AVG 2012 for Linux/FreeBSD

User Manual

Document revision 1.0 (16. 4. 2012)

Contents

1. Introduction

AVG 2012 Anti-Virus for Linux/FreeBSD is a free anti-malware suite for Linux and FreeBSD systems. Its main purpose is to detect and treat many kinds of malware like viruses, worms, trojans etc.

This User Manual describes both basic tasks that a user of AVG for Linux would need to do, such as installation, scanning and update, accessing and treating infected files, and some advanced tasks, such as setting up virus checking for an e-mail server, running virus-protected file server for heterogenous network and configuring automatic infection treatment.

As there is already comprehensive documentation provided within the AVG for Linux installation package, this documents does not aim to be totally descriptive; it rather gives the basics and offers advice on where to get further information.

1.1. Prerequisities

AVG for Linux is more or less distribution independent, and should be able to run on any recent x86 compatible hardware (for x64, the compatible versions of the 32-bit libraries are needed). The minimum requirements are:

- **CPU**: i686 or amd64, 800 MHz
- **RAM**: 512 MB (1 GB recommended)
- **Free harddisk space**: 500 MB

2. Installation

2.1. Installation Package

The installation package can be obtained from free.avg.com/download. Multiple formats are available; the choice depends on your distribution package management system. You can either use a dedicated installation package (for distributions using the RPM Package Manager or the Debian package system), or a universal package:

- Red Hat and distributions based on Red Hat (Fedora, SUSE, Mandriva, etc.): **rpm**
- Debian and distributions based on Debian (Ubuntu, Kanotix, etc.): **deb**
- Universal packages for all GNU/Linux distributions (mentioned above and others: Slackware, Gentoo, etc.): **sh** or **tar.gz**
- FreeBSD: **tar.gz**

All packages are named **avg2012{edition}-r{release number}-a{virus database version}.{platform}.{package type}**, for example "avg2012lms-r1234-a5678.i386.deb". Edition depends on license type: for full/trial product version it is **lms** (Linux) or **fms** (FreeBSD), for free product version it is **flx** (Linux) or **ffb** (FreeBSD).

Please note it is not within our limitations to test every version of every Linux distribution with the packages we provide, however we do test the major distributions in recent versions to run with our software smoothly.

2.2. How to Install

For installation from the local source, you can use a dedicated package manager such as YUM, Zypper, up2date, YaST (for .rpm packages), apt/aptitude, Synaptic (for .deb packages), etc.

For installation via your shell (terminal), switch to the directory that contains the downloaded package and use the appropriate command as follows:

rpm file:

```
# rpm -i avg2012lms-r{release}-a{version}.{platform}.rpm
```

deb file:

```
# dpkg -i avg2012lms-r{release}-a{version}.{platform}.deb
```

sh file:

```
# chmod +x avg2012lms-r{release}-a{vdb version}.{architecture}.
sh
```

```
# ./avg2012lms-r{release}-a{vdb version}.{architecture}.sh
```

tar.gz file:

```
# tar xzvf avg2012{edition}-r{release}-a{vdb version}.
{architecture}.tar.gz

# cd avg2012{edition}-r{release}-a{vdb version}.{architecture}

# ./install.sh
```

The installation process performs all necessary steps automatically. Once the installation is complete, you might need to register your AVG; see the chapter License Registration for details.

2.3. License Registration

During the installation process, AVG for Linux should be automatically registered using a free or trial license number provided within the package. If you purchase the product and obtain a full license number, you will need to register it manually, using the following command in your shell:

```
# avgctl --register <your license number>
```

To check the registered license info, use:

```
# avgctl --licinfo
```

2.4. Upgrading from v2011

Any previous version of AVG installed on your computer should have been upgraded to the current version AVG 2012 Anti-Virus for Linux/FreeBSD automatically, via update. If it has not, you can do it manually by using the following command in your shell:

```
# avgupdate --priority 5
```

3. Basic Usage

All functions of AVG for Linux are performed by **AVG daemons**; these can be controlled by **command-line tools**. The AVG daemons are launched upon system boot by the **init** script located here:

- **Linux:** /etc/init.d/avgd
- **FreeBSD:** /usr/local/etc/rc.d/avgd.sh

The main command-line tool **avgctl** can perform basic operations on AVG:

Starting AVG:

```
# avgctl --start
```

Alternatively, by using the init script:

On Linux: `# /etc/init.d/avgd start`

On FreeBSD: `# /usr/local/etc/rc.d/avgd.sh start`

Stopping AVG:

```
# avgctl --stop
```

Restarting AVG:

```
# avgctl --restart
```

Showing AVG statistics (`installed components/services and what state they are in`):

```
# avgctl --stat-all
```

Note: For any task or action on AVG, it is necessary that the daemons are running. They should launch automatically after AVG installation and upon each computer restart, however in case of any doubt, we recommend that you check the statistics using the command above to make sure.

Displaying documentation for a specific daemon or command-line tool (exit by Q):

```
man <name of a daemon or command-line>
```

For the complete overview of all AVG daemons and command-line tools, please refer to the README file located in **/opt/avg/av/doc**.

Useful common parameters for all commands:

```
--help

--version
```

Please note that for running any AVG command (except AVGSCAN), you will need to have root/superuser rights, or use the sudo command. Any launched action can be cancelled by Ctrl+C; only critical actions cannot be cancelled (for example an update at the time of changing configuration) and forced abort is not recommended.

3.1. Event Log

Useful information about all important AVG events can be also obtained via logs intended for common users, accessible via **avgevtlog** command-line. The logs contain well arranged data about running of AVG such as start of AVG daemons, performed updates, run scans, etc.

You can view or delete the log, save it to file, and filter out events from specific dates or components. For example, to view all basic events such as start of daemons, you need to filter out messages from the AVG WatchDog:

```
$ avgevtlog --source=WD
```

If you want to be thoroughly informed on regular basis, we recommend that you set up a task in **cron** to send you the log via e-mail daily or weekly. To learn more about cron and how to use it, please refer to its man pages:

```
$ man crontab
```

Example of a cron task:

```
PATH=/bin:/usr/bin:/usr/local/bin

SHELL=/bin/bash

MAILTO=user@localhost


0 23 * * * avgevtlog -D $(date +\%F) -s Update
```

This task will be launched every day at 11 p.m. and send log messages about AVG updates run on that day to the local mailbox. It can be saved to a file and added to crontab by using:

```
$ crontab <file>
```

3.2. Updating

Updates are critical for the AVG for Linux correct functioning, and as such should be performed on regular basis. There are predefined update schedules for this purpose, which you can adjust to your needs; see the sub-chapter Scheduling Updates. Sometimes, you might also need to run an update manually, to make sure that you have the latest protection available; see the sub-chapter Manual Update.

3.2.1. Manual Update

Checking for available updates, without running an actual update:

```
# avgupdate --check
```

Running an update (launching the **avgupd** daemon):

```
# avgupdate
```

By default, a complete program update (level 4) will be performed by this command. Other available update levels are as follows:

1 – **Critical update:** Update of the virus database containing definitions of a new aggressive virus currently spreading; should be applied at once

2 – **Virus update:** Update of the virus database

3 – **Recommended update:** Update of the virus database and the most important program changes

4 – **Program update** (default): Complete update of the virus database and program changes

5 – **Optional update:** Complete update of the virus database and all program changes including minor improvements

The most commonly released updates are of priority 2 and 3. To run an update of the selected level (for example level 2, AVI):

```
# avgupdate --priority 2
```

Please note that update of the Anti-Spam rules is not a part of any predefined update levels and therefore must be run separately!

Running an update of the Anti-Spam rules:

```
# avgupdate --antispam
```

Downloading update files to a local folder and using it for updating:

```
# avgupdate --download --path="/path/to/the/folder"

# avgupdate --source=folder --path="/path/to/the/folder"
```

Listing the complete overview of parameters for configuring the update:

```
# avgupdate --help
```

During any update, you will see progress details, and information about successful or unsuccessful finish will be shown at the end.

3.2.2. Scheduling Updates

The following predefined update schedules are available:

- **Virus update** (AVI) – update of the virus database. By default enabled, starts every four hours. If the scheduled time is missed, the task is launched after computer startup (delay 3 minutes).
- **Anti-Spam update** – update of the Anti-Spam rules. By default enabled, starts every two hours. If the scheduled time is missed, the task is launched after computer startup (delay 6 minutes).
- **Program update** – complete program update. By default disabled. If enabled, starts by default every day at 8 a.m. If the scheduled time is missed, the task is launched after next computer startup (delay 5 minutes).

Please note that the Anti-Spam update is not applicable for the free version of AVG for Linux (i.e. using free license).

Settings of the update tasks can be changed via the **avgcfgctl** command-line. To view all configuration items of the scheduled updates including current values, use:

```
$ avgcfgctl sched
```

To change an item, use parameter -w for writing. For example, to enable Program update, use:

```
$ avgcfgctl -w UpdateProgram.sched.Task.Disabled=false
```

All available configuration parameters are as follows ({update type} stands for "UpdateVir", "UpdateAspam", or "UpdateProgram" as described above):

{update type}.sched.Task.Disabled=[BOOL]

Specifies whether the task is currently disabled or not (true/false).

{update type}.sched.Task.StartType=[DWORD]

0 – starts the task only once at sched.Times.StartTime.

1 – starts the task repeatedly, every sched.Repeat.Interval units of sched.Repeat.Type.

2 – starts the task every day at sched.Times.StartTime.

3 – starts the task every week on sched.Times.DayOfWeek, at sched.Times.StartTime.

4 – starts the task at system startup, specifically, at AVG startup (not after restore from hibernation etc.).

5 – starts the task every month on sched.Times.DayOfMonth day, at sched.Times.StartTime.

6 – starts the task only on selected days, sched.Times.SelectedDays, at sched.Times.StartTime.

{update type}.sched.Times.StartTime=[DATE]

The [DATE] type values are in format YYYY-MM-DD/hh-mm-ss.

{update type}.sched.Repeat.Interval=[DWORD]

Specifies amount of units in sched.Repeat.Type.

{update type}.sched.Repeat.Type=[DWORD]

Specifies type of units for sched.Repeat.Interval (0 for minutes, 1 for hours).

{update type}.sched.Times.DayOfMonth=[DWORD]

Specifies a day of month when the task should start (1–31).

{update type}.sched.Times.DayOfWeek=[DWORD]

Specifies a day of week when the task should start (0–6; 0 for Sunday, 6 for Saturday).

{update type}.sched.Times.SelectedDays=[DWORD]

Specifies selected days of week (more than one) when the task should start, as a mathematical addition of the following values:

1 – Sunday

2 – Monday

4 – Tuesday

8 – Wednesday

16 – Thursday

32 – Friday

64 – Saturday

For example, if the task should be run on Monday, Tuesday and Friday, value 38 should be used (2+4+32).

{update type}.sched.Task.MissedStartAction=[DWORD]

Specifies what to do if the start time of the task has been missed:

0 – missed task will be ignored.

1 – missed task will be started at the nearest computer startup, with a delay specified by sched.Times.GracePeriod.

{update type}.sched.Times.GracePeriod=[DWORD]

Specifies a delay after system startup before the task should be launched (in seconds).

3.3. Scanning

AVG for Linux offers on-demand scanning with extensive setting options to match your needs. Please remember that before running a scan, it is well advised to perform an update (at least the Virus update) to make sure that your AVG contains definitions of the latest threats; the chapter Updating will give you more details.

Please note that this chapter only describes on-demand, i.e. manually run scan. To learn more about the automatic resident scanning service, please refer to the chapter On-Access Scanner, and concerning E-mail scanning, please see the chapter E-mail Scanner.

All scans are run multi-thread and independent, therefore it is possible to run more than one scan at a time. However, it is not a way to increase scanning performance. Also please note that if there is no shared memory available, each parallel running scan process will increase the memory usage; it is highly recommended that shared memory is enabled on the system.

Setting up shared memory on FreeBSD (otherwise prone to problems with running AVG processes):

```
# sysctl kern.ipc.shmmax=134217728

# sysctl kern.ipc.shmall=134217728
```

To see complete and detailed instructions on how to set the shared memory limits on various systems, please use:

```
# man avgavid
```

3.3.1. On-demand Scanning

Running a scan (launching the **avgscand** daemon):

```
$ avgscan <path>
```

This command will run a scan of the whole computer (except for /sys and /proc folders). By default, the command performs the most basic scan for viruses. If you

want to enhance the scan by any additional settings, you will need to enter all necessary parameters manually. *Please note that the used settings cannot be saved for future use.* To list the complete overview of parameters for configuring a scan, please use:

```
$ avgscan --help
```

If you want any detected infections to be dealt with automatically, make sure that you use the corresponding parameters.

Generally, the minimum recommended settings for a manual scan are as follows – automatic healing, making backup copies in the Virus Vault (see the chapter Virus Vault), using heuristic analysis and detecting potentially unwanted programs:

```
$ avgscan --heur --heal --vv-backup -pup
```

During a scan, you will see the files currently being scanned and reported files (ones that could not have been scanned, and optional reports such as files containing macros, files with hidden extension, password-protected files, etc.). A running scan can be aborted by Ctrl+C. Overall scan result (statistics) will be shown at the end, and will look like this:

Files scanned	: total number of processed files (successfully scanned)
Infections found	: total number of detected infections
PUPs found	: number of Potentially Unwanted Programs, suspicious files or spyware
Files healed	: number of healed files, i.e. successfully restored to the same state as before infection
Warnings reported	: number of other detections (not infections), e.g. password-protected files, hidden extensions, etc.
Errors reported	: number of files that could not be scanned

A scan log is not created by default. If you want a complete scan report, you need to run the scan with the following parameter:

```
$ avgscan --report=<filename>
```

After the scan, the report will be saved to a file with the filename specified.

3.3.2. Scheduling Scans

Scheduling scans is currently not supported in AVG 2012 Anti-Virus for Linux/FreeBSD. However, you can use the **cron** scheduler for this purpose: a program integrated by default in GNU/Linux systems, allowing you to schedule jobs to run at specific dates and times. To learn more about cron and how to use it, please refer to its man pages:

```
$ man crontab
```

Example of an AVG scan scheduled in cron:

```
PATH=/bin:/usr/bin:/usr/local/bin

SHELL=/bin/bash


0 23 * * * avgscan --heur --arc --coo --pup --pup2 --
report=$HOME/scan_reports/avgscan_$(date +\%s).rep $HOME
```

This task will be launched every day at 11 p.m. Home directory will be scanned, including archives, tracking cookies and potentially dangerous programs (spyware and similar questionable categories of malware). Heuristic analysis will be used for scanning (a simulation method allowing detection of unknown threats). A full scan report will be generated and saved to $HOME/scan_reports/avgscan_<timestamp>.rep.

The task can be saved to a file and added to crontab by using:

```
$ crontab <file>
```

3.4. Handling Infection

If there is infection detected on your computer, AVG for Linux will do nothing as automatic actions are by default disabled (it can be changed in the configuration for various scan types individually). Generally, all detected threats should be moved to the **Virus Vault**. You can then decide how to process the infected objects further – restore to the original location (if it turns out that the file is actually harmless, or if it is needed by another program), or delete forever (if it is not possible to heal the file, or if you no longer need it). Please refer to the sub-chapter Virus Vault for further details.

This document cannot fully cover removal of all different types of viruses, many of which need special treatment. If you need help or more detailed information about a specific threat, please refer to our online resources:

- **General info about viruses**: http://free.avg.com/about-viruses
- **AVG's online Virus Encyclopedia**: http://www.avgthreatlabs.com/webthreats/
- **FAQ section**: free.avg.com/faq -> group "Virus FAQ"
- **Forums (followed by our professional support staff)**: forums.avg.com/avg-forums -> topic "Virus Removal, Tools for Removing"
- **Targeted virus removal tools**: http://free.avg.com/cz-en/virus-removal

3.4.1. Virus Vault

AVG Virus Vault is a virus quarantine where infected files and backup copies of healed files can be safely stored. Virus Vault is managed via **avgvvctl** command-line.

There are generally two vaults used by AVG: **user** vault is used by on-demand scans and is placed in user's home directory, **system** vault is used by on-access scanner and e-mail scanner. Location of the vaults is managed via **avgcfgctl** command-line; to print the configuration items and the current path to the vaults, use:

```
$ avgcfgctl vv
```

The default is the **user** vault. To switch to the **system** vault, use:

```
$ avgvvctl --system-vault
```

Printing all items currently stored in the vault:

```
$ avgvvctl list
```

Restoring an item from the vault to its original location (an identification number must be specified):

```
$ avgvvctl restore --item-id <ID>
```

Deleting an item from the vault (an identification number must be specified):

```
$ avgvvctl remove --item-id <ID>
```

Deleting the complete contents of the vault:

```
$ avgvvctl remove-all
```

Please note that deleting items from a vault is an irreversible action.

To see the complete list of commands for available actions on the vault, use the common --help parameter.

4. On-access Scanner

AVG's on-access scanner is a resident protection that can guard the system continuously, by scanning all accessed files automatically in the background. On-access scanning can be arranged in the following ways:

1. By using **Fanotify** – see the chapter Fanotify (strongly recommended on Linux kernels 2.6.38 and later).

2. By integrating a special module in your Linux kernel – see the chapter Installing Kernel Modules (on Linux kernels up to 2.6.37).

3. By setting up on-access scanning on the **Samba** system – see the chapter Protecting Samba.

4.1. Fanotify

On Linux kernels version 2.6.36 and later, the best option for operating on-access scanning is the **Fanotify** notification system. On virtually all kernels version 2.6.38 and later, it is supported and enabled by default. If unsure, you can check the config file in the /boot directory for the following: CONFIG_FANOTIFY=y, CONFIG_FANOTIFY_ACCESS_PERMISSION=y.

To install on-access scanner with Fanotify, we recommend using a dedicated configuration wizard available newly in AVG 2012 Anti-Virus for Linux/FreeBSD:

```
# avgsetup

Enter "O" for deployment of the On-Access scanner.

Enter "F" for using Fanotify, and follow instructions on
screen.
```

4.2. Installing Kernel Modules

If you run somewhat aged Linux kernel that does not support Fanotify, you can make on-access scanning work by integrating a special module in the kernel. Please note that it is an advanced task that requires re-building your Linux system kernel, and it is therefore also necessary that the kernel supports module loading. Also, you will need the code of the kernel and installed header files. Depending on your Linux kernel version, you will then need to get one of the following supported modules:

- **RedirFS Anti-Virus Filter** – Linux kernel version 2.6.25–2.6.37
- **DazukoFS** kernel module – Linux kernel version up to 2.6.22
- **Dazuko** kernel module – FreeBSD up to 8.x

Note: Dazuko is an older version of DazukoFS and eventually, it is going to be discontinued.

To install the selected module, we recommend using a dedicated configuration wizard available newly in AVG 2012 Anti-Virus for Linux/FreeBSD:

```
# avgsetup

Enter "O" for deployment of the On-Access scanner, and follow
instructions on screen.
```

If for any reason you wish to install a module manually, you can find detailed instructions in the corresponding sub-chapter. Should you need detailed technical information, please refer to the **/opt/avg/av/doc/README.oad** file and to the documentation of your operating system. Please note that the **avgd** daemon must be running while installing a module.

4.2.1. RedirFS

To install the **RedirFS Anti-Virus Filter**, please first download the **redirfs** and **avflt** packages from http://www.redirfs.org and unpack them:

```
$ tar xzvf redirfs-x.y.tar.gz

$ tar xzvf avflt-x.y.tar.gz
```

To compile and install **redirfs**, switch to root using the su command and use:

```
# cd redirfs-X.y

# make -C /lib/modules/`uname -r`/build M=`pwd` modules

# make -C /lib/modules/`uname -r`/build M=`pwd` modules_install
```

Then compile and install **avgflt**:

```
# cd ../avflt-X.y

# cp <path to the redirfs-x.y directory>/Module.symvers ./

# make -C /lib/modules/`uname -r`/build M=`pwd` EXTRA_CFLAGS=-
I<full path to the redirfs> modules

# make -C /lib/modules/`uname -r`/build M=`pwd` EXTRA_CFLAGS=-
I<full path to the redirfs> modules_install
```

After successful installation, update and load both modules:

```
# depmod -a

# modprobe avflt

# modprobe redirfs
```

If you have changed the default configuration of AVG, please adjust the **Default.oad.avflt.paths.include** option, for example:

```
# avgcfgctl -w Default.oad.avflt.paths.include="|/home|"
```

And finally, restart the on-access daemon:

```
# avgctl --restart=oad
```

If you can adjust the startup options, it is recommended that you set the modules to load automatically at system startup, for example by adding **redirfs** and **avgflt** to the **/etc/modules** file.

4.2.2. DazukoFS

To install the **DazukoFS** module, please download the latest version from http://www.dazuko.org and unpack it:

```
# tar xzvf dazukofs-X.y.z.tar.gz
```

Before compiling the module, you will need to switch to root using the su command, and then you might need to patch the DazukoFS sources to match your kernel version precisely. This can be done using patches from the **dazukofs-x.y.z/patches** subfolder. For example, if you are using kernel 2.6.26, you will need to patch it with the corresponding patch as follows:

```
# patch -p1 < patches/patch-linux-2.6.26
```

Now you can compile, install and load the module:

```
# cd dazukofs-X.X.X

# make

# make dazukofs_install

# depmod -a

# modprobe dazukofs
```

To adjust AVG configuration accordingly, please use:

```
# avgcfgctl -w Default.oad.use=dazukofs
```

Then you need to restart the on-access daemon:

```
# avgctl --restart=oad
```

Finally, enable on-access scanning on the directories you want to protect:

```
# mount -t dazukofs /<folder name> /<folder name>
```

Please note that this setting is only valid until next computer restart.

If you want DazukoFS to be mounted over certain directories at computer startup, you will need to add the mounts to the end of **/etc/fstab**. The **dazukofs** module must be loaded in order for this to work.

Please note that is not possible to stack DazukoFS over the root filesystem (/).

Stacking over pseudo filesystems (/proc, /dev, /sys) has not been tested and should be avoided. The kernel will crash if you attempt to mount DazukoFS to a file instead of a directory (various kernel patches handle this error).

4.2.3. Dazuko

Please note that Dazuko is only recommended for FreeBSD systems.

To install the **Dazuko** module, please download the latest version from http://www.dazuko.org and unpack it:

```
# tar xzvf dazuko-X.y.z.tar.gz
```

After switching to root using the su command, you can compile, install and load the module:

```
#cd dazuko-X.X.X

# ./configure

# make

# make install

# depmod -a

# modprobe dazuko
```

To adjust AVG configuration accordingly, please use:

```
# avgcfgctl -w Default.oad.dazuko.paths.include="|/home|"

# avgcfgctl -w Default.oad.use=dazuko
```

Then you can restart the on-access daemon:

```
# avgctl --restart=oad
```

More information on compilation and installation of Dazuko can be found at http://dazuko.dnsalias.org/wiki/index.php/Installation_HOWTO.

4.3. Protecting Samba

If you use Samba (version 3.3.0 or later) on your file system, you can integrate it with AVG for Linux to be used for on-access scanning of files accessed through Samba. A special AVG module **avgsambamodule.so** is used for that purpose.

Note: On-access scanning on Samba is an independent way of protecting files you do not need to install any modules to make it functional, and no Linux kernel modification is required.

The AVG Samba modules (plugins) for various systems can be obtained from https://share.avg.com/linux/sambamodule.

To install a plugin, unzip the downloaded archive, switch to root using the su command, and run the installation script:

```
# ./install.sh
```

After successful installation, you will need to configure AVG, Samba and the Samba plugin according to the **avgsambamodule.conf** configuration file.

First, enable support for the plugin:

```
# avgcfgctl -w Default.tcpd.avg.
samba_plugin_support_enabled=true
```

If a Unix socket is used, it must be enabled in the AVG configuration, and the path to the socket must match:

```
# avgcfgctl -w Default.tcpd.avg.socket=avg-unix-socket

# avgcfgctl -w Default.tcpd.avg.use_socket=true
```

Optionally, you can also enable parallel running of the Unix socket and the TCP protocol (allows concurrent Samba and e-mail scanning):

```
# avgcfgctl -w Default.tcpd.avg.samba_plugin_socket=true
```

Then you will need to modify the **/etc/samba/smb.conf** file.

First, add the name and full path to the avgsambamodule.conf file. If you keep the default name and location of the configuration file, add the following line:

```
avgsambamodule: config-file = /etc/samba/avg/avgsambamodule.
conf
```

If a Unix socket is used, also add:

```
unix socket path = /opt/avg/av/var/run/avg-unix-socket
```

To use the AVG Samba plugin as the default for all shares, add the following to the [global] section of the file:

```
[global]

vfs objects = avgsambamodule

avgsambamodule: config-file = /etc/samba/avg/avgsambamodule.
conf
```

To use the AVG Samba plugin for a particular share only (for example "common"), add the following to the end of the file:

```
[common]

path = <shared directory path>
```

```
vfs objects = avgsambamodule

avgsambamodule: config-file = /etc/samba/avg/avgsambamodule.
conf

writeable = yes

browseable = yes

valid users = <valid users>
```

Please note that you need to add this for each share that you want to be protected by AVG on-access scanning.

Finally, you need to restart all related services:

```
# avgctl --restart

# sudo service smbd stop

# sudo service smbd start
```

To obtain more details, please refer to /opt/avg/avg/doc/README.samba and the Samba documentation.

4.4. Configuration

Settings of the On-Access Scanner can be changed via the **avgcfgctl** command-line. To view all configuration items, use:

```
$ avgcfgctl oad
```

To change an item, use parameter -w for writing. For example, to enable automatic actions on detected infections, use:

```
$ avgcfgctl -w Oad.scan.AutomaticActions.Enabled=true
```

Available parameters and the corresponding default values are as follows:

Oad.scand.maxscanproc=2

The size of the scanning thread pool.

Oad.scan.Options.ParanoidMode=false

Enable the "paranoid scan mode" that includes very detailed scanning, for example for old viruses or browser flaws that have long been fixed, etc. Please note that this mode makes heavy demands on system resources.

Oad.scan.AutomaticActions.Enabled=false

Enable automatic actions on infected files (healing or moving to the Virus Vault, if healing is not possible).

Oad.scan.AutomaticActions.PreferedAction=0

The preferred automatic action to be performed on infected files when scan. AutomaticActions.Enabled is set to "true":

1 – Heal

2 – Delete (without making a backup copy in the Virus Vault)

4 – Move to the Virus Vault

Oad.scan.AutomaticActions.BackupInVault=true

If an infected object is going to be deleted by an automatic action, first create a backup copy of it in the Virus Vault.

Oad.scan.Options.ArchiveLevel=32

Define the level of processing archives (resident, default):

0 – no archives, macros, cookies, real–time compression will be scanned, including MIME

32 – only macros, cookies, real–time compression will be scanned

256 – archives, macros, and cookies will be scanned

Oad.scan.Options.ReportArchiveBombs=false

Enable detection of malicious archive bombs. Upon decompression, these highly compressed archive files take up all available disk space and memory, and as a result, they bring down the application accessing them or the whole system.

5. E-mail Scanner

The AVG daemon for e-mail scanning is **avgtcpd**. It supports SMTP and AVG protocols and Milter library, and is compatible with the following e-mail servers:

Open Source

- Sendmail – http://www.sendmail.org
- Postfix – http://www.postfix.org
- Qmail – http://cr.yp.to/qmail.html and http://www.lifewithqmail.org
- Exim – http://www.exim.org

Other

- AMaViS – http://www.amavis.org

Setting up AVG on Sendmail and Postfix is described in the following sub-chapters for illustration. Also you can use a dedicated configuration wizard for this task:

```
# avgsetup

Enter "M" for deployment of the E-Mail scanner, and follow
instructions on screen.
```

To learn more about a specific e-mail server, please refer to the respective **/opt/avg/av/doc/README.<mailserver>** file.

5.1. Deployment

5.1.1. Postfix

To integrate AVG with Postfix, you will first need to edit the **/etc/postfix/main.cf** file. Add the following lines:

```
content_filter = avgtcpd:localhost:54321

receive_override_options = no_address_mappings
```

Note: The socket address "localhost:54321" is default.

Then edit the **/etc/postfix/master.cf** file by adding:

```
# ==========================================================

# service type private unpriv  chroot  wakeup  maxproc command

#              (yes)   (yes)   (yes)   (never) (100)

# ==========================================================
```

```
avgtcpd   unix  -       -       y/n     -       2       smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes
    -o disable_dns_lookups=yes
    -o max_use=20
```

Change the 'y/n' to either 'y' or 'n'. Check your regular smyp and smtpd service. Then continue adding:

```
# ==========================================================
# service type private unpriv  chroot  wakeup  maxproc command
#              (yes)   (yes)   (yes)   (never) (100)
# ==========================================================
localhost:10025 inet n -       n       -       10      smtpd
    -o content_filter=
    -o receive_override_options=no_unknown_recipient_checks,
no_header_body_checks
    -o smtpd_helo_restrictions=
    -o smtpd_client_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o mynetworks=127.0.0.0/8
    -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

If you run postfix version 2.3 and later, modify item **receive_override_options** by adding "no_milters":

```
-o receive_override_options=no_unknown_recipient_checks,
no_header_body_checks, no_milters
```

Note: The socket address "localhost:10025" is default.

To complete the deployment, you will need to make some configuration adjustments. For common settings, please refer to the chapter Configuration. To configure the **avgtcpd** service for Postfix specifically, use:

Default.tcpd.smtp.enabled=true

Enable smtp functions (Postfix related).

Default.tcpd.milter.enabled=false

For milter library; must be disabled if Default.tcpd.smtp.enabled is set to "true".

Default.tcpd.smtp.address=127.0.0.1

Default.tcpd.smtp.ports=|54321|

Define address and port for the AVG server service.

Default.tcpd.smtp.client_address=127.0.0.1

Default.tcpd.smtp.client_port=10025

Define address and port for the Postfix client service.

Default.tcpd.smtp.limiter_start=220

Default.tcpd.smtp.limiter_stop=250

These items control active connections.

Default.tcpd.smtp.read_buffer=102400

Define SMTP read buffer size.

5.1.2. Sendmail

To integrate AVG with Sendmail, you will first need to edit the configuration file. There are two options:

A) Edit and generate configuration: "mc file" (recommended).

B) Edit configuration directly: "cf file".

A)

Edit the **/etc/mail/sendmail.mc** file by adding the following to the end of the file:

```
INPUT_MAIL_FILTER('avgtcpd', 'S=inet:10024@localhost, F=T,
T=S:1m;R:1m;E:10m')
```

Re-create the cf file. Specific steps depend on your system; typically, you will use:

```
# make -C /etc/mail

# cd /etc/mail
```

```
# make sendmail.cf

# cd /etc/mail

# m4 /usr/share/sendmail.cf/m4/cf.m4 sendmail.mc >sendmail.cf
```

B)

Edit the /etc/mail/sendmail.cf file by adding the following below the section INPUT MAIL FILTERS:

```
# Input mail filters

O InputMailFilters=avgtcpd
```

Then add the following below the section MAIL FILTER DEFINITIONS:

```
##################################################################

##################################################################

#####

#####              MAIL FILTER DEFINITIONS

#####

##################################################################

##################################################################

Xavgtcpd, S=inet:1024@localhost, F=T, T=S:1m;R:1m;E:10m
```

Or, just add the following lines to the end of your cf file:

```
# AVG Input mail filter

O InputMailFilters=avgtcpd

# AVG Mail filter definitions

Xavgtcpd, S=inet:10024@localhost, F=T, T=S:1m;R:1m;E:10m
```

To complete the deployment, you will need to make some configuration adjustments. For common settings, please refer to the chapter Configuration. To configure the **avgtcpd** service for Sendmail specifically, use:

Default.tcpd.smtp.enabled=false

Disable smtp functions.

Default.tcpd.milter.socket=inet:10024@localhost

Define socket for communication with AVG.

Default.tcpd.milter.enabled=true

Enable and define milter interface.

Default.tcpd.milter.verbosity=0

Optional; sets milter logging verbosity (0 – disabled, 6 – highest severity).

5.2. Configuration

Common settings for the E-mail Scanner can be changed via the **avgcfgctl** command-line. To view all configuration items, use:

```
$ avgcfgctl tcpd
```

To change an item, use parameter -w for writing. To enable all e-mail functions, use:

```
$ avgcfgctl -w Default.setup.features.tcpd=true
```

All available parameters and the corresponding default values are listed and explained in detail in the following sub-chapters. You can also refer to the **avgtcpd** man pages.

Please note that after completing the configuration, you will need to restart all related services (AVG and the mail server).

5.2.1. Scanning

Tcpd.scan.DirOptions.ScanAllFiles=true

Enable including or excluding specific extensions. If set to "true", all file types will be scanned except for those specified in Tcpd.scan.DirOptions.Extensions; if set to "false", only those specified in Tcpd.scan.DirOptions.Extensions will be scanned.

Tcpd.scan.DirOptions.Extensions=[MULTISTRING]

A list of file extensions that will be scanned/excluded from scanning, based on Tcpd.scan.DirOptions.ScanAllFiles.

Tcpd.scan.DirOptions.ScanFilesWithoutExtensions=true

Scan files with no extension.

Tcpd.scan.mail.strip.enable=false

Enable filtering of file attachments based on file extension, as specified in Tcpd.scan.mail.strip.list.

Tcpd.scan.mail.strip.list=[MULTISTRING]

Custom list of file extensions when Tcpd.scan.mail.strip.enable is set to "true".

Tcpd.scand.maxscanproc=2

The size of the scanning thread pool.

Tcpd.scan.DirOptions.MaxRecursionDepth=16384

The maximum level of recursion for file and directory processing.

Tcpd.scan.Options.MaxRecursionDepth=3

The maximum level of recursion for archives.

Tcpd.scan.Options.MaxNumberOfFiles=50000

The maximum number of files that can be extracted from archives.

Tcpd.scan.Options.MaxFileSize=0x10000000

The maximum size of files that can be extracted from archives. Larger files will be processed up to this limit.

Tcpd.scan.Options.DetectCookies=false

Enable detection of tracking cookies (small files managed by a web server, to which they send potentially exploitable information collected from the user's computer such as browsing history, etc.).

Tcpd.scan.Options.DetectPup=true

Detect spyware or PUP (Potentially Unwanted Programs), apart from viruses. Spyware represents a questionable malware category: even though it usually represents a security risk, some of these programs can be installed intentionally.

Tcpd.scan.Options.DetectPup2=false

Detect an extended package of spyware: programs that are perfectly ok and harmless when acquired from the manufacturer directly, but which can be misused for malicious purposes later, or programs that are always harmless but might be unwanted (various toolbars etc.). This is an additional security measure, however it can possibly block legal programs, and is therefore set to "false" by default.

Note: This detection feature is in addition to Tcpd.scan.Options.DetectPup, so if you want protection from the basic types of spyware, always keep Tcpd.scan. Options.DetectPup set to "true".

Default.scan.Options.PupExceptions=[MULTISTRING]

List of files excluded from PUP detection, if activated by Tcpd.scan.Options. DetectPup or Tcpd.scan.Options.DetectPup2.

Tcpd.scan.Options.ParanoidMode=false

Enable the "paranoid scan mode" that includes very detailed scanning, for example for old viruses or browser flaws that have long been fixed, etc. Please note that this mode makes heavy demands on system resources.

Tcpd.scan.Options.ArchiveLevel=256

Define the level of processing archives (resident, default):

0 – no archives, macros, cookies, real–time compression will be scanned, including MIME

32 – only macros, cookies, real–time compression will be scanned

256 – archives, macros, and cookies will be scanned

Tcpd.scan.Options.ReportArchiveBombs=true

Enable detection of malicious archive bombs. Upon decompression, these highly compressed archive files take up all available disk space and memory, and as a result, they bring down the application accessing them or the whole system.

Tcpd.scan.Options.ReportHiddenExtensions=false

Enable detection of files with double extension. One of the extensions can be hidden on some systems, which can make, for example, an executable file look like a harmless plain text file. For this reason, hidden extensions are suspicious.

Tcpd.scan.Options.ReportMacros=false

Enable detection of document files containing macros. A macro is a predefined sequence of steps designed to make certain tasks easier, and as such, can contain potentially dangerous instructions.

Tcpd.scan.Options.ReportPwdProtectedArchs=false

Enable detection of archive files protected by a password (i.e. not possible to scan).

Tcpd.scan.Options.ReportPwdProtectedDocs=false

Enable detection of document files protected by a password (i.e. not possible to scan).

Tcpd.scan.Options.ScanMediaFiles=true

Scan media files (video, audio etc.). Setting this option to "false" can reduce the scanning time because these files are often quite large and not very likely to be infected by a virus.

Tcpd.scan.Options.UseHeuristics=true

Use heuristic analysis for scanning. Heuristic analysis is basically used to evaluate unknown and suspicious objects by simulating the object's behavior in a safe virtual computer environment. The behavior is then analyzed, and it is decided if the object can be dangerous to the computer or not. The principle of heuristic analysis allows detection of malicious code not yet described in the virus database, which is why we strongly recommend using it at all times.

Tcpd.scan.mail.strip.alldoc=false

Enable detection of document files with extensions specified in Tcpd.scan.mail.strip.alldoclist.

Tcpd.scan.mail.strip.alldoclist=DO?, XL?, VBX, RTF, PP?, POT, MDA, MDB, XML, DOC?, DOT?, XLS?, XLT?, XLAM, PPT?, POT?, PPS?, SLD?, PPAM, THMX

List of files that are treated as documents when Tcpd.scan.mail.strip.alldoc is set to "true".

Tcpd.scan.mail.strip.allexe=false

Enable detection of executable files with extensions specified in Tcpd.scan.mail.strip.allexelist.

Tcpd.scan.mail.strip.allexelist=COM, DRV, EXE, OV?, PGM, SYS, BIN, CMD, DEV, 386, SMM, VXD, DLL, OCX, BOO, SCR, ESL, CLA, CLASS, BAT, VBS, VBE, WSH, HTA, CHM, INI, HTT, INF, JS, JSE, HLP, SHS, PRC, PDB, PIF, PHP, ASP, LNK, PL, CPL, WMF

List of files that are treated as executables when Tcpd.scan.mail.strip.allexe is set to "true".

5.2.2. Other

The following items specify further processing of scanned e-mails:

Default.tcpd.parsing.mime_certification_enabled=false

Disables/enables AVG certification text in the e-mail body. The text is added to the end of e-mail and contains brief information that the e-mail has been checked by AVG.

Default.tcpd.rules.virus.action=0

Define action for each detected message. Values:

0 – PASS; a message will be only certified (header, subject, body)

1 – DROP; a message will be deleted

2 – BOUNCE; a message will be delivered to the address defined by Default.tcpd.rules.virus.bounce_addr

Default.tcpd.rules.virus.bounce_addr=

Define an address for the BOUNCE action if set by Default.tcpd.rules.virus.action.

Default.tcpd.scan.header.enabled=true

Add "AVG Anti-virus header" to e-mails.

Default.tcpd.scan.subj_prefix=[VIRUS]

Add the specified prefix to the subject of infected e-mails.

6. Anti-Spam

The Anti-Spam component included in AVG 2012 Anti-Virus for Linux/FreeBSD offers extensive, detailed setting options for protection from spam and phishing.

> *Please note that Anti-Spam is only included in trial and full versions of AVG for Linux, that means, not if a free license has been used to register. Of course, it is possible to upgrade from free to full at any time and obtain the Anti-Spam module.*

The AVG daemon for anti-spam control is **avgspamd**. To enable the Anti-Spam function, use:

```
$ avgcfgctl -w Default.tcpd.spam.enabled=true
```

To enable launching Anti-Spam automatically by **avgd**, use:

```
$ avgcfgctl -w Default.setup.features.antispam=true
```

Anti-Spam configuration can be changed via the **avgcfgctl** command-line. To learn more, please refer to the **avgspamd** man pages. Basic configuration items are the following:

Default.tcpd.spam.header.enabled=true

Add "AVG Anti-spam header" to e-mails.

Default.tcpd.spam.subj_prefix=[SPAM]

Add the specified prefix to the subject of unwanted e-mails.

Default.tcpd.spam.phish_subj_prefix=[PHISHING]

Add the specified prefix to the subject of phishing e-mails (attempts to coax exploitable information from people by faking a trustworthy institution).

Default.tcpd.spam.spamscore_level=90

Set score for spam identification. A score represents how close the message content is to spam. Typically used values:

50 – Very aggressive configuration; non-spam e-mail messages are as likely to be filtered out as real spam. Not recommended for normal use.

70 – Aggressive configuration; e-mail messages that are possibly spam will be filtered out, and non-spam messages are likely to be caught as well.

90 – Most incoming e-mail messages will be delivered normally, and a significant amount of spam may be allowed through.

Default.tcpd.rules.phishing.action=0

Default.tcpd.rules.spam.action=0

Define action for each detected message. Values:

0 – PASS; a message will be only certified (header, subject, body)

1 – DROP; a message will be deleted

2 – BOUNCE; a message will be delivered to the address defined by Default.tcpd.rules.*.bounce_addr

Default.tcpd.rules.phishing.bounce_addr=

Default.tcpd.rules.spam.bounce_addr=

Define an address for the BOUNCE action if set by Default.tcpd.rules.*.action.

7. Troubleshooting

7.1. Log Files

AVG logs are primarily intended for our internal diagnostic purposes, however in case of trouble, you can try to consult the logs as well. They are located in **/opt/avg/av/log**; those containing ".pub." in their name are public. You can also use the **avgevtlog** tool for reading and managing AVG event log; see the chapter Event Log for more information.

7.2. AVGDiag

AVGDiag is an automated diagnostic utility designed to collect data about AVG crashes and to send these to AVG for analysis and solution. If you want to run **AVGDiag**, please make sure that you first create a file with a thorough description of the problem, which will help our Technical Support understand it. Then you can run the utility itself, attaching the file to it:

```
$ avgdiag --dsc=<file>
```

If you run **AVGDiag** on the request of our Technical Support, typically you will have been assigned an ID. To run **AVGDiag** and idenfity the report with the ID, please use:

```
$ avgdiag --id=<id>
```

AVGDiag can also be set to run automatically upon any AVG process crash. You can activate the function as follows:

1. Open the /opt/avg/av/cfg/dump.ini file.
2. Add "AVG_DIAG" to "actions" (by default, the line will read "actions = GDB_DUMP CRASH INFO AVG_DIAG").

7.3. Support

There are the following support options for users of AVG 2012 Anti-Virus for Linux/FreeBSD:

- **FAQ**: free.avg.com/faq -> group "Technical FAQ" -> group "AVG for Linux"
- **Forums (followed by our professional support staff)**: forums.avg.com/avg-forums -> topic "AVG for Linux"

For information on viruses and how to remove them, please refer to the chapter Handling Infection.

8. Uninstallation

To uninstall AVG 2012 Anti-Virus for Linux/FreeBSD from your computer, use the appropriate command according to the type of package (see the chapter Installation for more information on packages):

rpm file:

```
# rpm -e avg2012lms
```

deb file:

```
# dpkg -r avg2012lms
```

tar.gz/sh file:

```
# /opt/avg/av/bin/uninstall.sh
```