTEMS, WEAPONS SYSTEMS, DIRECT OR INDIRECT LIFE-SUPPORT SYSTEMS, AIR TRAFFIC CONTROL, OR ANY APPLICATION OR INSTALLATION WHERE FAILURE COULD RESULT IN DEATH, SEVERE PHYSICAL INJURY OR PROPERTY DAMAGE.

GOVERNMENT RESTRICTED RIGHTS/RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software-Restricted Rights clause at 48 CFR 52.227-19, as applicable. Contact Central Command, Inc., at P.O. Box 468, Medina Ohio 44258-0468.

GENERAL: This License is deemed delivered in, and will be governed by, the laws of the State of Ohio, in the United States of America. This License may only be modified by a license addendum, which must accompany this License or by a written document which has been signed by both you and Central Command, Inc. This License has been written in the English language only and is not to be translated or interpreted in any other language. Prices, costs and fees for use of the Software are subject to change without notice to you. In the event of invalidity of any provision of this License, the invalidity shall not affect the validity of the remaining portions of this License. Vexira, Vexira logo, Central Command, Central Command's logo, EVRT, Emergency Virus Response Team, Without us, there's no defense, are trademarks of Central Command, Inc. Microsoft,



Windows, Excel, Word, the Windows logo, Windows NT, Windows 2000 are registered trademarks of Microsoft Corporation. All other trademarks or tradenames are the property of their respective owners.

CONTACT

This manual provides comprehensive information on operational of our virus protection product. If you have any additional questions about it or would like to share your experience or proposals with us do not hesitate to contact us! Turn to us with confidence, your demands and remarks will be respected.

Address Central Command, Inc.
Medina, Ohio 44258,
P. O. Box 468.
United States

Phone (+1) 330 723 2062
Fax (+1) 330 722 6517
Web www.centralcommand.com
E-mail sales@centralcommand.com
support@centralcommand.com