Dr.WEB

Enterprise Security Suite

Managing Dr.Web for UNIX Mail Servers

Trademarks

Dr.Web, SpIDer Mail, SpIDer Guard, CureIt!, CureNet!, AV-Desk, KATANA and the Dr.WEB logo are trademarks and registered trademarks of Doctor Web in Russia and/or other countries. Other trademarks, registered trademarks and company names used in this document are property of their respective owners.

Disclaimer

In no event shall Doctor Web and its resellers or distributors be liable for errors or omissions, or any loss of profit or any other damage caused or alleged to be caused directly or indirectly by this document, the use of or inability to use information contained in this document.

Dr.Web Enterprise Security Suite. Managing Dr.Web for UNIX Mail Servers
Version 11.0.2
Administrator Manual
5/27/2021

Doctor Web Head Office

2-12A, 3rd str. Yamskogo polya, Moscow, Russia, 125124

Website: https://www.drweb.com/

Phone: +7 (495) 789-45-87

Refer to the official website 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 anti-virus 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

Chapter 1. Introduction

1.1. About Manual

This manual is a part of documentation package of anti-virus network administrator and intended to provide detailed information on the organisation of the complex anti-virus protection of corporate computers and mobile devices using Dr.Web Enterprise Security Suite.

The manual is meant for anti-virus network administrator—the employee of organisation who is responsible for the anti-virus protection of workstations and servers of this network.

The manual contains the information about centralized configuration of anti-virus software of workstations which is provided by anti-virus network administrator via the Dr.Web Security Control Center. The manual describes the settings of Dr.Web for UNIX Mail Servers anti-virus solution and features of centralized configuration of the software.

To get additional information, please refer the following manuals:

- **Administrator Manual** of Dr.Web for UNIX Mail Servers anti-virus solution contains the information about configuration of anti-virus software provided on a station directly.
- **Administrator Documentation** of Dr.Web Enterprise Security Suite anti-virus network (includes **Administrator Manual**, **Installation Manual** and **Appendices**) contains the general information on installation and configuration of anti-virus network and, particularly, on operation with Dr.Web Security Control Center.

Before reading these document make sure you have the latest version of the manuals. The manuals are constantly updated and the current version can always be found at the official web site of Doctor Web at https://download.drweb.com/doc/?lng=en

1.2. Conventions and Abbreviations

Conventions

The following symbols and text conventions are used in this guide:

Convention	Comment
(!)	Important note or instruction.
⚠	Warning about possible errors or important notes to which you should pay special attention.
Anti-virus network	A new term or an accent on a term in descriptions.
<IP-address>	Placeholders.
Save	Names of buttons, windows, menu items and other program interface elements.
CTRL	Keyboard keys names.
`/home/user`	Names of files and folders, code examples.
Appendix A	Cross-references on the document chapters or internal hyperlinks to web pages.

Abbreviations

The following abbreviations will be used in the Manual without further interpretation:

- DNS—Domain Name System,
- Dr.Web GUS—Dr.Web Global Update System,
- FQDN—Fully Qualified Domain Name,
- FTP—File Transfer Protocol,
- HTML—HyperText Markup Language,
- HTTP—HyperText Transfer Protocol,
- HTTPS—Hypertext Transfer Protocol Secure,
- IMAP—Internet Message Access Protocol,
- IP—Internet Protocol,
- LAN—Local Area Network,
- MDA—Mail Delivery Agent,
- MIME—Multipurpose Internet Mail Extensions,
- MTA—Mail Transfer Agent,

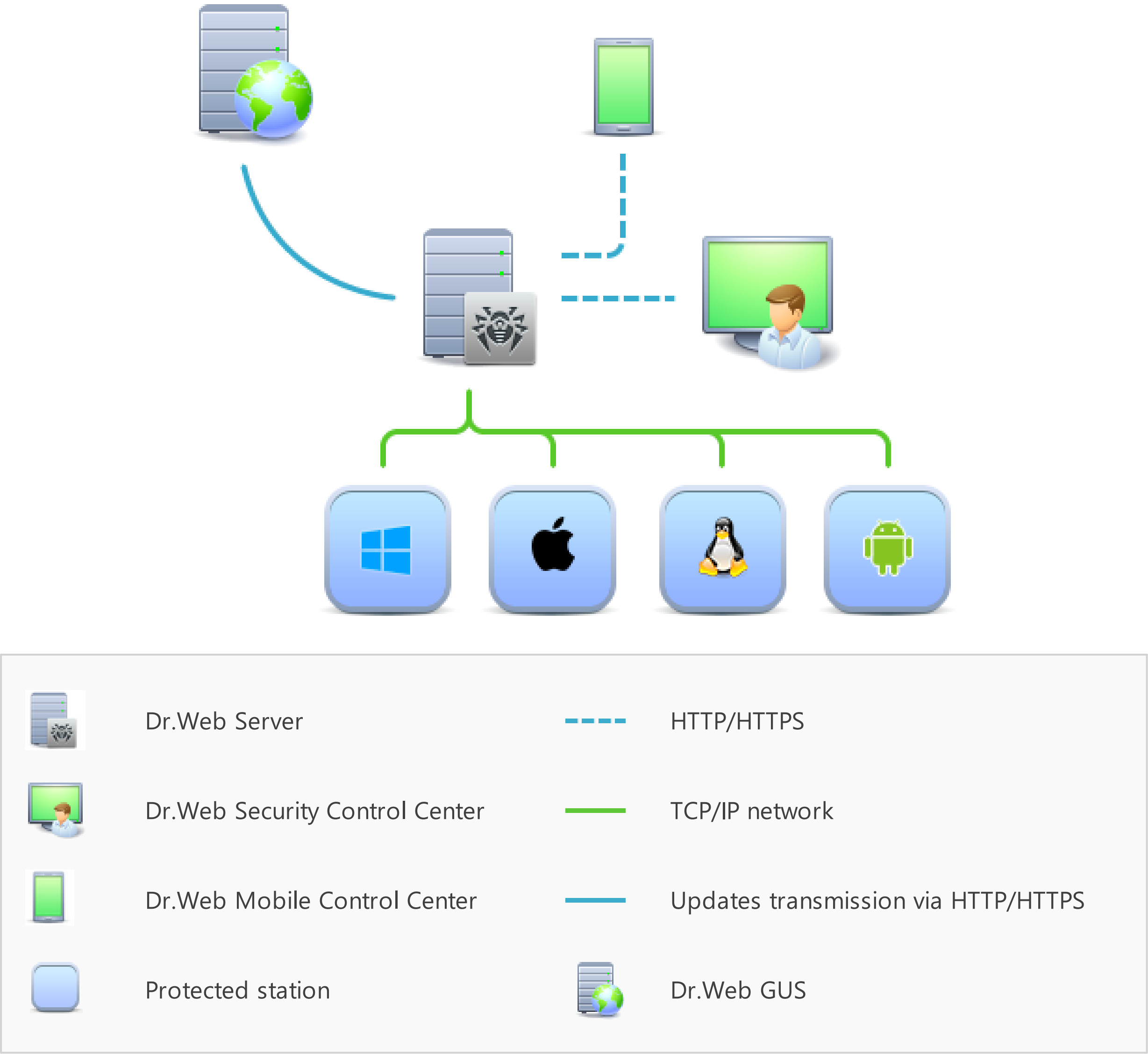
- MUA—Mail User Agent,
- OS—Operating System,
- PC—Personal Computer,
- POP—Post Office Protocol,
- SMTP—Simple Mail Transfer Protocol,
- SNI—Server Name Indication,
- SSL—Secure Socket Layers,
- TCP—Transmission Control Protocol,
- TLS—Transport Layer Security,
- URL—Uniform Resource Locator.

Chapter 2. Dr.Web Enterprise Security Suite

2.1. About Product

Dr.Web Enterprise Security Suite is designed for organization and management of integrated and secure complex anti-virus protection either local company network including mobile devices, or home computers of employers.

An aggregate of computers and mobile devices on which Dr.Web Enterprise Security Suite co-operating components are installed, represents a single *anti-virus network*.


Dr.Web Server
HTTP/HTTPS
Dr.Web Security Control Center
TCP/IP network
Dr.Web Mobile Control Center
Updates transmission via HTTP/HTTPS
Protected station
Dr.Web GUS

The logical structure of the anti-virus network

Dr.Web Enterprise Security Suite anti-virus network has a *client-server* architecture. Its components are installed on a computers and mobile devices of users and administrators as well as on a computers that function as LAN servers. Anti-virus network components exchange information via TCP/IP network protocols. Anti-virus software can be installed (and manage them afterwards) on protected stations either via the LAN, or via the Internet.

2.2. Workstations Protection

Workstations are protected by Dr.Web anti-virus packages designed for correspondent operating systems.

> Protected computer with installed anti-virus package as per its functions in the anti-virus network is called a *workstation* of anti-virus network. Please note: according to its LAN functions, such computer can be both a workstation or mobile device and a LAN server.

Anti-virus packages are installed on protected stations and get connected to Dr.Web Server. Each stations is included in one or several groups registered on this Server. Stations and Dr.Web Server communicate through the protocol used in the local network (TCP/IP of 4 or 6 version).

Installation

The anti-virus package can be installed on a workstation only locally. Local installation is performed directly on a user's computer. Installation may be implemented either by administrator or by user.

> Detailed description of anti-virus packages installation procedures on workstations you can find in the Dr.Web Enterprise Security Suite **Installation Manual**.

Management

When connection with Dr.Web Server is established, administrator is able to use the following functions implemented by anti-virus package on a station:

- Centralized configuration of anti-virus package on workstations via the Control Center.

  At this, administrator can either deny or grant user's permissions to change anti-virus package settings on stations on one's own.
- Configure the schedule for anti-virus scans and other tasks to execute on a station.
- Get scan statistics and other information on anti-virus components operation and on stations state.
- Start and stop anti-virus scans, etc. (depending on installed anti-virus package).

Update

Dr.Web Server downloads updates and distributes them to connected stations. Thus, optimal threats protection is implemented, maintained and adjusted automatically regardless of workstation users' computer skills.

In case an anti-virus station is disconnected from the anti-virus network, anti-virus package on station uses the local copy of the settings and the anti-virus protection on a workstation retains its functionality (up to the expiry of the user's license), but the software is not updated. If a station is allowed to use the *Mobile mode*, after connection with the Server is lost, the virus bases can be updated directly from the Dr.Web GUS.

> ⓘ The principle of stations operation in the Mobile mode is described in the Dr.Web Enterprise Security Suite **Administrator Manual**.

Chapter 3. Dr.Web for UNIX Mail Servers

This Manual describes management aspects of Dr.Web for UNIX Mail Servers anti-virus software designed for **GNU/Linux**, **FreeBSD**. The manual is designed for a person responsible for anti-virus protection and security ("Administrator" hereinafter).

Dr.Web for UNIX Mail Servers is an anti-virus solution designed to protect mail servers running under UNIX OSes (**GNU/Linux** and **FreeBSD**) from viruses and other types of malicious software, and to prevent distribution of the threats designed for all popular operating systems including mobile platforms.

Dr.Web for UNIX Mail Servers provides you with the following features:

1. **Detection and neutralization of threats.** Searches for malicious programs (for example, viruses, including those that infect mail files and boot records, Trojans, mail worms) and unwanted software (for example, adware, joke programs, dialers).

   Threat detection methods:

   - *Signature analysis*, which allows detection of known threats
   - *Heuristic analysis*, which allows detection of threats that are not present in virus databases
   - *Dr.Web Cloud* service that collects up-to-date information about recent threats and sends it to Dr.Web products.

   Note that the heuristic analyzer may raise false positive detections. Thus, objects that contain threats detected by the analyzer are considered “suspicious”. It is recommended that you choose to quarantine such files and send them for analysis to Doctor Web anti-virus laboratory.

   Scanning at user’s request can be performed in two modes: full scan (scan of all file system objects) and custom scan (scan of selected objects: directories or files that satisfy specified criteria). Moreover, the user can start a separate scan of volume boot records and executables that ran processes that are currently active. In the latter case, if a malicious executable is detected, it is neutralized and all processes run by this file are forced to terminate.
2. **Email message scanning.** The product supports the following modes of email message scanning:

   - *Mode of an external filter connected to the mail server (MTA)*. The product can be integrated into any mail server that supports interfaces for connection of external filters *Milter*, *Spamd* and *Rspamd*. In the filter mode, upon an initiative of MTA, all emails that arrive to the mail server are sent via the conjugation interface to Dr.Web for UNIX Mail Servers and scanned. Depending on the capability of the interface, Dr.Web for UNIX Mail Servers, that operates as a filter, can:

     - *Inform server of results of an email scanning*. In this case mail server must *independently* process an email message according to received results (reject the delivery, add headers or modify email contents, if scanning result contains information about presence of threats).
     - *Command the mail server to skip or reject an email message*.

- *Modify an email message* by adding the indicated headers or removing detected malicious or unwanted contents. Removed malicious contents are attached to the email message as an archive protected with a password. The recipient of the email message can request the password for unpacking the protected archive from the mail server administrator. If required, though not recommended, the administrator can configure the usage of the archives not protected with a password.

> Sending of commands to the mail server and return of the modified email message are supported only by the *Milter* interface. Interfaces *Spamd* and *Rspamd* do not allow Dr.Web for UNIX Mail Servers to send servers commands and return the modified email message. One of two verdicts will be returned to the server: "*email message is spam*" or "*email message is not spam*". In this case, for indirect modification of the rejected email message, you can use an action from the rules called `REJECT` <*description*>. Parameter <*description*>, if indicated, will be used as a header value `Message`', added by MTA to the email after the message about the scanning results.
>
> ---
>
> Function of scanning of email messages for the signs of spam could be unavailable depending on the distribution.

- *Invisible proxy mode for mail protocols*. In this mode, the product (using SpIDer Gate) implements the function of the proxy server embedded into the channel for sharing data between MTA and/or MUA transparently for the sharing parties and the function of the scanner of transmitted messages. The product can be transparently embedded into the main mail protocols: SMTP, POP3, IMAP. In this mode, and also depending on possibilities of the protocol it is embedded into, Dr.Web for UNIX Mail Servers can pass the email message to the recipient (it can be unmodified or have modifications in the form of added headers or repacked email message) or block its delivery, including the return of the correct protocol error to the sender or the recipient.

> Mode of the transparent proxy is available only for **GNU/Linux**.

Dr.Web for UNIX Mail Servers, depending on the distribution and settings, it executes the scanning of email messages:

- *Detection of malicious attachments* that contain threats;
- *Search for links to malicious websites* or websites from the unwanted categories;
- *Detection of signs of spam* (both using the automatically updated rule base of spam filtering and the mechanism of checking the presence of sender's address in the DNSxL black lists);
- *Compliance with the security criteria established by the administrator* of the mail system independently (scanning of a body and headers of messages using regular expressions).

To check links to unwanted websites, that can be present in email messages, the automatically updated databases of web resource categories is used. It is distributed along with Dr.Web for UNIX Mail Servers. Also, Dr.Web Cloud is requested to check the availability of information if the web source mentioned in the email message has been marked as malicious by other Dr.Web products.

3.1. Dr.Web for UNIX Mail Servers Components

For UNIX Internet gateways protection, the following anti-virus components are provided:

Dr.Web MailD

The component for scanning of emails. Analyzes the messages of email protocols, sorts out emails and prepares them for scanning for threats. It can operate in two modes:

- As a filter for mail servers(**Sendmail**, **Postfix**, etc.) connected via the interface *Milter*, *Spamd* or *Rspamd*.
- As a transparent proxy of mail protocols (SMTP, POP3, IMAP). In this mode, it uses SpIDer Gate.

Dr.Web Anti-Spam Engine

The component for checking of emails for spam. It is used by *Dr.Web MailD*, can be excluded from Dr.Web for UNIX Mail Servers on the station.

SpIDer Gate

The component which works in resident mode and monitors all network connections. Can be used by *Dr.Web MailD* in order to scan SMTP, POP3 and IMAP email connections.

Dr.Web Console Scanner (can be managed on station only)

Provides detection and neutralisation of viruses on the local machine. Managed via the console command line.

Dr.Web ClamD

Component emulating interface of the anti-virus daemon **clamd**, which is a component of **ClamAV**® anti-virus. Allows all applications that support **ClamAV**® to transparently use Dr.Web for UNIX Mail Servers for anti-virus scanning.

Quarantine

Isolates malicious and suspicious objects in the special folder.

Files on the workstation can be quarantined by Console Scanner only.

Description of how to manage Quarantine via the Control Center you can find in the **Administrator Manual**.

Auxiliary Components

Dr.Web Agent for UNIX

The component is used for interaction between Dr.Web for UNIX Mail Servers installed on the station and Dr.Web Enterprise Security Suite.

File Checker

The component is used by *Console Scanner* for checking files in *Scanning Engine* and for managing *Quarantine*.

Network Checker

The component is used to send data to the *Scanning Engine* for actual scanning. It is used by general components to check data transmitted over the network.

Scanning Engine

The component is used by *File Checker* and *Network Checker* for anti-virus scan and virus databases managing.

SNMP Agent

The component is designed for integration of Dr.Web for UNIX Mail Servers with external monitoring systems over the SNMP protocol.

Dr.Web ConfigD

The component that coordinates operation of all Dr.Web for UNIX Mail Servers components.

Dr.Web CloudD

The component that sends the following information to the *Dr.Web Cloud* service: visited URLs and information about the scanned files, to check them for threats not yet described in virus databases.

Dr.Web LookupD

Component retrieving data from external data sources (directory services, such as **Active Directory**) using LDAP protocol. The data are used in rules of traffic monitoring.

Dr.Web HTTPD

Web server for managing Dr.Web for UNIX Mail Servers components. It provides the management web interface for product installed on the station.

3.2. Dr.Web for UNIX Mail Servers Configuration

To view or edit the configuration of the anti-virus components on the workstation:

1. Select the **Anti-virus network** item in the main menu of the Control Center.

2. In the hierarchical list of the opened window, click the name of a station under required OS (**Linux**, **Solaris** or **FreeBSD**) or a group containing such stations.
3. In the **Configuration** section of the opened control menu, in the **UNIX** subsection, select the necessary component.
4. A window with the component settings will be opened.

   Managing settings of anti-virus components via the Control Center differs from managing settings directly via the corresponding components on station:

   - to manage separate parameters, use the options located on the right from corresponding settings:

     Reset to initial value—restore the value that parameter had before editing (last saved value).

     Reset to default value—set the default value for a parameter.

   - to manage a set of parameters, use the options located on the toolbar:

     Reset all parameters to initial values—restore the values that all parameters in this section had before current editing (last saved values).

     Reset all parameters to default values—restore default values of all parameters in this section.

     Propagate these settings to another object—copy settings from this section to settings of other station, group or several groups and stations.

     Set inheritance of settings from primary group—remove personal settings of a station and set inheritance of settings in this section from a primary group.

     Copy settings from primary group and set them as a personal—copy settings of this section from a primary group and set them for selected stations. Inheritance is not set and stations settings considered as a personal.

     Export settings from this section to the file—save all settings from this section to a file of a special format.

     Import settings to this section from the file—replace all settings in this section with settings from the file of a special format.

5. After settings changes were made via the Control Center, click **Save** to accept the changes. The settings will be passed to the stations. If the stations were offline when changes are made, the settings will be passed when stations connect to the Server.

   Administrator may forbid editing settings on station for a user (see the **Permissions of Station Users** section in the **Administrator Manual**). At this, only administrator will be ale to edit settings via the Control Center.

3.2.1. Dr.Web MailD Settings

The **Dr.Web MailD** page contains the following parameters of Dr.Web for UNIX Mail Servers operation:

- General—general parameters of the component.
- Notification templates—parameters of email notification templates.
- Milter Connections—parameters of connection to email servers (MTA) via *Milter* interface.
- Spamd Connections—parameters of connection to email servers (MTA) via *Spamd* interface.
- Rspamd Connections—parameters of connection to email servers (MTA) via *Rspamd* interface.

3.2.1.1. General

On this page you can manage the following options of Dr.Web MailD on the protected workstation (mail server):

- **Log level**—defines the log verbosity level that is used for Dr.Web MailD messages logging.
- **Logging method**—defines the logging method for Dr.Web MailD. The following values are allowed:
  - *Auto*—use the logging method which is defined in Dr.Web ConfigD settings for all components of the solution.
  - *Syslog*—use the **syslog** system service for Dr.Web MailD messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from Dr.Web MailD.
  - *Path*—use the specified file to store Dr.Web MailD log messages. If you specify this method, you must also specify a path to the file in the **Log file** field.
- **User**—determines under which user name the component should be run (the component takes rights and privileges of specified user).

  > ⚠ When a user name is not specified, the component operation terminates with an error after the starting up.

- **Configuration file of DNS resolver**—defines the path to the configuration file of a subsystem for domain names resolving (DNS resolver).
- **Path to the socket file of the fixed copy of the component**—defines the path to the special UNIX socket that is used by separate component instance. This instance is running permanently, if the socket is specified, and can be used by external programs. If the path is empty, the separated Dr.Web MailD instance is not running and is not available for external programs. The standard instance of Dr.Web MailD running and terminating automatically, when it necessary.

3.2.1.2. Notification templates

On this page you can manage options of generation by Dr.Web MailD on the protected mail server email notifications about detected threats :

- **Templates directory**—specifies the path to the directory containing the templates for emails returned to the user in case of email blocking..
- **Contacts of the mail system administrator**—allows to specify contacts of Dr.Web for UNIX Mail Servers Administrator for the insertion in the messages about threats (used in message templates). This information will allow to users to contact with administrator when they will receive a message with notification about a threat.
- **Report languages**—specifies the languages used for generation of service mail messages (for example, mail messages returned to the sender in case of email blocking).
- **Password generation mode**—specifies the method for generation of a password for archives with malicious objects placed in messages and sent to recipients. The following methods are allowed:
  - *None*—archives will not be protected with password (not recommended).
  - *Plain*—all archives will be protected with the same password. In this case, the password to be used is specified in the **Password** field.
  - *HMAC*—the unique password will be generated for each archive based on the message identifier and the secret word.In this case, the secret word to be used is specified in the **Secret word** field.

3.2.1.3. Milter Connections

On this page you can specify parameters of connection Dr.Web MailD on the protected mail server to email servers (MTA) via *Milter* interface:

- **Use heuristic analysis**—instructs Dr.Web MailD to use the heuristic analysis on the protected mail server to detect unknown threats in email messages received from MTA via the *Milter*. Note that heuristic analysis may slow down the checking but improves its reliability.
- **Timeout to scan one message**—restrictions on maximal time spent on scanning of one email message by Dr.Web MailD (for email messages received from MTA via the *Milter*). If the value is *0*, scan time is not limited.
- **Maximum nesting level.** In this section you can specify settings which Dr.Web MailD is used for checking compound files (containers) of the following types: archives, mail files (email messages, mailboxes), packed objects and other containers (i.e. compound files that are not classified as archives, mail files or packed objects) that attached to email messages received from MTA via the *Milter*.

  For each of the types you can specify in the corresponding field the maximum nesting level. The objects that are nested into container deeper than the specified level are skipped during scanning the container by Dr.Web MailD. For example, if you want scan the contents of the archives which are nested into archives, specify the maximum nesting level not less than *2*. To disable scanning of nested objects, specify *0* as the maximum nesting level for the corresponding type of containers.

Note that increasing of maximum nesting level slows down the file system monitoring.

The **Maximum compression ratio** field allows you to specify maximum compression ratio (as a ratio of size of compressed file to its original) for compressed files. If compression ratio of a file is greater than the maximum allowed, the file is skipped during the check.

- **Socket for connections of MTA via Milter**—defines the socket for integration Dr.Web MailD with MTA as a Milter filter of mail (MTA will connect to this socket when using Dr.Web MailD as the corresponding filter). You can specify here the UNIX socket (as a path) or the network socket (as an *IP address:port* pair).
- **Block unchecked email messages**—set the checkbox to block transmission of an email message received for scanning via the *Milter*, if its contents could not be scanned.
- **Email processing rules for Milter**—the section defines rules that are used by Dr.Web MailD to all email messages received from MTA for checking via the *Milter* interface.

  To add new rule in the list, click + near the corresponding list item. To delete some rule from the list, click − near the corresponding list item.

  For more information about the rules see Appendix A. Traffic Checking Rules.

3.2.1.4. Spamd Connections

On this page you can specify parameters of connection Dr.Web MailD on the protected mail server to email servers (MTA) via *Spamd* interface.

All the parameters presented here, are similar to corresponding parameters on the Milter Connections page.

3.2.1.5. Rspamd Connections

On this page you can specify parameters of connection Dr.Web MailD on the protected mail server to email servers (MTA) via *Rspamd* interface.

All the parameters presented here, are similar to corresponding parameters on the Milter Connections page.

3.2.2. SpIDer Gate Settings

The **SpIDer Gate** section consists of the following sections, containing the corresponding parameters of Dr.Web for UNIX Mail Servers operation:

- General—general SpIDer Gate settings
- Actions—actions on detection of threats by SpIDer Gate
- Web filtering—settings of web traffic check and control of access to Internet resources by SpIDer Gate

- Containers—settings of scanning of compound files (archives, email files, etc.)
- Additional—additional SpIDer Gate settings.

3.2.2.1. General

On this page you can manage the following parameters of SpIDer Gate on the protected station:

- **Enable SpIDer Gate**—enables or disables SpIDer Gate on the protected station.
- **Use heuristic analysis**—instructs SpIDer Gate to use the heuristic analysis on the protected station to detect unknown threats. Note that heuristic analysis may slow down the file system monitoring but improves its reliability.
- **Scanning time of one element**—restrictions on maximal time spent on scanning of one file by SpIDer Gate on the station. If the value is *0*, scan time is not limited.

3.2.2.2. Actions

On this page you can manage the following parameters of SpIDer Gate on the protected station:

- Set the **Scan received files** checkbox to enable check of incoming (downloaded from Internet) files.
- In the **Block files** and **Additionally block** sections, select types of incoming malicious objects which will be blocked by SpIDer Gate (**Infected**, **Suspicious**, etc.).

3.2.2.3. Web filtering

On this page you can manage the following parameters of SpIDer Gate on the protected station:

- Set the **Scan URL** flag to enable check of Internet resources by categories.
- Set the **Block non-recommended websites** flag to deny access to the websites that use social engineering techniques to misguide users.
- Set the **Block URLs listed due to a notice from copyright owner** flag to deny access to the websites due to a notice from copyright owner who has found out the violation of rights to the intellectual property in the Internet.
- In the **Block websites from the following categories** section select the categories of websites (**Adult content**, **Violence**, etc.) you need to block access to.
- In the **White list**/**Black list** sections add the paths to the websites you need to allow/restrict access to:
  - To add a certain website, enter its full domain address (for example, `www.example.com`). Access to all web pages located on this domain will be defined by this string.
  - To configure access to websites with similar names, enter the common part of their domain names. For example, if you enter `example`, the access to the `example.com`, `ex-`

`ample.test.com`, `test.com/example`, `test.example222.com` and other similar websites will be defined by this string.

- To configure access to websites within a particular domain, enter the domain name with a period "`.`". In this case, the access to all web pages located on this domain will be defined by this string. If specifying domain name, you use a forward slash "`/`", the substring before the "`/`" is considered a domain name, while the substring after the slash is considered a part of address for the websites that you want to access within this domain. For example, if you enter `example.com/test`, SpIDer Gate will configure access to web pages such as `example.com/test11`, `template.example.com/test22`, etc.
-

3.2.2.4. Containers

On this page you can specify settings which SpIDer Gate is used for checking compound files (containers) of the following types: archives, mail files (email messages, mailboxes), packed objects and other containers (i.e. compound files that are not classified as archives, mail files or packed objects).

For each of the types you can specify in the corresponding field the maximum nesting level. The objects that are nested into container deeper than the specified level are skipped during scanning the container by SpIDer Gate. For example, if you want scan the contents of the archives which are nested into archives, specify the maximum nesting level not less than *2*. To disable scanning of nested objects, specify *0* as the maximum nesting level for the corresponding type of containers.

Note that increasing of maximum nesting level slows down the file system monitoring.

The **Maximum compression ratio** field allows you to specify maximum compression ratio (as a ratio of size of compressed file to its original) for compressed files. If compression ratio of a file is greater than the maximum allowed, the file is skipped during the check.

3.2.2.5. Additional

On this page you can specify some advanced SpIDer Gate settings on the protected station.

The following advanced SpIDer Gate settings are available:

- **Log level**—defines the log verbosity level that is used for SpIDer Gate messages logging.
- **Logging method**—defines the logging method for SpIDer Gate. The following values are allowed:
  - *Auto*—use the logging method which is defined in Dr.Web ConfigD settings for all components of the solution.
  - *Syslog*—use the **syslog** system service for SpIDer Gate messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from SpIDer Gate.

  - *Path*—use the specified file to store SpIDer Gate log messages. If you select this method, you must also specify a path to the file in the **Log file** field.

3.2.3. Dr.Web Agent for UNIX Settings

The **Dr.Web Agent** page consists of the following sections, containing the corresponding parameters of Dr.Web for UNIX Mail Servers operation:

- General – Dr.Web Agent for UNIX settings.
- Configuration – editor of settings for all the Dr.Web for UNIX Mail Servers components.

3.2.3.1. General

On this page you can manage the following parameters of Dr.Web Agent for UNIX on the protected station:

- **Statistics sending period**—defines the time period of sending general statistics from Dr.Web Agent for UNIX to the server.
- **Mobile mode for updates**—allows the workstation to receive updates from GUS if the server is not available. The following values allowed:
  - *Auto*—instructs to use mobile mode, if allowed by the server, and perform updates both from GUS and from central protection server, depending on which connection is available and which connection quality is higher.
  - *Enable*—instructs to use mobile mode if it is allowed by the server (that is, perform updates from GUS using the updating component installed on the station).
  - *Disable*—instructs not to use mobile mode (updates are always received from the server).
- **Process the discovery requests**—set the flag, to allow Dr.Web Agent for UNIX to receive discovery requests from the server (discovery requests are used by the server to check the structure and state of the anti-virus network).
- **Log level**—defines the log verbosity level that is used for Dr.Web Agent for UNIX messages logging.
- **Logging method**—defines the logging method for Dr.Web Agent for UNIX. The following values are allowed:
  - *Auto*—use the logging method which is defined in Dr.Web ConfigD settings for all components of the solution.
  - *Syslog*—use the **syslog** system service for Dr.Web Agent for UNIX messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from Dr.Web Agent for UNIX.
  - *Path*—use the specified file to store Dr.Web Agent for UNIX log messages. If you select this method, you must also specify a path to the file in the **Log file** field.

⚠ Usually, for the parameters of this component the optimal values are specified. Thus, it is not recommended to change them, if it is not necessary.

3.2.3.2. Configuration

On this page you can specify settings for any of the Dr.Web for UNIX Mail Servers components installed on the station (an `.ini` configuration file format is used).

To specify settings, make the corresponding changes in the **Configuration file drweb.ini** field (*ini editor*).

Please note that:

- The *ini editor* shows only the configuration parameters having the values that have been changed on this page.
- Values of the configuration parameters that specified in the editor take precedence over values specified by the component setting pages: if on a settings page is one value of some parameter is specified and the other value for this parameter is specified in the *ini editor* on the **Configuration** page, the value that is specified on the **Configuration** page, will be used on the station. Moreover, if a section of some component is specified in the *ini editor*, for all parameters of the component that are not defined in the section, the default values are applied on the station.
- The context hints are supported by the *ini editor*: to show hint containing list of available parameters (or configuration section names, depending on the context), press CTRL+SPACE.
- You can export contents of the *ini editor* to `.ini` configuration file and import the contents from `.ini` configuration file. To do that click the corresponding icon at the top part of the page (above the *ini editor*).

For a complete list of components on the station that are available for configuration, and for a description of their parameters in the `drweb.ini` configuration file, refer to User manual or Administrator manual of the product installed on the station.

3.2.4. File Checker Settings

On this page you can manage parameters which are used by File Checker auxiliary component on the protected station.

The following parameters are available:

- **Maximum checked file cache size**—defines size of the cache that is used by File Checker for temporarily storing the results of files scan.
- **Cache validity period**—defines the duration of a time period when File Checker does not rescan the file, if its scan result is available in the cache.

- **Log level**—defines the log verbosity level that is used for File Checker messages logging.
- **Logging method**—defines the logging method for File Checker. The following values are allowed:
  - *Auto*—use the logging method which is defined in Dr.Web ConfigD settings for all components of the solution.
  - *Syslog*—use the **syslog** system service for File Checker messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from File Checker.
  - *Path*—use the specified file to store File Checker log messages. If you specify this method, you must also specify a path to the file in the **Log file** field.

Also, you can choose which additional data will be saved to the log on the *Debug* verbosity level.

- **IPC subsystem**—save IPC messages on component interaction
- **File scanning**—save file scan results
- **SpIDer Guard file monitoring**—save SpIDer Guard scan requests
- **Checked file cache status**—save the cache state changes.

> ⚠ Usually, for the parameters of this component the optimal values are specified. Thus, it is not recommended to change them, if it is not necessary.

3.2.5. Scanning Engine Settings

On this page you can manage parameters which are used by Scanning Engine auxiliary component on the protected station.

The following parameters are available:

- **Path to the socket file of the fixed copy of the component**—path to the special UNIX socket that is used by separate Scanning Engine instance. This instance is running permanently, if the socket is specified, and can be used by external programs for file scan via this socket. If the path is empty, the separated Scanning Engine instance is not running and is not available for external programs. The standard Scanning Engine instance running and terminating automatically, when it necessary for file scanning.
- **Number of scanning processes**—defines the maximum allowed number of child scanning processes that can be running by Scanning Engine during the scanning of files. If you want to change this value, evaluate the number of CPU cores available on the station.
- **Watchdog timer**—defines the duration of a time period which is used by Scanning Engine for automatic detection and termination termination the suspended scanning processes ("watchdog" timer).
- **Log level**—defines the log verbosity level that is used for Scanning Engine messages logging.

- **Logging method**—defines the logging method for Scanning Engine. The following values are allowed:
  - *Auto*—use the logging method which is defined in Dr.Web ConfigD settings for all components of the solution.
  - *Syslog*—use the **syslog** system service for Scanning Engine messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from Scanning Engine.
  - *Path*—use the specified file to store Scanning Engine log messages. If you specify this method, you must also specify a path to the file in the **Log file** field.

> ⚠ Usually, for the parameters of this component the optimal values are specified. Thus, it is not recommended to change them, if it is not necessary.

3.2.6. Dr.Web ConfigD Settings

On this page you can manage parameters which are used by Dr.Web ConfigD auxiliary component on the protected station.

The following parameters are available:

- **Public communication socket path**—path to internal UNIX socket that is used for interaction with Dr.Web ConfigD by Dr.Web for UNIX Mail Servers components.
- **Administrative communication socket path**—path to internal UNIX socket that is used for interaction with Dr.Web ConfigD by Dr.Web for UNIX Mail Servers components operating with superuser privileges.
- **Temporary files directory**—path to the directory with temporary files saved by Dr.Web for UNIX Mail Servers components.
- **Path to the directory with PID files and communication sockets**—path to the directory with PID files and UNIX sockets that used for Dr.Web for UNIX Mail Servers components interaction.
- **Log level**—defines the log verbosity level that is used for Dr.Web ConfigD messages logging.
- **Logging method**—defines the logging method for Dr.Web ConfigD. The following values are allowed:
  - *Syslog*—use the **syslog** system service for Dr.Web ConfigD messages logging. If you specify this method, you must also specify the value of **Syslog facility** parameter. It defines the label (or subsystem) which is used by **syslog** to save messages from Dr.Web ConfigD.
  - *Path*—use the specified file to store Dr.Web ConfigD log messages. If you specify this method, you must also specify a path to the file in the **Log file** field.

> ⚠ Usually, for the parameters of this component the optimal values are specified. Thus, it is not recommended to change them, if it is not necessary.

Appendix A. Traffic Checking Rules

The rules are represented by production rules such as IF <*condition*> THEN <*action*>. At that, in the part <*condition*> the following scanning types are specified: "*The variable value is (not) set*" or "*The variable value is (not) included in the specified set*". The part <*action*> contains *ultimate resolution* (skip or block traffic), or an action such as "*Assign a value to the variable*" or "*Add specified value to the set of variable values*".

The <*action*> part of the rule is executed, only if the <*condition*> part evaluates to true. If the <*condition*> part evaluates to false, the action is not performed, and the program jumps to the next rule. The rules are considered downwardly until any of the ultimate resolutions is performed. After this, all lower rules are ignored.

Rule Format

Format of the rule production

```
[<condition>[, <condition>[, ...]]] : <action>
```

The conditional part of the rule (before `':'`) can be missing, in this case the <*action*> part is executed without any condition. If the conditional part of the rule is missing, the `':'` separator can be omitted. The comma between conditions in the conditional part performs a role of a logical conjunction (that is, "and"), and the conditional part elevates to true, only if all its conditions are true. In the rules the register is not important for the key words, names of variables and configuration parameters.

Conditions

The following types of conditions can be use in the conditional part of the rules:

Condition	Meaning of the Condition
<*variable*> <*value* >	The value of the specified variable coincides with the set value. *Can be used only for those variables that can contain a set of values simultaneously.*
<*variable*> `[not] in` <*set of values*>	The value of the specified variable is contained in the specified set of values (*for not—does not match any value from the specified set*).
<*variable*> `[not] match` <*set of values*>	The value of the specified variable matches any regular expression listed in the specified set (*for not—does not match any expression from the specified set*).

Condition	Meaning of the Condition
	Regular expressions are specified using either the *POSIX* syntax (*BRE*, *ERE*) or the *Perl* syntax (*PCRE*, *PCRE2*).
<*variable*> `[not] gt` <*value*>	The value of the specified variable is (not) greater than the set value. *Can be used only for those variables that can have a single value.*
<*variable*> `[not] lt` <*value*>	The value of the specified variable is (not) less than the set value. *Can be used only for those variables that can have a single value.*

*) An optional key word `not` means negation.

Part <*set of values*> to which a variable is compared can be specified in the following ways:

Syntax	Meaning
`(`<*value 1*>`[,` <*value 2*>`[, ...]])`	In the parentheses you directly list the set of values to check against (not less then one value). In case there is only one value and the `in` condition is used, you can omit the parentheses (and you will end up with a case <*variable*> <*value*>).
`"`<*section*>`.`<*parameter*>`"`	The set of values currently assigned to a certain configuration parameter; where between the quotation marks you should specify the name of a configuration parameter whose value (or set of values) must be checked (note that you also need to specify the name of the section to which the parameter belongs). The lists of the parameters that can be used as conditions depend on the component for which the rules are set. The lists are provided below.
`file("`<*file name*>`")`	List of values is read from the text file <*file name*> (one file string—one list element, leading and trailing spaces in strings are ignored). A path to the file must be absolute. If a <*file name*> contains quotes and apostrophes, they must be escaped: '\'.

Syntax	Meaning
	! The file size must not exceed 64 MB. The file contents are read and inserted into the rules once, during the Dr.Web for UNIX Mail Servers starting up. If there is no file or the file size is exceeded, an error x102 appears during the starting. In case the file contents are changed during the process, in order to apply all changes, you should reboot your computer after the changes are saved using the. A set of values from the file is not available for all variables. Whether you can use a variable to scan its value by using a set of values from the file is indicated below.
<type_of_LOOKUP_request>`@`*<tag>*`[@`*<value>*`]`	A set of values is requested via Dr.Web LookupD from an external data source (LDAP, ActiveDirectory), where *<LOOKUP_request_type>* is the type of the data source used (LDAP or AD); *<tag>* is a section name describing the connection that is used to sample the data, and *<value>* (optional) is a value that must be contained in the set of values retrieved from the data source. ! Values from Dr.Web LookupD are not available for all variables. Also, the condition *<scanning>* cannot be applied to all variables. Whether you can use a variable to scan its value by using Dr.Web LookupD is indicated below.

`dnsxl(`*<DNSxL server>*`[: [mask]` *<IP>*`]`
`[, ...])`
	In the parentheses you list the DNSxL-servers (DNSBL, etc.) that must check the inclusion of an IP address or FQDN (resolved to IP addresses in advance) in their lists of IP addresses.

Syntax	Meaning
	If the checked IP address is registered in lists of one of the DNSxL servers listed in the parentheses, the response of this server contains one or more DNS logs of type A, and a fictitious IP address returned by the server could contain a reason, why the checked IP address was included in the lists of the server (generally, a type of the reason is defined by the value of the last octet of the returned fictitious IP address). For each DNSxL server in the list, it is possible to assign check of the expected returned value of the fictitious IP address. Check is indicated after the colon in the following way: • *<DNSxL server>: <IP address>* • *<DNSxL server>: mask <IP address>* In the first case, the indicated requirement states that the fictitious IP address returned by the server *<DNSxL server>* must exactly match the indicated address *<IP address>*. In the second case, the indicated requirement states that the fictitious IP address returned by the server *<DNSxL server>* must be equal to the indicated mask in its nonnull octets. If the check parameters are not indicated, the condition works if *<DNSxL server>* returns any fictitious IP address as a response to the request. **Examples:** *<IP>* `in dnsxl("dnsxl.server.org")` – for IP address in the variable *<IP>*, the server must return any fictitious IP address; *<IP>* `in dnsxl("dnsxl.server.org": 127.0.0.2)`—for IP address in the variable *<IP>*, the server must return fictitious IP address `127.0.0.2`; *<IP>* `in dnsxl("dnsxl.server1.org": mask 0.0.0.8, "dnsxl.server2.org": 127.0.0.3, "dnsxl.server3.org")`—for IP address contained in the variable *<IP>*, or the first server will return the fictitious IP address with the low octet `8`, or the second—fictitious IP address `127.0.0.3`, or the third—any fictitious IP address.

Syntax	Meaning
	Use of the check instruction <*variable*> `in dnsxl(`<*server list*>`)` is allowed only if <*variable*> is an IP address or a domain name that could be resolved by the DNS service to IP address (FQDN). Thus, only the following variables could be used as a variable for this condition: `src_ip`, `url_host` (see further).

Actions

The actions can be divided into *ultimate* resolutions, defining whether the traffic is blocked or allowed; *modifying* resolutions that do not interrupt the scanning but fix the action that should be applied to a connection or to a scanned object after reaching the permissive resolution that allows the traffic, and *actions that change the value of a variable*, which can be used to check the downward conditions.

Ultimate Resolutions

Resolution	Description (Meaning)
Common Resolutions	
`PASS`	Skip traffic (allow creating connection). The downward rules (if there are any) are ignored. For the rules of mail processing, there is merit in a command that allows a message to be transmitted to a recipient after all collected changes have been applied to it (i. e. all executed actions`REPACK`, `ADD_HEADER`, `CHANGE_HEADER`, see below).
`BLOCK as` <*reason*>	Block traffic (block creating connection). The downwards rules (if there are any) are ignored. A blocking <*reason*> is recorded in the log. The same reason is used to define a browser notification to be shown to a user. Two standard reasons can be used as <*reason*> for `BLOCK`: • `BlackList`—the data is blocked because it is included in user's black list. • `_match`—the block happens because a web resource or file containing threat belongs to a category that triggers rule executing (for conditions `*_category in (...)`). The `_match` variable con-

Resolution	Description (Meaning)
	tains the list of blocked categories for which the correspondence has been executed. *For the rules of mail processing, this action is synonymous to the action* `REJECT`. *The reason for blocking is <reason>, and it is ignored.*
Special resolutions for rules of mail processing	
`REJECT ["`*<description>*`"]`	Discard an email (prevent its receiving or sending). While working with data Transferred via SMTP protocol, form response code SMTP 541 (class of permanent errors). If an optional parameter *<description>* is indicated, it will be used as a response. When scanning an email message received from MTA via the *Spamd/Rspamd* interface, *<description>* will be used as the value of the header “`Message`”, which is added to the email after the message with scanning results.
`TEMPFAIL ["`*<description>*`"]`	Send to the sender as a “temporary error”. While working with data Transferred via SMTP protocol, form response code SMTP 451 (class of temporary errors). If an optional parameter *<description>* is indicated, it will be used as a response. When scanning an email message received from MTA via the *Spamd/Rspamd* interface, *<description>* will be used as the value of the header “`Message`”, which is added to the email after the message with scanning results.
`DISCARD`	Reject an email message, i.e. accept it without return of the error code to the sender, but delete it instead of sending to the recipient.

Modifying Resolutions

Modifying resolutions do not interrupt the scanning of rules but fix the actions that should be applied to the scanned data after reaching the permissive resolution PASS.

Resolution	Description (Meaning)
`REPACK [`*<reason>*`]`	Repack the email message, i.e. create (on the basis of one of the predetermined templates) a new email message that contains the contents of the old one and some text with information to the recipient on threats. The removed unwanted contents are placed in the archive protected with password. This archive will be added to the email message sent to the recipient as an attachment. Proceed with the scanning of the email message until the resolution *PASS*. There are the following predetermined templates for repacking: 1. *The email message is spam*;

Resolution	Description (Meaning)
	2. *One or more threats in the email message*; 3. *One or more malicious/unwanted URLs in the email message*; 4. *Violation of the security policy established by the administrator*. A REJECT <*reason*> is recorded in the log. The same reason is used to define which one of four templates was used to generate a notification email to the recipient. As a <*reason*> for `REPACK`, the following reasons could be used: • `as _match`—the repacking happens if an email message is considered to be spam or if it contains a web source or file with a threat that belong to a category that triggers the rule (for conditions `*_category in (...)`). The `_match` variable contains the list of unwanted categories for which the correspondence has been executed. For repacking, template 1, 2 or 3 (see above) is chosen depending on the contents detected in the email message. ▫ if the email message is spam, template 1 is chosen; ▫ if at least one threat is detected, template 2 is chosen; ▫ if at least one malicious/unwanted URL is detected, template 3 is chosen; ▫ if threats are not detected, template 4 is chosen. • "*text message*"—email message was repacked due to triggering of settings of an administrator, and the message indicates an arbitrary message from the administrator. For example: `REPACK "Virus found!"`. For repacking, template 4 will be chosen.
ADD_HEADER("<*Name*>", "<*Value*>")	Add the header <*Name*> to the email message with the value <*Value*> and continue scanning until the resolution *PASS*.For example: `ADD_HEADER ("X-SPAM", "Virus found!")`. The value will be recoded to ASCII according to RFC 2047.
CHANGE_HEADER("<*Name*>", "<*Value*>" | _value [+ "<*Value*>" | _value [+ ...]])	Replace the value of the first found header with the name <*Name*>. New value—concatenation of values after the comma separated by the "+" symbol. Each value could be either a string literal in quotation marks, or a special variable `_value`

Resolution	Description (Meaning)
	that is replaced with the initial value of the modified header. Continue scanning of the email message until the *PASS* resolution .

Aspects of Resolution Processing:

- `BLOCK as BlackList`, always processes as *"is included in a black list"* (without considering the condition specified in the rules with this resolution).
- `BLOCK as _match`, if `_match` is not empty, processes as *"belongs to the _match category"*.
- `BLOCK as _match`, if `_match` is empty, processes as *"is included in a black list"* (without considering the condition specified in the rules with this resolution).
- If all rules have been considered, and none of the rules with resolutions performs (or the rules do not have resolutions), this situation is the same as `PASS` action.

Changing Value of a Variable

To change the variable value, the following instruction is used:

```
SET<variable> = ([<value 1>[, <value 2>[, ...]]])
```

If nothing is enclosed in brackets, the list of variable values is cleared. If there is only one value, the brackets should be omitted, that is, the following syntaxes should be used:

```
SET <variable> = <value >
```

Variables used in the rules

When indicating variables in the rules, the register of symbols is not considered. The variables with compound names could be saved using underscore for spacing or without it. Thus, records `variable_name`, `VariableName` and `variablename` represent the same variable. In this section, all variables are saved using underscore (i.e. `variable_name`).

General purpose variables

Variable	Description	Can be used in	
		conditional part	action part (SET)
`protocol`	Network protocol type, used by the connection. *The variable can simultaneously contain a set of values.*	Yes	No

Variable	Description	Can be used in	
		conditional part	action part (SET)

	Allowed values: `HTTP`, `SMTP`, `IMAP`, `POP3`.
	Usage Aspects:
	• The variable value can be defined only if SSL/TLS is not used or it was allowed to unwrap SSL.
	• A set of values for checking a variable value is available from the file.
	Examples:
	`protocol in (HTTP, SMTP)`
	`protocol in (POP3)`
	`protocol in file("/etc/file")`		

`sni_host`	SNI host (address), with which the connection is established via SSL/TLS.
	Usage Aspects:
	• If SSL is not used, the value of a variable is not defined, the condition evaluates to false.
	• A set of values for checking a variable value is available from the file.
	Examples:
	`sni_host not in ('vk.com', 'ya.ru')`
	`sni_host in "LinuxFirewall.BlackList"`
	`sni_host in file("/etc/file")`
		Yes	No

`sni_category`	The list of categories (*AdultContent*, etc.) which the host (that is identified from the SNI-header) belongs to (according to the databases of web resource categories), for hosts to which your computer is attempting to connect over SSL/TLS. *The variable can simultaneously contain a set of values.*	Yes	No

Variable	Description	Can be used in	
		conditional part	action part (SET)

	Usage Aspects:
	• If SSL is not used, the value of a variable is not defined, the condition evaluates to false.
	• For rules used by Dr.Web MailD, condition with `not in` will be *true*, even if according to the scanning results, the host does not belong to any of the predetermined categories ("safe" host).
	• If databases of web resource categories are not installed, the variable could not be used in rules (attempts to check if a condition in the rule is true will lead to the error x112).
	• A set of values for checking a variable value is available from the file.
	Examples:
	`sni_category not in (AdultContent, Chats)`
	`sni_category in "LinuxFirewall.BlockCategory"`
	`sni_category in (FreeEmail)`
	`sni_category not in file("/etc/file")`		

`url_host`	URL/host with which the connection is established.
	Usage Aspects:
	• The variable value can be defined only if SSL/TLS is not used or it was allowed to unwrap SSL.
	• Dr.Web LookupD can be used to check the value of this variable.
	• This variable could be checked for inclusion in black lists of DNSxL (DNSBL, etc.).
	• A set of values for checking a variable value is available from the file.
	Examples:
	`url_host in ('vk.com', 'ya.ru')`
	`url_host not in "ICAPD.Whitelist"`
		Yes	No

<table>
<tr><th rowspan="2">Variable</th><th rowspan="2">Description</th><th colspan="2">Can be used in</th></tr>
<tr><th>conditional part</th><th>action part (SET)</th></tr>
<tr><td></td><td><code>url_host in LDAP@hosts</code>
<code>url_host not in file("/etc/file")</code>
<code>url_host not in dnsxl("multi.surbl.org": 127.0.0.2, "multi2.surbl.org")</code></td><td></td><td></td></tr>
<tr><td><code>url_category</code></td><td>The list of categories to which the URL/host belongs. The information is based according to the database of categories or Dr.Web Cloud replies.
<i>The variable can simultaneously contain a set of values.</i>
<b>Usage Aspects:</b>
• The variable value can be defined only if SSL/TLS is not used or it was allowed to unwrap SSL.
• For rules used by Dr.Web MailD, condition with <code>not in</code> will be <i>true</i>, even if according to the scanning results, URL/host does not belong to any of the predetermined categories (“safe” URL/host).
• If databases of web resource categories are not installed, the variable could not be used in rules (attempts to check if a condition in the rule is true will lead to the error x112).
• A set of values for checking a variable value is available from the file.
<b>Examples:</b>
<code>url_category not in (AdultContent, Chats)</code>
<code>url_category in "LinuxFirewall.BlockCategory"</code>
<code>url_category in (FreeEmail)</code>
<code>url_category in file("/etc/file")</code></td><td>Yes</td><td>No</td></tr>
<tr><td><code>threat_category</code></td><td>The list of categories to which the threat belongs, which is found in the transferred data (according to information from virus databases).</td><td>Yes</td><td>No</td></tr>
</table>

Variable	Description	Can be used in	
		conditional part	action part (SET)

	The variable can simultaneously contain a set of values.
	Usage Aspects:
	• The variable value can be defined only if SSL/TLS is not used or it was allowed to unwrap SSL.
	• For rules used by Dr.Web MailD, condition with `not in` will be *true*, even if according to the scanning results, the object does not contain threats from any of the predetermined categories (”safe” object).
	• A set of values for checking a variable value is available from the file.
	Examples:
	`threat_category in "LinuxFirewall.BlockThreat"`
	`threat_category not in (Joke)`
	`threat_category in file("/etc/file")`		

`user`	The name of the user with whose privileges the process that is sending (or receiving) the traffic has been launched. **Usage Aspects:** • Dr.Web LookupD can be used to check the value of this variable. • If you need to find out whether the user belongs to a certain user group, use an LDAP or an Active Directory data source that returns a list of groups and specify the name of the required group (for which you want to know whether the user is its member or not). Use the following format: <*type of the source for LookupD*>@<*source of groups*>@<*required group*>. Requests to Active Directory (AD@) return only lists of groups, therefore for these requests it is mandatory to use the @<*required group*> part. • A set of values for checking a variable value is available from the file.	Yes	No

Variable	Description	Can be used in	
		conditional part	**action part (SET)**

	Examples:
	`user in ('user1', 'user2')`
	`user in AD@Winusergroups@Admins`
	`user in LDAP@AllowedUsers`
	`user not in file("/etc/file")`		

`src_ip`	The IP address of a host establishing the connection.
	Usage Aspects:
	• Dr.Web LookupD can be used to check the value of this variable.
	• This variable cannot be used in rules of Dr.Web MailD for the interface *Spamd*: this protocol does not provide information about the email message sender.
	• This variable could be checked for inclusion in black lists of DNSxL (DNSBL, etc.).
	• A set of values for checking a variable value is available from the file.
	Examples:
	`src_ip not in (127.0.0.1, 10.20.30.41, 198.126.10.0/24)`
	`src_ip in LDAP@AllowedAddresses`
	`src_ip not in file("/etc/file")`
	`src_ip in dnsxl("zen.spamhouse.org": mask 0.0.0.2, "zen2.spamhouse.org")`
		Yes	No

`proc`	The process establishing the connection (the full path to the executable file).
	Usage Aspects:
	• A set of values for checking a variable value is available from the file.
	Examples:
	`proc in ('/usr/bin/ls')`
	`proc not in ('/home/user/myapp', '/bin/bin1')`
		Yes	No

Variable	Description	Can be used in	
		conditional part	**action part (SET)**

	`proc in "LinuxFirewall.ExcludedProc"`
	`proc in file("/etc/file")`		

`direction`	The type of traffic on the connection.
	Allowed values: `request` (client request), `response` (server reply).
	This variable cannot simultaneously contain a set of values; conditions of the `match` *and* `in` *type cannot be applied.*
	Examples:
	`direction request`
	`direction not response`
		Yes	No

`divert`	The direction of the connection.
	Allowed values: `input` (incoming—created/initiated from outside the local host), `output` (outgoing—created/initiated on the local host).
	This variable cannot simultaneously contain a set of values; conditions of the `match` *and* `in` *type cannot be applied.*
	Examples:
	`divert input`
	`divert not output`
		Yes	No

`content_type`	MIME type of data transferred during connection. **Usage Aspects:** • Can be defined if only SSL/TLS is not used or it was allowed to unwrap SSL. • The expression “`*/*`” matches data of any MIME type and HTTP replies without the header `Content-Type`. • Dr.Web LookupD can be used to check the value of this variable. • A set of values for checking a variable value is available from the file.	Yes	No

Variable	Description	Can be used in	
		conditional part	**action part (SET)**

	Examples:
	`content_type in ("multipart/byteranges", "application/octet-stream")`
	`content_type not in ("text/*", "image/*")`
	`content_type not in ("audio/*")`
	`content_type in ("*/*")`
	`content_type in LDAP@BlockedContent`
	`content_type not in file("/etc/file")`		

`unwrap_ssl`	Whether the traffic transferred via SSL/TLS is unwrapped.
	Allowed values: `true`, `false`.
	Usage Aspects:
	• The variable always has any value. The instruction `SET unwrap_ssl = ()` is impossible.
	• The variable cannot be used as a condition. It is necessary only to control SSL unwrapping (for example, to display a webpage containing notification about blocking triggered by our side).
	Examples:
	`SET unwrap_ssl = TRUE`
	`set Unwrap_SSL = false`
		No	Yes

`http_templates_dir`	The path to the directory where the notification page template on blocking HTTP request is stored. If the path starts with a / (forward slash), it is an absolute path; if it starts with any other symbol, then it is a relative path. In the latter case it is given relative to the directory specified in the **`TemplatesDir`** parameter. **Usage Aspects:** • It is useful only for the HTTP(S) protocol.	No	Yes

Variable	Description	Can be used in	
		conditional part	action part (SET)

	Examples:
	`SET http_templates_dir = "/etc/mytemplates"`
	`set http_templates_dir = "templates_for_my_site"`		

Variables used in the rules of mail processing

Variable	Description	Can be used in	
		conditional part	action part (SET)

`header`	Contents of email message headers.
	Usage Aspects:
	• Used for comparison of header areas with the list of specified templates (regular expressions are used).
	• Any of the headers represented in the email message could be checked.
	• Comparison is not case-sensitive, Unicode could be used.
	• A set of values for checking a variable value is available from the file.
	Examples:
	`header match ("su..ect: sp.m", "From: sales.*@.*")`
	`Header not match ("Subject: .*buy.*")`
	`header match file("/etc/file")`
		Yes	No

`body`	Text contents of the email message body. **Usage Aspects:** • Used for comparison of email message body with the list of specified templates (regular expressions are used). • Any text part of the email message could be checked. • Comparison is not case-sensitive, Unicode could be used.	Yes	No

Variable	Description	Can be used in	
		conditional part	action part (SET)

	• A set of values for checking a variable value is available from the file.
	Examples:
	`body match (“e.ternit[y] ")`
	`body not match file("/etc/file")`		

`body_part_header`	Headers of the parts of the email message body (MIME part).
	Usage Aspects:
	• Used for comparison of headers in sections of the email message body with the list of specified templates (regular expressions are used).
	• Any header of any part in the email message body could be checked.
	• Comparison is not case-sensitive, Unicode could be used.
	• A set of values for checking a variable value is available from the file.
	Examples:
	`body_part_header match ('Content-Disposition: attachment; .*filename=”virus.exe"')`
	`BodyPartHeader not match ("Content-Disposition: attachment; .*")`
	`body_part_header match file("/etc/file")`
		Yes	No

`attachment_name`	Name of attached files. **Usage Aspects:** • Used for comparison of names of files (*Content-Disposition: attachment*), attached to an email with a list of specified templates (regular expressions are used). • Comparison is not case-sensitive, Unicode could be used. • A set of values for checking a variable value is available from the file.	Yes	No

Variable	Description	Can be used in	
		conditional part	**action part (SET)**

	Examples:
	`attachment_name match ("\.ex.$", "\.js$", "^virus.*")`
	`attachment_name not match ("\.txt$", "\.rtf$")`
	`attachment_name not match file("/etc/file")`		

`total_spam_score`	Normalized rating of an email message as spam (from 0 to 100) received from Dr.Web ASE.
	Normalization of spam scoring received from Dr.Web ASE is performed according to the following rules:
	1. 0 points and less—0.0;
	2. 100 points—0.8;
	3. 1000 points and more—1.0.
	In the indicated intervals normalized scoring increases.
	Usage Aspects:
	• Numerical variable always has one value and could be used only with the conditions of the following types: `lt` and `gt`.
	• If Dr.Web ASE is not installed, scanning of email messages for spam is not performed, and the variable `total_spam_score` could not be used in rules (attempts to check if a condition in the rule is true will lead to the error "*Dr.Web ASE is unavailable*").
	Example:
	`total_spam_score gt 0.32`
	`total_spam_score gt 0.5,`
	`total_spam_score lt 0.95`
		Yes	No

`smtp_mail_from`	Address of the sender sent within the SMTP sessions by the command `MAIL FROM`.	Yes	No

Variable	Description	Can be used in	
		conditional part	action part (SET)

	Usage Aspects:
	• Used for comparison of the name of the sender indicated within an SMTP session with the list of specified templates (regular expressions are used).
	• Comparison is not case-sensitive.
	• This variable cannot be used in rules of the interface *Spamd*: this protocol does not provide information about the email message sender.
	• A set of values for checking a variable value is available from the file.
	Examples:
	`smtp_mail_from match ("^john@.*", ".*@domain.com$")`
	`smtp_mail_from not match ("^user@domain.com$")`
	`smtp_mail_from match file("/etc/file")`		

`smtp_rcpt_to`	List of addresses of the email message recipients sent within the SMTP sessions by the command `RCPT TO`. **Usage Aspects:** • Used for comparison of the Recipient names indicated within an SMTP session with the list of specified templates (regular expressions are used). • Comparison is not case-sensitive. • This variable cannot be used in rules of the interface *Spamd*: this protocol does not provide information about the email message recipient. • If before `match` there is `all`, then the condition with this variable will be true only in case of the match *of all values from the list* with the indicated templates. • A set of values for checking a variable value is available from the file.	Yes	No

Variable	Description	Can be used in	
		conditional part	action part (SET)

	Examples:
	`smtp_rcpt_to match ("^user1@domain.com$", ".*@domain2.com$")`
	`smtp_rcpt_to all match ("^john@.*", ".*@domain.com$")`
	`smtp_rcpt_to match file("/etc/file")`		

`maild_templates_dir`	The path to the directory with a template used for repacking of email messages.
	If the path starts with a / (forward slash), it is an absolute path; if it starts with any other symbol, then it is a relative path. In the latter case it is given relative to the directory specified in the **`TemplatesDir`** parameter.
	Usage Aspects:
	• It is useful only for mail protocols (*POP3*, *IMAP*, *SMTP*) and for MTA interfaces (*Milter*, *Spamd*, *Rspamd*).
	Examples:
	`SET maild_templates_dir = "/etc/my_mail_templates"`
	`set MaildTemplatesDir = "templates_for_my_MTA"`
		No	Yes

Categories of unwanted websites and threats

1. Categories of unwanted websites (for the variables `sni_category`, `url_category`)

Convention	Website category
InfectionSource	Websites containing malicious software (“infection sources”).
NotRecommended	Fraudulent websites (that use “social engineering”) visiting which is not recommended.
AdultContent	Websites containing adult content.
Violence	Websites containing graphic violence.
Weapons	Websites dedicated to weapons.

Convention	**Website category**
Gambling	Gambling websites.
Drugs	Websites dedicated to drugs.
ObsceneLanguage	Websites with obscene language.
Chats	Chat websites.
Terrorism	Websites that contain information about terrorism.
FreeEmail	Websites that offer free email registration.
SocialNetworks	Social networking websites.
DueToCopyrightNotice	Websites that were specified by the holders of copyrights pertaining to content or works protected by copyright law (movies, music, etc.).

As values of the variables `sni_category` *and* `url_category`*, it is also possible to use names of the parameters that control blocking (see below).*

2. Threat categories (for the `threat_category` variable)

Convention	**Threat categories**
KnownVirus	Known threat (virus).
VirusModification	Modification of the known threat (virus).
UnknownVirus	Unknown threat, suspicious object.
Adware	Adware.
Dialer	Dialer.
Joke	Joke.
Riskware	Riskware.
Hacktool	Hacktool.

As a value of the variable `threat_category`*, it is also possible to use names of the parameters that control blocking (see below).*

Appendix B. Technical Support

If you encounter any issues installing or using company products, before requesting for the assistance of the technical support, take advantage of the following options:

- Download and review the latest manuals and guides at https://download.drweb.com/doc/.
- Read the frequently asked questions at https://support.drweb.com/show_faq/.
- Browse the Dr.Web official forum at https://forum.drweb.com/.

If you have not found solution for the problem, you can request direct assistance from Doctor Web company technical support by one of the following ways:

- Fill in the web form in the corresponding section at https://support.drweb.com/.
- Call by phone in Moscow: +7 (495) 789-45-86.

Refer to the official website at https://company.drweb.com/contacts/offices/ for regional and international office information of Doctor Web company.