Doctor Web, Ltd.

Dr.Web® for Unix-systems (Linux, FreeBSD, OpenBSD and Solaris)

Administrator Manual

Version 4.33

There might be further improvements and changes in the software not described in this manual. The corrected and supplemented versions of this manual are available at http://www.drweb.com/

Russia, Moscow – Saint Petersburg

http://www.drweb.com/

Contents

1 Introduction

1.1. What is this manual about

The present manual describes the Doctor Web antivirus program (further named as Dr.Web) for Unix-based systems - Linux, FreeBSD, OpenBSD and SunOS Solaris.

The manual is meant for the employee concerned in the antivirus security (antivirus security system administrator), named administrator in this manual.

It is generally known, that counteraction to viruses in Unix-based operating systems (further named as Unix-systems) has two peculiarities. The first of them concerns the protection of the local system and user data from the destructive impact of viral programs. The second peculiarity deals with the diagnostics and neutralization of viruses when using Unix-systems as platforms for communication services, first of all, in mail servers, file servers of local networks, etc. While the viruses can be (and in most cases they are) designed not specifically for Unix-systems — through local networks and mail services they distribute ordinary Windows-viruses, including macro viruses for Word, Excel and other office applications.

The Dr.Web antivirus program for Unix-systems performs two functions. On the one hand, it is the Dr.Web scanner for detection and curing the viruses on the local computer. The Graphical interface module considerably facilitates setting of parameters and administration of the scanning process on the local computer.

On the other hand, the package includes the resident component (the daemon) Dr.Web Daemon (drwebd), which can be used almost in any data processing layouts as an external plug-in of the antivirus filter. For example, mail systems (such as Communigate Pro, Sendmail, Postfix, Exim, QMail, ZMailer and other) can be rather

easily and flexibly tuned for using the Dr.Web Daemon to check the e-mail messages transmitted by the mail server.

The aspect of using the Dr.Web program in Unix-systems, namely, the detection and neutralization of viruses in local computers will be covered in the present manual first. Here, we would like to emphasize the following points:

- general notions on the programs, the differences between the versions (chapter 1)
- program's installation from packages of different formats (chapter 2)
- using the virus scanner mainly on a local computer (chapter 3-4)

The second aspect will be reflected in chapters 5-9. At the end of this manual you will find the contact information of the developers and the technical support service.

The Dr.Web antivirus program is in constant development. The add-ons of the databases of the known viruses are released, as a rule, several times a day. The program itself gets upgraded too. The diagnostics techniques and counteraction to viruses, as well as integration with other applications of Unix-systems get constantly improved in the program. Besides, the list of applications compatible with Dr.Web is expanding. And it is not improbable, that some settings and functions of the current version will differ from those described in this manual. To get the present-day information on the program read the electronic documentation included into the delivery package, as it will be described in section 3.1.

1.2. Terms and abbreviations

The following terms are used in the manual (table 1).

Table 1. Legend

Legend	Interpretation
 *Important*	Important remark or instruction
 Attention	Warning on potentially dangerous or erroneous event
Scanner	A term used as definition or references to a definition
`/opt/drweb`	Names of files and directories, extracts from configuration files

The following abbreviations are used without further explanation in the Manual:

- OS — operating system.

1.3. Dr.Web® requirements to OS and Computer

Dr.Web for Linux is compatible with the Linux distributions based upon versions glibc 2.1, 2.2 and 2.3. Dr.Web for FreeBSD works with FreeBSD versions 3.x, 4.x, 5.x. Dr.Web for OpenBSD operates with OpenBSD versions 3.x. Dr.Web for Solaris works with Solaris versions 8, 9 and 10 (for Intel platforms only).

Regarding the hardware, the Dr.Web requirements are similar to those of the console (text) mode for the appropriate operating system — Linux, FreeBSD, OpenBSD and Solaris. To install the programs (the scanner and the daemon) approximately 2,5 Mb of the disk space is required.

When using the Graphical interface module the graphical mode support is required.

2 Installing Dr.Web®

The distribution filenames and names of intermediate directories created during the installation depend upon the antivirus version and the OS.

2.1. Installing Dr.Web® for Linux

The examples sited below, for the better understanding purpose, describe installation of the 4.33 antivirus for Linux based on glibc 2.2. In other cases, the names of files and directories will differ respectively (the section with the antivirus version number is changed and another version of glibc should be specified, or the relative indication of the Linux distribution should be specified instead of glibc).

The Dr.Web for Linux distribution is delivered in several formats:

- as tarball-archive (`drweb-4.33-glibc2.2.tar.gz`); this type of solution is not meant for a particular system distribution, but requires manual installation
- as sets of packages, designed for usage with definite Linux systems (rpm, deb and tgz) they are created for. The set consists of the following packages:
  - `drweb-base`: the package contains the search module, antivirus bases and main configuration file. The

package must be installed and it is required by all other packages

- `drweb-updater`: the package contains the updating utility of the search module and of the antivirus bases. The package must be installed and it is required by all other pack (except for `drweb-base`)
- `drweb-daemon`: the package contains executable files of the Dr.Web (R) daemon and documentation to it
- `drweb-scanner`: the package contains executable files of the Dr.Web (R) console scanner and the documentation to it
- `drweb-x`: the package contains executable files of the graphical interface for the Dr.Web (R) scanner and the documentation to it.

The installation from the packages requires support of this format by the system distribution.

The distribution designed as the set of packages is installed in a standard way with the help of the package managing utility under administrator rights. As the result, the packages are installed into the following directories: program files and documentation – to `/opt/drweb`, configuration files – to `/etc/drweb`, and antivirus bases – to `/var/drweb`.

Using RPM

Installation:

```
rpm -ih drweb-base-4.33-rh1.i586.rpm
rpm -ih drweb-updater-4.33-rh1.i586.rpm
rpm -ih drweb-daemon-4.33-rh1.i586.rpm
```

Updating:

```
rpm -Uh drweb-base-4.33-rh1.i586.rpm
rpm -Uh drweb-updater-4.33-rh1.i586.rpm
rpm -Uh drweb-daemon-4.33-rh1.i586.rpm
```

To install Dr.Web from the rpm-package you can also use graphical mode utilities, such as kpackage (in the KDE environment), Software Manager (included into the Linux Mandrake distribution), etc.

Using Debian-packages.

Installation/updating:

```
dpkg -i drweb-base_4.33_i386.deb
dpkg -i drweb-updater_4.33_i386.deb
dpkg -i drweb-daemon_4.33_i386.deb
```

Using Slackware-packages.

Installation:

```
installpkg drweb-base-4.33-i586-2.tgz
installpkg drweb-updater-4.33-i586-2.tgz
installpkg drweb-daemon-4.33-i586-2.tgz
```

Updating:

```
upgradepkg drweb-base-4.33-i586-2.tgz
upgradepkg drweb-updater-4.33-i586-2.tgz
upgradepkg drweb-daemon-4.33-i586-2.tgz
```

Or `pkgtool` utility can be used.

If installing from the tarball-archive, you should unpack the archive. The `drweb-4.33-glibc2.2` directory, which will contain the directory tree, will be created. Copy the tree to the root directory (you can also specify your own directory structure, but in this case you should edit the program configuration file, see p. 3.4).

Example:

```
> tar xzf drweb-4.33-glibc2.2.tar.gz
> cp -a drweb-4.33-glibc2.2/* /
```

Then, create the `drweb` account and set up correct permissions for the directories created:

```
> chown -R drweb:drweb /etc/drweb
> chown -R drweb:drweb /opt/drweb
> chown -R drweb:drweb /var/drweb
```

2.2. Installing Dr.Web® for FreeBSD

The examples sited below, for the better understanding purpose, describe installation of the 4.33 antivirus for FreeBSD version 4.x. In other cases, the names of files and directories will differ respectively.

The Dr.Web distribution for this OS is delivered in two formats:

- as bsd package `drweb-4.33-freebsd4.tgz`, designed for FreeBSD packages managing system
- as tarball-archive (`drweb-4.33-freebsd4.tar.gz`), for manual installation

Follow the instruction to install the package:

```
> pkg_add drweb-4.33-freebsd4.tgz
```

If you want to use the tarball-archive for installation, unpack the archive. The `drweb-4.33-freebsd4` directory containing the directory tree will be created. Copy the tree to the root directory (you can also specify your own directory structure, but in this case you should edit the program configuration file, see p. 1.3.4).

Example:

```
> tar xzf drweb-4.33-freebsd4.tar.gz
> cp -pR drweb-4.33-freebsd4/* /
```

Then, create the `drweb` account and set up correct permissions for the directories created:

```
> chown -R drweb:drweb /usr/local/etc/drweb
> chown -R drweb:drweb /usr/local/drweb
> chown -R drweb:drweb /var/drweb
```

2.3. Installing Dr.Web® for OpenBSD

The examples sited below, for the better understanding purpose, describe installation of the antivirus for OpenBSD version 4.33. In other cases, the names of files and directories will differ respectively.

The Dr.Web distribution for this OS is delivered as a tarball-archive.

For OpenBSD versions 3.1, 3.2 and 3.3 the distribution name is `drweb-4.33-openbsd3.tar.gz` (for OpenBSD 3.3 use the static program version).

For OpenBSD 3.4, 3.5 the distribution name is `drweb-4.33-openbsd34.tar.gz`

At present, installation of Dr.Web under OpenBSD is made manually only. Below go the examples for OpenBSD 3.1.

Unpack the archive. The `drweb-4.33-openbsd3` directory containing the directory tree will be created. Copy the tree to the root directory (you can also specify your own directory structure, but in this case you should edit the program configuration file, see p. 3.4).

Example:

```
> tar xzf drweb-4.33-openbsd3.tar.gz
> cp -pR drweb-4.33-openbsd3/* /
```

Then, create the `drweb` account and set up correct permissions for the directories created:

```
> chown -R drweb:drweb /usr/local/etc/drweb
> chown -R drweb:drweb /usr/local/drweb
> chown -R drweb:drweb /var/drweb
```

2.4. Installing Dr.Web® for Solaris

The examples sited below, for the better understanding purpose, describe installation of the 4.33 antivirus for Solaris version 9 (the same distribution is used for installation of Solaris 10). In other cases, the names of files and directories will differ respectively.

The Dr.Web distribution for this OS is delivered as tarball-archive. At present, installation of Dr.Web under Solaris is made manually only.

Unpack the archive. The `drweb-4.33-solaris9` directory containing the directory tree will be created. Copy the tree to the root directory (you can also specify your own directory structure, but in this case you should edit the program configuration file, see p. 3.4).

Example:

```
> gzip -d drweb-4.33-solaris9.tar.gz
> tar xf drweb-4.33-solaris9.tar
> cp -pR drweb-4.33-solaris9/* /
> gzip -d drweb-4.33-solaris9.tar.gz
> tar x -C / -f drweb-4.33-solaris9.tar
```

Then, create a `drweb` account and set up correct permissions for the directories created:

```
> chown -R drweb:drweb /etc/drweb
> chown -R drweb:drweb /opt/drweb
> chown -R drweb:drweb /var/drweb
```

If less than two weeks till the key expires left, the scanner will generate a message on it. The daemon in this situation can notify a user via e-mail. The messages are sent for every installed key file at every launch, restart or reload of the daemon, if less than two weeks

till the key expires left. To enable this option, the `MailCommand` parameter in the `[Daemon]` section of the `drweb32.ini` file should be set up (read).

2.5. Package registration. License key file

User's rights to use the antivirus are regulated by the special file called the *key file*. The key file contains, in particular, the following information:

- list of antivirus components licensed to the user
- the licensed version of the antivirus
- the license term of the antivirus
- virus definitions automatic updates period (also called the "subscription period" which may not be equal to the license period)
- other restrictions (i.e. the number of protected PCs, etc.)

The key file has the `key` extension and must be located by default in the installation directory.

The key file has a write-protected format and therefore must not be edited. Editing the file makes it invalid. Consequently, it is not recommended to open your key file with a text editor, which may occasionally corrupt it.

Users who have purchased the antivirus from the Doctor Web's certified partners obtain a *license key file.* The parameters of the key file are specified according to the license the user has paid for. The license key file contains the name of the user (or a company name), and the name of the selling company.

For evaluation purposes users may also obtain *demo key files*. Demo key files allow to enjoy full functionality of the program and the virus definitions updates, but have a limited term of use and no users' support is provided.

The key file may be supplied with the `key` extension, or as a zip archive containing the key file.

The key file may be received in one of the following ways:

- supplied or sent as a zip archive containing a file with the `key` extension (usually after the registration on the web site, explained below). Extract the key file using the appropriate archiving tool and place it to the directory where the program's executable files reside (by default it is `/opt/drweb` for Linux and Solaris, `/usr/local/drweb` for FreeBSD and OpenBSD)
- included into the distribution package
- supplied on a media as a file with the `.key` extension. The user should copy it manually to the directory specified above

The license key file is sent to users via email, as a rule, after the registration on the web site (the location of the web site is specified in the registration card accompanying the product). Visit the indicated site, fill in the web form with the customer data and put in the registration serial number (printed in the registration card). The key file will be sent to the specified address.

The key file can differ for the daemon and the scanner. Therefore, If necessary, the settings of the `Key` parameter in the `[Daemon]` section of the `drweb32.ini` file (read p. 5.3) should be changed. For example,

```
Key = DEFAULT_BIN_PATH/drweb32.key
```

If all key files specified in the `Key` parameters of the `[Daemon]` section are failed to read (wrong path, permission denied) or expired or blocked or invalid then the daemon terminates his work.

If less than two weeks till the key expires left, the scanner will generate a message on it. The daemon in this situation can notify a user via e-mail. The messages are sent for every installed key file at every launch, restart or reload of the daemon, if less than two weeks till the key expires left. To enable this option, the `MailCommand` parameter in the `[Daemon]` section of the `drweb32.ini` file should be set up (read p. 5.3).

2.6. Updating programs and virus bases

As any other antivirus package the Dr.Web antivirus requires regular updating of the virus bases of the known viruses. This task is implemented in the following way. The virus bases contain several `*.vdb` files, representing separate parts of it. When new viruses appear, small (of one or several Kb) files, which contain the base fragments describing these viruses, are released.

The add-ons are files (uniform for all supported platforms) presented as `drwtoday.vdb` (daily "hot" add-ons) and `drwXXXXX.vdb` (regular updates, usually issued weekly). To add the add-on to the main virus base place this file to the Dr.Web program directory (by default – to `/var/drweb/bases`) or to any other directory specified in the configuration file (see section 5.6).

Periodically (as brand new viruses and antivirus techniques appear), new versions of the antivirus package containing the updated algorithms, implemented in the virus kernel, are released. Simultaneously, all released add-ons are brought together and the new version is completed with the main updated base, bearing the descriptions of all viruses known for the moment of their appearance. Usually, when upgrading the package the succession of the bases format is preserved,

i.e. new bases can be linked up to the old kernel. This, however, does not secure detection or curing of new viruses, as it will require upgraded algorithms of the antivirus kernel.

Regular updating of the add-ons of package's virus bases predetermines the following structure:

- `drwebase.vdb` — main base, received with the new version of the package
- `drwXXXXX.vdb` — regular (weekly) virus bases updates
- `drwtoday.vdb` — "hot" add-ons issued daily or several times a day

The peculiarity of the installation of "hot" add-ons lies in the fact, that during the interval between the regular (numbered) add-ons release the `drwtoday.vdb` file is replenished with the new entries, i.e. it should be installed instead of the previous file. When the next regular add-on is released, all entries from this file are copied to the regular add-on, the file itself gets cleaned (the `drwtoday.vdb` file becomes clear of any entry of the base) is released.

Consequently, when updating the bases manually, you should install all missing regular updates and after that write the "hot" add-on file (instead of the previous file).

For automatic receipt and installation of the virus bases' add-ons a special script should be used from the program's executable files directory.

Script for Linux and Solaris:

`/opt/drweb/update.pl`

Script for FreeBSD and OpenBSD:

`/usr/local/drweb/update.pl`

The updating is made in the following way:

- the script reads the ini-file (it may be explicitly specified as the first and the only argument in the command line, e.g.,

for Linux and Solaris)
`update.pl /etc/drweb/drweb32.ini`
for FreeBSD and OpenBSD:
`update.pl /usr/local/etc/drweb/drweb32.ini`

- The following parameters are read from the ini-file:
  `EnginePath` (serves both to determine the daemon's version and to specify the directory the received file drweb32.dll is placed to)
  `VirusBase` (the first mask of this parameter specifies where the received bases will be placed to)
  `UpdatePath` (the location of all other received files),
  `PidFile` (the drwebd process identifier used for the daemon's reload is read from this file)
- the files are received and placed to the directories as it is described above (all files for the current Dr.Web version are accepted; if the sections of outdated versions are found in the add-on list, only `*.vdb` will be received)

Before using the script you should edit the lines containing the addresses for downloading the add-ons (there may be several of them) and specify what section of the configuration file the data should be used from (scanner of daemon).

3 Using Dr.Web® Scanner

This section describes the location of the Dr.Web program files, program's startup procedure, command line options and the configuration file structure. The use of the Graphical interface module is described in p. 4

3.1. Location of package files

The Doctor Web is installed by default to the directories further named as `%bin_dir`, `%etc_dir` and `%var_dir`. Depending on the OS version they indicate the following directories:

- for Linux and Solaris
  `%bin_dir` – it is `/opt/drweb`
  `%etc_dir` – it is `/etc/drweb`
  `%var_dir` – it is `/var/drweb`
- for FreeBSD and OpenBSD
  `%bin_dir` – it is `/usr/local/drweb`
  `%etc_dir` – it is `/usr/local/etc/drweb`
  `%var_dir` – it is `/var/drweb`

The OS-independent subdirectories structure is created in these directories.

The `%bin_dir` directory houses the executable program modules – the scanner (drweb) and the daemon (drwebd).

In its `lib` subdirectory the antivirus kernel as a loaded library is located (`drweb32.dll`).

The database of known viruses - `%var_dir/bases/drwebase.vdb`

Configuration file - `%etc_dir/drweb32.ini`

Language resource files (e.g., for operation with Russian interface) for scanner and daemon -
`%bin_dir/lib/ru_scanner.dwl`
`%bin_dir/lib/ru_daemon.dwl`

The documentation is located in the `%bin_dir/doc` subdirectory.

The program manuals are represented as usual text files in two variants — in English and Russian (KOI8-R encoded).

In the `%bin_dir` directory a perl-script for updating the program (`update.pl`) is located.

The subdirectory of the `%var_dir/infected` is meant for moving there files if the reaction of the program is set to detect the infected or suspicious files.

3.2. *Starting* the *scanner*

The Doctor Web scanner is a program with the textual interface functioning in the console mode (or in the terminal emulator window in X Window). It is launched by the sequence of instructions

```
> %bin_dir/drweb
```

or, if the `%bin_dir` directory is put into the PATH command shell environment variable — from an arbitrary directory. The last variant is not recommended for safety reasons, as well as making a reference to the drweb executable file in any directory like `/bin`, `/usr/bin` etc.

The scanner can be started both under the name of an administrator and under a user name. The check for viruses will be made only in the directories a user has a read access to, and the infected files will be cured in the directories such a user has a write access to (as a rule, this is a user's home directory, `$HOME`). Besides, there exist other restrictions of the program's functionality started in the user mode (e.g., moving and renaming of infected files).

The scanner being started, the headpiece with the name of the program and the platform (Linux, FreeBSD, OpenBSD or Solaris), the version number, the date of its release and the contact information are displayed. It is followed by a user's registration information and report on the antivirus database load, including add-ons (if any installed):

```
Dr.Web for Linux, version 4.32.1 (2004-08-30)
Copyright (c) Igor Daniloff, 1992-2004
Support service: http://support.drweb.com
To purshase: http://buy.drweb.com
Loading /var/drweb/bases/drwtoday.vdb - Ok,
                                    virus records: 5
Loading /var/drweb/bases/drw43203.vdb - Ok,
                                  virus records: 409
Loading /var/drweb/bases/drw43202.vdb - Ok,
                                  virus records: 543
Loading /var/drweb/bases/drwebase.vdb - Ok,
                                virus records: 51982
Loading /var/drweb/bases/drw43201.vdb - Ok,
                                  virus records: 364
Total virus records: 53303
Key file: /opt/drweb/drweb32.key
Key file number: 0000000007
```

After that the command shell invitation is returned. All other actions on detection and neutralization require specification of the command line options.

3.3. Command line options

The program gets started as follows (suppose that the `%bin_dir` directory is the current directory):

```
./drweb -path=<path> [options]
```

where <path> means the path to the checked directory or the tested files mask. The scanner started with empty options, with the path set as an argument only, will check the specified directory using the default set of options (we will further return to it in this section). The

next example shows how to enable checking of a user`s home directory:

```
./drweb -path=~
```

When the check is finished, the program outputs the report as follows:

```
Scan report for "/opt/drweb/tmp":
Scanned:         34/32
Cured:           0
Infected:        5/5
Deleted:         0
Modifications:   0/0
Renamed:         0
Suspicious:      0/0
Moved:           0
Scan time:       00:00:02
Scan speed:      5233 Kb/s
```

The numbers divided by the "/" character mean: the first — the total number of files, the second — the number of files in archives. Besides, if a file infected with the known virus, or suspicious files are detected, the following lines are displayed before the summary report:

```
/path/file infected [by a virus] VIRUS_NAME
```

Be aware that the Dr.Web distribution contains a specially designed `readme.eicar.rus` text file. With the help of a text editor one can easily make the eicar.com program (read this file for instructions), which acts as a virus causing the output of the following report:

```
%bin_dir/doc/eicar.com
         infected by Eicar Test File (Not a Virus!)
```

Actually, this file is not a virus, but it is usually used to test the reaction of the antivirus to the presence of a virus. All modern antivirus programs include information on it to its virus bases.

As for any other Unix-program, the Dr.Web scanner supports multiple command line options. They are separated from the specified path by

a blank and prefixed by the — (hyphen) symbol. To get the complete list of options run the drweb program with the following options:
`-?,-h, -help` or `--help`

The main program options can be classified in the following way:

- scan area options
- diagnostics options
- actions options
- interface options

The scan area options determine where the check for viruses should be performed. They include:

- @[+]<file> — check of objects listed in the given file, the + (plus) symbol instructs not to delete a file from the list of objects after the check is completed; the list file may contain paths to directories scanned periodically, or a list of files to be checked regular only
- **`sd`** — recursive search and testing of files in subdirectories starting with the current one
- **`fl`** — follow the references, both for files and directories; the references resulting in "circularity" are ignored

The diagnostics options determining what types of objects should be checked for viruses include:

- **`al`** — the diagnostics of all files on the specified drive or in the directory specified as an argument
- **`ar`**[**`d`**|**`m`**|**`r`**][`n`] — check of files in archives (ARJ, CAB, GZIP, RAR, TAR, ZIP);
  `d` – delete, `m` – move, `r` – rename archives containing infected objects;
  `n` – archiver name output disabled;
  by archives we mean not the archives themselves (e.g.,

such as `*.tar`), but also their compressed forms (in particular, compressed tar-archives such as `*.tar.gz and *.tgz`)

- **cn**[d|m|r][n] – check of files in containers (HTML, RTF, PowerPoint,..);
  d – delete, m – move, r – rename containers with infected objects; n – container type output disabled
- **ml** [d|m|r][n]— mail program files check
- **up**[n] — check executable files packed with LZEXE, DIET, PKLITE, EXEPACK;
  additional option -n disables displaying compression utilities names (n – packing utilities name output disabled)
- **ex** — diagnostics of files the name of which correspond to specified masks (see the `FilesTypes` option of the configuration file, section 3.4)
- **fm** — diagnostics of files with the program modules` inner structure
- **ha** —heuristic analysis of files, search of unknown viruses.

The actions options determine what actions should be performed if infected (or suspicious) files are detected. They include:

- **cu**[d|m|r] — curing of infected files; additional options instruct: d — delete, m — move, r — rename incurable files
- **ic**[d|m|r] — determines actions for incurable files: d — delete, m — move, r — rename incurable files
- **sp**[d|m|r] — determines actions for suspicious files: d — delete, m — move, r — rename suspicious files

- **adw**[d|m|r|] — specifies actions for files containing adwares: d — delete, m — move to, r — rename such files, '-' — ignore
- **dls**[d|m|r|] — specifies actions for files containing dialer programs: d — delete, m — move, r — rename such files, '-' — ignore
- **jok**[d|m|r|] — specifies actions for files containing joke-programs: d – delete, m — move to, r — rename such files, '-' — ignore
- **rsk**[d|m|r|] — specifies actions for files containing potentially dangerous programs: d — delete, m — move to, r — rename such files '-' — ignore
- **hck**[d|m|r|] — specifies actions for files with hacktools: d — delete, m — move to, r — rename such files, '-' — ignore

The interface options determine conditions of the program's report output and include:

- **ot** — output information to stdout, i.e. a standard output
- **oq** — disable information output
- **ok** — display 'Ok' for non-infected files
- **log**=<file> — logging to the specified file
- **ini**=<file> — using alternative ini-file
- **lng**=<file> — using alternative language file; if English interface is chosen during the installation specify ru_scanner.dwl as such file to display messages in Russian

Some options disable the correspondent action if postfixed by the – (minus) symbol. They include:

```
-ar -cu -ha -ic -fl -ml -ok –sd –sp -up
```

Example, if starting the scanner as

```
drweb -path=<path> -ha-
```

the heuristic check of files for viruses, usually enabled by default, will be disabled.

By default (with no options specified), if the program was not reconfigured (this will be described in the next section), the scanner will be started with the following options:

```
-ar -fm -ha -fl- -ml -sd -up
```

This set of the default scanner options (including the scanning of archives, packed files, mail programs, recursive search, heuristic analysis, etc.) is rather expedient for the diagnostics and can be applied in most standard cases. If the definite situation does not require applying some default option it can be disabled by placing the – (minus) symbols after it, as it was described above with the `-ha` (heuristic analysis) option.

Disabling the archived and packed files check significantly decreases the antivirus protection level, as the archives (often self-unpacking) distribute file viruses enclosed in mail attachments. The documents of certain applications, potentially susceptible to infection with macro viruses (Word, Excel), are also usually sent via e-mail being archived and packed.

When starting the scanner with the default options no curing of the infected files is provided. Nor any actions for incurable and suspicious files are taken. Such actions require specifying additional command line options — actions options.

The set of the actions options may vary in each case. Still, we recommend the following:

- `cu` — curing of infected files and system areas, without deletion, moving or renaming the infected files
- `icd` — deletion of incurable files
- `spm` — moving or `-spr` — renaming of suspicious files

When starting the scanner with the cure option, the program will make an attempt to restore the state of the infected object before the infection with a virus. This action is possible only if the known virus is detected, and the necessary instructions for curing are available in the base; still, even in these cases the attempt of curing may fail, for example, if the infected file has already been seriously damaged by the virus.

If the infected files were found inside the archives, they will not be cured, deleted, moved or renamed. To have the viruses deleted from such objects the archives should be manually unpacked by the correspondent software tools, advisably into a separate directory, which will be specified as an argument at the next scanner start.

If started with the delete option, the program will delete the infected file on the disk. This option is advisable for incurable (irreversibly damaged by a virus) files.

The rename option instructs to replace the file extension name with some specified extension (by default it is *.#??, i.e. the first extension symbol is replaced with #). Enable this option for files of other operating systems (e.g., DOS/Windows) detected heuristically as suspicious. Renaming will exclude the accidental launch of executable modules in these systems, loading of Word or Excel documents till the next check and avert the infection by a possible virus and its further proliferation.

The move option enabled will move the infected (or suspicious for a virus) file to the specified directory (by default, `%var_dir/infected`). By now, it is of theoretical value only: it is

useless for files of other operating systems, as they cannot damage a host system. Moving suspicious files in the Unix-system itself may produce operational errors and even result in its total failure.

The advisable mode of the scanner start for everyday usage will be as follows:

```
drweb -path=<path> -cu -icd -spm -ar -fm -ha -fl-
                                              -ml -sd -up
```

Such a command can be saved as text file, which, with the help of the following command `chmod a+x [name]` can be interpreted as a command shell script, or as a script for different situations. However, the default options sets can also be changed when setting the scanner. For details read the next section.

3.4. Configuring the scanner

You can certainly use the scanner with its default settings, but it is much more convenient to adjust it according to you requirements and conditions of operation. The scanner settings are stored in the program's configuration file, by default it is `drweb32.ini` and located in the `%etc_dir` directory. To use another configuration file specify the full path to it by the command line key at the scanner start, e.g.:

```
$ /usr/local/sbin/drweb
                        -ini=/usr/local/etc/drweb.ini
```

The configuration file is a text file (can be edited by any text editor) that has the following arrangement:

```
--- Beginning of file ---
[Name of section 1]
Parameter1 = value1, ..., valueK
.....
ParameterM = value1, ..., valueK
......
[Name of section X]
Parameter1 = value1, ..., valueK
```

```
.....
ParameterY = value1, ..., значениеK
--- end of file ---
```

If the line begins with ";" or "#", it is considered a line of comments; such lines are skipped by the scanner when reading parameters from the ini-file.

If some parameter of the scanner is not specified, this does not mean the parameter has no value, — the default value will be used (a common mistake when setting the scanner up). Only some parameters are optional or have no default values, which will be specially mentioned below. The values can be included in brackets (they must be included in brackets if contain blanks). Some parameters can have several values, the "," (comma) plays the role of the separator; in the parameter description it will be explicitly stated. If the values are included in {} in the parameter value field in this description, it means the parameter may take only one of these values.

Scanner settings section name — [Scanner].

The parameters will be described as follows:

Parametername = Parameterpseudovalue

Parameter description

{May have several values}

Default value: {value | unspecified}

Parameters description in their sequence order in the configuration file created when setting up the program:

EnginePath = {path to a file, usual extension — dll}

Location of the drweb32.dll module (search module). This parameter is also used by the updating module for updating the search module.

Default value: %bin_dir/lib/drweb32.dll

VirusBase = {list of paths (masks) to files, usual extension – vdb}

Masks for loaded virus bases. This parameter is also used by the updating module for updating the antivirus bases.

Listing of several masks allowable.

Default value: %var_dir/bases

UpdatePath = {directory}

This parameter is used by the updating module (update.pl) and if used, it should be obligatory specified.

Default value: unspecified

TempPath = {directory}

This directory is used by the antivirus module (search engine) to create temporary files. At normal operation this directory is almost not used, it is used for unpacking certain archives, or when a system lacks memory resources.

Default value: empty (used /tmp).

LngFileName = {path to the language resource file, usual extension – dwl}

Location of the localization file.

Default value: empty (in this case all messages are displayed in English).

Key = {path to a file, usual extension — key}

Location of a key file (license or demo).

Default value: %bin_dir/drweb32.key

OutputMode = {Terminal | Quiet}

Information output mode at start: Terminal outputs to a console, Quiet disables output.

Default value: `Terminal`.

`HeuristicAnalysis` = {`Yes` | `No`}

Enabling/disabling the heuristic detection of unknown viruses. Enabling the heuristic analysis makes possible detection of unknown viruses on the basis of a priori assumptions of the virus code structure. The approximate, probabilistic nature of this type of the virus detection, we'd rather speak about *suspicious*, than infected objects, distinguishes this type of the virus search. With this option disabled only known viruses will be detected by the virus base. A whole class of programs using similar with viruses code can cause an erroneous triggering of the heuristic analyser. Besides, this mode may somewhat increase the time of scanning for viruses (still, this increase is insignificant). These considerations may serves as reasons for disabling the heuristic analysis. At the same time, the analysis improves the reliability of the antivirus protection. We would advise to send all files detected by the heuristic analyser to developers through http://www.drweb.com(preferably) or via e-mail newvirus@drweb.com. Follow this procedure to upload the file. Archive the file in a password protected archive, include the password in the message body and attach the scanner report.

Default value: `No`.

`ScanPriority` = {value}

The daemon's scanning processes priority. The range of this parameter value should be within – 20 (highest priority) to 20 (lowest priority).

Default value: `0`.

`FilesTypes` = {list of extensions}

The types of files to be checked "by type", i.e. when the `ScanFiles` parameter (explained below) has the `ByType` value. The symbols "*" and "?" are allowable.

Several lines are allowable for such parameter name; in this case the specified lists are summed up.

Default value: EXE, COM, SYS, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, VXD, 386, DLL, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, AR?, ZIP, R??, PP?, OBJ, LIB, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SHS, SHB, PIF, SO, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, BMP, MPP, OCX, DVB, CPY, MSG, EML.

ScanFiles = {All | ByType | ByFormat}

Additional restriction for files to be checked. With the ByType value set the so-called file extensions (i.e. the last suffix after "." in the filename, if "." is absent it means the file has no extension), the values of which are specified or set either by default, or in the FilesTypes parameter (parameters), are taken into account. When setting the ByFormat value for files included by the scanner startup command, these files are checked whether or not they can be carriers of viruses, i.e., executable files (the name of the file and the extension are disregarded), and after that only the antivirus scanning will be performed for files presumably considered executable.

The All mode is always enabled in mail files. The ByType and ByFormat values can be interpreted only in the "local" scanning modes.

Default value: ByFormat.

ScanSubDirectories = {Yes | No}

Whether or not the scanner should check the content of all subdirectories in all the directories specified in the scanner command.

Default value: Yes.

CheckPackedFiles = {Yes | No}

Whether or not the unpacking of executable files packed with DIET, PKLITE, and etc. utilities should be performed.

Default value: Yes.

CheckArchives = {Yes | No}

Whether or not ZIP (WinZip, InfoZIP...), RAR, ARJ, TAR, GZIP and CAB archives should be unpacked.

Default value: Yes.

CheckEMailFiles = {Yes | No}

Whether or not the files of (e-mail) formats should be checked.

Default value: Yes.

ExcludePaths = {list of paths (masks) to be excluded from search}

The masks for files that should not be checked.

Default value: left empty.

FollowLinks = {Yes | No}

Whether or not the symbolic link should be followed at scanning.

Default value: No.

RenameFilesTo = {mask}

Masks for renaming files, if for the definite condition (an infected or suspicious file) the Rename action is set.

Default value: "#??" means that the first character of the file "extension" will be replaced with "#", and two subsequent characters will be preserved. If a file had no "extension", it will consist of just one "#" symbol.

EnableDeleteArchiveAction = {Yes | No}

This option enables/disables the scanner's Delete (deletion) action for various complex objects (archives, mailboxes, html-pages), if they contain an infected object.

With this option enabled the whole complex object will be deleted, i.e. the whole archive, or the whole mailbox, not only the infected message or a file in the archive. Apply this option carefully!

Default value: No.

InfectedFiles = {reaction}

Sets the program reaction for detection of a file infected with a known virus.

Allowable parameter values inlude:

- Report — report information to log only
- Cure — try to cure an object
- Delete — delete an infected file
- Move — move a file to the directory specified by the MoveFilesTo parameter
- Rename — rename a file using the masks specified by the RenameFilesTo parameter.

Default value - Report.

Some other similar parameters setting the program's reaction on detection of one or another object are also supported:

SuspiciousFiles — a file probably infected by an unknown virus;

IncurableFiles — a file is infected and incurable (makes sense if `InfectedFiles` = `Cure`); a message or a mailbox contains an infected object;

ActionInfectedArchive — an archive (ZIP, TAR, RAR and other) contains an infected file;

ActionInfectedContainer — a container (OLE, HTML, PowerPoint and other) contains an infected file.

For all these parameters the same values as for the `InfectedFiles` parameter, except for `Cure` can be specified.

Default value for each parameter: `Report`.

The delete, move and rename actions set for the infected objects detected in archives, containers and mailboxes are applied to the whole archive, container or a mailbox.

There also available some more similar action parameters:

ActionAdware – a file contains a program for displaying advertisements (so-called AdWare);

ActionDialers – a file contains an automatic dialer program, usually used by porno-sites;

ActionJokes – a file contains a joke program, which can frighten or irritate user;

ActionRiskware – a file contains a dangerous program, which can be used by malefactors;

ActionHacktools – a file contains a hacktool for breaking into computers and other computer techniques.

For al these parameters the same values as for the `InfectedFiles` parameter, except for `Cure`, can be specified.

In addition to these actions, `Ignore` is also avaliable, in console version this action is analogues to `Report` except an exit code does not contains an information about such objects, it is also used by the scanner graphical shell.

Default value for each parameter: `Report`

`LogFileName` = {filename}

Log file name. You may specify `syslog` for the log filename and the logging will be done by means of the syslogd system service. When using syslog the `SyslogFacility` and `SyslogPriority` parameters (explained below) should be taken into account. As syslog utilizes several files for logging different events and different degrees of their importance, these two parameters and the content of the syslog configuration file (usually `/etc/syslogd.conf`) determine the location where the report will be logged to.

Default value: `syslog`.

`SyslogFacility` = {`Daemon` | `Local0` .. `Local7` | `Kern` | `User` | `Mail`}

Sets the log type when using syslogd system service.

Default value: `User`.

`SyslogPriority` = {`Alert` | `Warning` | `Notice` | `Info` | `Error`}

Sets the log priority when using syslogd system service.

Default value: `Info`.

`LimitLog` = {`Yes` | `No`}

The parameter defines whether or not the log file size should be limited.

Default value: `No`.

MaxLogSize = {Value in Kb}

This parameter sets the maximum log file size. Can be used with `LimitLog` = `Yes` only.

Default value: `512`.

LogScanned = {`Yes` | `No`}

Whether or not the information on all scanned objects, regardless the viruses were detected or not, should be logged.

Default value: `No`.

LogInfected = {`Yes` | `No`}

Whether or not the information on infected objectes should be logged.

Default value: `Yes`

LogPacked = {`Yes` | `No`}

Whether or not additional information on files packed with DIET, PKLITE, etc. utilities should be logged.

Default value: `Yes`.

LogArchived = {`Yes` | `No`}

Whether or not additional information on the archiving tools should be logged.

Default value: `Yes`.

LogTime = {`Yes` | `No`}

Whether or not the time for each record should be logged. The parameter is not used if `LogFileName` is set to `syslog`.

Default value: `Yes`.

LogStatistics = {Yes | No}

Whether or not the total statistics of scanning should be logged.

Default value: Yes.

RecodeNonprintable = {Yes | No}

The output mode for symbols invisible for the given terminal.

Default value: Yes.

RecodeMode = {Replace | QuotedPrintable}

The RecodeNonprintable value set to Yes specifies decoding for invisible symbols. If the value is set to Replace, all such symbols are replaced with the RecodeChar parameter value (see below).

Default value: Replace.

RecodeChar = {"?" | "_" | ...}

The RecodeMode value set to Replace specifies the symbol the invisible symbols will be replaced with.

Default value: "_".

The following parameters can be used to reduce the archives' scan time (some objects in the archive will not be checked).

MaxCompressionRatio = {value}

Maximum compression ratio, i.e. the ratio of the unpacked file size to the packed file size (inside an archive). If the ratio exceeds this value the file will not be extracted, and therefore will not be checked.

Default value: empty (all files are checked).

CompressionCheckThreshold = {value in Kb}

The minimum size of a file inside an archive beginning from which the compression ratio check will be performed (if this is specified by the MaxCompressionRatio parameter).

Default value: empty, the ratio check is not performed.

`MaxFileSizeToExtract` = {value in Kb}

The maximum size of a file extracted from an archive. If the file size inside the archive exceeds this value it will be skipped.

Default value: empty (the files of any size get extracted).

`MaxArchiveLevel` = {value}

The maximum archive nesting level (archive in archive, then in archive, etc.)

Default value: empty (archives of any nesting level get extracted).

4 Scanner graphical interface module

Depending on the OS and its settings, the interface elements of the described program and user actions may vary. The present description is made on the bases of the program operated by Linux Fedora Core 3, GNOME graphical environment.

Many actions and settings of the program may be specified in several ways (through the main menu, contextual menu, hot keys and other). As a rule, only one option is described, for short.

4.1. Launching a program

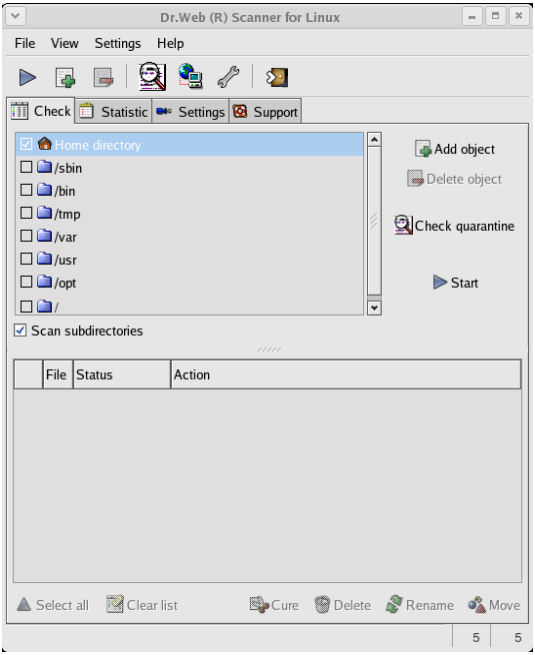
Installation of the Graphical interface module, read p. 2. Location of directories and files of the component, read. p. 3.1.

Before to launch the program, you should run X Window Server.

The launching mode of the Graphical interface module depends on the OS and the way the module was installed. If RPM packet manager was used, execute the `xdrweb` command in the console.

Regardless the installation procedure, you may also execute the `drweb-gui` file from the scanner installation directory.

The main program window will open (pic 1).



Picture 1. Main Graphical interface module window

Immediately after the launch, the program checks the settings of paths to the key file and the scanner. If a valid key file and the scanner executable file are not found in the specified location, a warning message is displayed.

Then, the Graphical interface module runs the scanner to obtain information on the antivirus protection state. If the scanner is not launched, a message is displayed.

If necessary, edit the corresponding program's settings (read below p. 2.2) and check the correctness of the scanner's installation.

If the program is launched from the command line, you may use the following parameters:

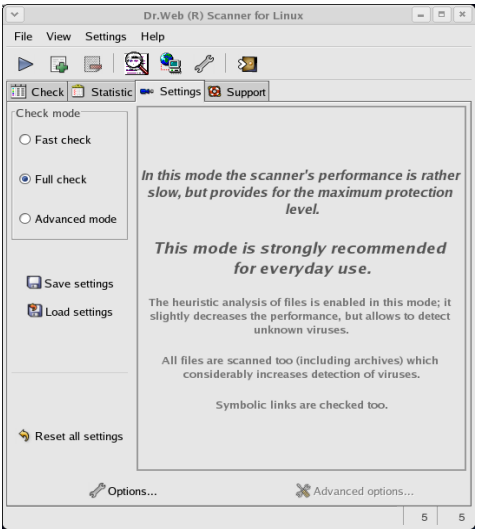
- **`-default-settings`** – not to load the configuration file

- **-eng-lng-save**=<path> - save the language resources files by default in the specified location
- **-ini-file**=<path> - path to the configuration file
- **-no-lng-load** – not to load the language resources files (use English, the program itself contains the resource)
- **-d** or **--debug** – run in the debug mode
- **-v** or **--version** – show program version
- **-h** or **--help** – display short prompt on action

4.2. Setting the scanner

4.2.1 Editing and saving the settings. Selecting the scanning mode

To start editing the settings, select the `Settings` pane (pic. 2).



Picture 2. Settings pane

To load the settings from the program's configuration file, press the `Load settings` button.

When launching the program, the settings from the configuration file will be loaded automatically. Use this button to cancel the changes in settings made with the help of the scanner's Graphical interface module only.

To save the changes in settings in the configuration file, press the `Save settings` button.

To load the default settings, press the `Reset all settings` button.

The settings of the Graphical interface module are also saved in the program's configuration file, in the `[GUI]` section. To learn more on the configuration file, read "Dr.Web® for Linux, FreeBSD, OpenBSD and Solaris. Administrator manual".

You can also set the scanning mode (the detail of the check level). For this, select one of the following variants in the `Check mode` group of buttons:

`Advanced mode` – in this mode, you can set all parameters specifying the detail of the check mode. The purpose of these parameters is described below.

`Full check` – this is the mode, when all the selected files are checked, archives including. The mode is recommended for every day computer check.

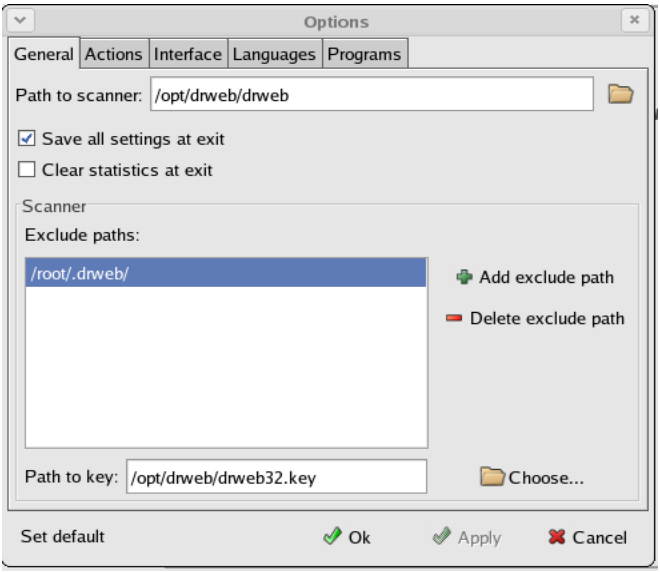
`Fast check` – this is the mode, when only the files the internal format of which lets consider them the "carriers" of viruses, are

checked; archives are not checked, the heuristic control is disabled. The check speed is higher than in the full check, due to some decrease of the control reliability.

4.2.2 Main program settings

Regardless the scanning mode selected, you can set the program's actions for detection of infected objects, as well as parameters of interaction with the OS and between different programs of the antivirus complex.

To view, and, if necessary, to edit the main settings, press the `Settings` button of the `Settings` pane of the main window. The `Options` window will open in the `General` pane (pic. 3).



Picture 3. General pane

In this pane you can specify the path to the scanner (as a rule, it is specified correctly at the installation and should not be edited). For

this, input the path into the `Path to scanner` field or press  and select it in the file manager.

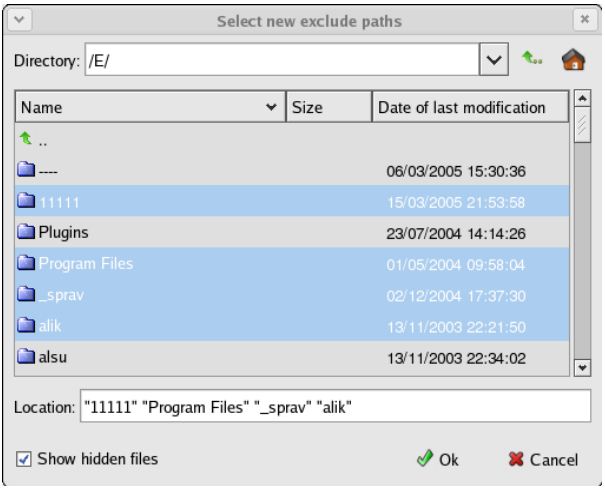
The same way, if necessary, specify the path to the license key file in the `Path to key` field.

Uncheck the `Save all settings at exit` box, if you want the settings to be saved to the configuration file when the corresponding button is pressed only. By default, the box is checked; the settings are also saved when the main window is closed.

Check the `Clear statistics at exit` box, if you want to keep statistics for each session only.

You can specify the list of paths excluded from the scanning. By default, the list contains only the directory used for the quarantine.

To add some directory or a file into the list, press the `Add exclude path` button. A window for selecting a path will open (pic. 4).



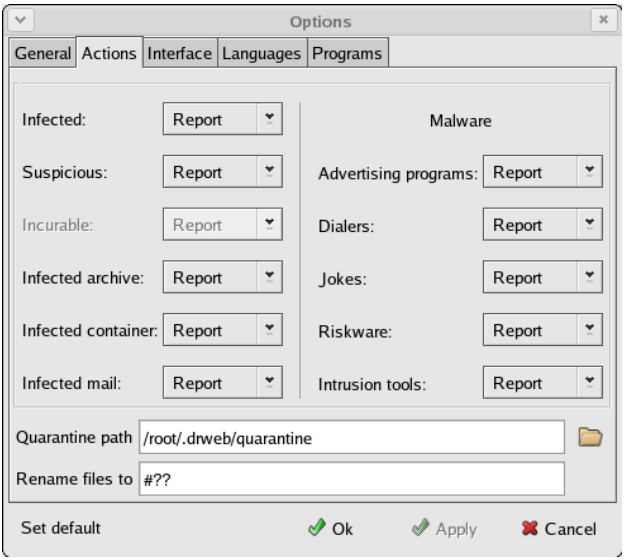
Picture 4. Adding the excluded path

To select the necessary files or directories, hold the [Ctrl] button and click the mouse over the corresponding objects (if just a single object should be selected, double click it). When the selection of necessary objects is finished, press the OK button.

By default, all files, including hidden files, are displayed in the file manager window. To forbid displaying hidden files, uncheck the Show hidden files box.

To remove any path from the list, select it in the list and then press the Delete exclude path button.

Select the Actions pane to set the program's reaction to detection of viral threats (pic. 5).



Picture 5. Setting actions

In the Infected dropdown list select an action to be made by the program when a file infected with a virus is found:

Report – report on an infection (read p. 4.2.3). A user may specify the reaction manually.

Cure – try to restore the state of the infected object before infection. If the attempt fails, or the action is impossible, the action for incurable objects will be applied (read below).

Delete – delete an infected file.

Quarantine – move an infected file to the quarantine directory (read below).

Rename – rename an infected file according to the renaming mask (read below).

By default, reporting is applied (Report action). We recommend to save it. The information on the detected files is reported to a user in the special field of the main window. A user may instruct to make necessary actions manually (read p. 4.3).

The same way the reaction is specified for detection of files infected with an incurable virus or suspicious for infection. In this case the Cure action is unavailable.

The Incurable option is available only when the Cure action is specified for infected files.

If an infected or suspicious file is found in file archives, mails or file containers, the program does not make any action with separate files in the archives of these types; still, you can specify the automatic actions with the archives of every type as a whole. The setting is the same as for incurable or suspicious files.

With the default settings, for archives of any type, the `Delete actions` is unavailable. Enabling of deletion of archives is specified in the advanced settings of the program, read p. 4.2.3.

In the right part of the window the program's action for detection of files with unsolicited programs of the following types is set:

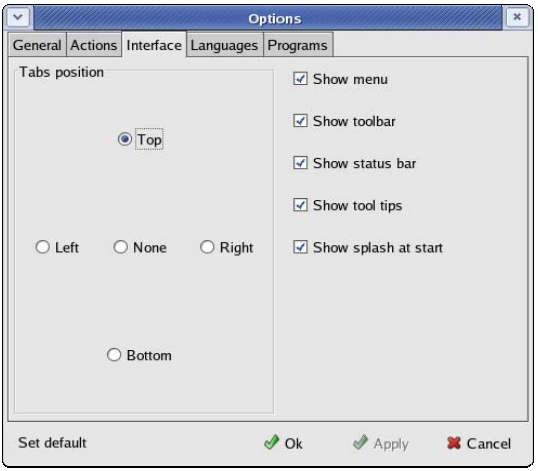
- advertising programs (displays advertisements)
- dialers (perform unauthorized connections to paid sites, often to porno sites)
- jokes (may threaten and draw away a user's attention)
- riskware (may be used by malefactors)
- hacktools (the tools for unauthorized access to computers and other electronic devices)

Setting of reactions in these cases is the same as setting actions for incurable or suspicious files, but on more `Ignore` actions is added.

You may edit the default path to the directory with moved objects (quarantine directory). Edit the path in the `Quarantine path` entry field or press  and select it in the file manager window (you may also create new directory).

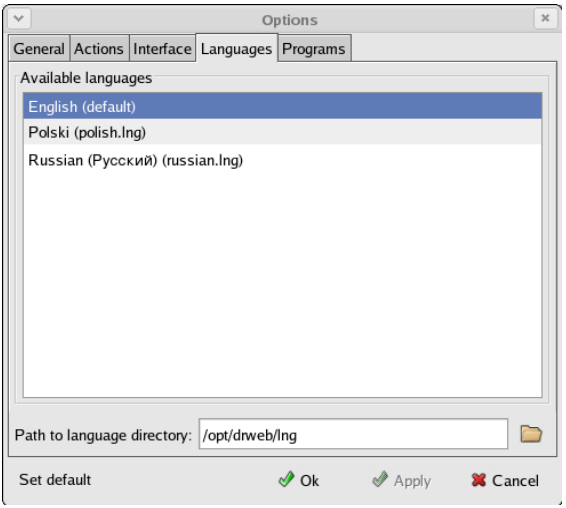
When renaming a file, its extension is replaced with the text specified by the renaming mask (by default, the mask looks like `#??`, i.e. the first symbol of extension is changed to `#`, the other remain unchanged). You may edit the mask in the `Rename files to` field.

Select the `Interface` pane to set the program's interface (pic. 6).



Picture 6. Setting the interface

Select the `Languages` pane to select the language of the program's interface (pic. 7).

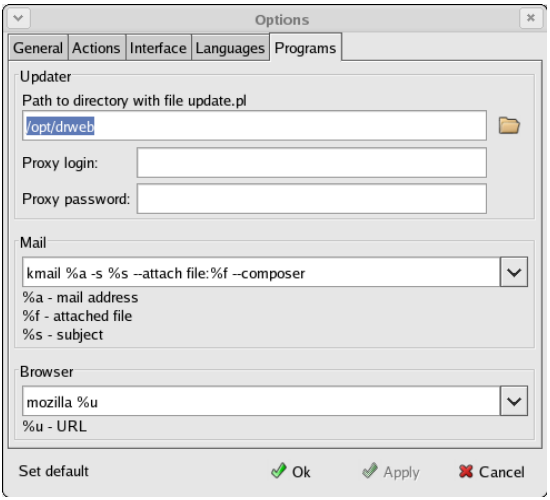


Picture 7. Setting the language

Choose the necessary language in the list of `Available languages`.

If necessary, edit the path to the language resources file in the `Path to language directory` field or press  and select it in the file manager window.

In the `Programs` pane you can set the parameters for interaction with the components of the antivirus complex and other programs (pic. 8).



Picture 8. Setting interaction with programs

If necessary, edit the path to the Updating module in the `Path to directory with file update.pl` field or press  and select it in the file manager window.

If a proxy server is used for the updates receipt, you may specify the login and the password to this server in the corresponding entry fields.

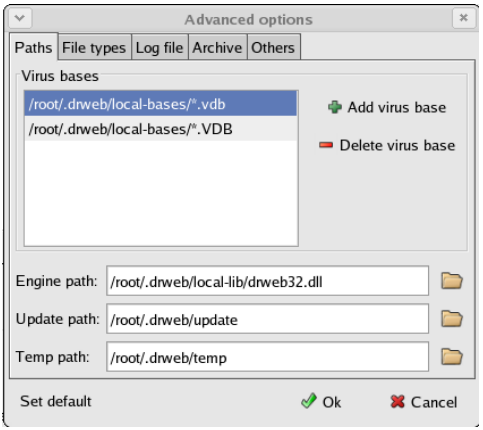
In the Mail field select and edit, if necessary, the command line for launching the mail program in the packet mode.

In the Browser field, select and edit, if necessary, the command line for launching the web browser.

When editing is finished, press the OK button to save the changes made.

4.2.3 Advanced settings

Advanced users may select Advanced mode for scanning (read above p. 4.2.1). The Advanced settings radio button becomes enabled (and the corresponding menu item of the Settings menu). Press the Advanced mode radio button. A window with advanced options in the Paths pane will open (pic. 9).

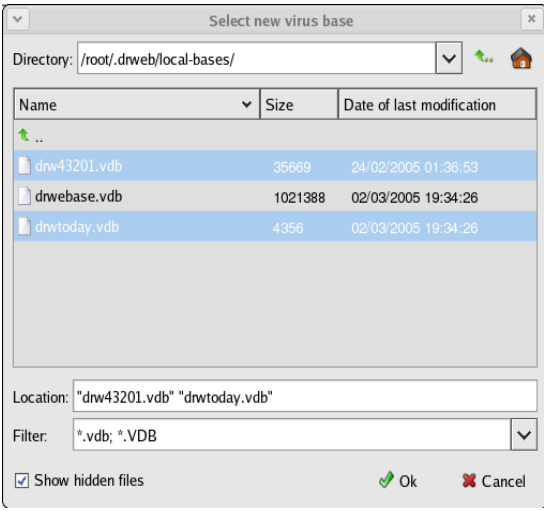


Picture 9. Setting paths

The Virus bases area lists the directories and files with the virus signature bases. By default, the bases are located in the directory specified during the program's installation in the configuration file. The additional bases are placed to the same directory by the

Updating module. In case of manual enabling of the bases, you should specify them in this list. When the bases have non-standard extension (even if they are located in the standard directory), they must be also included into this list.

To add an element into the list with the virus bases, press the `Add virus base` button. A window for adding a base will open (pic. 10).



Picture 10. Adding a virus base

To select the necessary files or directories, hold the [`Ctrl`] button and click the mouse over the corresponding objects (if just a single object should be selected, double click it). When the selection of necessary objects is finished, press the `OK` button.

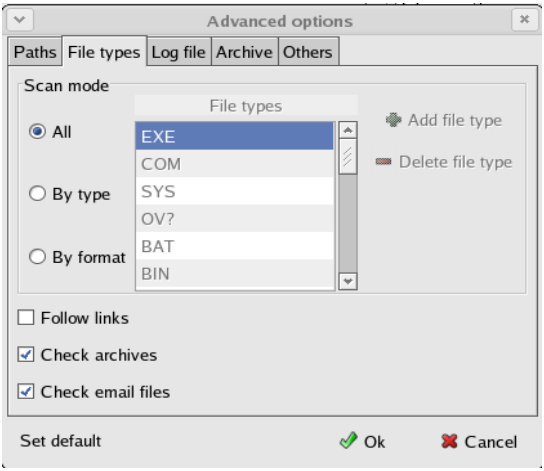
By default, all files, including hidden files, are displayed in the file manager window. To forbid displaying hidden files, uncheck the `Show hidden files` box.

The `Filter` dropdown list contains the mask for files of loaded bases. By default, there must be a list of two elements `*.vdb`; `*.VDB` (i.e. only files with `vdb` or `VDB` can be). You can also select the `*` value in the list (i.e. any files).

To exclude an element from the list, select it in the list and press the `Delete virus base` button.

If necessary, edit the paths to the antivirus engine, the updating directory and the temporary files directory in the corresponding fields.

Select the `File types` pane to set the restriction for files to be checked (pic. 11).



Picture 11. Selection of types of the scanned files

In the `Scan mode` group of buttons select the way for selecting files:

- `All` – all files are checked regardless their names and internal structure. This is the default mode in the `Full check` mode

- `By format` – the files, regardless their names, which can be carriers of viruses due to their internal structure are checked. This is the default mode in the `Fast check` mode
- `By type` – only files with specified in the `File types` extensions are checked. By default, the list contains extensions with executable files and files with macroses. To add the extension into the list, press the `Add file type` button and input the extension in the opened window; to exclude the extension from the list, select it and press the `Delete file type` button.

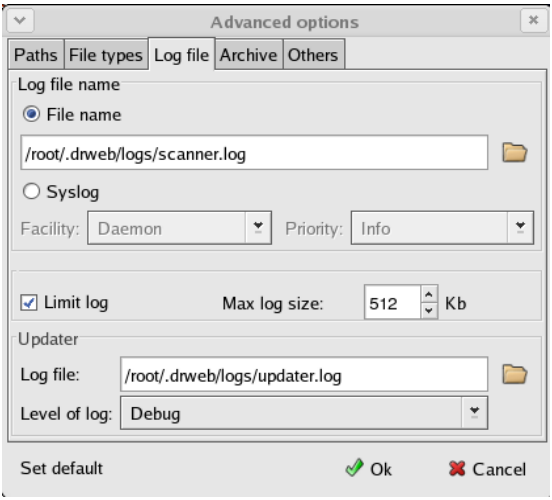
Check the `Follow links` box to instruct the scanner to check files the symbolic links to which fall within the checkable files.

Check the `Check archives` box to instruct the scanner to unpack file archives and check the files within them (in the `By format` mode — if they have the specified format, in the `By type` mode both the extension of an archive and the extension of the extracted file should be included into the list).

Check the `Check email files` box to instruct the scanner to check email files, including the attached files (the check of archives becomes enabled automatically).

In the `Full check` mode the above mentioned modes are enabled, in the `Fast check` mode they are disabled.

Select the `Log file` pane to set logging (pic. 12).



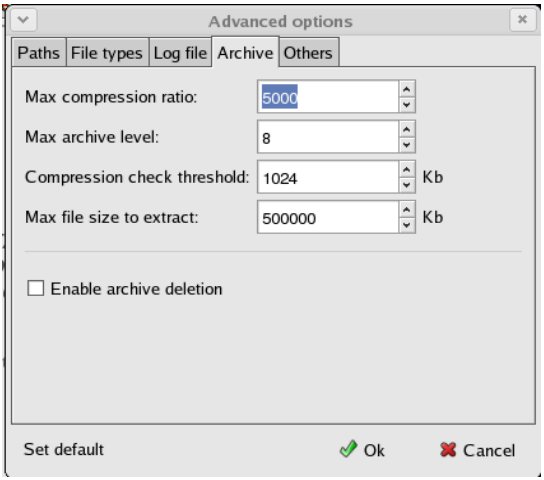
Picture 12. Setting the log file

In the `Log file name` group of buttons the mode of logging – into a file (`File name` variant) or using the Syslog system service is selected. In the first case, you can edit or select the path to the log file, in the second case you can select the logging facility and the priority.

We recommend to keep checked the default `Limit log` box and the value specified in the `Max log size` field (by default it is 512 Kb).

You can edit or select the Updater log file name in the `Updater` field and set the level of log.

Select the `Archives` pane to specify restrictions imposed onto the actions with archives for the security reasons (pic. 13).



Picture 13. Setting actions for archive

By default, the `Enable archive deletion` box is not checked and the `Delete` option for all types or archives is disabled in the `Actions` pane of the `Options` window (п. 4.2.2). To enable this action for archives, check this box.

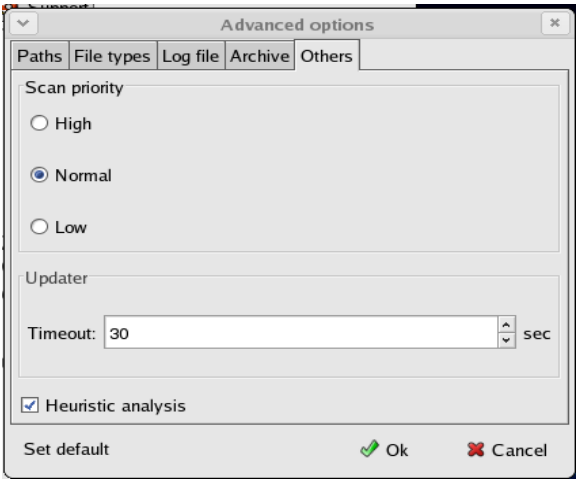
If the `Delete` action is selected, the scanner can automatically delete the archive with thousands of files, if just one infected or simply suspicious object is found. We strongly recommend to never enable deletion of archives.

The rest of parameters in the pane set the scanner's actions against "mail bombs" attacks. The parameters specify the characteristics of archives the exceeding of which terminates the scanning to avoid the exhaust of resources.

If the default settings must be changed, edit the values in the following fields:

- `Max compression ration` (by default, it is 5000)
- `Max archive level` (by default, it is 8)
- `Compression check threshold` (by default, it is 1024 Kb, smaller archives are checked regardless the compression ratio)
- `Max file size to extract` (by default, it is 500000 Kb, if greater file is detected, extraction terminates)

In the `Others` pane (pic. 14) the scanning priority, the updater timeout and some other parameters can be set.



Picture 14. Others pane

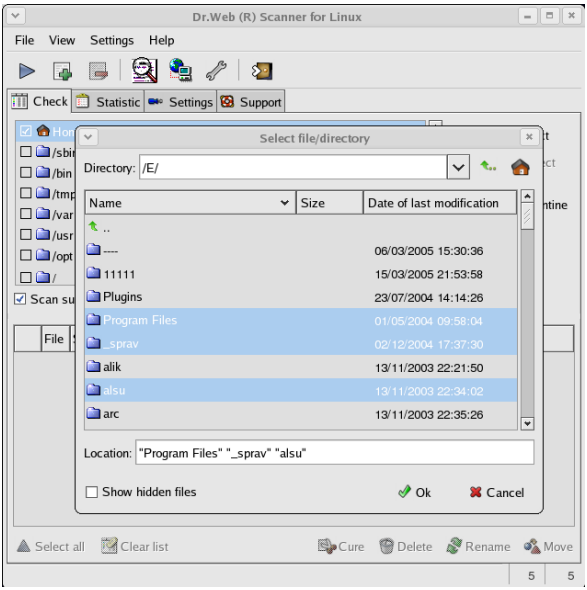
To enable the heuristic analysis (the search of unknown viruses, which may cause false alarms; the detected files are marked as "suspicious"), uncheck the `Heuristic analysis` box. In the `Fast check` mode it is enabled, in the `Full check` mode it is disabled.

4.3. Scanning under the Graphical interface module

The Graphical interface module can be used not only to facilitate the setting parameters of the scanner, but also for administration of the scanning process.

In the `Check` pane of the main window (see pic. 1 above) you can select the objects for scanning. By default, the directories of the upper level and the current user Home directory are listed there. The Home directory is selected and instructed to be checked, the other objects are not selected. To enable scanning of some objects from the list, you should select their names.

You can also edit the list of objects. To add the object into the list, press the `Add object` button. A window for selecting the objects for scanning will open (pic. 15).



Picture 15. Adding an element into the list for scanning

To select the necessary files or directories, hold the [Ctrl] button and click the mouse over the corresponding objects (if just a single object should be selected, double click it). When the selection of necessary objects is finished, press the OK button.

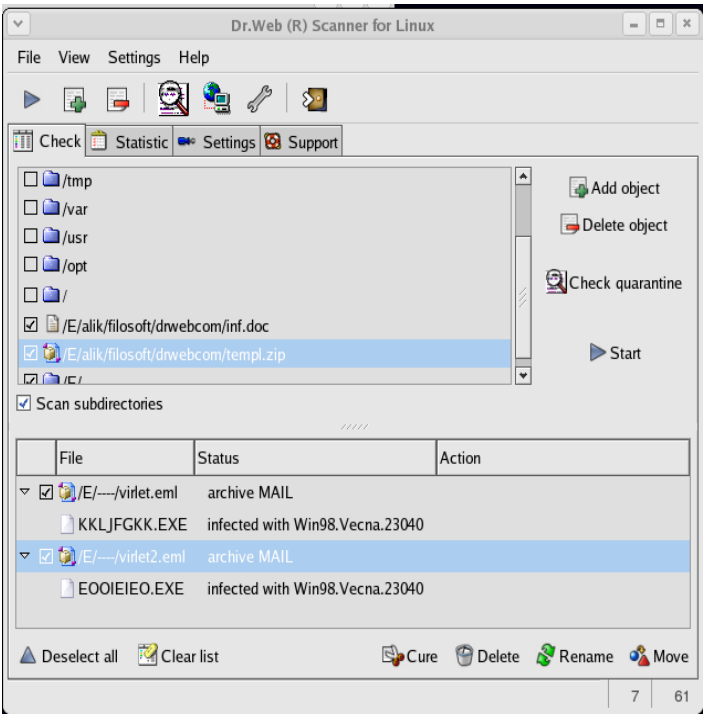
By default, all files, including hidden files, are displayed in the file manager window. To forbid displaying hidden files, uncheck the Show hidden files box.

To delete an object, select it in the list and press the Delete object button (only previously added by you objects can be deleted).

When editing the list of the object for scanning is finished, press the Start button.

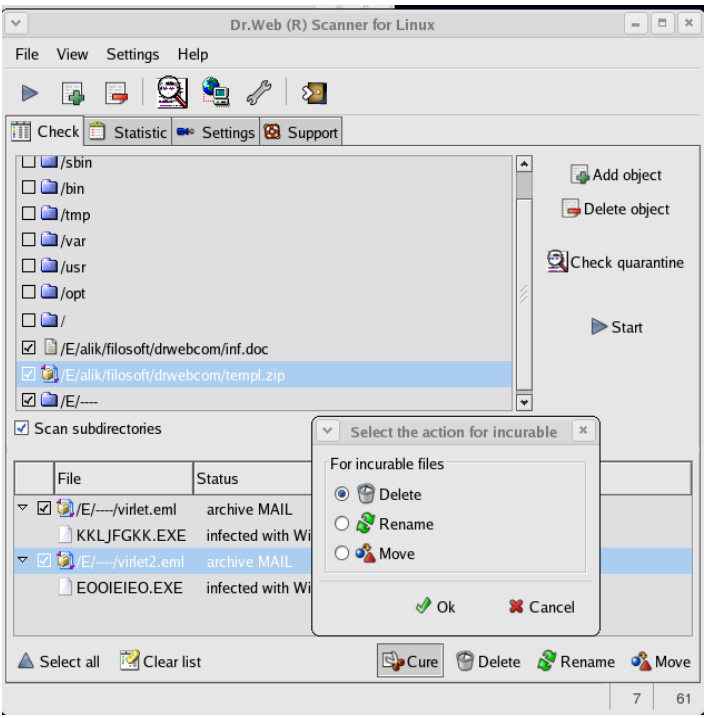
The scanning results are displayed in the report field table at the bottom of the main window. If different from the Report action was specified for the detected object, the result on the actions performed will be displayed in the Action column.

The list of detected objects is hierarchal; if a virus is detected in an archive, the infected archive is displayed as the nod of the hierarchy list, which can be unfolded or reduced (pic. 16).



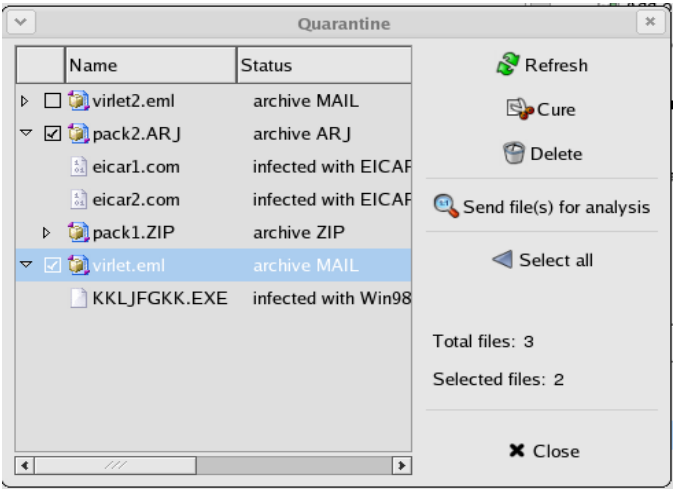
Picture 16. A virus is detected in a multilevel archive

To make the necessary action with the detected object manually, select it (or press the `Select all` button to select all detected objects) and press the `Cure`, `Delete`, `Rename` or `Move` button accordingly (the way these actions are performed is described in p. 4.2.2). If `Cure` is selected, additional window requesting an action, if curing fails will open (pic. 17).



Picture 17. Actions with infected objects

You may view files placed to the quarantine and specify actions for them. For this, press the `Check quarantine` button. The `Quarantine` window will open (pic. 18).



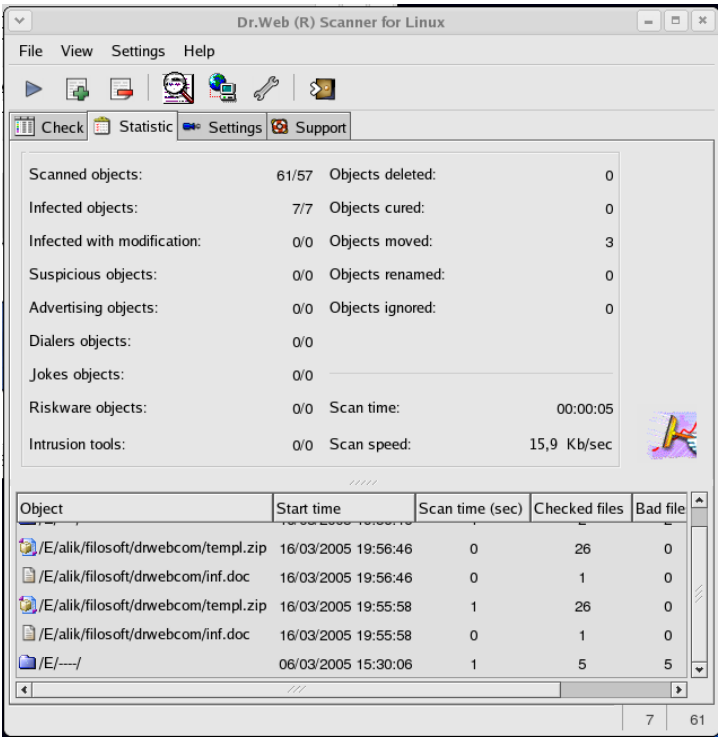
Picture 18. Quarantine

Select the files you want to take actions with (or press the `Select all` button). Press the `Cure` button to try to cure a file. Press the `Delete` button to delete a file. Press the `Send file(s) for analysis` to send files for analysis (using the installed mail client) to the technical support service of Doctor Web, Ltd.

If curing of a file fails, it is saved in the quarantine. You can delete it or send it for analysis, as described above.

4.4. Scanner's statistics

To view the scanner's statistics (depending on the program's current session settings or since the statistics' refreshment), select the `Statistics` pane (pic. 19).

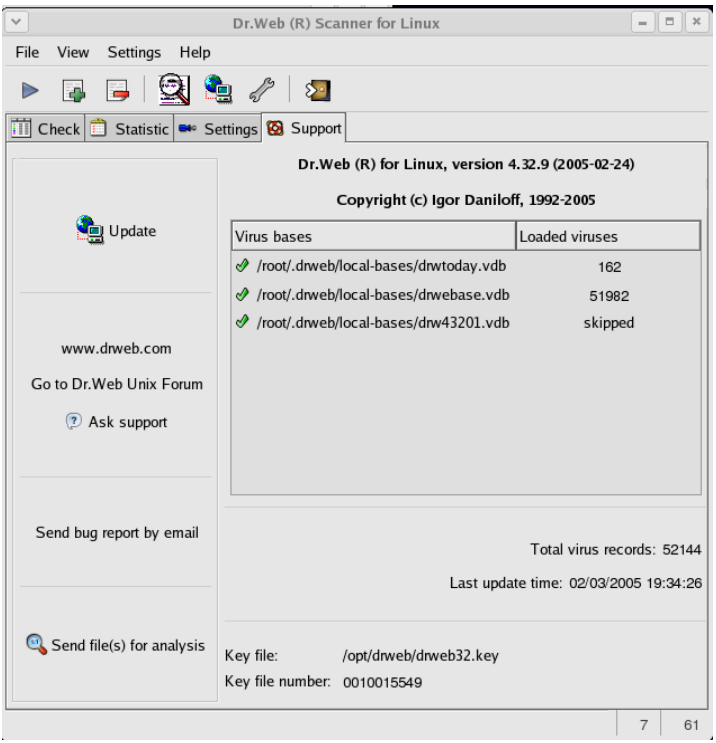


Picture 19. Statistics

To clear the statistics (reset the counters to zero), press  in the right part of the window.

4.5. About the program, updating and technical support

The Graphical interface module facilitates the updating of the program complex and application for the technical support. To use these services, select the `Support` pane (pic. 20).



Picture 20. Support pane

To run the Updating module, press the `Update` button in the left part of the window.

To open the browser at the web-site of Doctor Web, Ltd., Dr.Web Unix forum, to contact (through the HTML-web form) the technical support service or send a message by email, select the corresponding link in the left part of the window.

To send a file, presumably infected with an unknown virus, for analysis to Doctor Web, Ltd., press the

`Send file(s) for analysis` button. A window for selecting files will open.

In the right part of the window the information about the program can be viewed. The right part of the window contains information on the program version, on the bases loaded, the date of the last update and the key number. After the updating session, this information is refreshed.

If you tried to open one of the above listed web-sites, or send a message via email, a message that a browser or a mail program is not found was displayed, set the path to the mail program as it is described in the end of p. 4.2.2.

5 Using the Dr.Web® daemon

5.1. What is the Dr.Web® Daemon for Unix?

The Dr.Web Daemon is a constantly loaded antivirus module, which allows, by instruction from filtering programs to scan files on a drive or data transferred through a socket. The request is sent through a special protocol via unix- or tcp-sockets. In all other respects, the Dr.Web Daemon has similar with other products of the Dr.Web family functionalities:

- It uses the same kernel and virus bases as other scanners of the family
- Detects and cures all known viruses
- Checks packed files and archives

Besides, the Dr.Web Daemon has the option of mail filtering based on headers analysis.

Its constant readiness to function and comprehensible and simple protocol for scanning requests makes it an appropriate component for creating antivirus filters for mail systems and file servers. Developers of Dr.Web offer ready solutions for integration of the Dr.Web Daemon with CommuniGate Pro, Courier-MTA, Exim, Mobico MIO Server, Postfix, QMail, Sendmail, Zmailer mail systems, as well as with Samba file servers. There obviously may be other fields of application of the Dr.Web Daemon, those listed above are the principal by now.

The daemon is installed from the Dr.Web package; its installation procedure is described in the Dr.Web scanner documentation. It also includes the description of the program and the virus bases updating procedure.

5.2. The Dr.Web® Daemons command line options

As for any other Unix-program, the Dr.Web daemon supports command line options. They are separated from the specified path by a blank and prefixed by the – (hyphen) symbol. To get the complete list of options run the drweb program with the following options: `-?` or `-help`.

- `-ini=<file>` — using alternative ini-file;
- `-lng=<file>` — using alternative language file; if English interface is chosen during the installation specify `ru_daemon.dwl` as such file to display messages in Russian.

5.3. Configuring the Dr.Web® Daemon

You can run the daemon with the default settings, but it is better to set it in compliance with your requirements and conditions of operation. The `drweb32.ini` file is read by the Dr.Web Daemon from the `/etc/drweb` directory for Linux and Solaris, or from the `/usr/local/etc/drweb` directory for FreeBSD and OpenBSD. To use another configuration file, the full path to it should be specified by the command line key when running the Dr.Web Daemon, for example:

```
$ /usr/local/drweb/drwebd
                    -ini=/usr/local/drweb/drwebd.ini
```

the configuration file is described in details above in p. 3.4.

The settings for the daemon are in the [`Daemon`] section of this file.

Like for the scanner, the parameters will be specified in the following way:

`Parameter name` = `Pseudo Parameter Value`

Parameter description: {may have or not several values}

Default value: {value | `unspecified` }

Description of parameters in their succession mode in the configuration file created when installing the program:

`EnginePath` = {path to a file, usual extensions is `dll`}

The location of the `drweb32.dll` module (search module). This parameter is also used by the updating module to update the search module.

Default value: `%bin_dir/lib/drweb32.dll`

`VirusBase` = {list of paths (masks) to files, usual extension is `vdb`}

Masks for the linked up virus bases. This parameter is also used by the updating module to update the antivirus bases.

Listing of several masks is allowable.

Default value: `%var_dir/bases`

`UpdatePath` = {directory}

This parameter is used by the program's updating module (update.pl) and should be obligatory specified, if it is used.

Default value: `unspecified`

`TempPath` = {directory}

This directory is used by the antivirus module (search engine) to create temporary files. When normally operated, this directory is almost not used, it is used for unpacking certain types of archives, pr when a system lacks memory resources.

Default value: empty (used `/tmp`).

LngFileName = {path to the language resource file, usual extension — `dwl`}

Location of the localization file.

Default value: empty (in this case the dialogs will be displayed in English).

Key = {path to a file, usual extension is `key`}

Location of the key file (license or demo).

Default value: `%bin_dir/drweb32.key`

The key file may differ for the daemon and the scanner, therefore, if necessary, the settings for this parameter should be modified.

MailAddressesList = {path to a file}

This parameter is used only if you have a "per email" license for 15 or 30 addresses. A list of e-mail addresses (it should not exceed the licensed number), which will be checked (both incoming and out-going messages) should be specified in this file. The format of the file is simple – one line for one address. Aliases are considered as separate addresses.

Default value: empty.

OutputMode = {`Terminal` | `Quiet`}

The information output mode at start: `Terminal` – to console, `Quiet` cancels output.

Default value: `Terminal`.

RunForeground = {`Yes` | `No`}

The `Yes` value of this parameter disables the daemon's mode of the Dr.Web Daemon, i.e. its acting in the background without the controlling terminal. This option can be used by certain monitoring tools (or example, by daemon tools).

Default value: `No`.

User = {user name}

Defines a user whose rights the daemon is run under. It is recommended to add a special drweb user account on the host, which will be used in future by the daemon and some other filters. To use the Dr.Web Daemon with root rights is undesirable, still such solution requires less settings (especially in Samba-servers).

The value of this parameter is not subject to adjustment during the configuration reload "on the fly" (SIGHUP handling).

Default value: empty which means the daemon will use the privileges of the user it was run with.

UserID = {numeric ID}

GroupID = {numeric ID}

ID of a group or of a user the daemon will use the rights to operate. The parameters are ignored, if the `User` parameter is specified.

The values of these parameters should not be changed during the configuration reload "on the fly" (SIGHUP handling).

Default value: empty

PidFile = {path to a file}

A name of a file where the daemon's PID and socket will be written to at start (of the `Socket` parameter enables usage of the unix-socket) or the port number (if the `Socket` parameter enables usage of the tcp-socket).

If several `Socket` parameters are specified, this file will contain information on all sockets set (one per each line).

Default value: `%var_dir/drwebd.pid`

`BusyFile` = {path to a file}

Daemon's busy file name: it is created by a daemon's scanning "copy" if commanded and removed after transmission of the result of its execution. The filename created by each "copy" of the daemon, supplemented with a point and ASCIIZ representation of PID (e.g., `/var/run/drwebd.bsy.123456`).

Default value: empty (file is not created).

`MaxChildren` = {integer}

Sets maximum number of simultaneously running child scanning processes. The main process does not perform the scan, thus the maximum number of the daemon's processes in the system will be 1 process greater than the set value. Recommended value range – from 3 to 16 processes per CPU.

Default value: `16`.

`PreFork` = {`Yes` | `No`}

Sets mode of child processes spawn procedure. If the parameter is set to `No`, new child process will be spawned for each scanned object. If the parameter is set to `Yes`, the Dr.Web Daemon will create as many child processes, as equals to the `MaxChildren` value (explained above) immediately after start. PreFork mode is a highly productive process, but consumes more memory resources (as child processes are always present in memory).

The value of this parameter cannot be changed during the configuration reload "on the fly" (SIGHUP handling).

Default value: `No`.

`MailCommand` = `{command}`

The command used by the daemon and the updater to send notifications to a user (administrator) via e-mail. The daemon uses this feature at every start (restart, reboot), if less than two weeks till the key (one of key files) expires left. The daemon sends a message to the standard input stream of the command. The updater uses this feature to send information bulletins by Doctor Web, Ltd., including also on the updates procedures of the program files.

Example:

```
MailCommand = "/usr/sbin/sendmail -i -bm -f drweb -- root"
```

Default value: empty.

`FileTimeout` = {value in seconds}

Maximum file scan time.

Default value: `30`.

`StopOnFirstInfected` = {`Yes` | `No`}

Cancel or not message checking after the first virus is detected. The value set to `Yes` may considerably minimize the mail server load and messages scan time.

Default value: `No`.

`ScanPriority` = {value}

Priority of the daemon's scanning processes. The value of this parameter should be within – 20 (highest priority) to 20 (lowest priority).

Default value: `0`.

`FilesTypes` = {list of extensions}

Types of files to be checked at scanning "by type", i.e. when the `ScanFiles` parameter (explained below) is set to `ByType`. The "*" and "?" symbols are allowable.

Several lines with such parameter name are allowable; in this case the lines are summed up.

Default value: `EXE, COM, SYS, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, VXD, 386, DLL, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, AR?, ZIP, R??, PP?, OBJ, LIB, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SHS, SHB, PIF, SO, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, BMP, MPP, OCX, DVB, CPY, MSG, EML`.

`ScanFiles` = {`All` | `ByType` | `ByFormat`}

Additional restriction for files to be checked. With the `ByType` value set the so-called file extensions (i.e. the last suffix after "`.`" in the filename, if "`.`" is absent it means the file has no extension), the values of which are specified or set either by default, or in the `FilesTypes` parameter (parameters), are taken into account. When setting the `ByFormat` value for files included by the scanner startup command, these files are checked whether or not they can be carriers of viruses, i.e., executable files (the name of the file and the extension are disregarded), and after that only the antivirus scanning will be performed for files presumably considered executable.

The `All` mode is always enabled in mail files. The `ByType` and `ByFormat` values can be interpreted only in the "local" scanning modes.

Default value: `All`.

`CheckPackedFiles` = {`Yes` | `No`}

Whether or not unpacking of executable files packed by DIET, PKLITE, etc. utilities should be performed.

Default value: `Yes`.

CheckArchives = {Yes | No}

Whether or not unpacking of the archives in ZIP (WinZip, InfoZIP...), RAR, ARJ, TAR, GZIP and CAB formats should be performed.

Default value: Yes.

CheckEMailFiles = {Yes | No}

Whether or not files in e-mail formats should be scanned.

Default value: Yes.

ExcludePaths = {list of paths (masks) to be excluded from check}

Masks for files that should not be checked.

Default value: empty.

FollowLinks = {Yes | No}

Should the symbolic links be followed when scanning.

Default value: No.

RenameFilesTo = {mask}

Masks for renaming files if the Rename value is set for the given condition (an infected or suspicious file).

Default value: "#??" which means that the first character of a file "extension" will be replaced by "#", and two subsequent characters will remain. If the file has no “extension” it will consist of a "#" symbol.

MoveFilesTo = {directory}

Directory for moved files. This parameter is used only when the daemon is integrated with on-access monitor for Samba.

Default value: %var_dir/infected.

BackupFilesTo = {path}

This directory is used to backup infected files, which have been cured.

Default value: empty (i.e. a copy is not created).

LogFileName = {name of a file}

Log filename. You may specify `syslog` for the log filename and the logging will be done by means of the syslogd system service. When using syslog the `SyslogFacility` and `SyslogPriority` parameters (explained below) should be taken into account. As syslog utilizes several files for logging different events and different degrees of their importance, these two parameters and the content of the syslog configuration file (usually `/etc/syslogd.conf`) determine the location where the report will be logged to.

Default value: `syslog`.

SyslogFacility = {`Daemon` | `Local0` .. `Local7` | `Kern` | `User` | `Mail`}

Type of record when using syslogd system service.

Default value: `User`.

SyslogPriority = {`Alert` | `Warning` | `Notice` | `Info` | `Error`}

Priority of record when using syslogd system service.

Default value: `Info`.

LimitLog = {`Yes` | `No`}

The parameter determines whether or not to limit the log file size.

Default value: `No`.

MaxLogSize = {value in Kb}

This parameter sets the maximum log file size. It is used only if `LimitLog = Yes`.

Default value: `512`.

LogScanned = {`Yes` | `No`}

Whether or not information on all scanned objects, regardless the viruses were detected or not, should be logged.

Default value: `No`.

LogPacked = {`Yes` | `No`}

Whether or not additional information on files packed with DIET, PKLITE, etc. utilities should be logged.

Default value: `Yes`.

LogArchived = {`Yes` | `No`}

Whether or not additional information on the archiving tools should be logged.

Default value: `Yes`.

LogTime = {`Yes` | `No`}

Whether or not the time for each record should be logged. The parameter is not used if the `LogFileName` is set to `syslog`.

Default value: `Yes`.

LogProcessInfo = {`Yes` | `No`}

Whether or not each record in the log file should be prepended by pid of the scanning process and a filter address (host name or IP) the scanning has been activated from.

Default value: `Yes`.

RecodeNonprintable = {Yes | No}

The output mode of invisible for this terminal symbols into the log file.

Default value: Yes.

RecodeMode = {Replace | QuotedPrintable}

If RecodeNonprintable is set to Yes, the recode mode of invisible symbols is set. If the value is set to Replace, all these symbols will be replaced with the RecodeChar paramete value (explained below).

Default value: Replace.

RecodeChar = {"?" | "_" | ...}

If RecodeMode is set to "Replace", a symbol, which will replace all invisible symbols, is set.

Default value: "_".

Socket = {PORT [interfaces] | FILE [access]}

Description of a socket used for communication with the daemon. The first form describes a tcp-socket: parameter "PORT" — decimal port number, "interfaces" — list of listening interface names or IP-addresses.

Example:

```
Socket = 3000 127.0.0.1, 192.168.0.100
```

The second form describes unix-sockets, "FILE" — socket name, "access" — octal value of access rights.

Example:

```
Socket = %var_dir/.drwebd 0660
```

The number of "Socket..." keys is not limited; the daemon will work with all correctly described sockets.

Default value: 3000 127.0.0.1.

SocketTimeout = {value in seconds}

Time of the data receipt/transmission via the socket (file scanning time is disregarded).

Default value: 10.

ListeningQueue = {value}

Defines sockets queue maximum size. The value may vary from 0 to SOMAXCONN (its value depends on the OS version).

Default value: overridden by the SOMAXCONN value.

The following parameters can be used to reduce the archives' scan time (some objects in the archive will not be checked). If an object falls under the restriction set by this parameter the ArchiveRestriction action specified in configuration files of different filters is applied.

MaxCompressionRatio = {value}

Maximum compression ratio, i.e. the ratio of the unpacked file size to the packed file size (inside an archive). If the ratio exceeds the value the file will neither be extracted, nor checked.

Default value: empty (all files are checked).

CompressionCheckThreshold = {value in Kb}

Minimum size of a file inside an archive, beginning from which the compression ratio will be checked (if it is specified by the MaxCompressionRatio parameter).

Default value: empty (the parameter is not checked).

MaxFileSizeToExtract = {value in Kb}

Maximum size of a file extracted from an archive. If the size of the file inside the archive exceeds this value it will be skipped.

Default value: empty (files of any size get extracted).

MaxArchiveLevel = {value}

The maximum archive's nesting level (archive in archive, and then in archive, etc.)

Default value: empty (archives of any nesting level get extracted).

As it was stated above, the Dr.Web Daemon has the embedded options of mail messages filtrering based on the messages' headers analysis. Filtering rules are also specified in the configuration file, the rules get checked the way they are set, i.e. the rule set first is checked first. Setting the filtering rules does not mean they will be automatically applied. To enable the headers' analysis you should specify certain options in the delivered filters, or set special flags, if using customized settings on the basis of the Dr.Web Daemon. The search of correspondence to the rule is made till the first appropriate rule is found — the action set for the rule is returned.

ScanEncodedHeaders = {`Yes` | `No`}

Process or not message headers before decoding. For example, the `Yes` value and the

```
RejectCondition Subject = "iso-8859-5"
```

rule allow filtering of all messages with the Subject field in iso-8859-5 encoding. Be aware that with the `Yes` value set all headers will be scanned twice: before decoding and after it.

Default value: `No`.

RejectCondition {set of rules}

AcceptCondition {set of rules}

Description of filter rules by message headers. The rules consist of a header name and a regular expression describing the given field value. Several rules can be combined by round brackets and `OR` or `AND` operations.

Example:

```
RejectCondition Subject = "money"
                    AND "Content-Type" = "text/html"
```

Special filtering rules include conditions "`No HEADER`" (means absence of this field, e.g., following the rule "`RejectCondition No From`" the messages without the "`From`" field will be filtered), `HEADER = "8bit"` (the field contains 8-bit symbols).

Default value: empty.

`MissingHeader` = {list of fields}

Describes the list of header fields the absence of which in the message will enable filtering the message. Example:

```
MissingHeader "To", "From"
```

Default value: empty.

`FilterParts` = {`Yes` | `No`}

The `Yes` value allows using the rules set by the `RejectPartCondition` and `AcceptPartCondition` keys.

Default value: `No`.

`RejectPartCondition` {set of rules},
`AcceptPartCondition` {set of rules}

The parameters are similar to `RejectCondition` and `AcceptCondition`, but used for particular message parts. The rule set for these parameters admits using `FileName` = "mask", where "mask" is a POSIX 1003.2 compatible regular expression. Messages filtering by these rules is allowable only when the FilterParts key is set to `Yes` (explained above).

Default value: left empty.

5.4. Starting the Dr.Web® Daemon

The Daemon starts as follows:

- the configuration file is searched and loaded; if it is not found, the Dr.Web Daemon terminates loading. The path to the configuration file can be specified in the startup parameters by the key `-ini: {path/to/your/drweb32.ini}`, or the default value (`/etc/drweb/drweb32.ini`) can be used. Several parameters get validated at start, and, if the parameter value is not allowable the default value is applied
- the localization file is loaded, if it is specified; if not, English dialogs are used
- log file is created. Note: a user under the rights of which the daemon operates should have a write access to the directory where the log file is located
- The key file is loaded to the location specified in the configuration file. If the key file is not found, the daemon terminates loading
- if the User (or UserID) parameter is set, the daemon tries to modify its privileges
- the search module (`drweb32.dll`) is loaded. If the module is not found (errors in configuration file) or damaged, the loading terminates
- after that, the bases with the virus definitions are loaded. The bases are searched in the locations specified in the configuration file; their loading order is not fixed. If the base is damaged or absent the loading will continue
- the daemon "comes untied" of the terminal, that is why the reports on further problems cannot not be output to the terminal and get reported to the log file only

- a socket is created. In case with tcp, there may be more than one of them; if any of them was not created the loading continues. In case with unix-socket: make sure a user under the rights of which the daemon operates has a read and write access to the directory containing the socket. The users whose rights are used by the integrating modules should have execution access to the directory and write and read access to the socket file
- then, the so-called pid-file is created; it stores information on the descriptor of the daemon's process and the transport addresses at which the daemon is accessible. There is one more important note: a user under the rights of which the daemon operates should have a write access to the directory with pid-file, that is why if the User parameter is specified, you should also obligatory reset the pid-file parameter, as the default `/var/run` directory has no write access for users. If pid-file was not created, the loading terminates

5.5. Verifying availability of the Dr.Web® Daemon

The error free loading results in the ready-to-work daemon, i.e. a fixed number of scanners always ready to operate. You can check whether a socket is created by giving the `netstat -a` command. If tcp-sockets are used:

```
--- cut ---
Active Internet connections (servers and
                                        established)
Proto Recv-Q Send-Q Local Address Foreign Address
                                                State
tcp    0   0 localhost:3000   *:*  LISTEN
raw        0      0 *:icmp     *:*    7
raw        0      0 *:tcp      *:*    7
```

```
Active UNIX domain sockets (servers and
                                          established)
Proto RefCnt Flags  Type  State   I-Node Path
unix  0  [ ACC ]  STREAM  LISTENING  384
                                            /dev/gpmctl
unix  0  [ ]  STREAM  CONNECTED  190  @0000001b
unix  1  [ ]  STREAM  CONNECTED  1091  @00000031
unix  0  [ ACC ]  STREAM  LISTENING  403
                                /tmp/.font-unix/fs7100
unix  4  [ ]  DGRAM            293       /dev/log
unix  1  [ ]  STREAM  CONNECTED  1092
                                            /dev/gpmctl
unix  0      [ ]         DGRAM      450
unix  0      [ ]         DGRAM      433
unix  0      [ ]         DGRAM      416
unix  0      [ ]         DGRAM      308
--- cut ---
```

If unix-sockets are used:

```
--- cut ---
Active Internet connections (servers
                                       and established)
Proto Recv-Q Send-Q Local Address  Foreign
                                         Address  State
raw        0      0 *:icmp     *:*    7
raw        0      0 *:tcp      *:*    7
Active UNIX domain sockets   (servers
                                       and established)
Proto RefCnt Flags  Type  State   I-Node Path
unix  0    [ ACC ]  STREAM    LISTENING     384
                                          /dev/gpmctl
unix  0      [ ]    STREAM    CONNECTED     190
                                           @0000001b
unix  1      [ ]    STREAM    CONNECTED     1091
                                           @00000031
unix  0    [ ACC ]  STREAM    LISTENING     1127
                          /opt/drweb/run/drwebd.skt
unix  0    [ ACC ]  STREAM    LISTENING      403
                                /tmp/.font-unix/fs7100
unix  4      [ ]    DGRAM         293    /dev/log
unix  1      [ ]    STREAM     CONNECTED     1092
                                          /dev/gpmctl
unix  0      [ ]         DGRAM       450
```

```
unix  0      [ ]         DGRAM       433
unix  0      [ ]         DGRAM       416
unix  0      [ ]         DGRAM       308
--- cut ---
```

If there are no created sockets viewed, it means there have occurred some problems at loading.

To check the availability of the Dr.Web Daemon use the console client for the daemon (drwebdc); run it to obtain the service information on the daemon. If drwebdc is started with empty parameters the list of all supported options will be printed.

For tcp-socket:

```
$ drwebdc -nHOSTNAME -pPORTNUM -sv -sb
```

For unix-socket:

```
$ drwebdc -uSOCKETFILE -sv -sb
```

the following information should appear:

```
--- cut ---
- Version: DrWeb Daemon 4.32
- Loaded bases:
Base /var/drweb/bases/drwtoday.vdb
                                  contains 5 records.
Base /var/drweb/bases/drw43203.vdb
                                 contains 409 records.
Base /var/drweb/bases/drw43202.vdb
                                 contains 543 records.
Base /var/drweb/bases/drwebase.vdb
                               contains 51982 records.
Base /var/drweb/bases/drw43201.vdb
                                 contains 364 records.
Total 53303 virus-finding records.
--- cut ---
```

If such information is not displayed, try to enable the enhanced diagnostics. For tcp-socket:

```
$ drwebdc -nHOSTNAME -pPORTNUM -sv -sb
```

For unix-socket:

```
$ drwebdc -uSOCKETFILE -sv -sb
```

The more detailed output will clarify the situation:

```
--- cut ---
dwlib: fd: connect() failed - Connection refused
dwlib: tcp: connecting to 127.0.0.1:3300 - failed
dwlib: cannot create connection with a DrWeb daemon
ERROR: cannot retrieve daemon version
Error -12
--- cut ---
```

Then, take the `readme.eicar` test file (included into the distribution), make the eicar.com program in a text editor (see instructions inside the file) and try to check it by the daemon:

if you have license for mail server (50 and greater addresses):

For tcp-socket:

```
drwebdc -nHOSTNAME -pPORTNUM -e -f eicar.com
```

For unix-socket:

```
drwebdc -uSOCKETFILE -e -f eicar.com
```

if you have license for mail server (15 or 30 addresses):

For tcp-socket:

```
drwebdc -nHOSTNAME -pPORTNUM -e -FEMAIL_ADDRESS
                        -REMAIL_ADDRESS -f eicar.com
```

For unix-socket:

```
drwebdc -uSOCKETFILE -e -FEMAIL_ADDRESS
                        -REMAIL_ADDRESS -f eicar.com
```

(where `EMAIL_ADDRESS`, one of "protected" addresses (from `email.ini`))

if you have license for file server or internet-gateway:

For tcp-socket:

```
drwebdc -nHOSTNAME -pPORTNUM -f eicar.com
```

For unix-socket:

```
drwebdc -uSOCKETFILE -f eicar.com
```

The command must result in the following diagnostics:

```
--- cut ---
Results: daemon return code 0x20
                                (known virus is found)
--- cut ---
```

If failed, you should:

- study the daemon's log file for the record on the check of this file
- enable enhanced diagnostics (explained above)

If the check was successful, it means that the daemon is operable.

5.6. Check modes of the Dr.Web® Daemon (local scanning)

The Dr.Web Daemon provides for two basic check modes:

- check of the memory fragment received from a socket
- check of a file on a disk

Let us treat in details these two modes. The first mode enabled, the Dr.Web Daemon obtains data for check from a socket; actually, it is a fragment of data. This fragment can be named or not, — this will affect only the form of logging in the Dr.Web Daemon's log file. Read the previous section for the example of such operation of the daemon — the client read the file and sent it to the daemon for check. Basically, the daemon can check any data fragment; it should not always be a file.

But much more efficient is the mode when the daemon checks the definite file on a disk, — i.e. the client (be it a console client, or a mail filter) reports to the daemon the path to a file only, but does not transmit the whole file.

The path should be specified in relation to the daemon (as clients can be located on other computers, etc.).

What are the advantages of such mode? Firstly, it is more productive; secondly, it is much easier to create an operational circuitry with curing (e.g., on file servers). But this mode requires a careful rights setting, as the daemon should have a read access to such file; in case with mail files, if the curing and deletion are enabled it should also have a write access. In the skilfully set system the daemon, in most cases, does not require administrator's (root) rights. But you should pay special attention to it when using it with mail servers, because filters usually run under the name of the mail system (which also does not use root rights). In the most favourable mode the filter creates a file with a message (having received it from the mail system) and reports its location to the daemon. At this point the right access the directory where filters will create files should be carefully allotted. We would recommend either to enclose the user whose rights are used by the daemon into the mail subsystem group, or run the daemon under the mail system user.

6 Integration of the Dr.Web® Daemon with mail systems

6.1 Dr.Web® Daemon and CommuniGate Pro mail system

6.1.1 Requirements

- CommuniGate Pro (with Content Filtering support)
- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- drweb-cgp plug-in

6.1.2 Enabling support of drweb-cgp in CommuniGate Pro

To enable checking of the content of the delivered messages by CommuniGate Pro (further named as CGP) with the help of the antivirus daemon drwebd you should perform the following:

- Connect to CGP through WebAdmin
- Follow `Settings -> General -> Helpers`
- Choose `Content Filtering` and specify the full path to the agent program (check if the privileges with which CGP is executed are sufficient to start drweb-cgp and add start options, if necessary (explained below)). Then, to enable total message filtering you should get to `Settings -> Rules`. Create a new rule: choose its name first (e.g. drweb-filter) and press `Create New`, then set the `Action` field in `External Filter`, in the `Parameters` field put in the `Filter` field value: from `Settings -> General -> Helpers`.

For more detailed settings (especially, enabling/disabling filtration for every separate user) read documentation delivered with CGP.

6.1.3 Configuring drweb-cgp plug-in

Plug-in parameters are set up in the configuration file. If started with no parameters, the plug-in is trying to find the configuration file in standard (for the module) directories, which you can easily find out by starting the plug-in from the console:

```
$ drweb-cgp --help
```

If the given location of the configuration file does not suit you by any reason, you can explicitly set up the configuration filename by specifying in Content Filtering not only the path to the filter, but also the parameter

```
{path/to}/drweb-cgp –-conf={path/to/conf/file}
```

{path/to/conf/file} must contain a filename too.

For the description of the plug-in configuration file and notification templates read conf_file.txt and notify.txt accordingly.

To get the plug-in version run the plug-in:`$ drweb-cgp --version`

To test correctness of the plug-in configuration file run the plug-in:

```
$ drweb-cgp --check_only
```

The filter will run, read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

The Dr.Web daemon (drwebd) should run under the root account, or the `LocalScan=no` mode should be used, as CGP creates files in queue accessible by root only.

6.1.4 Configuring timeouts

Correct timeouts setup is an important factor of a stable interaction of CommuniGate mail system and the Dr.Web antivirus.

Timeout from the [DaemonCommunication] section of drweb_cgp.conf should be twice greater than SocketTimeout and FileTimeout from drweb32.ini taken together. If your version of CommuniGate supports timeout for filtering modules, then timeout from Settings -> General -> Helpers of the drweb filter settings should be greater than timeout from the [DaemonCommunication] section of drweb_cgp.conf

6.1.5 Known problems

Under Linux after changing (and updating) the command line in Helpers the previous filter process remains in zombie state until the next reloading of CGP.

Please don't use positional independent rules for header filtering (see description RejectCondition for drweb32.ini) for Subject: header, because notifications (at least to sender) contain original prefixed subject header and notification will be blocked again.

6.2 Dr.Web® Daemon and Sendmail mail system

6.2.1 Requirements

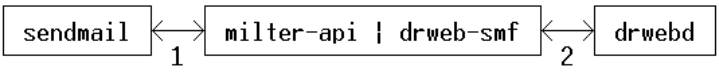
- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- drweb-smf filter
- Source texts of Sendmail version 8.11 or higher. The filter does not work with versions prior to 8.12.0 Beta1-Beta9 due to the error in Sendmail corrected in subsequent versions. They are required only if Sendmail has no Milter API support; some distributors (e.g., RedHat) now deliver Sendmail with Milter API support pre-included

6.2.2 Providing interaction of Sendmail and drweb-smf

Suppose that your Sendmail already supports MilterAPI; otherwise read p. 6.2.3, where it is described how to re-build Sendmail with necessary API support.

6.2.2.1 The way it works

The operation scheme of Sendmail Mail Filter API, pic. 21.



Picture21. Scheme of interaction of drwebd with Sendmail

1. Through the transport connection identified by drweb-smf by its transport address `__ADDRESS__` (specified in the previous topic), the inner Milter API commands and the message are transferred. As for the message, it is transferred not as a whole, but by parts, depending on the mailing session phase: helo, mail from:, rcpt to: etc. That is why the filter saves the message in temporary files (see the `-f` option). Through Milter API drweb-smf reports what action should be taken to the message. Milter API is a multi-stream library and thus several mailing sessions can be processed simultaneously.
2. So, considering this interaction, Sendmail is a client, and the mail filter performs the server function, that is why the filter address should be specified in (`sendmail.cf` and in the filter command line), and Sendmail will choose for this connection an appropriate client's address.
3. Through another transport connection the Dr.Web Daemon API drweb-smf transfers commands (and the message, if the `-l` option has not enabled the local mode) to the

daemon and waits for response. If a positive answer is received (the message is "clean"), a temporary file is deleted and Sendmail gets the permission to receive this message. If a virus is found, the message is moved to the infected messages archive (see option -a); instead of sending the message to the addressees, and (optionally) the sender (option -x) and administrator (option -g), notifications are sent only. In case of errors during the message processing the -b option is activated. Regarding the given interaction the filter acts as a client, while the daemon operates as a server that is why the address is specified in the daemon's ini-file and in the filter command line.

To sum up, drweb-smf is just an agent (or a converter) between the Sendmail interface and DrWebd. All three programs can be run on different computers. Still, the optimal variant will be to run the filter and the daemon on one and the same computer; with the option enabled that will allow to transfer the message only once, as the daemon will scan it as a local disk file.

6.2.2.2 Modifying sendmail.mc and sendmail.cf

If you do not want to re-build `sendmail.cf`, you can just insert or add (if the corresponding definitions already exist) to `sendmail.cf`

For version 8.11:

```
------------------- cut -----------------
############################
# Input mail filters
############################
O InputMailFilters=drweb-filter
############################
#      Xfilters
############################
Xdrweb-filter,  S=__ADDRESS__, F=T,
                                    T=S:5m;R:5m;E:1h
------------------- cut -------------------
```

For versions 8.12:

If you want to enable the virus checking for locally transferred messages (through calling mail or Sendmail utility), you should duplicate all changes in `submit.cf` (and `submit.mc`) too. Besides, you should add to `O PrivacyOptions` the `nobodyreturn` value.

Example:

```
------------------- cut -----------------
# privacy flags
O PrivacyOptions=goaway,noetrn,nobodyreturn
------------------- cut -----------------
```

or in `{sendmail_src}/cf/cf/feature/msp.m4`:

```
------------------- cut -----------------
define(`confPRIVACY_FLAGS'
       `goaway,noetrn,nobodyreturn,restrictqrun')dnl
------------------- cut -----------------
```

Now, let us consider changes in `sendmail.cf`:

```
-------------------- cut ------------------
############################
# Input mail filters
############################
O InputMailFilters=drweb-filter
O Milter.LogLevel=6
############################
#      Xfilters
############################
Xdrweb-filter,  S=__ADDRESS__, F=T,
                                T=C:1m;S:5m;R:5m;E:1h
------------------- cut -------------------
```

Now we can set the following flags (F=) for the case when the filter is inaccessible:

R — Reject delivery

T — Delivery temporary blocked (if both F=R and F=T are not specified, the message will be delivered without checking),

or add to `sendmail.mc` (additional options are described in `README` in the directory with the libmilter library source texts):

For Sendmail version 8.11 only:

```
------------------- cut -------------------
define(`_FFR_MILTER',1)
INPUT_MAIL_FILTER(`drweb-filter', `S=__ADDRESS__,
                            F=T, T=S:5m;R:5m;E:1h')
------------------- cut -------------------
```

and for Sendmail version 8.12:

```
------------------- cut -------------------
INPUT_MAIL_FILTER(`drweb-filter', `S=__ADDRESS__,
                      F=T, T=C:1m;S:5m;R:5m;E:1h')
define(`confMILTER_LOG_LEVEL',`6')
------------------- cut -------------------
```

Timeout value should correspond to Sendmail "`O Timeout.datablock=XX`" (by default, the value equals to one hour, XX=>1h).

After that you should recompile `sendmail.cf`

`__ADDRESS__` is an entry setting the filter connecting transport. Its format and value are identical to those used in the `MilterAddress` filter configuration file. It may be for TCP/IP sockets family: `inet:__PORT__@__HOST__`
(`__PORT__` & `__HOST__` should have definite values, e.g., `inet:3001@localhost`)
or for UNIX-DOMAIN sockets: `local:__SOCKPATH__`
(similarly `__SOCKPATH__` should specify the path accessible with the rights the filter will be run with, e.g., `local:/var/run/drweb-smf.sock`)

For details on setting up the filter see the Sendmail documentation (you can start with `{sendmaildir}/libmilter/README`).

Restart Sendmail.

The optimal way is to run the daemon and the filter (but not necessarily Sendmail), on one and the same computer; the scanning of local files will be performed without transferring a message through the sockets. For this the daemon's and the filter's rights should be sufficient to access the general directory and enough to read temporary files created by the filter.

To enable the filter to display sendmail's message ID (message identification number for Sendmail), make sure the following line is present in `sendmail.cf`:

```
------------------- cut ------------------
O Milter.macros.envfrom=i, ...
------------------- cut ------------------
```

(dots stand for other parameters - their values are not important).

To enable writing last relay IP address into the Received: header make sure the following line in `sendmail.cf` exists:

```
------------------- cut ------------------
O Milter.macros.connect=_, ...
------------------- cut ------------------
```

(dots stand for other parameters - their values are not important).

To suppress output into syslog of messages as follows:

```
------------------- cut ------------------
X-Authentication-Warning: some.domain.com: drweb
set sender to drweb-DAEMON@some.domain.com using -f
------------------- cut ------------------
```

add a drweb user (or the one who is running drweb-smf) into the trusted-users list in `submit.cf`

```
------------------- cut ------------------
####################
#   Trusted users   #
####################
Tdrweb
------------------- cut ------------------
```

or add to `submit.mc`

```
------------------- cut ------------------
define(`confTRUSTED_USERS', `drweb')"
------------------- cut ------------------
```

6.2.2.3 Configuring drweb-smf

Filter parameters are set up in the configuration file. If started with no parameters the filter is trying to find the configuration file in standard (for the filter) directories, which you can easily find out by starting the filter from the console:

```
$ drweb-smf --help
```

If the given location of the configuration file does not suit you by any reason, you can explicitly set up the configuration filename by specifying in Content Filtering not only the path to the filter, but also setting up the parameter:

```
{path/to}/drweb-smf --conf={path/to/conf/file}
```

{path/to/conf/file} should contain the filename too.

After that you should edit the configuration file and set up notification templates. When started, the filter is trying to check its operability in the given configuration by creating and checking a test file in the spool.

For the description of the filter configuration file, notification templates and the trusted users list see `conf_file.txt`, `notify.txt` and `users_list.txt` accordingly.

To get the filter version run the filter:

```
$ drweb-smf --version
```

To test correctness of the filter configuration file run the filter

```
$ drweb-smf --check_only
```

or

```
$ drweb-smf --check_only --check_user={USER}
```

The filter will run (change the level of privileges to {USER}), read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

6.2.2.4 SIGHUP processing

Correct processing of SIGHUP is possible in two cases only:

- your filter version is precompiled by the developer

- you apply the patch for Sendmail-8.12.9 and build from source texts. In all other cases the filter will be aborted by SIGHUP

Currently, the filter makes the following actions when processing the signal.

- Reloads DenyList (see the [Scanning] section of the configuration file)
- Reloads UnnotificableVirusesList and UnnotificableAddressesList files (see the [Actions] section)

6.2.3 Building Sendmail with Mail Filter API support

You need to build Sendmail with Mail Filter API support. If it is already done, proceed to p. 3, having previously copied libmilter.a and libsmutil.a for version 8.11.x (Linux), or libsm.a for version 8.12.x (libsmutil or libsm are needed in Linux, there is no need in FreeBSD in /usr/local/lib, and mfapi.h — in /usr/local/include.

6.2.3.1 Applying patches for Sendmail

Applying patches is not obligatory. There available three patches:

- listener-8.11.1 — fixes the bug in Sendmail-8.11.1 resulting in the filter core dumping on exit. The bug is not fatal and is corrected in next versions of Sendmail
- listener-8.12.0 — fixes the bug in Sendmail-8.12.0, when unix-socket and socket-file are not deleted at exit. The bug is fatal and is corrected in version 8.12.2

```
$ cp listener-8.1X.Y.patch
      /place/where/source/texts/are/libmilter
$ cd /place/where/source/texts/are/libmilter
$ patch < listener-8.1X.Y.patch
```

- Sendmail-8.12.2 — fixes the bug in Sendmail-8.12.2; the size of the transmitted message doubled if the filter was used with the demo-key. The bug is fatal and should be removed since version 8.12.3

```
$ cp sendmail-8.12.2.patch
         /place/where/sendmail/source/tree/is
$ cd /place/where/sendmail/source/tree/is
$ patch < sendmail-8.12.2.patch
```

6.2.3.2 Building from separate source texts

From now on {`sendmaildir`} stands for the directory where Sendmail has been unpacked. For example,

```
/usr/local/src/sendmail-8.xx.x):
$ cd {sendmaildir}
```

1. Specify that you want to build with the support of libmilter, for this:

   ```
   $ cd {sendmaildir}/devtools/Site/
   ```

   If the `site.config.m4` file does not exist, create it; and if it does, add the following lines to it.
   For Sendmail versions prior to 8.12.x:

   ```
   --- cut ---
   dnl Milter
   APPENDDEF(`conf_sendmail_ENVDEF', `-
   D_FFR_MILTER')
   APPENDDEF(`conf_libmilter_ENVDEF', `-
   D_FFR_MILTER')
   --- cut ---
   ```

   For Sendmail versions 8.12.0 – 8.12.8 or if your system does not support the poll system call:

   ```
   --- cut ---
   dnl Milter
   APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
   --- cut ---
   ```

   For Sendmail versions 8.12.9 and higher:

```
--- cut ---
dnl Milter
APPENDDEF(`conf_sendmail_ENVDEF', `-DMILTER')
APPENDDEF(`conf_libmilter_ENVDEF',
                                `-D_FFR_USE_POLL')
--- cut ---
```

If building Sendmail results in errors, the following lines might be added:

```
--- cut ---
APPENDDEF(`confLIBSEARCH', `db db2 bind resolve
                                          44bsd')
APPENDDEF(`confINCDIRS', `-I/usr/include/db2/
                           -I/usr/include/db1/')
APPENDDEF(`confMAPDEF', `-DNEWDB ')
--- cut ---
```

2. Building the libmilter library:

```
$ cd {sendmaildir}/libmilter
$ ./Build -c
```

Buidling and installing Sendmail:

```
$ cd {sendmaildir}
$ ./Build -c
```

Then, install Sendmail in the way it is described in `{sendmaildir}/INSTALL`.

6.2.3.3 Building Sendmail from FreeBSD distribution (/usr/src/contrib)

If you are running FreeBSD, the first thing to do is to add the following line into `make.conf` (by default, it resides in `/etc/defaults/`):

for Sendmail version prior to 8.12.x -
`SENDMAIL_CFLAGS=-D_FFR_MILTER`
for Sendmail version 8.12.x or higher-
`SENDMAIL_CFLAGS=-DMILTER`

Suppose that your directories arrangement is standard for FreeBSD 4.x. If it is not, changes in steps 1 and 2 are minimal and quite clear ($ is treated further as shell prompt):

```
$ cd /usr/src/lib/libsmutil
$ make clean all
$ cd /usr/src/lib
$ mkdir libmilter
$ cd libmilter
```

Create Makefile in this directory containing the following:

```
---------------------- cut -----------------
SENDMAIL_DIR=${.CURDIR}/../../contrib/sendmail
.PATH:  ${SENDMAIL_DIR}/libmilter
CFLAGS+=-I${SENDMAIL_DIR}/src
                            -I${SENDMAIL_DIR}/include
CFLAGS+=-DNEWDB -DNIS -DMAP_REGEX -DNOT_SENDMAIL
# User customizations to the sendmail build
                                          environment
CFLAGS+=${SENDMAIL_CFLAGS}
LIB=    milter
SRCS+=  comm.c engine.c handler.c listener.c
                main.c signal.c sm_gethost.c smfi.c
INTERNALLIB=        true
NOPIC=              true
INTERNALSTATICLIB=  true
.include <bsd.lib.mk>
---------------------- cut -----------------
```

Then:

```
$ make clean all
```

Rebuild Sendmail:

```
$ cd /usr/src/usr.sbin/sendmail
$ make clean all install
```

6.2.3.4 Building Sendmail from FreeBSD ports

If you build Sendmail (versions 8.11.5 or higher, or possibly lower) from FreeBSD ports collection, you can simply edit `/etc/make.conf` and enable the following option:

`SENDMAIL_WITH_MILTER=Yes`, and then follow a usual building of Sendmail.

6.2.4 Known problems

Description. When using Unix socket to provide communication between the filter and Sendmail, the Milter API supporting library (delivered together with Sendmail) did not remove the socket file (up to version 8.12.2).

Solution. For versions 8.12.x a `listener-8.12.0-1.patch` is available. The installation process is similar (except for the filename) to that described in p. 6.2.3.1. For versions 8.11 this file should be removed manually or from a script managing the filter. This bug is fixed in Sendmail 8.12.2

Description. When using "local" scanning and demo key, the message size value (reported to the next mail server) doubled (the message itself stayed as it was, or increased for a short "banner" length).

Solution. The problem is solved in Sendmail 8.12.3 (and higher).

Description. When using the filter in heavy loaded systems, you can see the following strings in mail logs:

```
"... Milter (drweb-filter): select(read):
                              interrupted system call"
```

Solution. The problem is solved in Sendmail 8.12.3 (and higher).

Description. When using the filter in heavy loaded systems, you can see the following strings in mail logs:

```
"... Milter (drweb-filter): select(read): timeout
                                    before data write"
"... Milter (drweb-filter): to error state"
```

Solution. Sendmail cannot establish connection with the filter for the specified timeout. In versions 8.11 it equals to 5 seconds and cannot

be changed, in versions 8.12 this timeout can be changed in the filter description (with the `C` value):

```
Xdrweb-filter,  S=__ADDRESS__, F=T,
                                T=C:1m;S:5m;R:5m;E:1h
```

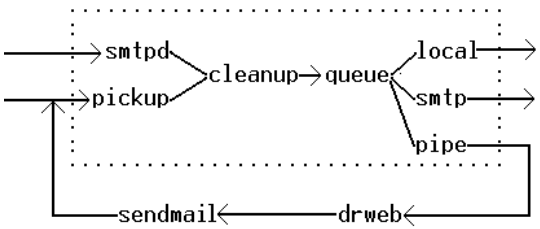
6.3 Dr.Web® Daemon and Postfix Mailer

6.3.1 Requirements

- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- Postfix versions 20000530 or higher

6.3.2 The way it works

This layout (pic. 22) shows the way the messages are filtered using the first method suggested by the Postfix mail server.



Picture22. Messages filtering scheme of Postfix mail server

The drweb-postfix filter receives an incoming message and saves it in the temporary file. Then, the client is called to inform the antivirus daemon where the object for check is located; the message itself is not transmitted through the transport connection between the filter and the daemon, and thus performance increases. After the check the filter performs the following:

- if everything goes well, the message gets queued until it is sent by Sendmail

- if viruses are detected, the message is moved to the archive and notifications are sent to the sender and administrator

6.3.3 Enabling antivirus check

To link up the Dr.Web antivirus daemon you will need to do the following.

1. Create a drweb user's budget, not allowing registration in the system. The filter will use the budget for safety purposes. It is advisable to run the daemon under the name of the same user.
2. Create a directory (e.g., `/var/spool/drweb`), accessible only for drweb user and having the read access only for the user under whose name drwebd is executed.
3. Add the following lines to the `Postfix master file` (`master.cf`), which location is specified at installation of Postfix (instead of {`drweb-directory`} insert the path to the `drweb-postfix` executable file):
   if the configuration file is on the default path:

```
filter    unix  -   n   n   -   -   pipe
                              flags=R user=drweb
            argv={drweb-directory}/drweb-postfix
                    -f ${sender} -- ${recipient}
```

   or

```
filter    unix  -   n   n   -   -   pipe
                              flags=R user=drweb
            argv={drweb-directory}/drweb-postfix
                --conf={/path/to/your/conf/file}
                    -f ${sender} -- ${recipient}
```

{`path/to/conf/file`} should contain the filename too.

You should specify `-s` option for `drweb-postfix` if you will use alternative spool directory (for small files, see also

conf_file.txt for RamSpool and RamThreshold parameters descriptions). So you should define the filter as:

```
filter    unix  -   n   n   -   -   pipe
                                  flags=R user=drweb
          argv={drweb-directory}/drweb-postfix
        -s ${size} -f ${sender} -- ${recipient}
```

4. Edit configuration file (see p. 6.3.4).
5. Instruct Postfix to check all messages circulating through SMTP. For this find in the Postfix master file the following entry (the n parameters may vary – n – –,):

   ```
   smtp    inet  n  -  n  -  -    smtpd
   ```

   and replace it with:

   ```
   smtp    inet  n  -  n  -  NN   smtpd -o
                                  content_filter= filter:dummy
   ```

   (Advice: it would be better if NN (maximum number of processes executed by postfix) equals to the MaxChildren option from the daemon's configuration files, you can also use – (minus), if do not want to enable this restriction.)
6. Restart Postfix.

Read the Postfix documentation for more details on the Postfix filter settings, for example here: http://www.postfix.org/FILTER_README.html

6.3.4 Settings

The filter parameters are set up in the configuration file. If started with no parameters, the filter is trying to find the configuration file in standard (for the filter) directories, which you can easily find out by starting the filter from the console:

```
$ drweb-postfix --help
```

For the description of the filter configuration file, notification templates and the trusted users list see `conf_file.txt`, `notify_template.txt` and `users_list.txt` accordingly.

To get the filter version run the filter:

```
$ drweb-postfix --version
```

To test the correctness of the filter configuration file run the filter:

```
$ drweb-postfix --check_only
```

or

```
$ drweb-postfix --check_only --check_user={USER}
```

The filter will run (change the level of privileges to {USER}), read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

6.4 Dr.Web® Daemon and Exim

6.4.1 Requirements

- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- Exim version 3.03 or hgher (3.15 or higher is recommended)

6.4.2 Enabling antivirus check in Exim

At present, we have discovered, implemented and tested two methods of integration of Dr.Web with the Exim mail system:

- using global system filter
- using special transportation

The first method has been implemented earlier; it can be considered more trusted, but is much slower.

6.4.2.1. Using system filter

For this, add the following into the Exim configuration file.

1. Filter settings. If it is already enabled you should only add at the beginning of the Exim configuration file (or change) some transport parameters for the filter (explained below).

```
--- cut ---
################################################
#      MESSAGE FILTER CONFIGURATION SETTINGS    #
################################################
--- cut ---
```

Add to existing or create trusted user and group for Exim:

```
--- cut ---
trusted_users = drweb
trusted_groups = drweb
--- cut ---
```

Modify (if available) or add the following options:

for Exim versions 3 (3.xx):

```
--- cut ---
message_filter = /path/to/system/filter
message_filter_pipe_transport =
                                _pipe_transport_name_
message_filter_reply_transport = address_reply
--- cut ---
```

for Exim versions 4 (4.xx):

```
--- cut ---
system_filter = /path/to/system/filter
system_filter_pipe_transport =
                                _pipe_transport_name_
system_filter_reply_transport = address_reply
--- cut ---
# Example:
# trusted_users = drweb
# trusted_groups = drweb
# message_filter = /usr/exim/system_filter.exim
# message_filter_pipe_transport = filter_pipe
# message_filter_reply_transport = address_reply
--- cut ---
```

The address_reply transport should be declared, at least as follows:

```
address_reply:
driver = autoreply
```

By default, this transport is declared in configuration of Exim, so just make sure it is available.

2. Then, in section:

```
--- cut ---
################################################
#            TRANSPORTS CONFIGURATION          #
################################################
#             ORDER DOES NOT MATTER            #
#Only one appropriate transport is called for  #
#                                  each delivery.#
################################################
--- cut ---
```

add the description of the correspondent transport:

```
--- cut ---
filter_pipe:
  driver = pipe
  user = drweb
  group = drweb
  return_fail_output
--- cut ---
```

Now, on the path specified above as (`/path/to/system/filter`) create a filter file, or, if the file already exists, just add the new filter into it:

```
--- cut ---
# Exim filter
# Version: 0.10
#
# Only run any of this stuff on the first pass
# through the filter - this is an optimisation
# for messages that get queued and have several
# delivery attempts we express this in reverse so
```

```
# we can just bail out on inappropriate messages
if $received_protocol is "drweb-scanned"
then
  # looks like a already scanned message
  finish
endif
if error_message and $header_from:
                            contains "Mailer-Daemon@"
then
  # looks like a real error message
                                      - just ignore it
  finish
endif
if not first_delivery
then
  # not first delivery attempt
  finish
endif
# Dr.Web Filter
pipe "{/PATH/TO}/drweb-exim [--conf={CONF}]
                              [-s ${message_body_size}]
                  -f $sender_address -- $recipients"
finish
--- cut ---
```

where the following symbols are used:

{CONF} — path (and filename) to the configuration file.

{/PATH/TO} — absolute path to the drweb-exim binary.

You should specify the -s option for drweb-exim, if you are going to use two directories for temporary files (for small and large messages, see also description of the RamSpool and RamThreshold parameters in conf_file.txt).

Examples:

1)

```
# Dr.Web Filter
pipe "/opt/drweb/drweb-exim –f $sender_address
                                    -- $recipients"
```

2)

```
# Dr.Web Filter
pipe "/opt/drweb/drweb-exim
       -s ${message_body_size} -f ${sender_address}
                                  -- ${recipients}"
```

3)

```
# Dr.Web Filter
pipe "/usr/local/drweb/drweb-exim
            --conf=/usr/local/drweb/drweb-exim.conf
                 -f $sender_address -- $recipients"
```

6.4.2.2. Using special transport (Exim 3.xx)

Make a drweb user "trusted":

```
--- cut ---
#################################################
#           MAIN CONFIGURATION SETTINGS         #
#################################################
trusted_users = drweb
--- cut ---
```

Then, you should add a special transport, a director and a router.

Find the section describing transports; it begins with the following header:

```
--- cut ---
################################################
#           TRANSPORTS CONFIGURATION           #
################################################
#            ORDER DOES NOT MATTER             #
#     Only one appropriate transport is called  #
#                               for each delivery.#
################################################
--- cut ---
```

here you should add the following transport description:

```
--- cut ---
# This transport is used for checking messages
                                          for viruses
drweb_transport:
  driver = pipe
  command = {/PATH/TO}/drweb-exim {CONF} -f
             ${sender_address} -- ${pipe_addresses}
  # If you want use SpamAssassin together with
                              drweb uncomment next line
  # transport_filter = /usr/bin/spamc -u drweb -s
                                                 500000
  current_directory = "/var/drweb/spool"
  # must use a privileged user to set
             $received_protocol on the way back in!
  user = drweb
  group = mail
  log_output = true
  return_fail_output = false
  # If you want use SpamAssassin uncomment next
                       line and comment out previous
  # return_fail_output = true
  return_path_add = false
--- cut ---
```

where the following symbols are used:

{CONF} – path (and filename) to the configuration file

{/PATH/TO} – absolute path to the drweb-exim binary

You should specify `-s` option for drweb-exim if you will use alternative spool directory (for small files, see also `conf_file.txt` for RamSpool and `RamThreshold parameters` description).

Now find the section describing the directors, it begins with the header:

```
--- cut ---
################################################
#               DIRECTORS CONFIGURATION        #
# Specifies how local addresses are handled    #
################################################
#                  ORDER DOES MATTER           #
#   A local address is passed to each in turn  #
#                           until it is accepted.#
################################################
--- cut ---
```

and add there the following director description:

```
--- cut ---
drweb_localuser:
  driver = localuser
  condition = "${if !eq {$received_protocol}
                            {drweb-scanned} {1}{0}}"
  transport = drweb_transport
--- cut ---
```

You should also edit the localuser director description:

```
--- cut ---
localuser:
  driver = localuser
  condition = "${if eq {$received_protocol}
                            {drweb-scanned} {1}{0}}"
  transport = local_delivery
--- cut ---
```

Now, find the section describing the routers, it begins with the header:

```
--- cut ---
################################################
#                    ROUTERS CONFIGURATION     #
# Specifies how remote addresses are   handled  #
################################################
#                  ORDER DOES MATTER           #
#  A remote address is passed to each in turn   #
#                            until it is accepted.#
################################################
--- cut ---
```

add the following router description:

```
--- cut ---
# This router routes messages to antivirus
                                    checking transport
drweb_router:
  no_verify
  driver = domainlist
  route_list = *
  condition = "${if !eq
       {$received_protocol}{drweb-scanned} {1}{0}}"
  transport = drweb_transport
--- cut ---
```

6.4.2.3. Using special transport (Exim 4.xx)

Firstly, make a drweb user "trusted":

```
--- cut ---
################################################
#             MAIN CONFIGURATION SETTINGS      #
################################################
trusted_users = drweb
--- cut ---
```

Now, add a special transport and a router. Find the section describing the routers; it begins with the header:

```
--- cut ---
#################################################
#                    ROUTERS CONFIGURATION      #
#   Specifies how remote addresses are handled  #
#################################################
#                     ORDER DOES MATTER         #
#  A remote address is passed to each in turn   #
#                            until it is accepted.#
#################################################
--- cut ---
```

add the following description of the router:

```
--- cut ---
# This router routes messages to antivirus
                                    checking transport
drweb_router:
  driver = accept
  condition = "${if eq
        {$received_protocol}{drweb-scanned}{0}{1}}"
  retry_use_local_part
  transport = drweb_transport
--- cut ---
```

Now, find the section describing transports, it begins with the header:

```
--- cut ---
#################################################
#               TRANSPORTS CONFIGURATION        #
#################################################
#                 ORDER DOES NOT MATTER         #
# Only one appropriate transport is called for  #
#                                  each delivery.#
#################################################
--- cut ---
```

add the description of the correspondent transport:

```
--- cut ---
# This transport is used for checking messages
                       for viruses drweb_transport:
  driver = pipe
  check_string =
```

```
  command = {/PATH/TO}/drweb-exim {CONF} -f
             ${sender_address} -- ${pipe_addresses}
  # If you want use SpamAssassin together with
                            drweb uncomment next line
  # transport_filter = /usr/bin/spamc -u drweb –s
                                                  500000
  current_directory = "/var/drweb/spool"
  escape_string =
  group = mail
  # headers_add = "X-Virus-Scanned: DrWEB for
                                                    Exim"
  message_prefix =
  message_suffix =
  path =
         "/bin:/sbin:/usr/bin:/usr/sbin:/opt/drweb"
  no_return_output
  no_return_path_add
  user = drweb
--- cut ---
```

where the following symbols are used:

{CONF} — path (and filename) to the configuration file.

{/PATH/TO} - absolute path to the drweb-exim binary.

You should specify the `-s` option to invoke `drweb-postfix`, if you plan to use two directories for temporary files (for small and large messages, read also `conf_file.txt` for the `RamSpool` and `RamThreshold` parameters descriptions).

6.4.3 Settings

Filter parameters are set up in the configuration file. If started with no parameters, the filter is trying to find the configuration file in standard (for the filter) directories, which you can easily find out by starting the filter from the console:

```
$ drweb-exim --help
```

For the description of the filter configuration file, notification templates and trusted users list see `conf_file.txt`, `notify.txt` and `users_list.txt` accordingly.

To get the filter version run the filter:

```
$ drweb-exim --version
```

To test correctness of the filter configuration file run the filter

```
$ drweb-exim --check_only
```

or

```
$ drweb-exim --check_only --check_user={USER}
```

The filter will run (change the level of privileges to {USER}), read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

6.4.4 Known problems

If drweb-exim, installed as system-wide filter, cannot read configuration then a message will been marked "defered", enqueued and would been resent to recipients without antivirus check.

6.5 Dr.Web® Daemon and Qmail mail system

6.5.1. Requirements

- Active Dr.Web Daemon (DrWebd) v.4.33 or higher
- QMail-1.03

6.5.2. How to configure

You should stop the qmail system before installation of the filter otherwise a losing of corresponse is possible.

6.5.2.1. Replacing the original qmail-queue

The first thing you should do is to save the original qmail-queue, because the principle of operating QMail filter is based on replacing

(or, it would be more correct to say, proxying) this component. That means that the filter receives the message through the interface specified for qmail-queue, checks it and, if it's "clean" transmits it forward to the original qmail-queue. This principle leads to the first restriction: the filter configuration file is located on the set paths (you can get to know them by running qmail-queue with the `--help` parameter). Remember the name and the path where you stored the original qmail-queue – we will further need it.

Now copy qmail-queue from the package to the `qmail/bin` directory. Then, don't forget to specify the correct rights and user name both for the new qmail-queue (drweb filter) and for the original `qmail-queue.original` you have copied. The most appropriate configuration is when the daemon and the filter work with drweb pseudo-account privileges. You should set up the following rights for files in qmail/bin for such configuartion:qmail-queue:

```
-rws--x--x  X drweb  qmail  SIZE DATE qmail-queue
-rws--x--x  X qmailq qmail  SIZE DATE qmail-
                                    queue.original
```

You can use the following commands for this:

```
$ chown drweb:qmail qmail-queue
$ chmod 4711 qmail-queue
$ chown qmailq:qmail qmail-queue.original
$ chmod 4711 qmail-queue.original
```

Also you should add drweb user to nofiles group (name of the group could be changed by a maintainer of the qmail package so please see qmail package documentaion for details):

```
$ cat /etc/group | grep nofiles
nofiles:x:NN:drweb
```

You should modify the `User` parameter in drweb32.ini to run the daemon as drweb:

```
$ cat /etc/drweb/drweb32.ini | grep User
User = drweb
```

```
;UserID =
```

Now, you should set up the rights for the spool and the infected directories:

```
$ ls -l /var | grep drweb
drwxr-xr-x  X drweb     drweb   SIZE DATE  drweb
$ ls -l /var/drweb
drwxr--r--  X drweb     drweb   SIZE DATE  bases
drwxrwx---  X drweb   nofiles   SIZE DATE  infected
drwxr--r--  X drweb     drweb   SIZE DATE  log
drwxr--r--  X drweb     drweb   SIZE DATE  run
drwxrwx---  X drweb   nofiles   SIZE DATE  spool
```

6.5.2.2. Setting up the filter

The filter parameters are set up in the configuration file. As it is impossible to set the filter parameters at start, it tries to find the configuration file in standard (for the filter) directories, which you can easily find out by running the filter from the console.

```
$ qmail-queue --help
```

For the description of the filter configuration file, notification templates and trusted users list read `conf_file.txt`, `notify.txt` and `users_list.txt` accordingly.

To get the filter version run the filter:

```
$ qmail-queue --version
```

To test correctness of the filter configuration file run the filter

```
$ qmail-queue --check_only
```

The filter will run, read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

6.6 Dr.Web® Daemon and Zmailer mail system

6.6.1 Requirements

- Active Dr.Web Daemon (drwebd) version 4.33 or higher

- ZMailer with ContentFiltering support

6.6.2 Enabling Dr.Web® support in ZMailer

For correct operation of ZMailer and other filters it is advisable to apply patches, if the source texts of Zmailer are available and you wish to rebuild it. To patch you should:

- Go to the `$(ZMAILER_SRCHOME)/smtpserver` directory
- To make sure the patch is appropriate for your version you should do the following:
  `$ patch -C < smtpdata.c.XXX.patch`
  (where XXX — is the version of Zmailer, for which the patch was designed)
- to put the patch do the following:
  `$ patch < smtpdata.c.XXX.patch`

If you want to apply fast method of rejecting messages from <> (errors, bounces and widely used by spammers), you can apply the `policytest.c.XXX.patch` in the same way

To enable the Dr.Web Daemon support in Zmailer, you should:

- copy drweb-zmailer into the `$MAILBIN` directory (this value is specified in `zmailer.conf`)
- and edit `smtpserver.conf`, by adding or modifying it:
  `PARAM contentfilter $MAILBIN/drweb-zmailer`

Standard ZMailer filtering API allows to check only messages received through SMTP-protocol. To check locally transmitted messages read the example in the `local_scan.example` subdirectory.

6.6.3 Setting up the drweb-zmailer filter

The filter parameters are set up in the configuration file. As it is impossible to set up the filter parameters at start, it will try to find

the configuration in the standard (for the filter) directories, which you can find out by, running the filter from the console:

```
$ drweb-zmailer --help
```

For the description of the filter configuration file and notification templates see `conf_file.txt` and `notify.txt` accordingly.

To get the filter version run it with the following parameter:

```
$ drweb-zmailer --version
```

To test correctness of the filter configuration file run the filter

```
$ drweb-zmailer --check_only
```

or

```
$ drweb-zmailer --check_only --check_user={USER}
```

The filter will run (change the level of privileges to {USER}), read the configuration file and test options. It will report on actions made into a standard output stream (stdout).

The Dr.Web daemon (drwebd) should be run under the root account, or the mode with `LocalScan=no` should be used, as ZMailer creates files in queue accessible for root only.

6.7 Dr.Web® Daemon Courier-MTA mail system

6.7.1 Requirements

- Courier-MTA
- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- drweb-courier plug-in

6.7.2 Enabling drweb-courier support in Courier-MTA

- Copy drweb-coureir or create a symbolic link into Courier filter directory (usually `/usr/lib/courier/libexec/filters`)
- Register the filter as global in the Courier system:

  ```
  /usr/lib/courier/sbin/filterctl start
                                    drweb-courier
  ```

  if you want to stop filtering with the drweb-courier:

  ```
  /usr/lib/courier/sbin/filterctl stop
                                    drweb-courier
  ```

- Create (edit) the `enablefiltering` control file to set services for check (esmtp, local or uucp — if several services are specified they should be separated with blanks)
- Enable filtration in Courier:
  `/usr/lib/courier/sbin/courierfilter start`

6.7.3 Setting up the drweb-courier plug-in

The plug-in parameters are set up in the configuration file. If started with no parameters, the plug-in will try to find the configuration file in standard (for the plug-in) directories, which you can easily find out by starting the plug-in from the console:

```
$ drweb-courier --help
```

The descriptions of the plug-in configuration file and the notification templates are stored in `conf_file.txt and notify.txt` accordingly.

To get the plug-in version run it with the following parameter:

```
$ drweb-courier --version
```

To test correctness of the plug-in configuration file run the plug-in

```
$ drweb-courier --check_only
```

or

```
$ drweb-courier --check_only --check_user={USER}
```

The plug-in will starts (drop privileges to {USER} if `--check_user=` specified), read the configuration file and test options. The plug-in will report on the actions made into a standard output stream (stdout).

6.8 Dr.Web® Daemon and Mobico MIO Server

6.8.1 Requirements

- Mobico MIO Server (version 3.24 or higher)
- Started Dr.Web Daemon (drwebd) version 4.30 or higher

The plug-in is incompatible with "per email" licenses for the daemon, it can be used only with "traffic" or "unlimited" licenses.

The filter will check only incoming messages for mail boxes on this server. Relayed mail will ***not be checked***. This is the restriction of the Mobico MIO server antivirus API.

6.8.2 Enabling drweb-mio support in Mobico MIO Server

Add the following line into the Service mqueue section of the MIO server configuration file (`mio.conf`):

```
ext_content_filter={/path/to/}drweb-mio
```

e.g.,

```
ext_content_filter=/opt/drweb/drweb-mio
```

For more details on external filter settings in MIO read the "MIO Administrator manual".

6.8.3 Setting up the drweb-mio plug-in

The plug-in parameters are set up in the configuration file. If started with no parameters, the plug-in will try to find the configuration file in standard (for the plug-in) directories, which you can easily find out by starting the plug-in from the console:

```
$ drweb-mio --help
```

If, due to some reasons, you find the given paths for the location of the configuration file inappropriate, specify explicitly the name of the configuration file. Specify in the Content Filtering not only the path to the plug-in, but also set up the following parameter:

```
{path/to}/drweb-mio --conf={path/to/conf/file}
```

{`path/to/conf/file`} should contain the filename too.

The descriptions of the plug-in configuration file and the notification templates are stored in `conf_file.txt` and notify.txt accordingly.

To get the plug-in version run the plug-in with the following parameter:

```
$ drweb-mio --version
```

To test the correctness of the plug-in configuration file run the plug-in

```
$ drweb-mio --check_only
```

The plug-in will start, read configuration file and test options. Plug-in will report into standard output stream (stdout) about actions.

DrWeb daemon (drwebd) should work under root account or `LocalScan = no` mode should be used.

6.9 Description of the unnotificable viruses list

6.9.1 Purpose

This file (`viruses.conf`) allows to block notifications to specified person (a sender, recipients or an administrator) depending on the virus name.

The lines starting with the # character are considered commentaries and therefore get skipped, blank lines are also allowable. If the first meaningful line contains a similar entry:
`[version=NN]`

this means that the entries are in the NN-version of the file. If there is no such a line, the file is considered to be in the first version format.

6.9.2 Version 1 format

```
TO_ADMIN TO_SENDER TO_RCPTS VIRUSNAME
```

TO`_xxx` — {`allow` | `deny`} (i.e. either `allow` or `deny` `value`) — either allow or deny notifications to a specified person.

VIRUSNAME — virus name. All virus names should be written in terms of POSIX regular expressions.

E.g., `HLLM.Generic.95` should be written as "`HLLM\.Generic\.95`". It is strongly recommended to enclose all names in quotes.

6.9.3 Version 2 format

```
TO_ADMIN TO_SENDER TO_RCPTS QUARANTINE VIRUSNAME
```

Description of `TO_xxx and VIRUSNAME` is similar to those from the previous paragraph.

QUARANTINE — {`allow` | `deny`} allows or denies moving a message to the quarantine. It is highly useful during the mass epidemics, especially with viruses forging addresses.

6.9.4 Cure, remove and redirect actions peculiarities (for all filters except for drweb-mio)

- If `cure` or `remove` actions are specified, a virus is removed (for some types of viruses, i.e. trojans, curing is equal to removing) and notification to recipients is blocked, then the `discard` action (or `reject`, if `discard` is not supported by MTA) is applied.
- If action is set to `redirect`, and the quarantine function is blocked for this virus, then redirection is not performed and the `reject` action is applied.

6.10 Description of the file with the "blocked" masks

6.10.1 Purpose

This file (`users.conf`) allows to specify the addresses not subject to antivirus check. The first variant of the file is already out-of-date, but is supported by all the filters. The new file format allows to extend possibilities by specifying the file version (do not mix up with the product version).

The lines starting with the # character are considered commentaries and therefore get skipped, blank lines are also allowable. If the first meaningful line contains a similar entry: [`version=NN`],

this means that the entries are in the NN-version of the file. If there is no such a line, the file is considered to be in the first version format.

When checking a message, its addresses (that of a sender and an addressee) get compared to each line of the file (the way they follow in the file). If the correspondence to the address in some line is found the search terminates. If the correspondence to the address is not found in any line the message check is permitted for the given address in any cases.

6.10.2 Version 2 format

```
OPERATION    WHO    METHOD    MASK
```

OPERATION — {`allow` | `deny`} (either `allow`, or `deny` `value`).

The `allow` value means, that the checking of the addresses corresponding to this entry is allowed, the `deny` value means that the checking of this address if prohibited.

WHO — {`from` | `to` | `any`}. The `from` value means, that the entry is considered only if the address is the sender's one; `to` means the entry is considered only if it will be the recipient's address, WHO means the entry is considered in both cases.

METHOD — {`exact` | `subst` | `regex` | `cregex`} determines the method of correspondence of the address to the mask. The `exact` value means that the address should precisely correspond to the MASK. The `subst` value means, that it is sufficient that the MASK is a substring in the address for the address to correspond with it.

The `regex` or `cregex` values mean that the address should correspond to a regular expression written in the mask.

The `exact`, `subst` and `cregex` methods are case-sensitive, the `regex` method is case-insensitive.

MASK is the line, which contains no blanks; otherwise it should be quoted to contain blanks.

6.10.3 Version 1 format (outdated)

```
OPERATION     MASK
```

The fields descriptions correspond to the specified above. Any entry of the first version can be written as the entry of the second version as follows:

```
OPERATION    any     subst   MASK
```

6.10.4 Check denying algorithm in pseudocode

```
IF DenyMode EQUAL "byAll" THEN
{
   IF sender_is_uncheckable AND
all_rcpts_are_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
ELSE IF DenyMode EQUAL "byOne" THEN
{
   IF sender_is_uncheckable OR
one_of_rcpts_is_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
ELSE IF DenyMode EQUAL "bySender" THEN
{
   IF sender_is_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
ELSE IF DenyMode EQUAL "bySenderAndOneRecipient"
THEN
{
   IF sender_is_uncheckable AND
one_of_rcpts_is_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
```

```
ELSE IF DenyMode EQUAL "byOneRecipient" THEN
{
   IF one_of_rcpts_is_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
ELSE IF DenyMode EQUAL "byAllRecipients" THEN
{
   IF all_rcpts_are_uncheckable
   THEN pass_message_without_check
   ELSE check_message
}
```

* `DenyMode` is the option from the [`Scanning`] section of the `drweb_{mta}.conf file`, where {mta} is one of: `smf`, `cgp`, `postfix`, `exim`, `qmail`, `zmailer` or `courier`.

* `sender_is_uncheckable true`, if a sender`s address corresponds to one of `any`— or `from`— entry with the `deny` action.

* `recipient_is_uncheckable true`, if a recipient`s address coresponds to one of `any`— or `to`— entry with `deny` action.

6.10.5 Examples

1) Deny checking of incoming messages for all users except for asv@ .. .ru:

   ```
   deny    to      regex   ^asv@(.*)\.ru$
   ```

2) Deny checking of out-going messages of users of the drweb.com domain:

   ```
   deny   from    regex  @(.*)\.drweb\.com$
   ```

6.10.6 Frequently asked questions

Q: What "allow" is needed for if all the addresses are permitted by default?

A: To create exceptions from general rules.

Example:

```
allow   any   exact  someuser@any.domain.com
deny    any   subst     @any.domain.com
```

The example shows that the mail for someuser of the any.domain.com domain will be checked, while the mail of the rest of users will not be checked.

6.11 Virus statistics. Dr.Web® Statistics agent

6.11.1 Requirements

- One of Dr.Web mail filters version 4.32 or higher

6.11.2 What does the agent do?

The drweb-agent is designed to collect statistics from the Dr.Web mail filters and send it from time to time to our statistics server http://stat.drweb.com. We believe, that it might be useful both for our company (Doctor Web, Ltd.) and you (users). We receive statistics for future processing, and you receive the tool for collection, storage and display of the virus statistics. You can view both the virus statistics on your server and consolidated statistics (see p. 6.11.4 for details). The agent collects the following information:

- Total quantity of the viruses detected
- The list of pairs (virus, quantity) for all virus names
- Quantity of scanned messages (at present, this parameter is not sent to http://stat.drweb.com.)
- To use the agent you should be either our registered user (in this case the UUID for you will be md5 of the key file), or obtain a personal UUID. This UUID has to be written in the drweb_agent.conf configuration file

6.11.3 Configuring the agent

You can find the description of the agent's configuration file `drweb_agent.conf` in `conf_file.txt`

In the mail filter configuration file you should add the [`Agent`] section; the description of the parameters of this section is in the `conf_file.txt` resided in the directory with the name of the mail system.

To get the agent automatically loaded at a system startup, find the startup scripts for the correspondent systems in the linux and bsd directories. In addition, do not forget to start the agent before the mail filter's startup.

If the agent receives the instruction to reboot (through initscript or -HUP), the filters for CommuniGate Pro (drweb-cgp), Sendmail (drweb-smf), Courier-MTA (drweb-courier) and Mobico MIO Server (drweb-mio) should also be rebooted.

6.11.4 Administrating the statistics

During the operation of the antivirus on the Unix mail servers the statistics on virus events can be collected. The received information is sent to the Dr.Web statistics server. You can view there both the statistics of your server and the total statistics collected from all servers.

The statistics contains the data on most frequently detected viruses (quantity of detections and the per cent of the total amount) during a certain term.

The data can be represented either in HTML format or in XML. The last option is a very convenient way if you wish to publish the data received from your web-site: you can transform the data in accordance with your site design and your way of presentation on it.

To receive a summarized statistics from all servers, open http://stat.drweb.com in the web browser. The page contains the list of viruses detected on the servers (in a detection frequency descending order) stating for each of them the quantity of detections in figures and per cents (pic. 23, represents a fragment of the table). The page may differ in viewing depending upon the browser.

12.04.2005 00:00 - 12.04.2005 08:00		
viruses: 157, infected messages: 213424		
Win32.HLLM.Netsky.35328	52830	24.75%
Win32.HLLM.Netsky	28492	13.35%
Win32.HLLM.MyDoom.22	27897	13.07%
Win32.HLLM.Netsky.based	19960	9.35%
Win32.HLLM.Netsky.22016	9209	4.31%
Win32.HLLM.MyDoom.33808	6796	3.18%
Win32.HLLM.MyDoom.21	6208	2.91%
Trojan.Bankfraud	6036	2.83%
Win32.HLLM.MyDoom.54464	5839	2.74%
Win32.HLLM.Lovgate.9	5014	2.35%
Win32.Parite.2	4410	2.07%

Picture 23. Virus statistics

You can change the enquiry parameters and repeat it:

1) In the dropdown list choose the time zone (by default, it is GMT). Check the `is daylight savings time in effect?` box, if summer-time settings are used.
2) In the `start` and `end` drop down lists, set the time and date of the beginning and end of the term you need to view a statistics.
3) Specify the maximum size of the table in the `max` entry field (only most frequently detected viruses will be viewed).

4) Check the `daily` box, if you want to view the statistics in separate tables for separate days (if an enquiry is made for several days).

5) Press [ ].

The file with summarized statistics in XML can be found here http://info.drweb.com/export/xml/top. The example of this file is sited below:

```
- <drwebvirustop period="24" top="5"
updatedutc="2005-04-12 09:02:01">
- <item>
  <vname>Win32.HLLM.Netsky</vname>
  <place>1</place>
  <percents>58.3163167883155</percents>
  </item>
- <item>
  <vname>Win32.HLLM.MyDoom</vname>
  <place>2</place>
  <percents>30.0401772318628</percents>
  </item>
- <item>
  <vname>Trojan.Bankfraud</vname>
  <place>3</place>
  <percents>3.19479137845705</percents>
  </item>
- <item>
  <vname>Win32.HLLM.Dasha</vname>
  <place>4</place>
  <percents>2.36114304419496</percents>
  </item>
- <item>
  <vname>Win32.HLLM.Hazafi</vname>
  <place>5</place>
  <percents>2.28885520219163</percents>
  </item>
  </drwebvirustop>
```

The following attributes are used:

- `period` – the duration (in hours) of the statistics' collection

- `top` – the quantity of the most frequently detected viruses presented in the table
- `updatedutc` – the time of the statistics' last update
- `vname` – name of a virus
- `place` – place in the statistics
- `percents` – per cent of the total quantity of detections

The "the duration (in hours) of the statistics' collection" and the table size parameters cannot be modified by a user.

To get a statistics for your server, open http://stat.drweb.com/view/<UID>, where `<UID>` is a user identifier (md5-key). To get the md5-key, contact our support service.

The personal statistics page has the same format as the summarized statistics page.

A personalized statistics in XML can be found at http://stat.drweb.com/xml/<UID>, where `<UID>` is a user identifier (md5-key) (see above). The brief example of such file is sited below:

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <drwebvirustop period="24" top="20" user="<UID>"
                lastdata="2005-04-12 07:00:00+04">
- <item>
  <caught>69</caught>
  <percents>24.1258741258741</percents>
  <place>1</place>
  <vname>Win32.HLLM.Netsky.35328</vname>
  </item>
- <item>
  <caught>57</caught>
  <percents>19.9300699300699</percents>
  <place>2</place>
  <vname>Win32.HLLM.MyDoom.54464</vname>
  </item>
.........................................
  </drwebvirustop>
```

The following attributes are used:

- `period` – the duration (in hours) of the statistics' collection
- `top` – most frequently caught viruses presented in the table
- `user` – user identifier
- `lastdata` – time of the last data receipt from a user
- `vname` – name of a virus
- `place` – place in the statistics
- `caught` – the quantity of detections of a given virus
- `percents` – per cent of the total quantity of detections

The same as with the summarized statistics, a user cannot set specify the duration of the statistics' collection and the size of the table.

7 Integration of Dr.Web® Daemon with file systems

7.1. Dr.Web® Daemon and Samba file server

7.1.1. Requirements

- Active drwebd version 4.30 (or higher)
- Samba-2.2.1 (or higher)
- For Samba versions 2.2.1, 2.2.2 and 2.2.3a source texts of Samba are needed, as vfs-interface in these versions was experimental and was not included into a standard build

The daemon (drwebd) and Samba SpIDer should be run at the same host or the daemon should have access (not through a smbfs, i.e. through a nfs) to files at protected shares.

7.1.2. What is this

Samba SpIDer is a monitor of file operations for the Samba file server. It is designed as a plug-in module for new vfs-interface (virtual file system) in Samba. At the same time, it is a client to the Dr.Web Daemon, and as such requires the configured and active daemon.

7.1.3. Installation

7.1.3.1. Installing Samba SpIDer

Make the following modifications in the [`Daemon`] section of the antivirus configuration file.

If you are going to use the `quarantine` function (moving of infected files), add the following (if it is missing):
`MoveFilesTo = {path/to/your/favorite/quarantine/dir}`

If you are going to use the `rename` function (renaming of infected files), you should add the following (if it is missing):
`RenameFilesTo = #??` (i.e. the files will be renamed from `file.ext` to `file.#xt`, or from file to `file.#`).

Add the section listed below into the samba configuration file and edit it accordingly to your paths. Now the resource is protected by Dr.Web:

For Samba versions 2.2.x:

```
--- cut ---
[drweb_audit]
comment = Dr.Web protected directory
path = /DIR/TO/PROTECT
vfs object = /FULL/PATH/TO/smb_spider.so
writeable = yes
browseable = yes
guest ok = yes
public = yes
--- cut ---
```

For Samba version 3.0.x:

```
--- cut ---
[drweb_audit]
comment = Dr.Web protected directory
path = /DIR/TO/PROTECT
vfs objects = smb_spider
writeable = yes
browseable = yes
guest ok = yes
public = yes
--- cut ---
```

Correspondence of `smb_spider` and Samba versions:

`smb_spider.so.1` — Samba 2.2.1, 2.2.2
`smb_spider.so.2` — Samba 2.2.3

smb_spider.so.3 — Samba 2.2.4, 2.2.5
smb_spider.so.4 — Samba 2.2.6 – 2.2.12
smb_spider.so.9 — Samba 3.0.0, 3.0.1
smb_spider.so.10 — Samba 3.0.2
smb_spider.so.11 — Samba 3.0.3, 3.0.10
smb_spider.so.11 — Samba 3.0.12

For Samba servers compiled with LFS support (e.g., in linux with kernel 2.4 and glibc 2.2) you should select the file with suffix – lfs.

- for Samba 2.2.x:

copy the appropriate smb_spider to any directory (advisably to the daemon directory), which is specified as /FULL/PATH/TO/ in smb.conf

- for Samba 3.0.x:

copy the correspondent smb_spider.so.NN to the lib/vfs subdirectory of Samba directory tree and rename to smb_spider.so.

Copy smb_spider.conf to the /usr/local/etc/drweb directory (alas, still there is no way to transmit even a path to the ini-file to SpIDer) and edit this file (the description of the configuration file is in conf_file.txt).

Restart Samba server.

The monitor will activate when one of the clients tries to open the shared resource on the server. The following actions are performed during the initialisation:

1) the versions of the monitor and the Samba-server interfaces are checked;
2) the monitor reads the configuration file (/etc/drweb/smb_spider.conf);
3) then, it monitors file operations made by the client. At steps 1 and 2 the monitor outputs information into the system log

(syslog) with the following parameters `Facility.Priority = Daemon.Info`

(see `syslogd.conf`). After the configuration file is loaded the output of the information is made with the parameters `Facility.Priority = SyslogFacility.SyslogPriority`

7.1.3.2. Building Samba with VFS support

Samba SpIDer uses the vfs-interface (virtual file system) of the samba server. As the interface was pilot in Samba 2.2.1, 2.2.2 and 2.2.3a, the corresponding binary packages with Samba do not have a build-in support. So the source texts for building this interface with vfs-interface support are required. And traditionally some corrections should be made too.

Firstly, you should patch Makefile.in:

`samba-2.2.1a-vfs.patch` for Samba 2.2.1
`samba-2.2.2-vfs.patch` for Samba 2.2.2
`samba-2.2.3a-vfs.patch` for Samba 2.2.3

The patch (`samba-2.2.X-vfs.patch`) is applied as follows:

```
# cd {to-dir-where-samba-dir-extracted}
# Patch -p0 < samba-2.2.X-vfs.patch
```

Example:

You have put `samba-2.2.2.tgz` to the `/home/tmp` directory, then you should:

```
# cd /home/tmp
# tar xzf samba-2.2.2.tgz
# patch -p0 < samba-2.2.2-vfs.patch
```

For Samba versions 2.2.1 — 2.2.3 only: FreeBSD has no `libdl.so` library (as it is not necessary, the necessary resides in `libc.so`), it seems that Samba authors do not known or forgot about it, to build correctly in FreeBSD you should:

- download `libdl.so.1`, for example, from here: http://www.mit.edu/afs/sipb/system/i386_nbsd1/emul/linux/lib/libdl.so.1
- put it to `/usr/lib/` and obligatory make a symlink to it `/usr/lib/libdl.so`
- delete `config.cache` (if it exists) in the `samba-2.2.x` directory
- delete `config.cache` (if it exists) in the `samba-2.2.x` directory

Now you should build samba with vfs-interface support:

```
# configure --with-vfs
# make
# make install
```

and then configure as usual.

7.1.4. Known problems

When started, the daemon should have read access (if we are going to search for viruses only) and write access (deletion, curing, etc.) to all files written and read on the shared resource.

8 Integration of the Dr.Web® with applications using icap

8.1. Dr.Web® Daemon and Squid proxy server

drweb-icapd connects a proxy server supporting icap and the Dr.Web daemon to scan incoming http-traffic for viruses. It also allows to filter access to html resources by mime-type/size and by the host name. Thus, drweb-icapd is a solution for protection of internet gateways.

At present the support of icap is included into Squid, SafeSquid and Shweby proxy servers and is planned in Opps! proxy server.

8.1.1. Requirements

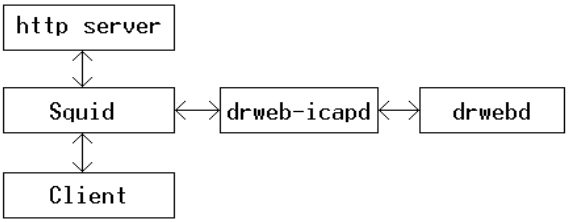
- Active Dr.Web Daemon (drwebd) version 4.33 or higher
- Active:
  1) either Squid version 2.5.STABLE5 or higher with icap version 6-pre3 or heigher support. If the preview mode is enabled, use the source code version of squid-icap-2.5-200409161544-src with patch patch.diff from the documentation directory, or download the patched version from www.drweb.com (recommended option). For details read p. 8.1.3;
  2) or Shweby version 1.0 or higher;
  3) or SafeSquid version 3.0 or higher

8.1.2. Providing for compatibility of proxy server and drweb-icapd

Suppose that Squid with the icap support is already assembled and started. If not, read paragraph 8.1.3 on how to compile Squid with the icap support.

8.1.2.1. The way it works

Below is the general scheme of interaction of Squid, drweb-icapd and a client. (pic. 24).



Picture 24. General scheme of interaction of Squid, drweb-icapd and the client

The scheme shows that the client interacts with an http-server through a proxy server. The proxy server is, at the same time, an icap-client of the icap-server (drweb-icap). Drweb-icap, in its turn, is a client of Dr.Web daemon. Drweb-icap allows to scan for viruses (by drwebd daemon) the http-traffic incoming from an http-server and passed to the proxy server on icap. This scheme does not allow scanning of ftp-traffic. Read paragraph 8.1.4 to enable scanning ftp-traffic with Squid.

8.1.2.2. Settings of Squid for interaction with drweb-icapd

To provide for interaction between Squid and drweb-icapd you should edit the file squid.conf (it usually resides in `/usr/local/squid/etc`) in order to enable icap functions.

You should either find the lines listed below, to uncomment them and edit (if necessary) the default values, or add the given settings at the end of the file:

```
# enable icap
icap_enable on

# register new service icap
```

```
# ICAP service description:
# icap_service <name> <type> <pass> <url>
#     <name> - name of service
#     <type> - type of service
#     <pass> - can content be passed (1) beside of
                                ICAP server or not (0)
#     <ur> - url of the service
icap_service service_1 respmod_precache 0
                        icap://localhost:1344/respmod
```

when using the Squid-STABLE10 version, you should write the following instead of the previous line:

```
icap_service service_1 respmod_precache 0
         icap://localhost:1344/respmod no-keep-alive
# make a class for the new service
icap_class class_1 service_1

# allow access of the new class to HTTP, GET etc.
icap_access class_1 allow all
```

When using the preview mode with patched version of Squid (see p. 8.1.3) you should enable additional settings:

```
# enable preview mode
icap_preview_enable on
# specify the size of a message (in bytes)
                                        sent to preview
icap_preview_size 0
```

At present in Squid the respmod-postcache mode is not implemented, therefore when using this proxy server the check of content in cash is impossible!

8.1.2.3. Setting Shweby for interaction with drweb-icapd

To provide for interaction between Shweby and drweb-icapd you should edit the `config.xml` file. Create a new service for sending RESPMOD requests to the icap-server at the beginning of the <`icap`> section:

```
<service>
   <name>drweb_resp</name>
   <comment>drweb_icapd service,
                     for HTTP responses</comment>
   <vpoint>respmod_postcache</vpoint>
   <uri>icap://localhost:1344/</uri>
   <keepalive>false</keepalive>
   <shortcut>false</shortcut>
</service>
```

Make sure `false` is specified in the <`keepalive`> section. Then, create a class to use this service:

```
<class>
   <name>drweb</name>
    <comment>Services for drweb_icapd</comment>
   <service_list>drweb_resp</service_list>
</class>
```

Allow using the new class for specified ip-addresses and the port in the <`access`> section, for example:

```
<allow>
   <enabled>true</enabled>
    <comment>localhost, drweb class</comment>
   <username></username>
   <password></password>
   <ip_list>127.0.0.1-
              127.0.0.254,81.57.150.39</ip_list>
   <port_list>4000</port_list>
   <class>drweb</class>
   <access>config,proxy,connect,http,
                              transparent</access>
</allow>
```

Make sure the example services from the original `config.xml` (for example, `test_resp`) are disabled in the <`access`> section.

8.1.2.4. Setting interaction of SafeSquid and drweb-icapd

To provide for interaction between SafeSquid and drweb-icapd you should edit `config.xml`, or use the web-interface.

If using the web-interface, you should choose the `ICAP` section from the drop-down menu, then select the `Add` item (add new icap interface). In the opened form, fill in the following fields:
`Enabled=true`; `Host=ip` or host name where drweb-icapd is run (by default, it is `localhost`);
`File=/respmod`; `Port`=port number where drweb-icapd listens (by default, it is 1344);
`Applies to=responses`; and then Submit.

You can also edit `config.xml` yourself: for this, for example, you should add

```
<icap>
   <enabled>true</enabled>
   <icap>
      <enabled>true</enabled>
      <comment>Dr.Web icap server</comment>
      <profiles></profiles>
      <host>localhost</host>
      <file>/respmod</file>
      <port>1344</port>
      <which>responses</which>
   </icap>
</icap>
```

into `<safesquidb </safesquid>` section

8.1.2.5. Launching order

The launching of the interacting elements should be as follows:

1) Dr.Web daemon;
2) drweb-icapd;
3) proxy-server.

Regardless the order of launching the elements, not a single object will pass unchecked, as either proxy-server blocks the transmission, if there is no connection with the drweb-icapd, or the drweb-icapd itself will block the receipt of information, if there is no connection with the Dr.Web daemon (a user will receive a correspondent page with notification in return).

8.1.3. Assembling Squid with icap support

There are two ways to obtain the patched version of Squid.

1. Patch yourself source codes of Squid. For this:
   - download the source codes of Squid version STABLE10 or higher, for example, from here: http://www.squid-cache.org/Versions/v2/2.5/
   - download patch with icap support, for example, from here: http://squid.sourceforge.net/projects.html#icap
   - copy the patch to the directory with the Squid source codes
     `cp patch_with_icap_support /path/to/directory/squid/`
   - go to the directory with the Squid source codes:
     `cd /path/to/directory/squid/`
   - execute the command
     `patch -p1 < patch_with_icap_support`
   - then compile Squid as it is described below.
2. Download the patched version of Squid from http://www.drweb.com, unpack the archive and compile as described above.

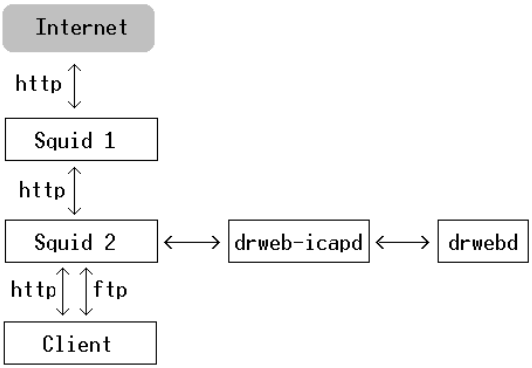
After the Squid source code with icap suport is received (if, due to your own reasons, the unpatched version of Squid should be used, read the end of the text for the addresses for downloads) you should execute the script `bootstrap.sh` in the Squid root directory and then the script `./configure` with the following parameters:

`--enable-icap-support --prefix=/usr/local/squid`
the last parameter sets the directory where Squid will be installed.

Then follow the instructions described in the `INSTALL` file in the `Squid` directory.

8.1.4. Transmitting ftp-traffic through drweb-icapd with Squid

To scan both http- and ftp-traffic you should install and set up two Squids at a time. Below goes an interaction scheme for such scanning (pic. 25).



Picture25. Enabling scanning of http- and ftp-traffic

Following the scheme both http- and ftp-traffic from the internet will be scanned by drweb-icapd, as in both cases for Squid 2 this traffic is an http-traffic. To implement the scheme you should do the following:

1. install 2 independent Squids in different folders;
2. in Squid 1 change the http_port parameter. Set the new port parameter (e.g., 3129);
3. enable interaction of Squid 2 with drweb-icapd, as it is described in p. 8.1.2.2;

4. change settings of Squid 2 for interaction with Squid 1 by setting the following parameter:

```
cache_peer localhost  parent  3129  3130
                default connect-timeout=80000
```

   3129 means here port number installed above, localhost means a host Squid 1 is installed, and 80000 is a time-out value which should be big enough to support connection between Squid 2 and Squid 1.

5. enable interaction of clients through proxy Squid 2 both on http- and ftp-traffic;
6. run both Squid and drweb-icapd.

8.1.5. The preview mode

The preview mode allows to specify files that should not be scanned and therefore should not be loaded by the icap-server (e.g., streaming video and audio). It also allows to considerably decrease both external traffic when using the option of filtering access by mime-type/size, or by a host name, and internal traffic when using `Allow 204` and `preview_size=0` modes. It speeds up the operation and renders it more comfortable for the end user.

When using the unpatched version of Squid (read p. 8.1.3) we strongly advise to disable the preview mode both in drweb-icapd (`UsePreview=No`) and in Squid (`icap_preview_enable` off and `icap_preview_size -1`).

At present the preview mode is not done in Shweby proxy server. In SafeSquid proxy server, instead of preview mode, the output of total number of downloaded bytes of data is implemented.

8.1.6. Testing the drweb-icapd availability

To test the drweb-icapd availability you should do the following:

1. make sure the `Infected`, `Suspicious` and `Incurable` parameters in the `drweb-icapd.ini` file are set with the `report` value
2. visit http://www.eicar.org/download/eicar.com, a message warning on an infected file will be displayed in the browser

If no warning message is displayed, check the following:

1. the browser uses proxy Squid or Shweby set for interaction with drweb-icapd to provide access for http-traffic
2. Check that the templates are copied to the `%etc_dir/templates/icapd` directory and the paths to them in the drweb-icapd.ini file are specified correctly

8.1.7. Web sites of Squid, Shweby and Oops! projects

Squid project cache: http://www.squid-cache.org/

Support of icap for Squid:
http://squid.sourceforge.net/projects.html#icap ,
http://squid.sourceforge.net/icap/

Shweby Project: http://shweby.sourceforge.net

SafeSquid Project: http://www.safesquid.com

Oops! Project cache: http://zipper.paco.net/~igor/oops.eng/

9 CONTACTS

The Dr.Web antivirus program is in constant development. The latest news on its updates and informative notices are available on the web site:

http://www.drweb.com

Sales department:

http://buy.drweb.com
e-mail: sales@drweb.com

WWW: http://buy.drweb.com

e-mail: sales@drweb.com

Technical support service:

http://support.drweb.com

e-mail: support@drweb.com

When addressing our technical support the following information, which can help to thoroughly examine the case, will be greatly appreciated:

- full name and version of the UNIX distribution
- the Dr.Web program version
- versions of applications and filters the Dr.Web Daemon is integrated with
- configuration files of the daemon and the applications the Dr.Web Daemon is integrated with
- log files: daemon, filters and other applications the Dr.Web Daemon s integrated with

APPENDIX.
Dr.Web® for Unix-like systems User Licenses

There exist four main types of licenses for Unix-based versions of Dr.Web, one for the scanner and three for the daemon.

The licenses can be purchased for definite terms, i.e. for 6, 12 or 24 months. The license terms and the quantitative parameters and limitations for different regional partners of Doctor Web, Ltd. may vary or be revised. To learn more about regional license terms, contact our partner in your region listed on the web-site of Doctor Web, Ltd. (http://partners.drweb.com/list/)

When buying a license, during the whole license term, you have the right to update from the Dr.Web Global Updating system servers and to receive a standard technical support of Doctor Web, Ltd. and its partners.

1. Command line scanners (http://buy.drweb.com/home/)

This license allows operation with Dr.Web antivirus scanner for Unix (both with graphical interface module and with the console scanner), as well as with other types of command line scanners for Windows, MS DOS, OS/2.

2. Corporate mail protection (http://buy.drweb.com/business/)

This is a license for Dr.Web daemon which enables scanning of mail messages passed to a daemon by filters from Communigate Pro, Sendmail, Postfix, Exim, QMail, ZMailer, Courier-MTA, MIO Mail Server mail servers.

This license is not for integration of the daemon with file servers and Internet gateways, but enables their complex usage.

Below are described the licenses for mail servers.

2.1. Licensing per traffic

This license allows to use the daemon for scanning messages of any number of users, with the number of messages limited per day. The messages containing viruses, or those that can not be scanned due to their internal errors, are not taken into account.

When the day limit is exhausted, the daemon will suspend scanning messages and the relevant return code will be sent to the mail filter.

Administrator can set up the filter (see the description of the `LicenseLimit` parameter of the configuration file) so that either to admit, or reject such messages, in CommuniGate Pro mail filter there exists an option to postpone the delivering of a message for some time for another scanning.

2.2. Licening per addresses

This license allows to use the daemon for scanning messages (incoming and out-going) of specified number of mail addresses and of an unlimited quantity of messages per day.

If 15- or 30-address license is chosen, all the protected addresses should be explicitly specified in a special file (see the description of the `MailAddressesList` parameter in p. 5.2). The file format is rather simple: one line per one address. First several lines (as specified in the license) are read only, regardless whether the line is empty or not, the rest is ignored.

To determine whether a particular address should be scanned or not, the daemon uses the addresses from the SMTP-envelope (see RFC 2821). To make the daemon scan a message, one of the addresses from the SMTP-envelope must be in full conformity with one of the "protected" addresses (case-insensitive comparison is applied).

That is why, if, for the address foo@bar.example.com the messages can be sent with the address `foo`, `foo@bar.example.com` or `foo@example.com`, the three of them should be listed in the correspondent file.

If a message is not identified as "for scanning", the daemon will not scan it and the relevant return code will be sent to a filter. Administrator can adjust the filter (see the description of the `LicenseLimit` parameter of the filter's configuration file) so that either to admit, or reject such messages.

If 50-address license is chosen, explicit specification of addresses is not required so far. Still, Doctor Web, Ltd. reserves the right to control in future the conformity of number of scanned addresses with the number specified in the license.

2.3. Licensing per servers

This license allows to use the daemon for scanning unlimited number of messages of an unlimited number of user, but is limited per server.

2.4. Unlimited license

This license allows to user the daemon for scanning unlimited number of messages of an unlimited number of users on an unlimited number of servers.

3. File servers protection *(http://buy.drweb.com/business/)*

This is the license for the Dr.Web daemon which allows to use the daemon for scanning the shared resources in the Samba file servers (versions 2.2.2 and higher).

This license does not allow to use the daemon for integration with mail servers or Internet gateways, but enables their complex usage.

4. Internet Gateway protection *(http://buy.drweb.com/business/)*

This is the license which enables scanning of incoming HTTP traffic going through proxy servers supporting the icap-protocol (Squid, SafeSquid and Shewby). Scanning of incoming FTP traffic for Squid proxy servers is also possible.

The solution is licensed per users working through the gateway. Minimal license is for 20 users.

This license does not allow to use the daemon for integration with mail or file servers, but enables their complex usage.