Dr.WEB®

Anti-virus

for Unix Internet gateways

Defend what you create

Administrator Manual

Dr.Web® Anti-virus for Unix Internet gateways
Version 6.0.2
Administrator Manual
03.12.2014

Doctor Web Head Office
2-12A, 3rd str. Yamskogo polya
Moscow, Russia
125124

Web site: www.drweb.com
Phone: +7 (495) 789-45-87

Refer to the official web site for regional and international office information.

Doctor Web

Doctor Web develops and distributes Dr.Web® information security solutions which provide efficient protection from malicious software and spam.

Doctor Web customers can be found among home users from all over the world and in government enterprises, small companies and nationwide corporations.

Dr.Web antivirus solutions are well known since 1992 for continuing excellence in malware detection and compliance with international information security standards. State certificates and awards received by the Dr.Web solutions, as well as the globally widespread use of our products are the best evidence of exceptional trust to the company products.

We thank all our customers for their support and devotion to the Dr.Web products!

Table of Contents

Introduction 8

Terms and Abbreviations 10

System Requirements 12

Compatibility with Linux Distributions 13

Package File Location 14

Configuration Files 15

Logging 18

Allowed Actions 19

Installation and Deinstallation 20

Installation from Distribution Package for UNIX Systems 20

Using GUI Installer 22

Using Console Installer 26

Removing Distribution Package for UNIX Systems 29

Using GUI Uninstaller 30

Using Console Uninstaller 32

Updating Distribution Package for UNIX Systems 34

Installing from Native Packages 34

Starting Dr.Web for Unix Internet gateways 39

For Linux and Solaris OS 39

For FreeBSD OS 40

Configuring SeLinux Security Policies 41

Registration Procedure 44

Dr.Web Updater 46

Updating Anti-Virus and Virus Databases 46

Cron Configuration 47

Command Line Parameters 48

Blocking Updates for Selected Components 48

Restoring Components 49

Configuration 50

Updating Procedure 53

Dr.Web Agent 55

Operation Mode 55

Command Line Parameters 57

Configuration File **58**
[Logging] Section 58
[Agent] Section 58
[Server] Section 59
[EnterpriseMode] Section 60
[StandaloneMode] Section 61
[Update] Section 62
Running Dr.Web Agent **62**
Interaction with Other Suite Components **63**
Integration with Dr.Web Enterprise Security Suite **63**
Configuring Components to Run in Enterprise Mode 64
Automatic Creation of New Account by ES Server 64
Manual Creation of New Account by Administrator 65
Configuring Components via Dr.Web Control Center (embedded in Enterprise Security Suite) 65
Export of Existing Configuration to ES Server 65
Starting the System 65
Integration with Dr.Web ESS 10 **66**
Gathering Virus Statistics **67**
Dr.Web Monitor **71**
Operation Mode **71**
Command Line Parameters **72**
Configuration File **73**
[Logging] Section 73
[Monitor] Section 73
Running Dr.Web Monitor **76**
Interaction with Other Suite Components **76**
Dr.Web Command Line Scanner **78**
Running Dr.Web Scanner **78**
Command Line Parameters **79**
Configuration **84**
Exit Codes **91**
Dr.Web Daemon **92**
Command-Line Parameters **92**
Running Dr.Web Daemon **93**
Dr.Web Daemon Testing and Diagnostics **93**
Scan Modes **95**

Processed Signals 96
Log Files and Statistics 96
Configuration 97
Dr.Web ICAPD 106
Configuring Interaction between Dr.Web ICAPD and Squid 107
Configuring Interaction between Dr.Web ICAPD and SafeSquid 108
Configuring Squid to Scan FTP Traffic 109
ICAP Preview Mode 110
Black and White Lists 111
Content-Specific Black Lists 111
User-Defined Lists 111
Command Line Parameters 113
Settings of Dr.Web ICAPD 113
Configuration parameters 113
Redefining Parameters for User Groups 122
Variables 123
Logical expressions 123
Redefining Parameters: [match] section 125
Functions: [def] section 126
Example Usage 126
Configuring Squid to Operate with Variables 128
Setting Content Filtering by MIME Type and Size 128
Interaction between Dr.Web Agent and Dr.Web Monitor 129
Startup 130
How to Test Dr.Web ICAPD 130
Links to Squid and SafeSquid Project Websites 131
Notification Templates 131
Dr.Web Console for UNIX Internet Gateways 135
Installation 135
Basic Configuration 138
User Interface 139
Configuration 139
Actions Applied to Threats 140
Logging 142
Content Filter 143
System Settings 143

Traffic Filtering Rules 145
Quarantine 146
Templates 147
Running in Enterprise Mode 148
Configuring User Permissions 148
Configuring Workstation 150
Types of Administrator Accounts 151
Contacts 153
Appendix. The License Policy 154
Protection of Internet gateways 154

Introduction

This Manual describes the following anti-virus software:

- **Dr.Web® Anti-virus for Unix Internet gateways** for **Linux**;
- **Dr.Web® Anti-virus for Unix Internet gateways** for **FreeBSD**;
- **Dr.Web® Anti-virus for Unix Internet gateways** for **Solaris** x86.

As far as all these solutions for UNIX systems differ from each other only slightly, all of them will be referred to as **Dr.Web for Unix Internet gateways**. Critical differences are described in the corresponding chapters and paragraphs.

The manual is designed for a person responsible for anti-virus protection and security ("Administrator" hereinafter).

Protection of Internet gateways in UNIX systems has the following features:

- Monitoring of all incoming HTTP and FTP traffic to provide virus detection and neutralization.

  In most cases, viruses are not directly aimed at UNIX systems. For example, through the Internet ordinary **Windows** viruses are distributed, including macro viruses for **Word, Excel** and other **MS Office** applications.

- Filtration of access to HTML resources by their MIME type, size and host name.
- Restriction of access to Internet resources according to the black lists that are regularly updated.

Dr.Web for Unix Internet gateways solution consists of three major components and performs all of the tasks mentioned above.

Dr.Web for Unix Internet gateways includes the following components:

- **Dr.Web Scanner** - console anti-virus scanner that provides detection and neutralization of viruses on the local machine and in the shared directories;
- **Dr.Web Daemon** - a background that performs functions of an external anti-virus filter;
- **Dr.Web Monitor** - a resident component that runs and terminates other **Dr.Web** modules in the required order;
- **Dr.Web Agent** - a resident component that helps to configure and manage **Dr.Web** components, gathers statistics and provides integration with **Dr.Web Enterprise Security Suite** (**Dr.Web ESS**);

  By default, the solution includes **Dr.Web Agent**, designed for integration with **Dr.Web ESS** 6.0. If you want to integrate the suite with **Dr.Web ESS** 10.0, install the updates for **Dr.Web Agent** and perform additional configuration steps. For details, refer to the Dr.Web Agent section.

- **Dr.Web Engine** and virus databases that are regularly updated;
- **Dr.Web Updater** (implemented as a **Perl** script) - a component that provides regular updates to virus databases;
- **Dr.Web ICAP Daemon** (hereinafter **Dr.Web ICAPD**) allows to integrate other **Dr.Web** components with HTTP/FTP-proxy server using *ICAP* protocol;
- **Dr.Web Console for UNIX Internet Gateways** – web management interface, a **Webmin** built-in module, used for **Dr.Web for Unix Internet gateways** management and configuration via the web interface from any browser.

The following picture shows the structure of **Dr.Web for Unix Internet gateways** and its components.

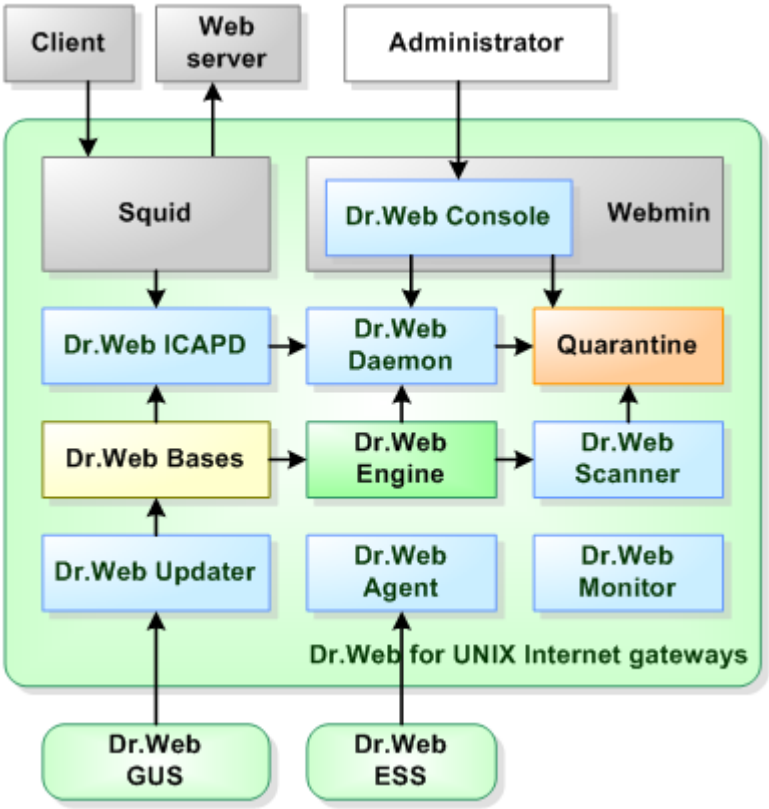


Figure 1. Structure of Dr.Web for Unix Internet gateways and its components

The present manual provides information on setup, configuration, and usage of **Dr.Web for Unix Internet gateways**, that is:

- General product description
- Installation of **Dr.Web for Unix Internet gateways**
- Running **Dr.Web for Unix Internet gateways**
- Usage of **Dr.Web Updater**
- Usage of **Dr.Web Agent**
- Usage of console scanner **Dr.Web Scanner**
- Usage of background on-demand scanner **Dr.Web Daemon**
- Usage of **Dr.Web Monitor**
- Usage of module **Dr.Web ICAPD**

At the end of this manual, you can find contact information for technical support.

Doctor Web products are constantly developed. Updates to virus databases are issued daily or even several times a day. New product versions appear. They include enhancements to detection methods, as well as to the means of integration with UNIX systems. Moreover, the list of applications compatible with **Doctor Web** is constantly expanding. Therefore, some settings and functions described in this Manual can slightly differ from those in the current program version. For details on updated program features, refer to the documentation delivered with an update.

Terms and Abbreviations

The following conventions are used in the Manual:

Convention	Description
Bold	Names of buttons and other elements of the graphical user interface (GUI), and required user input that must be entered exactly as given in the guide.
Green and bold	Names of **Doctor Web** products and components.
Green and underlined	Hyperlinks to topics and web pages.
`Monospace`	Code examples, input to the command line and application output.
Italics	Placeholders which represent information that must be supplied by a user. For command-line input, it indicates parameter values.
CAPITAL LETTERS	Names of keys and key sequences.
Plus sign ('+')	Indicates a combination of keys. For example, ALT+F1 means to hold down the ALT key while pressing the F1 key.
Exclamation mark	A warning about potential errors or any other important comment.

To define directories, where the suite components are installed, the following conventions are used: `%bin_dir`, `%etc_dir` and `%var_dir`. Depending on the OS, these symbols refer to the following directories:

for Linux and Solaris:

```
%bin_dir = /opt/drweb/
%etc_dir = /etc/drweb/
%var_dir = /var/drweb/
```

for FreeBSD:

```
%bin_dir = /usr/local/drweb/
%etc_dir = /usr/local/etc/drweb/
%var_dir = /var/drweb/
```

The following conventions are used in the Manual:

Abbreviation	Description
ASCII	American Standard Code for Information Interchange
CIDR	Classless Inter-Domain Routing
DEB	Extension for package files for software distribution in **Debian** (and others used **dpkg**)
DNS	Domain Name System
HTML	HyperText Markup Language
IP	Internet Protocol
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
IPC	Inter-Process Communication
MD5	Message Digest 5 algorithm
OS	Operating System
PID	Process IDentifier in UNIX based OS
POSIX	Portable Operating System Interface for Unix
RFC	Request for Comments

Abbreviation	Description
RPM	Package files format (and extension) for **Red Hat Package Manager**
SSL	Secure Socket Layers protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security protocol
URL	Uniform Resource Locator
UUID	Unique User IDdentifier
XML	eXtensible Markup Language

The following abbreviations are used in chapters about components **Dr.Web ICAPD** and **Dr.Web Console for UNIX Internet Gateways**:

Abbreviation	Description
CGI	Common Gateway Interface
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICAP	Internet Content Adaptation Protocol
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions

System Requirements

Dr.Web for Unix Internet gateways is compatible with

- **Linux** distributions that meet requirements listed in Compatibility with Linux Distributions;
- **FreeBSD** version 6.x and higher for Intel x86 and amd64 platform;
- **Solaris** version 10 for Intel x86 and amd64 platform.

Used platform must be fully compatible with x86 processor architecture in 32-bit or 64-bit modes. 64-bit systems must support 32-bit applications.

The products, operating in **FreeBSD** 6.x, cannot be integrated with **Dr.Web ESS** 10.

For example:

To enable support for 32-bit applications in systems based on **Debian**/**Ubuntu Linux** the `libc6-i386` library must be installed, in systems based on **ALT Linux** - the `i586-glibc-core` library.

For successful operation of **Dr.Web for Unix Internet gateways**, it is required to:

- Install and run **Dr.Web Daemon** and anti-virus **Dr.Web Engine** version 6.0.2 or later.
- Install and run **Squid** version 3.0.STABLE1 or later, or **SafeSquid** version 3.0 or later.
- Installed **Perl** 5.8.0 or later for **Dr.Web Updater**.

Dr.Web for Unix Internet gateways hardware requirements are the same as requirements for the command line interface of the compatible operating system.

Installation requires 205 megabytes.

GUI installer of **Dr.Web for Unix Internet gateways** requires **X Window System**. Execution of interactive configuration script in graphical mode requires `xterm` or `xvt` terminal emulators.

In addition to that, the following packages must be installed in your system:

- `base64`
- `unzip`
- `crond`

For successful installation of **Dr.Web for Unix Internet gateways** in **FreeBSD** OS (version later than 8.0), the `compat7x` library is required.

Depending on the range of problems to be solved by **Dr.Web for Unix Internet gateways** and operational load, meeting additional hardware requirements can be necessary.

Compatibility with Linux Distributions

Dr.Web for Unix Internet gateways solution is compatible with x86 and x86-64 **Linux** distributions.

Requirements for kernel versions and glibc library depend on the type of the installation package:

- Universal package for UNIX systems (**Linux x86**):
  - **`kernel`** version 2.4.x, **`glibc`** version 2.2 (not recommended) and later,

  OR
  - **`kernel`** version 2.6.x, **`glibc`** version 2.3 and later;
- Universal package for UNIX systems (**Linux x86-64**):
  - **`kernel`** version 2.6.x, **`glibc`** version 2.3 (recommended) and later;
- Native RPM distribution packages (**rpm-apt**, **urpmi**, **yum**, **zypper**):
  - **`kernel`** version 2.6.18 and later, **`glibc`** version 2.5 and later;
- Native DEB distribution packages (**apt**):
  - **`kernel`** version 2.6.26 and later, **`glibc`** version 2.7 and later.

Performance of **Dr.Web for Unix Internet gateways** was tested on the following distributions:

Linux distribution	Versions	
	32-bit	64-bit
ALT Linux	4.0 – 5.0 СПТ 6.0 (ru)	5.0 СПТ 6.0 (ru)
Arch Linux	–	all
ASPLinux	12.0 – 14.0	–
Debian	3.1 – 6.0	4.0 – 6.0
Fedora	–	14.0
Gentoo	all	
Mandriva Linux	higher than 2009, CS4	2010.x
Mandrake	10.x	10.x
openSUSE	10.3 – 11.0	10.3 – 11.0
PCLinux	2010	2010
RedHat Enterprise Linux (RHEL)	4.0 – 6.0	5.0 – 6.0
Suse Linux Enterprise Server	9.0 – 11.0	10.0 – 11.0
Ubuntu	7.04 – 11.04	7.04 – 11.04

Compatibility with MSVS OS

Dr.Web for Unix Internet gateways is compatible with the following versions of **MSVS** OS:

- **MSVS** 3.0 80001-12 (rev. 0, 1, 2, 3);
- **MSVS** 3.0 80001-14 (rev. 0, 1, 2);
- **MSVS** 3.0 80001-08;
- **MSVS** 3.0 80001-16;
- **MSVS** 3.0 FSTEK.

Other **Linux** distributions that meet the requirements mentioned above are also supported (but they were not tested). If you encounter any compatibility problems with the used **Linux** distribution, please

contact technical support at http://support.drweb.com/request/.

Package File Location

Dr.Web for Unix Internet gateways solution is installed to the default `%bin_dir`, `%etc_dir` and `%var_dir` directories. OS independent directory tree is created in the following directories:

- `%bin_dir` – directory with executable modules of **Dr.Web for Unix Internet gateways** and **Dr.Web Updater** (perl script `update.pl`);
- `%bin_dir/doc/` – documentation on the product. All documentation is available in both Russian and English languages and represented in `KOI8-R` и `UTF-8` text files.

`%bin_dir/lib/` – directory with various service libraries and supporting files for **Dr.Web for Unix Internet gateways** component operation, for example:
  - `ru_scanner.dwl` – file of **Dr.Web Scanner** language resources.
-
- `%bin_dir/web/` – **Dr.Web for Unix Internet gateways** web interface module for connection to **Webmin**.
- `%etc_dir/` – directory with **Dr.Web for Unix Internet gateways** configuration and enable files that manage startup of components operating in daemon mode*
- `%etc_dir/agent/` – directory with additional configuration files for **Dr.Web Agent**;
- `%etc_dir/monitor/` – directory with additional configuration files for **Dr.Web Monitor**;
-
- `%var_dir/bases/` – directory with virus databases (`*.vdb` files);
- `%var_dir/infected/` – **Quarantine** folder that serves for isolation of infected or suspicious files if the corresponding action is specified in **Dr.Web for Unix Internet gateways** settings.
- `%var_dir/lib/` – anti-virus engine implemented as a loadable library (**`drweb32.dll`**).

*) Directory of the `enable` files depends on **Dr.Web for Unix Internet gateways** installation method:

- Installation using the **universal package for UNIX systems**:

  Files are stored in the `%etc_dir` directory and named as follows
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.

- Installation using the **native DEB packages**:

  Files are stored in the `/etc/defaults` directory and named as follows
  `drweb-icapd`,
  `drwebd`,
  `drweb-monitor`.

- Installation using **native RPM packages**:

  Files are stored in the `/etc/sysconfig` directory and named as follows
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.

Configuration Files

General format of configuration files

All **Dr.Web for Unix Internet gateways** settings are stored in configuration files which you can use to configure all suite components. Configuration files are text files, so they can be edit in any text editor. They have the following format:

```
--- beginning of file ---

[Section 1 name]
Parameter1 = value1, ..., valueK
...
ParameterM = value1, ..., valueK

[Section X name]
Parameter1 = value1, ..., valueK
...
ParameterY = value1, ..., valueK

--- end of file ---
```

Configuration files are formed according to the following rules:

- Symbols ';' or '#' mark the beginning of a comment. Text that follows these symbols is ignored by **Dr.Web for Unix Internet gateways** modules when reading a file.
- Contents of the file is divided into sets of named sections. Possible section names are hardcoded and cannot be changed. The section names are specified in square brackets.
- Each file section contains configuration parameters, grouped by meaning.
- One line contains a value (or values) only for one parameter.
- General format for parameter value setting (spaces enclosing the '=' signed are ignored) is the following:

  ```
  <Parameter name> = <Value>
  ```

- Parameter names are hardcoded and cannot be changed.
- Names of all sections and parameters are case insensitive.
- Order of sections in a file and order of parameters in sections are of no consequence.
- Parameter values in a file may be enclosed in quotation marks (and must be enclosed in quotation marks if they contain spaces).
- Some parameters can have more than one value. In this case, parameter values are separated by a comma or each parameter value is set separately in different lines of the configuration file. If values of a parameter are separated by commas, spaces between a comma and a value are ignored. If a space is a part of a value, the whole value must be enclosed in quotation marks.

  If a parameter can have several values, that is explicitly designated. If the possibility to assign several values to a parameter is not explicitly designated, the parameter can have only one value.

  Example of assigning several values to a parameter:

  1) Separating values by commas:

  ```
  Parameter = Value1, Value2,"Value 3"
  ```

2) Setting of each parameter value separately:

```
Parameter = Value2
Parameter = Value1
Parameter = "Value 3"
```

If a parameter is not specified in a configuration file, this does not mean that the parameter does not have any value. In this case, the parameter value is assigned by default. Only a few parameters are optional or do not have default values, which is mentioned separately.

Parameter description rules used in this Manual

Each parameter in this manual is described as follows:

ParameterName = {Parameter type | Possible values}	Description {Whether more than one value is possible} {Special remarks} {Important remarks}
	Default value: **ParameterName** = {value | nothing}

Description of parameters is provided in this document in the same order as they are specified in the corresponding configuration file created upon **Dr.Web for Unix Internet gateways** installation.

The `Parameter type` field can be one of the following:

- **`numerical value`** — parameter value expressed as a whole non-negative number.
- **`time`** — parameter value expressed as a date unit. The value is a whole number that can be followed by a symbol defining the type of a date unit (`s` – seconds, `m` – minutes, `h` – hours; symbol is case insensitive). If the value does not have a symbol, the parameter is expressed in seconds (by default).

  Examples: `30h`, `15m`, `6` (in the last example, time is expressed in seconds).
- **`size`** — parameter value expressed as a unit of memory size (disk space or RAM). The value is a combination of a whole number that can be followed by a symbol defining the type of a memory size unit (`b` – bytes, `k` – kilobytes, `m` – megabytes, `g` – gigabytes; symbol is case insensitive). If the value does not have a symbol, the parameter is expressed in bytes.

  Examples: `20b`, `15k`
- **`permissons`** — parameter value expressed as a three-digit number which determines file access permissions in UNIX format:
  Each permission is a combination (sum) of three base permissions:
  - Read permission (r) is specified by 4;
  - Write permission (w) is specified by 2;
  - Execute permission (x) is specified by 1.

  First digit in the value defines permissions for the file owner, second digit - for owner's group, and third digit - for all other users (neither owners nor members of the group).

  Examples: `755`, `644`
- **`logical (Yes/No)`** — parameter value expressed as a string that can be one of the following: "`Yes`" or "`No`".

- **path to file/directory** — parameter value expressed as a string which contains a path to a file or folder in the file system. Note, that names of files and folders are case sensitive. If mentioned, you can specify a file mask as a parameter value. A **mask** can include the following symbols:
  - ? – replaces one symbol in the file (folder) name;
  - * – replaces any sequence of symbols (including an empty sequence) in the file (folder) name.

  Example: "?.e*" – this mask defines all files with a name consisting of only one character and with an extension which is of any length and starts with "e" (x.exe, g.e, f.enable and others).
- **action** — parameter value expressed as a string which contains actions (those that are applied to objects by **Dr.Web for Unix Internet gateways** components). In some cases, the parameter can have one basic and three additional actions specified (in such a case, the name of the parameter type is **actions list)**. Basic action must be the first in the list. Different parameters can have a different action list and, in this case, it is specified separately for each parameter. For information on available actions, see Allowed actions.
- **address** — parameter value expressed as a string which contains socket address of a **Dr.Web for Unix Internet gateways** component or used external program.
  Address is of the following format: TYPE:ADDRESS. There are three available TYPEs:
  - inet — a TCP socket, ADDRESS is specified in the following format: PORT@HOST_NAME, where HOST_NAME can be either a direct IP address or domain name of the host.

    Example:

    ```
    Address = inet:3003@localhost
    ```

  - local — a local UNIX socket, ADDRESS is a path to the socket file.

    Example:

    ```
    Address = local:%var_dir/.daemon
    ```

  - pid — a real process address that is to be read from the process PID file. This address type is allowed only in certain cases that are explicitly designated in the parameter description.
- **text value, string** — parameter value expressed as a text string. The text can be enclosed in quotation marks (and the text must be enclosed in quotation marks if it contains spaces).
- **log level** — parameter value expressed as a string which contains the verbosity level of logging into the file or **syslog** system service.
- **value** — parameter has the type that is not described in the previous items of the list. In this case, all available values are provided.

Behaviour of the modules if configuration file parameters are ill-defined

- If any parameter value is incorrect, the respective **Dr.Web for Unix Internet gateways** module outputs an error message and terminates.
- If any unknown parameter is found when loading a configuration file, **Dr.Web for Unix Internet gateways** logs the corresponding message and continues operation in the normal mode.

Some parameters can use regular expressions as values (that is mentioned in the description of the corresponding parameter). Regular expression syntax of **Perl** is used by default. For information on regular expressions, see a corresponding article, for example, on the **Wikipedia** website (Regular expressions article).

Logging

All **Dr.Web for Unix Internet gateways** components keep records about their operation in the logs. You can set a log mode for each component (output of information into the file or to **syslog**).

You can also select a log verbosity level: for example, set high level of verbosity (the `Debug` option) or disable logging (the `Quiet` option). To set the verbosity level, use the **`LogLevel`** parameter. You can also specify additional parameters for certain plug-ins to configure their verbosity log level (for example, keeping records of IPC subsystem operation is modified by the **`IPCLevel`** parameter).

If the **`LogLevel`** configuration parameter is not available for a plug-in, it is not allowed to adjust its log mode. In this case, the default log mode has a verbosity level similar to `Debug`.

Log verbosity levels

If allowed, you can set one of the following log verbosity levels for a **Dr.Web for Unix Internet gateways** component (the list is arranged in ascending order of detail):

- `Quiet` – Logging is disabled.
- `Error` – The component logs only fatal errors.
- `Alert` – The component logs errors and important warnings.
- `Warning` – The component logs errors and all warnings.
- `Info` – The component logs errors, warnings and information messages.
- `Notice` – This mode is similar to the `Info` mode, but the component also logs notifications.
- `Debug` – This mode is similar to the `Notice` mode, but the component also logs debug information.
- `Verbose` – The component logs all details on its activity (this mode is not recommended, because a large volume of logged data can considerably reduce performance of both the program and **syslog** service if it is enabled).

Each **Dr.Web for Unix Internet gateways** component can have different set of allowed log verbosity levels. For information on available verbosity levels, see description of the corresponding parameters.

Logging into syslog

If you select the mode of logging information into **syslog**, it is necessary to specify a verbosity log level and a message source label. The label can be used by the **syslog** service for internal routing of messages to different logs. Routing rules are configured in the **syslog** daemon configuration file (usually, the path to the file is `/etc/syslogd.conf`).

To set a flag for syslog messages, specify **`SyslogFacility`** parameter value in configuration files. You can specify one of the following parameter values:

- `Daemon` – label of a resident system service (daemon) message;
- `Local0, ..., Local7` – label of a user application message (8 values are reserved `Local0` to `Local7`);
- `Kern` – label of a system kernel message;
- `User` – label of a user process message;
- `Mail` – label of a mail system message.

Note that if information is logged into **syslog**, an additional parameter - **`SyslogPriority`** - can be specified in configuration files. **`SyslogPriority`** defines a verbosity level of logging into **syslog** and is modified by one of the values available for the **`LogLevel`** parameter. If you select the mode of logging into the file, **`SyslogPriority`** is ignored. Otherwise, information is logged into **syslog** with

the less verbosity level.

Example:

Let us assume that logging of component operation is defined by the following parameter values: **`LogLevel`** = `Debug`, **`SyslogPriority`** = `Error`. If mode of logging into **syslog** is selected, the log verbosity level is `Error` (that means only records about errors are to be logged and the `Debug` value is ignored).

Allowed Actions

You can configure **Dr.Web for Unix Internet gateways** components to apply specified actions to objects that are detected to be malicious, suspicious or potentially dangerous.

Different parameters can have different available actions, they are listed in each parameter description.

You can use the following actions when configuring the settings:

- `Move` – move the file to the **Quarantine** folder, send to the user the HTML page with notification;
- `Truncate` – truncate the file to a zero length and send it to the user;
- `Pass` – pass the file to the user;
- `Report` – log the information, send to the user the HTML page with notification;
- `Cure` – try to cure the infected object.

You can use the following actions when configuring **Dr.Web Scanner**:

- `Move` – move the file to the **Quarantine** folder;
- `Delete` – delete the infected file;
- `Rename` – rename the file;
- `Ignore` – ignore the file;
- `Report` – only log information about the file;
- `Cure` – try to cure the infected object.

Please note that action names are case insensitive (for example, value `Report` equals to `report`).

Installation and Deinstallation

Below you can find detailed description of **Dr.Web for Unix Internet gateways** installation, update and uninstallation procedures in UNIX systems. You need superuser (`root`) privileges to perform these operations. To get it, use the **`su`** command or **`sudo`** prefix.

If previously the product was installed from packages of other formats (for example, RPM or DEB), ensure that they are carefully uninstalled.

Dr.Web for Unix Internet gateways distribution package for UNIX systems is delivered in EPM format (script-based distribution package with installation and uninstallation scripts and standard install/uninstall GUIs) designed to use with ESP Package Manager (EPM). Please note that all these scripts relate to the EPM package, not to any of the **Dr.Web for Unix Internet gateways** components.

You can install, deinstall, and update **Dr.Web for Unix Internet gateways** in one of the following ways:

- using GUI;
- using console scripts.

During installation, dependencies are supported, that is if a component installation requires other components to be installed in the system (for example, `drweb-daemon` package requires **`drweb-common`** and **`drweb-bases`** packages), they will be installed automatically.

If you install **Dr.Web for Unix Internet gateways** to a computer where other **Dr.Web** products have been previously installed from EPM packages, then at every attempt to remove a module via graphical installer you will be prompted to remove absolutely all **Dr.Web** modules, including those from other products.

Please, pay special attention to the actions you perform and selections you make during uninstallation to avoid accidental removal of some useful components.

Installation from Distribution Package for UNIX Systems

Dr.Web for Unix Internet gateways solution is distributed as a self-extracting package `drweb-internet-gateways_[version number]~[OS name].run`.

The following components are included in this distribution:

- `drweb-common`: contains the main configuration file - `drweb32.ini`, libraries, documentation and directory structure. During installation of this component, `drweb` user and `drweb` group are created;
- `drweb-bases`: contains Anti-virus search Engine (**Dr.Web Engine**) and virus databases. It requires `drweb-common` package to be installed;
- `drweb-libs`: contains common libraries for all the components of the suite;
- `drweb-epm6.0.2-libs`: contains libraries for graphical installer and uninstaller. It requires `drweb-libs` package to be previously installed;
- `drweb-epm6.0.2-uninst`: contains files of graphical uninstaller. It requires `drweb-libs` package to be previously installed;
- `drweb-boost147`: containes common libraries for **Dr.Web Agent** and **Dr.Web Monitor**. It requires `drweb-libs` package to be previously installed;
- `drweb-updater`: contains update utility - **Dr.Web Updater** for **Dr.Web Engine** and virus databases. It requires `drweb-common` and `drweb-libs` packages to be installed;
- `drweb-agent`: contains **Dr.Web Agent** executable files and its documentation. It requires `drweb-common` and `drweb-boost147` packages to be installed;

- `drweb-agent-es`: contains files required for communication between **Dr.Web Agent** and **Dr.Web ESS** server version 6 in central protection mode. It requires `drweb-agent`, `drweb-updater and drweb-scanner` packages to be installed;
- `drweb-agent10`: contains executable files and documentation for the updated **Dr.Web Agent** (designed for operation with **Dr.Web ESS** server version 10).
- `drweb-agent10-es`: contains files required for communication between the updated **Dr.Web Agent** and **Dr.Web ESS** server version 10 in central protection mode.
- `drweb-daemon`: contains **Dr.Web Daemon** executable files and its documentation. It requires `drweb-bases` and `drweb-libs` packages to be previously installed;
- `drweb-scanner`: contains **Dr.Web Scanner** executable files and its documentation. It requires `drweb-bases` and `drweb-libs` packages to be installed;
- `drweb-monitor`: contains **Dr.Web Monitor** executable files and its documentation. It requires `drweb-agent`, `drweb-common` and `drweb-boost147` packages to be installed;
- `drweb-icapd`: contains **Dr.Web ICAPD** executable files and its documentation. It requires `drweb-common`, `drweb-icapd-dws` and `drweb-libs` packages to be installed;
- `drweb-icapd-dws`: contains content-specific black and white lists of Internet resources. It requires `drweb-common` package to be installed;
- `drweb-icapd-web`: contains web interface of **Dr.Web for Unix Internet gateways**;
- `drweb-internet-gateways-doc`: contains documentation for **Dr.Web for Unix Internet gateways**.

In distributions for 64-bit systems, two additional packages are included: `drweb-libs` and `drweb-libs32`, which contain libraries for 64 and 34-bit systems correspondingly.

To install all **Dr.Web for Unix Internet gateways** components automatically, use either console (CLI) or the default file manager of your GUI-based shell. In the first case, allow the execution of the corresponding self-extracting package with the following command:

```
# chmod +x drweb-internet-gateways_[version number]~[OS name].run
```

and then run it:

```
# ./drweb-internet-gateways_[version number]~[OS name].run
```

As a result,
`drweb-internet-gateways_[version number]~[OS name]`
directory is created, and the GUI installer starts. If it starts without root privileges, the GUI installer tries to gain required privileges.

If the GUI installer fails to start, then interactive console installer starts automatically.

If you need only to extract the content of the package without starting the GUI installer, use `--noexec` command line parameter:

```
# ./drweb-internet-gateways_[version number]~[OS name].run --noexec
```

After you extract the content, you can start the GUI installer and continue setup with the following command:

```
# drweb-internet-gateways_[version number]~[OS name]/install.sh
```

To install with the use of the console installer, use the following command:

```
# drweb-internet-gateways_[version number]~[OS name]/setup.sh
```

Installation, regardless of the used method, includes the following steps:

- Original configuration files are recorded to the `%etc_dir/software/conf/` directory with the

following names: `[configuration_file_name].N`.

- Operational copies of configuration files are installed to the corresponding directories.
- Other files are installed. If a file with the same name already exists in the directory (e.g. after inaccurate removal of previous package versions), it is overwritten with the new file, and a copy of the old one is saved as `[file_name].O`. If a file with the `[file_name].O` name already exists in this directory, it is replaced with the new file.
- If you select the **Run interactive postinstall script** check box in the corresponding window of the GUI installer, then after installation of the components completes, the post-install script is initialized for **Dr.Web for Unix Internet gateways** basic adjustment.

Please note that if the used **Linux** distribution features **SELinux**, installation can be interrupted by the security subsystem. If such situation occurs, set **SELinux** to `(Permissive)` mode. To do this, enter the following command:

```
# setenforce 0
```

and restart the installer.

After the installation completes, configure **SELinux** security policies to enable correct operation of anti-virus components.

You can remove the `drweb-internet-gateways_[version number]~[OS name]` directory and .run file after successful completion of installation.

Using GUI Installer

To install with GUI

1. Enter the following command:

```
# drweb-internet-gateways_[version number]~[OS name]/install.sh
```

The setup program launches. On the Welcome screen, click **Next**.

At any step you can return to the previous one by clicking **Back**. To continue installation, click **Next**. To abort installation, click **Cancel**.

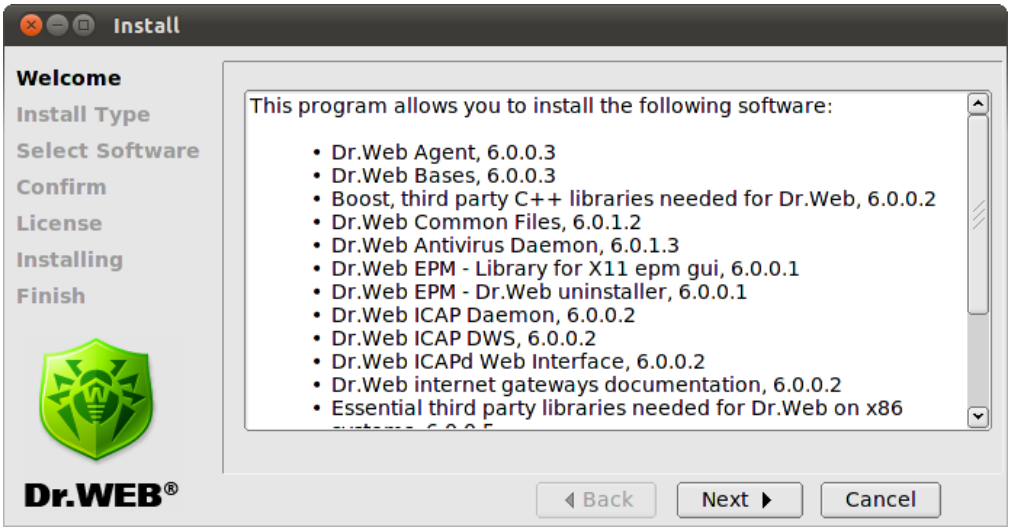


Figure 2. Welcome screen

2. On the **Install Type** screen, select the installation type: typical configuration for **Dr.Web for Internet Gateways** with all the necessary components selected by default or custom configuration.

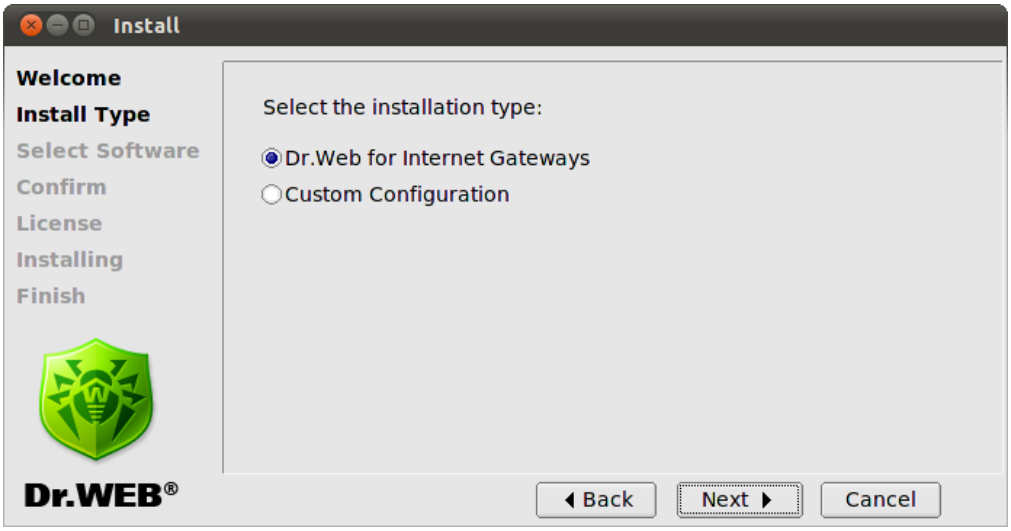


Figure 3. Install type window

If you selected **Custom Configuration**, then select necessary components on the **Select Software** screen:

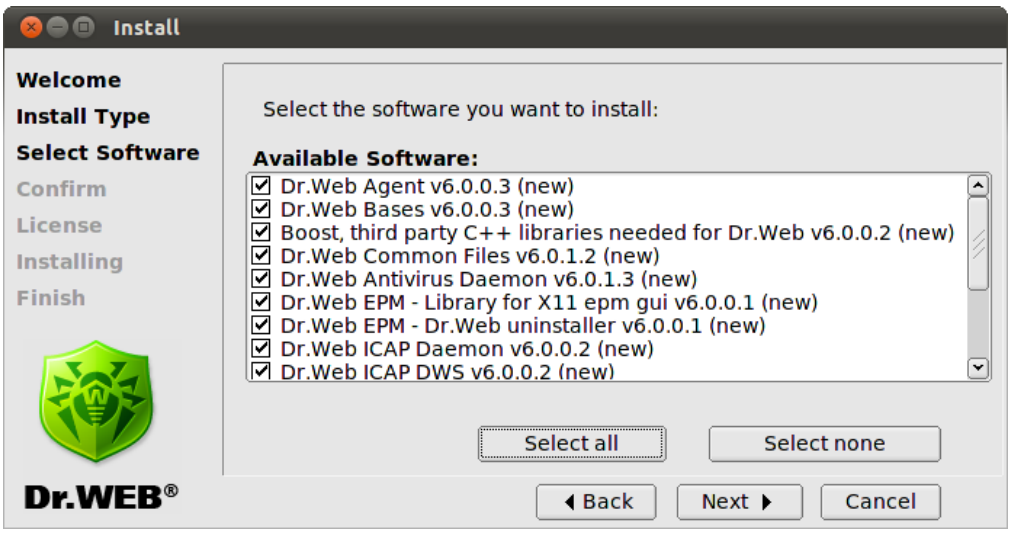


Figure 4. Select Software screen

If installation of a component requires some other components to be previously installed, all corresponding dependencies are selected for installation automatically. For example, if you select to install **Dr.Web Antivirus Daemon**, then **Dr.Web Bases** and **Dr.Web Common Files** are installed automatically.

Click to **Select all** to select all components. Click **Install None** to clear selection.

3. On the **Confirm** screen, review and confirm the list of components to install:

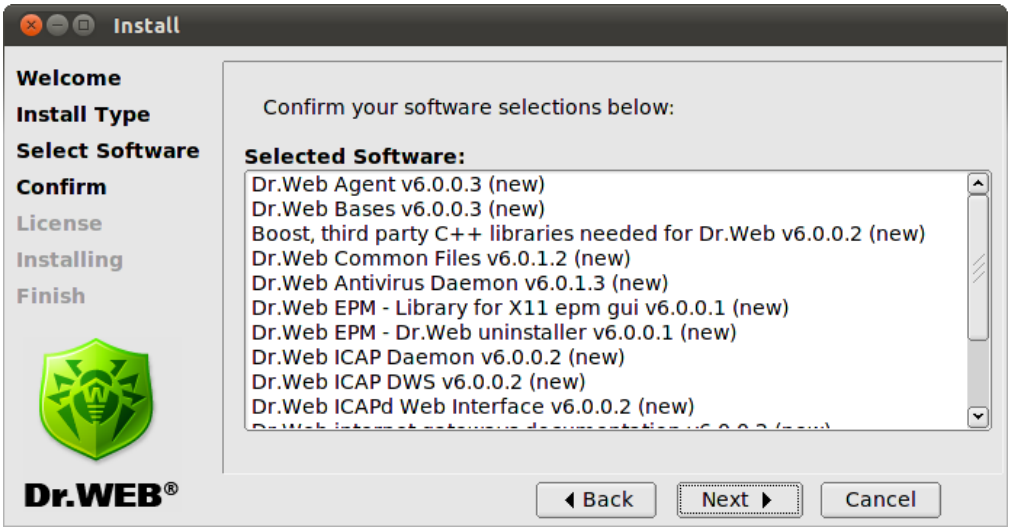


Figure 5. Confirm screen

Click **Next** to confirm selection, or click **Back** to make changes.

4. Review the **License Agreement**. To proceed, you need to accept it. If necessary, use the **Language** list to select a preferred language of the agreement (Russian and English languages are available):

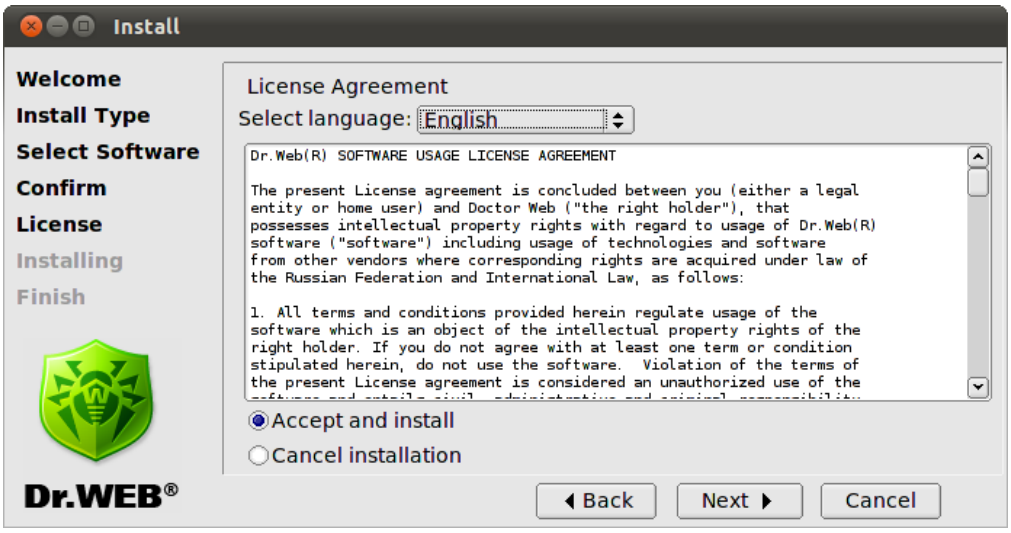


Figure 6. License Agreement screen

5. After you accept the **License Agreement**, installation starts. On the **Installing** screen, you can review the installation process in real-time:

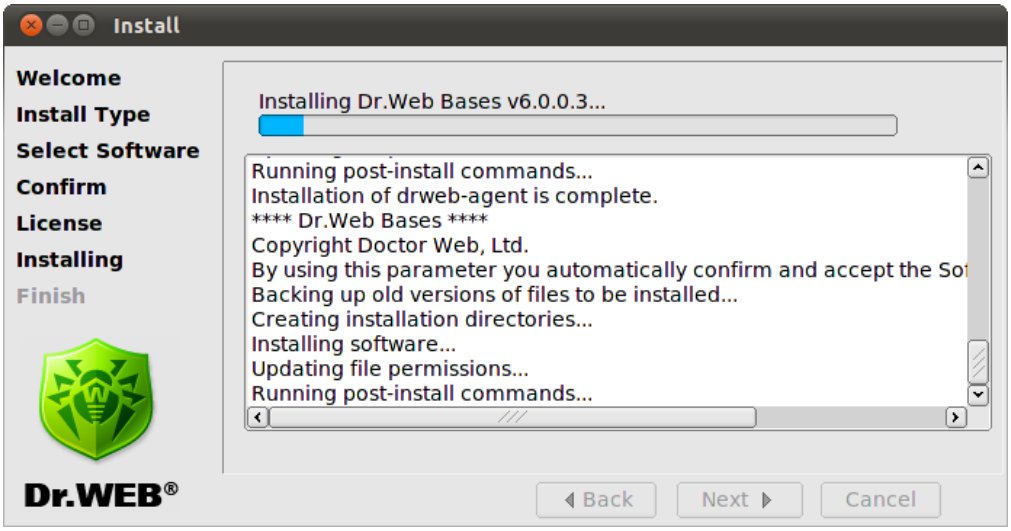


Figure 7. Installing screen

This report is logged at the same time in the `install.log` log file located at the `drweb-internet-gateways_[version number]~[OS name]` directory. If you selected **Run interactive post-install script**, once component installation completes, the post-install script for **Dr.Web for Unix Internet gateways** basic configuration initializes.

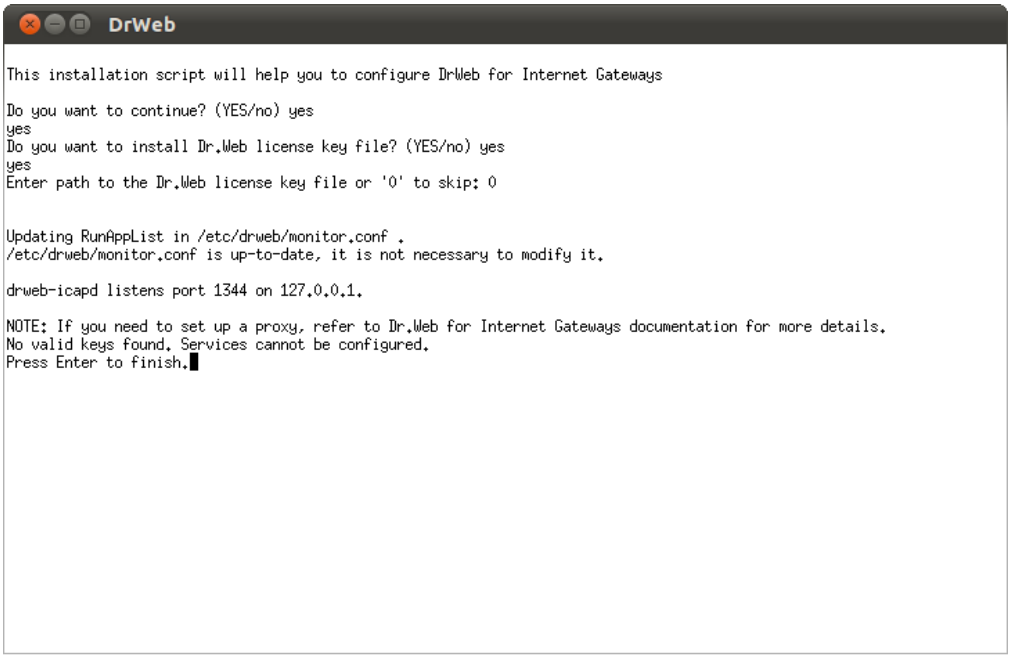


Figure. 8. Interactive post-install script

After initialization of the script, you can specify a path to the key file, set an order of mail processing by the plug-ins and automatically enable services necessary for **Dr.Web for Unix Internet gateways** proper operation (for example, **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**).

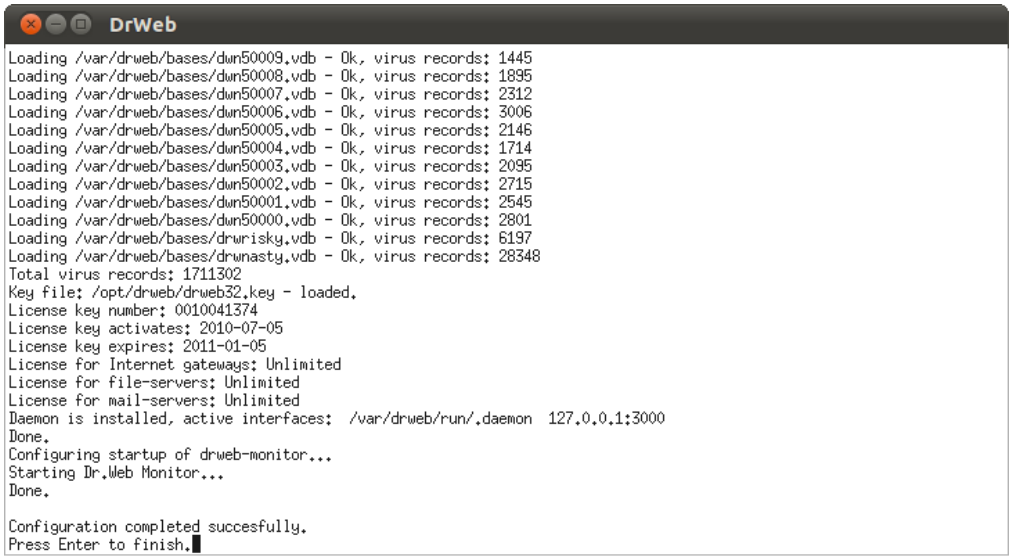


Figure 9. Starting services

On the **Finish** screen, you can see a notification that further adjustment is required to provide proper operation of **Dr.Web for Unix Internet gateways**, click **Close** to exit setup:

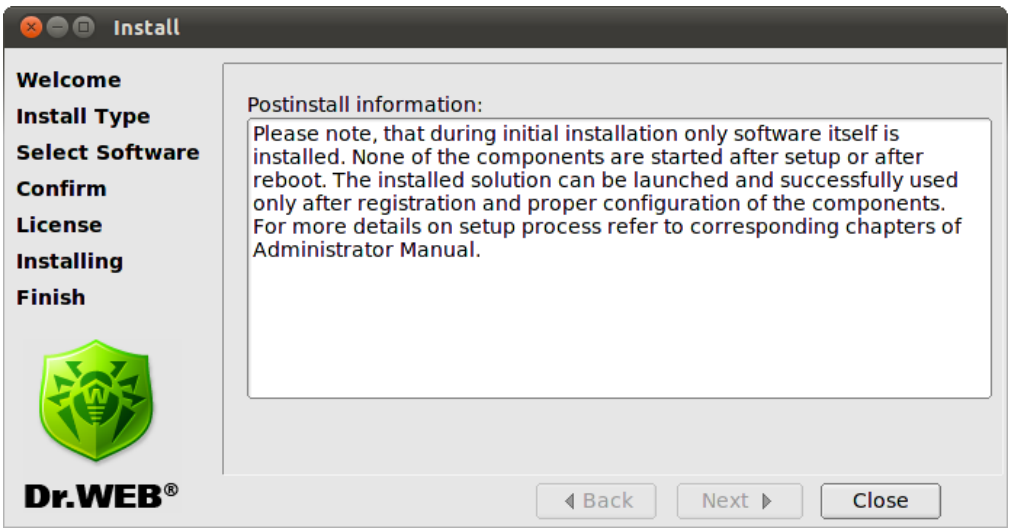


Figure 10. Finish screen

Using Console Installer

Console installer starts automatically if the GUI installer fails to start. If the console installer also fails to start (for example, if it is impossible to gain necessary privileges), you can try to run the following command with `root` privileges:

```
# drweb-internet-gateways_[version number]~[OS name]/setup.sh
```

To install from console

1. Once the console installer starts, the following dialog window opens:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка

This installation script will help you install DrWeb for Internet Gateways

Do you want to continue? (YES/no)
```

2. If you want to install **Dr.Web for Unix Internet gateways**, enter **Y** or **Yes** (values are case insensitive), otherwise enter **N** or **No**. Press ENTER.
3. If you chose to install **Dr.Web for Unix Internet gateways**, installer suggests you to select the installation type:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Select the installation type:
        1       Dr.Web for Internet Gateways
        2       Custom Configuration

Choose one configuration to install [1] :
```

To select a required mode, enter the respective number and press ENTER.

4. If you selected **Custom Configuration**, specify required components to install:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Select the software you want to install:
        [ ] 1 Dr.Web Agent v6.0.0.3 (new)
        [ ] 2 Dr.Web Bases v6.0.0.3 (new)
        [ ] 3 Boost, third party C++ libraries needed for Dr.Web v6.0.0.2 (new)
        [ ] 4 Dr.Web Common Files v6.0.1.2 (new)
        [ ] 5 Dr.Web Antivirus Daemon v6.0.1.3 (new)
        [ ] 6 Dr.Web EPM - Library for X11 epm gui v6.0.0.1 (new)
        [ ] 7 Dr.Web EPM - Dr.Web uninstaller v6.0.0.1 (new)
        [ ] 8 Dr.Web ICAP DWS v6.0.0.2 (new)
        [ ] 9 Dr.Web ICAPd Web Interface v6.0.0.2 (new)
        [ ] 10 Dr.Web ICAP Daemon v6.0.0.2 (new)
        [ ] 11 Dr.Web internet gateways documentation v6.0.0.2 (new)
        [ ] 12 Essential third party libraries needed for Dr.Web on x86 systems
v6.0.0.5 (new)
        [ ] 13 Dr.Web Monitor v6.0.0.3 (new)
        [ ] 14 Dr.Web Antivirus Scanner v6.0.1.3 (new)
        [ ] 15 Dr.Web Updater v6.0.0.4 (new)

To select a package you want to install or deselect some previously
selected package - enter the corresponding package number and press Enter.

You may enter A or All to select all the packages, and N or None to deselect all
 of them.
Enter I or Install to install selected packages.
Enter 0, Q or Quit to quit the dialog.
All values are case insensitive.
Select:
```

To specify a required component, enter the respective number and press ENTER.

5. Review the **License Agreement**. To scroll the text, press SPACEBAR:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Dr.Web(R) SOFTWARE USAGE LICENSE AGREEMENT

The present License agreement is concluded between you (either a legal
entity or home user) and Doctor Web ("the right holder"), that
possesses intellectual property rights with regard to usage of Dr.Web(R)
software ("software") including usage of technologies and software
from other vendors where corresponding rights are acquired under law of
the Russian Federation and International Law, as follows:

1. All terms and conditions provided herein regulate usage of the
software which is an object of the intellectual property rights of the
right holder. If you do not agree with at least one term or condition
stipulated herein, do not use the software.  Violation of the terms of
the present License agreement is considered an unauthorized use of the
software and entails civil, administrative and criminal responsibility.

2. If you are a legal owner of the Software's copy, you receive the
--More--(24%)
```

To continue the installation, you need to accept the **License Agreement**. If you agree to the terms, enter **Y** or **Yes**. Otherwise, the installation aborts.

6. The installation process starts immediately. You can review results of the installation steps in the console in real time:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Creating installation directories...
Installing software...
Updating file permissions...
Running post-install commands...
Installation of drweb-libs is complete.
Backing up old versions of files to be installed...
Creating installation directories...
Installing software...
Updating file permissions...
Installation of drweb-boost144 is complete.
Backing up old versions of files to be installed...
Creating installation directories...
Installing software...
Checking configuration files...
Updating file permissions...
Running post-install commands...
Installation of drweb-agent is complete.
Copyright Doctor Web, Ltd.
```

7. Once installation of the components completes, the post-install script runs automatically to set up **Dr.Web for Unix Internet gateways** basic configuration. You are offered to specify the path to the license key file and automatically enable all the services necessary for **Dr.Web for Unix Internet gateways** proper operation (for example, **Dr.Web Daemon**, **Dr.Web Agent**, **Dr.Web Monitor**). In addition, you can .

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка

This installation script will help you to configure DrWeb for Internet Gateways

Do you want to continue? (YES/no) yes
yes
Do you want to install Dr.Web license key file? (YES/no) yes
yes
Enter path to the Dr.Web license key file or '0' to skip: 0


Updating RunAppList in /etc/drweb/monitor.conf .
/etc/drweb/monitor.conf is up-to-date, it is not necessary to modify it.

drweb-icapd listens port 1344 on 127.0.0.1.

NOTE: If you need to set up a proxy, refer to Dr.Web for Internet Gateways docum
entation for more details.
No valid keys found. Services cannot be configured.
Press Enter to finish.
```

Removing Distribution Package for UNIX Systems

To remove all the components of **Dr.Web for Unix Internet gateways** via GUI uninstaller, start it with the following command:

```
# %bin_dir/remove.sh
```

If startup is performed without root privileges, the GUI uninstaller tries to gain appropriate privileges.

If the GUI uninstaller fail to start, then interactive console uninstaller is initialized.

After uninstallation you can also remove `drweb` user and `drweb` group from your system.

During uninstallation, the following actions are performed:

- Original configuration files are removed from the `%etc_dir/software/conf/` directory.
- If operational copies of configuration files are not modified by the user, they are also removed. If the user made any changes to them, they are preserved.
- Other **Dr.Web** files are removed. If a copy of an old file was created during installation, this file is restored under the name it had before the installation. Such copies are usually named `[file_name].O`.
- License key files and log files are saved to their corresponding directories.

Using GUI Uninstaller

To uninstall with GUI

1. Enter the following command:

```
# %bin_dir/remove.sh
```

On the Welcome screen, click **Next**:

At any step, you can return to the previous stage by clicking **Back**. To continue installation, click **Next**. To abort uninstallation, click **Cancel**.

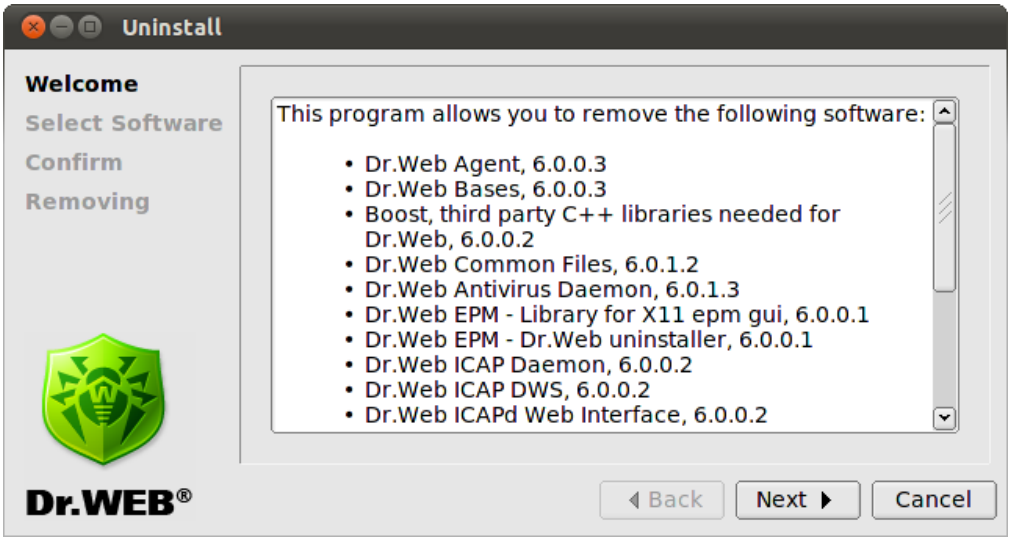


Figure 11. Welcome screen

2. On the **Select Software** screen, select components to remove:

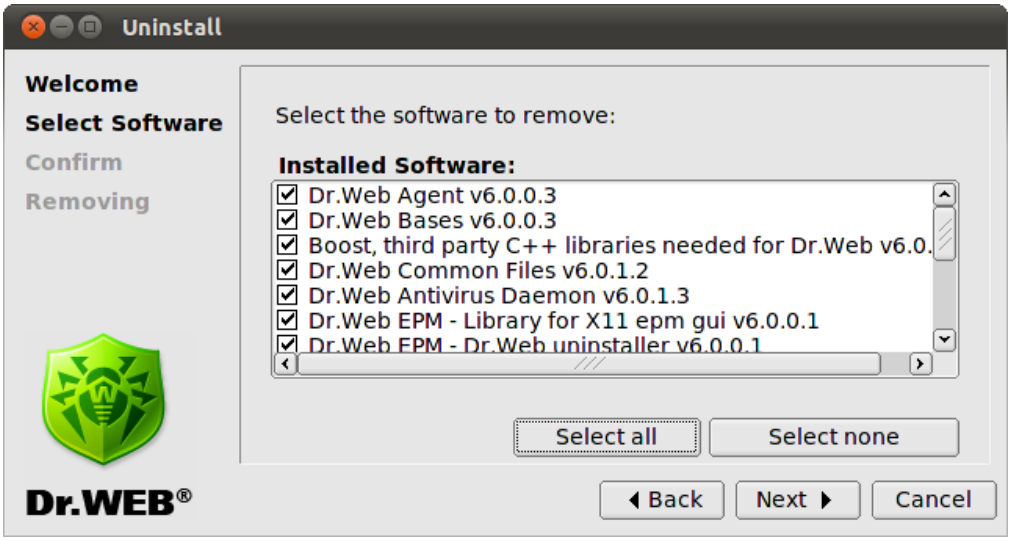


Figure 12. Select Software screen

All corresponding dependencies are selected to be uninstalled automatically.

If you installed **Dr.Web for Unix Internet gateways** on the computer with another **Dr.Web** product installed from EPM-packages, then the setup lists all **Dr.Web** modules for both **Dr.Web for Unix Internet gateways** and the older product. Please pay attention to the actions you perform and selection you make during uninstallation to avoid accidental removal of useful components.

Click **Select All** to select all components. To clear selection, click **Select None**.

When you complete selection, click **Next**.

3. On the **Confirm** screen, review and confirm the list of components to remove:

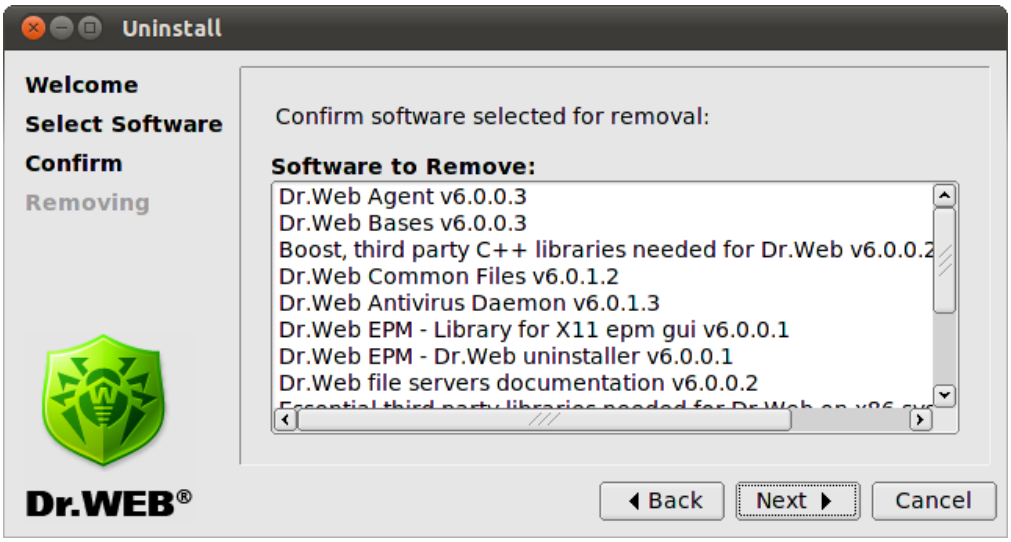


Figure 13. Confirm screen

Click **Next** to confirm selection, or click **Back** to make changes.

4. On the **Removing** screen, you can review results of the uninstallation steps in real time:

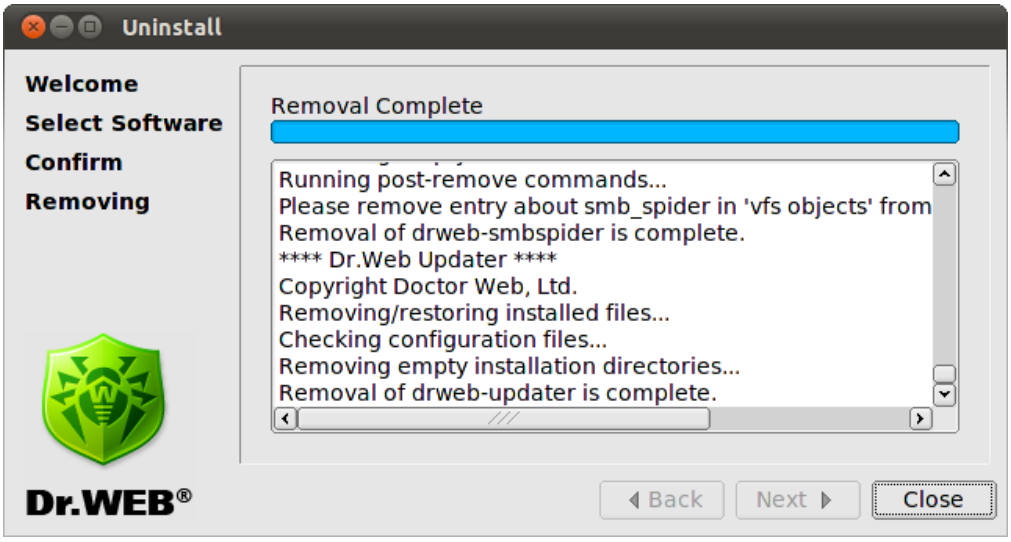


Figure 14. Removing screen

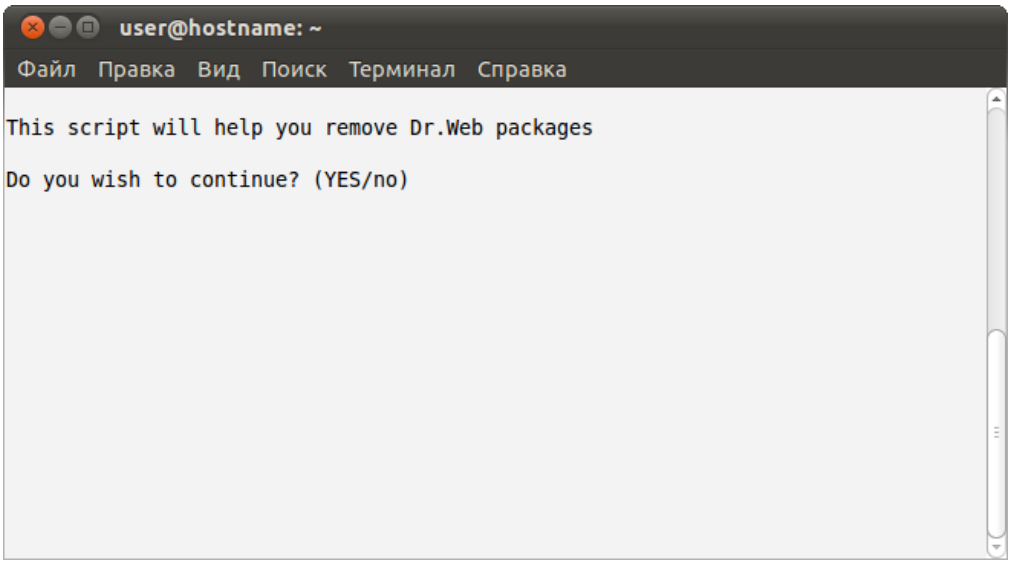
5. Click **Close** to exit setup.

Using Console Uninstaller

Console uninstaller starts automatically when graphical uninstaller fails to start.

To uninstall from console

1. Once the console uninstaller starts, a dialog window opens:



   If you want to uninstall **Dr.Web for Unix Internet gateways**, enter **yes**, otherwise enter **no**. Press ENTER.

2. Review the list of components available for removal:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
Select the software you want to remove:
        [ ] 1 Dr.Web Agent (6.0.0.3)
        [ ] 2 Dr.Web Bases (6.0.0.3)
        [ ] 3 Boost, third party C++ libraries needed for Dr.Web (6.0.0.2)
        [ ] 4 Dr.Web Common Files (6.0.1.2)
        [ ] 5 Dr.Web Antivirus Daemon (6.0.1.3)
        [ ] 6 Dr.Web EPM - Library for X11 epm gui (6.0.0.1)
        [ ] 7 Dr.Web EPM - Dr.Web uninstaller (6.0.0.1)
        [ ] 8 Dr.Web ICAP DWS (6.0.0.2)
        [ ] 9 Dr.Web ICAP Daemon (6.0.0.2)
        [ ] 10 Dr.Web ICAPd Web Interface (6.0.0.2)
        [ ] 11 Dr.Web internet gateways documentation (6.0.0.2)
        [ ] 12 Essential third party libraries needed for Dr.Web on x86 systems
(6.0.0.5)
        [ ] 13 Dr.Web Monitor (6.0.0.3)
        [ ] 14 Dr.Web Antivirus Scanner (6.0.1.3)
        [ ] 15 Dr.Web Updater (6.0.0.4)

To select a package you want to remove or deselect some previously
selected package - enter the corresponding package number and press Enter.

You may enter A or All to select all the packages, and N or None to deselect all
 of them.
Enter R or Remove to remove selected packages.
Enter 0, Q or Quit to quit the dialog.
All values are case insensitive.
Select:
```

3. To select components to remove, follow the prompts .
4. To confirm you selection and start uninstallation, enter **Y** or **Yes** (they are case insensitive) and press ENTER:

```
user@hostname: ~
Файл  Правка  Вид  Поиск  Терминал  Справка
A list of packages marked for removal:
        drweb-agent
        drweb-bases
        drweb-boost144
        drweb-common
        drweb-daemon
        drweb-epm6.0.0-libs
        drweb-epm6.0.0-uninst
        drweb-icapd-dws
        drweb-icapd
        drweb-icapd-web
        drweb-internet-gateways-doc
        drweb-libs
        drweb-monitor
        drweb-scanner
        drweb-updater
Are you sure you want to remove the selected packages? (YES/no)
```

5. You can results of the uninstallation steps in the console in real time.
6. Once the process completes, exit setup.

Updating Distribution Package for UNIX Systems

Updating procedure combines installation and deinstallation procedures. To update **Dr.Web for Unix Internet gateways**, download the latest version of the corresponding software, remove the previous version and install the new one.

After an update, license key files, log files, and configuration files modified by the user are remained in the corresponding directories.

Installing from Native Packages

You can install **Dr.Web for Unix Internet gateways** from native packages for common **Linux** distributions or **FreeBSD** operating system.

All packages are located in the **Dr.Web** official repository http://officeshield.drweb.com/drweb/. Once you added the repository to the package manager of your system, you can install, update or remove necessary packages like any other program from repository. All dependencies are resolved automatically.

After installing packages from repository, automatic post-install script for installing license key file is not initiated. Licence key file must be manually copied to `%bin_dir`.

For the updates to take effect, you need to restart all **Dr.Web** services after updating from repository.

All the following commands to add repositories, import keys, install and remove packages must be run with administrator privileges (`root`).

If it is necessary, use the **`sudo`** or **`su`** commands.

Debian, Ubuntu (apt)

1. Installation:

Debian repository is signed with the digital key. It is necessary to import the key or correct operation. To do this, use the following command

```
wget -O - httwp://officeshield.drweb.com/drweb/drweb.key | apt-key add -
```

or

```
curl http://officeshield.drweb.com/drweb/drweb.key | apt-key add -
```

To add the repository to your system, add the following line to `/etc/apt/sources.list` file:

```
deb http://officeshield.drweb.com/drweb/debian stable non-free
```

To install **Dr.Web for Unix Internet gateways**, use the following commands:

```
apt-get update
apt-get install drweb-internet-gateways
```

2. Deinstallation:

To remove **Dr.Web for Unix Internet gateways**, use the following command:

```
apt-get remove drweb-internet-gateways
```

To remove all installed packages from **Dr.Web**, use the following command (in some systems, it is required to escape the '*' character with a backslash: '*'):

```
apt-get remove drweb*
```

To automatically remove unused packages from the system, use the following command:

```
apt-get autoremove
```

Removal with the use of **apt-get** has the following features:

1. The first variant of the command removes only the `drweb-internet-gateways` package, but other packages (which could be automatically installed on the package installation to resolve dependences) remain in the system.
2. The second variant of the command removes from the system all packages which names start with '`drweb`' (this is a standard pattern for a **Dr.Web** package name). Please note that this command removes from the system all packages which name corresponds to the pattern, not only those of **Dr.Web for Unix Internet gateways**.
3. The third variant of the command removes from the system all unused packages which were automatically installed for resolving dependences of some removed packages. Please note that this command removes all unused packages from the system, not only those of **Dr.Web for Unix Internet gateways**.

You can also use alternative package managers (for example, **Synaptic**, **aptitude**) to install or remove the packages. Moreover, it is recommended to use alternative managers, such as **aptitude**, to resolve a package conflict if it occurs.

ALT Linux, PCLinuxOS (apt-rpm)

1. Installation:

To add the repository to you system, add the following line to the `/etc/apt/sources.list` file:

32-bit version:

```
rpm http://officeshield.drweb.com/drweb/altlinux stable/i386 drweb
```

64-bit version:

```
rpm http://officeshield.drweb.com/drweb/altlinux stable/x86_64 drweb
```

To install **Dr.Web for Unix Internet gateways**:

```
apt-get update
apt-get install drweb-internet-gateways
```

2. Uninstallation:

In this case, uninstallation process is the same as for **Debian** and **Ubuntu** (see above).

You can also use alternative package managers (for example, **Synaptic**, **aptitude**) to install or remove the packages.

Mandriva (urpmi)

1. Installation:

Download a repository key from http://officeshield.drweb.com/drweb/drweb.key and save it to the disk. After that, import the key with the following command:

```
rpm --import <path to repository key>
```

Open the following file:

http://officeshield.drweb.com/drweb/drweb-i386.urpmi-media

or

http://officeshield.drweb.com/drweb/drweb-x86_64.urpmi-media

After you open a file, you will be offered to add a repository to the system.

Alternatively, you can add the repository via console using one of the following commands:

```
urpmi.addmedia drweb http://officeshield.drweb.com/drweb/mandriva/stable/i386/
```

or

```
urpmi.addmedia drweb http://officeshield.drweb.com/drweb/mandriva/stable/x86_64/
```

To install **Dr.Web for Unix Internet gateways**:

```
urpmi.update drweb
urpmi drweb-internet-gateways
```

2. Deinstallation:

To remove **Dr.Web for Unix Internet gateways**:

```
urpme drweb-internet-gateways
```

To automatically remove unused packages from the system:

```
urpme --auto-orphans drweb-internet-gateways
```

Removal with the use of **urpme** has the following features:

1. The first variant the command removes only the `drweb-internet-gateways` package, but other packages (which could be automatically installed on the package installation to resolve dependences) remain in the system.
2. The second variant of the command removes the `drweb-internet-gateways` package and all unused packages, which were automatically installed to resolve dependences of some removed packages. Please note that this command removes all unused packages from the system, not only those of **Dr.Web for Unix Internet gateways**.

You can also use alternative package managers (for example, **rpmdrake**) to install or remove the packages.

Red Hat Enterprise Linux, Fedora, CentOS (yum)

1. Installation:

Add to the `/etc/yum.repos.d` directory the file with following content:

32-bit version:

```
[drweb]
name=DrWeb - stable
baseurl=http://officeshield.drweb.com/drweb/el5/stable/i386/
gpgcheck=1
enabled=1
gpgkey=http://officeshield.drweb.com/drweb/drweb.key
```

64-bit version:

```
[drweb]
name=DrWeb - stable
baseurl=http://officeshield.drweb.com/drweb/el5/stable/x86_64/
gpgcheck=1
enabled=1
gpgkey=http://officeshield.drweb.com/drweb/drweb.key
```

To install **Dr.Web for Unix Internet gateways**:

```
yum install drweb-internet-gateways
```

2. Deinstallation:

To remove **Dr.Web for Unix Internet gateways**:

```
yum remove drweb-internet-gateways
```

To remove all installed packages from **Dr.Web**, use the following command (in some systems, it is required to escape the '*' character with a backslash: '*'):

```
yum remove drweb*
```

Removal with the use of **yum** has the following features:

1. The first variant of the command removes only the `drweb-internet-gateways` package, but other packages (which could be automatically installed on the package installation to resolve dependences) remain in the system.
2. The second variant of the command removes from the system all packages, names of which start with the '`drweb`' string (this is a standard pattern for a **Dr.Web** package name). Please note that this command removes from the system all packages which name corresponds to the pattern, not only those of **Dr.Web for Unix Internet gateways**.

You can also use alternative package managers (for example, **PackageKit**, **Yumex**) to install or remove the packages.

Zypper package manager (SUSE Linux)

1. Installation:

To add the repository, use the following command:

```
zypper ar -t YUM http://officeshield.drweb.com/drweb/el5/stable/i386/ drweb
```

or

```
zypper ar -t YUM http://officeshield.drweb.com/drweb/el5/stable/x86_64/ drweb
```

To install **Dr.Web for Unix Internet gateways**, use the following commands:

```
zypper refresh
zypper install drweb-internet-gateways
```

2. Deinstallation:

To remove **Dr.Web for Unix Internet gateways**, use the following command:

```
zypper remove drweb-internet-gateways
```

To remove all installed packages from **Dr.Web**, use the following command (in some systems, it is required to escape the '*' character with a backslash: '*'):

```
zypper remove drweb*
```

Removal with the use of **zypper** has the following features:

1. The first variant of the command removes only the `drweb-internet-gateways`, package but other packages (which could be automatically installed on the package installation to resolve dependences) remain in the system.
2. The second variant of the command removes from the system all packages, names of which start with the '`drweb`' string (this is a standard pattern for a **Dr.Web** package name). Please note that this command removes from the system all packages which name corresponds to the pattern, not only those of **Dr.Web for Unix Internet gateways**.

You can also use alternative package managers (or example, **YaST**) to install or remove the packages.

FreeBSD operating system

Installation:

You can install **Dr.Web** products from meta-ports for **FreeBSD**. Download the `drweb-internet-gateways_current-current~freebsd_all.tar.gz` archive from http://officeshield.drweb.com/drweb/freebsd/ports/. After that, unpack the archive and use the `make install` command to compile and install **Dr.Web for Unix Internet gateways**. If you install **Dr.Web for Unix Internet gateways** in **FreeBSD** 6.1, specify the path to the `/usr/ports/Mk/` directory using the `-I` parameter. That directory contains the ports tree.

Example:

```
tar -xzvf drweb-internet-gateways_current-current~freebsd_all.tar.gz
make install -I /usr/ports/Mk/
```

Starting Dr.Web for Unix Internet gateways

This section describes startup of **Dr.Web for Unix Internet gateways** in **Linux**, **Solaris** or **FreeBSD** operating systems.

For Linux and Solaris OS

To run **Dr.Web for Unix Internet gateways**:

1. Register the software.
2. Copy or move the key file to the directory with **Dr.Web for Unix Internet gateways** executable files (the default directory for UNIX systems is `%bin_dir`). Name of the key file can be different in different distribution packages (for details, see Software Registration):
   - If **Dr.Web for Unix Internet gateways** was purchased as a standalone product, license key file is named `drweb32.key`. In this case, copy the file to the `%bin_dir` directory without changing its name.
   - If **Dr.Web for Unix Internet gateways** was purchased as a part of **Dr.Web Enterprise Security Suite**, archive received during registration contains a key file for **Dr.Web Enterprise Server** (`enterprise.key`) and a key file for workstations (`agent.key`). Rename `agent.key` as `drweb32.key` and copy it to the `%bin_dir` directory.

   To use a key file from a different location or with another name (for example, `agent.key`), specify its full path as a `Key` parameter value in the `drweb32.ini` configuration file. In the `Standalone` mode, alternative path to the key file must be specified as a value of the **`LicenseFile`** parameter in `agent.conf` (a configuration file of **Dr.Web Agent**).
3. Configure the software by making necessary changes to the configuration files. For details on configuration parameters, see the corresponding sections of this Manual.
4. Set `1` as a value of the `ENABLE` variable in the `drwebd.enable` file to run **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** on the local machine (properly configured **Dr.Web Daemon** is working on another local network computer), the value of the ENABLE variable must be 0 (its default value).
5. Set `1` as a value of the `ENABLE` variable in the `drweb-monitor.enable` file to run **Dr.Web Monitor**.

Location of the `enable` files depends on **Dr.Web for Unix Internet gateways** installation type:

- Installation from **universal package for UNIX systems**:
  Files are saved to the `%etc_dir` directory and named as follows
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation from **native DEB packages**:
  Files are saved to the `/etc/defaults` directory and named as follows
  `drweb-icapd`,
  `drwebd`,
  `drweb-monitor`.
- Installation from **native RPM packages**:
  Files are saved to the `/etc/sysconfig` directory and named as follows
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.

---

6. Run **Dr.Web Daemon** and **Dr.Web Monitor** either from the console or a file manager of your operation system. After startup, **Dr.Web Monitor** starts all other **Dr.Web for Unix Internet gateways** components.

   In case of installation from native packages in Solaris:

   During **Dr.Web for Unix Internet gateways** installation, the SMF service management system attempts to run **Dr.Web Monitor**. If **Dr.Web Monitor** cannot find a licence key file (for example, on the first installation of **Dr.Web for Unix Internet gateways**), it stops its operation and SMF goes into the maintenance state.

   To run **Dr.Web Monitor**, reset the maintenance state:

   - Enter the following command

     ```
     # svcs -p <FMRI>
     ```

     where FMRI is a unique identifier of a controlled resource. In this case, a unique identifier of **Dr.Web Monitor** is required.

   - Force termination of the process from `svcs -p` output list.

     ```
     # pkill -9 <PID>
     ```

     where PID is a number of the process listed above.

   - Restart **Dr.Web Monitor** with the following command:

     ```
     # svcadm clear <FMRI>
     ```

   While installing **Dr.Web for Unix Internet gateways** from native packages in Solaris, run **Dr.Web for Unix Internet gateways** with the SMF service management system:

   ```
   # svcadm enable <drweb-monitor>
   # svcadm enable <drweb-daemon>
   # svcadm enable <drweb-icapd>
   ```

   To stop the service:

   ```
   # svcadm disable <service_name>
   ```

7. Run a proxy server.

The `drwebd` and `drweb-icapd` modules can be launched in one of the following two modes:

1. with the `init` script (standard launch)
2. with the **Dr.Web Monitor**

In the second mode, set the `ENABLE` parameter to 0 in the `enable` file.

Each of the components can be run independently as well, but note that **Dr.Web Agent** must be started first since all other modules receive configuration from **Dr.Web Agent**.

For FreeBSD OS

To run **Dr.Web for Unix Internet gateways**:

1. Register the software.
2. Copy or move the key file (with the `.key` extension) to the directory with **Dr.Web for Unix Internet gateways** executable files (the default directory for UNIX systems is `%bin_dir`). Name of the key file can differ in different distribution packages (for details, see Software Registration):
   - If **Dr.Web for Unix Internet gateways** was purchased as a standalone product, license key file is named `drweb32.key`. In this case, copy the file to the `%bin_dir` directory without

changing its name.

- If **Dr.Web for Unix Internet gateways** was purchased as a part of **Dr.Web Enterprise Security Suite**, archive received during registration contains a key file for **Dr.Web Enterprise Server** (`enterprise.key`) and a key file for workstations (`agent.key`). Rename `agent.key` to `drweb32.key` and copy the file to the `%bin_dir` directory.

To use a key file from a different location or with another name (for example, `agent.key`), specify its full path as a **`Key`** parameter value in the `drweb32.ini` configuration file. In the `Standalone` mode, alternative path to the key file must be specified as a value of the **`LicenseFile`** parameter in `agent.conf` (a configuration file of **Dr.Web Agent**).

3. Configure the software by making necessary changes to the configuration files. For details on configuration parameters, see the corresponding sections of this Manual.
4. Add the following lines to the `/etc/rc.conf` file:
   - `drwebd_enable="YES"` – to run **Dr.Web Daemon**. If it is not required to run **Dr.Web Daemon** on the local machine (properly configured **Dr.Web Daemon** is working on another local network computer), then you do not need to add the line to the `rc.conf` file;
   - `drweb_monitor_enable="YES"` – to run **Dr.Web Monitor.**
5. Run **Dr.Web Daemon** and **Dr.Web Monitor** either from the console or from a file manager of your operation system. After startup, **Dr.Web Monitor** starts all other **Dr.Web for Unix Internet gateways** components.
6. Run a proxy server.

Each of the components can be run independently as well, but note that **Dr.Web Agent** must be started first since all other modules receive their configuration from **Dr.Web Agent**.

Configuring SeLinux Security Policies

If the used **Linux** distribution features **SELinux** security subsystem (`Security-Enhanced Linux`), you need to configure security policies used by **SELinux** in order to enable correct operation of anti-virus components (**Dr.Web Daemon** and **Dr.Web Console Scanner**) after the installation.

Moreover, if **SELinux** is enabled, product installation from distribution packages (`.run`) can fail because an attempt to create `drweb` user, whose privileges are used by **Dr.Web for Unix Internet gateways**, will be blocked.

Thus, before installing the product, check **SELinux** operation mode with the use of **`getenforce`** command. This command outputs the current operation mode which can be one of the following:

- **`Permissive`** – protection is active, but permissions are supported: actions that violate the security are not denied but logged.
- **`Enforced`** – protection is active and restrictions are enforced: actions that violate the security are logged and blocked.
- **`Disabled`** – **SELinux** is installed but not active.

If **SELinux** is operating in the `Enforced` mode, temporarily (until the product is installed and security policies are configured) enable `Permissive` mode. To do this, enter the **`setenforce 0`** command that temporarily (until the next restart) sets **SELinux** operation mode to `Permissive`. To enable the `Enforced` mode again, enter the **`setenforce 1`** command.

Note that regardless of the mode enabled with the **`setenforce`** command, after system restart **SELinux** will operate in the mode specified in the settings (normally, **SELinux** configuration file is located in the `/etc/selinux` directory).

In general, if **`audit`** daemon is used, the log file resides in `/var/log/audit/audit.log`.

Otherwise, notifications on forbidden actions are logged to the following log file: `/var/log/messages`.

For correct operation of anti-virus components when **SELinux** is enabled, compile special security policies once the product installation completes.

Please note that some Linux distributions may not have the below mentioned utilities installed by default. In this case you need to additionally install the required packages.

To create required policies:

1. Create a new file with **SELinux** policy source code (.te file). The file defines restrictions applied to the described module. The source file can be created in one of the two ways:
   1) **With the use of `audit2allow` utility.** This way is more simple. The utility generates permissive rules based on the messages on denial of access to system log files. You can set automatic search of messages in log files or set path to the log file manually.

      `audit2allow` utility resides in the `policycoreutils-python` package, or `policycoreutils-devel` package (for **RedHat Enterprise Linux, CentOS, Fedora** OS, depending on the version), or `python-sepolgen` package (for **Debian, Ubuntu** OS).

      Example usage:

      ```
      # audit2allow -M drweb -i /var/log/audit/audit.log
      ```

      OR

      ```
      # cat /var/log/audit/audit.log | audit2allow -M drweb
      ```

      In this example, **`audit2allow`** utility searches for access denied messages in the `audit.log` file.

      ```
      # audit2allow -a -M drweb
      ```

      In this example, **`audit2allow`** searches for access denied messages in log files automatically.

      In both cases two files are created as a result of the utility operation: `drweb.te` policy source file and `drweb.pp` policy module which is ready for installation.

      In most cases you do not need to adjust policies created by the utility. So, it is recommended to go to step 4 for installation of the `drweb.pp` policy module. Note that **`audit2allow`** utility outputs **`semodule`** command invocation string. Copy the string to the command line and execute. That way, you will do instructions of step 4. Go to step 2 only if you want to adjust the policies which are automatically formed for **Dr.Web for Unix Internet gateways** components.

   2) **With the use of `policygentool` utility.** As a parameter, specify the name of the module which operation you want to configure and the path to its executable file.

      Note that **`policygentool`** utility included in `selinux-policy` package for **RedHat Enterprise Linux** and **CentOS Linux** OS might not function correctly. In this case, use **`audit2allow`** utility.

      Example of creating policies with policygentool:

      - For **Dr.Web Console Scanner**:

        ```
        # policygentool drweb-scanner /opt/drweb/drweb.real
        ```

- For **Dr.Web Daemon**:

```
# policygentool drweb-daemon /opt/drweb/drwebd.real
```

You will be prompted to get information on some domain features and then for each of the modules, 3 files will be created which determine the policy:

`[module_name].te`, `[module_name].fc` и `[module_name].if`.

2. If necessary, edit generated source file of the `[module_name].te` policy and then use the **checkmodule** utility to create a binary representation (`.mod`) of the policy source file.

Please note that for successful policy compilation, a `checkpolicy` package must be installed in the system.

Usage example:

```
# checkmodule -M -m -o drweb.mod drweb.te
```

3. Create a policy module (`drweb.pp`) with the use of **semodule_package** utility.

Example:

```
# semodule_package -o drweb.pp -m drweb.mod
```

4. To install a new policy module into the module store, use the **semodule** utility.

Example:

```
# semodule -i drweb.pp
```

After system restart, **SELinux** security subsystem will be configured to enable correct operation of **Dr.Web for Unix Internet gateways**.

For details on how to configure **SELinux** and on its operation features, refer to documentation for the used **Linux** distribution.

Registration Procedure

Permissions to use **Dr.Web for Unix Internet gateways** are specified in the key file.

License key file contains the following information:

- list of **Dr.Web for Unix Internet gateways** components licensed to the user;
- license period;
- other restrictions (for example, number of protected workstations).

By default, the license key file is located in the directory with **Dr.Web for Unix Internet gateways** executables.

License key file is digitally signed to prevent its editing. Edited key file becomes invalid. It is not recommended to open your key file in text editors in order to avoid its accidental invalidation.

Users who have purchased **Dr.Web for Unix Internet gateways** from **Doctor Web** certified partners obtain the license key file. Key files contain the following information which depends on the license type. The license key file also contains information on the user and seller of the product.

For evaluation purposes users may also obtain a demo key file. It allows them to enjoy full functionality of the **Dr.Web for Unix Internet gateways** solution, but has a limited term of use, and no technical support is provided.

License key file can be supplied as:

- a `drweb32.key` file license key for workstations, or as a zip archive containing a license key file in case of purchasing **Dr.Web for Unix Internet gateways** as a standalone product;
- a zip-archive, which contains a key file for **Dr.Web Enterprise Server** `(enterprise.key)` and a key file for workstations `(agent.key)` in case of purchasing **Dr.Web for Unix Internet gateways** as a part of **Dr.Web Enterprise Security Suite**.

License key file can be received in one of the following ways:

- by email as a ZIP-archive containing license key file with `*.key` extension (usually after registration on the website). Extract the license key file using an appropriate archiving utility and copy (or move) it to the directory with **Dr.Web for Unix Internet gateways** executable files (default directory for UNIX systems is `%bin_dir`);
- within the distribution package;
- on a separate data carrier as a file with `*.key` extension. In this case, a user must copy it manually to the `%bin_dir` directory.

License key file is sent to a user via email usually after registration on the website (website location is specified in the registration card supplied with the product). Visit the website, fill in the web form with your customer data and submit your registration serial number (printed on the registration card). After that, your license is activated and a key file is created according to the specified serial number. The key file is sent to the specified email address.

It is recommended to keep the license key file until it expires, and use it to reinstall or restore **Dr.Web for Unix Internet gateways**. If the license key file is damaged or lost, it can be recovered by the same procedure as during license activation. In this case, you must use the same product serial number and customer data that you provided during the registration; only the email address can be changed (in this case, a license key file will be sent to the new email address). If the serial number matches any entry in **Dr.Web for Unix Internet gateways** database, the corresponding key file will be automatically dispatched to the specified email address.

One serial number can be registered no more than 25 times. If you need to recover a lost license key file after its 25th registration, send a request for license key file recovery at http://support.drweb.com/

request/ stating the data input during registration, valid email address, and detailed description of your problem. The request will be considered by **Dr.Web for Unix Internet gateways** technical support service engineers. If the request is approved, a license key file will be provided via automatic support system or dispatched via email.

Path to a license key file of the certain component must be specified as a **Key** parameter value in the corresponding configuration file (drweb32.ini).

Example:

```
Key = %bin_dir/drweb32.key
```

If a license key file specified as a **Key** parameter value failed to be read (wrong path, permission denied) or is expired, blocked or invalid, the corresponding component terminates its operation.

If the license expires in less than two weeks, **Dr.Web Scanner** outputs a warning message on its startup and **Dr.Web Daemon** notifies the user via email. Messages are sent on every startup, restart or reload of **Dr.Web Daemon** for every license key file installed. To enable this option, set up the **MailCommand** parameter in the [Daemon] section of the drweb32.ini configuration file.

If you want to use a key file from another location, specify the full path to it as a **LicenseFile** parameter value in the [StandaloneMode] section of the **Dr.Web Agent** configuration file (see [StandaloneMode] section description).

Dr.Web Updater

You can use **Dr.Web Updater** to enable automatic updates of virus databases and content-specific black and white lists of Internet resources for **Dr.Web for Unix Internet gateways**. **Dr.Web Updater** is implemented as a console script `update.pl` written in **Perl**, and you can find the module in the directory with **Dr.Web for Unix Internet gateways** executable files.

Dr.Web Updater requires installed **Perl** 5.8.0 or later.

Dr.Web Updater settings are located in the `[Updater]` section of the `drweb32.ini` configuration file in `%etc_dir` directory. To use an alternative configuration file, specify the full path to it with a command line parameter on the startup.

To run the script, use the following command:

```
$ %bin_dir/update.pl [parameters]
```

For details on allowed parameters, see Command Line Parameters.

In the standard mode, updates are downloaded and installed automatically under the `drweb` user.

Do not start updating under the `root` superuser as this results in changing the ownership of updated files to `root` superuser and may cause an error on attempt to update them automatically in the future.

Updating Anti-Virus and Virus Databases

To provide reliable protection, **Dr.Web for Unix Internet gateways** requires regular updates to virus databases.

Dr.Web for Unix Internet gateways virus databases are stored as files with the *.`vdb` extension. Update servers of **Dr.Web Global Updating System** (**Dr.Web GUS**) can also store them within lzma-archives. When new viruses are discovered, small files (only several KBytes in size) with database segments describing these viruses are released to provide quick and effective countermeasures.

Updates are the same for all supported platforms. There are daily "hot" updates (`drwtoday.vdb`) and regular weekly updates (`drwXXXYY.vdb`), where `XXX` is a version number of an anti-virus engine, and YY is a sequential number, starting with 00 (for example, the first regular update for version 6.0 is named `drw60000.vdb`).

"Hot" updates are issued daily or even several times a day to provide effective protection against new viruses. These updates are installed over the old ones: that is, a previous `drwtoday.vdb` file is overwritten. When a new regular update is released, all records from `drwtoday.vdb` are copied to `drwXXXYY.vdb`, and a new empty `drwtoday.vdb` file is issued.

If you want to update virus databases manually, you must install all missing regular updates first, and then overwrite `drwtoday.vdb` file.

To add an update to the main virus databases, place the corresponding file to the directory with **Dr.Web for Unix Internet gateways** executable files (`/var/drweb/bases/` by default) or to any other directory specified in the configuration file.

Signatures for virus-like malicious programs (adware, dialers, hacktools and others) are supplied in two additional files - `drwrisky.vdb` and `drwnasty.vdb` - with the structure similar to virus databases. These files are also regularly updated: `dwrXXYYY.vdb` and `dwnXXYYY.vdb` are for regular updates, and `dwrtoday.vdb` and `dwntoday.vdb` are for "hot" updates.

From time to time (as new anti-virus techniques are developed), new versions of the anti-virus package are released, containing the updated algorithms, implemented in the anti-virus engine **Dr.Web Engine**. At the same time, all released updates are brought together, and the new package version is completed with the updated main virus databases with descriptions of all known viruses. Usually after an upgrade of a package version, new databases can be linked to the old **Dr.Web Engine**. Please note that this does not guarantee detection or curing of new viruses, as it requires upgrading of algorithms in **Dr.Web Engine.**

Being regularly updated, virus databases have the following structure:

- `drwebase.vdb` – general virus database, received with the new version of the package;
- `drwXXXYY.vdb` – regular weekly updates;
- `drwtoday.vdb` – "hot" updates released daily or several times a day;
- `drwnasty.vdb` – general database of other malware, received with the new version of the package;
- `dwnXXXYY.vdb` – regular weekly updates for other malware;
- `dwntoday.vdb` – "hot" updates for other malware;
- `drwrisky.vdb` – general database of riskware, received with the new version of the package;
- `dwrXXXYY.vdb` – regular weekly updates for riskware;
- `dwrtoday.vdb` – "hot" updates for riskware.

Virus databases can be automatically updated with **Dr.Web Updater** module (`%bin_dir/update.pl`). After installation, a user crontab file (`/etc/cron.d/drweb-update`) is automatically created to run **Updater** every 30 minutes. That ensures regular updates and maximum protection. You can modify this file to change update period.

Black and white lists of Internet resources are stored as files with the `dws` extension:

- `dwfXXXNN.dws` – black list, where `XXX` stands for an abbreviation of the main content topic, and `NN` is the ordinal number of the list;
- `white_dwfXXX.dws` – white list, where `XXX` stands for an abbreviation of the main content topic.

Update servers of **Dr.Web GUS** may also store these files in lzma-archives. If you do not want to update these lists, delete or move `icapd.drl` file from the directory that contains `drl`-files (path to this directory is specified as the **`DrlDir`** parameter value in the `[Updater]` section of `drweb32.ini` configuration file).

Cron Configuration

For Linux: a special file with user settings is created in the `/etc/cron.d/` directory during installation of the software. It enables interaction between **`cron`** and **Dr.Web Updater**.

In the task created for **`crond`**, the vixie cron syntax is used. If you use a different **`cron`** daemon, such as **`dcron`**, create a task to start **Dr.Web Updater** automatically.

For FreeBSD and Solaris: manual configuration of **`cron`** is required to enable its interaction with **Dr.Web Updater**.

For example, when you use **FreeBSD** you may add the following string to **`crontab`** of `drweb` user:

```
*/30 * * * * /usr/local/drweb/update.pl
```

If you work with **Solaris**, the following set of commands is used:

```
# crontab -e drweb
# 0,30 * * * * /opt/drweb/update.pl
```

Please note that by default the **cron** daemon launches **Dr.Web Updater** once in 30 minutes (at the 0 and 30 minutes of every hour). This may result in increased load on the **Dr.Web GUS** update servers and cause update delays. To avoid such situation, it is recomended to change default values to arbitrary.

Command Line Parameters

- `--help` – shows brief help.
- `--ini` – specifies another (not default) configuration file to be used. To use another configuration file, specify the full path to it with the `--ini` command line parameter. If the name of the configuration file is not specified, `%etc_dir/drweb32.ini` is used.

  Example:

  ```
  $ /opt/drweb/update.pl --ini=/path/to/conf_file
  ```

- `--what` – temporarily overrides value of the **Section** parameter on **Updater** startup. The new specified value is used until next start of the script. Possible values: `scanner` or `daemon`.

  Example:

  ```
  $ /opt/drweb/update.pl --what=Scanner
  ```

- `--components` – displays a list of all product components available for update.

  Example:

  ```
  $ /opt/drweb/update.pl --components
  ```

- You can also use the command line parameter `--not-need-reload`:
  - if this parameter is not specified, all daemons (**Dr.Web Daemon** for **Dr.Web for Unix Internet gateways**) which components were updated, removed, or added are restarted after **update.pl** script finishes;
  - if the `--not-need-reload` parameter is specified without any value, after the **update.pl** script finishes no daemon of **Dr.Web for Unix Internet gateways** is restarted;
  - if some daemon names are specified as the `not-need-restart` value, the corresponding daemons are not restarted after the **update.pl** script finishes. Names of non-restarted daemons must be separated be commas and listed without white spaces. The names are case insensitive.

  Example:

  ```
  $ /opt/drweb/update.pl --not-need-reload=drwebd
  ```

Blocking Updates for Selected Components

You can configure **Dr.Web Updater** to block updates to selected components of your **Dr.Web for Unix Internet gateways**.

To view the list of available components, use the `--components` command line parameter:

Example:

```
# ./update.pl --components

Available Components:
    agent
    drweb          (frozen)
    icapd          (frozen)
    vaderetro_lib
```

If updates to a component are blocked, that component is marked as *frozen*. Frozen components are not updated when **Dr.Web Updater** is started.

Blocking updates

To block updates for specific component, use the `--freeze=<components>` command-line parameter, where `<components>` is a comma separated list of components to be frozen.

Example:

```
# ./update.pl --freeze=drweb
Updates for component 'drweb' are frozen.
Run command './updater --unfreeze=drweb' to start updates again.
```

Unblocking updates

To enable updates for a frozen component, use the `--unfreeze=<components>` command-line parameter, where `<components>` is a comma separated list of components to be unfrozen.

Example:

```
# ./update.pl --unfreeze=drweb
Updates for component 'drweb' are no longer frozen.
```

Unfreezing will not update the component.

Restoring Components

When **Dr.Web for Unix Internet gateways** components are being updated, **Dr.Web Updater** saves their back-up copies to the working directory. It enables you to restore any component to its previous state if any problem occurs during an update.

To restore component to its previous state, use the `--restore=<components>` command line parameter, where `<components>` is a comma separated list of components to be restored.

Example:

```
# ./update.pl --restore=drweb

Restoring backup for component 'drweb'...
Updates for component 'drweb' are frozen.
Run command './updater --unfreeze=drweb' to start updates again.

Backup for component 'drweb' has been restored!
Dr.Web (R) restore details:

Following files has been restored:
     /var/drweb/bases/drwtoday.vdb
     /var/drweb/bases/dwntoday.vdb
     /var/drweb/bases/dwrtoday.vdb
     /var/drweb/bases/timestamp
     /var/drweb/updates/timestamp
```

Restored components are automatically frozen. To enable updates for a restored component, unfreeze it.

Configuration

Dr.Web Updater settings are stored in the `Updater` section of the configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory:

`Section [Updater]`

UpdatePluginsOnly = {logical}	If `Yes` value is specified, **Dr.Web Updater** does not update **Dr.Web Daemon** and **Dr.Web Scanner**. It updates only the plug-ins. Default value: **UpdatePluginsOnly** = No
Section = {Daemon | Scanner}	Specifies the section of configuration file where **Dr.Web Updater** takes the settings, such as a path to the key file, paths to virus databases and others. Possible values: `Scanner, Daemon`. Value of this parameter can be temporarily overriden by the `--what` command line parameter. The specified value is used until the next start of the script. Default value: **Section** = Daemon
ProgramPath = {path to file}	Path to the executable file of **Dr.Web Daemon** or **Dr.Web Scanner**. It is used by **Dr.Web Updater** to get the product version. Default value: **ProgramPath** = %bin_dir/drwebd
SignedReader = {path to file}	Path to the program which is used to read digitally signed files. Default value: **SignedReader** = %bin_dir/read_signed
LzmaDecoderPath = {path to directory}	Path to the directory that contains a program used for unpacking of lzma-archives. Default value: **LzmaDecoderPath** = %bin_dir/

LockFile = {path to file}	Path to the file used to prevent sharing of certain files during their processing by **Dr.Web Updater**. Default value: **LockFile** = %var_dir/run/update.lock
CronSummary = {logical}	If you specify Yes, **Dr.Web Updater** outputs an update report for each session to stdout. This mode can be used to send notifications to administrator by email, if **Dr.Web Updater** is run by the cron daemon. Default value: **CronSummary** = Yes
DrlFile = {path to file}	Path to the file (*.drl) with the list of **Dr.Web GUS** servers. **Dr.Web Updater** selects a server from this list in random order to download updates. For details on downloading updates, see Updating Process. This file is signed by **Doctor Web** and must not be modified by a user. The file is updated automatically. Default value: **DrlFile** = %var_dir/bases/update.drl
CustomDrlFile = {path to file}	Path to the file (*.drl) with the alternative list of **Dr.Web GUS** servers. **Dr.Web Updater** also selects a server from this list in random order to download updates. For details on downloading updates, see Updating Process. This file is signed by **Doctor Web** and must not be modified by a user. It is updated automatically. Default value: **CustomDrlFile** = %var_dir/bases/custom.drl
FallbackToDrl = {logical}	Allows using the file specified by **DrlFile** when connection to one of the servers listed in **CustomDrlFile** failed. If the parameter value is No, the file specified in **DrlFile** is not used. If the file specified in **CustomDrlFile** does not exist, the file specified in **DrlFile** is used regardless of the **FallbackToDrl** parameter value. For details on downloading updates, see Updating Process. Default value: **FallbackToDrl** = Yes
DrlDir = {path to directory}	Path to the directory that contains drl files with lists of **Dr.Web GUS** servers for each plug-in. These files are signed by **Doctor Web** and must not be modified by a user. Default value: **DrlDir** = %var_dir/drl/

Timeout = {numerical value}	Maximum wait time for downloading updates from the selected **Dr.Web GUS** server, in seconds. Default value: **Timeout** = 90
Tries = {numerical value}	Number of attempts by **Dr.Web Updater** to establish connection with the selected update server. Default value: **Tries** = 3
ProxyServer = {host name | IP address}	Host name or IP address of the proxy server which is used for Internet access. If the proxy server is not used, the value of this parameter must be empty. Default value: **ProxyServer** =
ProxyLogin = {string}	User login to access the used proxy server (if it requires authentication). Default value: **ProxyLogin** =
ProxyPassword = {string}	The password to access the used proxy server (if it requires authentication). Default value: **ProxyPassword** =
LogFileName = {syslog | file name}	Path to the log file name. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label which is used by `syslogd` system service. Default value: **SyslogFacility** = Daemon
LogLevel = {log level}	Log verbosity level. The following levels are allowed: • Quiet • Error • Warning • Info • Debug • Verbose Default value: **LogLevel** = Info

IcapdPidFile = {path to file}	Path to **Dr.Web ICAPD** PID file. Default value: **IcapdPidFile** = %var_dir/run/drweb_icapd.pid
BlacklistPath = {path to directory}	Path to the directory with .dws files. Default value: **BlacklistPath** = %var_dir/dws
AgentConfPath = {path to file}	Path to **Dr.Web Agent** configuration file. Default value: **AgentConfPath** = %var_dir/agent.conf
ExpiredTimeLimit = {numerical value}	Number of days left before license expiration during which **Dr.Web Updater** is attempting to update license key file. Default value: **ExpiredTimeLimit** = 14
ESLockfile = {path to file}	Path to the lock file. If the lock file exists, **Dr.Web Updater** can not be automatically initialized by cron daemon. Default value: **ESLockfile** = %var_dir/run/es_updater.lock

Updating Procedure

Updating is performed in the following stages:

1. **Dr.Web Updater** reads the configuration file (drweb32.ini by default, or specified with the --ini command line argument).
2. **Dr.Web Updater** uses parameters from the [Updater] section of the configuration file (see the description above) as well as the following parameters: **EnginePath**, **VirusBase**, **UpdatePath** and **PidFile**.
3. **Dr.Web Updater** selects **Dr.Web GUS** server for downloading updates. The server is selected in the following way:
   - Reading of the files which contain lists of update servers. The filenames are specified in the **DrlFile** and **CustomDrlFile** parameters;
   - If both files are not accessible, updating process stops and terminates;
   - If only one of the files is accessible, it is used regardless of the value specified for the **FallbackToDrl** parameter;
   - If both files are accessible, **Dr.Web Updater** uses the file specified in the **CustomDrlFile** parameter;
   - If it is impossible to connect to any of the servers from this file (specified in **CustomDrlFile**), and the **FallbackToDrl** value is set to Yes, **Dr.Web Updater** tries to establish connection with the servers from the file specified in the **DrlFile** parameter. If the connection fails, the updating process stops and terminates.
4. **Dr.Web Updater** tries to connect to servers from the selected file in random order until connection is established (**Dr.Web Updater** waits for the server to respond during the period specified in the **Timeout** parameter).

5. **Dr.Web Updater** requests the list of available updates from the selected **Dr.Web GUS** server and then requests the corresponding lzma archives. If the archives are not available on the server, the updates are downloaded as `vdb` files. To unpack lzma-archives, **`lzma`** utility is used. Path to the directory with the utility is specified in the **`LzmaDecoderPath`** parameter.
6. After updates are unpacked, they are saved to the corresponding directories as described in Updating.

Dr.Web Agent

Dr.Web Agent is a resident module used to manage settings of **Dr.Web for Unix Internet gateways** modules, define anti-virus policy depending on available licenses and collect virus statistics. Statistics, depending on **Dr.Web Agent** operational mode, is sent with the predetermined frequency either to the public server of the company or to the central protection server that works under **Dr.Web Agent**. When **Dr.Web for Unix Internet gateways** modules are started or settings are changed, **Dr.Web Agent** sends all necessary configuration to these modules.

Note that **drweb-agent** can operate in enterprise mode only with **Dr.Web ESS** 6. If you want to ensure connection to the central protection server **Dr.Web ESS** 10, install and configure the new agent version, implemented as **drweb-agent10** module. For details on how to install and configure **drweb-agent10**, refer to the Migration to Dr.Web ESS 10 section.

Dr.Web Agent can interact with other modules through exchanging control signals.

Since all **Dr.Web for Unix Internet gateways** components (except for **Dr.Web Monitor**) receive their configuration via **drweb-agent** module, it must be run before all these modules, but after the **drweb-monitor** module.

Please note that when several parameters with the same name are specified in the configuration file, **Dr.Web Agent** unites them in one comma delimited string. You can also use a backslash symbol "\" to define parameter value in several lines. New line after backslash is added to the previous line when **Dr.Web Agent** is reading configuration. Note that using of a space character after a slash is not allowed.

Operation Mode

If necessary, **Doctor Web** can be connected to a corporate or private anti-virus network managed by **Dr.Web Enterprise Security Suite** (**Dr.Web ESS**). To operate in the central protection mode, you do not need to install additional software or uninstall your **Dr.Web** solution.

To provide you with this option, **Dr.Web Agent** can operate in one of the two following modes:

- **Standalone mode** when a protected computer is not included in an anti-virus network or managed remotely. In this mode, configuration files and key files reside on local drives, and **Dr.Web Agent** is fully controlled from the protected computer.
- **Enterprise mode** (or central protection mode), when protection of the computer is managed from the central protection server. In this mode, some features and settings of **Dr.Web for Unix Internet gateways** may be modified and blocked for compliance with a general (for example, company) security policy. Licence key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used.

Note that **drweb-agent** can operate in enterprise mode only with **Dr.Web ESS** 6. If you want to ensure connection to the central protection server **Dr.Web ESS** 10, install and configure the new agent version, implemented as **drweb-agent10** module. For details on how to install and configure **drweb-agent10**, refer to the Migration to Dr.Web ESS 10 section.

To use central protection mode

1. Contact the anti-virus network administrator of your company for a public key file and parameters of connection to the central protection server.
2. In the **Dr.Web Agent** configuration file (by default, `%etc_dir/agent.conf`), adjust the following parameters in the `[EnterpriseMode]` section:

- Set the **PublicKeyFile** parameter value to location of a public key file received from anti-virus network administrator (usually, `%var_dir/drwcsd.pub`).This file includes an encryption public key for access to **Dr.Web ESS**. If you are the anti-virus network administrator, you can locate the file in the corresponding directory on the **Enterprise Server**.
- Set the **ServerHost** parameter value to the IP-address or host name of the **Enterprise Server**.
- Set the **ServerPort** parameter value to the **Enterprise Server** port number.

3. To connect to the central protection server, set the **UserEnterpriseMode** parameter value to `Yes`.

   In the central protection mode, some features and settings of **Dr.Web for Unix Internet gateways** may be modified and blocked in compliance with the general security policy. A key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used.

> To run **Dr.Web Agent** in the central protection mode, `drweb-agent-es` package must be installed.
>
> To enable **Dr.Web for Unix Internet gateways** to fully support the central protection mode, set **Dr.Web Monitor** to operate in enterprise mode. For more details, see Operation Mode of **Dr.Web Monitor**.

To use standalone mode

1. Ensure that all parameters in the `[StandaloneMode]` section of the **Dr.Web Agent** configuration file (by default, `%etc_dir/agent.conf`) are adjusted properly.
2. In the `[EnterpriseMode]` section of the **Dr.Web Agent** configuration file, set the **UseEnterpriseMode** parameter to `No`.

   When switching to this mode, all settings of **Dr.Web for Unix Internet gateways** are unlocked and restored to their previous or default values. You can access all features of **Dr.Web for Unix Internet gateways** solutions again and configure them.

> For correct operation in the standalone mode, **Dr.Web for Unix Internet gateways** requires a valid personal key file. The key files received from the central protection server cannot be used in this mode.

Using Dr.Web for Unix Internet gateways and Dr.Web Anti-virus for Linux together in the central protection mode

Because of the implementation features, **Dr.Web for Unix Internet gateways** and **Dr.Web Anti-virus for Linux** cannot be simultaneously operate in the central protection mode if they are both installed on the same computer. To enable **Dr.Web for Unix Internet gateways** to operate in the central protection mode, change the operation mode of **Dr.Web Anti-virus for Linux** to the Standalone mode and delete or move to another directory the following files: `%etc_dir/agent/drweb-cc.amc` and `%etc_dir/agent/drweb-spider.amc`.

If you want to switch **Dr.Web Anti-virus for Linux** back to the central protection mode later, we recommended to save the files as a back up copy in a directory that is different from `%etc_dir/agent`. In this case, disable the central protection mode of **Dr.Web for Unix Internet gateways**, copy back up copies of `drweb-cc.amc` and `drweb-spider.amc` files to the `%etc_dir/agent/` directory and follow the instructions provided in the **Dr.Web Anti-virus for Linux** User Manual.

Command Line Parameters

To run **Dr.Web Agent**, use the following command:

```
drweb-agent [parameters]
```

where the following `parameters` are available:

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and terminate the module		
`-v`	`--version`	
Description: Show **Dr.Web Agent** version on the screen and terminate the module		
`-u`	`--update-all`	
Description: Start updating all **Dr.Web for Unix Internet gateways** components		
`-f`	`--update-failed`	
Description: Start updating **Dr.Web for Unix Internet gateways** components, updating of which failed in the standard mode		
`-C`	`--check-only`	
Description: Check correctness of **Dr.Web Agent** configuration. This parameter cannot be used if a **Dr.Web Agent** process is already running in the system		
`-c`	`--conf`	`<path to file>`
Description: Enable the module to use the specified configuration file		
`-d`	`--droppwd`	
Description: Discard registration data required to access **Dr.Web Enterprise Server** (username, password). At the next connection attempt, a new process of workstation registration will start.		
`-p`	`--newpwd`	
Description: Change username and password required to access **Dr.Web Enterprise Server**		
`-s`	`--socket`	`<path to file>`
Description: Use the specified socket for interaction with the controlled modules		
`-P`	`--pid-file`	`<path to file>`
Description: Use the specified file as a PID file of **Dr.Web Agent**		
`-e`	`--export-config`	`<application name>`
Description: Export configuration of the specified application to **Dr.Web Enterprise Server**. Use the application name specified in the header of the `Application "<application name>"` section in the corresponding `amc` file (see Interaction with other Suite components). This parameter cannot be used if a **Dr.Web Agent** process is already running in the system or if you want to export **Dr.Web Anti-virus for Linux** configuration.		

Configuration File

Configuration of **Dr.Web Agent** is specified in the following file: `%etc_dir/agent.conf`.

For general organization concept of **Dr.Web for Unix Internet gateways** configuration files, see Configuration Files.

[Logging] Section

The `[Logging]` section contains **Dr.Web Agent** logging settings:

`[Logging]`

`Level` = `{log level}`	**Dr.Web Agent** log verbosity level. The following levels are available: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`Level`** = `Info`
`IPCLevel` = `{log level}`	Log verbosity level of IPC library. The following levels are available: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`IPCLevel`** = `Error`
`SyslogFacility` = `{syslog label}`	Log type label used by `syslogd` system service. Default value: **`SyslogFacility`** = `Daemon`
`FileName` = `{path to file | syslog}`	Path to the log file. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. Default value: **`FileName`** = `syslog`

[Agent] Section

The `[Agent]` section contains general **Dr.Web Agent** settings:

`[Agent]`

`MetaConfigDir` = `{path to directory}`	Name of the directory where meta-configuration files of **`drweb-agent`** are located.

	These files contain settings of interaction between **Dr.Web Agent** and other modules of the **Dr.Web** suite. Meta-configuration files are provided by **Dr.Web** developers and do not need to be modified. Default value: **MetaConfigDir** = %etc_dir/agent/
UseMonitor = {logical}	Yes value indicates to **drweb-agent** that **Dr.Web Monitor** is used as a part of **Dr.Web for Unix Internet gateways**. Default value: **UseMonitor** = Yes
MonitorAddress = {address}	Socket used by **Dr.Web Agent** for interaction with **Dr.Web Monitor** (the parameter value must be the same as the **Address** parameter value in the **Dr.Web Monitor** configuration file). Default value: **MonitorAddress** = local:%var_dir/ipc/.monitor
MonitorResponseTime = {numerical value}	Maximum time to get a response from **drweb-monitor** module, in seconds. If **Dr.Web Monitor** does not respond during this period, **Dr.Web Agent** considers **drweb-monitor** not running and stops trying to establish connection with **Dr.Web Monitor**. Default value: **MonitorResponseTime** = 5
PidFile = {path to file}	Name of the file where **Dr.Web Agent** PID is written on **Dr.Web Agent** startup. Default value: **PidFile** = %var_dir/run/drweb-agent.pid

[Server] Section

The [Server] section contains parameters that control interaction of **Dr.Web Agent** with other **Dr.Web for Unix Internet gateways** modules:

[Server]

Address = {address}	Socket used by **Dr.Web Agent** to interact with other modules of the suite. You can specify multiple sockets separating them by comma. Default value: **Address** = local:%var_dir/ipc/.agent, inet:4040@127.0.0.1
Threads = {numerical value}	Number of **drweb-agent** simultaneous threads. This parameter determines maximum number of simultaneous connections to modules that report virus statistics to **Dr.Web Agent**. The parameter value cannot be changed with SIGHUP signal. If 0 is specified, number of threads is unlimited (not recommended). Default value: **Threads** = 2

Timeout = {numerical value}	Maximum time (in seconds) for establishing connection between **Dr.Web Agent** and other **Dr.Web** modules. If the value is set to 0, time for establishing connection is unlimited. Default value: **Timeout** = 15

[EnterpriseMode] Section

The [EnterpriseMode] section contains parameters of **Dr.Web Agent** operation in the **Enterprise** mode:

[EnterpriseMode]

UseEnterpriseMode = {logical}	If the value is set to Yes, **Dr.Web Agent** operates in the Enterprise mode, if the value is set to No - in the Standalone mode. Default value: **UseEnterpriseMode** = No
ComputerName = {text value}	Name of the computer in **Anti-virus network**. Default value: **ComputerName** =
VirusbaseDir = {path to directory}	Path to the directory where virus databases are located. Default value: **VirusbaseDir** = %var_dir/bases
PublicKeyFile = {path to file}	Path to the public key file required to access **Dr.Web Enterprise Server**. Default value: **PublicKeyFile** = %bin_dir/drwcsd.pub
ServerHost = {IP address}	IP address of **Dr.Web Enterprise Server** . Default value: **ServerHost** = 127.0.0.1
ServerPort = {port number}	Number of the port required to access **Dr.Web Enterprise Server**. Default value: **ServerPort** = 2193
CryptTraffic = {Yes | Possible | No}	Encryption of traffic between **Dr.Web Enterprise Server** and **Dr.Web Agent**: • Yes – force encryption • Possible – encrypt if possible • No – do not encrypt Default value: **CryptTraffic** = possible
CompressTraffic = {Yes | Possible | No}	Compression of traffic between **Dr.Web Enterprise Server** and **Dr.Web Agent**: • Yes – force compression

	• `Possible` – compress if possible • `No` – do not compress Default value: **`CompressTraffic`** = `possible`
`CacheDir` = `{path to directory}`	Path to the directory, where different utility files are stored: configuration files, files with access privileges for applications managed by **Dr.Web Enterprise Server**, files with registration information on **Dr.Web Enterprise Server**, etc. Default value: **`CacheDir`** = `%var_dir/agent`

[StandaloneMode] Section

The `[StandaloneMode]` section contains parameters of **Dr.Web Agent** operation in the **Standalone** mode:

`[StandaloneMode]`

`StatisticsServer` = `{text value}`	Address (URL) of the virus statistics server If the value is not specified, statistics is not sent. Default value: **`StatisticsServer`** = `stat.drweb.com:80/update`
`StatisticsUpdatePeriod` = `{numerical value}`	Period (in minutes) for statistics updating. Value cannot be less than 5 Default value: **`StatisticsUpdatePeriod`** = `10`
`StatisticsProxy` = `{hostname | IP address}`	IP address or host name of proxy server for sending virus statistics. Please note that if the parameter value is not set, the value of `http_proxy` environment variable is used. **Example**: `StatisticsProxy = localhost:3128` Default value: **`StatisticsProxy`** =
`StatisticsProxyAuth` = `{text value}`	Authentication string (`<username>:<password>`) to access proxy server. **Example**: `StatisticsProxyAuth = test:testpwd` Default value: **`StatisticsProxyAuth`** =
`UUID` = `{text value}`	Unique user ID for the statistics server http://stat.drweb.com/. Please note that this parameter is mandatory for sending statistics. Thus, if you want to enable this option, specify the personal `UUID` as the parameter value (`md5` sum of license key file is usually used as UUID). Default value: **`UUID`** =

LicenseFile = {paths to files}	Location of **Dr.Web** license key files or demo key files. Paths in the list are separated by commas (if the list contains more than one path). Default value: **LicenseFile** = %bin_dir/drweb32.key

[Update] Section

The [Update] section contains parameters of **Dr.Web for Unix Internet gateways** update via **Dr.Web Enterprise Server**:

[Update]

CacheDir = {path to directory}	Directory where **Dr.Web Agent** temporarily stores downloaded update files. Default value: **CacheDir** = %var_dir/updates/cache
Timeout = {numerical value}	Maximum time (in seconds) for **Dr.Web Agent** to process downloaded update files. If 0 is specified, time for process is unlimited. Default value: **Timeout** = 120
RootDir = {path to directory}	Path to the root directory. Default value: **RootDir** = /

For more information, see *Administrator Manual* for **Dr.Web ESS**.

Running Dr.Web Agent

Please note that if at the post-install script runtime you select the "Configure Services" option in the conversation, all services including **Dr.Web Agent**, will be started automatically.

When **Dr.Web Agent** starts with the default settings, the following actions are performed:

- **Dr.Web Agent** searches and loads its configuration file. If the configuration file is not found, **Dr.Web Agent** terminates.
- If the parameters in the [EnterpriseMode] section are set correctly and **Dr.Web for Unix Internet gateways** is operating within **Anti-virus network**, **Dr.Web Agent** starts in the Enterprise mode. Otherwise, if parameters in the [Standalone] section are set correctly, **Dr.Web Agent** starts in the Standalone mode. If the parameters in the [Standalone] section are not set, **Dr.Web Agent** terminates.
- Socket for interaction of **Dr.Web Agent** with other **Dr.Web** modules is created. If a TCP socket is used, several connections can be established (loading continues if at least one connection is established). If a UNIX socket is used, it can only be created if the user, whose privileges are used to run **drweb-agent**, has read and write access to its directory. If a socket cannot be created, **Dr.Web Agent** terminates.

Further loading process depends on the selected operation mode.

If **Dr.Web Agent** operates in the **Enterprise mode**:

- **Dr.Web Agent** connects to **Dr.Web Enterprise Server**. If the server is unavailable or authorization process fails during the first connection attempt, **Dr.Web Agent** terminates. If **Dr.Web Agent** worked previously with this server and now the server is temporary unavailable (for example, if any connection problem occurs), **Dr.Web Agent** uses backup copies of configuration files received from the server earlier. These files are encrypted and must not be edited by a user. An attempt to edit the files makes them invalid.
- If the connection is established, **Dr.Web Agent** receives key files and settings from **Dr.Web Enterprise Server**. After all settings and key files are received, **Dr.Web Agent** is fully operational.

If **Dr.Web Agent** operates in the **Standalone mode**, meta-configuration files (.amc) that manage **Dr.Web Agent** interaction with other **Dr.Web** modules are loaded. Location of meta-configuration files is set in the **`MetaConfigDir`** parameter in the `[Agent]` section of the **Dr.Web Agent** configuration file. When meta-configuration files are successfully loaded, **Dr.Web Agent** is ready to operate.

Interaction with Other Suite Components

Interaction with other suite components is performed by **Dr.Web Agent** metaconfiguration files (`amc` files). These files contain configuration parameters that are sent to the respective **Dr.Web** modules by **Dr.Web Agent**. The files reside in the directory specified in the **`MetaConfDir`** parameter (by default - `%etc_dir/agent`). Usually, one file contains configuration parameters of one component and name of the file matches to the name of the **Dr.Web for Unix Internet gateways** component.

Each module is described in the `Application` section with the corresponding name. At the end of the section **`EndApplication`** must be specified.

The following parameters must be present in the module description:

- **`id`**: identifier of the module in **Dr.Web ESS**.
- **`ConfFile`**: path to the module configuration file.
- **`Components`**: description of the modules. At the end of this section, `EndComponents` must be specified. Description of each module must contain the following information: name and list of sections in the configuration file with parameters that are necessary for proper operation. The list of sections and parameters is comma separated.
  To describe individual parameters properly, specify the full path to them (for example, `/Quarantine/DBISettings)`. In the section descriptions, only their names can be specified (for example, `General)`.
  To denote line breaks, a back slash (\) is used.
  If the component requires all settings from the configuration file, you can specify a path "/*" instead of the list of sections and/or parameters.

Example of amc file for Dr.Web ICAPD for Linux:

```
Application "ICAPD"
      id  49
      ConfFile  "/etc/drweb/drweb-icapd.ini"
      Components
                  drweb-icapd   Icapd
    EndComponents
EndApplication
```

Integration with Dr.Web Enterprise Security Suite

There are two possible situations which require integration of **Dr.Web for Unix Internet gateways**

with **Dr.Web Enterprise Security Suite**:

- Setup and initial configuration of **Dr.Web for Unix Internet gateways** in the existing **Anti-virus Network** operated by **Dr.Web ESS**;
- Embedding of working UNIX server with already installed and configured **Dr.Web for Unix Internet gateways** in the **Anti-virus Network** operated by **Dr.Web ESS**.

To enable **Dr.Web for Unix Internet gateways** to work in **Dr.Web ESS** environment, configure **Dr.Web Agent** and **Dr.Web Monitor** components for operation in the `Enterprise` mode, and register the suite on **Dr.Web Enterprise Server**.

According to the connection policy for new working stations (for details, see **Dr.Web Enterprise Security Suite** administrator manual), **Dr.Web for Unix Internet gateways** can be connected to **Dr.Web Enterprise Server** in two different ways:

- when a new account is automatically created by the central protection server
- when a new account is created by administrator manually.

Configuring Components to Run in Enterprise Mode

To start the components in the `Enterprise` mode after installation, it is necessary to adjust parameter values in the local configuration files of **Dr.Web Agent** and **Dr.Web Monitor**.

For Dr.Web Agent

In the `[EnterpriseMode]` section of **Dr.Web Agent** `configuration file` (`%etc_dir/agent.conf`) set the following parameter values:

- **`UseEnterpriseMode`** = `Yes`;
- **`PublicKeyFile`** = `%var_dir/drwcsd.pub` (public encryption key used to access **Dr.Web Enterprise Server**. Administrator must move this file from the corresponding directory of **Dr.Web Enterprise Server** to the specified path);
- **`ServerHost`** = IP address or host name of **Dr.Web Enterprise Server**;
- **`ServerPort`** = **Dr.Web Enterprise Server** `port` (`2193` by default).

For Dr.Web Monitor

In the `[Monitor]` section of the **Dr.Web Monitor** configuration file `%etc_dir/monitor.conf` set the following parameter values:

- **`UseEnterpriseMode`** = `Yes`.

Automatic Creation of New Account by ES Server

When a new account is created automatically:

1. On the first run in the `Enterprise` mode, **Dr.Web Agent** sends a request for the account details (station ID and password) to **Dr.Web Enterprise Server**;
2. If **Dr.Web Enterprise Server** is set to the **Approve access manually** mode (used by default; for details, see the administrator manual for **Dr.Web ESS**), system administrator must confirm registration of a new station via **Dr.Web Control Center** web interface in one minute;
3. After the first connection, **Dr.Web Agent** records the hash of the station ID and password into the `pwd` file. This file is created in the directory specified in the **`CacheDir`** parameter of the `[EnterpriseMode]` section (default value is `%var_dir/agent/)`;
4. Data from this file is used every time **Dr.Web for Unix Internet gateways** connects to **Dr.Web Enterprise Server**;
5. If you delete the password file, repeated registration request will be sent to **Dr.Web Enterprise Server** on the next **Dr.Web Agent** startup.

Manual Creation of New Account by Administrator

To create a new account manually:

1. Create a new account on **Dr.Web Enterprise Server**: specify the station ID and password (for details, see the administrator manual for **Dr.Web ESS**).
2. Start **Dr.WebAgent** with the **`--newpwd`** command line parameter (or **`-p`**) and enter the station ID and password. **Dr.Web Agent** records the hash of station ID and password into the `pwd` file. This file is created in the directory that is specified in the **`CacheDir`** parameter of the [`EnterpriseMode`] section (default value is `%var_dir/agent/`) .
3. Data from this file is used every time **Dr.Web for Unix Internet gateways** connects to **Dr.Web Enterprise Server**.
4. If you delete the password file, retry registration on the next **Dr.Web Agent** startup.

Configuring Components via Dr.Web Control Center (embedded in Enterprise Security Suite)

You can configure **Dr.Web for Unix Internet gateways** and **Dr.Web Daemon** (anti-virus module included in the standard installation package) via **Dr.Web Control Center**.

The standard installation package **Dr.Web Enterprise Security Suite** includes basic configuration files for **Dr.Web for Unix Internet gateways** and **Dr.Web Daemon** for **Linux**, **FreeBSD** and **Solaris**. When you configure certain components via the web interface (**Dr.Web Control Center**), values of the corresponding parameters change in these configuration files on **Dr.Web Enterprise Server**. After that, every time the components start, **Dr.Web Agent** requests configuration from **Dr.Web Enterprise Server**.

Export of Existing Configuration to ES Server

You can export configuration from the local computer to **Dr.Web Enterprise Server** automatically when **Dr.Web Agent** is operating in the `Enterprise` mode. To export configuration, use the command line parameter **`--export-config`** (or **`-e`**).

You must specify the name of the component (`DAEMON`, `ICAPD`).

Example:

```
# %bin_dir/drweb-agent --export-config ICAPD
```

Starting the System

To start the system:

1. In **Dr.Web Control Center**, open **Dr.Web Monitor** settings and select the **Daemon** and **ICAP** check boxes to start the corresponding components;
2. Start **Dr.Web Monitor** on the local computer:

   For **Linux** and **Solaris**:

   ```
   # /etc/init.d/drweb-monitor start
   ```

   For **FreeBSD**:

   ```
   # /usr/local/etc/rc.d/00.drweb-monitor.sh start
   ```

Integration with Dr.Web ESS 10

Dr.Web for Unix Internet gateways 6.0.2 includes two versions of the **Dr.Web Agent**:

- **Dr.Web Agent**, implemented as **`drweb-agent`** module, in **enterprise** mode can interact only with **Dr.Web ESS** server version 6.
- **Dr.Web Agent**, implemented as **`drweb-agent10`** module, in **enterprise** mode can interact only with **Dr.Web ESS** server version 10.

To start using the central protection server **Dr.Web ESS** 10, configure standard integration and also make additional settings.

The products, operating in **FreeBSD** 6.x, cannot be integrated with **Dr.Web ESS** 10.

Configuring connection to Dr.Web ESS 10

As **Dr.Web ESS** does not support management of **Dr.Web Monitor** and **Dr.Web Daemon**, **`drweb-agent10`** uses two supplementary configuration files in addition to the standard file `%etc_dir/agent.conf`: `es_monitor.conf` and `es_daemon.conf`. They are located in the same directory. These files store configuration for **Dr.Web Monitor** and **Dr.Web Daemon**. The configuration settings will be used for adjusting operation of these modules in **enterprise** mode.

Each file line contains the parameter value of the corresponding module configuration. The format is as follows: `<section>/<parameter> <value>`, where `<section>` is the name of the section from the component configuration file, `<parameter>` is the parameter name, and `<value>` is the value specified for this parameter.

Example (for `es_monitor.conf` file that contains settings for **Dr.Web Monitor** component operation in **enterprise** mode):

```
Monitor/RunAppList DAEMON
```

This line contains the value of **`RunAppList`** parameter stored in `[Monitor]` section in **Dr.Web Monitor** configuration file. This parameter value is used when the suite is running in **enterprise** mode. In this case, **Dr.Web Monitor** starts only **Dr.Web Daemon**.

Example (for `es_daemon.conf` file that contains settings for **Dr.Web Daemon** component operation in **enterprise** mode):

```
Daemon/MaxCompressionRatio 500
```

This line contains the value of **`MaxCompressionRatio`** parameter stored in `[Daemon]` section in **Dr.Web Daemon** configuration file. This parameter value is used when the suite is running in **enterprise** mode. In this case, **Dr.Web Daemon** uses 500 as the threshold value of compression ratio.

To connect **Dr.Web for Unix Internet gateways** to the central protection server **Dr.Web ESS** 10:

1. Open `agent.mmc` meta-configuration file (used by **Dr.Web Monitor** for communication with **Dr.Web Agent**) and replace the specified binary file name **`drweb-agent`** with **`drweb-agent10`**.

2. In `es_monitor.conf` file, specify components to be started int **enterprise** mode. For that purpose, edit the `es_monitor.conf` accordingly. The set of started components must be similar to the set of components started in **standalone** mode (specified as the value of **`RunAppList`** parameter stored in `[Monitor]` section in **Dr.Web Monitor** configuration file). If more than one component must be started, they are specified as a comma-separated list. Note that white spaces are not allowed. Example:

   ```
   Monitor/RunAppList DAEMON,ICAPD
   ```

   As the component names, here should be used the names specified in `Application` section of mmc-files.

3. If required, configure parameters in `es_daemon.conf` file that is used by **Dr.Web Daemon** respectively in **enterprise** mode.
4. If **standalone** mode was previously used, switch operation of **Dr.Web Agent** and **Dr.Web Monitor** components to **enterprise** mode by specifying appropriate settings in their configuration files, as described in the Configuring Components to Run in Enterprise Mode section.
5. Restart **Dr.Web Monitor** by using the following command:

   ```
   # service drweb-monitor restart
   ```

Gathering Virus Statistics

Dr.Web Agent receives statistics on computer threats from the controlled modules and sends it either to the official **Doctor Web** statistics website: http://stat.drweb.com/ (if the Internet connection is available) or to **Dr.Web ESS** (if **Dr.Web Agent** is operating in the Enterprise mode).

Dr.Web Agent needs the *unique user identifier* (UUID) to connect to this website. By default, MD5 hash of the key file is used as a UUID. Also you can get a personal UUID from **Doctor Web Technical Support**. In this case, specify your UUID explicitly in the **Dr.Web Agent** configuration file (`StandaloneMode]` section).

Statistics is gathered only for those **Dr.Web** modules that receive settings from **Dr.Web Agent**. Instructions on how to set up interaction with **Dr.Web Agent** are given in the sections describing the modules.

On the statistics website (at http://stat.drweb.com/), you can view aggregate statistics on computer threats both for a given server and for all servers supported by **Dr.Web Anti-virus for UNIX** or by **Dr.Web for Unix Internet gateways** with an anti-virus plug-in. **Dr.Web Agent** can simultaneously process statistics on computer threats from several different **Dr.Web** products which are able to interact with **Dr.Web Agent**.

If **Dr.Web Agent** is operating in the Enterprise mode, you can view statistics on the special page of **Dr.Web Control Center**. In this case, statistics gathered by **Dr.Web Enterprise Server** is also sent to the **<%COMPANYNAME%>** statistics server as a summary of the **Anti-virus network** statistics.

Statistics is available in both HTML and XML formats. The second format is convenient if you plan to publish this statistics on another website, since data in the XML format can be transformed according to the website concept and design.

To view aggregate statistics on computer threats for all supported servers, visit http://stat.drweb.com/. You can view a list of detected threats for all supported servers (in descending order) with overall percentage of detections.

Appearance of the webpage can differ depending on the used browser.

The following figure shows threats statistics page.

Figure 15. Computer threats statistics

You can change search options and repeat the search. To do this:

1. Select either **Mail** or **Files** check boxes to get statistics on computer threats detected in emails or files.
2. In the drop-down lists for **Start date** and **End date**, select **start/end date** and **time** for the required period.
3. In the **Top** field, enter the required number of rows in the statistics table (most frequently detected threats will be shown).
4. Click **Query**. The file with aggregate statistics in the XML format can be found at http://info.drweb.com/export/xml/top

Example:

```
<drwebvirustop period="24" top="5"
  vdbaseurl="http://info.drweb.com/virus_description/"
  updatedutc="2009-06-09 09:32:02">
 <item>
  <vname>Win32.HLLM.Netsky</vname>
  <dwvlid>62083</dwvlid>
  <place>1</place>
  <percents>34.201062139103</percents>
 </item>
 <item>
  <vname>Win32.HLLM.MyDoom</vname>
  <dwvlid>9353</dwvlid>
  <place>2</place>
  <percents>25.1303270912579</percents>
 </item>
 <item>
  <vname>Win32.HLLM.Beagle</vname>
  <dwvlid>26997</dwvlid>
  <place>3</place>
  <percents>13.4593034783378</percents>
 </item>
 <item>
  <vname>Trojan.Botnetlog.9</vname>
  <dwvlid>438003</dwvlid>
  <place>4</place>
  <percents>7.86446592583328</percents>
 </item>
 <item>
  <vname>Trojan.DownLoad.36339</vname>
  <dwvlid>435637</dwvlid>
  <place>5</place>
  <percents>7.31494163115527</percents>
 </item>
</drwebvirustop>
```

In this file, the following XML attributes are used:

- `period` – duration (in hours) of the statistics collection process;
- `top` – number of the most frequently detected threats shown in the statistics table (number of rows);
- `updatedutc` – last statistics update time;
- `vname` – threat name;
- `place` – place of the virus in the statistics;
- `percents` – percentage of the total number of detections.

Value of the period parameter and size of the sample cannot be changed by user.

To get personalized threat statistics

Visit one of the following webpages:

- For statistics in HTML format, go to http://stat.drweb.com/view/*<UUID>*. Page with the personalized statistics is similar to the aggregate statistics page.
- For the file with the personalized threat statistics in XML format, go to http://stat.drweb.com/xml/*<UUID>*.

The *<UUID>* in both cases stands for the MD5 hash of your license key file (unless you have a personal UUID received from **Doctor Web Technical Support**).

Example:

```
<drwebvirustop period="24" top="2" user="<UUID>"
  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>
```

In this file, the following XML attributes are used:

- `period` – duration (in hours) of the statistics collection process;
- `top` – number of the most frequently detected threats shown in the table (number of rows);
- `user` – user identifier;
- `lastdata` – time when user last sent data to the server;
- `vname` – threat name;
- `place` – threat place in the statistics;
- `caught` – number of detections of the certain threat;
- `percents` – percentage of the total number of detections.

Value of the period parameter and size of the sample cannot be changed by user.

Dr.Web Monitor

Dr.Web Monitor is a memory resident module **`drweb-monitor`**.

It is used to increase fault-tolerance of the whole **Dr.Web for Unix Internet gateways** suite. It ensures correct startup and termination of suite components as well as restart of any component if it is operating abnormally. **Dr.Web Monitor** starts all modules and loads, if necessary, some extra components of these modules. If **Dr.Web Monitor** fails to start a module, it repeats an attempt later. Number of attempts and time period between them are defined by **Dr.Web Monitor** settings.

After all modules are loaded, **Dr.Web Monitor** permanently controls their operation. If any module or one of its components operates abnormally, **Dr.Web Monitor** restarts the application. Maximum number of attempts to restart a component and a period of time between them are defined by **Dr.Web Monitor** settings. If any of the modules starts to operate abnormally, **Dr.Web Monitor** notifies the system administrator.

Dr.Web Monitor can interact with **Dr.Web Agent** by exchanging control signals.

Operation Mode

If necessary, **Doctor Web** solutions can be used to connect to a corporate or private **Anti-virus network** managed by **Dr.Web Enterprise Security Suite**. To operate in the central protection mode, it is not required to install additional software or uninstall your **Dr.Web** solution.

To provide you with this option, **Dr.Web Monitor** can operate in one of the following modes:

- **Standalone mode** when a protected computer is not included in an anti-virus network and is managed locally. In this mode, configuration files and key files reside on local drives, **Dr.Web Monitor** is fully controlled from the protected computer, and all modules start in accordance with the settings specified in the **Dr.Web Monitor** configuration file.
- **Enterprise mode** (or **central protection mode**) when protection of the local computer is managed from the central protection server. In this mode, some features and settings of **Dr.Web for Unix Internet gateways** can be modified and blocked for compliance with a general security policy (for example, corporate security policy). A key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used.

To enable central protection mode

1. Contact anti-virus network administrator of your company for a public key file and parameters of connection to the central protection server.
2. In **Dr.Web Monitor** configuration file (by default, `%etc_dir/monitor.conf`), set the **`UseEnterpriseMode`** parameter value to `Yes`.

   In the central protection mode, some features and settings of **Dr.Web for Unix Internet gateways** can be modified or blocked for compliance with the general security policy. A key file for operation in this mode is received from the central protection server. Your personal key file on the local computer is not used.

For **Dr.Web for Unix Internet gateways** to fully support the central protection mode, also enable **Dr.Web Agent** to operate in the Enterprise mode. For details, see Operation Mode of **Dr.Web Agent**.

To enable standalone mode

1. Ensure that all modules that you want **Dr.Web Monitor** to start are listed in the **RunAppList** parameter in the [Monitor] section of **Dr.Web Monitor** configuration file (by default, %etc_dir/monitor.conf). The modules must be installed and configured properly.
2. In the [Monitor] section of **Dr.Web Monitor** configuration file, set the **UseEnterpriseMode** parameter value to No.

   On switching to this mode, all settings of **Dr.Web for Unix Internet gateways** are unlocked and restored to their previous or default values. You can access all settings of **Dr.Web for Unix Internet gateways** again and configure them.

For correct operation in the standalone mode, **Dr.Web for Unix Internet gateways** requires a valid personal key file. The key files received from the central protection server cannot be used in this mode.

Command Line Parameters

To run **Dr.Web Monitor**, use this command:

```
drweb-monitor [parameters]
```

where the following parameters are allowed:

Short case	Extended case	Arguments
-h	--help	
Description: Show information about supported command line parameters on the screen and terminate the module		
-v	--version	
Description: Show **Dr.Web Monitor** version on the screen and terminate the module		
-u	--update	
Description: Start updating all **Dr.Web for Unix Internet gateways** components		
-C	--check-only	
Description: Check correctness of **Dr.Web Monitor** configuration. This parameter cannot be used if a **Dr.Web Monitor** process is already running in the system.		
-A	--check-all	<path to file>
Description: Check correctness of configuration of all **Dr.Web for Unix Internet gateways** components		
-c	--conf	<path to file>
Description: Module must use the specified configuration file		
-r	--run	<application name>[,<application name>,...]
Description: Run applications, name of which are specified. Use the application name specified in the header of the Application "<application name>" section in the corresponding mmc file (for details, see Interaction with other Suite Components). This parameter cannot be used if a **Dr.Web Monitor** process is already running in the system.		

Example usage:

```
drweb-monitor -r AGENT
```

Configuration File

Adjustment of **Dr.Web Monitor** settings is performed in its configuration file `%etc_dir/monitor.conf`.

For general organization concept of **Dr.Web for Unix Internet gateways** configuration files, see Configuration Files.

[Logging] Section

In the `[Logging]` section, parameters responsible for logging information on operation of **Dr.Web Monitor** are collected:

`[Logging]`

`Level` = `{log level}`	**Dr.Web Monitor** log verbosity level. The following levels are available: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`Level`** = `Info`
`IPCLevel` = `{log level}`	Log verbosity level for IPC library. The following levels are available: • `Quiet` • `Error` • `Alert` • `Info` • `Debug` Default value: **`IPCLevel`** = `Error`
`SyslogFacility` = `{syslog label}`	Log type label which is used by `syslogd` system service. Default value: **`SyslogFacility`** = `Daemon`
`FileName` = `{syslog | path to file}`	Path to the log file. You can specify `syslog` as a log file name and logging will be performed by `syslogd` system service. In this case, you must also specify the **`SyslogFacility`** parameter. Default value: **`FileName`** = `syslog`

[Monitor] Section

The `[Monitor]` section contains main settings of **Dr.Web Monitor**:

`[Monitor]`

RunForeground = {logical}	**Yes** value forbids **Dr.Web Monitor** to operate in daemon mode. This option can be used by some monitoring utilities (for example, **daemontools**). Default value: **RunForeground** = No
User = {text value}	Name of the user whose privileges are used by **Dr.Web Monitor.** Default value: **User** = drweb
Group = {text value}	User group name used to run **Dr.Web Monitor** with certain user privileges. Default value: **Group** = drweb
PidFileDir = {path to directory}	Path to the directory of a file where information on **Dr.Web Monitor** process identifier (PID) is written upon the module startup. Default value: **PidFileDir** = %var_dir/run/
ChDir = {path to directory}	Change of working directory upon **Dr.Web Monitor** startup. If this parameter is set, **Dr.Web Monitor** changes directory to the one specified in this parameter value. Otherwise, working directory is not changed. Default value: **ChDir** = /
MetaConfigDir = {path to directory}	Path to the directory where metaconfiguration files reside. These files contain settings defining **Dr.Web Monitor** interaction with other **Dr.Web** components. Metaconfiguration files are provided by **Dr.Web** developers and do not require editing. Default value: **MetaConfigDir** = %etc_dir/monitor/
Address = {address}	Socket used by **Dr.Web Monitor** to receive control signals from other **Dr.Web** components. Default value: **Address** = local:%var_dir/ipc/.monitor
Timeout = {numerical value}	Maximum time (in seconds) to establish connection between **Dr.Web Monitor** and other **Dr.Web** components. Default value: **Timeout** = 5
TmpFileFmt = {text value}	Name templates for **Dr.Web Monitor** temporary files. Template format: path_to_file.XXXXXX where X is a random symbol (letter or digit), used in temporary file names. Default value: **TmpFileFmt** = %var_dir/msgs/tmp/monitor.XXXXXX

RunAppList = {text value}	List of modules started by **Dr.Web Monitor**; use comma as a delimiter. Please note that this parameter is not modified upon uninstalling a **Dr.Web** component. You must manually remove the uninstalled component from this parameter value. Otherwise, **Dr.Web Monitor** will not be able to run and start other **Dr.Web** components. Default value: **RunAppList** = AGENT
UseEnterpriseMode = {logical}	If the value is set to Yes, **Dr.Web Monitor** receives the list of modules to be started from **Dr.Web Agent** rather than from the **RunAppList** parameter value. Default value: **UseEnterpriseMode** = No
RecoveryTimeList = {numerical values}	Time intervals between attempts to restart components that are not responding (in seconds). This parameter can have multiple values, separated by commas. First attempt to restart a component is made after a period of time specified in the first parameter value, second attempt – using the second parameter value, and so on. Default value: **RecoveryTimeList** = 0,30,60
InjectCmd = {string}	Command to send reports. Please note that if you want to send reports to other addresses (not only to root@localhost), you need to specify the addresses in the command. Default value: **InjectCmd** = "/usr/sbin/sendmail -t"
AgentAddress = {address}	Socket used by **Dr.Web Monitor** to interact with **Dr.Web Agent** (parameter value must be the same as the **Address** parameter value from **Dr.Web Agent** configuration file). Default value: **AgentAddress** = local:%var_dir/ipc/.agent
AgentResponseTime = {numerical value}	Maximum time to wait a response from drweb-agent module in seconds. If **Dr.Web Agent** does not respond during this time period, **Dr.Web Monitor** considers drweb-agent not working and tries to restart it. If 0 is specified, response time is unlimited. Default value: **AgentResponseTime** = 5

Running Dr.Web Monitor

When **Dr.Web Monitor** is started with the default settings, the following actions are performed:

1. **Dr.Web Monitor** searches for and loads its configuration file. If the configuration file is not found, loading process stops;
2. **Dr.Web Monitor** starts operating in the `daemon` mode. So, information about loading problems cannot be output to the console and, thus, is logged to the file;
3. Socket for **Dr.Web Monitor** interaction with other **Dr.Web for Unix Internet gateways** modules is created. If a TCP socket is used, several connections can be established (loading process continues if at least one connection is established). If a UNIX socket is used, it can be created only if the user whose privileges are used to run **`drweb-monitor`** has read and write access to the certain directory. If a socket cannot be created, loading process stops;
4. PID-file with information on **`drweb-monitor`** process identifier is created. If the PID-file cannot be created, loading process stops;
5. **`drweb-monitor`** module starts other suite components. If a module cannot load, **Dr.Web Monitor** tries to restart it. If all **Dr.Web Monitor** attempts to start the module failed, **Dr.Web Monitor** unloads all previously loaded modules and terminates. **Dr.Web Monitor** reports problems connected with the modules startup in one of the available ways (logging to the file, notifying via email, startup of a custom program). Notification methods used for various modules are set in the **Dr.Web Monitor** meta-configuration file (.mmc).

To start **Dr.Web Monitor** in the automatic mode, do one of the following:

- change the value of the `ENABLE` variable to `1` in the **`drweb-monitor`** `enable` file (for **Linux** and **Solaris**);
- add `drweb_monitor_enable="YES"` line to the `/etc/rc.conf` file (for **FreeBSD**).

Please note that if at the post install script runtime you select the "Configure Services" option in the conversation, all services including **Dr.Web Agent** will be started automatically.

Location of the enable files depends on **Dr.Web for Unix Internet gateways** installation type:

- Installation from the **universal package for UNIX systems**:
  Files will be saved to `%etc_dir` directory and have the following names
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.
- Installation from **native DEB packages**:
  Files will be saved to `/etc/defaults` directory and have the following names
  `drweb-icapd`,
  `drwebd`,
  `drweb-monitor`.
- Installation from **native RPM packages**:
  Files will be saved to `/etc/sysconfig` directory and have the following names
  `drweb-icapd.enable`,
  `drwebd.enable`,
  `drweb-monitor.enable`.

Interaction with Other Suite Components

Interaction with other suite components is performed with the use of **Dr.Web Monitor** meta-configuration files (`mmc` files). These files are included in packages of those products which can interact with **Dr.Web Monitor** and reside in reside in the directory specified in the **`MetaConfDir`** parameter

(by default - `%etc_dir/monitor`). The files contain information on component composition, location of binary files, their launch order and startup options. Usually, one file contains information on one component and name of the file matches to the name of the **Dr.Web for Unix Internet gateways** component.

Each component is described in the `Application` section with the corresponding name. At the end of the section, `EndApplication` must be specified.

The following parameters must be present in the component description:

- **`FullName`** – full name of the component.
- **`Path`** – path to the binary files.
- **`Depends`** – names of the components which must be started before the described component. For example, `AGENT` component must be started before **Dr.Web Daemon,** therefore in the `mmc` file for **Dr.Web Daemon** **`Depends`** parameter has the `AGENT` value. If there are no dependencies, this parameter can be skipped.
- **`Components`** – list of binary files of modules started together with the component. Modules are started in the same order as they are specified in this parameter. For each module the following information must be specified (space separated): command line parameters (can be enclosed in quotation marks), timeouts for startup and stop (`StartTimeout` and `StopTimeout`), notification type and startup privileges. *Notification type* – defines where notifications on component failure are sent. When `MAIL` value is specified, notifications are sent by mail, when `LOG` value is specified, information is only logged to the file. *Startup privileges* – defines a group and a user, whose privileges are used by the component.

Example of mmc file for Dr.Web Daemon:

```
Application "DAEMON"
 FullName  "Dr.Web (R) Daemon"
 Path      "/opt/drweb/"
 Depends   "AGENT"
 Components
  # name  args  MaxStartTime  MaxStopTime  NotifyType  User:Group
  drwebd "-a=local:/var/drweb/ipc/.agent --foreground=yes" 30 10 MAIL drweb:drweb
 EndComponents
EndApplication
```

Example of mmc file for Dr.Web ICAPD:

```
Application "ICAPD"
 FullName    "Dr.Web (R) icapd"
 Path        "/opt/drweb/"
 Depends     "AGENT"
 Components
  # name  args  MaxStartTime  MaxStopTime  NotifyType  User:Group
  drweb-icapd   "-m -f local:/var/drweb/ipc/.agent"  5  5  MAIL  drweb:drweb
 EndComponents
EndApplication
```

Dr.Web Command Line Scanner

Command line **Dr.Web Scanner** provides you with detection and neutralization of malware on the local machine. The component is presented by the **drweb** module.

Dr.Web Scanner checks files and boot records specified on its startup. For anti-virus checking and curing, **Dr.Web Scanner** uses **Dr.Web Engine** and virus databases, but does not use the resident module **Dr.Web Daemon** (operation is performed independently of it).

Running Dr.Web Scanner

You can run **Dr.Web Scanner** with the following command:

```
$ %bin_dir/drweb
```

If `%bin_dir` directory is added to the `PATH` environment variable, you can run **Dr.Web Scanner** from any directory. However, doing so (as well as making a symbolic link to **Dr.Web Scanner** executable file in directories like `/bin/`, `/usr/bin/`, etc.) is not recommended for security reasons.

Dr.Web Scanner can be run with either root or user privileges. In the latter case, virus scanning can be performed only in those directories, where the user has read access, and infected files will be cured only in directories, where the user has write access (usually it is the user home directory, `$HOME`). There are also other restrictions when **Dr.Web Scanner** is started with user privileges, for example, on moving and renaming infected files.

When **Dr.Web Scanner** is started, it displays the program name, platform name, program version number, release date and contact information. It also shows user registration information and statistics, list of virus databases and installed updates:

```
Dr.Web (R) Scanner for Linux, v6.0.1 (February 19, 2010)
Copyright (c) Igor Daniloff, 1992-2010
Support service: http://support.drweb.com/
To purchase: http://buy.drweb.com/
Program version: 6.0.0.10060 <API:2.2>
Engine version: 6.0.0.9170 <API:2.2>
Loading /var/drweb/bases/drwtoday.vdb - Ok, virus records: 1533
Loading /var/drweb/bases/drw60012.vdb - Ok, virus records: 3511
-----------------------------------------
Loading /var/drweb/bases/drw60000.vdb - Ok, virus records: 1194
Loading /var/drweb/bases/dwn60001.vdb - Ok, virus records: 840
Loading /var/drweb/bases/drwebase.vdb - Ok, virus records: 78674
Loading /var/drweb/bases/drwrisky.vdb - Ok, virus records: 1271
Loading /var/drweb/bases/drwnasty.vdb - Ok, virus records: 4867
Total virus records: 538681
Key file: /opt/drweb/drweb32.key
Key file number: XXXXXXXXXX
Key file activation date: XXXX-XX-XX
Key file expiration date: XXXX-XX-XX
```

After displaying this report, **Dr.Web Scanner** terminates and command line prompt. To scan for viruses or neutralize detected threats, specify additional command line parameters.

By default, **Dr.Web Scanner** starts with the following parameters:

-ar -ha -fl- -ml -sd -al -ok

These parameters are optimal for thorough anti-virus protection and can be used in most typical cases. If any of the parameters is not required, disable it with "-" postfix as described above.

Disabling scan of archives and packed files will significantly decrease an anti-virus protection level, because viruses are often distributed in archives (especially, self-extracting archives) attached to an email message. Office documents (Word, Excel) dispatched within an archive or a container can also pose a threat to security of your computer as they are vulnerable to macro viruses.

When you start **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are performed. To enable these actions, specify the corresponding command line parameter explicitly.

The following actions are recommended:

- **cu** – cure infected files and system areas without deleting, moving or renaming infected files;
- **icd** – delete incurable files;
- **spm** – move suspicious files;
- **spr** – rename suspicious files.

When **Dr.Web Scanner** is started with **cu** action specified, it tries to restore the original state of an infected object. It is possible only if a detected virus is a known virus, and cure instructions for it are available in virus database; even in this case a cure attempt may fail if the infected file is seriously damaged by a virus.

When an infected file is found within an archive, the file is not cured, deleted, moved or renamed. To cure such a file, manually unpack the archive to the separate directory and instruct **Dr.Web Scanner** to check it.

When **Dr.Web Scanner** is started with **icd** action specified, it removes all infected files from the disk. This option is suitable for incurable (irreversibly damaged by a virus) files.

The **spr** action instructs **Dr.Web Scanner** to replace a file extension with another one (`*.#??` by default, that is the first extension character is replaced with the "#" character). Enable this parameter for files of other operating systems, detected heuristically as suspicious. Renaming helps to avoid accidental execution of such files in these operating systems and therefore prevents infection.

The **spm** action instructs **Dr.Web Scanner** to move infected or suspicious files to the **Quarantine** directory (`%var_dir/infected/` by default). This option is of insignificant value since infected and suspicious files of other operating systems cannot infect or damage a UNIX system. Moving of suspicious files of a UNIX system may cause system malfunction or failure.

Thus, the following command is recommended for day-to-day scanning:

```
$ drweb <path> -cu -icd -spm -ar -ha -fl- -ml -sd
```

You can save this command to the text file and convert it into simple shell script with the following command:

```
# chmod a+x [filename]
```

Dr.Web Scanner default settings could be adjusted in the configuration file.

Command Line Parameters

You can run **Dr.Web Scanner** with the following command:

```
$ %bin_dir/drweb <path> [parameters]
```

where `<path>` – is either the path (or paths) to scanned directories or mask for checked files. If a path is specified with the following prefix: `disk://<path to device file>` (files of the devices are

located in the /dev directory), **Dr.Web Scanner** checks the boot sector of the corresponding device and cure it, if necessary. The path can start with an optional parameter - path.

When **Dr.Web Scanner** is started only with the <path> argument, without any parameters specified, it scans the specified directory using the default set of parameters (for details, see below).

The following example shows a command to check the user home directory:

```
$ %bin_dir/drweb ~
```

Once scanning completes, **Dr.Web Scanner** displays all detected threats (infected and suspicious files) in the following format:

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

After that, **Dr.Web Scanner** outputs summary report in the following format:

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

Numbers separated by slash "/" mean the following: the first number – total number of files, the second one – number of files in archives.

You can use readme.eicar file, included in the distribution package, to test **Dr.Web Scanner**. Open this file in any text editor and follow the instructions from the file to transform it into eicar.com program.

When you check the program with **Dr.Web Scanner**, the following message must be output:

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

This program is not a virus and is used only for testing of anti-virus software.

Dr.Web Scanner has numerous command-line parameters. In accordance with UNIX conventions, the parameters are separated from a path by a space character and start with a hyphen ("-"). To get a full list of parameters, run **Dr.Web Scanner** with either -?, -h, or -help parameters.

The **Console Scanner** basic parameters can be divided into the following groups:

- Scan area parameters
- Diagnostic parameters
- Action parameters
- Interface parameters

Scan Area Parameters

These parameters determine where to perform a virus scan:

Parameter	Description
-path [=] <path>	Sets the path to be scanned. Symbol '=' can be skipped, in this case a path for scanning is separated from the -path parameter by a space. You can specify several paths in one -path parameter (paths will be combined into one list). You can also specify paths without the -path parameter. If in the startup options the <path> parameter is specified with following prefix: disk://<path to device file>,

Parameter	Description
	the boot sector (MBR) of the corresponding device is checked and cured, if necessary. Device file is a special file, located in the `/dev` directory and named as **sdX** or **hdX**, where **X** is a letter of the Latin alphabet (`a`, `b`, `c`, ...). For example: **hda**, **sda**. Thus, to check MBR of disk **sda**, specify the following: disk:///dev/sda
`-@[+]<file>`	Instructs to scan objects listed in the specified file. Add a plus '+' if you do not want the file with the list of objects to be deleted when scanning completes. The file can contain paths to directories that must be periodically scanned or list of files to be checked regularly.
`--`	Instructs to read the list of objects for scanning from the standard input stream (`stdin`).
`-sd`	Sets recursive search for files to scan in subfolders.
`-fl`	Instructs to follow symbolic links to both files and folders. Links that cause loops are ignored.
`-mask`	Instructs to ignore filename masks.

Diagnostic Parameters

These parameters determine object types to be scanned for viruses:

Parameter	Description
-al	Instructs to scan all objects defined by scan paths regardless of their file extension and structure. This parameter is opposite to the **-ex** parameter.
-ex	Instructs to scan only files of certain types in the specified paths. The list of file types must be specified in the **FileTypes** variable of the configuration file. The configuration file is defined by the **-ini** parameter. By default, objects with the following file extensions are scanned: EXE, COM, DLL, SYS, VXD, OV?, BAT, BIN, DRV, PRG, BOO, SCR, CMD, 386, FON, DO?, XL?, WIZ, RTF, CL*, HT*, VB*, JS*, INF, PP?, OBJ, LIB, PIF, HLP, MD?, INI, MBR, IMG, CSC, CPL, MBP, SH, SHB, SHS, SHT*, CHM, REG, XML, PRC, ASP, LSP, MSO, OBD, THE*, NWS, SWF, MPP, OCX, VS*, DVB, CPY, BMP, RPM, ISO, DEB, AR?, ZIP, R??, GZ, Z, TGZ, TAR, TAZ, CAB, LHA, LZH, BZ2, MSG, EML, 7Z, CPIO. This parameter is opposite to the **-al** parameter.
-ar`[d|m|r][n]`	Instructs to scan files within archives (ARJ, CAB, GZIP, RAR, TAR, ZIP, etc.). An archive is understood to be a tar archive (*.tar) or compressed archive (*.tar.bz2, *.tbz). If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious files in archives. Otherwise, it applies the specified actions to detected threats.
-cn`[d|m|r][n]`	Instructs to scan files within containers (HTML, RTF, PowerPoint). If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious files in containers. Otherwise, it applies the specified actions to detected threats.
-ml`[d|m|r][n]`	Instructs to scan contents of mail files. If additional modifiers (`d`, `m` or `r`) are not specified, **Dr.Web Scanner** only informs the user on detected malicious or suspicious objects. Otherwise, it applies the specified actions to detected threats.
-upn	Scan executable files packed with LZEXE, DIET, PKLITE, EXEPACK without output of the compression type.
-ha	Enables heuristic analysis to detect unknown threats.

For some parameters, you can use the following additional modifiers:

- Add `d` to delete objects to avert the threat
- Add `m` to move objects to **Quarantine** to avert the threat
- Add `r` to rename objects to avert the threat (that is, replace the first character of the file extension with '#')
- Add `n` to disable logging of the archive, container, mail file or packer type

If malicious objects are detected within complex objects such as archives, containers, packed or mail files, the reaction is applied to the whole complex object, and not to the included malicious object only.

Action Parameters

These parameters determine which actions are applied to infected (or suspicious) objects:

Parameter	Description
-cu[d|m|r]	Defines an action applied to infected files and boot sectors. If an additional modifier is not specified, **Dr.Web Scanner** cures infected objects and deletes incurable files (unless another action is specified in the **-ic** parameter). Additional modifiers allow to set another action instead of curing, but the new action can be applied only to infected files. In this case, action for incurable files must be set with **-ic** parameter.
-ic[d|m|r]	Defines an action applied to incurable files. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-sp[d|m|r]	Defines an action applied to suspicious files. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-adw[d|m|r|i]	Defines an action applied to adware. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-dls[d|m|r|i]	Defines an action applied to dialers. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-jok[d|m|r|i]	Defines an action applied to joke programs. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-rsk[d|m|r|i]	Defines an action applied to potentially dangerous programs. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.
-hck[d|m|r|i]	Defines an actionapplied to hacktools. If an additional modifier is not specified, **Dr.Web Scanner** only informs the user about the threat.

Additional modifiers indicate actions that is applied in order to avert threats:

- Add `d` to delete objects.
- Add `m` to move objects to **Quarantine**.
- Add `r` to rename objects, that is, replace the first character of extension with '#'.
- Add `i` to ignore threats (available for minor threats only such as adware etc), that is, apply no action and do not list such threats in the report.

If malicious objects are detected within complex objects such as archives, containers, packed or mail files, the action is applied to the whole complex object, and not to the included malicious object only.

Interface Parameters

These parameters configure **Dr.Web Scanner** output:

Parameter	Description
-v, -version, --version	Instructs to output information on the product and engine versions and exit **Dr.Web Scanner**.
-ki	Instructs to output information about the license and its owner (in `UTF8` encoding only).
-go	Instructs to run **Dr.Web Scanner** in batch mode when all questions implying answers from a user are skipped and all decisions implying a choice are taken automatically. This mode is useful for automatic scanning of files, for example, during a daily (or weekly) check of the hard drive.
-ot	Instructs to use the standard output (`stdout`).
-oq	Disables information output.
-ok	Instructs to list all scanned objects in the report and mark the "clean" object with **Ok**.
-log=[+]<path to file>	Instructs to log **Dr.Web Scanner** operations in the specified file. The file name is required for enabling logging. Add a plus '+' if you want to append the log file instead of overwriting it.
-ini=<path to file>	Instructs to use the specified configuration file. By default, **Dr.Web Scanner** uses **drweb32.ini** (this configuration file is shared by **Dr.Web Daemon**, **Dr.Web Scanner** and **Dr.Web Updater**). **Dr.Web Scanner** uses parameters specified in the `[Scanner]` section of this file. The list of the scanner parameters and available values are similar to the those specified in the `[Daemon]` section.
-lng=<path to file>	Instructs to use the specified language file. The default language is English.
-a = <Control Agent address>	Run **Dr.Web Scanner** in the central protection mode.
-ni	Disables the use of the configuration file for adjusting scanner settings. **Dr.Web Scanner** is configured via command line parameters.
-ns	Disables interruption of scanning process even upon receipt of interruption signals (SIGINT).
--only-key	On startup, only key file is received from **Dr.Web Agent**.

You can use the hyphen «-» postfix (no space) to disable the following parameters:

-ar -cu -ha -ic -fl -ml -ok -sd -sp

For example, if you start **Dr.Web Scanner** with the following command:

```
$ drweb <path> -ha-
```

heuristic analysis (enabled by default) will be disabled.

For the `-cu`, `-ic` and `-sp` parameters, the "negative" form disables any action specified with additional modifiers, that is, information on detection of infected or suspicious object is logged, but no action is performed to avert threats.

The `-al` and `-ex` parameters have no "negative" form, but specifying one of them cancels actions of the other.

By default (if **Dr.Web Scanner** configuration is not customized and no parameters are specified), **Dr.Web Scanner** is started with the following parameters:

-ar -ha -fl- -ml -sd -al -ok

Default **Dr.Web Scanner** parameters (including scan of archives, packed files, files of email programs, recursive search, heuristic analysis and others) are sufficient for everyday diagnostics and can be used in most typical cases. You can also use hyphen «-» postfix to disable required parameters (as it is shown above with an example of heuristic analysis).

Disabling scanning of archives and packed files significantly decreases anti-virus protection level, because viruses are often distributed as archives (especially, self-extracting ones) attached to an email message. Office documents are potentially susceptible to infection with macro viruses (e.g., **Word, Excel**) and can also be dispatched via email within archives and containers.

When you run **Dr.Web Scanner** with default parameters, no cure actions and no actions for incurable and suspicious files are performed. To enable these actions, specify the corresponding command line parameters explicitly.

Configuration

Dr.Web Scanner can be used with default settings, but it could be convenient to configure it according to your needs. **Dr.Web Scanner** settings are stored in the configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory.

To use another configuration file, specify the full path to it as a command line parameter, for example:

```
$ %bin_dir/drweb -ini=%bin_dir/etc/drweb.ini
```

For general principles of the **Dr.Web for Unix Internet gateways** configuration files organization, see Configuration files.

`[Scanner]`

EnginePath = {path to file}	Location of `drweb32.dll` module (anti-virus engine **Dr.Web Engine**). This parameter is also used by **Dr.Web Updater**. Default value: **EnginePath** = `%bin_dir/lib/drweb32.dll`
VirusBase = {list of file masks}	Masks for loading virus databases. This parameter is also used by **Dr.Web Updater**. Multiple values are allowed (separated by commas). By default, virus databases files has a `.vdb` extension Default value: **VirusBase** = `%var_dir/bases/*.vdb`
UpdatePath = {path to directory}	This parameter is used by **Dr.Web Updater** (`update.pl`) and is mandatory. Default value: **UpdatePath** = `%var_dir/updates/`
TempPath = {path to directory}	Directory where anti-virus engine **Dr.Web Engine** stores temporary files. It is used for unpacking archives or when the system is low on memory Default value: **TempPath** = `/tmp/`

LngFileName = {path to file}	Language file location. By default, language files have a `.dwl` extension Default value: **LngFileName** = %bin_dir/lib/ru_scanner.dwl
Key = {path to file}	Key file location (license or demo). By default, key files have a `.key` extension Default value: **Key** = %bin_dir/drweb32.key
OutputMode = {Terminal | Quiet}	Output mode: • `Terminal` – console output • `Quiet` – no output Default value: **OutputMode** = Terminal
HeuristicAnalysis = {logical}	Enables or disables heuristic detection of unknown viruses. Heuristic analysis can detect previously unknown viruses which are not included in the virus database. It relies on advanced algorithms to determine if scanned file structure is similar to the virus architecture. Because of that, heuristic analysis can produce false positives: all objects detected by this method are considered suspicious. Please send all suspicious files to **Dr.Web** through http://vms.drweb.com/sendvirus/ for checking. To send a suspicious file, put it in a password protected archive, include password in the message body and attach **Dr.Web Scanner** report. Default value: **HeuristicAnalysis** = Yes
ScanPriority = {signed numerical value}	**Dr.Web Scanner** process priority. Value must be between `-20` (highest priority) and `19` (**Linux**) or `20` (other UNIX-like operating systems). Default value: **ScanPriority** = 0
FilesTypes = {list of file extensions}	File types to be checked "by type", i.e. when the **ScanFiles** parameter (explained below) has `ByType` value. "*" and "?" wildcard characters are allowed. Default value: FilesTypes = 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
FilesTypesWarnings = {logical}	Notifies about files of unknown types. Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Instructs to scan all files (`All` value) or only files with the extensions specified in the **FileType** parameter (`ByType` value).

	The parameter can have the `ByType` value only in the **local scan** mode. In other modes, the value must be set to `All`. All mail fails are scanned regardless of the **`ScanFiles`** parameter value. Default value: **`ScanFiles`** = `All`
`ScanSubDirectories` = `{logical}`	Enables or disables scanning of subdirectories. Default value: **`ScanSubDirectories`** = `Yes`
`CheckArchives` = `{logical}`	Enables or disables checking of files in archives (`RAR`, `ARJ`, `TAR`, `GZIP`, `CAB` and others). Default value: **`CheckArchives`** = `Yes`
`CheckEMailFiles` = `{logical}`	Enables or disables checking mail files. Default value: **`CheckEMailFiles`** = `Yes`
`ExcludePaths` = `{list of path | file masks}`	Masks for files to be skipped during scanning. Multiple values are allowed (separated by commas). Default value: **`ExcludePaths`** = `/proc,/sys,/dev`
`FollowLinks` = `{logical}`	Allows or forbids **Dr.Web Scanner** to follow symbolic links during scanning. Default value: **`FollowLinks`** = `No`
`RenameFilesTo` = `{mask}`	Mask for renaming files when the `Rename` action is applied. Default value: **`RenameFilesTo`** = `#??`
`MoveFilesTo` = `{path to directory}`	Path to the **Quarantine** directory. Default value: **`MoveFilesTo`** = `%var_dir/infected/`
`EnableDeleteArchiveAction` =`{logical}`	Enables or disables `Delete` action for complex objects (archives, mailboxes, HTML pages) if they contain infected files. Please note, if the action is enabled, a whole complex object is to be deleted. Use this option carefully! Default value: **`EnableDeleteArchiveAction`** = `No`
`InfectedFiles` = `{action}`	Sets one of the following actions upon detection of an infected file: `Report`, `Cure`, `Delete`, `Move`, `Rename`, `Ignore`. `Delete` and `Move` actions are applied to a whole complex object upon detection of infected files within it. Default value: **`InfectedFiles`** = `Report`

SuspiciousFiles = {action}	Sets one of the following actions upon detection of a suspicious file: Report, Delete, Move, Rename, Ignore. Default value: **SuspiciousFiles** = Report
IncurableFiles = {action}	Sets one of the following actions applied if an infected file cannot be cured (use only if **InfectedFiles** = Cure): Report, Delete, Move, Rename, Ignore. Default value: **IncurableFiles** = Report
ActionAdware = {action}	Sets one of the following actions upon detection of adware: Report, Delete, Move, Rename, Ignore. Default value: **ActionAdware** = Report
ActionDialers = {action}	Sets one of the following actions upon detection of a dialer program: Report, Delete, Move, Rename, Ignore. Default value: **ActionDialers** = Report
ActionJokes = {action}	Sets one of the following actions upon detection of a joke program: Report, Delete, Move, Rename, Ignore. Default value: **ActionJokes** = Report
ActionRiskware = {action}	Sets one of the following actions upon detection of a potentially dangerous program: Report, Delete, Move, Rename, Ignore. Default value: **ActionRiskware** = Report
ActionHacktools = {action}	Sets one of the following actions upon detection of a hacktool: Report, Delete, Move, Rename, Ignore. Default value: **ActionHacktools** = Report
ActionInfectedMail = {action}	Sets one of the following actions upon detection of an infected file in a mailbox: Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedMail** = Report
ActionInfectedArchive = {action}	Sets one of the following actions upon detection of an infected file in an archive (ZIP, TAR, RAR, etc.): Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedArchive** = Report

ActionInfectedContainer = {action}	Sets one of the following actions upon detection of an infected file in a container (OLE, HTML, PowerPoint, etc.): Report, Delete, Move, Rename, Ignore. Default value: **ActionInfectedContainer** = Report
	Logging parameters:
LogFileName = {syslog | file name}	Log file name. You can specify syslog as a log file name to use syslogd system service for logging. In this case you must also specify the **SyslogFacility** and **SyslogPriority** parameters. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label which is used by syslogd system service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Log verbosity level when syslogd system service is used. The following levels are allowed: • Error • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
LimitLog = {logical}	Enables or disables limit of log file size (if **LogFileName** value is not set to syslog). With this parameter enabled, **Dr.Web Scanner** checks log file size on startup. If log file size exceeds the **MaxLogSize** parameter value, log file content will be erased and logging will start from scratch. Default value: **LimitLog** = No
MaxLogSize = {numerical value}	Maximum log file size in Kbytes. Used only with **LimitLog** = Yes. If this parameter value is set to 0, log file size is not checked. Default value: **MaxLogSize** = 512
LogScanned = {logical}	Enables or disables logging of information about all scanned objects regardless whether they are infected or not. Default value: **LogScanned** = Yes
LogPacked = {logical}	Enables or disables logging of additional information about files packed with DIET, PKLITE and other utilities.

	Default value: **LogPacked** = Yes
LogArchived = {logical}	Enables or disables logging of additional information about files archived with various archiving utilities. Default value: **LogArchived** = Yes
LogTime = {logical}	Enables or disables logging of time for each record. Parameter is not used if **LogFileName** = syslog. Default value: **LogTime** = Yes
LogStatistics = {logical}	Enables or disables logging of scan statistics. Default value: **LogStatistics** = Yes
RecodeNonprintable = {logical}	Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters). Default value: **RecodeNonprintable** = Yes
RecodeMode = {Replace | QuotedPrintable}	Decoding mode for non printable characters if **RecodeNonprintable** = Yes. When **RecodeMode** = Replace, all non-printable characters are substituted with the **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable, all non-printable characters are converted to the Quoted Printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets character for replacing non-printable characters if **RecodeMode** = Replace. Default value: **RecodeChar** = "?"

The following parameters can be used to reduce time of scanning archives (by skipping some objects in an archive).

MaxCompressionRatio = {numerical value}	Maximum compression ratio, that is ratio between size of unpacked file and its size within an archive. If a ratio exceeds the specified value, the file will not be extracted and therefore will not be checked. An email message with such an archive is considered as a "mail bomb". Parameter can have only natural values. If the value is set to 0, compression ratio will not be checked Default value: **MaxCompressionRatio** = 5000
CompressionCheckThreshold = {numerical value}	Minimum size of a file enclosed within an archive, in Kbytes. If a file size is less than the specified value, the compression ratio will not be checked (if such a check is enabled by the **MaxCompressionRatio** parameter).

	Default value: **CompressionCheckThreshold** = 1024
MaxFileSizeToExtract = {numerical value}	Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped. An email message with such a file is considered as a "mail bomb". Default value: **MaxFileSizeToExtract** = 500000
MaxArchiveLevel = {numerical value}	Maximum archive nesting level. If an archive nesting level exceeds the specified value, the archive is skipped. An email message with such a file is considered as a "mail bomb". If the value is set to 0, archive nesting level will not be checked Default value: **MaxArchiveLevel** = 8
MaximumMemoryAllocationSize = {numerical value}	Maximum size of the memory (in Mbytes) that can be used by **Dr.Web Scanner** to check one file. If the value is set to 0, memory allocation is not limited. Default value: **MaximumMemoryAllocationSize** = 0
ScannerScanTimeout = {numerical value}	Maximum time period allowed for scanning one file (in seconds). If the value is set to 0, scanning time is not limited. Default value: **ScannerScanTimeout** = 0
MaxBasesObsolescencePeriod = {numerical value}	Maximum time (in hours) after last update when virus databases are considered as up-to-date. Upon the expiration of this time period, notification displays informing that the databases are obsolete. If the value is set to 0, database actuality will not be checked. Default value: **MaxBasesObsolescencePeriod** = 24
ControlAgent = {address}	**Dr.Web Agent** socket address. **Example**: **ControlAgent** = inet:4040@127.0.0.1,local:%var_dir/ipc/.agent **Dr.Web Scanner** receives a license key file and configuration from **Dr.Web Agent**. (if **OnlyKey** = No). Default value: **ControlAgent** = local:%var_dir/ipc/.agent
OnlyKey = {logical}	Enables receiving only a license key file from **Dr.Web Agent**, without configuration. At that, **Dr.Web Scanner** uses the local configuration file. If the value is set to No and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Agent** also receives statistics on **Dr.Web Scanner** operation (information is sent after scanning of each file).

	Default value: **OnlyKey** = No

Exit Codes

When the scan task ends, **Dr.Web Scanner** returns an exit code which determines result of scanning.

The exit code is always constructed as an combination (sum) of codes that are related to the corresponding events of scanning process. The possible events and related codes are following:

Code	Event
1	Known virus detected
2	Modification of known virus detected
4	Suspicious object found
8	Known virus detected in archive, mailbox or other container
16	Modification of known virus detected in archive, mailbox or other container
32	Suspicious file found in archive, mailbox or other container
64	At least one infected object succesfully cured
128	At least one infected or suspicious file deleted/renamed/moved

The actual value returned by **Dr.Web Scanner** is equal to the sum of codes for the events that occurred during scanning. Obviously, the sum can be easily decomposed into separate event codes. For example, return code 9 = 1 + 8 means that known viruses were detected, including viruses in archives, mail archives or containers; curing and others actions were not executed; no other threat events occurred during scanning.

If no threat events occurred during scanning, **Dr.Web Scanner** returns the exit code 0.

Dr.Web Scanner has one feature: in some cases, when no threats were found during scanning, it can return the exit code 128 instead of exit code 0. This case is similar to the case "no threats found" (exit code 0).

Dr.Web Daemon

Dr.Web Daemon is a background anti-virus module **drwebd**, designed to perform scanning for viruses on request received from other **Dr.Web** components. It can scan files on the disk or data transferred through a socket. Requests for anti-virus scanning are sent using a special protocol via UNIX or TCP sockets. **Dr.Web Daemon** uses the same anti-virus engine (**Dr.Web Engine**) and virus databases, like **Dr.Web Scanner**, and is able to detect and cure all known viruses.

Dr.Web Daemon is always running and has simple and intelligible protocol for sending scanning requests, which makes it a perfect solution to be used as an anti-virus filter for file servers. **Dr.Web for Unix Internet gateways** is a ready-made solution for integrating **Dr.Web Daemon** with for Internet gateways. In the **Dr.Web for Unix Internet gateways** solution, **Dr.Web Daemon** is integrated with applications that use ICAP protocol..

Note that **Dr.Web Daemon** cannot scan the contents of the encrypted files because in this case it is necessary to know the password that been used for encryption. So, these files will be passed without the scan, and for the client application the special return code will be returned.

Command-Line Parameters

To run **Dr.Web Daemon**, use the following command:

```
drwebd [parameters]
```

where the following `parameters` are available:

Short case	Extended case	Arguments
-h, -?	-help, --help	
Description: Show information about supported command line parameters on the screen and terminate the module		
-a		<Agent socket address>
Description: Start **Dr.Web Daemon** in the central protection mode under control of the specified copy of **Dr.Web Agent**		
-ini		<path to file>
Description: Module must use the specified configuration file		
	--foreground	<yes|no>
Description: Operation mode of **Dr.Web Daemon**. If `yes` is specified, **Dr.Web Daemon** is a foreground process. Otherwise (`no`), **Dr.Web Daemon** is a background process		
	--check-only	<command line parameters for checking>
Description: Check **Dr.Web Daemon** configuration correctness on startup. If any command line parameter is specified, correctness of the value is also checked		
	--only-key	
Description: On startup, **Dr.Web Daemon** receives from **Dr.Web Agent** only the license key file		

Running Dr.Web Daemon

When **Dr.Web Daemon** is started with the default settings, the following actions are performed:

- Search and load of the configuration file. If the configuration file is not found, loading of **Dr.Web Daemon** terminates. Path to the configuration file can be specified on startup with the `-ini` command line parameter: `{path/to/your/drweb32.ini}`, otherwise, the default value (`%etc_dir/drweb32.ini`) can be used. On startup, correctness of several configuration parameters is checked, and if a parameter value is incorrect, the default parameter value is set;
- Creation of a log file. A user account under which **Dr.Web Daemon** is started must have appropriate privileges to write to the log file directory. Users do not have write permission for the default log directory (`/var/log/`). Therefore, if the **User** parameter is specified, adjust the **LogFileName** parameter and provide alternative log file directory;
- Load of a key file from the location specified in the configuration file. If the key file is not found, loading of **Dr.Web Daemon** terminates;
- If the **User** parameter is specified, **Dr.Web Daemon** attempts to change its privileges;
- Load of **Dr.Web Engine** (**drweb32.dll**). If **Dr.Web Engine** is damaged or not found (because of errors in the configuration file), initialization of **Dr.Web Daemon** terminates;
- Load of virus databases in arbitrary sequence from the location specified in the configuration file. If virus databases are damaged or absent, initialization of **Dr.Web Daemon** proceeds;
- **Dr.Web Daemon** enters daemon mode, so all information about initialization problems cannot be output to the console and is logged to the log file;
- Creation of a socket for interaction between **Dr.Web Daemon** and other **Dr.Web for Unix Internet gateways** modules. When TCP-sockets are used, there can be several connections (loading continues if at least one connection is established). When a UNIX socket is used, **Dr.Web Daemon** user account must have appropriate privileges to read and write from the directory of this socket. User accounts for modules must have execution access to the directory and write and read access to the socket file. Users do not have write permission for the default socket directory (`/var/run/`). If the **User** parameter is specified, adjust the **Socket** parameter and provide alternative path to the socket file. If creation of the UNIX socket was unsuccessful, initialization of **Dr.Web Daemon** terminates;
- Creation of a PID file with **Dr.Web Daemon** PID information and transport addresses. User account under which **Dr.Web Daemon** is started must have appropriate privileges to write to the directory of the PID file. Users do not have write permission for the default socket directory (`/var/run/`). So, if the **User** parameter is specified, adjust the **PidFile** parameter and provide alternative path to the PID file. If creation of the PID file was unsuccessful, initialization of **Dr.Web Daemon** terminates.

Dr.Web Daemon Testing and Diagnostics

If no problems occurred during initialization, **Dr.Web Daemon** is ready to use. To ensure that the daemon is initialized correctly, use the following command:

```
$ netstat -a
```

and check whether required sockets are created.

TCP sockets:

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

Unix socket:

```
. . .
Active UNIX domain sockets (servers and established)
Proto   RefCnt   Flags   Type    State      I-Node   Path
. . .
unix    0       [ ACC ]  STREAM  LISTENING  1127     %var_dir/.daemon
. . .
```

Missing of the required sockets in the list indicates problems with **Dr.Web Daemon** initialization.

To perform a functional test and obtain service information, use **Dr.Web Daemon console client** (**drwebdc**).

TCP sockets:

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

Unix socket:

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

Report, similar to the following example, is output to the console:

```
- Version: DrWeb Daemon 6.00
- Loaded bases:
Base /var/drweb/bases/drwtoday.vdb
contains 5 records.
Base /var/drweb/bases/drw60003.vdb
contains 409 records.
Base /var/drweb/bases/drw60002.vdb
contains 543 records.
Base /var/drweb/bases/drwebase.vdb
contains 51982 records.
Base /var/drweb/bases/drw60001.vdb
contains 364 records.
Total 53303 virus-finding records.
```

If the report was not output, run extended diagnostics.

For TCP socket:

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

For UNIX socket:

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

More detailed report can help to identify the problem:

```
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
```

You can test **Dr.Web Daemon** with the special **`eicar.com`** program included in the installation package. Use any text editor to transform `readme.eicar` into `eicar.com` (see instructions within the file).

For TCP-socket:

```
$ drwebdc -n<HOST> -p<PORT> eicar.com
```

For UNIX socket:

```
$ drwebdc -u<SOCKETFILE> eicar.com
```

The following result are output:

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

If the results were not output, check **Dr.Web Daemon** log file to see whether the file was scanned. If the file was not scanned, run extended diagnostic (see above).

If file was scanned successfully, **Dr.Web Daemon** is fully operational.

When scanning very large archives, some issues with timeout expiration may occur. To fix this, increase values of the `FileTimeout` and `SocketTimeout` parameters.

Please note that **Dr.Web Daemon** cannot scan files larger than **2 Gbytes**. Such files will not be sent for scanning.

Scan Modes

Dr.Web Daemon has two scan modes:

- scan of chunks received from the socket (**remote scan mode**);
- scan of files on the disk (**local scan mode**).

In the **remote scan mode**, client sends data to be scanned to **Dr.Web Daemon** through a socket. **Dr.Web Daemon** can scan both anonymous memory and memory mapped objects with only one difference - in logging. This mode enables scanning of files without read access but is less efficient than the local scan mode.

Local scan mode is easier to use and provides better performance since client sends to **Dr.Web Daemon** only a file path instead of the file. For the reason that clients can be located on different computers, the path must be specified in relation to the actual location of **Dr.Web Daemon**.

Local scan mode requires careful configuration of user privileges. **Dr.Web Daemon** must have read access to each file that is to be scanned. To perform **`Cure`** and **`Delete`** actions to files in mailboxes, you must also permit write access.

If the system is configured correctly, **Dr.Web Daemon** does not require `root` superuser privileges..

If required, name of the user with whose privileges **Dr.Web Daemon** must run is set as the **`User`** parameter value in **Dr.Web Daemon** settings. In addition, you can configure user and their group used on module startup. For that purpose, edit mmc-file of **Dr.Web Monitor** if it is used for management of **Dr.Web for Unix Internet gateways** components.

Processed Signals

Dr.Web Daemon can receive and process the following signals:

- `SIGHUP` – reload the configuration file;
- `SIGTERM` – correct termination of **Dr.Web Daemon**;
- `SIGKILL` – force termination of **Dr.Web Daemon** (if any problem occurs);
- `SIGUSR1` – save process pool statistics to the log file.

Please note that `SIGUSR1` signal must be sent to its parent process only, because child processes are terminated after receiving of `SIGUSR1`.

Log Files and Statistics

Daemon Log

Since **Dr.Web Daemon** is a resident program, information on its operation can be obtained only from a log file. Log file contains details on processing of all scanning request sent to **Dr.Web Daemon**. You can specify the log file location in a value of the **`LogFileName`** parameter.

Dr.Web Daemon can log information to different files depending on a client that sent the request. You can specify different log files for every **Dr.Web** clients (for example, **Dr.Web for Unix Internet gateways**) in the **`ClientsLogs`** parameter value.

Regardless of the **`ClientsLogs`** parameter, if **Dr.Web Daemon** recognizes its client, scanning results will marked with a prefix indicating the client. The following prefixes are available:

- `<web>` – **Dr.Web ICAPD**;
- `<smb_spider>` – **Dr.Web Samba SpIDer**;
- `<mail>` – **Dr.Web MailD**;
- `<drwebdc>` – console client for **Dr.Web Daemon**;
- `<kerio>` – **Dr.Web for Kerio Internet Gateways**;
- `<lotus>` – **Dr.Web for IBM Lotus Domino**.

In the **FreeBSD** operating system, **`syslog`** service can intercept information output by **Dr.Web Daemon** to the console. In this case, the information is logged character-by-character. That occurs when the logging level is set to `*.info` in the **`syslog`** configuration file (`syslog.conf`).

Statistics on process pool

Statistics on pool used for processing scanning request is output to the log file upon receipt of `SIGUSR1` signal (the signal must be sent only to parent process, as if a child process receives `SIGUSR1`, it terminates).

Output of statistics on process pool is regulated by the **`stat`** value (`yes` or `no`), specified for the **`ProcessesPool`** parameter. Collected statistics is not aggregated. Each time the saved record

contains statistics on the pool state between previous and current moment of saving.

Example of pool statistics output record:

```
Fri Oct 15 19:47:51 2010 processes pool statistics: min = 1 max = 1024
(auto) freetime = 121 busy max = 1024 avg = 50.756950 requests for new
process = 94 (0.084305 num/sec) creating fails = 0 max processing time =
40000 ms; avg = 118646 ms curr = 0 busy = 0
```

where:

- `min` – minimal number of processes in the pool;
- `max` – maximal number of processes in the pool;
- `(auto)` – displays if limits on number of processes in the pool are determined automatically;
- `freetime` – maximum idle time for a process in the pool;
- `busy max` – maximum number of simultaneously used processes, `avg` - average number of simultaneously used processes;
- `requests for new process` – number of requests for new process creation (frequency of requests per second is displayed in parenthesis);
- `creating fails` – number of failed attempts to create a new process (failures usually occur when the system is running low on resources);
- `max processing time` – maximum time for processing a single scanning request;
- `avg` – average time for processing a single scanning request;
- `curr` – number of all current processes in the pool;
- `busy` – number of currently used processes in the pool.

Configuration

Dr.Web Daemon can be run with default settings, but you can configure it according to your specific requirements. **Daemon** settings are stored in the `[Daemon]` section of the configuration file (`drweb32.ini` by default) which is located in `%etc_dir` directory. To use another configuration file, specify the full path to it as a command-line option.

[Daemon]

EnginePath = {path to file}	Location of `drweb32.dll` module (anti-virus engine **Dr.Web Engine**). This parameter is also used by the **Dr.Web Updater**. Default value: **EnginePath** = `%bin_dir/lib/drweb32.dll`
VirusBase = {list of files (masks)}	Masks for virus databases. This parameter is also used by **Dr.Web Updater**. Multiple values are allowed (separated by commas). By default, virus databases files has the `.vdb` extension Default value: **VirusBase** = `%var_dir/bases/*.vdb`
UpdatePath = {path to directory}	Directory to store updates. The parameter is mandatory. Default value: **UpdatePath** = `%var_dir/updates/`
TempPath = {path to directory}	Directory where the **Dr.Web Engine** anti-virus engine puts temporary files.

	It is used when system has insufficient memory or to unpack certain types of archives. Default value: **TempPath** = %var_dir/spool/
Key = {path to file}	Key file location (license or demo). By default, a key file has the .key extension. Please note that **Dr.Web Daemon** and **Dr.Web Scanner** can have different license key files. In this case, change the value of this parameter correspondingly. The parameter value can be set several times to specify several license key files. In this case, **Dr.Web Daemon** tries to combine all license permissions from all available license key files. Default value: **Key** = %bin_dir/drweb32.key
OutputMode = {Terminal | Quiet}	Output mode: • Terminal – console output • Quiet – no output Default value: **OutputMode** = Terminal
RunForeground = {logical}	Allows to disable or enable daemon mode for **Dr.Web Daemon**. With Yes value specified **Dr.Web Daemon** runs as a foreground process. This parameter can be used for certain monitoring utilities (for example, **Dr.Web Monitor**). Default value: **RunForeground** = No
User = {text value}	User under which **Dr.Web Daemon** operates. It is strongly recommended to create a separate **drweb** user account, which will be used by **Dr.Web Daemon** and filters. It is not recommended to run **Dr.Web Daemon** with root privileges, even though it may take less time to configure. This parameter cannot be changed when reloading configuration using SIGHUP. Default value: **User** = drweb
PidFile = {path to file}	File to store **Dr.Web Daemon**'s PID and UNIX socket (if it is enabled by the **Socket** parameter) or port number (if TCP socket is enabled by the **Socket** parameter). If more than one **Socket** parameter is specified, this file contains information on all the sockets (one per line). This file is created every time **Dr.Web Daemon** starts. Default value: **PidFile** = %var_dir/run/drwebd.pid
BusyFile = {path to file}	File where **Dr.Web Daemon** busy flag is stored. This file is created by a **Dr.Web Daemon** child process upon receipt of the scan command and is removed after successful command execution.

	Filenames created by each **Dr.Web Daemon** child process are appended by a dot and ASCII representation of the PID (for example, `/var/run/drwebd.bsy.123456`). Default value: **`BusyFile`** = `%var_dir/run/drwebd.bsy`
`ProcessesPool` = `{process pool settings}`	Settings of dynamic process pool. At first, specify the number of processes in the pool: • `auto` – number of processes is set automatically depending on system load; • `N` – nonnegative integer. Pool will have at least `N` active processes, additional processes will be created if necessary; • `N-M` – positive integer, `M>=N`. The pool will have at least `N` active processes, additional processes will be created if necessary, but maximum total number of processes cannot exceed `M`. Then specify optional secondary parameters: • **`timeout`** = `{time in seconds}` – timeout for closing an inactive process. This parameter does not affect the first `N` processes which wait for requests indefinitely. • **`stat`** = `{yes|no}` – statistics on processes in a pool. If `yes`, it is saved to the log file each time `SIGUSR1` system signal is received. • **`stop_timeout`** = `{time in seconds}` – maximum time to wait for a running process to stop. Default value: **`ProcessesPool`** = `auto`,**`timeout`** = `120`, **`stat`** = `no`, **`stop_timeout`** = `1`
`OnlyKey` = `{logical}`	Enables receiving only a license key file from **Dr.Web Agent**, without configuration. At that, **Dr.Web Scanner** uses the local configuration file. If the value is set to `No` and the address of a **Dr.Web Agent** socket is specified, **Dr.Web Daemon** sends operational statistics to **Dr.Web Agent** (information is sent after scanning of every file). Default value: **`OnlyKey`** = `No`
`ControlAgent` = `{address}`	**Dr.Web Agent** socket address. **Example**: **`ControlAgent`** = `inet:4040@127.0.0.1,local:%var_dir/ipc/.agent` **Dr.Web Daemon** receives from **Dr.Web Agent** a license key file (and configuration if **`OnlyKey`** = `No`. Moreover, in this case the socket is used for sending statistics on **Dr.Web Daemon** operation to **Dr.Web Agent**). Default value: **`ControlAgent`** = `local:%var_dir/ipc/.agent`
`MailCommand` = `{string}`	Shell command used by **Dr.Web Daemon** and **Dr.Web Updater** for sending notifications on new updates to the user (administrator) via email. If the period before the key file (or one of the key files) expiration

	is less than the period specified by the **NotifyPeriod** parameter, **Dr.Web Daemon** starts sending notifications upon every system startup, restart or reboot. Default value: **MailCommand** = "/usr/sbin/sendmail -i -bm -f drweb -- root"
NotifyPeriod = {numerical value}	This parameter value specifies the period (in days) before license key expiration date when **Dr.Web Daemon** starts prompting a user to renew the license. If the parameter value is set to 0, **Dr.Web Daemon** starts sending out notifications immediately after the key file expires. Default value: **NotifyPeriod** = 14
NotifyFile = {path to file}	Path to the file with a timestamp of the last license expiration notification. Default value: **NotifyFile** = %var_dir/.notify
NotifyType = {Ever | Everyday | Once}	Frequency of sending license expiration notifications. • Once – notification is sent only once. • Everyday – notification is sent daily. • Ever – notification is sent upon every **Dr.Web Daemon** restart and every database update. Default value: **NotifyType** = Ever
FileTimeout = {numerical value}	Maximum time (in seconds) allowed for **Dr.Web Daemon** to perform scanning of one file. If the parameter value is set to 0, time to scan of one file is unlimited. Default value: **FileTimeout** = 30
StopOnFirstInfected = {logical}	Enables or disables interruption of file scanning upon detection of the first virus. If the value is set to yes, it can significantly reduce mail server load and scan time. Default value: **StopOnFirstInfected** = No
ScanPriority = {signed numerical value}	Priority of **Dr.Web Daemon** process. Value must be in the following range: -20 (highest priority) to 19 (lowest priority for **Linux**) or 20 (lowest priority for **FreeBSD** and **Solaris**). Default value: **ScanPriority** = 0
FilesTypes = {list of file extensions}	Types of files to be checked "by type", that is, when the **ScanFiles** parameter value (described below) is set to ByType. "*" and "?" wildcard characters are allowed.

	Default value: **FilesTypes** = 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
FilesTypesWarnings = {logical}	Notify on files of unknown types Default value: **FilesTypesWarnings** = Yes
ScanFiles = {All | ByType}	Scan only files with extensions specified in the **FileTypes** parameter (the ByType value) or all files (the All value). This parameter can have the ByType value only in the **local scan** mode (in other modes, only the All value can be set). In mailboxes, all files are always checked (regardless of the **ScanFiles** parameter value). Default value: **ScanFiles** = All
CheckArchives = {logical}	Enables or disables checking of files in archives. The following formats are supported: ZIP (WinZip, InfoZIP, etc.), RAR, ARJ, TAR, GZIP, CAB and others. Default value: **CheckArchives** = Yes
CheckEMailFiles = {logical}	Enables or disables checking of email files. Default value: **CheckEMailFiles** = Yes
ExcludePaths = {list of path | file masks}	Masks for files to be skipped during scanning. Default value: **ExcludePaths** = /proc,/sys,/dev
FollowLinks = {logical}	Enables or disables **Dr.Web Daemon** to follow symbolic links during scanning. Default value: **FollowLinks** = No
RenameFilesTo = {mask}	Mask for renaming files when the Rename action is applied. Default value: **RenameFilesTo** = #??
MoveFilesTo = {path to directory}	Path to the **Quarantine** directory. Default value: **MoveFilesTo** = %var_dir/infected/
BackupFilesTo = {path to directory}	Directory for backup copies of cured files. Default value: **BackupFilesTo** = %var_dir/infected/

LogFileName = {syslog | file name}	Log file name. You can specify syslog as a log file name and logging will be performed by syslogd system service. In this case, also specify the **SyslogFacility** and **SyslogPriority** parameter values. Default value: **LogFileName** = syslog
SyslogFacility = {syslog label}	Log type label used by syslogd system service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Logging priority (log verbosity level) when syslogd system service is used. There are the following levels allowed: • Error • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
LimitLog = {logical}	Enables or disables limit for log file size (if **LogFileName** value is not specified to syslog). If limit is enabled, **Dr.Web Daemon** checks the size of a log file on startup or on receipt of HUP signal. If the log file size is greater than MaxLogSizevalue, the log file is overwritten with an empty file and logging starts from scratch. Default value: **LimitLog** = No
MaxLogSize = {numerical value}	Maximum log file size in Kbytes. Used only with **LimitLog** = Yes. Set this parameter value to 0 if you do not want a log file to be unexpectedly modified on startup. Default value: **MaxLogSize** = 512
LogScanned = {logical}	Enables or disables logging of information about all scanned objects regardless whether they are infected or not. Default value: **LogScanned** = Yes
LogPacked = {logical}	Enables or disables logging of additional information about files packed with DIET, PKLITE and other utilities. Default value: **LogPacked** = Yes
LogArchived = {logical}	Enables or disables logging of additional information about files archived with various archiving utilities.

	Default value: **LogArchived** = Yes
LogTime = {logical}	Enables or disables logging of time for each record. The parameter is not used if **LogFileName** = syslog. Default value: **LogTime** = Yes
LogProcessInfo = {logical}	Enables or disables logging PID of the scanning process and filter address (host name or IP address) from which scanning has been activated. This data is logged before each record. Default value: **LogProcessInfo** = Yes
RecodeNonprintable = {logical}	Enables or disables transcoding of characters that are undisplayable on a given terminal (see also the description of the following two parameters). Default value: **RecodeNonprintable** = Yes
RecodeMode = {Replace | QuotedPrintable}	Decoding mode for non-printable characters (if **RecodeNonprintable** = Yes). When **RecodeMode** = Replace, all non-printable characters are substituted with the **RecodeChar** parameter value (see below). When **RecodeMode** = QuotedPrintable, all non-printable characters are converted to Quoted Printable encoding. Default value: **RecodeMode** = QuotedPrintable
RecodeChar = {"?" | "_" | ...}	Sets a character to replace all non-printable characters if **RecodeMode** = Replace. Default value: **RecodeChar** = "?"
Socket = {address list}	List of sockets to be used for communication with **Dr.Web Daemon** (separated by commas). **Example**: Socket = inet:3000@127.0.0.1,local:%var_dir/.daemon You can also specify a socket address in the following format: PORT [interfaces] | FILE [access]. For a TCP socket, specify a decimal port number (PORT) and the list of interface names or IP addresses for incoming requests (interfaces). **Example:** **Socket** = 3000 127.0.0.1, 192.168.0.100 For UNIX sockets, specify a socket name (FILE) and access permissions in the octal form. **Example:** **Socket** = %var_dir/.daemon 0660 Number of **Socket** parameter values is not limited. **Dr.Web Daemon** will work with all sockets described correctly.

	To enable connections on all available interfaces, set 3000 0.0.0.0 as a value of this parameter. Default value: **Socket** = %var_dir/run/.daemon
SocketTimeout = {numerical value}	Maximum time (in seconds) allowed for transferring data through socket (file scanning time is not included). If the parameter value is set to 0, the time is unlimited. Default value: **SocketTimeout** = 10
ClientsLogs = {string list}	Enables splitting of log files. If during communication with **Dr.Web Daemon** a client uses the option to transfer its ID, log file will be substituted with the file specified in this parameter. Descriptions of log files are separated by commas or spaces. If more than six values are set, the configuration file is considered invalid. Log files are defined in the following way: <client name1>:<path to file>, <client name2>:<path to file> Client name may be one of the following: • web — **Dr.Web ICAPD**; • smb_spider — **Dr.Web Samba SpIDer**; • mail — **Dr.Web MailD**; • drwebdc — console client for **Dr.Web Daemon**; • kerio — **Dr.Web for Kerio Internet Gateways**; • lotus — **Dr.Web for IBM Lotus Domino**. **Example**: drwebdc:/var/drweb/log/drwebdc.log, smb:syslog, mail:/var/drweb/log/drwebmail.log Default value:
MaxBasesObsolescencePeriod = {numerical value}	Period, in hours, after last update, during which virus databases are considered up-to-date. When this period is over, a message notifying that databases are obsolete is output. If value is set to 0, database obsolescence is not checked. Default value: **MaxBasesObsolescencePeriod** = 24

The following parameters can be used to reduce scanning time in archived files (some objects in archives are not checked). Actions applied to skipped depend on the **ArchiveRestriction** parameter value of the corresponding modules.

MaxCompressionRatio = {numerical value}	Maximum compression ratio, that is a ratio between size of unpacked file and its size within an archive. The parameter can have only natural values. If the ratio exceeds

	the specified value, file will not be extracted and therefore will not be checked. Value of this parameter must be not less than 2. Default value: **MaxCompressionRatio** = 5000
CompressionCheckThreshold = {numerical value}	Minimum size of a file enclosed within an archive (in Kbytes) for which compression ratio check is performed (if such a check is enabled by the **MaxCompressionRatio** parameter). Value of this parameter must be greater than 0. Default value: **CompressionCheckThreshold** = 1024
MaxFileSizeToExtract = {numerical value}	Maximum size of a file enclosed in an archive, in Kbytes. If a file size exceeds the specified value, the file is skipped. Default value: **MaxFileSizeToExtract** = 40960
MaxArchiveLevel = {numerical value}	Maximum allowed archive nesting level. If an archive nesting level exceeds the specified value, an archive is not scanned. Default value: **MaxArchiveLevel** = 8
MessagePatternFileName = {path to file}	Path to template for a license expiration message. You can configure output of an expiration message according to your needs. To do this, use the following variables in the template. The specified variables are substituted with the corresponding values: • $EXPIRATIONDAYS — number of days left until license expiration; • $KEYFILENAME — path to license key file; • $KEYNUMBER — license number; • $KEYACTIVATES — license activation date; • $KEYEXPIRES — license expiration date. If there is no user-defined template, standard message in English is output. Default value: **MessagePatternFileName =** %etc_dir/templates/drwebd/msg.tmpl
MailTo = {email address}	Email address of an administrator where the following information is sent: messages about license expiration, virus databases obsolescence, etc. Default value: **MailTo =**

Dr.Web ICAPD

Dr.Web ICAPD module (**drweb-icapd**) allows integration of all **Dr.Web for Unix Internet gateways** components with applications which use the ICAP protocol. This protocol is currently supported by **Squid** and **SafeSquid** proxy servers.

Dr.Web ICAPD establishes connection between **Dr.Web Daemon** and the corresponding proxy server to enable scanning of incoming FTP and HTTP traffic for viruses. It also allows filtering access to HTML resources by both the MIME type and size of downloaded files and the name of the host where these files reside. Moreover, it is possible to restrict access to webpages with the use of content-specific black lists, which are regularly updated, and white and black lists defined by the user (administrator of the suite).

Interaction scheme:

1) Client requests an Internet resource (with a `HTTP GET` request);
2) Proxy server requests **Dr.Web ICAPD** a permission to access the required resource via the ICAP protocol;
3) If access to the requested resource is not forbidden (for example, if the user added the server to the white list, or this server is not included in the user-defined black list and in the content-specific black lists, or if the applied rules allow access to the resource), **Dr.Web ICAPD** does not block the request. Otherwise, **Dr.Web ICAPD** instructs the proxy server to respond with an HTML page notifying that access to the requested resource is blocked;
4) If access to the remote server is allowed, the proxy server connects to it, receives response and then, via the ICAP protocol, transmits the received content to **Dr.Web ICAPD** for anti-virus scanning;
5) If the user added the remote server to the white list, the received content is not checked and **Dr.Web ICAPD** instructs the proxy server to transmit the content to the client. Otherwise, **Dr.Web ICAPD** checks the content with the use of content-filtering rules, and, if the rules instruct to apply a **scan** action, the content is transmitted to **Dr.Web Daemon** for anti-virus scanning;
6) According to the received results, one of the following actions is applied to to the requested content:
   - **pass** – the requested content is returned to the client;
   - **report** – an HTML page notifying that the requested file is rejected displays;
   - **move** – **Dr.Web ICAPD** moves the received file to **Quarantine** and instructs the proxy server to return an HTML page notifying that the requested file is quarantined;
   - **truncate** – an empty file is returned to the client.

The same actions (except for moving to **Quarantine**) can be specified in content-filtering rules. Thus, you can instruct **Dr.Web ICAPD** to pass content of certain types without scanning, or, on the contrary, to reject it unconditionally. For that purpose, enable and configure the ICAP preview mode.

Configuring Interaction between Dr.Web ICAPD and Squid

General scheme of interaction between a **Squid** proxy server, **drweb-icapd** and a client is as follows:

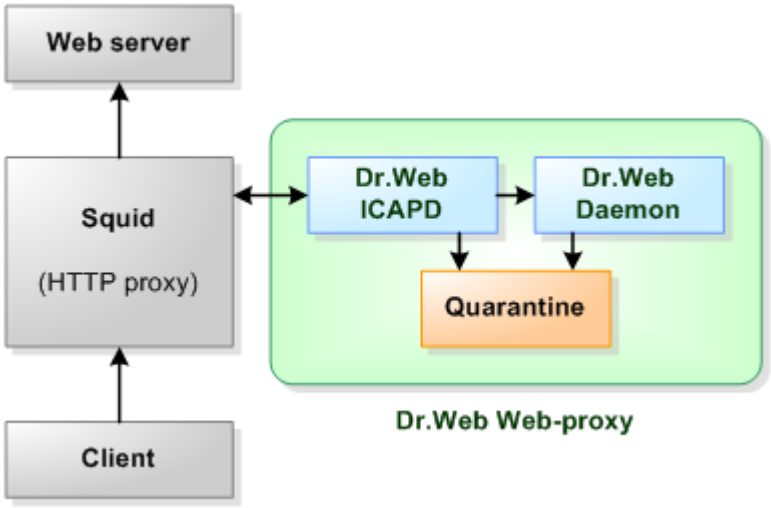


Figure 16. General interaction scheme

In this scheme, a client interacts with an HTTP server through a proxy server. The proxy server is a client of the ICAP server (**Dr.Web ICAPD**). **Dr.Web ICAPD** module, in turn, is a client of **Dr.Web Daemon**. **Dr.Web ICAPD** allows to perform virus scanning (using **Dr.Web Daemon**) of all HTTP traffic coming from the HTTP server and transmitted by the proxy server via the ICAP protocol. The given scheme does not allow FTP traffic scanning. For details on how to enable scanning of FTP traffic, refer to the Configuring Squid to Scan FTP Traffic section.

Note that HTTPS traffic is not scanned as it is encrypted and cannot be decrypted without the public key of the HTTPS server.

To enable **Squid** to use **Dr.Web ICAPD**, edit the `squid.conf` configuration file (usually located at `/usr/local/squid/etc`) to allow usage of the ICAP protocol.

For this purpose, uncomment the lines mentioned below and edit the specified default values if necessary. If the lines are not present, add them to the end of the configuration file:

1. Enable usage of the ICAP protocol:

```
icap_enable on
```

2. Register new ICAP service:

For Squid 3.0:

```
icap_service service_1 respmod_precache 0 icap://127.0.0.1:1344/respmod
icap_class class_1 service_1
icap_access class_1 allow all
```

For Squid 3.1:

```
icap_service service_1 respmod_precache bypass=0 icap://127.0.0.1:1344/respmod
adaptation_access service_1 allow all
```

Please note that address and port that are specified in **icap_service**, must be equal to the corresponding values of **BindAddress** and **BindPort** parameters of the **Dr.Web ICAPD** configuration file.

When ICAP preview mode is enabled, configure additional settings.

3. Enable the ICAP preview mode:

```
icap_preview_enable on
```

4. Specify size of the message (in bytes) sent to ICAP preview:

```
icap_preview_size 0
```

Note that specifying any value other than `0` for the `icap_preview_size` parameter when `ICAP preview` is enabled, as well as specifying any value other than `-1` when `ICAP preview` is disabled has no effect (that is, size of previewed objects currently cannot be adjusted).

5. If necessary, enable logging the IP address of the client that requested resource:

```
icap_send_client_ip on
```

6. If necessary, enable persistent connections between drweb-icapd and Squid, which improves performance:

```
icap_persistent_connections on
```

`respmod-postcache` mode is not implemented in the current version of **Squid**, thus checking of cached content is not possible.

Configuring Interaction between Dr.Web ICAPD and SafeSquid

To enable **SafeSquid** to use **Dr.Web ICAPD**, edit the `config.xml` configuration file manually or use the web interface.

When using the web interface, select the ICAP section in the drop-down menu and then select the **Add** item to add a new ICAP interface. In the open window, specify the following information in the corresponding fields:

- **Enabled** = `true`;
- **Host** = `IP address or hostname where` **drweb-icapd** `is running (127.0.0.1` by default);
- **File** = `/respmod`;
- **Port** = `drweb-icapd listening port number (1344` by default);
- **Applies to** = `responses`;

After that, click the **Submit** button.

You can also edit `config.xml` manually. For example, you can add the following block to the **`<safesquid></safesquid>`** section:

```
<icap>
  <enabled>true</enabled>
  <icap>
      < enabled>true</enabled>
      < comment>Dr.Web icap server</comment>
      < profiles></profiles>
      < host>127.0.0.1</host>
      < file>/respmod</file>
      < port>1344</port>
      < which>responses</which>
  </ icap>
</icap>
```

Please note that host and port must be equal to the corresponding values of **`BindAddress`** and **`BindPort`** parameters of the **Dr.Web ICAPD** configuration file.

Configuring Squid to Scan FTP Traffic

FTP traffic can be transferred through **Dr.Web ICAPD** only when **Squid** proxy server is used. If you need to scan both HTTP and FTP traffic, implement one of the two schemes, mentioned below:

1. Chain of two **Squid** proxy servers.
2. Chain of **Frox** FTP proxy server, which converts FTP to HTTP, and **Squid** proxy server.

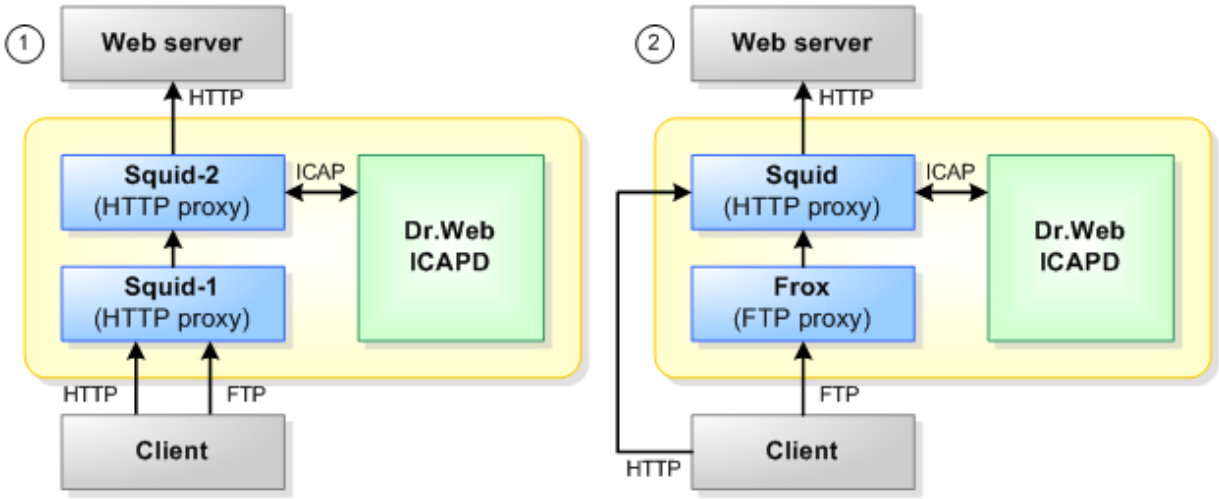


Figure 17. HTTP and FTP traffic scanning schemes

1. Chain of two Squid proxy servers (Squid-Squid)

All traffic, both HTTP and FTP, is treated as HTTP by the second proxy server (**Squid-2**) as **Squid-1** converts FTP traffic to HTTP (and back). Thus, all transmitted traffic is checked by **Dr.Web ICAPD**. To implement this scheme, do the following:

1. Install two independent **Squid** proxy servers (**Squid-1** and **Squid-2**) in different directories;
2. Adjust the **`http_port`** parameter value for **Squid-1** by setting a new port number (for example, `3129`). It must differ from the default value;
3. Configure **Squid-2** to enable interaction with **Dr.Web ICAPD** as described above;

4. Configure **Squid-2** to enable interaction with **Squid-1** as follows:

```
cache_peer localhost parent 3129 3130 default connect-timeout=80000
```

   where:
   - `3129` – number of the port used by **Squid-1**
   - `localhost` – host where **Squid-1** is installed
   - `80000` – timeout for interaction between **Squid-2** and **Squid-1**. This value must be large enough to provide correct interaction between the proxy servers.
5. Configure clients to operate via **Squid-2** using both the HTTP and FTP protocols.
6. Start both **Squid** proxy servers and **Dr.Web ICAPD**.

2. Chain of Frox and Squid proxy servers

HTTP and FTP traffic is checked by **Dr.Web ICAPD**: **Frox** converts FTP traffic to HTTP and transmits it to the **Squid** proxy server, thus, **Squid** treats them both as HTTP traffic. To implement this scheme, do the following:

1. Install **Frox** and **Squid** proxy servers;
2. Configure **Frox** to transmit data to **Squid** as HTTP traffic (FTP traffic from all FTP clients is registered as HTTP traffic transmitted by **Frox**);
3. Configure **Squid** to enable interaction with **Dr.Web ICAPD**, as described above;
4. Configure clients to operate with the **Squid** proxy server via the HTTP protocol, and with the **Frox** proxy server via the FTP protocol;
5. Start both proxy servers (**Squid** and **Frox**) and **Dr.Web ICAPD**.

ICAP Preview Mode

`ICAP Preview` mode allows to specify files which do not require scanning (e.g., streaming video and audio) and, therefore, **Dr.Web ICAPD** does not need to download them. This feature allows to decrease the amount of both external traffic (by filtering access by MIME type, size or host name) and internal traffic (using **`Allow`** `204` and **`preview_size`** `= 0` modes). As a result, `ICAP preview` mode helps improve overall speed and enhance comfort of user operation.

Rules to define files which do not require scanning (depending on the content type and size) are specified in the configuration file. For details, refer to Setting Content Filtering by MIME Type and Size.

When using **Squid** version `2.*`, it is *highly recommended* to disable `ICAP preview` mode both in **`drweb-icapd`** (**`UsePreview`** `= No`) and **Squid** (**`icap_preview_enable`** `off` and **`icap_preview_size`** `-1`).

Note that specifying any value other than `0` for the `icap_preview_size` parameter when `ICAP preview` is enabled, as well as specifying any value other than `-1` when `ICAP preview` is disabled has no effect (that is, size of previewed objects currently cannot be adjusted).

Currently, instead of `ICAP preview` mode, **SafeSquid** proxy server allows to view total amount of downloaded data.

Black and White Lists

Dr.Web ICAPD uses access control lists where addresses of Internet resources, access to which is blocked or allowed, are specified. Apart form the content-specific black lists, that are distributed with the product and updated automatically by **Doctor Web**, the user can create and configure unlimited number of user-defined lists.

Note that the user can create both black and white access control lists.

Content-Specific Black Lists

Dr.Web ICAPD uses content-specific black lists. Each of these lists is devoted to a specific topic and contains the corresponding set of URLs. The lists are updated automatically by **Dr.Web Updater.** The lists contain URL to the websites of the following categories:

- **Adult** – websites with adult content.
- **Violence** – websites that contain photos and videos of car accidents, plane crashes, natural disasters, acts of terrorism, etc.
- **Weapon** – websites that contain texts, photos, and videos of weapons (from cold steel to weapons of mass destruction) and information on manufacture of explosives.
- **Gamble** – Internet casinos, gambling and bookmaking websites.
- **Drugs** – websites that contain information on drug production, distribution, and use.
- **Obscenity** – websites with obscene language;
- **Chats**
- **Terrorism** – websites that contain detailed description of terroristic acts, manufacture of explosives, terrorist propaganda materials.
- **Email** – websites that offer free e-mail registration.
- **SocialNetwork** – dating websites, business social networking sites, corporate social networks, etc.
- **SocialEngineering** – website used for phishing and fraud;
- **MalwareLinks** – known infection sources.

Note that an Internet resource can be included in several categories. In this case, access to this resource is blocked if it matches at least one of the categories. If it is required to access a certain resource, allow access to all categories which this resource matches.

You can enable and disable block of a certain category with the use of **`Block<NAME>`** configuration parameter, where **`<NAME>`** is the name of the category. Moreover, you can specify rules to permit access depending on certain conditions.

If it is necessary to allow unconditional access to a certain resource regardless of the categories it matches, you can add the resource to the user-defined white list.

Content-specific black lists are specified in files with `.dws` extension.

User-Defined Lists

You can create black or white lists for **Dr.Web ICAPD**. User-defined black lists, as well as content-specific black lists, block access to certain hosts. User-defined white lists can be of the following types:

- **Trusted white list - WhiteHosts**. All content from the specified hosts is passed without scanning for viruses.

- **Permissive white list - WhiteDWS**. Users can access the specified hosts regardless whether or not they match a category of a content-specific black list; however, access to the hosts is forbidden if they are specified in a user-defined black list.

Note the following features of user-defined lists:

- if a host is included in a trusted white list (of the **WhiteHosts** type), access to it is controlled as usual: the host is checked whether it is included in a content-specific list in compliance with the rules and then - whether it is included in a user-defined black list.
- if a host is included in a user-defined black list (of the **BlackHosts**), access to this host is blocked unconditionally; that is, you cannot create a redefining rule that allows access to such a resource. Moreover, user-defined black lists (of the **BlackHosts** type) take precedence over user-defined permissive white lists (of the **WhiteDWS** type), that is, if a host is added both to a user-defined white list and to a user-defined black list, **access to this host is blocked**.

To create and manage user-defined lists, you can either use the **Web Interface** of **Dr.Web for Unix Internet gateways** or edit the **`drweb-icapd.ini`** configuration file.

To create a user-defined black/white list

- Create a text file containing names or IP addresses of the hosts access to which must be blocked or allowed. Each host must be specified on a separate line.
- Configure required reaction of **Dr.Web ICAPD** on attempt to access these hosts:
  - **To add hosts to a user-defined black list**, specify the path to the text file, where the hosts are listed, as a value of the **`BlackHosts`** parameter in the **Dr.Web ICAPD** configuration file. You can specify several file paths, separated by commas.

    Example:

    ```
    BlackHosts = /home/user/host_list_1, /home/user/host_list_2
    ```

    In the given example, all hosts included in the `host_list_1` and `host_list_2` files, are added to a user-defined black list; thus, access to them is blocked.
  - **To add hosts to a permissive white list**, specify the path to the text file, where the hosts are listed, as a value of the **`WhiteDWSFiles`** parameter in the **Dr.Web ICAPD** configuration file. However, if the same host is added in both permissive white list and user-defined black list, **access to this host is blocked**.

    Example:

    ```
    WhiteDWSFiles = /home/user/host_list_1, /home/user/host_list_3
    ```

    In the given example, users can access only hosts included in the `host_list_3` file, even though `host_list_1` is specified in the **`WhiteDWSFiles`** parameter. However, if the same host is included in both permissive white list and content-specific black list, **access to the host is allowed**.
  - **To add the specified hosts to a trusted white list**, to transmit the content from the hosts without scanning, specify the path to the text file as a value of the **`WhiteHosts`** parameter in the **Dr.Web ICAPD** configuration file.

    Example:

    ```
    WhiteHosts = /home/user/host_list_1, /home/user/host_list_2, /home/user/
    host_list_3
    ```

    In the given example, content received from the hosts listed in files `host_list_1`, `host_list_2` and `host_list_3` is not scanned for viruses.

Please note that the **`WhiteHosts`** parameter only disables anti-virus scanning of files received from the specified hosts, but does not manage access to them. Thus, in the example given above (according to the **`BlackHosts`** and **`WhiteDWSFiles`** values), access will be allowed only to the hosts specified in the `host_list_3` list, and content received from these hosts will not be scanned for viruses.

Command Line Parameters

The following command line parameters are supported by **Dr.Web ICAPD** (`drweb-icapd`):

Short case	Extended case	Arguments
`-h`	`--help`	
Description: Show information about supported command line parameters on the screen and exit		
`-v`	`--version`	
Description: Display **Dr.Web ICAPD** version on the screen and exit		
`-d`		
Description: Output debug log to the console screen		
`-f`		`<path to file | path to the Agent socket>`
Description: Set a new path to the **Dr.Web ICAPD** configuration file or to the **Dr.Web Agent** socket, if **Dr.Web ICAPD** is to receive its configuration from **Dr.Web Agent**		
`-m`		
Description: **Dr.Web ICAPD** is started under **Dr.Web Monitor** control		

Settings of Dr.Web ICAPD

Dr.Web ICAPD can be started with default settings, but if you want to ensure optimal performance, you may adjust it according to your specific requirements. Configuration file of **Dr.Web ICAPD** (`drweb-icapd.ini`) is located in the `%etc_dir` directory.

This file consists of the mandatory section `[Icapd]`, where main configuration parameters of **Dr.Web ICAPD** operation are specified, and `[match]` and `[def]` additional sections where parameters of **Dr.Web ICAPD** operation can be redefined for certain user groups or depending on conditions.

- For details on how to specify parameters in the main section and short description of parameter types, refer to the Configuration Files section.
- For the list of parameters specified in the `[Icapd]` section, refer to the Configuration Parameters section.
- For details on content filtering rules, refer to the Settings of Content Filtering by MIME Type and Size section.
- For details on how to redefine access parameters for certain users or depending on conditions, refer to the Redefining Parameters for User Groups section.

Configuration parameters

General settings of **Dr.Web ICAPD** operation are specified in the `[Icapd]` section of the `drweb-icapd.ini` configuration file. This section contains the following parameters:

`Logfile` = `{path to file | syslog}`	Log file name. You can specify `syslog` to enable logging with the **syslog** service. In this case, you must also specify **`SyslogFacility`** and **`SyslogPriority`** parameters.

	Default value: **Logfile** = syslog
SyslogFacility = {syslog label}	Facility label for logging with the syslog service. Default value: **SyslogFacility** = Daemon
SyslogPriority = {log level}	Verbosity level for logging with the syslog service. You can specify one of the following levels: • Alert • Warning • Info • Notice Default value: **SyslogPriority** = Info
Loglevel = {numerical value}	Log verbosity level. The value is a sum of an arbitrary set that can consist of the following values: • 0 – output information on errors and detected viruses • 1 – output information at the Info level: on checked clean files and other service information • 2 – output general messages • 4 – output results of chunk analysis • 8 – output extended messages on chunks • 16 – output activity log of the syntax analyzer • 32 – output other debugging messages **Example**: The value 18, which is a sum of the following values: 0 + 2 + 16, enables logging of information on errors and detected viruses as well as logging of general messages and messages of syntax analyzer. Thus, maximum possible parameter value equals to 63. Please note that **Loglevel** = -1 disables logging. Default value: **Loglevel** = 1
MaxLogSize = {size}	Maximum log file size. Each time **Dr.Web Daemon** starts, size of the log file is checked. If it is greater than the **MaxLogSize** parameter value, log file is overwritten. Set this parameter value to 0 to disable check of log file size at startup. Default value: **MaxLogSize** = 1m
Hostmaster = {e-mail address}	Administrator's e-mail address. Default value: **Hostmaster** = root@localhost

Reactions of **Dr.Web ICAPD** on detection of viruses and other threats in scanned files:

Infected = {action}	Reaction to an infected object. You can specify one of the following actions: Cure, Move, Truncate, Report. Default value: **Infected** = Cure
Incurable = {action}	Reaction to an incurable object (if Cure action was applied but failed). You can specify one of the following actions: Move, Truncate, Report. Default value: **Incurable** = Report
Suspicious = {action}	Reaction to a suspicious object detected by the heuristic analyzer. You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Suspicious** = Report
Adware = {action}	Reaction to an object containing an advertising program (adware). You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Adware** = Report
Dialers = {action}	Reaction to an object containing a dialer program. You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Dialers** = Report
Jokes = {action}	Reaction to an object containing a joke program. You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Jokes** = Pass
Riskware = {action}	Reaction to riskware (programs that can be used to harm the system). You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Riskware** = Pass
Hacktools = {action}	Reaction to a program used for hacking. You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **Hacktools** = Pass

ArchiveRestriction = {action}	Reaction to an archive that cannot be scanned by **Dr.Web Daemon** because a threshold value specified in the main configuration file was exceeded. You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **ArchiveRestriction** = Report
DaemonError = {action}	Reaction to an object that caused errors during scanning (for example, **Dr.Web Daemon** is out of memory or does not have permissions required for further processing). You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **DaemonError** = Report
SkipObject = {action}	Reaction to an object that cannot be scanned by **Dr.Web Daemon** (for example, password protected or broken archive, symbolic link or non-regular files). You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **SkipObject** = Pass
LicenseError = {action}	Reaction to an object during scanning of which a license error occurred (for example, license expired). You can specify one of the following actions: Pass, Move, Truncate, Report. Default value: **LicenseError** = Report
Heuristic = {logical}	Enables or disables the heuristic analyzer mode. The detection method used by the *heuristics analyzer* is based on certain knowledge about the attributes that characterize malicious code. Each attribute or characteristic has a weight coefficient that determines the level of its severity and reliability. Depending on the sum weight of a file, the *heuristics analyzer* calculates the probability of unknown virus infection. As with any system of hypothesis testing under uncertainty, the *heuristics analyzer* may commit type I or type II errors (i.e., it may omit viruses or raise false alarms). It is recommended to send copies of such files to the virus laboratory of **Doctor Web** for analysis at http://vms.drweb.com/sendvirus/. Note that object detected by the *heuristic analyzer* are treated by suspicious. Default value: **Heuristic** = Yes

Global parameters of blocking Internet resources that are included in predefined content-specific black lists (can be overridden by rules):

BlockAdult = {logical}	Enables or disables blocking of Internet resources included in the Adult content-specific black list.

	Default value: **BlockAdult** = Yes
BlockViolence = {logical}	Enables or disables blocking of Internet resources included in the Violence content-specific black list. Default value: **BlockViolence** = Yes
BlockWeapon = {logical}	Enables or disables blocking of Internet resources included in the Weapon content-specific black list. Default value: **BlockWeapon** = Yes
BlockGamble = {logical}	Enables or disables blocking of Internet resources included in the Gamble content-specific black list. Default value: **BlockGamble** = Yes
BlockDrugs = {logical}	Enables or disables blocking of Internet resources included in the Drugs content-specific black list. Default value: **BlockDrugs** = Yes
BlockObscenity = {logical}	Enables or disables blocking of Internet resources included in the Obscenity content-specific black list. Default value: **BlockObscenity** = Yes
BlockChats = {logical}	Enables or disables blocking of Internet resources included in the Chats content-specific black list. Default value: **BlockChats** = No
BlockTerrorism = {logical}	Enables or disables blocking of Internet resources included in the Terrorism content-specific black list. Default value: **BlockTerrorism** = Yes
BlockEmail = {logical}	Enables or disables blocking of Internet resources included in the Email content-specific black list. Default value: **BlockEmail** = No
BlockSocialNetwork = {logical}	Enables or disables blocking of Internet resources included in the SocialNetwork content-specific black list. Default value: **BlockSocialNetwork** = No
BlockSocialEngineering = {logical}	Enables or disables blocking of Internet resources included in the SocialEngineering content-specific black list.

	Default value: **BlockSocialNetwork** = Yes
BlockMalwareLinks = {logical}	Enables or disables blocking of Internet resources included in the MalwareLinks content-specific black list. Default value: **BlockMalwareLinks** = Yes
BlockAll = {logical}	Enables or disables Internet access both to allowed and forbidden resources. Note that effect of this parameter is not the same as assigning Yes or No value to all **Block<NAME>** parameters (where **<NAME>** is the name of a corresponding content-specific black list). • When this parameter is set to Yes, access to all Internet resources is blocked regardless whether or not they belong to a white or black list • When this parameter is set to No, access is allowed only to the Internet resources that are not included in the content-specific black lists or are included in the user-defined white lists. If it is required to allow access to all Internet resources regardless whether or not they belong to black lists, set values of both the **BlockAll** and all **Block<NAME>** parameters to No. Moreover, clear the user-defined black list, specified in the **BlackHosts** parameter. Default value: **BlockAll** = No

Definitions of User-defined black and white lists:

WhiteDwsFiles = {paths to files list}	Permissive user-defined white list. The parameter value is a list of paths to text files, separated by commas. The specified files contain hosts which content is not to be checked for matching a black list category (of both content-specific and user-defined black lists). However, the content is to be scanned for viruses. The parameter is necessary to allow access to those websites which are blocked due to being included in a black list. Hosts are specified in files in the following ways: host1 host2 ... You can also redefine access parameters with the use of rules to allow conditional access. If it required to allow access to certain hosts without scanning traffic for viruses, add these hosts to the trusted white list (in the **WhiteHosts** parameter value). Default value: **WhiteDwsFiles** =
WhiteHosts = {paths to files list}	Trusted user-defined white list. The parameter value is a list of paths to text files, separated by commas. The specified files contain hosts which content is not to be scanned for viruses. However, the content is to be checked for matching a black list, both content-specific and user-defined lists.

	In order to allow user access to a host, include it to the permissive white list (in the **WhiteDwsFiles** parameter value). This parameter is used to prevent false alarms of **Dr.Web Daemon**. You can specify the host name or its IP addresses. Default value: **WhiteHosts** =
BlackHosts = {paths to files list}	User-defined black list. The parameter value is a list of paths to text files, separated by commas. The specified files contain hosts access to which is to be blocked. You can specify the host name or its IP addresses. Note that if a host is included in this list, access to the host is blocked unconditionally; that is, this setting cannot be redefined with the use of rules. Default value: **BlackHosts** =

Other configuration parameters:

SendUrlsWithViruses = {logical}	Enables or disables an option to send addresses of web pages containing viruses and names of detected viruses to **Doctor Web** company automatically. Please note that this option requires **Dr.Web Agent** to be installed. Default value: **SendUrlsWithViruses** = No
MaxBlocksize = {size}	Sets maximum size of the memory block which can be allocated by **Dr.Web ICAPD** at a time. If random access memory is enough, this parameter value can be increased for better performance. Default value: **MaxBlocksize** = 10m
LocalScan = {logical}	Enables or disables the **local** scan mode. If **LocalScan** = Yes, **Dr.Web Daemon** scans files in the local mode; that is, only paths to the files are transmitted to the component. Otherwise, it receives the content of files for scanning. The parameter value can be set to Yes only if **Dr.Web Daemon** and **Dr.Web ICAPD** are operating on the same host. Default value: **LocalScan** = Yes
User = {user name}	User whose privileges are used by **Dr.Web ICAPD**. It is strongly recommended to create drweb user and enable **Dr.Web ICAPD** to use its privileges. Default value: **User** = drweb
Cache =	Path to the directory where temporary files are created and stored.

	Default value: **Cache** = %var_dir/cache/
DwsDirectory = {path to directory}	Path to the directory with predefined content-specific black lists (files with the .dws extension). Default value: **DwsDirectory** = %var_dir/dws/
Templates = {path to directory}	Path to directory containing templates for report generation. Default value: **Templates** = %etc_dir/templates/icapd
PidFile = {path to file}	Name of a file where information on the PID, Unix socket (if enabled with the **Socket** parameter) or port number (if enabled with the **Socket** parameter) is saved on the **Dr.Web ICAPD** startup. If more than one **Socket** parameter is specified, this file contains information on all of the sockets (one per line). This file is created every time **Dr.Web ICAPD** starts. Default value: **PidFile** = %var_dir/run/drweb_icapd.pid
Key = {path to file}	Path to the key file (license or demo). Usually a key file has the .key extension. Default value: **Key** = %bin_dir/drweb32.key
BindPort = {numerical value}	Number of the port to which ICAP clients (e.g. **Squid**) connect on attempt to establish connection with **Dr.Web ICAPD**. Note that this value must be equal to corresponding value, specified for the used HTTP proxy server. Default value: **BindPort** = 1344
BindAddress = {host name | IP address}	Host where drweb-icapd operates. Note that this value must be equal to corresponding value, specified for the used HTTP proxy server. Default value: **BindAddress** = 127.0.0.1
DrwebAddress = {addresses list}	List of sockets used for connection with **Dr.Web Daemon**. Addresses in the list are separated by commas. **Examples**: **DrwebAddress** = inet:3000@localhost **DrwebAddress** = local:%var_dir/.daemon **DrwebAddress** = pid:/usr/local/drweb/run/drwebd.pid Note that if the used **Dr.Web Daemon** is running on a remote machine, **LocalScan** parameter value must be set to No. If a socket address or path to **Dr.Web Daemon** PID file is specified first in the list, local scanning will be forced to terminate if connection to this address cannot be established.

	If this list is empty, **Dr.Web ICAPD** operates without connection to **Dr.Web Daemon** and anti-virus check is not performed. Default value: **DrwebAddress** = pid:%var_dir/run/drwebd.pid
PathToQuarantine = {path to directory}	Path to the **Quarantine** directory. Default value: **PathToQuarantine** = %var_dir/infected/
QuarantineFilesMode = {access permissions}	Permissions to access files in **Quarantine**. Default value: **QuarantineFilesMode** = 0660
Timeout = {numerical value}	Timeout for a socket to wait for data to be received, in seconds. When at least one byte is received/dispatched, the counter is reset. If 0 is specified, the wait time is unlimited. Default value: **Timeout** = 300
SendMail = {logical}	Enables or disables sending notifications to administrator on attempt to download a malicious object. Notifications are sent to the address specified in the **Hostmaster** parameter. Default value: **SendMail** = No
SendMailDwsBlock = {logical}	Enables or disables sending notifications to administrator on attempt to open a web page blocked due to matching a black list category. Notifications are sent to the address specified in the **Hostmaster** parameter. Default value: **SendMailDwsBlock** = No
MailCommand = {text}	Shell command executed to send a notification to administrator. Placeholder %s in the command text is replaced with the **Hostmaster** parameter value. Default value: **MailCommand** = "/usr/sbin/sendmail -i -bm -f drweb -- %s"
MailCache = {numeric value}	Time period, in seconds, within which notifications on repeated attempts to open the same "bad" page are not sent to the administrator. If the parameter value is set to 0, notification is sent every time a page is blocked. Default value: **MailCache** = 60
AclList = {paths to files list}	The parameter value is a list of paths to text files, separated by commas. The specified files contain IP addresses and host names, for which access to **Dr.Web ICAPD** via the ICAP protocol is

	allowed. If the list is empty or the specified files do not contain any address, access to **Dr.Web ICAPD** is allowed for all clients. Default value: **AclList** =
SendStat = {logical}	Enables or disables sending statistics on detected viruses to **Dr.Web Agent**. Default value: **SendStat** = No
KeepAlive = {logical}	Enables or disables maintenance of permanent connection with the proxy server. Default value: **KeepAlive** = Yes
UsePreview = {logical}	Enables or disables the ICAP preview mode. If the proxy server does not work correctly in this mode, disable this option by specifying No. Default value: **UsePreview** = Yes

Note that one Internet resource can be included in several content-specific black lists as well as in a user-defined black list. In this case, access to this resource is blocked if at least one content-specific black list is active. If it is necessary to allow access to such a resource, deactivate all content-specific black lists where it is included.

At the end of the [Icapd] section, subsection with filtering rules is located. It starts with the **MimeStart** string and ends with **MimeEnd**. For detailed information on content filtering rules, see Settings of Content-filtering by MIME type and size.

Redefining Parameters for User Groups

You can specify individual settings to configure access to Internet resources for certain users or user groups. For that purpose, adjust basic settings in the main section of the configuration file or specify redefining rules.

In the current version of **Dr.Web for Unix Internet gateways**, you can redefine the following parameters:

- **BlockAdult**
- **BlockViolence**
- **BlockWeapon**
- **BlockGamble**
- **BlockDrugs**
- **BlockObscenity**
- **BlockChats**
- **BlockTerrorism**
- **BlockEmail**
- **BlockSocialNetwork**

- **BlockSocialEngineering**
- **BlockMalwareLinks**
- **BlockAll**

The rules are specified in the [match] section of the drweb-icapd.ini configuration file. Functions, used in these rules, are specified in the [def] section of the same configuration file. These sections can be declared in random order, but every function used in rules must be predefined in the [def] section. Expressions for rules and functions inside the [match] and [def] sections can be specified on multiple lines.

Note that one Internet resource can be included into several content-specific black lists as well as in a user-defined black list. In this case, access to this resource is blocked if at least one content-specific black list is active. If it is necessary to allow access to such a resource, deactivate all content-specific black lists where it is included.

However, to block access to other resources matching the deactivated black list categories, it is strongly recommended to include check of both URL and client properties (e.g., IP address) in the condition of an allow rule.

Note that if an Internet resource is included in a user-defined black list, access to this resource cannot be allowed by a rule.

Variables

Every request sent from a client to the proxy server has a set of unique parameters. You can use these parameters in rules, after specifying them as variables:

Variable name	Variable type	Description
request_url	string	Request URL
request_username	string	Name under which the user authorized on the proxy server. The name is extracted from the X-Client-Username header. If the header is not present, the variable value is treated as an empty line.
request_ip	IP-address with network mask (CIDR)	IP address of the user who sent a request to the proxy server. The address is extracted from the X-Client-IP header. If the header is not present, the variable value is treated as undefined.
system_time	time	Current system time (hours and minutes).

Please note that use of the **request_ip** and **request_username** variables requires the **Squid** HTTP proxy server to be configured accordingly.

Logical expressions

Logical expressions are operations of function call united by the following operators: **&&** - conjunction (logical AND), **||** - disjunction (logical OR), **!** - negation (logical NOT). To group operations and change their priority, use brackets.

Note that only **&&**, **||** and **!** operators can be used in a logical expression.

Standard mnemonics (AND, OR, NOT) are not allowed.

Syntax of BOOL_EXPR logical expressions is as follows:

```
func_name() | COMPARE |
(BOOL_EXPR) | !BOOL_EXPR |
BOOL_EXPR && BOOL_EXPR |
BOOL_EXPR || BOOL_EXPR
```

Where BOOL_EXPR is a logical expression, func_name() is a call of function with the func_name name, and COMPARE is one of the comparison operations listed below. The function must be defined in the [def] section.

Comparison operation can be one of the following:

Notation	Description
string_var cidr_var time_var	Variable of the corresponding type (STRING, CIDR or TIME).
TIME	String of the following format: "HH: MM" or "H: MM" (hours, minutes); must be enclosed in quotation marks.
STRING	Random string enclosed in quotation marks.
REGEX	Regular expression of the POSIX extended format; must be enclosed in quotation marks.
FILE_NAME	File path enclosed in quotation marks.
CIDR	IPv4 address enclosed in quotation marks (you can specify a network mask after a stroke character). If the network mask is not specified, it is treated equal to /32. An empty string "" indicates an undefined value.

The following comparison operations are supported for variables of the string type:

Operation	Description
string_var == STRING	Variable matches the string.
string_var != STRING	Variable does not match the string.
string_var ~ REGEX	Variable contains the substring that is checked for matching the regular expression (search method is used).
string_var == file:FILE_NAME	Variable matches at least one string in the specified file.
string_var ~ file:FILE_NAME	Variable corresponds to at least one regular expression in the specified file.

== and ~ operations are case insensitive.

The following comparison operations are supported for variables of the cidr type:

Operation	Description
cidr_var <<= CIDR	IP address is within the specified network range.
cidr_var <<= file:FILE_NAME	IP address is within at least one of the networks listed in the file.

If both arguments of <<= operation have undefined value, the operation result is true. If only one parameter has undefined value, the operation result is false.

The following comparison operations are supported for variables of the time type:

Operation	Description
time_var > TIME time_var >= TIME time_var < TIME	Time comparison.

Operation	Description
time_var <= TIME	

Every operation has a certain priority relative to other operations. Sorted in descending order, comparison operation priority is as follows:

1. ! ("logical NOT")
2. < ("less than"), <= ("less than or equal to"), > ("greater than"), >= ("greater than or equal to")
3. == ("equal to"), != ("not equal to"), ~ ("matches"), <<= ("belongs to")
4. && ("logical AND")
5. || ("logical OR")

Operations listed in the same line have equal priority and are processed from left to right.

For certain operations, reading of a value array from a file (specified with the `file:` prefix) is available. Lines beginning with the "#" or ";" characters as well as with empty lines are skipped when reading values. The content of the `file:FILE_NAME` file is read while the configuration file is processed. Thus, after changing content of the file that contains values (or a path to such a file), force **drweb-icapd** to reread its configuration (for example, by sending the **drweb-icapd** daemon `SIGHUP` signal).

Redefining Parameters: [match] section

The rules are set in the `[match]` sections of the `drweb-icapd.ini` configuration file. In these rules, special `if` statements are used.

Syntax of an `if` statement:

```
if BOOL_EXPR {
      configuration section
}
```

where `BOOL_EXPR` is a logical expression and `configuration section` is a list of parameters to which new values (different from the global values, specified in the configuration file) are to be assigned.

Value for a parameter is defined as follows:

1. **Dr.Web ICAPD** checkes whether the requested resource matches any of the black list category.
2. If the corresponding URL is found in a black list (for example, in the **Terrorism** list), **Dr.Web ICAPD** requests value of the **BlockNAME** parameter, where **NAME** is a category of the black list (in the given example, **BlockTerrorism**).
3. The value is first searched in the `[match]` section according to the following algorithm:
   - for all request variables, value of the `if`-statement expression is calculated.
   - if true, the required parameter is searched in the configuration section.
   - if this parameter is found, its value is returned and the search completes
   - if this parameter is not found or if the `if` statement is false, **Dr.Web ICAPD** goes to the next `if`-statement.
4. If in the `[match]` section none of the rules is matching or does not contain the required parameter, the global parameter value is returned (or its default value, if this parameter is not specified in the configuration file).

Search of the parameter value is performed until the first match; thus, the first found value is returned (from the configuration blocks of those `if` statements that have `true` expressions).

Functions: [def] section

Functions can be used in any logical expressions; however, each function must be predefined in the [def] section before use. In one section, several functions can be defined. Moreover, you can specify several `[def]` sections in the configuration file (functions defined in different sections are combined into one list while reading configuration).

Function definition syntax:

```
func_name_1 = { BOOL_EXPR }
```

where `BOOL_EXPR` is a logical expression.

All functions return a Boolean value, arguments are not supported. In fact, a function is just a shorthand notation for an expression.

Example:

Definition of the `is_localhost` and `local_ip` functions: these functions are to be `true` if the request was sent from one of the specified IP addresses or from one of the IP addresses listed in the file.

```
[def]
is_localhost = { request_ip <<= "127.0.0.0/8" }
local_ip = {
      request_ip <<= "127.0.0.0/8"
      || request_ip <<= "192.168.0.0/16"
      || request_ip <<= "172.16.0.0/12"
      || request_ip <<= file:"/tmp/icapd/other_local_ips.txt"
}
```

Definition of a `worktime()` function: if the current system time is between 9:30 and 13:00 or 14:00 and 18:15, the function is to be `true`.

```
[def]
worktime = {
 (system_time>="9:30" && system_time<="13:00")
 ||
 (system_time>="14:00" && system_time<"18:15")
}
```

Example Usage

If it is necessary to block access to Internet resources from the **Adult** and **Email** lists during working hours of the local network users, as well as block access from a certain IP address, you can specify the following rule:

```
[match]
if (local_ip() ||
    request_ip <<= "87.249.57.20") &&
    worktime() {
      BlockAdult = yes
      BlockEmail = yes
}
```

If you want to block access to Internet resources from the **Terrorism** list during night time (from 23:00 to 8:00) for certain IP addresses, you can specify the following rule:

```
[match]
if (request_ip <<= "93.185.182.46" ||
    request_ip <<= "195.98.93.66") &&
    (system_time>="23:00" ||
    system_time<="8:00")
{
     BlockTerrorism = yes
}
```

To prevent Internet access during nonworking time for "`edx`" user:

```
[match]
if request_username=="edx" && !worktime()
{
     BlockAll = yes
}
```

Note that `worktime()` function, used in the examples, must be predefined in the `[def]` section.

To block access to a certain Internet resource for all users whose name either matches the "`john .*`" regular expression, or any regular expression specified in the file, or one of the lines in the file, use the following rule:

```
[match]
if (request_username ~ "john.*" ||
 request_username ~ file:"/tmp/icapd/users_re_block.txt"
 || request_username == file:"/tmp/icapd/users_block.txt")
 && (request_url=="http://example.com/mega_music.mp3")
{
     BlockAll = yes
}
```

Note that setting the **`BlockAll`** parameter value to `No` does not mean enabling access to all resources when the rule is true. In this case, access to a resource is allowed if it is either included in user-defined white list or not included in active black lists (in both `<NAME>` content-specific lists, which are active when the corresponding **`Block<Name>`** parameters are set to **`Yes`**, and user-defined black lists).

If in normal mode access to resources is blocked due to being on the black lists, but it is required to allow access to some of these resources, specify a corresponding rule.

For example, let it be required to allow access to `socialnetwork.com` for users whose IP address is within `192.168.1.1/32` network range, despite this resource being included in **`SocialNetwork`** and **`Chats`** active black lists.

```
if (request_ip <<= "192.168.1.1/32") && (request_url ~ "socialnetwork.com")
{
  BlockSocialNetwork = no
  BlockChats = no
}
```

This rule allows access to resources included in **`SocialNetwork`** and **`Chats`** black lists only if both of the following conditions are true:

- client's IP address is within the `192.168.1.1/32` range
- the requested URL contains the `socialnetwork.com` substring.

Otherwise, global settings, specified in the configuration file, are applied. Note that if a resource matches several black list categories, it is required to disable blocking of the resource by all of the black

lists.

Configuring Squid to Operate with Variables

Usage of the `request_username` and `request_ip` variables requires additional configuration of the **Squid** proxy server. For that purpose, adjust the `squid.conf` configuration file (typically `/usr/local/squid/etc/squid.conf`).

If the following lines are already present in the configuration file, you can uncomment them and adjust the values if necessary. Otherwise, add the lines at the end of the file.

To enable use of request_ip:

```
# request_ip
icap_send_client_ip on
```

To enable use of request_username:

```
# request_username
icap_send_client_username on
icap_client_username_header X-Client-Username
icap_client_username_encode off
```

Setting Content Filtering by MIME Type and Size

Rules of filtering content by file size and MIME type are defined at the end of the main `[Icap]` section of the `drweb-icapd.ini` configuration file. This section always starts with **`MimeStart`** line and ends with **`MimeEnd`** line. The section contains filtering rules (one per line).

Content filtering requires the proxy server to support the ICAP preview mode. Moreover, ensure that the **`UsePreview`** parameter value is set to `Yes`.

Filtering rules are specified as follows:

```
<MIME type> <action1> <size> <action2>
```

where:

- **`MIME type`** – it is a MIME type of content, for example:
  - `*` – file of any type;
  - `application` – executables, archives, MS Office and PDF documents, etc
  - `audio` – audio files (`mp3`, `wav`, `wma`, etc.)
  - `image` – images (`gif`, `jpg`, `png`, `svg`, etc.)
  - `message` – messages between web servers and clients
  - `multipart` – containers (mail files, packed files)
  - `text` – text or source code (`html`, `xml`, `css`, etc.)
  - `video` – video files (`mpeg-1`, `mp4`, `wma`)
  - `model` – 3D models.

  You can specify either a family of MIME types or a concrete type (for example, `video` indicates any video files, `video/mpeg` – only file of MPEG type).

  The rule specified for the nearest matching MIME type is applied to an object. Thus, the rule specified for files of any type (`"*"`) is applied only if no other rule matching the object MIME type is found.
- `<action1>` – action (`scan`, `pass`, `reject`) that is applied if the object size is not greater than

the specified <size> value.

- <size> – threshold size. If the object size is not greater than this threshold, <action1> is applied; otherwise <action2> is applied.
- <action2> – action (scan, pass, reject) that is applied if the object size is greater than the specified <size> value.

If all is specified in the <size> field, only the first action (<action1>) is applied to the object. In this case, it is not required to specify <action2>.

The following actions are allowed:

- **scan** – send the file for scanning
- **pass** – pass the file to the user without scanning
- **reject** – reject the file and return another object. This action must be specified with a switch that defines what data is returned to the user:
  - -report – return an HTML page notifying the user that the file is blocked
  - -trunc – return a requested file truncated to zero length (empty file).

  Note that the **reject** action must not be specified without a switch!

The order in which filtering rules are specified is of no importance.

Examples of filtering rules:

```
MimeStart
 *                          scan 1M pass
 application                scan 1M pass
 image                      scan 1M pass
 message                    scan 1M pass
 multipart                  scan 1M reject -report
 text                       scan 1M pass
 audio                      pass all
 video                      pass all
 application/x-mms-framed   pass all
MimeEnd
```

The first rule from the given example is applied to objects which MIME type does not correspond to any of the types specified in the subsequent rules. If size of that object is less than 1MB, it is sent for scanning; otherwise, it is passed to the user without scanning. The rule specified for objects of the multipart type, instructs to reject such objects if their size is greater than 1 MB, and return an HTML page notifying on the rejection. The last rule is applied to all objects of application/x-mms-framed MIME type and instructs to pass all these objects to the user without scanning regardless of the object size.

Please note that file is sent for scanning only after the **scan** action is applied to it (in this case, the file might be rejected due to the results of scanning, depending on the specified settings). Otherwise, if the **reject** action (with '-report' or '-trunc' switch) is applied, the file is not scanned and the user receives either the corresponding notifying HTML page or the empty file.

Interaction between Dr.Web Agent and Dr.Web Monitor

Dr.Web ICAPD can interact with **Dr.Web Agent** and **Dr.Web Monitor**.

Dr.Web ICAPD receives configuration and key file from **Dr.Web Agent**. Thus, if **Dr.Web ICAPD** is started under control of **Dr.Web Agent**, the **Key** parameter value from the drweb-icapd.ini configuration file is ignored. Instead of this, the path to the key file specified in **Dr.Web Agent** configuration (by default, %etc_dir/agent.conf file) is used.

Dr.Web ICAPD can send information on detected viruses to **Dr.Web Agent** (this process is managed by **SendStat** parameter in the drweb-icapd.ini configuration file).

Dr.Web Agent can send the received information to the **Doctor Web** website. For the purpose, specify md5 sum of the key file as the **UUID** parameter value in the agent.conf file.

You can view gathered statistics at http://stat.drweb.com/view/<md5sum>/, where <md5sum> is the value of the **UUID** parameter.

You can connect **Dr.Web Agent** on **Dr.Web ICAPD** startup by specifying a path to the **Dr.Web Agent** socket in -f command line parameter.

Moreover, **Dr.Web Agent** can automatically send addresses of infected Internet resources for analysis to **Doctor Web**, which allows to improve the operating quality.

Dr.Web Monitor allows to automate startup of **Dr.Web ICAPD** and control its operation. When starting **Dr.Web ICAPD** via **Dr.Web Monitor**, **Dr.Web ICAPD** automatically requests configuration from **Dr.Web Agent**. To start **Dr.Web ICAPD** via **Dr.Web Monitor**, add ICAPD value to **RunAppList** parameter of **Dr.Web Monitor** configuration file (by default, %etc_dir/monitor.conf) or specify ICAPD value in -r command line parameter on **Dr.Web Monitor** startup. The script for automatic **Dr.Web ICAPD** startup must be removed from the system paths, as the function to start and stop the component is now performed by **Dr.Web Monitor**.

For more detailed information on **Dr.Web Agent** and **Dr.Web Monitor**, refer to **Dr.Web Agent** and **Dr.Web Monitor** sections of this document.

Startup

It is recommended to start the interactive components in the following order:

- **Dr.Web Daemon**;
- **Dr.Web ICAPD**;
- Used proxy server.

Not a single object will be pass without check, regardless of the order the components are started: either the proxy server blocks data transfer if not connected to **Dr.Web ICAPD** or **Dr.Web ICAPD** blocks data transfer if not connected to **Dr.Web Daemon** (the user will be notified on that).

How to Test Dr.Web ICAPD

To test **Dr.Web ICAPD**

1. Ensure that values of **Infected**, **Suspicious** and **Incurable** parameter values in the drweb-icapd.ini configuration file are set to Report.
2. Visit the following webpage: http://www.eicar.org/download/eicar.com. Notification that the file is infected will appear in the browser window.

If the warning message does not display, check the following:

- to access HTTP traffic, the browser uses **Squid** proxy server which is configured to operate with **Dr.Web ICAPD**.
- templates are copied to %etc_dir/templates/icapd/ subdirectory and paths to them are specified correctly in the drweb-icapd.ini file.

Links to Squid and SafeSquid Project Websites

To visit the **Squid** project website, go to http://squid-cache.org/.

For description of how **Squid** supports the ICAP protocol, go to http://squid.sourceforge.net/projects.html#icap

To visit the **SafeSquid** project website, go to http://safesquid.com/.

Notification Templates

Dr.Web ICAPD uses two types of notification templates:

1) **HTML templates** – they are used for generation of notifications that display as an HTML page in the browser. These notifications are displayed to the user on attempt to access a blocked resource or download an infected or suspicious object, if `'Move'` or `'Report'` actions was applied according to the settings.
2) **Mail templates** – they are used for generation of notifications that are sent to the administrator on certain events (e.g., on attempt to access to a blocked resource, download infected objects), according to the specified settings.

All notification templates are ordinary text files that reside in the following directory: `%etc_dir/templates/icapd[_lng]`, where `_lng` is a suffix that denotes a language used for the templates presented in this directory (for example, `ru` – Russian, `ja` – Japanese). Note that the folder with templates for English language `/icapd` must be always present and the language prefix is not specified in its name.

If necessary, change the folder with used notification templates. To do this, specify the correct path to it as a **`Templates`** parameter value in the configuration file. Folder with notification templates must contain the following files (the file names must not be changed):

File name	Description
`ERR_FAILED_CHECK`	HTML template to report a file scan error
`ERR_DOWNLOAD_BLOCK`	HTML template to report an access error on attempt to open a blocked web page (from a black list)
`ERR_ACCESS_DENIED`	HTML template to report a download error on attempt to download a forbidden object. This template is used if `Report` action was applied to the object.
`ERR_ACCESS_DENIED_MOVED`	HTML template to report a download error on attempt to download an forbidden object. This template is used if `Move` action was applied to the object.
`email.templ`	Mail template for e-mail messages sent to the administrator (to the address specified in **`Hostmaster`** parameter value).

1. HTML templates

HTML template is a correct HTML file which can include style sheets and JavaScript codes, but must not include embedded objects, images, and URLs to external resources (outside the HTML page): for example, a style sheet from an external CSS file cannot be used.

The HTML template can contain special macros - placeholders (anywhere in the HTML document). These embedded placeholders are replaced with corresponding text while an HTML page is generated based on the template.

The following placeholders can be used:

Placeholder	Description
$DAEMON_REPORT$	Is replaced with a line from **Dr.Web Daemon** report containing the reason of the occurred incident (e.g., virus detection or failure to check the file due to specified restrictions)
$DATE$	Is replaced with a line containing date and time of when the notification was generated
$FILE_NAME_ERROR$	Is replaced with the path to a file moved to **Quarantine**
$HOSTMASTER$	Is replaced with the Administrator's e-mail address (specified as the **Hostmaster** parameter value)
$MANUAL_SUBMIT$	Is replaced with the HTML form containing only one button. When the button is clicked, information on the event is sent to **Doctor Web**. Caption for the button must be specified directly after the placeholder, for example: $MANUAL_SUBMIT$Send!
$RELEASE$	Is replaced with information on the installed **Dr.Web** product (including product name, product version, and version of **Anti-virus Engine**)
URL	Is replaced with the full URL to the resource requested by the user
URL_SHORT	Is replaced with short URL to the resource requested by the user
$VERSION$	Is replaced with the current version of **Dr.Web ICAPD**

HTML template example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Blocked by Dr.Web ICAPD</title>
 </head>
 <body>
  <H1>Content blocked!</H1>
  <p>Access to this resource is denied due to administration policy.<br/>
     Please save content of this page and ask your
     <a href="mailto:$HOSTMASTER$">system administrator</a>
     for further instructions.
  </p>
  <H2>Details:</H2>
  <p><strong>Restricted URL: </strong><a href="$URL$">$URL_SHORT$</a></p>
  <p><strong>Reason: </strong><pre>$DAEMON_REPORT$</pre></p>
  <H2>Product information:</H2>
  <p><strong>Release: </strong><pre>$RELEASE$</pre></p>
  <p><strong>Version: </strong><pre>Dr.Web ICAPD $VERSION$</pre></p>
  <p>$MANUAL_SUBMIT$Notify Dr.Web</p>
 </body>
</html>
```

2. Mail templates

The current version of the **Dr.Web ICAPD** uses only one mail template – `email.templ`. It used for messages sent to the administrator if an incident occurs. This template has the same internal structure as the structure of a correct e-mail message of the `multipart/mixed` MIME format.

The mail template can contain special macros - placeholders (anywhere in the message). These placeholders are replaced with corresponding text while a message for the administrator is generated based on the template.

The following placeholders can be used:

Placeholder	Description
`$ACTION$`	Is replaced with the name of the action applied to the infected object
`$HOSTMASTER$`	Is replaced with the Administrator's e-mail address (specified as the **`Hostmaster`** parameter value)
`$HTML_PAGE$`	Is replaced with the text of HTML page displayed to the user
`$IP$`	Is replaced with the IP address of the user who tried to download the rejected object
`$REASON$`	Is replaced with the reason of the occurred event (for example, virus detection or failure to check the file due to specified restrictions)
`$SIZE$`	Is replaced with size of the rejected object
`$TIME$`	Is replaced with with the string containing date and time when the notification was generated
`$URL$`	Is replaced with the full URL to the resource requested by the user

Mail template example:

```
From: "DrWeb-ICAP" <drweb-icapd>
To: "System Administrator" <$HOSTMASTER$>
Subject: $REASON$
Content-Type: multipart/mixed;
              boundary="001-DrWeb-Icapd-Notification"
MIME-Version: 1.0
Precedence: junk
X-Antivirus-Ticket: Dr.Web notification.

--001-DrWeb-Icapd-Notification
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 7bit

Url: $URL$
Reason: $REASON$
Action: $ACTION$
Time: $TIME$
Client-IP: $IP$
Object size: $SIZE$

--001-DrWeb-Icapd-Notification
Content-Type: text/html; charset=iso-8859-1
Content-Transfer-Encoding: 7bit

$HTML_PAGE$
--001-DrWeb-Icapd-Notification--
```

It is strongly recommended not to change the mail template internal structure unless it is essential. Otherwise, the structure of MIME parts and headers might be accidentally corrupted.

You can change text of templates in the web interface **Dr.Web Console for UNIX Internet Gateways** on the Template page.

Dr.Web Console for UNIX Internet Gateways

Setup and configuration of **Dr.Web for Unix Internet gateways** can be performed via the web interface **Dr.Web Console for UNIX Internet Gateways**. It is implemented as a plug-in to **Webmin** (for detailed information on **Webmin** interface, visit its official website at http://www.webmin.com/).

To achieve optimal performance of web interface **Dr.Web Console for UNIX Internet Gateways**, ensure that the following **Perl** modules are installed on your system:

- **`XML::Parser`** – module for parsing XML documents
- **`XML::XPath`** – set of modules for parsing XPath statements
- **`CGI`** – module enabling operation with Common Gateway Interface
- **`CGI::Carp`** – module for operation with error log
- **`Cwd`** – module for detection of current working directory of any process
- **`Data::Dumper`** – module for writing arbitrary data structures to memory and reading from it
- **`Text::Iconv`** — module that provides a Perl interface to `iconv()` encoding conversion function
- **`Encode`** and **`Encode::Detect`** – modules used for encoding conversion
- **`perl-devel`** (or **`libperl-dev`**, depending on the UNIX distribution)
- **`File::Basename`** – module for parsing file names
- **`File::Stat`** – object-oriented interface for a `stat()` function
- **`POSIX`** – interface for POSIX-compliant functions
- **`JSON`** – **Perl** module for parsing and converting to JSON (JavaScript Object Notation)
- **`Encode::CN`** – module used for Chinese character encoding
- **`Encode::HanExtr`** – module with additional set of Chinese character encodings
- **`Switch`** – module that enables use of `switch-case` statements.

It is recommended to install missing modules from the command line. For that purpose, `root` privileges are required. Names of the modules may vary, but typically they are included in the following packages: `perl-Convert-BinHex`, `perl-IO-stringy`, `perl-MIME-tools`, `perl-XML-Parser`, `perl-XML-XPath`. To install modules in `rpm` systems, it is recommended to choose `noarch.rpm` packages.

Appearance of the web interface may differ from the given screenshots depending on the **Webmin** version and used browser.

Due to features of **Webmin** implementation, **Dr.Web Console for UNIX Internet Gateways** web interface does not display correctly in **Internet Explorer 7**. If problems with displaying of web pages occur, try to use **Internet Explorer 8** or **9** (and later) or use another browser.

Installation

To start working with **Dr.Web Console for UNIX Internet Gateways**, do the following:

- install **Webmin**
- install **Dr.Web Console for UNIX Internet Gateways** plug-in located in the `%bin_dir/web/` directory.

Webmin configuration and installation of modules is performed with the use of **Webmin** web

interface.

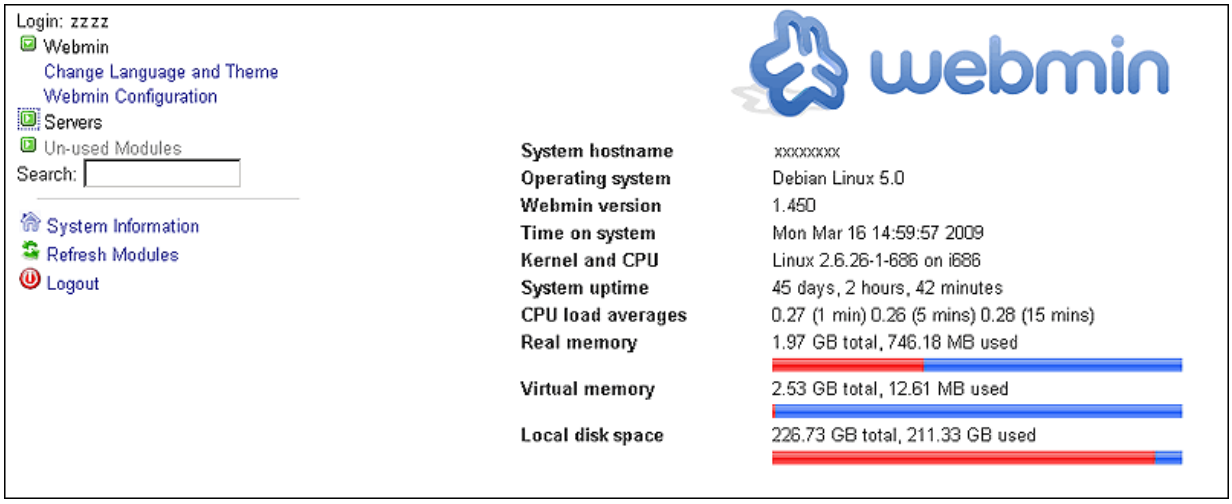


Figure 18. Main page of Webmin web interface

To install additional modules, click **Webmin Configuration** on the main menu and then click **Webmin Modules** on the open page.

Figure 19. Webmin configuration

To install required modules

1. Click the **Browse** button near the **From local file** text field on the **Webmin Modules** page. A new browser window opens to provide navigation through folders and files.
2. Choose the corresponding installation package from the list (`%bin_dir/web/drweb-icapd-web.wbm.gz` by default).

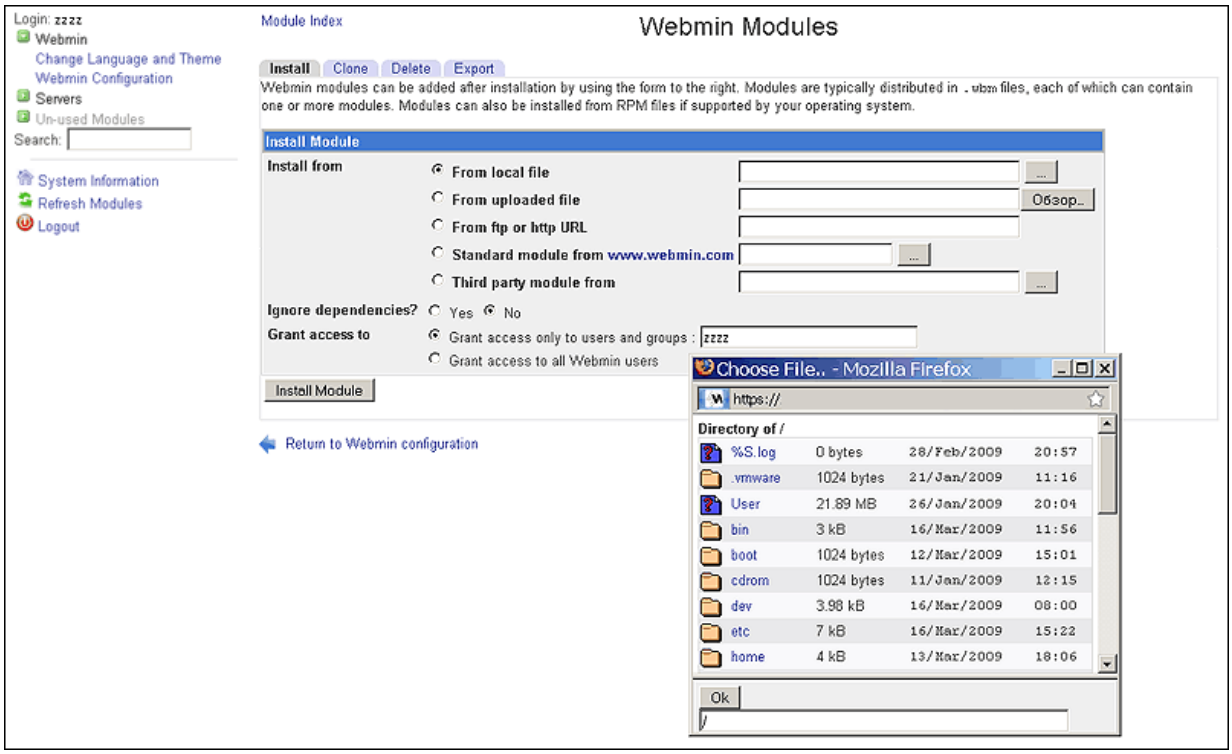


Figure 20. Webmin modules

3. After you click an item from the list, path to this item is added to the field below. If you click the item twice, the folder opens. With the second click on the previously selected file, navigation window closes, and the full path to the selected file appears in **From local file text** field. You may also click **OK** after you select a required file.
4. After you select an installation package file, click **Install Module**.
5. When the installation completes, a link to the new **Dr.Web Console for UNIX Internet Gateways** module appears in the **Servers** section of the main menu.

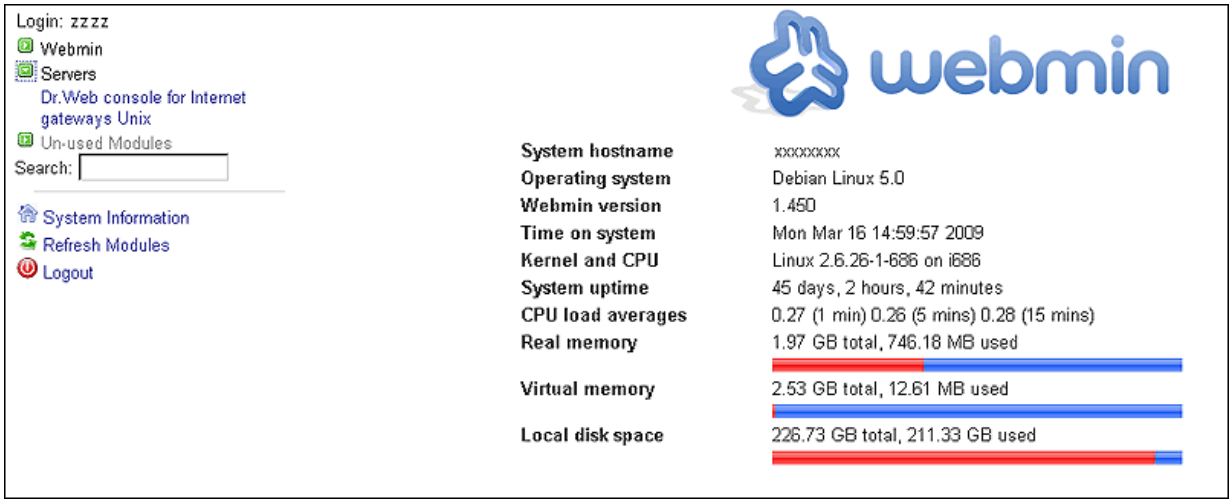


Figure 21. Dr.Web Console for UNIX Internet Gateways module

If you use **Webmin** 1.680 or later, you should also add the following line to its configuration file (usually, this is the file `/etc/webmin/config`):

```
no_content_security_policy=1
```

Basic Configuration

To open general settings of **Dr.Web Console for UNIX Internet Gateways**, click on the top pane of the corresponding page. On the open page, you can specify the path to the configuration file `drweb-icapd.ini`, path to the init script, path to the directory containing black and white lists, number of files shown per page in the **Quarantine** section, and the operation mode.

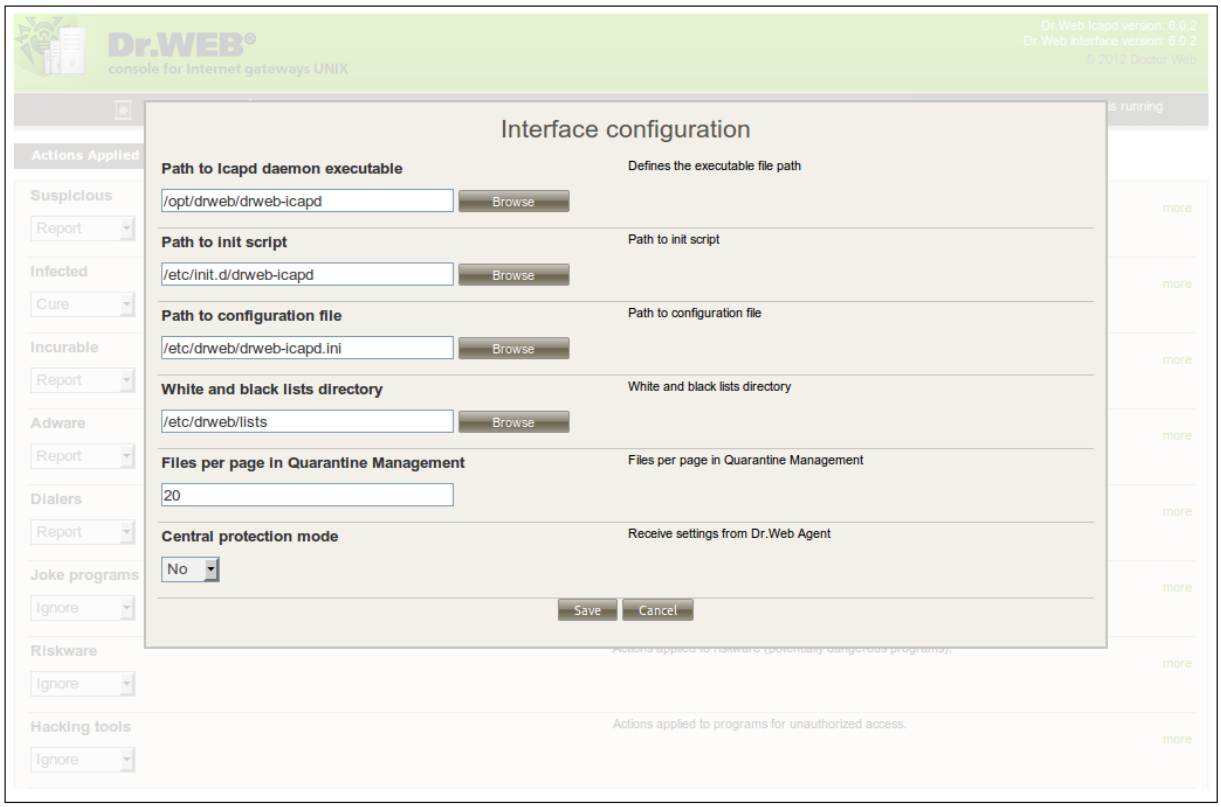


Figure 22. Module configuration

User Interface

When navigating within the **Dr.Web Console for UNIX Internet Gateways** sections, you cannot open the previous page using the standard **Back** function. If you click **Back** or use the corresponding key combination, the previous section of the main menu opens.

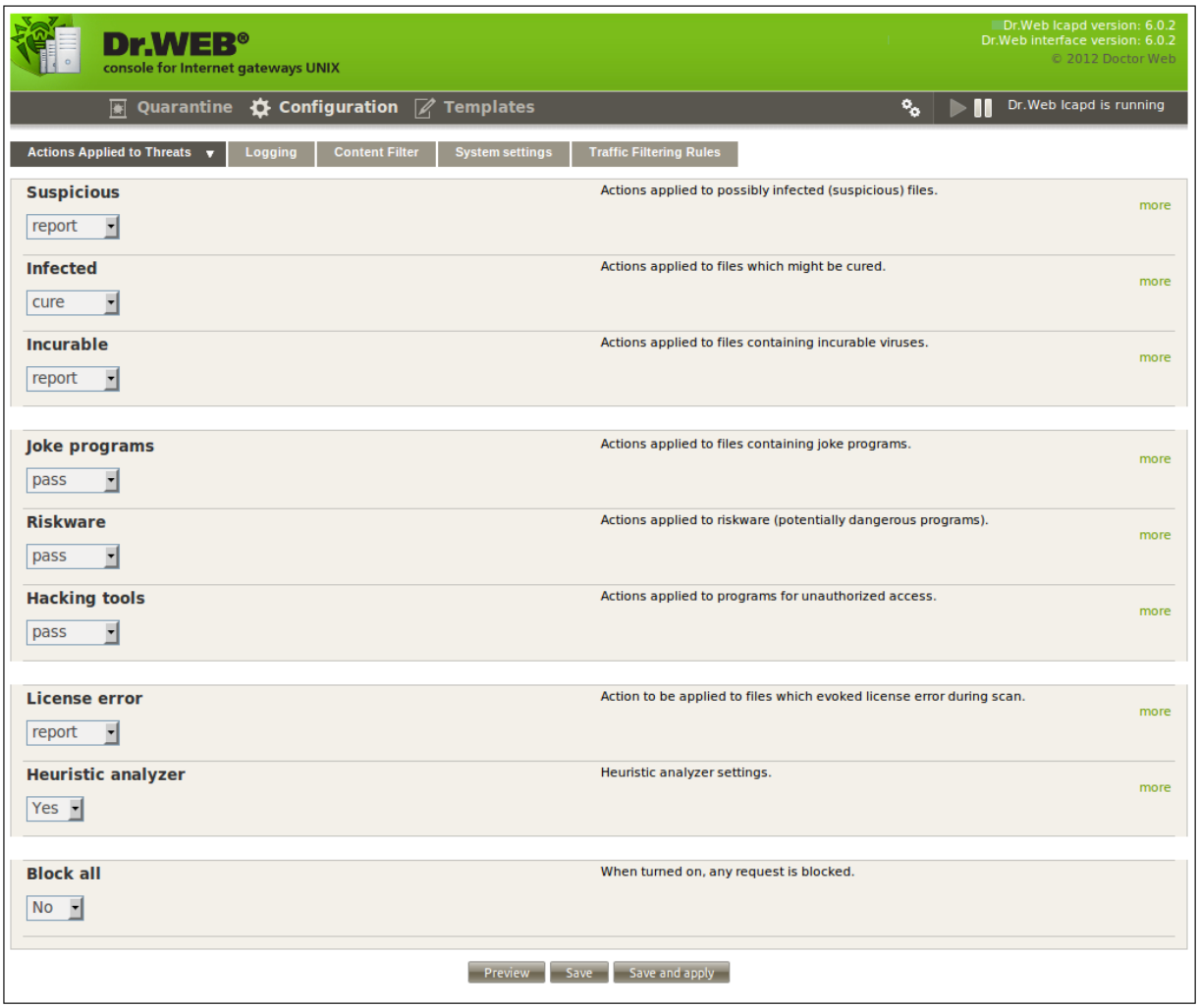


Figure 23. Dr.Web console for UNIX Internet gateways

Next to the module header, information on the current versions of **Dr.Web ICAPD** and **Dr.Web for Unix Internet gateways** web interface displays.

Under the module header, you can see the following three sections: **Quarantine**, **Configuration** and **Templates**. By default, the **Actions Applied to Threats** tab of the **Configuration** section opens.

Next to the section headers, you can see the following buttons: **Interface Setup**, **Start Dr.Web Icapd** and **Stop Dr.Web Icapd**, as well as the current **Dr.Web ICAPD** status.

Configuration

The **Configuration** page contains the following tabs, where you can configure **Dr.Web for Unix Internet gateways** operation:

- Actions Applied to Threats, where you can specify actions to be applied to detected threats of

different types.

- Logging, where you can adjust logging settings
- Content Filter, where you can configure blocking of web pages by their content type.
- System settings , where you can configure user-defined black and white lists of Internet resources, specify paths to the license key file and quarantine directory, as well as configure administrator notification settings.
- Traffic Filtering Rules, where you can specify rules to process files depending on their type and size.

If **Dr.Web for Unix Internet gateways** operates in the central protection mode, the administrator of the central protection server can block an option to adjust the settings. If so, users cannot configure **Dr.Web for Unix Internet gateways** settings.

To adjust the settings, either select required parameter values from the drop-down lists or type the values in the corresponding text fields. If it is required to add another value, click . After you change a parameter value, you can immediately undo the change by clicking or restore the default value by clicking .

To view the changes, click **Preview**. On the open page, you can choose whether or not to save the adjustments (to undo a change, clear the corresponding checkbox). To continue adjusting the settings, click **Continue Editing** and the previous page will open. To cancel all of the changes, click **Cancel changes.** To save the changes, click either **Save** or **Apply and Save.**

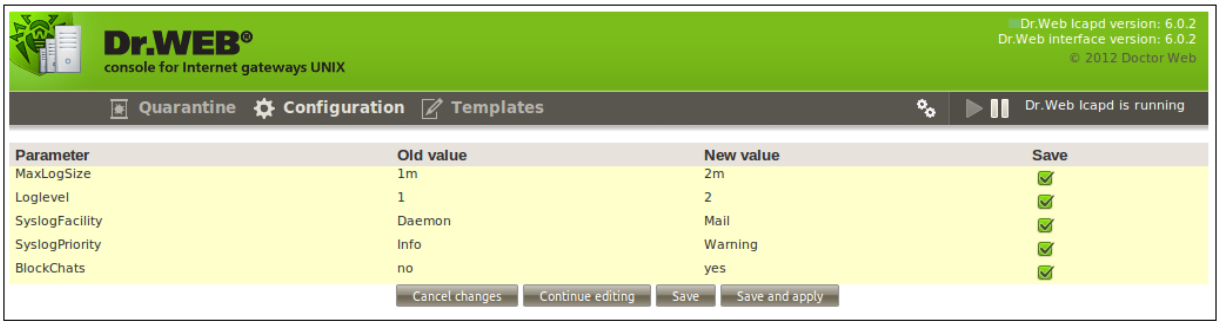


Figure 24. Preview page

After you click either the **Save** or **Apply and Save** button, notification on the configuration being saved displays on the screen. Click the notification to return to the settings page.

Actions Applied to Threats

On this tab, you can specify actions to be applied to detected threats of different types.

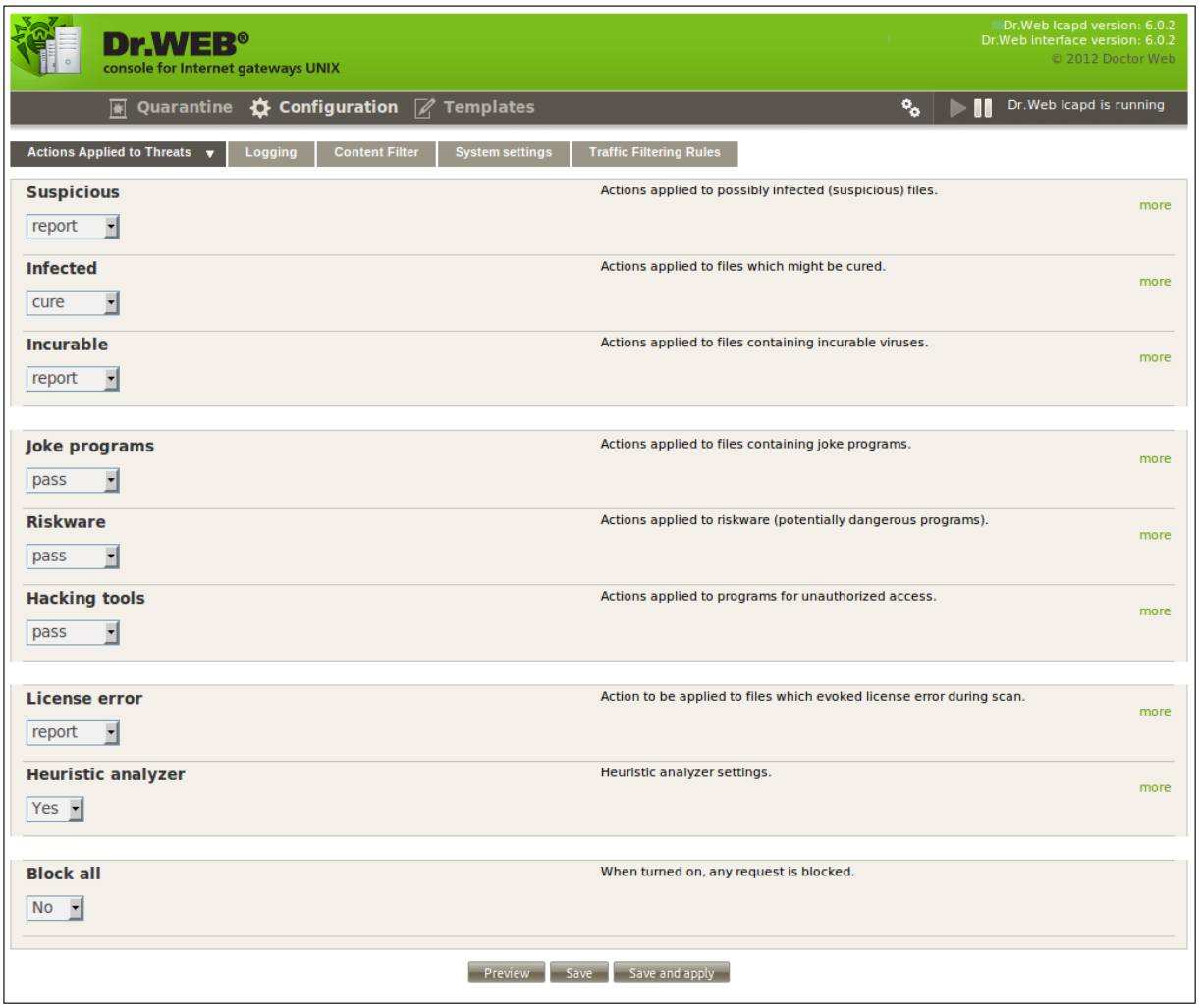


Figure 25. Actions Applied to Threats

Every parameter has a drop-down menu with the list of possible actions. For details on available actions, click **more**.

Logging

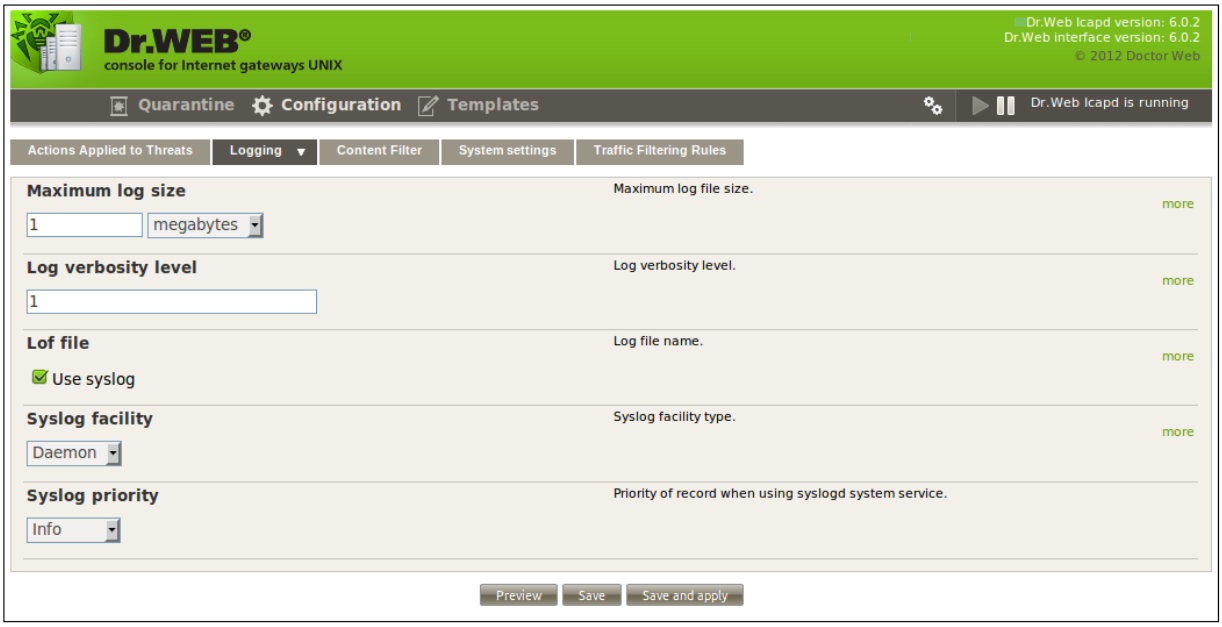


Figure 26. Logging

On this tab, you can select required parameter values either by selecting them in the corresponding drop-down lists or typing the values in the text fields.

To select a file where logged information is to be saved, clear the **Use syslog** checkbox and click **Browse**. A separate browser window will open, where you can select the required file.

Content Filter

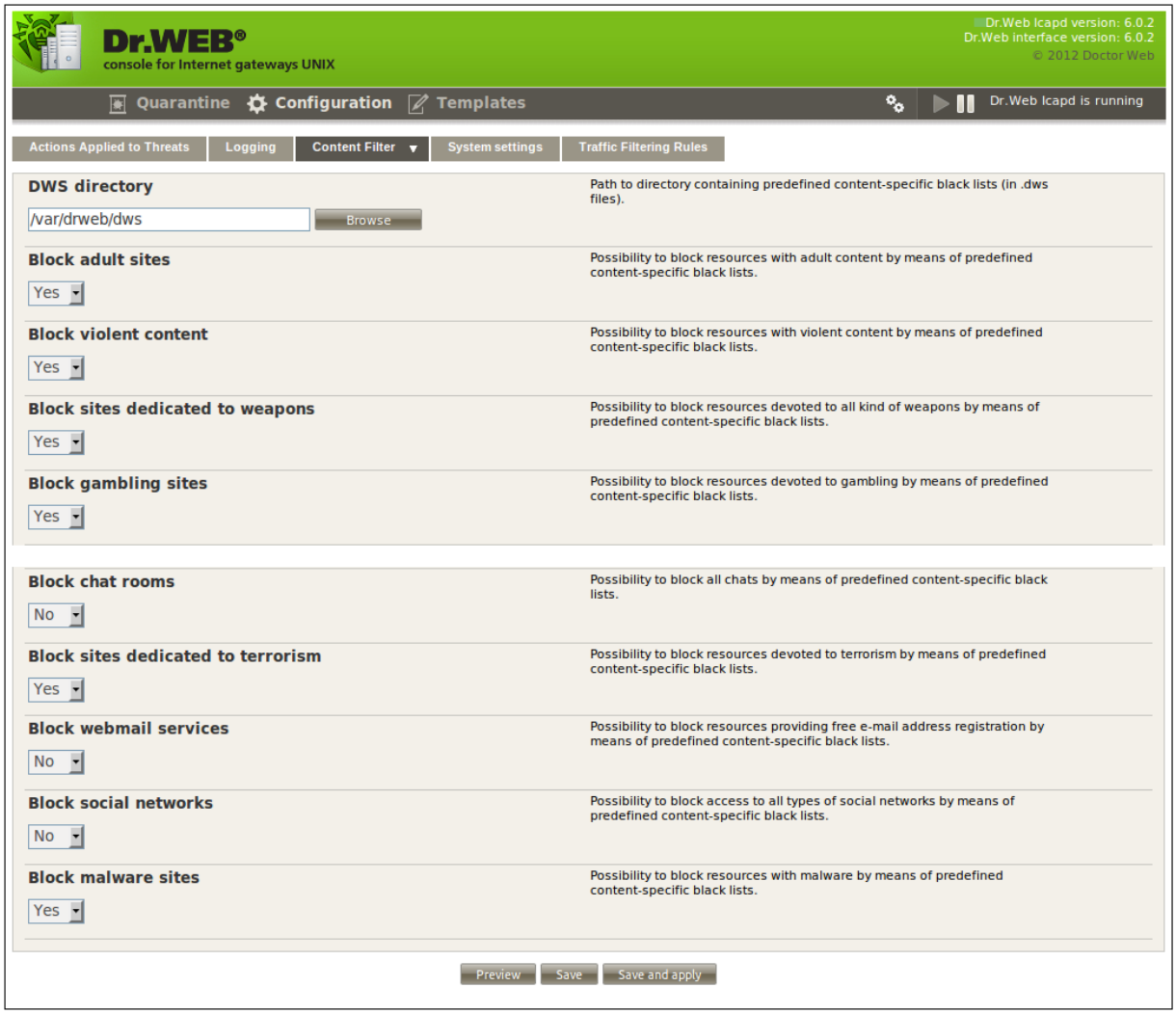


Figure 27. Content Filter

On this tab, you can configure blocking of web pages by their content type: for example, websites with adult content, websites providing information on drugs. To set the required value, select it in the corresponding drop-down list. Only the following two values are available for each parameter: **Yes** and **No**. You can specify a path to the directory with updated black lists either by typing the path in the text field of the `DwsDirectory` parameter or by clicking **Browse** and selecting the required folder in the open window.

System Settings

On this page, you can adjust user-defined black and white lists of Internet resources, specify paths to the license key file and **Quarantine** directory, and configure administrator notification settings.

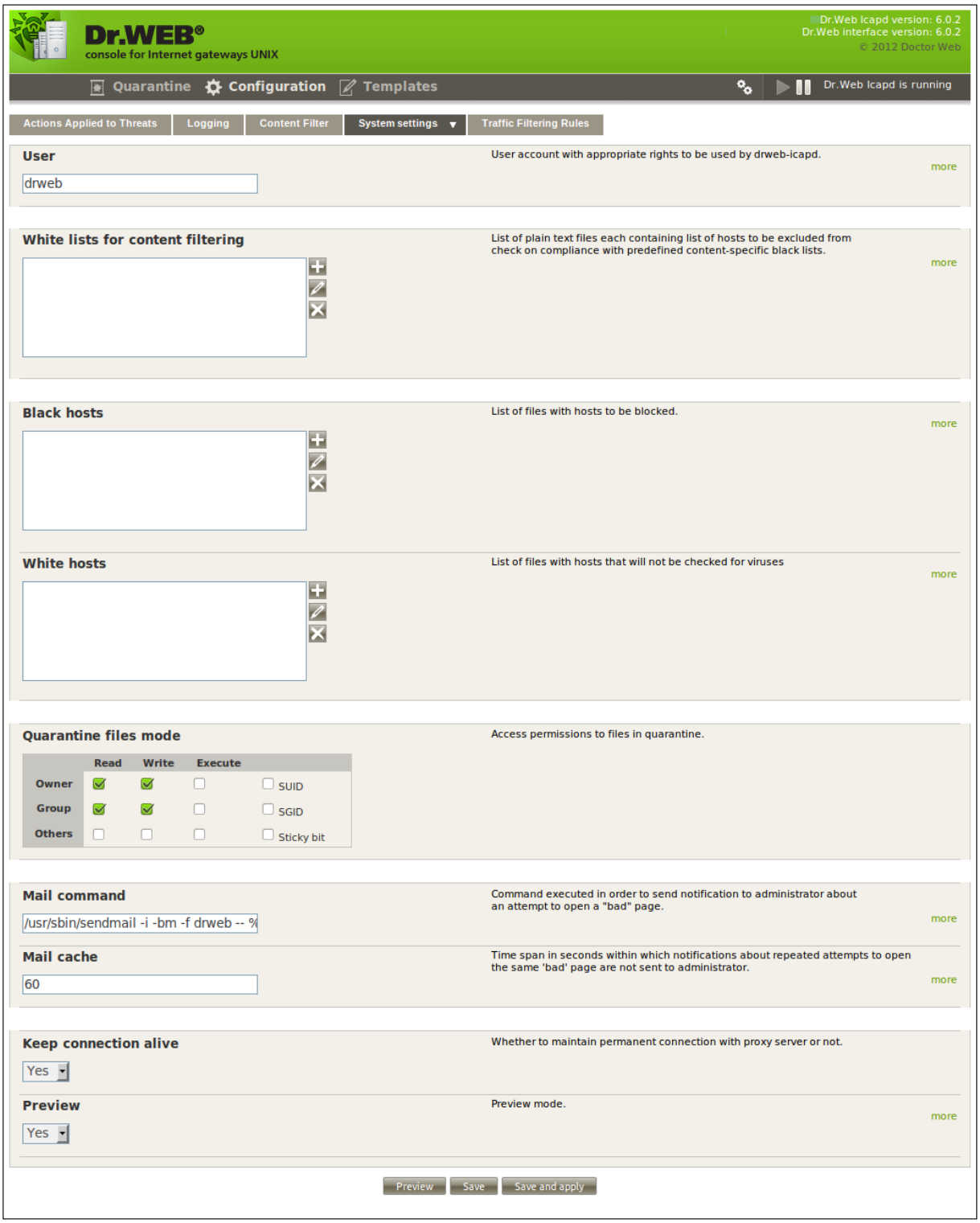


Figure 28. System settings

In the **Black Hosts** section, you can create user-defined black lists. For that purpose, click  , enter the name of the new list (no spaces) and names of hosts that are to be blocked.

Figure 29. Creating a user-defined list

In the **White hosts** and **White lists for content filtering** sections, you can create a user-defined white list. The procedure of creating a white list is the same as for a black list. Hosts specified in the **White Hosts** list will not be checked for viruses. Hosts specified in files from the **White lists for content filtering** list will not be checked for matching the content-specific black lists.

Traffic Filtering Rules

On this page, you can specify rules to filter files by their MIME type, which is a file type identifier used on the Internet.

Mime type identifier consists of two parts: main and additional. For example, `application/octet-stream` corresponds to executables with `.com` and `.exe` extension; `image/any` corresponds to any file containing images; `audio/mpeg` corresponds to audio files of the `mp1`, `mp2` и `mp3` formats.

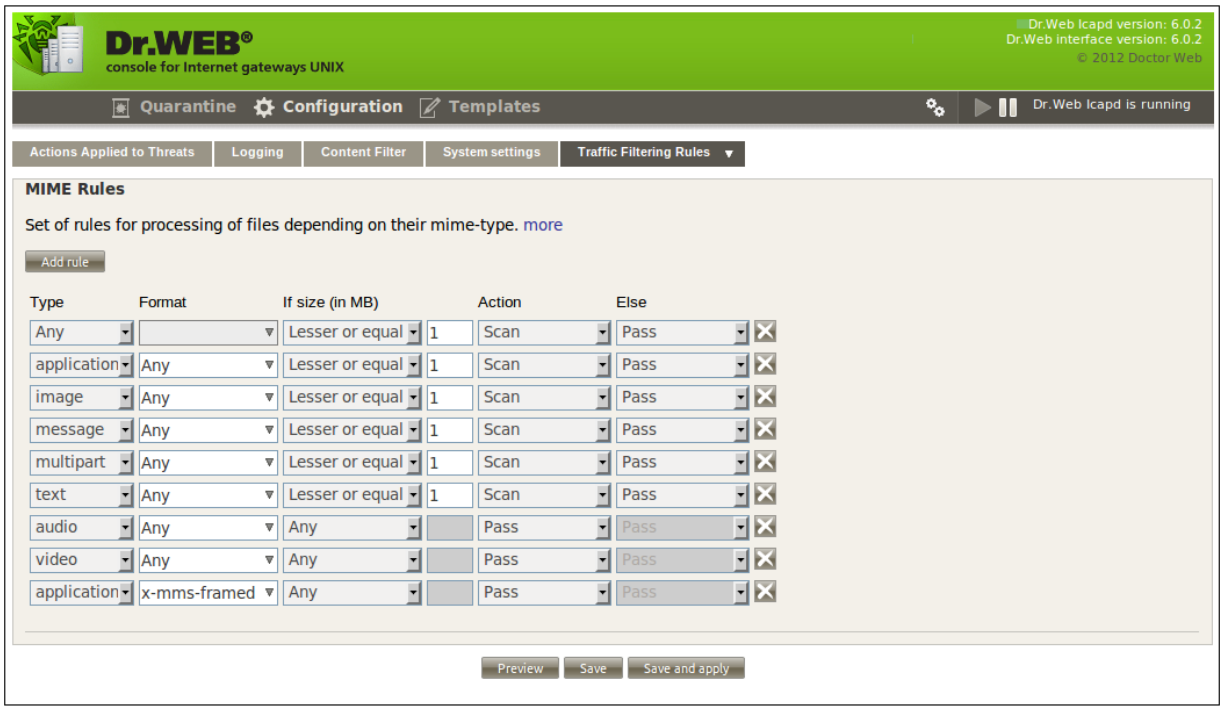


Figure 30. Traffic Filtering Rules

Click **Add rule** to add a new rule. To delete a rule, click the corresponding button next to the rule.

Each rule contains the following fields:

- **Type** - basic MIME type of the file.
  - **Any** - file of any type
  - **application** - executables and archived files, documents in PDF, MS Word, etc.
  - **audio** - audio files (mp3, wav, wma, etc.)
  - **image** - images (gif, jpg, png, svg, etc.)
  - **message** - messages between web servers and clients
  - **multipart** - containers (mail files, packed files)
  - **text** - text or source code (html, xml, css, etc.)
  - **video** - video files (mpeg-1, mp4, wma)
  - **model** - 3D model files.
- **Format** - additional MIME type of the file. Can be selected from a list or entered manually.
- **If size** - specify whether or not to filter files by size. If yes, enter the size (in megabytes) in the corresponding field.
- **Action** - specify an action for the files of the specified size.
- **Else** - specify an action for other files of this type.

When specifying a traffic filtering rule, use the following syntax:

```
<MIME type> <action1> <size> <action2>
```

`<action1>` is applied to the file of the `<MIME-type>` type if the file size (in megabytes) exceeds the specified `<size>` value. Otherwise, `<action2>` is applied.

This syntax is used to write traffic filtering rules to the configuration file and display them in **Dr.Web Console for UNIX Internet Gateways** web interface. If the value in the **If size** field is set to **Greater,** the rule will be adjusted to the form described above.

Example:

Let us assume that a user specified a rule according to which images of pgn format and size of which is greater than 10 MB must be checked for viruses. If the file size is less than or equal to 10 MB, the file is passed.

After being saved, the rule is adjusted to the form described above and displays as follows:

Thus, despite the difference in notation, the adjusted rule has the similar effect as the original one.

Quarantine

Quarantine page contains the list of links to quarantined files. A suspicious file is moved to **Quarantine** and its name is created from address of those web pages from which the file was downloaded.

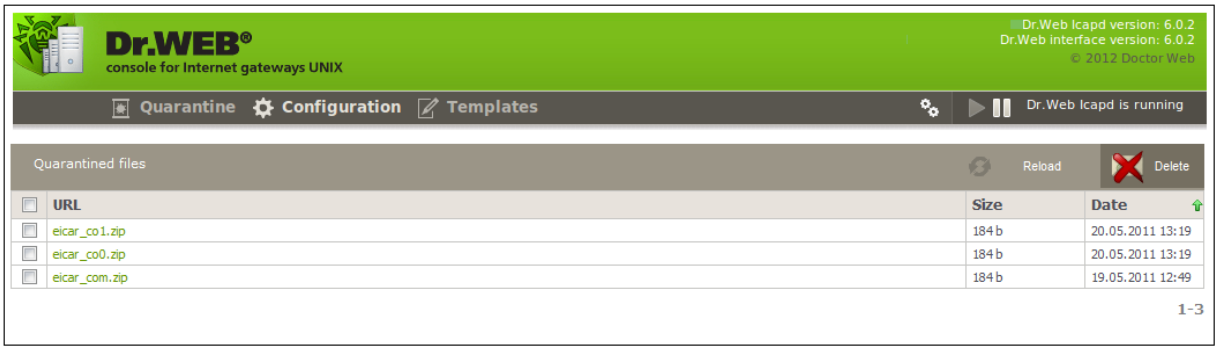


Figure 31. Quarantine

To remove a file from the **Quarantine** directory, select the corresponding checkbox and click **Delete**.

Templates

This section contains templates of web pages that are generated and displayed to the end user on attempt to access blocked content (if actions **report** or **move** are specified for the corresponding parameters in the configuration section), as well as a template to generate a notification for administrator.

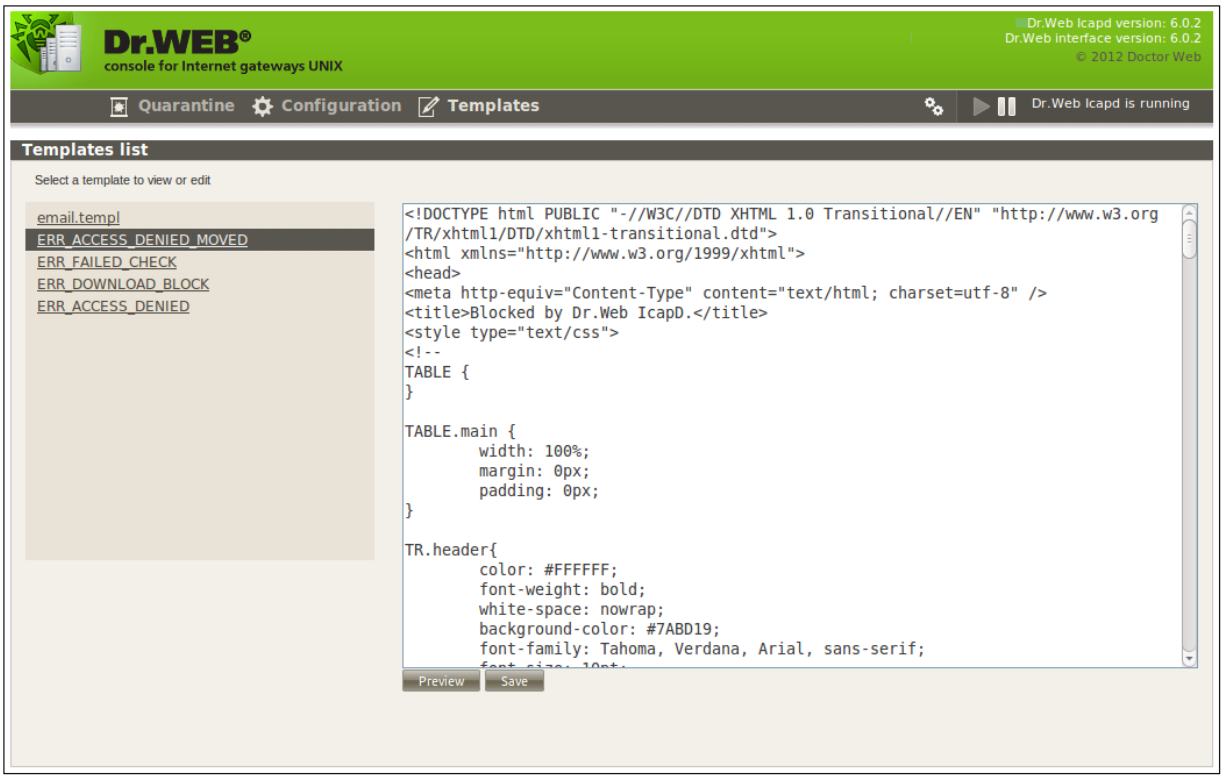


Figure 32. Templates

Template name	Description
ERR_FAILED_CHECK	Template to report a file scan error.
ERR_DOWNLOAD_BLOCK	Template to report an access error on attempt to open a blocked web page (from a black list).
ERR_ACCESS_DENIED	Template to report a download error on attempt to download an forbidden object. This template is used if **report** action is specified.

ERR_ACCESS_DENIED_MOVED	Template to report a download error on attempt to download an forbidden object. This template is used if **move** action is specified.
email.templ	Template or e-mail messages sent to the administrator on attempt to access a blocked content.

If necessary, you can change a structure and text of any template. For details, refer to the Notification Templates section.

Running in Enterprise Mode

To start **Dr.Web Console for UNIX Internet Gateways** in the central protection mode, configure **Dr.Web Agent** as described in the corresponding section. After making necessary changes, click on the navigational menu at the top of the page. In the open window, set `Central Protection Mode` parameter value to `Yes`.

`Central Protection Mode` parameter can have one of the following two values:

- `No` – in this mode **Dr.Web Console for UNIX Internet Gateways** interacts with local configuration file and does not have access to the configuration received by **Dr.Web Agent** from **Dr.Web Enterprise Server**. Changes made to the configuration in this mode take effect only after **Dr.Web Agent** is set to operate in the `Standalone` mode.
- `Yes` – in this mode **Dr.Web Console for UNIX Internet Gateways** receives configuration from the **Dr.Web Agent** socket. If **Dr.Web Agent** is operating in the `Stanalone` mode, the following warning is output to the **Dr.Web Console for UNIX Internet Gateways**:
  `Receiving settings error: unable to establish connection with Dr.Web Agent.`

If there is a problem connecting to **Dr.Web Enterprise Server**, the following behaviours of **Dr.Web Console for UNIX Internet Gateways** are possible:

- If **Dr.Web Enterprise Server** is unavailable upon the initial connection or authorization process fails, **Dr.Web Agent** terminates. In this case, check the settings and try to restart **Dr.Web Agent** and **Dr.Web Console for UNIX Internet Gateways**.
- If connection to **Dr.Web Enterprise Server** was established earlier, but now the server is temporary unavailable (for example, in the event of connection problems), **Dr.Web Agent** uses backup copies of configuration files that were previously received from the server. These files are encrypted and must not be edit by users. Edited files become invalid.

Configuring User Permissions

When **Dr.Web Agent** is running in the `Enterprise` mode, **Dr.Web Control Center** administrator can partially or completely block user permission to configure **Dr.Web** components installed on the workstation.

To set permissions of a workstation user:

- Enter **Dr.Web Control Center**. Note that the administrator must have sufficient privileges to adjust settings of **Dr.Web** anti-virus software.
- On the main menu, select **Network**, then click the workstation name in the hierarchical list. On the open control menu (left pane), select **Permissions.** This opens the permission configuration window.

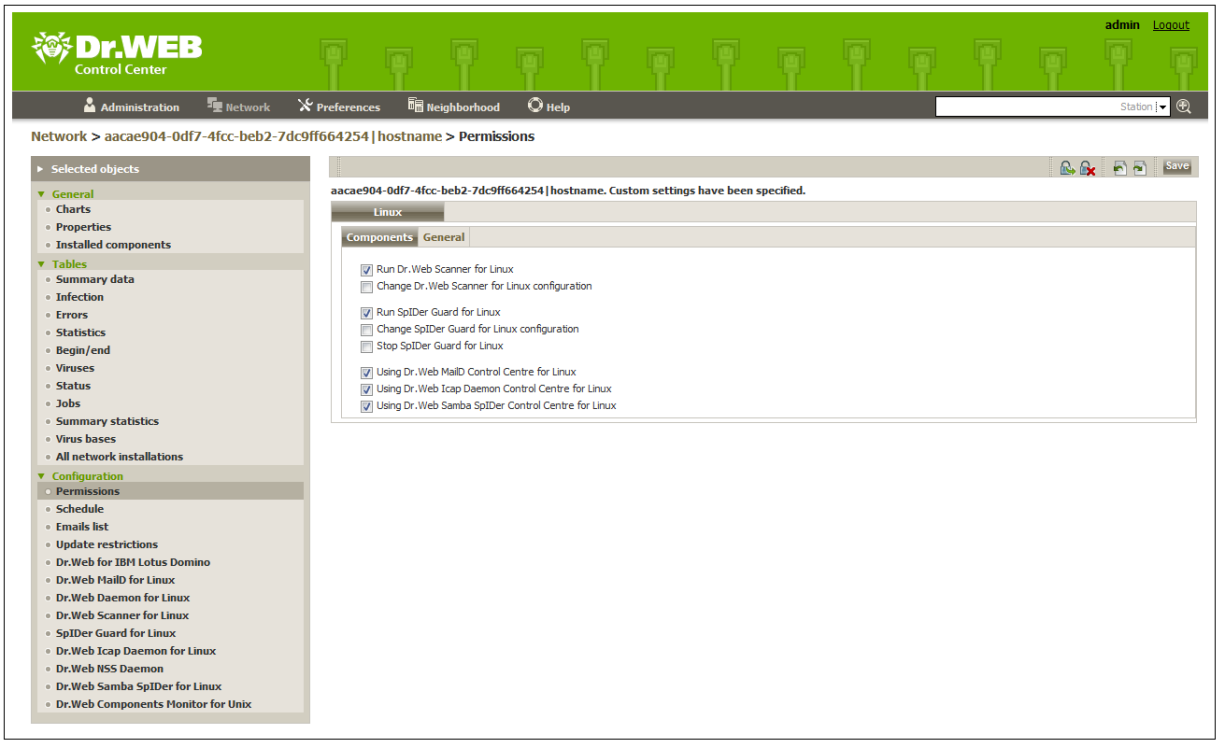


Figure 33. User permissions configuration

- In the **Components** section, select components to be available for the workstation user to change. For example, to allow the workstation user to adjust **Dr.Web for Unix Internet gateways** configuration, select the **Using Dr.Web Icap Daemon Control Centre for Linux** checkbox and click **Save.**
- To disable the workstation user to adjust **Dr.Web for Unix Internet gateways** configuration, clear the **Using Dr.Web Icap Daemon Control Centre for Linux** checkbox and click **Save.** In this mode, **Console** displays the corresponding warning and **Apply and Save Settings, Preview** and **Save** buttons become unavailable.

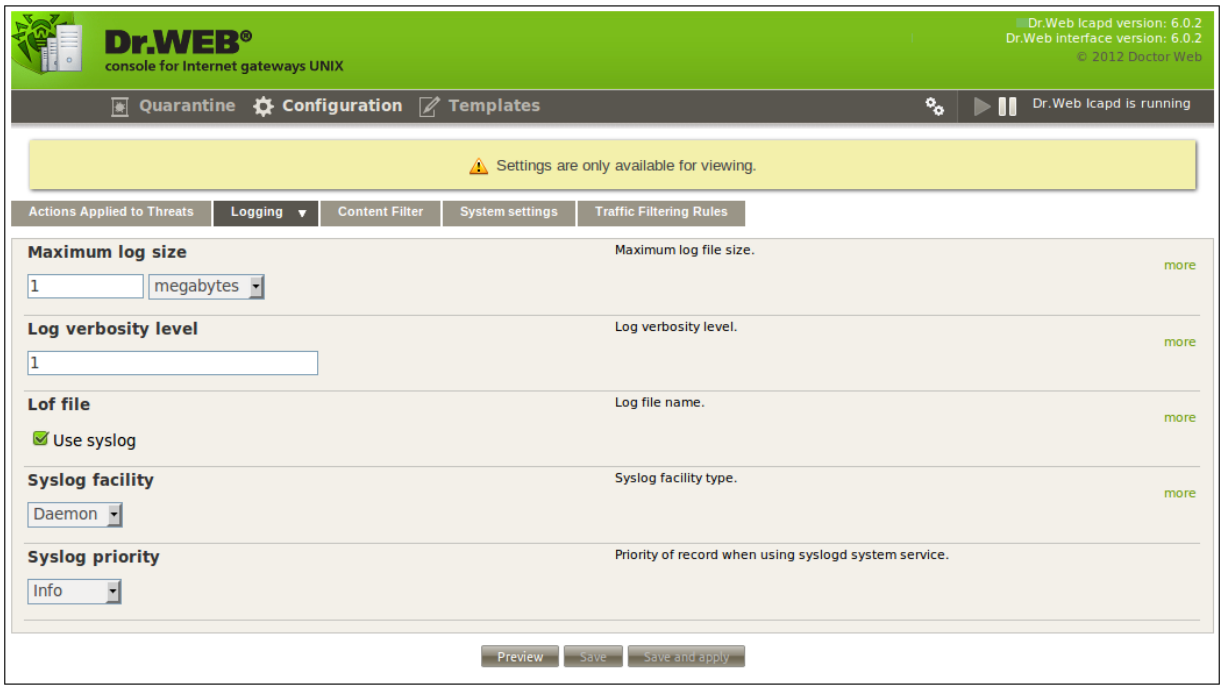


Figure 34. Read-only user permissions

Configuring Workstation

When a new workstation is created, its configuration settings are inherited from a group it belongs to. That group is called the *primary group*. If the settings of the primary group are modified, these changes are inherited by all workstations included into the group, unless the workstation configuration is customized. When creating a workstation, you can specify what group is to be treated as primary. By default, the primary group is the **Everyone** group.

Inheritance in nested groups depends on the group hierarchy. If for a station no custom settings are specified, it inherits configuration from its parent group, and this process repeats recursively. Therefore, search for the group configuration is performed upwards through the hierarchical tree of nested groups, starting from the primary group of the station and further until the root group is reached. If no custom settings are found, the workstation inherits configuration of the **Everyone** group.

Example:

The structure of a hierarchical list is as follows:

```
Network
├Everyone
└Group1
  └Group2
    └Group3
      └ Group4
        └  Station1
```

`Group4` is the primary group for `Station1`. To determine the settings to be inherited by `Station1`, the search is performed in the following order: `Station1` → `Group4` → `Group3` → `Group2` → `Group1` → `Everyone`.

You can edit configuration inherited from the primary group in two ways:

- Using **Dr.Web Control Center** interface. To edit configuration, select **Network** on the main menu, then click the workstation name in the hierarchical list. On the control menu (on the left pane), select the component you want to configure. You need the corresponding permissions to perform this operation. The configuration process is similar to the one via Console. When necessary changes are made, click **Save** to save them.

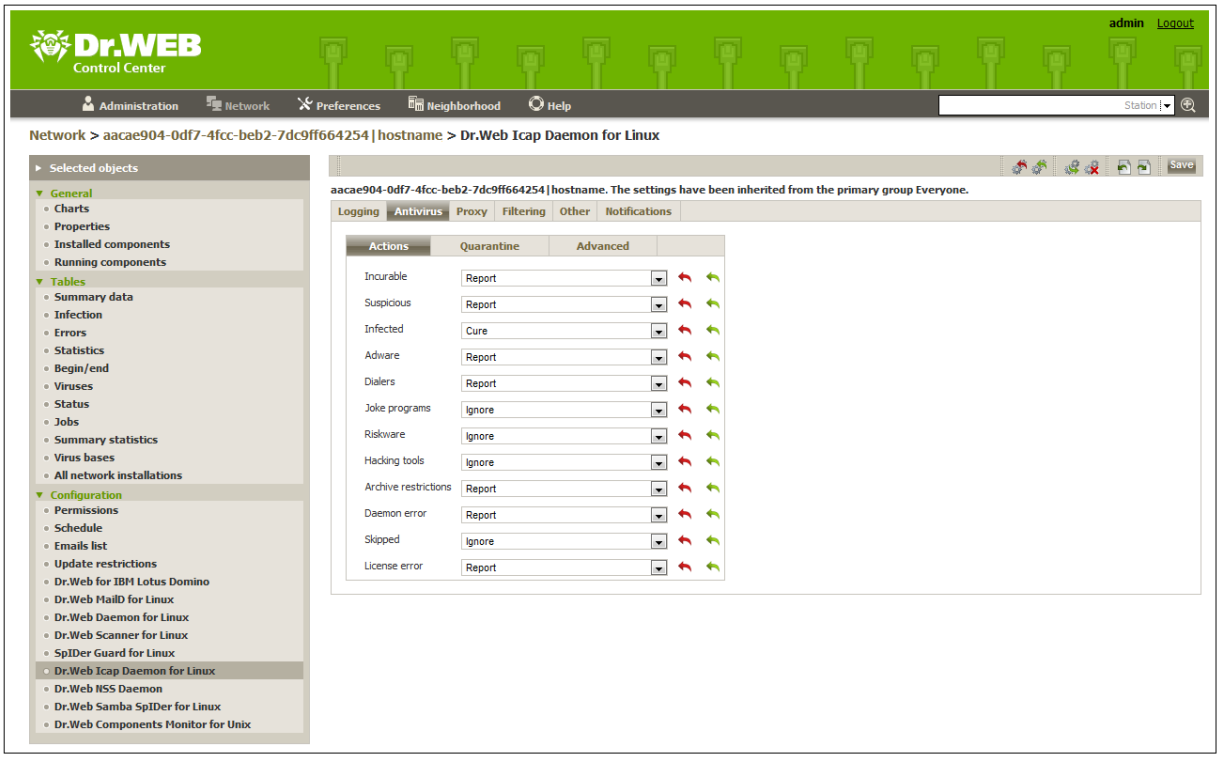


Figure 35. Configuration of Dr.Web Icap Daemon for Linux via Dr.Web Control Center interface

If appropriate permissions are set, parameters can be reconfigured via **Dr.Web Console for UNIX Internet Gateways**. The configuration process is similar to the one in the Standalone mode. If the workstation user has insufficient privileges for that, settings are open in read-only mode.

Types of Administrator Accounts

There are four types of administrator accounts:

- *Administrators with full rights* have exclusive rights for management of **Dr.Web Enterprise Server** and **Anti-virus network**. They can view and edit the **Anti-virus network** configuration and create new administrator accounts. An administrator with full rights can configure the anti-virus software installed on the workstation, limit and disable user intervention into anti-virus software administration.

  An administrator with full rights can view and edit the list of current administrator accounts.

- *Administrators with read-only rights* can only view **Anti-virus network** settings and its separate elements, but cannot modify them.

- *Group Administrators with full rights* have access to all system groups and those custom groups which they are allowed to manage (including nested groups). *Group Administrator* accounts can be created for custom groups only (see Administrator manual for **Dr.Web® Enterprise Security Suite**). In the hierarchical tree, only those groups are displayed for *group administrators* which they are allowed to access.

  The list of current administrator accounts is not available for *Group Administrators*.

- *Group Administrators* with read-only rights can be granted full rights to adjust the available groups or read-only rights.

- *Default administrators* with full rights created automatically during **Dr.Web Enterprise Server** (the

admin account).

Thus, *Administrators with full rights* can:

- Add new and delete already existing administrator accounts.
- Adjust settings for all administrators of **Anti-virus network**.

Group administrators and *administrators with read-only rights* can:

- Adjust some of their account settings.

Contacts

Dr.Web for Unix Internet gateways solution is constantly improved. You can find news and the latest information on available updates on the website at:

http://www.drweb.com/

Sales department:

http://buy.drweb.com/

Technical support:

http://support.drweb.com/

Please include the following information in the problem report:

- full name and version of your operating system;
- versions of **Dr.Web for Unix Internet gateways** modules;
- configuration files of all modules;
- log files of all modules.

Appendix. The License Policy

Dr.Web for Unix Internet gateways solution is available as a separate product and as a part of «universal» and «economy» **Dr.Web** kits. Types of licenses vary correspondingly.

All licenses can be purchased for definite terms, i.e. for 1, 2 or 3 years. Amount of protected file servers may also vary. License terms, their quantitative parameters and limitations may be different for various regional partners of **Doctor Web**, or may be revised hereafter.To learn more about regional license terms, contact our partner in your region. List of the **Doctor Web** trusted partners can be found on the corporate web site http://partners.drweb.com/.

During the whole license term client have the right to download updates from the **Dr.Web Global Updating System** (**Dr.Web GUS**) servers and to receive a technical support from **Doctor Web** and its partners.

Protection of Internet gateways

This license for **Dr.Web Daemon** and **Dr.Web ICAPD** components allow to use **Dr.Web Daemon** for scanning of incoming HTTP-traffic for viruses, and **Dr.Web ICAPD** – for integration of the whole system with proxy servers that support ICAP protocol (**Squid** and **SafeSquid**). In solutions based on **Squid** proxy server it is also possible to set up scanning of incoming FTP-traffic.

Dr.Web for Unix Internet gateways solution is being licensed according to the number of users working through an Internet gateway. Minimal license covers protection of 25 users.

Components continue to work 24 hours after the license has expired.

Address of product's page: http://products.drweb.com/gateway/unix/?lng=en