



Dr.WEB
for UNIX File Servers

Administrator Manual



© Doctor Web, 2018. All rights reserved

This document is the property of Doctor Web. No part of this document may be reproduced, published or transmitted in any form or by any means for any purpose other than the purchaser's personal use without proper attribution.

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 for UNIX File Servers
Version 11.0
Administrator Manual
9/7/2018

Doctor Web Head Office
2-12A, 3rd str. Yamskogo polya
Moscow, Russia
125040

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 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

Conventions and Abbreviations	8
Introduction	9
About this Product	10
Main Functions	10
Program Structure	12
Quarantine Directories	18
File Permissions and Privileges	19
Operation Modes	20
System Requirements	23
Licensing	26
Installing and Removing the Product	27
Installing the Product	27
Installing the Universal Package	28
Installing from Command Line	30
Installing from Repository	31
Upgrading the Product	34
Getting Current Upgrades	34
Upgrading to a Newer Version	35
Removing the Product	38
Uninstalling the Universal Package	38
Uninstalling from Command Line	39
Uninstalling the Product Installed from the Repository	39
Additional Information	42
Product Packages and Files	42
Custom Component Installation and Uninstallation	45
Configuring Security Subsystems	50
Configuring SELinux Security Policies	51
Configuring the Permissions of PARSEC (Astra Linux)	54
Getting Started	56
Registration and Activation of the Product	56
Testing the Operation of the Product	59
File System Monitoring Setting	60
Integration with Samba File Server	61



Integration with NSS Volumes	63
Brief Instructions	65
Components of the Product	70
Dr.Web ConfigD	70
Operating Principles	70
Command-Line Arguments	72
Configuration Parameters	73
Dr.Web Ctl	76
Command-Line Call Format	77
Usage Examples	102
Configuration Parameters	104
Dr.Web Web Management Interface	105
Managing the Components	106
Threats Management	107
Managing the Settings	109
Managing the Central Protection	113
Scanning local files	115
SpIDer Guard	119
Operating Principles	119
Command-Line Arguments	121
Configuration Parameters	122
File System Monitoring Setting	128
Use kernel module for SpIDer Guard	128
SpIDer Guard for SMB	131
Operating Principles	131
Command-Line Arguments	133
Configuration Parameters	134
Integration with Samba File Server	140
Building the VFS SMB Module	140
SpIDer Guard for NSS	143
Operating Principles	143
Command-Line Arguments	145
Configuration Parameters	145
Integration with NSS Volumes	151
Dr.Web ClamD	152
Operating Principles	152



Command-Line Arguments	153
Configuration Parameters	154
Integration with External Applications	159
Dr.Web File Checker	161
Operating Principles	161
Command-Line Arguments	162
Configuration Parameters	163
Dr.Web Network Checker	165
Operating Principles	165
Command-Line Arguments	167
Configuration Parameters	168
Dr.Web Scanning Engine	173
Operating Principles	173
Command-Line Arguments	175
Configuration Parameters	177
Dr.Web Updater	180
Operating Principles	180
Command-Line Arguments	181
Configuration Parameters	182
Dr.Web ES Agent	187
Operating Principles	187
Command-Line Arguments	188
Configuration Parameters	189
Dr.Web HTTPD	192
Operating Principles	192
Command-Line Arguments	194
Configuration Parameters	194
Dr.Web SNMPD	198
Operating Principles	198
Command-Line Arguments	200
Configuration Parameters	201
Integration with SNMP Monitoring Systems	204
Dr.Web SNMP MIB	213
Dr.Web CloudD	246
Operating Principles	246
Command-Line Arguments	247





Configuration Parameters	248
Appendices	249
Appendix A. Types of Computer Threats	249
Appendix B. Neutralizing Computer Threats	253
Appendix C. Contacting Technical Support	256
Appendix D. Configuration File	257
File Structure	257
Parameter Types	259
Appendix E. Generating SSL certificates	262
Appendix F. Known Errors	264
Appendix G. List of Abbreviations	309
Index	311



Conventions and Abbreviations

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.
<i>Anti-virus network</i>	A new term or an accent on a term in descriptions.
<code><IP-address></code>	Placeholders.
Save	Names of buttons, windows, menu items and other program interface elements.
CTRL	Keyboard keys names.
<code>/home/user</code>	Names of files and folders, code examples.
Appendix A	Cross-references on the document chapters or internal hyperlinks to web pages.



Command-line commands, which are entered via a keyboard (in the terminal or terminal emulator), are marked with the command prompt character \$ or # in the current manual. The character indicates the privileges required for execution of the specified command. According to the standard convention for UNIX-based systems

\$—indicates that the command can be executed with user rights.

#—indicates that the command can be executed with superuser (usually *root*) privileges. To elevate the privileges, use **su** and **sudo** commands.

List of abbreviations is in section [Appendix G. List of Abbreviations](#).



Introduction

Thank you for purchasing Dr.Web for UNIX File Servers. It offers reliable protection of your server and its users from distribution of various types of [computer threats](#) using the most advanced virus detection and neutralization [technologies](#).

This manual is intended to help administrators of the servers that run an OS of the **GNU/Linux** family or other UNIX-like operating systems, such as **Solaris** and **FreeBSD**, to install and use Dr.Web for UNIX File Servers 11.0..

Convention for Paths to Product Files

The product described in the present document is designed for operation in different **UNIX**-based operating system. Real paths to product files depend on the operating system installed on the user's computer. For notational convenience, the following conventions are used:

- `<opt_dir>`—directory where main product files are located (including executable files and libraries).
- `<etc_dir>`—directory where the configuration file and a key file are located.
- `<var_dir>`—directory where supporting and temporary product files are located.

Real paths corresponding to the conventions in different operating systems are given in the table below.

OS Type	Convention	Real Path
GNU/Linux, Solaris	<code><opt_dir></code>	<code>/opt/drweb.com</code>
	<code><etc_dir></code>	<code>/etc/opt/drweb.com</code>
	<code><var_dir></code>	<code>/var/opt/drweb.com</code>
FreeBSD	<code><opt_dir></code>	<code>/usr/local/libexec/drweb.com</code>
	<code><etc_dir></code>	<code>/usr/local/etc/drweb.com</code>
	<code><var_dir></code>	<code>/var/drweb.com</code>

For space considerations, examples use paths for **GNU/Linux** operating systems. In some places of the document, where it is possible, examples contain real paths for all of the operating systems.



About this Product

Dr.Web for UNIX File Servers is designed to protect file servers running under UNIX (**GNU/Linux**, **Solaris** and **FreeBSD**) from viruses and other types of any malicious software, as well as to prevent distribution of these threats developed for various platforms.

Main components (anti-virus engine and virus databases) are not only highly effective and resource-sparing, but also cross-platform, which lets Doctor Web specialists create reliable anti-virus solutions protecting computers and mobile devices under popular operating systems from threats that target different platforms. Currently, along with Dr.Web for UNIX File Servers, Doctor Web offers anti-virus solutions for both **UNIX**-based operating systems (such as **GNU/Linux**, **Solaris** and **FreeBSD**) and **IBM OS/2**, **Novell NetWare**, **macOS** and **Windows**. Moreover, other anti-virus products have been developed to deliver protection for devices that run **Android**, **Symbian**, **BlackBerry**, and **Windows Mobile**.

Components of the Dr.Web for UNIX File Servers are constantly updated, and virus databases, databases of web resources categories and databases of rules for spam filtering of email messages are regularly supplemented with new signatures to ensure up-to-date protection of servers, workstations and mobile users and their programs and data. To provide additional protection against unknown viruses heuristic analysis methods are implemented in the anti-virus engine and to the Dr.Web Cloud service that stores information about the latest threats, signatures of which are absent in the database (this function is not available for all products).

Main Functions

Dr.Web for UNIX File Servers main functions:

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). To find more information on computer threat types, refer to [Appendix A. Types of Computer Threats](#).

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. For details on methods used to neutralize threats, refer to [Appendix B. Neutralizing Computer Threats](#).

When scanning the file system on the user's request, it is possible of either full scan of all the file system objects available to user, or selective scan of the specified objects only (separate directories or files that meet the specified criterias). In addition, it is possible to perform



separate checks of boot records of volumes and executable files which support currently active processes in the system. In the latter case, when a threat is detected, it is not only neutralized the malicious executable file, but all processes running from it are forcibly terminated. In systems that implement a mandatory model of access to files with a set of different access levels, the scanning of files that are not available at the current access level can be done in special [autonomous copy](#) mode .

The [Dr.Web Ctl](#) command-line management tool included in the product allows to scan for threats file systems of remote network hosts, that provide remote terminal access via SSH.



The remote scanning can be used only for detection of malicious and suspicious files on a remote host. To eliminate detected threats on the remote host, it is necessary to use administration tools provided directly by this host. For example, for routers and other "smart" devices, a mechanism for a firmware update can be used; for computing machines, it can be done via a connection to them (as an option, using a remote terminal mode) and respective operations in their file system (removal or moving of files, etc.), or via running an anti-virus software installed on them.

2. Monitoring access to files:

- **File system in the OS.** Monitors file events and attempts to run executables. This feature allows to detect and neutralize malware at an attempt to infect the server's file system.

*Note that volume monitoring function is available only for operating systems of **GNU/Linux** family. For other [supported](#) operating systems the component which provides this feature is not in the package.*

- **Samba shared directories.** Read and write operations of local and remote users of the file server are monitored. This feature allows to detect and neutralize malware at an attempt to save a malicious program to the file storage, which prevents its distribution over the network.
- **NSS (Novell Storage Services) volumes.** Monitors write operations of the NSS file storage users. This feature allows to detect and neutralize malware at an attempt to save the malicious program to NSS storage, which prevents its distribution over the network.

*Note that the **Novell Storage Services** volume monitoring function is available only for **Novell Open Enterprise Server SP2 SUSE Linux Enterprise Server** operating system 10 SP3 or later. For other [supported](#) operating systems the component which provides this feature is not in the package.*

3. **Reliable isolation of infected or suspicious objects.** Such objects detected in the server's file system are moved to a special storage, quarantine, to prevent any harm to the system. When moved to quarantine, objects are renamed according to special rules and, if necessary, they can be restored to their original location only on demand.
4. **Automatic update** of the anti-virus engine, virus databases for the maintenance of the high level of protection against malware.
5. **Collection of statistics** on virus events, logging threat detection events. Notification on detected threats over SNMP to external monitoring systems and to the central protection server (if the product operates in [central protection mode](#)).




6. **Operation in central protection mode** (when connected to the central protection server, such as Dr.Web Enterprise Server or as a part of Dr.Web AV-Desk service). This mode allows implementation of a [unified security policy](#) on computers within the protected network. It can be a corporate network, a private network (VPN), or a network of a service provider (for example, a provider of Internet service).

Program Structure



Dr.Web for UNIX File Servers is a product that consists of several components, each of which has its individual set of functions. The list of components included in Dr.Web for UNIX File Servers are listed below.

Component	Description
Dr.Web ConfigD	<p>Configuration daemon Dr.Web for UNIX File Servers, which performs the following functions:</p> <ul style="list-style-type: none">• Starts and stops the product's components depending on the settings. Automatically restarts components if a failure in their operation occurs. Starts components at the request of other components. Informs active components when another component starts or shuts down.• Stores information about present license keys and settings and provides this data to all components. Receives adjusted settings and license keys from the components of Dr.Web for UNIX File Servers expected to provide such information. Notifies other components on changes in license keys and settings. <p>Executable file: drweb-configd</p> <p>Internal name output to the log file: <code>ConfigD</code></p>
Dr.Web Virus-Finding Engine	<p>Anti-virus engine. The main component of the anti-virus protection. Implements algorithms to detect viruses and malicious programs as well as algorithms to analyze suspicious behavior (by using signature and heuristic analysis).</p> <p>Used by all Dr.Web for UNIX File Servers components via Dr.Web Scanning Engine.</p> <p>Executable file: drweb32.dll</p> <p>Internal name output to the log file: <code>CoreEngine</code></p>
Dr.Web Scanning Engine	<p>Scanning engine. The component responsible for loading the anti-virus engine Dr.Web Virus-Finding Engine and virus databases. It transmits the contents of files and disk boot records to the anti-virus engine for scanning at the request of other components of Dr.Web for UNIX File Servers. It queues files that are waiting to be scanned. Cures the files that can be cured. From the point of view of other components of Dr.Web for UNIX File Servers, this component provides</p>



Component	Description
	<p>the anti-virus scanning service. Can operate under the control of the Dr.Web ConfigD configuration daemon or in an autonomous mode (autonomously from other components).</p> <p>Used by all Dr.Web for UNIX File Servers components for the anti-virus scanning.</p> <hr/> <p>Executable file: drweb-se</p> <p>The internal name, displayed in log: <code>ScanEngine</code></p>
Dr.Web virus database	<p>Automatically updated database of these virus signatures and other threats, also algorithms of detection and neutralization of malicious software.</p> <p>Used by the anti-virus engine Dr.Web Virus-Finding Engine and provided along with it.</p>
Dr.Web File Checker	<p>The component which scans file system objects and manages quarantined files. It receives scanning tasks from other Dr.Web for UNIX File Servers components and searches file system directories according to a received task, transmits files for scanning to Dr.Web Scanning Engine scanning engine and notifies components on scanning progress. It also removes infected files, moves them to quarantine, restores them from quarantine, and manages quarantine directories. The component creates and updates cache that stores information on scanned files to lessen the frequency of repeated file scanning.</p> <p>Used by components that scan file system objects, such as SpIDer Guard, SpIDer Guard for SMB, SpIDer Guard for NSS.</p> <hr/> <p>Executable file: drweb-filecheck</p> <p>The internal name, displayed in log: <code>FileCheck</code></p>
SpIDer Guard	<p>Linux file system monitor. Operates in background mode and controls file operations (such as creation, opening, closing, running) in GNU/Linux file systems. It sends the file scanning component requests to scan new or changed files as well as executables of programs when they are run.</p> <p>Depending on its options, OS uses the fanotify mechanism (API provided by the OS) or a special kernel module, developed by Dr.Web (LKM-module is supplied together with SpIDer Guard in a separate package).</p> <div> It is included only in the distributions for GNU/Linux OS.</div>



Component	Description
	<p>Executable file: drweb-spider</p> <p>The internal name, displayed in log: <code>LinuxSpider</code></p>
GNU/Linux kernel module for SpIDer Guard	<p>GNU/Linux kernel module (LKM-module) used by the SpIDer Guard to have access to the file system events in some operating systems, where API fanotify is unavailable or implemented with limited functions (for example, in systems with mandatory access model).</p> <p>The component is included as compiled (for set of operation systems, where fanotify is not implemented or is unavailable) and as source codes, that allow to build and install the operating system kernel module manually (for the instruction, refer to the Building kernel module for SpIDer Guard section).</p> <div> It is included only in the distributions for GNU/Linux OS.</div> <p>Executable file: drweb.ko</p>
SpIDer Guard for SMB	<p>Samba shared directories monitor. Operates in background mode and monitors file system operations (such as creation, opening, closing, read and write operations) in the directories selected as the Samba server's file storages. Sends content of new or modified files for scanning to the file scanning component. Integration with the file server is performed via VFS SMB modules that operate on Samba server side.</p> <p>Executable file: drweb-smbspider-daemon</p> <p>The internal name, displayed in log: <code>SMBSpider</code></p>
SpIDer Guard for NSS	<p>NSS (Novell Storage Services) volumes monitor. Operates in background mode and controls file system operations (such as creation, opening, closing and write operations) on NSS volumes that are mounted on the file system. Sends content of new or modified files for scanning to the file scanning component.</p> <div> It is included only in the distributions designed for GNU/Linux OSes. The component can operate only on Novell Open Enterprise Server SP2 based on SUSE Linux Enterprise Server 10 SP3 and older.</div> <p>Executable file: drweb-nss</p> <p>The internal name, displayed in log: <code><%NSSSPIDER_SECTION%></code></p>
Dr.Web ES Agent	<p>Central protection agent. Makes it possible for the product to operate in centralized and mobile modes. Provides communication between</p>



Component	Description
	<p>the product and the central protection server, a license key file, updates to the virus databases and components. Sends to the server information on the components included in Dr.Web for UNIX File Servers and their state as well as statistics of virus events.</p> <hr/> <p>Executable file: drweb-esagent</p> <p>The internal name, displayed in log: <code>ESAgent</code></p>
Dr.Web Network Checker	<p>An agent of the network data scanning. Used to send data to the scanning engine for actual scanning. The data is sent by components of the product via the network (such components as Dr.Web ClamD).</p> <p>Besides, it allows Dr.Web for UNIX File Servers to arrange a distributed scanning of files: to receive/transmit files for scanning from/to remote hosts. For that purpose, remote hosts must feature an installed and running Dr.Web for UNIX-based operating systems. In the distributed scanning mode, it allows automatic distribution of scanning load among remote hosts by reducing load on hosts with a large number of scanning tasks (for example, on mail servers, file servers, Internet gateways).</p> <p>For security reasons, files are transmitted over SSL.</p> <hr/> <p>Executable file: drweb-netcheck</p> <p>The internal name, displayed in log: <code>NetCheck</code></p>
Dr.Web HTTPD	<p>Web interface for managing Dr.Web for UNIX File Servers components. It consists of management web interface (it should be installed separately) and service interface for operation of Dr.Web Link Checker (can be installed additionally) browser extension. You can access the interface via any browser on a local or remote host. In-built web interface enables the product to use neither third-party web servers (such as Apache HTTP Server) nor remote administration tools, such as Webmin.</p> <p>For security reasons, web interface interacts with user over HTTPS.</p> <hr/> <p>Executable file: drweb-httpd</p> <p>The internal name, displayed in log: <code>HTTPD</code></p>
Dr.Web Ctl	<p>Tool for managing Dr.Web for UNIX File Servers from the command line.</p> <p>Allows the user to start file scanning, to view quarantined objects, to start a virus database update procedure, to connect the product to or to disconnect it from the central protection server, to view and to configure parameters.</p> <hr/> <p>Executable file: drweb-ctl</p> <p>The internal name, displayed in log: <code>Ctl</code></p>



Component	Description
Dr.Web Updater	<p>An update component. Downloads from Doctor Web servers updates of the virus databases, anti-virus engine.</p> <p>The updates can be downloaded automatically, according to a schedule, and on user's demand (via Dr.Web Ctl or management web interface).</p> <hr/> <p>Executable file: drweb-update</p> <p>The internal name, displayed in log: <code>Update</code></p>
Dr.Web SNMPD	<p>An SNMP agent. Designed for integration of Dr.Web for UNIX File Servers with external monitoring systems over SNMP. Such integration allows you to monitor the state of the product's components and to collect statistics on threat detection and neutralization. Supports SNMP v2c and v3.</p> <hr/> <p>Executable file: drweb-snmpd</p> <p>The internal name, displayed in log: <code>SNMPD</code></p>
Dr.Web ClamD	<p>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 File Servers for anti-virus scanning.</p> <hr/> <p>Executable file: drweb-clamd</p> <p>The internal name, displayed in log: <code>ClamD</code></p>
Dr.Web CloudD	<p>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.</p> <hr/> <p>Executable file: drweb-cloudd</p> <p>The internal name, displayed in log: <code>CloudD</code></p>



The figure below shows the structure of Dr.Web for UNIX File Servers and its operation with external applications.

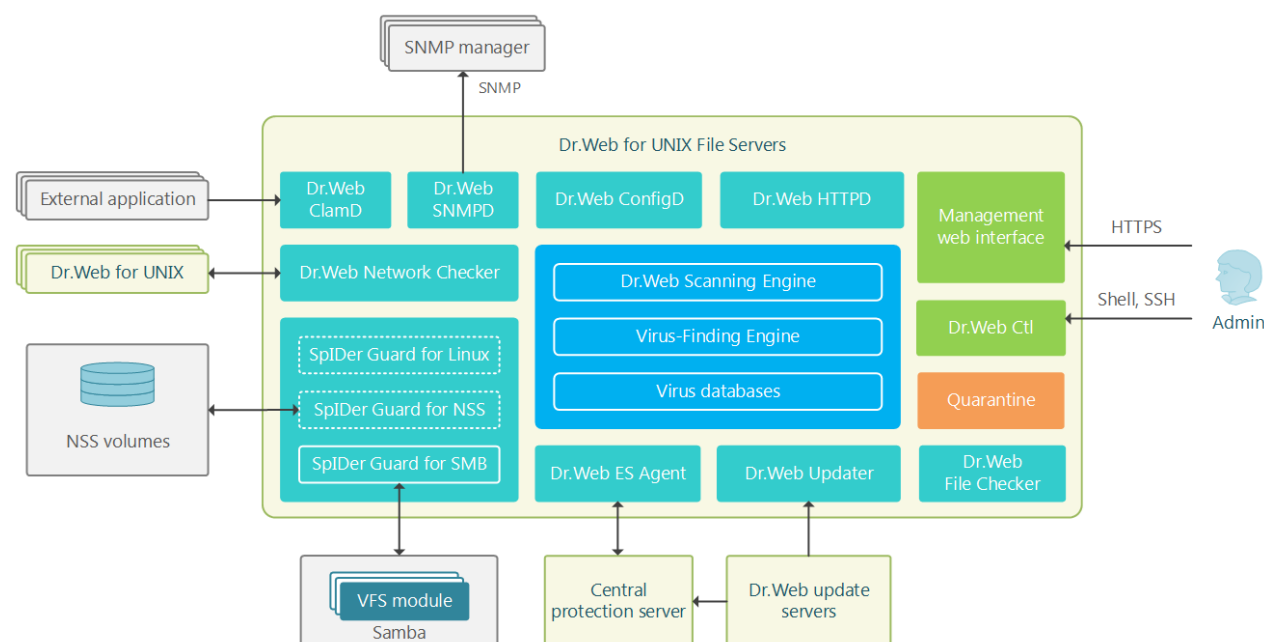


Figure 1. Dr.Web for UNIX File Servers structure

In this scheme, the following notations are used:

	— Dr.Web for UNIX File Servers as a whole and external Dr.Web applications together with systems which are not included in the solution.
	— external to Dr.Web for UNIX File Servers programs and products for its integration.
	— Components that are included in Dr.Web for UNIX File Servers engine. Other product components use the engine as a service that performs anti-virus checks.
	— Service components designed to perform particular anti-virus protection functions (for example, scanning file system objects, updating virus databases, establishing connection to central protection servers, managing the operation of the product).
	— Components that provide the user with the interface for Dr.Web for UNIX File Servers.
	— Quarantine as a set of file system directories which store isolated malicious files.

Components marked with a dashed line can be missing depending on the distribution.

For details on Dr.Web for UNIX File Servers components, refer to [Components of the Product](#).



Quarantine Directories

Quarantine directories of Dr.Web for UNIX File Servers 11.0 serve for isolation of files that pose a threat to system security and cannot be currently cured. Such threats are those that are unknown to Dr.Web for UNIX File Servers (that is, a virus is detected by the heuristic analyzer but the virus signature and method to cure are absent in the databases) or those that caused an error during curing. Moreover, a file can be quarantined at user request if the user selected this [action](#) in the list of detected threats or specified this action in settings as reaction to this threat [type](#).

When a file is quarantined, it is renamed according to special rules. Renaming of isolated files prevents their identification by users or applications and complicates access to them in case of attempt to bypass quarantine management tools implemented in Dr.Web for UNIX File Servers. Moreover, when a file is moved to quarantine, the execution bit is reset to prevent an attempt to run this file.

Quarantine directories are located in

- *user home directory* (if multiple user accounts exist on the computer, a separate quarantine directory can be created for each of the users)
- *root directory of each logical volume* mounted to the file system

Dr.Web quarantine directories are always named as `.com.drweb.quarantine` and are not created until the Quarantine [action](#) is applied. At that, only a directory required for isolation of a concrete object is created. When selecting a directory, the file owner name is used: search is performed upwards from the location where the malicious object resides and if the owner home directory is reached, the quarantine storage created in this directory is selected. Otherwise, the file is isolated in the quarantine created in the root directory of the volume (which is not always the same as the file system root directory). Thus, any infected file moved to quarantine is always located on the volume, which provides for correct operation of quarantine in case several removable data storages and other volumes are mounted to different locations in the system.

A user can manage quarantine contents from the command line using the utility [Dr.Web Ctl](#), or via the [management web interface](#) (if it is installed). Every action is applied to the consolidated quarantine; that is, changes affect all quarantine directories available at the moment.





Operation with quarantined objects is allowed even if no [active license](#) is found. However, isolated objects cannot be cured in this case.

Not all anti-virus components of Dr.Web for UNIX File Servers can use Quarantine for threat isolation. For example, it is not used by the Dr.Web ClamD, as well as by Dr.Web ICAPD and Dr.Web MailD components (may not be included in the your product).



File Permissions and Privileges

To scan objects of the file system and neutralize threats, Dr.Web for UNIX File Servers (or rather the user under whom it runs) requires the following permissions:

Action	Required rights
<i>Listing all detected threats</i>	Unrestricted. No special permission required.
<i>List archive contents</i> (display only corrupted or malicious elements)	Unrestricted. No special permission required.
<i>Moving to quarantine</i>	Unrestricted. The user can quarantine all infected files regardless of read or write permissions on them.
<i>Deleting threats</i>	<p>The user needs to have write permissions for the file that is being deleted.</p> <div> If threat is detected in a file located in a container (an archive, email message, etc.), its removal is replaced with moving of a container to quarantine.</div>
<i>Curing</i>	<p>Unrestricted. The access permissions and owner of a cured file remain the same after curing.</p> <div> The file can be removed if deletion can cure the detected threat.</div>
<i>Restoring a file from quarantine</i>	The user should have permissions to read the file and to write to the restore directory.
<i>Deleting a file from quarantine</i>	The user must possess write permissions to the file that was moved to quarantine.

To enable operation of the command-line management [Dr.Web Ctl](#) tool with superuser (*root*) privileges, you can use the **su** command, which allows to change the user, or the **sudo** command, which allows you to execute a command as another user.



Note that Dr.Web Scanning Engine scanning engine cannot check file which size exceeds 4 Gbytes (on attempt to scan such files, the following error message displays: "File is too large").



Operation Modes

Dr.Web for UNIX File Servers can operate both in standalone mode and as a part of an *anti-virus network* managed by a *central protection server*. Operation in *central protection mode* does not require installation of additional software or Dr.Web for UNIX File Servers re-installation or removal.

- *In Standalone mode*, the protected computer is not connected to an anti-virus network and its operation is managed locally. In this mode, configuration and license key files are located on local disks and Dr.Web for UNIX File Servers is fully controlled from the protected computer. Updates to virus databases are received from Doctor Web update servers.
- *In Central protection mode (Enterprise mode)*, protection of the computer is managed by the central protection server. In this mode, some functions and settings of Dr.Web for UNIX File Servers can be adjusted in accordance with the general (corporate) anti-virus protection policy implemented on the anti-virus network. The license key file used for operating in enterprise mode is received from the central protection server. The key file stored on the local computer, if any, is not used. Statistics on virus events is sent to the central protection server. Updates to virus databases are also received from the central protection server.
- *In Mobile mode*, Dr.Web for UNIX File Servers receives updates from Doctor Web update servers, but operation of the product is managed with the local settings. The used key file is received from the central protection server. You can switch to mobile mode only if it is allowed in the central protection server settings.

Central Protection Concept

Doctor Web solutions for central protection use client-server model (see the figure below).

Workstations and servers are protected by *local anti-virus components* (herein, Dr.Web for UNIX File Servers) installed on them, which provides for anti-virus protection of remote computers and allows connection between the workstations and the central protection server.

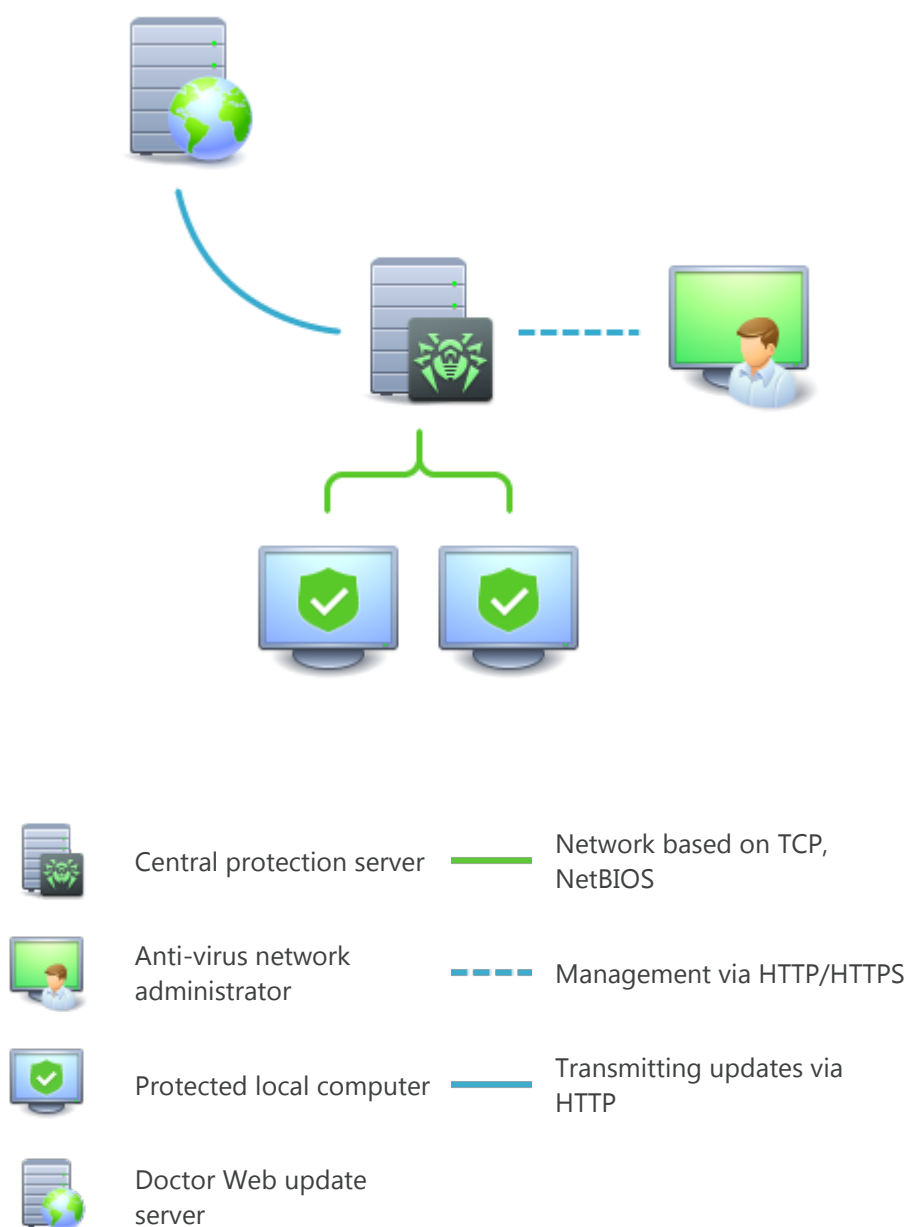


Figure 2. Logical structure of the Anti-virus Network

Local computers are updated and configured from the *central protection server*. The stream of instructions, data and statistics in the anti-virus network goes also through the central protection server. The volume of traffic between protected computers and the central server can be quite sizeable, therefore solutions provide options for traffic compression. To prevent leak of sensitive data or substitution of software downloaded onto protected computers, encryption is also supported.

All necessary updates are downloaded to the central protection server from Doctor Web update servers.



Local anti-virus components are configured and managed from the central protection server according to commands received from anti-virus network administrators. Administrators manage central protection servers and topology of anti-virus networks (for example, validate connections to the central protection server from remote computers) and configure operation of local anti-virus components when necessary.



Local anti-virus components are not compatible with anti-virus products of other companies or anti-virus solutions of Dr.Web if the latter do not support operation in central protection mode (for example, Dr.Web Anti-virus, version 5.0). Installation of two anti-virus programs on the same computer can cause a system crash and loss of important data.

Note that the current version of Dr.Web for UNIX File Servers *does not fully* implement the central protection mode: the central protection server cannot manage operation settings of the program components and cannot send scan tasks for the program.

Connecting to an Anti-Virus Network

Dr.Web for UNIX File Servers can be connected to the central protection server of an anti-virus network using the `esconnect` [command](#) of the [Dr.Web Ctl](#) command-line-based management tool.



Verification of central protection server requires use of public encryption keys, that is, each server is supplied with a unique public key. By default, the central protection agent [Dr.Web ES Agent](#) does not allow connection to the server unless you provide a file containing a public key for authentication of the used server. Such public key file should be obtained from the administrator of your anti-virus network serviced by the server to which you want to connect Dr.Web for UNIX File Servers.

If Dr.Web for UNIX File Servers is connected to the central protection server, you can switch the product into the Mobile mode or switch it back into the Central protection mode. Switching the Mobile mode on or off is accomplished with the help of the **MobileMode** [configuration parameter](#) of the [Dr.Web ES Agent](#) component.



Operation can switch to Mobile mode only if it is allowed in the settings on the central protection server.


Disconnecting from an Anti-Virus Network

Dr.Web for UNIX File Servers can be disconnected from the central protection server of an anti-virus network using the `esdisconnect` [command](#) of the [Dr.Web Ctl](#) command-line-based management tool.



System Requirements

You can use Dr.Web for UNIX File Servers on a computer that meets the following requirements:

Component	Requirement
Platform	CPU with the Intel/AMD architecture and command system are supported: 32-bit (IA-32, x86); 64-bit (x86_64, x64, amd64).
Hard disk space	At least 1 GB of free disk space on a volume where the Dr.Web for UNIX File Servers directories are located.
Operating System	GNU/Linux (kernel 2.6.37 or newer and library glibc 2.13 or newer), FreeBSD or Solaris for Intel x86/amd64 platforms. <div> For systems operating on 64-bit platforms, support of 32-bit applications <i>must</i> be enabled (probably, additional libraries must be installed for this, see below).<hr/>Operation system must support the PAM authentication mechanism.</div> <p>Tested operating system distributions are listed below.</p>
Other	The following valid network connections: <ul style="list-style-type: none">Valid Internet connection to enable updates for virus databases and Dr.Web for UNIX File Servers components.When operating in central protection mode, connection to the server on the local network is enough; connection to the Internet is not required.

Tested Operating System Distributions

The product was tested on the following distributions:

- **GNU/Linux:**

Linux distribution name	Versions	Platforms
Astra Linux Special Edition (Smolensk)	1.5	x86_64
CentOS	6.9, 7.4	x86, x86_64
Debian	7.11, 8.10, 9.3	x86_64
Fedora	27	x86, x86_64



Linux distribution name	Versions	Platforms
Red Hat Enterprise Linux	7.4	x86_64
SUSE Linux Enterprise Server	11 SP4, 12 SP3	x86_64
Ubuntu	14.04, 16.04	x86_64

Other **GNU/Linux** distributions that meet the above-mentioned requirements have not been tested for compatibility with Dr.Web for UNIX File Servers but may be supported. If a compatibility issue occurs, contact technical support on the official website at <https://support.drweb.com/request/>.

- **FreeBSD:**

Versions	Platforms
10.3, 11.1	x86, x86_64

- **Solaris:**

Versions	Platforms
10 u11	x86, x86_64



For **FreeBSD** and **Solaris**, the product can be installed only from the [universal package](#).

Additional Packages

- For **CentOS, Debian, Fedora, Red Hat Enterprise Linux, Ubuntu** on the platform **x86_64**, the package that enables support for 32-bit applications (**libc6-i386** or **glibc.i686**, depending on OS).



For convenient work with Dr.Web for UNIX File Servers in the [command line](#), you can enable command auto-completion in the used command shell (if disabled).

If you encounter any problem with installation of additional packages and components, refer to manuals for the used distribution of the operating system.



Disclaimer

- Operation of SpIDer Guard in the operating system kernel mode (LKM module) is *not supported* for operating systems launched in the **Xen** hypervisor environment. An attempt to load the SpIDer Guard kernel during the OS operation in the **Xen** environment can lead to a [critical error](#) of the OS kernel (so called “Kernel panic” error).

Supported File Servers

1. Samba File Service

For [integration](#) with **Samba** file service, the installed and configured file server **Samba** is required (supported Samba versions: 3.0.33, 3.0.36, 3.2.0 – 3.6.0, 4.0.0 – 4.8.0).



The SpIDer Guard for SMB monitor of Dr.Web for UNIX File Servers uses a special VFS SMB module for the integration with the **Samba** server. With the SpIDer Guard for SMB component, several versions of this module are supplied. They are built for various versions of **Samba**. However, the supplied versions of the VFS SMB module may be incompatible with the version of **Samba** installed on your file server. It may occur, for example, if your **Samba** server uses the `CLUSTER_SUPPORT` option.

If VFS SMB modules are incompatible with your **Samba** server, *the corresponding message is shown* during the product [installation](#). In this case, build the VFS SMB module for your **Samba** server manually (including the compatibility with the `CLUSTER_SUPPORT` option if necessary).

The procedure of building the VFS SMB module from the supplied source code files is described in the [Building the VFS SMB Module](#) section.

2. NSS File Service

For [integration](#) with **NSS** file service, the installed and configured **Novell Open Enterprise Server** SP2 based on the operating system **SUSE Linux Enterprise Server** 10 SP3 or newer (11 SP1, SP2) is required.

Compatibility with Security Subsystems

By default, Dr.Web for UNIX File Servers does not support **SELinux**. In addition, Dr.Web for UNIX File Servers operates in reduced functionality mode in the **GNU/Linux** systems that use mandatory access models (for example, in systems supplied with the **PARSEC** mandatory access subsystem that appends different privilege levels to users and files).

If installation of Dr.Web for UNIX File Servers is required for systems with **SELinux** (as well as for systems that use mandatory access models). It is necessary to execute additional settings of a security subsystem so that Dr.Web for UNIX File Servers operates in full functionality mode. For details, refer to the section [Configuring Security Subsystems](#).



Licensing

Permissions to use Dr.Web for UNIX File Servers are granted by the *license* purchased from Doctor Web company or from its partners. License parameters determining user rights are set in accordance with the License agreement (see <https://license.drweb.com/agreement/>), which the user accepts during product installation. The license contains information on the user and the vendor as well as usage parameters of the purchased product, including:

- List of components licensed to the user
- License period
- Other restrictions (for example, number of computers on which the purchased product is allowed for use).

For evaluation purposes users may also activate *demo period*. After successful activation, demo period provides users with full functionality of Dr.Web for UNIX File Servers for the whole activated period.

Each Doctor Web product license has a unique serial number associated with a special file stored on the user computer. This file regulates operation of product components in accordance with the license parameters and is called a *license* key file. Upon activation of a demo period, a special key file, named a *demo* key file, is automatically generated.

If a license or a demo period are not activated on the computer, Dr.Web for UNIX File Servers components are blocked. Moreover, updates for virus databases and components cannot be downloaded from Doctor Web update servers. But you can activate the product by connecting it to the central protection server as a part of the [anti-virus network](#) administered by the enterprise or Internet service provider. In this case, operation of Anti-virus and updating are managed by the central protection server.



Note that the current version of Dr.Web for UNIX File Servers *does not* fully implement the central protection mode: the central protection server cannot manage operation settings of the program components.



Installing and Removing the Product

This section describes how to [install](#) and [uninstall](#) the Dr.Web for UNIX File Servers version 11.0. In this section, you can also find information on how to obtain [current updates](#) and a procedure of [upgrading to a new version](#), if the previous version of Dr.Web for UNIX File Servers is already installed on your computer.

Besides, this section describes the procedure of [custom installation and uninstallation](#) of the product components (for example, to resolve errors that occurred during the course of the Dr.Web for UNIX File Servers operation or to get an installation with a limited function set) and [configuration of advanced security subsystems](#) (such as **SELinux**) that could be necessary for installation and operation of the product.

To perform these procedures, root permissions are required (i.e. privileges of the *root* user). To elevate privileges when installing or uninstalling the product, use the **su** command for changing the current user or the **sudo** command to execute the specified command with the privileges of another user.



Compatibility *is not guaranteed* for Dr.Web for UNIX File Servers and anti-virus products of other developers. Due to the fact that installation of two anti-viruses on one machine can lead to *errors in the operation system and loss of important data*, before the installation of Dr.Web for UNIX File Servers, *it is strongly recommended* that you delete anti-virus products of other developers from the computer.

If your computer *already has* other Dr.Web anti-virus product installed from the [universal package](#) (`.run`), and you want to install one more Dr.Web anti-virus product (for example, you have Dr.Web for Linux product from the universal package installed, and in addition you want to install Dr.Web for UNIX File Servers), it is necessary to make sure that the version of the installed product *is the same* as the version of the product you want to install. If the product version that you plan on installing is newer than the installed product version, *before* installation, it is necessary to [upgrade](#) the installed product to the version of the product you want to install additionally.

For **FreeBSD** and **Solaris**, the product can be installed only from the [universal package](#).

Installing the Product

To install Dr.Web for UNIX File Servers, do one of the following:

1. From the Doctor Web's official website, download the installation file that contains a [universal package](#) for UNIX systems. The package includes an installer (due to the fact that the installation program is developed for the command line mode, for its operation in the mode of the graphical desktop, you will need to have a terminal emulator available).
2. Install the product in the form of a set of [native packages](#) (to do this, you will need to connect to the corresponding package repository of Doctor Web).



For **FreeBSD** and **Solaris**, the product can be installed only from the [universal package](#).

Regardless of the selected way to install Dr.Web for UNIX File Servers, after the installation completes, you need to activate the license and to install the received key file. Moreover, you can [connect](#) the product to a central protection server. For details, refer to [Licensing](#).

Otherwise, *anti-virus protection remains disabled*.

After you installed the product by any of the mentioned means, you can [uninstall](#) or [update](#) it if there are fixes for its components available or if a new product versions is released. If required, you can also [configure security subsystems](#) of **GNU/Linux** for correct operation of the installed product. If there is a problem with functioning of any individual components, you can perform their [custom installation and uninstallation](#), without uninstalling the entire installed product.

Installing the Universal Package

Dr.Web for UNIX File Servers is distributed as an installation file named `drweb-<version>-av-srv-<OS>-<platform>.run`, where `<OS>` is a type of **UNIX-based** operation system, `<Platform>` is the platform for which the product is intended (x86 for 32-bit platforms and amd64 for 64-bit platforms). For example:

```
drweb-11.0.7-av-srv-linux-x86.run
```

Note that the installation file name corresponding to the above-mentioned format is referred to as `<file_name>.run` below in this section.

To install Dr.Web for UNIX File Servers components:

1. If you do not have the installation file containing the universal package, download it from the Doctor Web's official website: <https://download.drweb.com/>.
2. Save the installation file to the hard disk drive of your computer.
3. Allow the archive to be executed, for example, by using the following command:

```
# chmod +x <file_name>.run
```

4. Execute the archive using the following command:

```
# ./<file_name>.run
```

or use the standard file manager of the graphical shell for both changing the file properties (permissions) and running the file.

This will run an integrity check of the archive, after which the archived files are unpacked to a temporary directory and an installation program is started. If the user does not have root privileges, the installation program attempts to elevate its privileges asking you for the root password (**sudo** is used). If the attempt fails, the installation process aborts.



If the path to the temporary directory in the file system has not enough free space for the unpacked files, the installation process is aborted and an appropriate message is displayed. In this case, change the value of the `TMPDIR` system environment variable so that it points to a directory with enough free space and repeat the installation. You can also use the `--target` option.

After that the installer for the [command-line mode](#) is automatically started (to run it in a graphical desktop environment, you need any terminal emulator).

5. Follow the installer's instructions.
6. You can also start the installation program in a silent mode by executing the following command:

```
# ./<file_name>.run -- --non-interactive
```

In this case the installation program is started in the silent mode and will operate without a user interface (this means it also will not have any dialogs that are normally displayed in the command-line mode).

Note that

- Using this option means that you *accept* the terms of the Dr.Web License Agreement. The License Agreement's text is located in the `/opt/drweb.com/share/doc/LICENSE` file. The file extension indicates the language of the License Agreement. If the `LICENSE` file does not have any extension, the Dr.Web License Agreement is written in English. If you *do not accept* the terms of the License Agreement, you must [uninstall](#) the product after its installation.
- Administrative (root) privileges are required to start the uninstall program in silent mode. To elevate the privileges, you can use the **su** and **sudo** commands.



If the used **GNU/Linux** distribution features **SELinux**, the installation process can be interrupted by the security subsystem. If such situation occurs, set **SELinux** to the *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 the product components.

For details on conventions for `<opt_dir>`, `<etc_dir>`, and `<var_dir>`, refer to the [Introduction](#).

All unpacked installation files are deleted once the installation process completes.



It is recommended that you save file `<file_name>.run`, from which the installation was performed, for the possibility of reinstallation of the product or its components without the need to update the product version.



Installing from Command Line

Once you start the program for the command-line-based installation, a message will be displayed inviting you to install the product.

1. To start the installation process, enter *Yes* or *Y* in response to the “Do you want to continue?” question. If you choose not to install the Anti-virus on your computer, enter *No* or *N*. In this case, the installation will be canceled.
2. After that, you need to view the terms of Dr.Web License Agreement which is displayed on the screen. Press ENTER to scroll the text down line by line or SPACEBAR to scroll it down one screenful at a time. Note that options to scroll the License agreement up are not provided.
3. After you read the License agreement text, you are prompted to accept the terms. Type *Yes* or *Y* if you accept the License agreement. If you refuse to accept it, type *No* or *N*. In the latter case, the installer exits.
4. Once you accept the terms of the License Agreement, installation starts automatically. During the procedure, information about the installation process, including the list of installed components, will be displayed on the screen.
5. Once the installation successfully completes, then—in case an automated configuration procedure is available in the product—an interactive setup script for the product is automatically started. After it finishes its operation, an appropriate message will be displayed on the screen, informing you on how to manage the operation of the product.

If an error occurs, a message describing the error is displayed on the screen and then the installer exits. When the installation process fails due to an error, remove the problems that caused this error and start the installation again.

Interactive Setup Script

The interactive setup script allows you install the product’s license key file that you have as well as to automatically [integrate](#) Dr.Web for UNIX File Servers with the **Samba** file server and helps to specify the list of shared directories that must be monitored by the [SpIDer Guard for SMB](#) monitor.

1. If you enter *n* or *no*, execution of the script will end. If you wish to configure such an integration, enter *y* or *yes* as the answer to the question “Do you want to continue?”.
2. If a valid [key file](#) is not available on your computer (in the product’s standard directory for keeping the key file), the script will offer you to specify the path to a valid key file. Otherwise (i.e. if a valid key file has been found), this step will be automatically skipped.

To skip this step, enter *0*. Later, you can [install](#) a key file manually. If a valid key file is already available on your computer, specify the path to it and press ENTER. The file will be copied to the product’s standard directory for keeping the key file.
3. Next, allow or deny modification of the `smb.conf` configuration file of the **Samba** server and confirm that the installation script has found the right path to the **samba** server’s daemon. Otherwise, specify the correct path.
4. Select any shared directories, managed by **Samba**, that must be monitored by SpIDer Guard for SMB. For that, follow the instructions given by the script:



- If you specify the number of a shared directory that is not yet marked with an [X], this directory will be added to the list of monitored directories. Conversely (if the directory is already marked with an [X]), then entering its number will exclude it from monitoring.
- If you enter the letter *A* or *All*, all available shared directories will be added for monitoring; if you enter *N* or *None*, all shared directories will be excluded from monitoring.

To finish the selection and save the changes to the configuration file, enter *0*, *Q*, or *Quit*.

5. After that, all changes are saved to the configuration file. Additionally, the required version of the VFS SMB module's library will be determined and the corresponding link will be added to the **Samba** server's directory.
6. After you finish adjusting the settings, press ENTER to end the execution of the script.

After you finish adjusting the settings, press ENTER to end the execution of the script.

Installing from Repository

Dr.Web for UNIX File Servers's native packages are stored in the Dr.Web official repository at <https://repo.drweb.com/>. Once you have added the Dr.Web repository to the list of those used by your operating system's package manager, you can install the product from native packages as you install any other programs from the operating system's repositories. Required dependencies are automatically resolved.



All the commands mentioned below—the commands used to add repositories, to import digital signature keys, to install and remove packages—must be performed with superuser (**root**) privileges. To elevate the privileges, use the **su** command (to change the current user) or the **sudo** command (to execute the specified command with another user's privileges).

Note that for the **FreeBSD** and **Solaris** operating systems, the product can be installed only from the [universal package](#).

Debian, Mint, Ubuntu (apt)



The Dr.Web for UNIX File Servers anti-virus engine uses a 32-bit architecture *x86*; in 64-bit systems **Debian**, **Mint**, **Ubuntu** (for platforms *x86-64*, *x64*, *amd64*), a permission could be required for installation of packages for the platform *x86*. It could be obtained via the following command:

```
# dpkg --add-architecture i386
```



1. The repository for these operating systems is digitally signed by Doctor Web. To access the repository, import and add to the package manager storage the digital signature key via execution of the following command:

```
# apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 10100609
```

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

```
deb http://repo.drweb.com/drweb/debian 11.0 non-free
```



Besides, you can execute items 1 and 2 by downloading from the repository and installing a special DEB package <https://repo.drweb.com/drweb-repo11.deb>.

3. To install Dr.Web for UNIX File Servers from the repository, use the following commands:

```
# apt-get update
# apt-get install drweb-file-servers
```

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

ALT Linux, PCLinuxOS (apt-rpm)

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

```
rpm http://repo.drweb.com/drweb/altlinux 11.0/<arch> drweb
```

where `<arch>`—representation of the used packet architecture:

- For the **32-bit** version: `i386`
- For **64-bit** version: `x86_64`

2. To install Dr.Web for UNIX File Servers from the repository, use the following commands:

```
# apt-get update
# apt-get install drweb-file-servers
```

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

Mageia, OpenMandriva Lx (urpmi)

1. Connect the repository using the following command:

```
# urpmi.addmedia drweb https://repo.drweb.com/drweb/mandriva/11.0/<arch>/
```




where *<arch>*—representation of the used packet architecture:

- For the **32-bit** version: `i386`
- For **64-bit** version: `x86_64`

3. To install Dr.Web for UNIX File Servers from the repository, use the following command:

```
# urpmi drweb-file-servers
```

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

Red Hat Enterprise Linux, Fedora, CentOS (yum, dnf)

1. Add a file `drweb.repo` with the contents described below to the `/etc/yum.repos.d` directory:

```
[drweb]
name=DrWeb - 11.0
baseurl=https://repo.drweb.com/drweb/el5/11.0/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://repo.drweb.com/drweb/drweb.key
```



If you plan on logging the indicated above contents to a file using such commands as **echo** with redirecting of an output, a symbol `$` must be escaped: `\$`.

Besides, you can execute item 1 by downloading from the repository and installing a special RPM package <https://repo.drweb.com/drweb-repo11.rpm>.

2. To install Dr.Web for UNIX File Servers from the repository, use the following command:

```
# yum install drweb-file-servers
```

In the **Fedora** operating system, starting from version 22, it is recommended that instead of manager **yum** the manager **dnf** is used, for example:

```
# dnf install drweb-file-servers
```

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

SUSE Linux (zypper)

1. To add the repository, use the following command:

```
# zypper ar -t YUM 'https://repo.drweb.com/drweb/el5/11.0/$basearch/' drweb
```



2. To install Dr.Web for UNIX File Servers from the repository, use the following commands:

```
# zypper refresh
# zypper install drweb-file-servers
```

You can also use alternative package managers (for example, **YaST**) to install the product.

Upgrading the Product

There are two modes for updating Dr.Web for UNIX File Servers:

1. [Getting updates of packages and components](#) released in the course of operation of the current product version (usually such updates contain error fixing and minor improvements in component functioning);
2. [Upgrading to a newer version](#). This upgrading option is used if Doctor Web released a new version of the product you use, and it has new features.

Getting Current Upgrades

After installation of the product using any method described in the [corresponding section](#), the package manager automatically connects to the Dr.Web [package](#) repository:

- If installation was performed from the [universal package](#) (file `.run`), and the system uses DEB packages (for example, such operating systems as **Debian**, **Mint**, **Ubuntu**), there is no package manager in an operating system (**FreeBSD**, **Solaris**), for operation with Dr.Web packages, an individual version of package managers **zypper** is used. It is automatically installed during the product installation.

To get and install the updated Dr.Web packages with this manager, go to the `<opt_dir>/bin` directory (for **GNU/Linux**—`/opt/drweb.com/bin`), and execute the following commands:

```
# ./zypper refresh
# ./zypper update
```

- In all other cases use commands for updating of the package manager used in your OS, for example:
 - For **Red Hat Enterprise Linux** and **CentOS**, use the command **yum**
 - For **Fedora**, use the command **yum** or **dnf**
 - For **SUSE Linux**, use the command **zypper**
 - For **Mageia**, **OpenMandriva Lx**, use the command **urpmi**
 - For **Alt Linux**, **PCLinuxOS**, **Debian**, **Mint**, **Ubuntu**, use the command **apt-get**.

You can also use alternate package managers developed for your operating system. If necessary, refer to the instruction manual for the package manager you use.



If a new product version is released, packages with its components are put into the section of the Dr.Web repository corresponding to the new product version. In this case, an update requires switching of the package manager to a new Dr.Web repository section (refer to [Upgrading to a Newer Version](#)).

Upgrading to a Newer Version

Introductory Remarks

Please note that your version of Dr.Web for UNIX File Servers must be upgraded in the same way that was used to install the product:

- If the current version was installed from the repository, an upgrade requires updating program packages from the repository.
- If the current version was installed from the universal package, then to upgrade the product, you need to install another universal package that contains a newer version of the product.



To identify how the product version, which requires an update, has been installed, check whether the directory with the product's executables contains a program removal script `uninst.sh`. If it does contain this script, then the current version was installed from the universal package; otherwise, it was installed from the repository.

Note that for the **FreeBSD** and **Solaris** operating systems, the product can be installed only from the [universal package](#).

If you cannot update the product the way you installed it initially, uninstall your current version of Dr.Web for UNIX File Servers, and then install a new version using any convenient method. Installation and uninstallation procedures for previous Dr.Web for UNIX File Servers versions are the same as [installation](#) and [uninstallation](#) described in the current manual for version 11.0. For additional information, see Administrator manual for your current version of Dr.Web for UNIX File Servers.

If the current version of the product is operating in the [central protection](#) mode, it is recommended that you record the address of the used central protection server. For example, to determine the address to which Dr.Web for UNIX File Servers of the version higher than 6.0.2, you can use the following command:

```
$ drweb-ctl appinfo
```

In the output provided by this command, from the line that looks like:

```
ESAgent; <PID>; RUNNING 1; Connected <address>, on-line
```



save the `<address>` part (which can look like `tcp://<IP address>:<port>`, for example: `tcp://10.20.30.40:1234`). In addition, it is recommended that you save the server public key file.

In case there are any problems with finding out the parameters of the connection that you are currently using, refer to the Administrator's Manual for the product version that you are currently using and to the administrator of your anti-virus network.

Installing Universal Package for an Upgrade

Install Dr.Web for UNIX File Servers 11.0 from the [universal package](#). If an automatic update of the installed product is impossible, during the installation of the new version, you will get an offer to automatically remove the components of the older version of the product installed on your computer.



If during the update process you need to remove the installed product version, and there are multiple Dr.Web's server products are installed *together* on your server (for example, products for file servers, for mail servers, and for Internet gateways), you need to select *only the packages listed below for removal, in order to keep other server products—that will not be upgraded—fully functional (i.e. to keep the products for mail servers and for Internet gateways intact)*:

- `drweb-file-servers-doc`
- `drweb-samba-web`
- `drweb-smbspider`

Upgrading from the Repository



Note that you *cannot* upgrade Dr.Web for UNIX File Servers 6.0.2 to version 11.0 from the repository if several Dr.Web's version 6.0.2 server products are installed *together* on your server (for example, if the product for file servers, the product for mail servers, and the product for Internet gateways are installed). In this case, install the new version of Dr.Web for UNIX File Servers on a separate machine.

To upgrade your current version of Dr.Web for UNIX File Servers that was installed from the Doctor Web's repository, do one of the following, depending on the required type of packages:

• RPM packages (yum, dnf).

1. Change the used repository (from the package repository of your current version to the package repository 11.0).



You can find the name of the repository that stores 11.0 packets in the [Installing from the Repository](#) section. For details on how to change repositories, refer to help guides of the used operating system distribution.



2. Install the new version using the following command:

```
# yum update
```

or, if the manager **dnf** is used (similar to the **Fedora** OS of the version 22 and earlier):

```
# dnf update
```



If during the update of packages there is an error, uninstall and repeat the installation of the product. If necessary, see sections [Uninstalling the Product Installed from the Repository](#) and [Installing from the Repository](#) (items for the OS and the package manager that you are using).

- **DEB packages (apt-get).**

1. Change the used repository (from the package repository of your current version to the package repository 11.0).
2. Update the product using the following commands:

```
# apt-get update  
# apt-get dist-upgrade
```



Please note that for the **Ubuntu 14.04** (64-bit version) OS, the **apt-get dist-upgrade** command may fail. In this case use the **aptitude** package manager (to upgrade the product, issue the **aptitude dist-upgrade** command).

Key File Transfer

Regardless of the selected method to upgrade the product, the license [key file](#) which you already have (if you have one) will be automatically transferred and installed to the correct location required for the new version of the product.



If any problem occurs during the automatic installation of the key file, you can [install it manually](#).

If a valid license key file was lost, contact the [technical support](#).



Restoring Connection to the Central Protection Server

If it is possible, your connection to the central protection server will be restored automatically after the upgrade (if the product had been connected to a central protection server before the upgrade). In case the connection has not been automatically restored, then to reestablish the connection of the upgraded Dr.Web for UNIX File Servers to the anti-virus network, execute the following [command](#):

```
$ drweb-ctl esconnect <address> --Key <path to a file of the server public key>
```

In case there are any problems with the connection process, contact the administrator of your anti-virus network.

Removing the Product

Depending on the method that you used to install Dr.Web for UNIX File Servers, you can uninstall the product in one of the following ways:

1. [Starting the uninstaller](#) to uninstall the universal package.
2. [Uninstalling the packages](#) installed from the Doctor Web's repository with the help of the system's package manager.



Please note that after the uninstalling of Dr.Web for UNIX File Servers, you need to manually remove the symbolic link to the Dr.Web's **VFS** module from the Samba server's directory and to edit the configuration file of **Samba** (`smb.conf`) by removing the following line from each section that describes a shared directory:
`vfs objects = smb_spider` (where `smb_spider` is the name of the symbolic link to the Dr.Web's VFS SMB module).

Uninstalling the Universal Package

Dr.Web for UNIX File Servers that was installed from the [universal package](#) for UNIX systems can be uninstalled via the command line (if you are using a graphical desktop environment, you will need a terminal emulator for this option).



Note that the uninstallation tool uninstalls not only Dr.Web for UNIX File Servers, but also *all the other* Dr.Web products installed on your computer.

If any other Dr.Web products are installed on your computer, besides Dr.Web for UNIX File Servers, then, to delete only Dr.Web for UNIX File Servers, use the custom [components installation/removal](#) procedure, instead of running the automatic removal tool.



Uninstalling the Product via the Command Line

The uninstallation tool is started by the `uninst.sh` script, which is located in the `<opt_dir>/bin` directory (in **GNU/Linux** this is `/opt/drweb.com/bin`). Uninstallation procedure of Dr.Web for UNIX File Servers is described in section [Uninstalling from the Command Line](#).

You can also start the uninstallation tool in silent mode by executing the command

```
# env DRWEB_NON_INTERACTIVE=yes /opt/drweb.com/bin/uninst.sh
```

In this case, the uninstallation tool is run in silent mode and operates without the user interface (including program dialogs for command-line mode). Note that root privileges are required to start the uninstallation tool in silent mode. To elevate the privileges, you can use the **su** and **sudo** commands.

Uninstalling from Command Line

Once the command-line-based uninstallation program starts, an offer to remove the product is displayed in the command line.

1. To start the uninstalling, enter *Yes* or *Y* in response to the "Do you want to continue?" question. To exit the removal program, type *No* or *N*. In this case, removal will be canceled.
2. An automatic uninstallation procedure will be launched. During this procedure messages about the removal process will be displayed on the screen and logged into an uninstallation log.
3. Once the process is completed, the uninstallation program will automatically terminate.

Uninstalling the Product Installed from the Repository



All commands mentioned below for package uninstallation require superuser (**root**) privileges. To elevate the privileges, use the **su** command (to change the current user) or the **sudo** command (to execute the specified command with other user's privileges).

Debian, Mint, Ubuntu (apt)

To uninstall the root meta-package of Dr.Web for UNIX File Servers, enter the following command:

```
# apt-get remove drweb-file-servers
```

To uninstall all the installed Dr.Web packages, enter the following command (in certain operating systems, the '*' character must be escaped: '*'):

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



To automatically uninstall all packages that are no longer used, enter also the following command:

```
# apt-get autoremove
```



Please, note that uninstallation with the help of the **apt-get** command has the following special aspects:

1. The first mentioned variant of the command uninstalls only the `drweb-file-servers` package; any other packages that could have been automatically installed to resolve the dependencies of this package will remain in the system.
2. The second mentioned variant of the command uninstalls all the packages whose name starts with "drweb" (the standard name prefix for Dr.Web's products). Note that this command uninstalls all packages with this prefix, not only those of Dr.Web for UNIX File Servers.
3. The third mentioned variant of the command uninstalls all the packages that have been automatically installed to resolve dependencies of other packages and are no longer necessary (e.g., due to the uninstallation of the dependent packages). Note that this command uninstalls all packages that are not used, not only those of Dr.Web for UNIX File Servers.

You can also use alternative managers (for example, **Synaptic** or **aptitude**) to uninstall packages.

ALT Linux, PCLinuxOS (apt-rpm)

In this case, uninstalling of Dr.Web for UNIX File Servers is the same as on **Debian** and **Ubuntu** operating systems (see above).

You can also use alternative managers (for example, **Synaptic** or **aptitude**) to uninstall packages.

Mageia, OpenMandriva Lx (urpme)

To uninstall Dr.Web for UNIX File Servers, enter the following command:

```
# urpme drweb-file-servers
```

To automatically uninstall all packages that are no longer used, enter the following command:

```
# urpme --auto-orphans drweb-file-servers
```




Please, note that uninstallation with the help of the **urpme** command has the following special aspects:

1. The first mentioned variant of the command uninstalls only the `drweb-file-servers` package; any other packages that could have been automatically installed to resolve the dependencies of this package will remain in the system.
2. The second mentioned variant of the command uninstalls the `drweb-file-servers` package as well as all the packages that have been automatically installed to resolve dependencies of other packages and are no longer necessary (e.g., due to the uninstallation of the dependent packages). Note that this command uninstalls all packages that are not used, not only those of Dr.Web for UNIX File Servers.

You can also use alternative managers (for example, **rpmdrake**) to uninstall packages.

Red Hat Enterprise Linux, Fedora, CentOS (yum, dnf)

To uninstall all the installed Dr.Web packages, enter the following command (in certain operating systems, the '*' character must be escaped: '*'):

```
# yum remove drweb*
```

In the **Fedora** operating system, starting from version 22, it is recommended that instead of manager **yum** the manager **dnf** is used, for example:

```
# dnf remove drweb*
```



Please, note that uninstallation with the help of the **yum (dnf)** command has the following special aspects:

This variant of the command uninstalls all the packages whose name starts with "drweb" (the standard name prefix for Dr.Web's products). Note that this command uninstalls all packages with this prefix, not only those of Dr.Web for UNIX File Servers.

You can also use alternative managers (for example, **PackageKit** or **Yumex**) to uninstall packages.

SUSE Linux (zypper)

To uninstall Dr.Web for UNIX File Servers, enter the following command:

```
# zypper remove drweb-file-servers
```

To uninstall all the installed Dr.Web packages, enter the following command (in certain operating systems, the '*' character must be escaped: '*'):

```
# zypper remove drweb*
```



Please, note that uninstallation with the help of the **zypper** command has the following special aspects:

1. The first mentioned variant of the command uninstalls only the `drweb-file-servers` package; any other packages that could have been automatically installed to resolve the dependencies of this package will remain in the system.
2. The second mentioned variant of the command uninstalls all the packages whose name starts with "drweb" (the standard name prefix for Dr.Web's products). Note that this command uninstalls all packages with this prefix, not only those of Dr.Web for UNIX File Servers.

You can also use alternative managers (for example, **YaST**) to uninstall packages.

Additional Information

Product Packages and Files

Packages

Dr.Web for UNIX File Servers consists of the following packages:

Package	Contents
<code>drweb-bases</code>	Files of virus databases and of the anti-virus engine (Dr.Web Virus-Finding Engine)
<code>drweb-boost</code>	Boost libraries
<code>drweb-clamd</code>	Files of the Dr.Web ClamD component.
<code>drweb-cloudd</code>	Files of the Dr.Web CloudD component
<code>drweb-common</code>	<p>The main configuration file—<code>drweb.ini</code>, main libraries, documentation, and a hierarchy of the product's directories.</p> <p>During the installation of this package, a user named <code>drweb</code> and a group named <code>drweb</code> are created.</p>
<code>drweb-configd</code>	Files of the Dr.Web ConfigD and the Dr.Web Ctl components
<code>drweb-esagent</code>	Files of the Dr.Web ES Agent component.
<code>drweb-filecheck</code>	Files of the Dr.Web File Checker component.
<code>drweb-file-servers-doc</code>	PDF documentation for the product
<code>drweb-file-servers</code>	The root meta-package of the product



Package	Contents
drweb-httpd	Files of the Dr.Web HTTPD component and of the management web interface (a meta-package).
drweb-httpd-bin	Files of the Dr.Web HTTPD component.
drweb-httpd-webconsole	Files of the management web interface.
drweb-icu	Libraries for Unicode support and internationalization
drweb-libs	Main libraries of the product *)
drweb-netcheck	Files of the Dr.Web Network Checker component.
drweb-nss	Files of the SpIDer Guard for NSS component.
drweb-openssl	OpenSSL libraries
drweb-protobuf	Protobuf libraries
drweb-se	Files of the Dr.Web Scanning Engine component.
drweb-smbspider-daemon	Files of the SpIDer Guard for SMB component (the SMB monitoring daemon).
drweb-smbspider	Files of the SpIDer Guard for SMB component.
drweb-smbspider-modules	Files of the SpIDer Guard for SMB component (VFS SMB modules).
drweb-smbspider-modules-src	Files of the SpIDer Guard for SMB component (the source code of the VFS SMB module).
drweb-snmpd	Files of the Dr.Web SNMPD component.
drweb-spider	Files of the SpIDer Guard component.
drweb-spider-kmod	Files of the GNU/Linux kernel module for operation of SpIDer Guard in <i>LKM</i> mode
drweb-update	Files of the Dr.Web Updater component

*) The product's versions intended for 64-bit systems include two packages: `drweb-libs` and `drweb-libs32` that contain libraries for 64-bit and 32-bit components respectively.

In the section [Custom Component Installation and Uninstallation](#) there are typical component sets for a custom installation that provide solutions for typical tasks of the product.



Files

After the installation of Dr.Web for UNIX File Servers, its files are located in the `/opt`, `/etc`, and `/var` directories of the file system.

Structure of the used directories

Directory	Contents
<code><etc_dir>/</code>	The integrated configuration file and the product's license-key file.
<code>/etc/init.d/</code>	Managing startup script for configuration daemon Dr.Web ConfigD.
<code><opt_dir>/</code>	Main directory of the product.
<code>bin/</code>	Executable files of all the product's components (except for Dr.Web Virus-Finding Engine).
<code>include/</code>	Header files of the used libraries.
<code>lib/</code> <code>lib64/</code>	The libraries used for 32- and 64-bit platforms.
<code>man/</code>	System help files: man .
<code>share/</code>	Auxiliary product files.
<code>doc/</code>	Product documentation (<code>readme</code> files and the text of the license agreement).
<code>drweb-bases/</code>	Files of Dr.Web's virus databases (source files supplied during installation).
<code>drweb-spider-kmod/</code>	Files of kernel module for operation of SpIDer Guard (the source codes to the manual built).
<code>scripts/</code>	Auxiliary script files.
<code><var_dir>/</code>	Auxiliary and temporary files of the product.
<code>bases/</code>	Files of Dr.Web virus databases (the updated version).
<code>cache/</code>	Cache of updates.
<code>drl/</code>	Lists of servers that are used to get updates.
<code>lib/</code>	Anti-virus engine—Dr.Web Virus-Finding Engine—as a dynamic-link library (<code>drweb32.dll</code>) and the settings for working in the central protection mode.



Directory	Contents
update/	Directory for a temporary storage of updates during their download.

For details on conventions used for directories, refer to the [Introduction](#).

Custom Component Installation and Uninstallation

If necessary, you can choose to install or uninstall only certain product components by installing or uninstalling the respective [packages](#). Custom component installation or uninstallation should be performed the same way the product was installed.

To reinstall a component, you can uninstall it first and then install again.

Typical Component Kits for a Custom Installation

If it is required to install the product with the limited functionality, instead of installation of the product's root meta-package from the [repository](#) or from the [universal package](#), you can install only component packages that provide the required functionality. The packages required to resolve dependencies will be automatically installed. The table below displays component sets designed to resolve typical product tasks. In the column **Package for Installation**, there is a list of packages required for installation to obtain the specified component suite.

Custom Component Kit	Package for Installation	Will be Installed
Minimum kit for console scanning	drweb-filecheck	<ul style="list-style-type: none">• Dr.Web Ctl• Dr.Web ConfigD• Dr.Web Scanning Engine• Dr.Web File Checker• Dr.Web Updater• Virus database
Suite for file system monitoring of Linux	drweb-spider	<ul style="list-style-type: none">• Dr.Web Ctl• Dr.Web ConfigD• Dr.Web Scanning Engine• Dr.Web File Checker• Dr.Web Updater• SpIDer Guard• Virus database
Suite for monitoring of shared Samba directories	drweb-smbspider	<ul style="list-style-type: none">• Dr.Web Ctl• Dr.Web ConfigD• Dr.Web Scanning Engine• Dr.Web File Checker



Custom Component Kit	Package for Installation	Will be Installed
		<ul style="list-style-type: none">• Dr.Web Updater• SpIDer Guard for SMB• Virus database
Suite for monitoring of NSS volumes	drweb-nss	<ul style="list-style-type: none">• Dr.Web Ctl• Dr.Web ConfigD• Dr.Web Scanning Engine• Dr.Web File Checker• Dr.Web Updater• SpIDer Guard for NSS• Virus database
Suite for the emulation ClamAV (clamd)	drweb-clamd	<ul style="list-style-type: none">• Dr.Web Ctl• Dr.Web ConfigD• Dr.Web Scanning Engine• Dr.Web File Checker• Dr.Web Network Checker• Dr.Web Updater• Dr.Web ClamD• Virus database

1. Installation and Uninstallation of Product Components Installed from Repository

If your product is installed from repository, for custom component installation or uninstallation use the respective command of the package manager, used in your OS. For example:

1. To uninstall Dr.Web ClamD (package `drweb-clamd`) from the product installed on OS **CentOS**, use the command:

```
# yum remove drweb-clamd
```

2. To additionally install Dr.Web ClamD (package `drweb-clamd`) to the product installed on OS **Ubuntu Linux**, use the command:

```
# apt-get install drweb-clamd
```

If necessary, use a help file of the package manager used in your OS.



The Dr.Web for UNIX File Servers anti-virus engine uses a 32-bit architecture *x86*; in 64-bit systems **Debian, Mint, Ubuntu** (for platforms *x86-64, x64, amd64*), a permission could be required for installation of packages for the platform *x86*. It could be obtained via the following command:

```
# dpkg --add-architecture i386
```

2. Installation and Uninstallation of Product Components Installed from the Universal Package

If the product is installed from the universal package and you want to additionally install or reinstall a package of a component, you will need an installation file (with the *.run* extension), from which the product was installed. In case you did not save this file, download it from the Doctor Web's official website.

Unpacking the Installation File

When you launch the *.run* file, you can also specify the following command-line parameters:

--noexec—unpack the product's installation files instead of starting the installation process. The files will be placed to the directory that is specified in the *TMPDIR* environment variable (usually, */tmp*).

--keep—do not delete the product installation files and the installation log automatically after the installation completes.

--target <directory>—unpack the product's installation files to the specified *<directory>*.

For a full list of command-line parameters that can be specified for the launching of the *.run* file, enter the following command:

```
$ ./<file_name>.run --help
```

For a custom installation, you need to use the unpacked installation files. If there is no directory containing these files, you should first unpack them. To do that, enter the following command:

```
$ ./<file_name>.run --noexec --target <directory>
```

After the command is executed, a nested directory named *<file_name>* will appear in the directory *<directory>*.



Custom Installation of the Components

Installation RUN file contains packages of all components of Dr.Web for UNIX File Servers (in the RPM format) and supporting files. Package files of each component have the following structure:

```
<component_name>_<version>~linux_<platform>.rpm
```

where *<version>* is a string that contains the version and time of the product's release, and *<platform>* is a platform for which the product is intended. Names of all the packages containing the components of Dr.Web for UNIX File Servers start with the "drweb" prefix.

Package manager is enabled for the installation of packages to the installation kit. For the custom installation, you should use a service script `installpkg.sh`. To do that, first, you need to unpack the contents of the installation package to a directory.



To install packages, superuser permissions are required (i.e. privileges of the *root* user). To elevate your privileges, use the **su** command for changing the current user or the **sudo** command to execute the specified command with the privileges of another user.

To start installation or reinstallation of a component package, go to the directory which contains the unpacked installation kit, and execute the following command via the console (or via a console emulator—terminal for the graphical mode):

```
# ./scripts/installpkg.sh <package_name>
```

For example:

```
# ./scripts/installpkg.sh drweb-clamd
```

If it is necessary to start the full product installation, launch the automatic installation script. To do that, use the following command:

```
$ ./install.sh
```

Besides that, you can install all product packages (to install the missing or accidentally deleted components as well) by launching the installation of the root meta-package of the product:

```
# ./scripts/installpkg.sh drweb-file-servers
```




Custom Uninstallation of the Components

For the custom uninstallation of a component, use the appropriate uninstallation command of the package manager of your OS if your OS uses the RPM format of packages:

- In **Red Hat Enterprise Linux** and **CentOS**, use the command **yum remove <package_name>**
- In **Fedora**, use the command **yum remove <package_name>** or **dnf remove <package_name>**
- In **SUSE Linux**, use the command **zypper remove <package_name>**
- In **Mageia**, **OpenMandriva Lx**, use the command **urpme <package_name>**
- In **Alt Linux** and **PCLinuxOS**, use the command **apt-get remove <package_name>**.

For example (for **Red Hat Enterprise Linux**):

```
# yum remove drweb-clamd
```

If your OS uses DEB packages (also if you use **MSVS 3.0 OS**), or if there is no package manager in your system (**FreeBSD**, **Solaris**), for the custom uninstallation, you should use the package manager **zypper**, which is automatically installed within the product installation. To do that, go to the directory **<opt_dir>/bin** (for **GNU/Linux**—**/opt/drweb.com/bin**) and execute the following command:

```
# ./zypper remove <package_name>
```

For example:

```
# ./zypper remove drweb-clamd
```

If it is necessary to start the full product uninstalling, launch the automatic uninstallation script. To do that, use the following command:

```
# ./uninst.sh
```

To reinstall a component, you can uninstall it first and then install by launching the custom or full installation from the installation kit.



Configuring Security Subsystems

Presence of the **SELinux** enhanced security subsystem in the OS (as well as the use of mandatory access control systems, such as **PARSEC** (as opposed to the classical discretionary model used by UNIX) causes problems in the work of Dr.Web for UNIX File Servers when its default settings are used. To ensure correct operation of Dr.Web for UNIX File Servers in this case, it is necessary to make additional changes to the settings of the security subsystem and/or to the settings of Dr.Web for UNIX File Servers.

This section discusses the settings that ensure correct operation of Dr.Web for UNIX File Servers in the following cases:

- [Configuring SELinux](#) Security Policies.
- [Setting up the permissions](#) of the **PARSEC** mandatory access control system (the **Astra Linux** OS)



Configuring the permissions of the **PARSEC** mandatory access control system for Dr.Web for UNIX File Servers will allow the components of Dr.Web for UNIX File Servers to bypass the restrictions of the set security policies and to get access to the files that belong to different privilege levels.

Note that even if you have not configured the permissions of the **PARSEC** mandatory access control system for Dr.Web for UNIX File Servers, you still will be able to launch file scanning directly from the [command line](#). To do this, use the **drweb-ctl**[command](#) in the autonomous mode, by specifying the `--Autonomous` option in the command call. When scanning is launched this way, it is possible to scan only those files that can be accessed with the privileges not exceeding those of the user who launched the scanning. This mode has several features:

- To launch the autonomous copy you need the valid [key file](#), the work with [central protection](#) server is not supported (it is possible to [install](#) the key file, exported from central protection server). Herewith, even if Dr.Web for UNIX File Servers is connected to central protection server, the autonomous copy do not send to it any notifications on threats, detected during the work in autonomous mode.
- All additional components that support the functioning of the autonomous copy, will be launched under the current user and will work with specially generated configuration file.
- All the used temporary files and UNIX sockets are created only in the directory with an unique name, which is created when the autonomous copy is launched. The unique temporary directory is created in the system directory for temporary files (path to this directory is available in the `TMPDIR` environment variable).
- All the required paths to virus databases, anti-virus engine and executable files used during scanning are defined by default or retrieved from the special environment variables.
- The number of the autonomous copies working simultaneously is not limited.
- When the autonomous copy is terminated, the set of supporting components also terminates.



Configuring SELinux Security Policies

If your **GNU/Linux** distribution includes **SELinux** (*Security-Enhanced Linux*), you may need to configure **SELinux's security policies to get the servicing components of the Dr.Web product (such as the [scanning engine](#))** to operate correctly after the installation of the Dr.Web product.

1. Universal Package Installation Issues

If **SELinux** is enabled, installation from the [installation file](#) (.run) can fail because an attempt to create the *drweb* user, under which Dr.Web for UNIX File Servers components operate, can be blocked.

In case of failure, check the **SELinux** operation mode with the **getenforce** command. The command outputs one of the following:

- **Permissive**—protection is active but a permissive strategy is used: actions that violate the security policy are not denied but information on the actions is logged.
- **Enforced**—protection is active and restrictive strategy is used: actions that violate security policies are blocked and information on the actions is logged.
- **Disabled**—**SELinux** is installed but not active.

If **SELinux** is operating in *Enforced* mode, change it to *Permissive*. For that purpose, use the following command:

```
# setenforce 0
```

This command (until the next reboot) enables *Permissive* mode for **SELinux**.



Note that regardless of the operation mode enabled with the **setenforce** command, after the restart of the operating system, **SELinux** returns to the safe operation mode specified in its settings (file with **SELinux** settings usually resides in the `/etc/selinux` directory).

After the successful product installation, enable *Enforced* mode again before starting the product. For that, use the following command:

```
# setenforce 1
```

2. Problems with the Product's Operation

In some cases, when **SELinux** is enabled, some Dr.Web for UNIX File Servers's components (for example, **drweb-se** and **drweb-filecheck**) cannot start. If so, object scanning and file system monitoring become unavailable. In this case errors *119* and *120* can appear in the system log **syslog** (normally located in the `/var/log/` directory).



Messages on 119 and 120 errors can also indicate an attempt to start Dr.Web for UNIX File Servers on 64-bit version of the operating system if the 32-bit application support library is missing (see [System Requirements](#)).

When the **SELinux** security system denies access, such an event is logged. In general, when the **audit** daemon is used on the system, the log of the audit is stored in the `/var/log/audit/audit.log` file. Otherwise, messages about blocked operations are saved to the general log file (`/var/log/messages` or `/var/log/syslog`).

If the scanning components of the product do not function because they are blocked by **SELinux**, you will need to compile special *security policies* for them.



Note that certain **Linux** distributions do not feature the utilities mentioned below. If so, you may need to install additional packages with the utilities.

Configuring SELinux Security Policies:

1. Create a new file with the **SELinux** policy source code (a `.te` file). This file defines restrictions related to the described policy module. The policy's source code can be created in one of the following ways:

- 1) Using the **audit2allow** utility, which is the simplest method. The utility generates permissive rules from messages on access denial in system log files. You can set to search messages automatically or specify a path to the log file manually.

Note that you can use this method only if Dr.Web for UNIX File Servers's components have violated **SELinux** security policies and these events are registered in the audit log file. If not, wait for such an incident to occur or force-create permissive policies by using the **policygentool** utility (see below).



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

Example of using **audit2allow**:

```
# grep drweb-se.real /var/log/audit/audit.log | audit2allow -M drweb-se
```

In the given example, the **audit2allow** utility performs a search in the `/var/log/audit/audit.log` file to find access denial messages for the **drweb-se** component.

The following two files are created: policy source file `drweb-se.te` and the `drweb-se.pp` policy module ready to install.

If no security violation incidents are found in the system audit log, the utility returns an error message.



In most cases, you do not need to modify the policy file created by the **audit2allow** utility. Thus, it is recommended to go to [step 4](#) for installation of the `drweb-se.pp` policy module. Note that the **audit2allow** utility outputs invocation of the **semodule** command. By copying the output to the command line and executing it, you complete [step 4](#). Go to [step 2](#) only if you want to modify security policies which were automatically generated for Dr.Web for UNIX File Servers components.

- 2) Using the **policygentool** utility. For that purpose, specify the name of the component that you want to be treated differently and the full path to its executable file.



Note that the **policygentool** utility, included in the `selinux-policy` package for **RedHat Enterprise Linux** and **CentOS Linux** OS, may not function correctly. If so, use the **audit2allow** utility.

Example of policy creation using **policygentool**:

- For the **drweb-se** component:

```
# policygentool drweb-se /opt/drweb.com/bin/drweb-se.real
```

- For the **drweb-filecheck** component:

```
# policygentool drweb-filecheck /opt/drweb.com/bin/drweb-filecheck.real
```

You will be prompted to specify several general properties for created the domain. After that, three files that determine the policy will be created (for each of the components):

`<module_name>.te`, `<module_name>.fc` and `<module_name>.if`.

2. If required, edit the generated policy source file `<module_name>.te` and then use the **checkmodule** utility to create a binary representation (a `.mod` file) of this source file of the local policy.



Note that to ensure successful execution of the command, the `checkpolicy` package must be installed in the system.

Example usage

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

3. Create a policy module for installation (a `.pp` file) with the help of the **semodule_package** utility.

Example:

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

4. To install the created policy module, use the **semodule** utility.

Example:

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



For details on **SELinux** operation and configuration, refer to documentation for the used **Linux distribution**.

Configuring the Permissions of PARSEC (Astra Linux)

In operating systems equipped with the **PARSEC** security subsystem (mandate access control system), due to the variation in privilege levels required to access different files, the [SpIDer Guard](#) monitor working in its default mode (**Mode** = **AUTO**) cannot intercept file-access events for any files whose required access privilege level is higher than the one with which SpIDer Guard was launched. Moreover, if the user works at any privilege level other than the zeroth, the command-line-based management tool [Dr.Web Ctl](#) for Dr.Web for UNIX File Servers cannot interact with the SpIDer Guard monitor and with the [Dr.Web ConfigD](#) configuration daemon, if they work at a different privilege level; access to the [consolidated quarantine](#) may also become unavailable.

To configure permissions, superuser permissions are required (i.e. privileges of the *root* user). To elevate your privileges, use the **su** command for changing the current user or the **sudo** command to execute the specified command with the privileges of another user.

Configuring SpIDer Guard to intercept attempts to access files with any privilege level

To give the SpIDer Guard file monitor an ability to detect attempted access, when any files that have any level of access privileges are accessed, it is necessary to switch SpIDer Guard into an *LKM* operating mode (this will use a special loadable kernel module for the **Linux** kernel; this module is supplied together with Dr.Web for UNIX File Servers).

To switch SpIDer Guard into the *LKM* operating mode, execute the following [command](#):

```
# drweb-ctl cfset LinuxSpider.Mode LKM
```

To get additional information, use the following command:

```
$ man drweb-spider
```

Configuring the Correct Launch of Dr.Web for UNIX File Servers at Any Privilege Level

In order for all the components of Dr.Web for UNIX File Servers to be able to correctly interact with each other when they are launched with different privilege levels, modify the script that launches the Dr.Web ConfigD configuration daemon (**drweb-configd**):

1. Log into the system using the privilege level zero
2. Open the `/etc/init.d/drweb-configd` script file in any text editor (root privileges are required).
3. In this file find the definition of the `start_daemon` function and replace the line:



```
"$DAEMON" -d -p "$PIDFILE" >/dev/null 2>&1
```

with the line:

```
execaps -c 0x100 -- "$DAEMON" -d -p "$PIDFILE" >/dev/null 2>&1
```

4. In some OSes, (for example, **Astra Linux SE 1.3**), an additional indication of component launch dependence from the **PARSEC** subsystem could be required. In this case, it is also necessary to modify a string in the file:

```
# Required-Start: $local_fs $network
```

Change this string in the following way:

```
# Required-Start: $local_fs $network parsec
```

5. Save the file and reboot the operating system.



Getting Started

1. To start using the installed Dr.Web for UNIX File Servers, you need to [activate](#) it by obtaining and installing a [key file](#).
2. Further [scanning of the operability of the product](#) is recommended.
3. Integrate Dr.Web for UNIX File Servers with the required file services (please, see the [instruction manual for Samba](#), please, see the [instruction manual for NSS](#)).
4. If necessary, [configure the monitoring parameters](#) for the **Linux** file system objects outside the NSS volumes and outside the Samba shared directories.
5. Check what components are running and enable additional components, which are disabled by default, if you need them for the protection of your server (for example, the [SpIDer Guard](#), [Dr.Web ClamD](#) or [Dr.Web SNMPD](#) component, depending on the distribution). Note that you may also need to perform other actions apart from enabling the additional components, for example, you may need to adjust their default configuration. To view the list of installed and running components and their settings, use one of the following:
 - The [command-line-based management tool](#)—Dr.Web Ctl (use the **drweb-ctl** appinfo, **drweb-ctl** cfshow and **drweb-ctl** cfset commands).
 - The management [web interface](#) of Dr.Web for UNIX File Servers (by default, you can access it via a web browser at <https://127.0.0.1:4443/>).

Registration and Activation of the Product

Purchasing and Registering License

After a license is purchased, updates to product components and virus databases are regularly downloaded from Doctor Web update servers. Moreover, if the customer encountered any issue when installing or using the purchased product, they can take advantage of technical support service provided by Doctor Web or its partners.

You can purchase any Dr.Web product as well as obtain a product serial number either on the [online store](#) or from our [partners](#). For details on license periods and license types, visit the Doctor Web official website at <https://www.drweb.com/>.

License registration is required to prove that you are a legal user of Dr.Web for UNIX File Servers and to activate the functions of the anti-virus, including the regular updates of virus databases. It is recommended that you register the product and activate the license once the installation completes. A purchased license can be activated on the Doctor Web's official website at <https://products.drweb.com/register/>.

During activation, it is required to enter the serial number of the purchased license. The serial number is supplied with the product or via email when purchasing or renewing the license online.



If you have used the product in the past, you may be eligible for a 150-day extension to your new license. To enable the bonus, enter your registered serial number or provide the license key file.

If you have several licenses for using Dr.Web for UNIX File Servers on several servers, but choose to use the product only on one server, you can specify this and, hence, license validity period will be automatically extended.

Obtaining Demo License

A demo period for your copy of the product can be obtained on the Doctor Web official website at <https://download.drweb.com/demoreq/biz/>. After you select the product and fill the registration form, you will receive an email with a serial number or key file for Dr.Web for UNIX File Servers activation.



Another demo period for the same computer can be obtained after a certain time period.

You can use the license [command](#) of the [Dr.Web Ctl](#) (**drweb-ctl**) command-line tool, which allows to get a demo key file or a licensed key file for a serial number of a registered license automatically.

Key File Installation

The key file is a special file stored on the local computer. It corresponds to the purchased license or activated demo period for Dr.Web for UNIX File Servers. The file contains information on the provided license or demo period and regulates usage rights in accordance with it.



During Dr.Web for UNIX File Servers operation, the key file must be located in the default `<etc_dir>` directory (`/etc/opt/drweb.com` for **Linux**) under the name `drweb32.key`.

Components of the product regularly check whether the key file is available and valid. The key file is digitally signed to prevent its editing. So, the edited key file becomes invalid. It is recommended that you do not open your key file in text editors in order to avoid its accidental invalidation.

If no valid key file (license or demo) is found, or if the license is expired, operation of the anti-virus components is blocked until a valid key file is installed.

It is recommended that you keep the license key file until it expires, and use it to reinstall the product or install it on a different computer. In this case, you must use the same product serial number and customer data that you provided during the registration.



Dr.Web key files are usually packed in a ZIP archive if sent via email. The archive with a key file is named `agent.zip` (note that if there is *several* archives in an email message, you should use only `agent.zip`). Before installing a key file, unpack it using any suitable tool and extract a key file to any directory (for example, to your home directory or to a USB flash drive).

If you have a key file corresponding to the valid license for the product (for example, if you obtained the key file by email or if you want to use Dr.Web for UNIX File Servers on another server), you can activate the product by specifying the path to the key file. For that, do the following:

1. Unpack the key file if archived
2. Do one of the following:
 - Copy the key file to the `<etc_dir>` directory and rename the file to `drweb32.key` if necessary.
 - In the Dr.Web for UNIX File Servers [configuration file](#) specify the key file path as the **KeyPath** parameter value.
3. Uninstall Dr.Web for UNIX File Servers by entering the following [command](#):

```
# drweb-ctl reload
```

to apply all changes.

You can also use the following [command](#):

```
# drweb-ctl cfset Root.KeyPath <path to the key file>
```

In this case, restart of Dr.Web for UNIX File Servers is not required. The key file will not be copied to the `<etc_dir>` directory and will remain in its original location.



For details on conventions for `<opt_dir>`, `<etc_dir>`, and `<var_dir>`, refer to the [Introduction](#).

If the key file is not copied to the `<etc_dir>` directory, the user becomes responsible for ensuring that the file is protected from corruption or deletion. This installation method is not recommended as the key file can be accidentally deleted from the system (for example, if the directory, where the key file resides, is periodically cleaned up). Remember that if a key file is lost, you can request the support for a new one, but the number of such requests is limited.



Subsequent Registration

If a key file is lost but the existing license is not expired, you must register again by providing the personal data you specified during the previous registration. You may use a different email address. In this case, the license key file will be sent to the newly specified address.

The number of times you can request a key file is limited. One serial number can be registered *no more than 25 times*. If requests in excess of that number are sent, no key file will be delivered. To receive a lost key file, contact Doctor Web [technical support](#), describe your problem in detail, and state personal data you entered upon serial number registration. The license key file will be sent by email.

After the key file is sent to you by email, you need to [install](#) it manually.

Testing the Operation of the Product

The *EICAR (European Institute for Computer Anti-Virus Research)* Test helps testing performance of anti-virus programs that detect viruses using signatures. This test was designed specially so that users could test reaction of newly-installed anti-virus tools to detection of viruses without compromising security of their computers.

Although the *EICAR* test is not actually a virus, it is treated by the majority of anti-viruses as if it were a virus. On detection of this “virus”, Dr.Web anti-virus products report the following: **EICAR Test File (NOT a Virus!)**. Other anti-virus tools alert users in a similar way. The **EICAR** test file is a 68-byte COM-file for **MS DOS/MS Windows** that outputs the following line on the console when executed:

```
EICAR-STANDARD-ANTIVIRUS-TEST-FILE!
```

The *EICAR* test contains the following character string only:

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

To create your own test file with the “virus”, you may create a new file with the line mentioned above.

If Dr.Web for UNIX File Servers operates correctly, the test file is detected during a file system scan regardless of the scan type, and the user is notified on the detected threat: **EICAR Test File (NOT a Virus!)**.

An example of a command that checks operation of the program by means of **EICAR** test from the command line:

```
$ tail <opt_dir>/share/doc/drweb-common/readme.eicar | grep X5O > testfile &&  
drweb-ctl rawscan testfile && rm testfile
```



From the file `<opt_dir>/share/doc/drweb-common/readme.eicar` (supplied with the product), this command retrieves a string that represent a body of the **EICAR** test file, then writes it to the file `testfile` located in the current catalog, checks the received file, and removes the created file.



The above-mentioned test requires write access to the current catalog. In addition, make sure that it does not contain a file named `testfile` (if necessary, change the file name in the command).

For details on conventions for `<opt_dir>`, `<etc_dir>`, and `<var_dir>`, refer to the [Introduction](#).

If a test virus is detected, the following message is displayed:

```
<path to the current directory>/testfile - infected with EICAR Test File (NOT a Virus!)
```

If an error occurs during the test, refer to the description of known errors (see [Appendix F. Known Errors](#)).



If SpIDer Guard is enabled, a malicious file can be immediately removed or quarantined (depending on the configuration of the component). In this case, the command **rm** will inform that the file is missing, which implies that the monitor operates in normal mode.

File System Monitoring Setting

To configure the **Linux** file system monitoring with the SpIDer Guard monitor, specify values for parameters that are in the `[LinuxSpider]` [settings](#) section of the configuration file:

- Enable the monitor by setting the **Start** value to `Yes`;
- Specify the mode of operation with the file system monitor in the **Mode** parameter (it is recommended that you use the `Auto` value).
- If required, in the **ExcludedProc** parameter, list the paths to the executable files of trusted applications, i.e. applications whose access to files will not be controlled by the monitor;
- In the **IncludedPath** and **ExcludedPath** parameters specify the monitoring scope and the exclusion scope in the file system (the lists of paths to objects monitored and excluded from monitoring). For example, if some paths are controlled by the file server **Samba** or are the **NSS** volumes, these paths should be added to the exclusion scope in order to avoid conflicts during the scanning by different monitors;
- Specify the parameters of file scanning and monitor's reaction on detection of various types of threats .

After all settings are adjusted, restart Dr.Web for UNIX File Servers (use the [command](#) **drwebctl reload**). You can also restart the configuration daemon Dr.Web ConfigD (use the



service drweb-configd restart command).

Integration with Samba File Server



The SpIDer Guard for SMB monitor uses a special VFS SMB module for the integration with the **Samba** server. With SpIDer Guard for SMB, several versions of this module are supplied. They are built for various versions of **Samba**. However, the supplied versions of the VFS SMB module may be incompatible with the version of **Samba** installed on your file server. It may occur, for example, if your **Samba** server uses the `CLUSTER_SUPPORT` option.

If VFS SMB modules are incompatible with your **Samba** server, *the corresponding message is shown during the product installation*. In this case, build the VFS SMB module for your **Samba** server manually (including the compatibility with the `CLUSTER_SUPPORT` option if necessary).

The procedure of building the VFS SMB module from the supplied source code files is described in the [Building the VFS SMB Module](#) section.

Steps for integration with Samba

To integrate SpIDer Guard for SMB with the **Samba** file server, do the following:

1. In the directory from which **Samba** loads its VFS SMB modules (the default directory in **Linux** is `/usr/lib/samba/vfs`), create a symbolic link `smb_spider.so` that points to the Dr.Web-supplied VFS SMB module that corresponds to your version of **Samba**.

The VFS SMB modules that are supplied by Dr.Web reside in the directory that holds the libraries of:

- `<opt_dir>/lib/samba` – for the 32-bit platform.
- `<opt_dir>/lib64/samba` – for the 64-bit platform.

The modules have file names that look as follows: `libsmb_spider.so.<ver>`, where `<ver>` is the version of the **Samba** server for which the module is intended.

For instance: `/opt/drweb.com/lib/samba/libsmb_spider.so.3.6.0` is a VFS SMB module for the **Samba** server version 3.6.0 that runs on a 32-bit platform in the **Linux** environment.

2. In the configuration file of the **Samba** `server-smb.conf` (the default location in **Linux** is `/etc/samba`)—create sections for the shared directories. Such a section should look like:

```
[<share_name>]
comment = <any comment>
path = <path to the protected directory>
vfs objects = smb_spider
writeable = yes
browseable = yes
guest ok = yes
public = yes
```



where the *<share name>* is any name for the shared resource and *<any comment>* is an arbitrary line with a comment (optional). The object's name specified in `vfs objects` must be the same as the name of the symbolic link (here `smb_spider`).

After that, this directory will be monitored by SpIDer Guard for SMB. Interaction between SpIDer Guard for SMB and the VFS SMB module will be performed via a UNIX socket `/<samba chroot path>/var/run/.com.drweb.smb_spider_vfs`. By default, the path to this UNIX socket is specified in the SpIDer Guard for SMB settings and in the settings of the VFS SMB module.

3. If you need to change the path to the socket, specify the new path both in the [settings](#) of SpIDer Guard for SMB (the **SmbSocketPath** parameter) and in the configuration file of **Samba**—`smb.conf`. For that, add the following line to the `[<share name>]` section:

```
smb_spider:socket = <path to socket>
```

where *<path to socket>* must be an absolute path to the UNIX socket, relative to the root directory that was set for the **Samba** server by using **chroot** (`<samba chroot path>`).

4. If required, you can use **ExcludedPath** and **IncludedPath** parameters to exclude paths to objects located in the protected shared directories or to include them in SpIDer Guard for SMB checks. You can specify paths to directories or paths to files. If you specify a directory, all content of the directory is skipped or scanned. Note that the **IncludedPath** parameter takes precedence over the **ExcludedPath** parameter, that is, if the same object (file or directory) is included in both parameter values, this object will be checked.
5. If you need to specify personal scanning settings for this shared directory (different from the default settings used for all shared directories), create a tag identifier for the VFS SMB module that controls this directory:

```
smb_spider:tag = <share name>
```

Then specify personal settings for the protection of this shared directory in SpIDer Guard for SMB settings as a [separate section](#) `separate section`
`[SMBSpider.Share.<share name>]`.

To add a new section identified by a *<share name>* tag with the help of the Dr.Web Ctl command-line tool, it is necessary to use the following command: **drweb-ctl** `cfset SmbSpider.Share.<share name>.<parameter> <value>`, for example:

```
# drweb-ctl cfset SmbSpider.Share.BuhFiles.OnAdware Quarantine
```

This command adds the `[SMBSpider.Share.BuhFiles]` section into the configuration file. This added section will contain all the available parameters adjusting the scanning of this shared directory, at that, values for all parameters, except the **OnAdware** parameter specified in the command, will coincide with parameter values from the general `[SMBSpider]` [section](#).

6. Enable SpIDer Guard for SMB by setting the **Start** value to **Yes**.

After all settings are adjusted, restart Dr.Web for UNIX File Servers (use the [command](#) **drweb-ctl** `reload`). You can also restart the configuration daemon Dr.Web ConfigD (use the **service** `drweb-configd restart` command).



To avoid conflicts between SpIDer Guard for SMB and SpIDer Guard, which may occur in the process of scanning the files located in the shared directories of **Samba**, it is recommended that you additionally configure SpIDer Guard by performing one of the following actions:

- add **Samba** shared directories to the exclusion scope (specify these directories in the **ExcludedPath** parameter);
- add the **Samba** process (**smbd**) to the list of ignored processes (specify **smbd** in the **ExcludedProc** parameter).

Scripts to Support Integration

For convenient integration of SpIDer Guard for SMB with the **Samba** file server, Dr.Web for UNIX File Servers is supplied with special shell scripts for integration setting. They are located in:

`<opt_dir>/share/drweb-smbspider-modules:`

The script file	Function
<code>drweb_smbspider_configure.sh</code>	The script that allows you to modify the Samba configuration file— <code>smb.conf</code> —via a dialog window (the script sets up monitoring and protection for the shared directories described in the Samba configuration file).
<code>update-links.sh</code>	The script that creates/updates a link to the Dr.Web VFS SMB module in the Samba directory.
<code>vfs-versions.sh</code>	Auxiliary script that determines the version of a Dr.Web VFS SMB module; it is used by the <code>update-links.sh</code> script

The `update-links.sh` shell script automatically runs when product is installed. If required, you can also run it manually afterwards. The `drweb_smbspider_configure.sh` script runs automatically if Dr.Web for UNIX File Servers is installed from a universal package; it is recommended that you run this script manually after installing the Dr.Web for UNIX File Servers product if you installed the product from a repository, or if you declined the offer to run the script during the installation. It is allowed to run this script repeatedly, when it is necessary to enable or disable monitoring of certain shared directories. The script also saves the original (unmodified) copy of the **Samba** `smb.conf`—adding a `.drwebsave` extension to its name.

Integration with NSS Volumes

To integrate Dr.Web for UNIX File Servers with **Novell Storage Services** volumes, specify values for parameters that are in the [NSS] [section](#) of the configuration file:

- In the **NssVolumesMountDir** parameter, specify the path to the file system directory, where **NSS** volumes are mounted (the default directory is `/media/nss`).



- In the **ProtectedVolumes** parameter, specify the names of **NSS** file system volumes located in the directory indicated in the **NssVolumesMountDir** parameter, and subject to protection. If the parameter is empty, all volumes in the directory indicated in the **NssVolumesMountDir** parameter are subject to protection.
- If required, you can use **ExcludedPath** and **IncludedPath** parameters to exclude paths to objects located in the protected volumes or to include them in SpIDer Guard for NSS checks. You can specify paths to directories or paths to files. If you specify a directory, all content of the directory is skipped or scanned. Note that the **IncludedPath** parameter takes precedence over the **ExcludedPath** parameter, that is, if the same object (file or directory) is included in both parameter values, this object will be checked. Parameters **IncludedPath** and **ExcludedPath** allow to use file masks (*wildcards*). Case sensibility of paths indicated in these parameters is defined by the **NSS** settings.
- Enable SpIDer Guard for NSS by setting the **Start** parameter to the value **Yes**.

After all settings are adjusted, restart Dr.Web for UNIX File Servers (use the [command](#) **drwebctl reload**). You can also restart the configuration daemon Dr.Web ConfigD (use the **service** **drweb-configd restart** command).



Brief Instructions

How to Connect Dr.Web for UNIX File Servers to a Samba Server

Follow the instructions provided in the [Integration with Samba File Server](#) section.

How to Connect Dr.Web for UNIX File Servers to Novell Storage Services

Follow the instructions provided in the [Integration with NSS Volumes](#) section.

How to Configure GNU/Linux File System Monitoring

Follow the instructions provided in the [File System Monitoring Setting](#) section.

How to Restart Dr.Web for UNIX File Servers

To restart the product when it is already running, you can also use the script that controls the Dr.Web ConfigD configuration daemon. Startup, stop, or restart of the daemon cause respectively the startup, stop or restart of Dr.Web for UNIX File Servers.

The default directory of the shell script that controls the operation of Dr.Web ConfigD is `/etc/init.d`. The name of the script is `drweb-configd`. It has the following parameters:

Parameter	Description
<code>start</code>	Instructs to start Dr.Web ConfigD if it is not running. When Dr.Web ConfigD starts, Dr.Web ConfigD launches all the required modules of Dr.Web for UNIX File Servers.
<code>stop</code>	Instructs to shut down Dr.Web ConfigD if it is running. When Dr.Web ConfigD is shutting down, Dr.Web ConfigD also shuts down all the components of Dr.Web for UNIX File Servers.
<code>restart</code>	Instructs to restart (shut down and then start) Dr.Web ConfigD. Dr.Web ConfigD shuts down and then starts all the modules of Dr.Web for UNIX File Servers. If Dr.Web ConfigD is not running, the parameter has the same effect as <code>start</code> .
<code>condrestart</code>	Instructs to restart Dr.Web ConfigD only if it is running.
<code>reload</code>	Instructs to send a HUP signal to Dr.Web ConfigD if the component is running. Dr.Web ConfigD forwards this signal to all the components of Dr.Web for UNIX File Servers. The parameter is used to make all components reread their configuration.
<code>status</code>	Instructs to output the current state of Dr.Web ConfigD to the console.



To restart Dr.Web for UNIX File Servers (or start it, if it is not running), use the following command:

```
# /etc/init.d/drweb-configd restart
```

How to Connect to the Central Protection Server

1. Obtain the address of the central protection server and the file of its public key from your anti-virus network administrator. You may also need additional parameters, such as an identifier and password for your workstation or identifiers of the main group and tariff group.
2. Use the `esconnect` [command](#) of the [Dr.Web Ctl](#) command-line tool provided in Dr.Web for UNIX File Servers.

For connection it is required to use the option `--Key`, by specifying the path to the public key file of the server. You can additionally enter the identifier of your host (the ID of your "workstation", if we use the terminology used by the central protection server) and a password for authentication on the central protection server by using the `--Login` and `--Password` parameters. In this case, connection to the server will be established only if you specify a correct identifier-password pair. If the parameters are not specified, connection to the server will be established only if it is approved on the server (automatically or by the administrator of the anti-virus network, depending on the server's settings).

Moreover, you can use the `--Newbie` option (connect as a new user). If this mode is allowed on the server, then after this connection is approved, the server automatically generates a unique identifier/password pair, which will be further used for connection of this agent to the server. Note that in this mode the central protection server generates a new account for the host even if this host already has another account on the server.

A standard example of the command instructing Dr.Web for UNIX File Servers to connect to the central protection server:

```
# drweb-ctl esconnect <server address> --Key <path to the server's public key file>
```

After establishing a connection to the central protection server, the product will operate in central protection mode or in mobile mode, depending on the permissions set on the server and the value of the [configuration parameter](#) **MobileMode** of the Dr.Web ES Agent component. To allow unconditional use of mobile mode, set the parameter's value to `On`. For operation in central protection mode, set the parameter's value to `Off`.

A standard example of the command instructing Dr.Web for UNIX File Servers that is connected to a central protection server to switch into mobile mode is as follows:

```
# drweb-ctl cfset ESAgent.MobileMode On
```



If the used central protection server does not support or does not allow mobile mode, adjusting the **MobileMode** parameter cannot switch operation of Dr.Web for UNIX File Servers to mobile mode.



How to Disconnect From the Central Protection Server

To disconnect the product from the central protection server and switch its operation into standalone mode, use the `esdisconnect` [command](#) of the [Dr.Web Ctl](#) command-line tool provided in Dr.Web for UNIX File Servers:

```
# drweb-ctl esdisconnect
```

To use the product in standalone mode, a valid license [key file](#) is required. Otherwise, anti-virus functions of the product will be *blocked* after the operation is switched to standalone mode.

How to Activate the Product

1. Register on the official website of Doctor Web at <https://products.drweb.com/register/>.
2. At the email address that you specified during the registration you will receive an archive containing a valid license key file (you can also download this archive directly from the website after you have finished the registration).
3. Carry out the key file [installation procedure](#).

How to Add a New SMB Shared Directory

1. Edit the configuration file—`smb.conf`—of the **Samba** server by adding a section describing the shared directory. The section describing the shared directory must be as follows:

```
[<share_name>]
comment = <any comment>
path = <path to the protected directory>
vfs objects = smb_spider
writeable = yes
browseable = yes
guest ok = yes
public = yes
```

where `<share_name>` is any name of the shared resource and `<any comment>` is an arbitrary line with a comment (optional).

2. If you need to specify scanning settings for the added shared directory and these settings differ from the default SpIDer Guard for SMB settings, use steps 3 and 4 of the instruction given in the [Integration with Samba File Server](#) section.
3. Restart the **Samba** server and Dr.Web for UNIX File Servers.

How to Add a New Protected NSS Volume

1. Specify the name of the volume, which is to be protected, in the **ProtectedVolumes** parameter (in the [NSS] [section](#) of the configuration file). If this parameter has no value assigned to it, then all the volumes from the directory for set by the **NssVolumesMountDir** parameter are protected.



2. Restart Dr.Web for UNIX File Servers.

How to Upgrade the Product

[Update](#) component versions or [upgrade to a new version](#).

Note that during the upgrade you can be asked to remove the current product version.

How To Add or Remove Component of the Product

Follow the [Custom Component Installation and Uninstallation](#) procedure.

Note that when installing and uninstalling the component, other product components could be additionally installed or uninstalled to resolve dependencies.

How to Manage Components Operation

To view the status of the product's components and to manage their operation, you can use:

- The [command-line-based management tool](#) Dr.Web Ctl (use the **drweb-ctl** `appinfo`, **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` commands. To view the list of available management commands, use the command **drweb-ctl** `--help`).
- The management [web interface](#) of Dr.Web for UNIX File Servers (by default, you can access it via a web browser at `https://127.0.0.1:4443/`).

How to View Log of the Product

According to default settings the general log of all product's components is displayed in **syslog** file (the file for logging messages by the system component **syslog** depends on the system and is located in the directory `/var/log`). General log settings are defined in the [configuration file](#) in the [section](#) [Root] (parameters **Log** and **DefaultLogLevel**). For each [component](#) in their settings section, parameters **Log** and **LogLevel** are available. They set the log storage location and the logging level of messages that the component outputs in the log.

To change the logging settings, use the Dr.Web Ctl command line management tool and the Dr.Web for UNIX File Servers management web interface (if it is installed).

- To identify errors, we recommend you to configure output of the general log of all components to a separate file and enable output of extended debug information to the log. For that, execute the following commands:

```
# drweb-ctl cfset Root.Log <path to log file>
# drweb-ctl cfset Root.DefaultLogLevel DEBUG
```

- To return to the default logging method and verbosity level for all components, execute the following commands:



```
# drweb-ctl cfset Root.Log -r  
# drweb-ctl cfset Root.DefaultLogLevel -r
```



Components of the Product

This section contains a description of the components of Dr.Web for UNIX File Servers. For each of them, you can find information about its functions, operation principles, and parameters stored in the [configuration file](#) of the product.

Dr.Web ConfigD

The configuration daemon—Dr.Web ConfigD—is the core component of Dr.Web for UNIX File Servers. It provides central storage of configuration information for all program components, manages operation of all components, and organizes trusted data exchange between them.

Operating Principles

Main Functions

1. Starts and stops the product's components depending on the settings. Automatically restarts components if a failure in their operation occurs. Starts components at the request of other components. Informs active components when another component starts or shuts down.
2. Provides for a centralized access of all components to configuration settings. Provides special components with interface for centralized management of configuration parameters. Notifies all required components about changes in configuration.
3. Provides components with information from the used license key file. Receives new license information from special components. Notifies running components on changes in license data or in configuration parameters.

The configuration daemon Dr.Web ConfigD is always started with *root* privileges. It launches other components of Dr.Web for UNIX File Servers and communicates with them via a preliminarily open socket. The configuration daemon receives connections from other components via an information socket (publicly available) and a management socket (available only for components with superuser privileges). Loads configuration parameters and license data from files or delivers them from the used central protection server via the [Dr.Web ES Agent](#), as well as substitutes default correct values for configuration parameters. Thus, by the moment when any component starts or `SIGHUP` signal is sent, the configuration daemon has an integral and consistent set of parameters for Dr.Web for UNIX File Servers.

Upon receipt of `SIGHUP` signal, the configuration management daemon rereads configuration parameters and license data. If required, the daemon sends all components notifications instructing them to reread their configuration. Upon receipt of `SIGTERM` signal, the daemon shuts all components down and only after that finishes its own operation. The daemon also removes all temporary files of components after they are shut down.



Communication Principles

1. All components use only configuration parameters and license data received from the configuration daemon Dr.Web ConfigD on their startup.
2. The daemon collects messages from all the controlled components into an integrated log. All information output by a component to the error stream *stderr* is collected by the daemon and written to the integrated log of the product with a mark indicating what component has output this.
3. When shutting down, the controlled components return an exit code. If the code differs from 101, 102, or 103, the configuration daemon restarts this component. Thus, abnormal termination of a component triggers its restart and registration of an error message from *stderr* in the product's log.
 - If a component exits with code 101, the component will be started again only after license parameters are changed. Thus, if a component cannot operate because of license restriction, it terminates its operation and outputs code 101 to *stderr*.
 - If a component exits with code 102, the component will be started again only after configuration parameters change. Thus, if a component cannot operate because of its configuration, it terminates its operation and outputs code 102 to *stderr*. The configuration daemon will attempt to start the component again only after any parameters are changed.
 - Components started by the configuration daemon at request can terminate their operation when idle and output code 103. It is such components as [Dr.Web Scanning Engine](#) and [Dr.Web File Checker](#).
 - If new parameter values received by the component from the configuration daemon cannot be applied "on the fly", that is, if the restart is required, the component exits with code 0. If so, Dr.Web ConfigD restarts the component.
 - If a component cannot connect to the configuration daemon or a communication protocol error occurs, the component outputs an appropriate message to *stderr* and exits with code 1.
4. Signal exchange:
 - The configuration daemon sends the component `SIGHUP` signal, which instructs to change parameters of configuration.
 - The configuration daemon sends the component `SIGTERM` signal, which instructs the component to terminate operation in 30 seconds.
 - `SIGKILL` signal is sent by the configuration daemon to trigger force termination of components which failed to shut down within 30 seconds after they received a `SIGTERM` signal.



Command-Line Arguments

To run the configuration daemon Dr.Web ConfigD, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-configd [<parameters>]
```

The configuration daemon Dr.Web ConfigD can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.
--config	Description: Instructs to use the specified configuration file for further operation. Short form: -c Arguments: <path to the file>—the path to the configuration file that you want to use.
--daemonize	Description: Instructs to run the component as a daemon; that is, without access to the terminal. Short form: -d Arguments: None.
--pid-file	Description: Instructs to use the specified PID file for further operation. Short form: -p Arguments: <path to the file>—the path to a file into which you would like to the process ID (PID) to be Stored.

Example:

```
$ /opt/drweb.com/bin/drweb-configd -d -c /etc/opt/drweb.com/drweb.ini
```

The command runs Dr.Web ConfigD as a daemon which uses the following configuration file: /etc/opt/drweb.com/drweb.ini.

Startup Notes

To enable the operation of the product, Dr.Web ConfigD must be running as a daemon. During standard booting, Dr.Web ConfigD is automatically launched when the operating system starts; for this purpose Dr.Web ConfigD comes together with a standard management script located



in `/etc/init.d`. To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-configd**

Configuration Parameters

The daemon Dr.Web ConfigD uses configuration parameters which are specified in the `[Root]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

DefaultLogLevel <i>{logging level}</i>	Defines default logging level of event logging for all Dr.Web for UNIX File Servers components. <i>The value of this parameter is used for all the components in the product which do not have their own different logging levels set up in their configuration.</i> Default value: Notice
LogLevel <i>{logging level}</i>	Logging level of event logging for Dr.Web ConfigD. Default value: Notice
Log <i>{log type}</i>	Logging method of the configuration daemon and logging method of those components for which another value of this parameter is not specified. Note that upon its initial startup, before the configuration file is read, the configuration daemon uses the following values of the parameter: <ul style="list-style-type: none">• As a daemon (if run with the <code>-d</code> option)—<code>SYSLLOG:Daemon</code>• Otherwise—<code>Stderr</code> If a component is working in a background mode (was launched with the <code>-d</code> option from the command line), then the <code>Stderr</code> value <i>cannot be used</i> for this parameter. Default value: On
PublicSocketPath <i>{path to file}</i>	Path to the socket used for interaction between all Dr.Web for UNIX File Servers components. Default value: <code>/var/run/.com.drweb.public</code>
AdminSocketPath <i>{path to file}</i>	Path to the socket used for interaction between Dr.Web for UNIX File Servers components with elevated (administrative) privileges. Default value: <code>/var/run/.com.drweb.admin</code>



CoreEnginePath <i>{path to file}</i>	<p>Path to the dynamic library of the anti-virus engine Dr.Web Virus-Finding Engine.</p> <p>Default value: <var_dir>/lib/drweb32.dll</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/lib/drweb32.dll• For FreeBSD: /var/drweb.com/lib/drweb32.dll
VirusBaseDir <i>{path to directory}</i>	<p>Path to the directory with virus database files.</p> <p>Default value: <var_dir>/bases</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/bases• For FreeBSD: /var/drweb.com/bases
KeyPath <i>{path to file}</i>	<p>Path to the product key file (license or demo).</p> <p>Default value: <etc_dir>/drweb32.key</p> <ul style="list-style-type: none">• For Linux, Solaris: /etc/opt/drweb.com/drweb32.key• For FreeBSD: /usr/local/etc/drweb.com/drweb32.key
CacheDir <i>{path to directory}</i>	<p>Path to the cache directory (used to hold cache for updates as well as cache for information about checked files).</p> <p>Default value: <var_dir>/cache</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/cache• For FreeBSD: /var/drweb.com/cache
TempDir <i>{path to directory}</i>	<p>Path to the directory with temporary files.</p> <p>Default value: <i>Path copied from the system environment variable TMPDIR, TMP, TEMP or TEMPDIR (the environment variables are searched in this particular order). Otherwise /tmp, if there are no these environment variables.</i></p>
RunDir <i>{path to directory}</i>	<p>Path to the directory with all PID files of running components and sockets used for interaction between the product's components.</p> <p>Default value: /var/run</p>
VarLibDir <i>{path to directory}</i>	<p>Path to the directory with libraries used by product components.</p> <p>Default value: <var_dir>/lib</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/lib• For FreeBSD: /var/drweb.com/lib
VersionDir <i>{path to directory}</i>	<p>The path to a directory, where the information on Dr.Web for UNIX File Servers components current versions is stored.</p> <p>Default value: <var_dir>/version</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/version• For FreeBSD: /var/drweb.com/version
DwsDir	<p>The parameter is not used.</p>



{path to directory}	Default value: <var_dir>/dws <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/dws• For FreeBSD: /var/drweb.com/dws
AdminGroup {group name GID}	Group of users with administrative privileges for Dr.Web for UNIX File Servers management. These users, in addition to the <i>root</i> superuser, are allowed to elevate privileges of Dr.Web for UNIX File Servers components to superuser privileges. Default value: <i>Is determined during the installation of the product.</i>
TrustedGroup {group name GID}	The parameter is not used. Default value: drweb
DebugIpc {Boolean}	Indicates whether detailed IPC messages are included into the log file on the debug level (i.e. when LogLevel = DEBUG). IPC messages show the interaction between the configuration daemon and other components. Default value: No
UseCloud {Boolean}	Indicates whether to refer to Dr.Web Cloud service to receive information about malicious files and URLs. Default value: No
AntispamCorePath {path to file}	The parameter is not used. Default value: <var_dir>/lib/vaderetro.so <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/lib/vaderetro.so• For FreeBSD: /var/drweb.com/lib/vaderetro.so
VersionNotification {Boolean}	Notify a user on availability of updates to update the currently installed product version. Default value: Yes



Dr.Web Ctl

You can manage operation of Dr.Web for UNIX File Servers from the command line with the help of a special command-line tool—Dr.Web Ctl (**drweb-ctl**).

You can do the following actions from the command line:

- Start scanning file system objects including boot records
- Launch of scanning of files on remote network hosts (see note [below](#)).
- Start updating anti-virus components (virus databases, anti-virus engine, etc. depending on the distribution).
- View and change parameters of Dr.Web for UNIX File Servers configuration
- View the status of the product's components and statistics on detected threats
- View quarantine and manage quarantined objects (via the Dr.Web Ctl [component](#)).
- Connect to the central protection server or disconnect from it.

[Commands](#) entered by the user to control the product can have an effect only if the [Dr.Web ConfigD](#) configuration daemon is running (by default, it is automatically launched at the operating system's startup).



Note that some control commands require superuser privileges.

To elevate privileges, use the **su** command (change the current user) or the **sudo** command (execute the specified command with other user privileges).

The Dr.Web Ctl tool supports auto-completion of commands for managing Anti-virus operation if this option is enabled in the used command shell. If the command shell does not allow auto-completion, you can configure this option. For that purpose, refer to the instruction manual for the used OS distribution.



When shutting down, the tool returns the exit code according to convention for the POSIX compliant systems: 0 (zero)—if an operation is successfully completed, non-zero—if otherwise.

Note that the tool returns a non-null exit code only in case of internal error (for example, the tool could not connect to a component, a requested operation could not be executed, etc.). If the tool detects (and possibly) neutralizes a threat, it returns the null exit code, because the requested operation (such as `scan`, etc.) is successfully completed. If it is necessary to define the list of detected threats and applied actions, analyze the messages displayed on the console.

Codes of all errors are listed in the [Appendix F. Known Errors](#) section.



Remote host scanning

Dr.Web for UNIX File Servers allows to perform scanning for threats of files located on remote network hosts. Such hosts can be not only full computing machines (workstations and servers) but also routers, set-top boxes and other “smart” devices that form the so-called Internet of things. To perform the remote scanning, it is necessary for the remote host to provide a remote terminal access via SSH (Secure Shell). Besides, it is required to know an IP address and a domain name of the remote host, name and password of the user, who could remotely access the system via SSH. The indicated user must have access rights to the scanned files (at least the reading rights).

This function can be used only for detection of malicious and suspicious files on a remote host. Elimination of threats (i.e. isolation in the quarantine, removal and curing of malicious objects) using means of the remote scanning is impossible. To eliminate detected threats on the remote host, it is necessary to use administration tools provided directly by this host. For example, for routers and other “smart” devices, a mechanism for a firmware update can be used; for computing machines, it can be done via a connection to them (as an option, using a remote terminal mode) and respective operations in their file system (removal or moving of files, etc.), or via running an anti-virus software installed on them.

Remote scanning is performed only via the command-line tool Dr.Web Ctl (the `remotescan` [command](#) is used).

Command-Line Call Format

1. Command Format for Calling the Command-Line Utility to Manage the Product

The call format for the command-line tool which manages Dr.Web for UNIX File Servers operation is as follows:

```
$ drweb-ctl [<general options>] | <command> [<argument>] [<command options>]
```

Where:

- *<general options>*—options that can be applied on startup when the command is not specified or can be applied for any command. Not mandatory for startup.
- *<command>*—command to be performed by Dr.Web for UNIX File Servers (for example, start scanning, output the list of quarantined objects, and other commands).
- *<argument>*—command argument. Depends on the specified command. It can be missing for certain commands.
- *<command options>*—options for managing the operation of the specified command. They can be omitted for some commands.



2. General Options

The following general options are available:

Option	Description
-h, --help	Show general help information and exit. To display the help information on any command, use the following call: <pre>\$ drweb-ctl <command> -h</pre>
-v, --version	Show information on the module version and exit
-d, --debug	Instructs to show debug information upon execution of the specified command. It cannot be executed if a command is not specified. Use the call <pre>\$ drweb-ctl <command> -d</pre>

3. Commands

Commands to manage Dr.Web for UNIX File Servers can be divided into the following groups:

- [Anti-virus scanning](#) commands.
- Commands to [manage updates](#) and operation in central protection mode.
- [Configuration management](#) commands.
- Commands to [manage detected threats and quarantine](#).
- [Information commands](#).




To request documentation about this component of the product from the command line, use the following command **man 1 drweb-ctl**

3.1. Anti-virus Scanning Commands

The following commands to manage anti-virus scanning are available:

Command	Description
scan <path>	<p>Purpose: Start checking the specified file or directory via the Dr.Web File Checker component.</p> <p>Arguments:</p> <p><path>—path to the file or directory which is selected for scanning.</p>




Command	Description
	<p><i>This argument may be omitted, if you use the <code>--stdin</code> or the <code>--stdin0</code> option. To specify several files that satisfy a certain criterion, use the find utility (see the Usage Examples) and the <code>--stdin</code> or <code>--stdin0</code> option.</i></p> <p>Options:</p> <p><code>-a [--Autonomous]</code>—run a separate instance of Dr.Web Scanning Engine and Dr.Web File Checker to perform specified checks and terminate their operation after the scanning task is completed. Note that threats detected during stand-alone scanning are not added in the common threat list that is displayed using the <code>threats</code> command (see below).</p> <p><code>--stdin</code>—get the list of paths to scan from the standard input string (<i>stdin</i>). Paths in the list need to be separated by the next line character (<code>'\n'</code>).</p> <p><code>--stdin0</code>—get the list of paths to scan from the standard input string (<i>stdin</i>). Paths in the list need to be separated by the zero character NUL (<code>'\0'</code>).</p> <div><p>When using <code>--stdin</code> and <code>--stdin0</code> options, the paths in the list should not contain patterns or regular expressions for a search. Recommended usage of the <code>--stdin</code> and <code>--stdin0</code> options is processing a path list (generated by an external utility, for example, find) in the <code>scan</code> command (see Usage Examples).</p></div> <p><code>--Report <BRIEF DEBUG></code>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>BRIEF</i>—brief report.• <i>DEBUG</i>—detailed report. <p>Default value: <i>BRIEF</i></p> <p><code>--ScanTimeout <number></code>—specify timeout to scan one file, in ms.</p> <p>If the value is set to <i>0</i>, time on scanning is not limited.</p> <p>Default value: <i>0</i></p> <p><code>--PackerMaxLevel <number></code>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--ArchiveMaxLevel <number></code>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--MailMaxLevel <number></code>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p>



Command	Description
	<p>Default value: 8</p> <p>--ContainerMaxLevel <number>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MaxCompressionRatio <ratio>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: 3000</p> <p>--HeuristicAnalysis <On Off>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: On</p> <p>--OnKnownVirus <action>—action applied to a threat detected by using signature-based analysis.</p> <p>Allowed values: REPORT, CURE, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnIncurable <action>—action applied on failure to cure a detected threat or if a threat is incurable.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnSuspicious <action>—action applied to a suspicious object detected by heuristic analysis.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnAdware <action>—action applied to detected adware programs.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnDialers <action>—action applied to dialers.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnJokes <action>—action applied to joke programs.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnRiskware <action>—action applied to potentially dangerous programs (riskware).</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnHacktools <action>—action applied to hacktools.</p>



Command	Description
	<p>Allowed values: <i>REPORT, QUARANTINE, DELETE</i>.</p> <p>Default value: <i>REPORT</i></p> <div><p>If threat is detected in a file located in a container (an archive, email message, etc.), its removal (<i>DELETE</i>) is replaced with moving of a container to quarantine (<i>QUARANTINE</i>).</p></div>
<code>bootscan</code> <code><disk drive> ALL</code>	<p>Purpose: Start checking boot records on the specified disks via the Dr.Web File Checker component. Both MBR and VBR records are scanned.</p> <p>Arguments:</p> <p><code><disk drive></code>—path to the block file of a disk device whose boot record you want to scan. You can specify several disk devices separated by spaces. The argument is mandatory. If <code>ALL</code> is specified instead of the device file, all boot records on all available disk devices will be checked.</p> <p>Options:</p> <p><code>-a [--Autonomous]</code>—run a separate instance of Dr.Web Scanning Engine and Dr.Web File Checker to perform specified checks and terminate their operation after the scanning task is completed. Note that threats detected during stand-alone scanning are not added in the common threat list that is displayed using the <code>threats</code> command (see below).</p> <p><code>--Report <BRIEF DEBUG></code>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>BRIEF</i>—brief report.• <i>DEBUG</i>—detailed report. <p>Default value: <i>BRIEF</i></p> <p><code>--ScanTimeout <number></code>—specify timeout to scan one file, in ms.</p> <p>If the value is set to <i>0</i>, time on scanning is not limited.</p> <p>Default value: <i>0</i></p> <p><code>--HeuristicAnalysis <On Off></code>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: <i>On</i></p> <p><code>--Cure <Yes No></code>—enable or disable attempts to cure detected threats.</p> <p>If the value is set to <i>No</i>, only a notification about a detected threat is displayed.</p> <p>Default value: <i>No</i></p> <p><code>--ShellTrace</code>—enable display of additional debug information when scanning a boot record.</p>




Command	Description
proscan	<p>Purpose: Start checking executable files containing code of currently running processes with the Dr.Web File Checker. If a malicious executable file is detected, it is neutralized, and all processes run by this file are forced to terminate.</p> <p>Arguments: None.</p> <p>Options:</p> <p>-a [--Autonomous]—run a separate instance of Dr.Web Scanning Engine and Dr.Web File Checker to perform specified checks and terminate their operation after the scanning task is completed. Note that threats detected during stand-alone scanning are not added in the common threat list that is displayed using the <code>threats</code> command (see below).</p> <p>--Report <<i>BRIEF DEBUG</i>>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>BRIEF</i>—brief report.• <i>DEBUG</i>—detailed report. <p>Default value: <i>BRIEF</i></p> <p>--ScanTimeout <<i>number</i>>—specify timeout to scan one file, in ms.</p> <p>If the value is set to <i>0</i>, time on scanning is not limited.</p> <p>Default value: <i>0</i></p> <p>--HeuristicAnalysis <<i>On Off</i>>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: <i>On</i></p> <p>--PackerMaxLevel <<i>number</i>>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p>--OnKnownVirus <<i>action</i>>—action applied to a threat detected by using signature-based analysis.</p> <p>Allowed values: <i>REPORT, CURE, QUARANTINE, DELETE</i>.</p> <p>Default value: <i>REPORT</i></p> <p>--OnIncurable <<i>action</i>>—action applied on failure to cure a detected threat or if a threat is incurable.</p> <p>Allowed values: <i>REPORT, QUARANTINE, DELETE</i>.</p> <p>Default value: <i>REPORT</i></p> <p>--OnSuspicious <<i>action</i>>—action applied to a suspicious object detected by heuristic analysis.</p> <p>Allowed values: <i>REPORT, QUARANTINE, DELETE</i>.</p> <p>Default value: <i>REPORT</i></p>



Command	Description
	<p>--OnAdware <action>—action applied to detected adware programs.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnDialers <action>—action applied to dialers.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnJokes <action>—action applied to joke programs.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnRiskware <action>—action applied to potentially dangerous programs (riskware).</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnHacktools <action>—action applied to hacktools.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p><i>Note that if a threat is detected in an executable file, Dr.Web for UNIX File Servers terminates all processes started from the file.</i></p>
netscan <path>	<p>Purpose: Start distributed scanning of the specified file or directory via the Dr.Web Network Checker agent for network data scanning. If there are no configured connections to other hosts that are running Dr.Web for UNIX, then the scanning will be done only via the locally-available scanning engine (similar to the scan command).</p> <p>Arguments:</p> <p><path>—path to the file or directory which is selected to be scanned.</p> <p>Options:</p> <p>--Report <BRIEF DEBUG>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• BRIEF—brief report.• DEBUG—detailed report. <p>Default value: BRIEF</p> <p>--ScanTimeout <number>—specify timeout to scan one file, in ms.</p> <p>If the value is set to 0, time on scanning is not limited.</p> <p>Default value: 0</p> <p>--HeuristicAnalysis <On Off>—enable or disable heuristic analysis during the scanning.</p>





Command	Description
	<p>Default value: <i>On</i></p> <p><code>--PackerMaxLevel <number></code>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--ArchiveMaxLevel <number></code>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--MailMaxLevel <number></code>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--ContainerMaxLevel <number></code>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to <i>0</i>, nested objects will be skipped during scanning.</p> <p>Default value: <i>8</i></p> <p><code>--MaxCompressionRatio <ratio></code>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: <i>3000</i></p> <p><code>--Cure <Yes No></code>—enable or disable attempts to cure detected threats.</p> <p>If the value is set to <i>No</i>, only a notification about a detected threat is displayed.</p> <p>Default value: <i>No</i></p>
<code>flowscan <path></code>	<p>Purpose: to start scanning the specified file or directory via Dr.Web File Checker using the “flow” method (normally this method is used internally by SpIDer Guard).</p> <div> For on-demand scanning of files and directories, it is recommended that you use the <code>scan</code> command.</div> <p>Arguments:</p> <p><code><path></code>—path to the file or directory which is selected to be scanned.</p> <p>Options:</p> <p><code>--ScanTimeout <number></code>—specify timeout to scan one file, in ms.</p> <p>If the value is set to <i>0</i>, time on scanning is not limited.</p> <p>Default value: <i>0</i></p>




Command	Description
	<p>--HeuristicAnalysis <On Off>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: On</p> <p>--PackerMaxLevel <number>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ArchiveMaxLevel <number>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MailMaxLevel <number>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ContainerMaxLevel <number>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MaxCompressionRatio <ratio>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: 3000</p> <p>--OnKnownVirus <action>—action applied to a threat detected by using signature-based analysis.</p> <p>Allowed values: REPORT, CURE, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnIncurable <action>—action applied on failure to cure a detected threat or if a threat is incurable.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnSuspicious <action>—action applied to a suspicious object detected by heuristic analysis.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnAdware <action>—action applied to detected adware programs.</p> <p>Allowed values: REPORT, QUARANTINE, DELETE.</p> <p>Default value: REPORT</p> <p>--OnDialers <action>—action applied to dialers.</p>



Command	Description
	<p>Allowed values: <i>REPORT, QUARANTINE, DELETE.</i></p> <p>Default value: <i>REPORT</i></p> <p>--OnJokes <i><action></i>—action applied to joke programs.</p> <p>Allowed values: <i>REPORT, QUARANTINE, DELETE.</i></p> <p>Default value: <i>REPORT</i></p> <p>--OnRiskware <i><action></i>—action applied to potentially dangerous programs (riskware).</p> <p>Allowed values: <i>REPORT, QUARANTINE, DELETE.</i></p> <p>Default value: <i>REPORT</i></p> <p>--OnHacktools <i><action></i>—action applied to hacktools.</p> <p>Allowed values: <i>REPORT, QUARANTINE, DELETE.</i></p> <p>Default value: <i>REPORT</i></p> <div> If threat is detected in a file located in a container (an archive, email message, etc.), its removal (<i>DELETE</i>) is replaced with moving of a container to quarantine (<i>QUARANTINE</i>).</div>
proxyscan <i><path></i>	<p>Purpose: Start scanning the specified file or directory via Dr.Web File Checker using the “flow” method (normally this method is used internally by the SpIDer Guard for SMB monitor and Dr.Web ClamD component).</p> <div> Note that threats detected by this scanning method are not included into the list of detected threats that is displayed by the <code>threats</code> command (see below).</div> <div>For on-demand scanning of files and directories, it is recommended that you use the <code>scan</code> command.</div> <p>Arguments:</p> <p><i><path></i>—path to the file or directory which is selected to be scanned.</p> <p>Options:</p> <p>--Report <i><BRIEF DEBUG></i>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>BRIEF</i>—brief report.• <i>DEBUG</i>—detailed report. <p>Default value: <i>BRIEF</i></p> <p>--ScanTimeout <i><number></i>—specify timeout to scan one file, in ms.</p> <p>If the value is set to 0, time on scanning is not limited.</p>




Command	Description
	<p>Default value: 0</p> <p>--HeuristicAnalysis <On Off>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: On</p> <p>--PackerMaxLevel <number>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ArchiveMaxLevel <number>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MailMaxLevel <number>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ContainerMaxLevel <number>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MaxCompressionRatio <ratio>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: 3000</p>
rawscan <path>	<p>Purpose: to start “raw” scanning of the specified file or directory by Dr.Web Scanning Engine directly, without the use of Dr.Web File Checker.</p> <div><p>Note that threats detected by “raw” scanning are not included into the list of detected threats that is displayed by the threats command (see below).</p><hr/><p>It is recommended that you use this command only to debug the functioning of Dr.Web Scanning Engine. Note that the command outputs the “cured” status, if at least <i>one</i> threat is neutralized of those threats that are detected in a file (not <i>all</i> threats might be neutralized). Thus, it is <i>not recommended</i> to use this command if you need thorough file scanning. In the latter case it is recommended to use the scan command.</p></div>



Command	Description
	<p>Arguments:</p> <p><path>—path to the file or directory which is selected to be scanned.</p> <p>Options:</p> <p>--ScanEngine <path>—path to the UNIX socket of the Dr.Web Scanning Engine. If not specified, an autonomous instance of the scanning engine is started (which will be shut down once the scanning is completed).</p> <p>--Report <BRIEF DEBUG>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• BRIEF—brief report.• DEBUG—detailed report. <p>Default value: BRIEF</p> <p>--ScanTimeout <number>—specify timeout to scan one file, in ms.</p> <p>If the value is set to 0, time on scanning is not limited.</p> <p>Default value: 0</p> <p>--PackerMaxLevel <number>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ArchiveMaxLevel <number>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MailMaxLevel <number>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--ContainerMaxLevel <number>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p>--MaxCompressionRatio <ratio>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: 3000</p> <p>--HeuristicAnalysis <On Off>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: On</p>



Command	Description
	<p>--Cure <Yes No>—enable or disable attempts to cure detected threats.</p> <p>If the value is set to <i>No</i>, only a notification about a detected threat is displayed.</p> <p>Default value: <i>No</i></p> <p>--ListCleanItem—enable outputting the list of clean (non-infected) files found inside a container that was scanned.</p> <p>--ShellTrace—enable display of additional debug information when scanning a file.</p>
remotescan <host> <path>	<p>Purpose: Connect to the specified remote host and start scanning the specified file or directory using SSH.</p> <div><p>Note that threats detected by remote scanning will not be neutralized and also will not be included into the list of detected threats that is displayed by the <code>threats</code> command (see below).</p><p>This function can be used only for detection of malicious and suspicious files on a remote host. To eliminate detected threats on the remote host, it is necessary to use administration tools provided directly by this host. For example, for routers and other “smart” devices, a mechanism for a firmware update can be used; for computing machines, it can be done via a connection to them (as an option, using a remote terminal mode) and respective operations in their file system (removal or moving of files, etc.), or via running an anti-virus software installed on them.</p></div> <p>Arguments:</p> <p><host>—IP address or a domain name of the remote host.</p> <p><path>—path to the file or directory which is selected to be scanned.</p> <p>Options:</p> <p>-l [--Login] <name>—login (user name) used for authorization on the remote host via SSH.</p> <p><i>If a user name is not specified, there will be an attempt to connect to a remote host on behalf of the user who has launched the command.</i></p> <p>-i [--Identity] <path to file>—path to the file containing a private key used for authentication of the specified user via SSH.</p> <p>-p [--Port] <number>—number of the port on the remote host for connecting via SSH.</p> <p>Default value: 22</p>




Command	Description
	<p><code>--Password <password></code>—password used for authentication of a user via SSH.</p> <p><i>Please note that the password is transferred as a plain text.</i></p> <p><code>--Report <BRIEF DEBUG></code>—specify the type of the report with scanning results.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>BRIEF</i>—brief report.• <i>DEBUG</i>—detailed report. <p>Default value: <i>BRIEF</i></p> <p><code>--ScanTimeout <number></code>—specify timeout to scan one file, in ms.</p> <p>If the value is set to 0, time on scanning is not limited.</p> <p>Default value: 0</p> <p><code>--PackerMaxLevel <number></code>—set the maximum nesting level when scanning packed objects.</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p><code>--ArchiveMaxLevel <number></code>—set the maximum nesting level when scanning archives (zip, rar, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p><code>--MailMaxLevel <number></code>—set the maximum nesting level when scanning email messages (pst, tbb, etc.).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p><code>--ContainerMaxLevel <number></code>—set the maximum nesting level when scanning other containers (HTML and so on).</p> <p>If the value is set to 0, nested objects will be skipped during scanning.</p> <p>Default value: 8</p> <p><code>--MaxCompressionRatio <ratio></code>—set the maximum compression ratio of scanned objects.</p> <p>The ratio must be at least equal to 2.</p> <p>Default value: 3000</p> <p><code>--HeuristicAnalysis <On Off></code>—enable or disable heuristic analysis during the scanning.</p> <p>Default value: On</p>




3.2. Commands to manage updates and operation in Central protection mode

The following commands for managing updates and operation in central protection mode are available:

Command	Description
update	Purpose: Instructs to initiate the updating process of the anti-virus components (virus databases, anti-virus engine, etc. depending on the distribution) from Doctor Web's update servers
esconnect <server> [: <port>]	<p>Purpose: Connect Dr.Web for UNIX File Servers to the specified central protection server (for example, Dr.Web Enterprise Server). For details, refer to Operation Modes.</p> <p>Arguments:</p> <ul style="list-style-type: none">• <server>—IP address or network name of the host on which the central protection server is operating. This argument is mandatory.• <port>—port number used by the central protection server. The argument is optional and should be specified only if the central protection server uses a non-standard port. <p>Options:</p> <p>--Key <path>—path to the public key file of the central protection server to which connection is performed.</p> <p>--Login <ID>—login (workstation identifier) used for connection to the central protection server.</p> <p>--Password <password>—password for connection to the central protection server.</p> <p>--Group <ID>—identifier of the group to which the workstation is added on connection.</p> <p>--Rate <ID>—identifier of the tariff group applied to your workstation when it is included in one of the central protection server groups (can be specified only together with the --Group option).</p> <p>--Compress <On Off>—enables (On) or disables (Off) forced compression of transmitted data. If not specified, usage of compression is determined by the server.</p> <p>--Encrypt <On Off>—enables (On) or disables (Off) forced encryption of transmitted data. If not specified, usage of encryption is determined by the server.</p> <p>--Newbie—connect as a "newbie" (get a new account on the server).</p> <div> This command requires drweb-ctl to be started with root privileges. If necessary, use the su or sudo commands.</div>




Command	Description
esdisconnect	<p>Purpose: Disconnect Dr.Web for UNIX File Servers from the central protection server and switch its operation to standalone mode.</p> <p><i>The command has no effect if Dr.Web for UNIX File Servers already operates in standalone mode.</i></p> <p>Arguments: None.</p> <p>Options: None</p> <div> This command requires drweb-ctl to be started with <i>root</i> privileges. If necessary, use the su or sudo commands.</div>

3.3. Configuration Management Commands

The following commands to manage configuration are available:

Command	Description
<code>cfset</code> <code><section> . <parameter></code> <code><value></code>	<p>Purpose: to change the active value of the specified parameter in the current configuration.</p> <p><i>Note that an equals sign is not allowed.</i></p> <p>Arguments:</p> <ul style="list-style-type: none">• <code><section></code>—name of the configuration file's section where the parameter resides. This argument is mandatory.• <code><parameter></code>—name of the parameter. The argument is mandatory.• <code><value></code>—new value that is to be assigned to the parameter. The argument is mandatory. <p><i>The following format is always used to specify a parameter value:</i> <code><section> . <parameter> <value></code>.</p> <p><i>Note that if you want to indicate several parameter values, you need to repeat the call of the command <code>cfset</code> as many times as the number of parameter values you want to ass. In addition, to ass a new value to the list of the parameter values, you need to use an option <code>-a</code> (see below). You cannot use the command option <code><parameter> value1, value2</code>, because the string <code>value1, value2</code> will be considered a unified parameter value.</i></p> <p><i>For description of the configuration file, refer to the section Appendix D. Configuration File, or to the documentation page displayed by <code>man 5 drweb.ini</code>.</i></p>



Command	Description
	<p>Options:</p> <p><code>-a [--Add]</code>—do not substitute the current parameter value but add the specified value to the list (allowed only for parameters that can have several values, specified as a list). You should also use this option to when adding a new parameter group identified by a tag.</p> <p><code>-e [--Erase]</code>—do not substitute the current parameter value but remove the specified value from the list (allowed only for parameters that can have several values, specified as a list). You can also use this option to delete the whole group of parameters with a tag.</p> <p><code>-r [--Reset]</code>—reset the parameter value to the default. At that, <code><value></code> is not required in the command and is ignored if specified.</p> <p>Options are not mandatory. If they are not specified, then the current parameter value (the entire list of values, if the parameter currently holds several values) are substituted with the specified value.</p> <p><i>If you use the <code>-r</code> option for sections that contain individualized parameter settings for different connection points to the Dr.Web ClamD component or for sections that contain individualized parameter settings for different shared directories for the SpIDer Guard for SMB monitor, parameter value in the individualized settings section will be changed to the value of its “parent” parameter having the same name and located in the general settings section of this component.</i></p> <p>If it is necessary to add a new connection point <code><point></code> for Dr.Web ClamD or a section containing parameters for a Sambashared directory with the <code><tag></code>, use the following command:</p> <pre>cfset ClamD.Endpoint.<point> -a, for example: cfset ClamD.Endpoint.point1 -a cfset SmbSpider.Share.<tag> -a, for example: cfset SmbSpider.Share.BuhFiles -a</pre> <div> This command requires drweb-ctl to be started with <i>root</i> privileges. If necessary, use the su or sudo commands.</div>
<code>cfshow</code> <code>[<section>]</code> <code>[. <parameter>]</code>	<p>Purpose: to display parameter values in the current configuration. The parameters are output to the display as follows <code><section>.<parameter> = <value></code>. Sections and parameters of non-installed components are not displayed.</p> <p>Arguments:</p> <ul style="list-style-type: none"><code><section></code>—name of the configuration file section parameters of which are to be displayed. The argument is optional. If not specified, parameters of all configuration file sections are displayed.



Command	Description
	<ul style="list-style-type: none">• <i><parameter></i>—name of the displayed parameter. If not specified, all parameters of the section are displayed. Otherwise, only this parameter is displayed. If a parameter is specified without the section name, all parameters with this name from all of the configuration file sections are displayed. <p>Options:</p> <p>--Uncut—display all configuration parameters (not only those used with the currently installed set of components). If the option is not specified, only parameters used for configuration of the installed components are displayed.</p> <p>--Changed—output only those parameters which have values different from the default ones.</p> <p>--Ini—display parameter values in the INI file format: at first, the section name is specified in square brackets, then the section parameters listed as <i><parameter> = <value></i> pairs (one pair per line).</p> <p>--Value—output only value of the specified parameter (the <i><parameter></i> argument is mandatory in this case).</p>
reload	<p>Purpose: to send the SIGHUP signal to the Dr.Web ConfigD configuration daemon.</p> <p>On receiving this signal, the Dr.Web ConfigD configuration daemon rereads the configuration and sends the required changes of it to Dr.Web for UNIX File Servers components. Then the configuration daemon reopens the program log, restarts the components that use virus databases (including the anti-virus engine), and attempts to restart those components which were terminated abnormally.</p> <p>Arguments: None.</p> <p>Options: None</p>

3.4. Commands to Manage Detected Threats and Quarantine

The following commands for managing threats and quarantine are available:

Command	Description
threats [<i><action></i> <i><object></i>]	<p>Purpose: Apply the specified action to detected threats, selected by their identifiers. Type of the action is specified by the command's option.</p> <p>If the action is not specified, displays information on detected but not neutralized threats. For each threat the following information is displayed:</p> <ul style="list-style-type: none">• Identifier assigned to the threat (its ordinal number)• The full path to the infected file





Command	Description
	<ul style="list-style-type: none">• Information about the threat (name of the threat, threat type according to the classification used by the Doctor Web company)• Information about the file: size, the file owner's user name, the time of last modification• History of operations applied to the threat: detection, applied actions etc. <p>Arguments: None.</p> <p>Options:</p> <p><code>-f [--Follow]</code>—wait for new messages about new threats and display them once they are received (CTRL+C interrupts the waiting).</p> <p><i>If this option is applied along with any options mentioned below, it is ignored.</i></p> <p><code>--Cure <threat list></code>—attempt to cure the listed threats (list threat identifiers separating them with commas).</p> <p><code>--Quarantine <threat list></code>—move the listed threats to quarantine (list threat identifiers separating them with commas).</p> <p><code>--Delete <threat list></code>—delete the listed threats (list threat identifiers separating them with commas).</p> <p><code>--Ignore <threat list></code>—ignore the listed threats (list threat identifiers separating them with commas).</p> <p>If it is required to apply the command to all detected threats, specify <code>All</code> instead of <code><threat list></code>. For example:</p> <pre>\$ drweb-ctl threats --Quarantine All</pre> <p>moves all detected malicious objects to quarantine.</p>
<code>quarantine</code> [<action> <object>]	<p>Purpose: Apply an action to the specified object in quarantine.</p> <p>If an action is not specified, information on quarantined objects and their identifiers together with brief information on the original files moved to quarantine is displayed. For every isolated (quarantined) object the following information is displayed:</p> <ul style="list-style-type: none">• Identifier assigned to the quarantined object• The original path to the file, before it was moved to quarantine.• The date when the file was put in quarantine• Information about the file: size, the file owner's user name, the time of last modification• Information about the threat (name of the threat, threat type according to the classification used by the Doctor Web company) <p>Arguments: None.</p>



Command	Description
	<p>Options:</p> <p><code>-a [--Autonomous]</code>—start a separate instance of the Dr.Web File Checker component for checking files for performing the specified quarantine command and shut it down after the command is completed.</p> <p><i>This option can be applied along with any options mentioned below.</i></p> <p><code>--Delete <object></code>—delete the specified object from quarantine.</p> <p><i>Note that objects are deleted from quarantine permanently—this action is irreversible.</i></p> <p><code>--Cure <object></code>—try to cure the specified object in the quarantine.</p> <p><i>Note that even if the object is successfully cured, it will remain in quarantine. To restore the cured object from quarantine, use the <code>--Restore</code> command.</i></p> <p><code>--Restore <object></code>—restore the specified object from the quarantine to its original location.</p> <p><i>Note that this command may require drweb-ctl to be started with superuser privileges. You can restore the file from quarantine even if it is infected.</i></p> <p><code>--TargetPath <path></code>—restore an object from the quarantine to the specified location: either as a file with the name specified here (if the <code><path></code> is a path to a file), or just to the specified directory (if the <code><path></code> is a path to a directory). Can be used only in combination with the <code>--Restore</code> command.</p> <p>As an <code><object></code> specify the object identifier in quarantine. To apply the command to all quarantined objects, specify <code>All</code> instead of <code><object></code>. For example,</p> <pre>\$ drweb-ctl quarantine --Restore All</pre> <p>restores all quarantined objects.</p> <p><i>Note that for the <code>--Restore All</code> variant the additional option <code>--TargetPath</code>, if specified, must set a path to a directory, not a path to a file.</i></p>
<code>nss_threats</code> [<code><action> <object></code>]	<p>Purpose: Apply the specified action to threats detected on NSS volumes; threats are selected by their identifiers. Type of the action is specified by the command's option.</p> <p>If the action is not specified, displays information on detected but not neutralized threats. The displayed information is essentially the same as the one that is displayed by calling the <code>threats</code> command (see above); but the scope of the information may be extended or modified based on the peculiarities of NSS storage.</p>



Command	Description
	<div> To use this command, it is necessary to have SpIDer Guard for NSS installed and started.</div> <p>Arguments: None.</p> <p>Options:</p> <ul style="list-style-type: none"><code>-f [--Follow]</code>—wait for new messages about new threats and display the messages once they are received (CTRL+C interrupts the waiting).<code>--Cure <threat list></code>—attempt to cure the listed threats (list threat identifiers separating them with commas).<code>--Quarantine <threat list></code>—move the listed threats to NSS quarantine (specify threat identifiers as a comma-separated list).<code>--Delete <threat list></code>—delete the listed threats (list threat identifiers separating them with commas).<code>--Ignore <threat list></code>—ignore the listed threats (list threat identifiers separating them with commas). <p>If it is required to apply the command to all the threats detected on NSS volumes specify <code>All</code> instead of a <code><threat list></code>. For example:</p> <div><pre>\$ drweb-ctl nss_threats --Quarantine All</pre></div> <p>moves all detected malicious objects to NSS quarantine.</p>
<code>nss_quarantine</code> [<action> <object>]	<p>Purpose: to apply an action to the specified object located in quarantine on NSS volumes.</p> <p>If not specified, the following information is output: object identifier in NSS quarantine and brief information on original files. The displayed information is essentially the same as the one that is displayed by calling the <code>quarantine</code> command (see above); but the scope of the information may be extended or modified based on the peculiarities of NSS storage.</p> <div> To use this command, it is necessary to have SpIDer Guard for NSS installed and started.</div> <p>Arguments: None.</p> <p>Options:</p> <ul style="list-style-type: none"><code>--Delete <object></code>—delete the specified object from quarantine NSS. <i>Note that objects are deleted from quarantine permanently—this action is irreversible.</i><code>--Cure <object></code>—try to cure the specified object in the quarantine NSS.



Command	Description
	<p><i>Note that even if the object is successfully cured, it will remain in quarantine. To restore the cured object from quarantine, use the <code>--Restore</code> command.</i></p> <p><code>--Rescan <object></code>—rescan the specified object in the NSS quarantine.</p> <p><i>Note that even if the rescanning will determine that the object is clean (not infected), it will stay in quarantine. To restore the object from quarantine, use the <code>--Restore</code> option.</i></p> <p><code>--Restore <object></code>—restore the specified object from the NSS quarantine.</p> <p><i>Note that this command may require drweb-ctl to be started with superuser privileges. You can restore the file from quarantine even if it is infected.</i></p> <p><code>--TargetPath <path></code>—restore an object from the quarantine to the specified location: either as a file with the name specified here (if the <code><path></code> is a path to a file), or just to the specified directory (if the <code><path></code> is a path to a directory).</p> <p><i>Can be used only in combination with the <code>--Restore</code> command.</i></p> <p>As an <code><object></code> specify the object identifier in NSS quarantine. To apply the command to all quarantined objects, specify <i>all</i> as an <code><object></code>. For example, the following command:</p> <pre>\$ drweb-ctl quarantine --Restore All</pre> <p>restores all quarantined objects.</p> <p><i>Note that for the <code>--Restore All</code> variant the additional option <code>--TargetPath</code>, if specified, must set a path to a directory, not a path to a file.</i></p>



If the *Quarantine* action is specified for some threat type in the [settings](#) of SpIDer Guard for NSS, the object containing a threat of this type will be placed to quarantine again on attempt to restore this object from quarantine to an NSS volume by the `nss_quarantine` command. For example, the following default settings:

```
NSS.OnKnownVirus = Cure
NSS.OnIncurable = Quarantine
```

move all incurable objects to quarantine. This is why, when any incurable object is restored from quarantine to an NSS volume by the `nss_quarantine` command, this object is automatically returned to quarantine.



3.5. Information Commands


The following information commands are available:

Command	Description
appinfo	<p>Purpose: Output information on active Dr.Web for UNIX File Servers components.</p> <p>The following information is displayed about each component that is currently running:</p> <ul style="list-style-type: none">• Internally-used name• Process identifier GNU/Linux (PID)• State (running, stopped etc.)• Error code, if the work of the component has been terminated because of an error• Additional information (optionally). <p>For the configuration daemon Dr.Web ConfigD the following is displayed as additional information:</p> <ul style="list-style-type: none">• The list of installed components—<i>Installed</i>• The list of components which must be launched by the configuration daemon—<i>Should run</i>. <p>Arguments: None.</p> <p>Options:</p> <p><code>-f [--Follow]</code>—wait for new messages on component status change and output them once such a message is received (interrupt waiting by pressing CTRL+C).</p>
baseinfo	<p>Purpose: Display the information on the current version of the Virus-Finding Engine and status of virus databases.</p> <p>The following information is displayed:</p> <ul style="list-style-type: none">• Version of the anti-virus engine• Date and time when the virus databases that are currently used were issued.• The number of available virus records (in the virus databases)• The time of the last successful update of the virus databases and of the anti-virus engine• The time of the next scheduled automatic update <p>Arguments: None.</p> <p>Options: None.</p>
certificate	<p>Purpose: Display the contents of the trusted certificate of Dr.Web used by Dr.Web for UNIX File Servers. To save the certificate to a <code><cert_name>.pem</code> file,</p>



Command	Description
	<p>you can use the following command:</p> <pre>\$ drweb-ctl certificate > <cert_name>.pem</pre> <p>Arguments: None.</p> <p>Options: None</p>
idpass <identifier>	<p>Purpose: Display the password that has been generated by the scanning component of email messages Dr.Web MailD for the email message with the indicated identifier and used for the protection of enclosed archive with threats removed from the email message (i.e. if RepackPassword parameter is set in the component settings to HMAC (<secret>)).</p> <p>Arguments:</p> <ul style="list-style-type: none">• <identifier>—identifier of email messages. <p>Options:</p> <p>-s [--Secret] <secret>—Secret word used for generation of an archive password.</p> <p><i>If a secret word is not indicated when the command is called, the current secret word <secret> is used. It is indicated in the Dr.Web MailDsettings. And if RepackPassword parameter is not available or is set to a value different from HMAC (<secret>), the command will return an error.</i></p>
license	<p>Purpose: Show the information about the currently active license, or get a demo-version license, or get the key file for a license that has already been registered (for example, that has been registered on the company's website).</p> <p>If no options are specified, then the following information is displayed (if you are using a license for the standalone mode):</p> <ul style="list-style-type: none">• License number• Date and time when the license will expire <p>If you are using a license provided to you by a central protection server (for the use of the product in the central protection mode or in the mobile mode), then the following information will be displayed:</p> <p>Arguments: None.</p> <p>Options:</p> <p>--GetRegistered <serial number>—get a license key file for the specified serial number, if the conditions for the provision of a new key file have not been breached (for example, breached by using the product not in the central protection mode, when the license is managed by a central protection server).</p> <p><i>If the serial number is not the one provided for the demo period, you must first register it at the company's website.</i></p>



Command	Description
	<p>For further information about the licensing of Dr.Web products, refer to the Licensing section.</p> <div> To register a serial number, an Internet connection is required.</div>
stat	<p>Purpose: Output statistics about the operation of components that process files (pressing CTRL+C or Q interrupts the statistics display) or about the operation of the network data scanning agent Dr.Web Network Checker.</p> <p>The statistics output includes:</p> <ul style="list-style-type: none">• Name of the component that initiated scanning• PID of the component• Average number of files processed per second during the last minute, 5 minutes, 15 minutes• Usage percentage of the scanned files cache.• Average number of scan errors per second. <p>For the distributed scanning agent, the following information is output:</p> <ul style="list-style-type: none">• List of local components that initiated scanning• List of remote hosts that received files for scanning• List of remote hosts that sent files for scanning <p>For local clients of the distributed scanning agent, their PID and name are specified; for remote clients—address and port of the host.</p> <p>For both clients—local and remote—the following information is output:</p> <ul style="list-style-type: none">• Average number of files scanned per second• Average number of sent and received bytes per second• Average number of errors per second <p>Arguments: None.</p> <p>Options:</p> <p>-n [--netcheck]—Output statistics on operation of the network data scanning agent.</p>



Usage Examples

Usage examples for Dr.Web Ctl (**drweb-ctl**):

1. Object scanning

1.1. Simple Scanning Commands

1. Perform scanning of the `/home` directory with default parameters:

```
$ drweb-ctl scan /home
```

2. Scan paths listed in the `daily_scan` file (one path per line):

```
$ drweb-ctl scan --stdin < daily_scan
```

3. Perform scanning of the boot record on the **sda**:

```
$ drweb-ctl bootscan /dev/sda
```

4. Perform scanning of the running processes:

```
$ drweb-ctl procscan
```

1.2. Scanning of Files Selected by Criteria

Examples for selection of files for scanning are listed below and use the result of the operation of the utility **find**. The obtained list of files is sent to the command **drweb-ctl scan** with the parameter `--stdin` or `--stdin0`.

1. Scan listed files returned by the utility **find** and separated with the NUL (`'\0'`) character:

```
$ find -print0 | drweb-ctl scan --stdin0
```

2. Scan all files in all directories, starting from the root directory, on one partition of the file system:

```
$ find / -xdev -type f | drweb-ctl scan --stdin
```

3. Scan all files in all directories, starting from the root directory, with the exception of the `/var/log/messages` and `/var/log/syslog` files:

```
$ find / -type f ! -path /var/log/messages ! -path /var/log/syslog |  
drweb-ctl scan -stdin
```

4. Scan all files of the *root* user in all directories, starting from the root directory:

```
$ find / -type f -user root | drweb-ctl scan --stdin
```

5. Scan files of the *root* and *admin* users in all directories, starting from the root directory:



```
$ find / -type f \( -user root -o -user admin \) | drweb-ctl scan --stdin
```

6. Scan files of users with UID in the range 1000–1005 in all directories, starting from the root directory:

```
$ find / -type f -uid +999 -uid -1006 | drweb-ctl scan --stdin
```

7. Scan files in all directories, starting from the root directory, with a nesting level not more than five:

```
$ find / -maxdepth 5 -type f | =drweb-ctl scan --stdin
```

8. Scan files in a root directory ignoring files in subdirectories:

```
$ find / -maxdepth 1 -type f | =drweb-ctl scan --stdin
```

9. Scan files in all directories, starting from the root directory, with following all symbolic links:

```
$ find -L / -type f | drweb-ctl scan --stdin
```

10. Scan files in all directories, starting from the root directory, without following symbolic links:

```
$ find -P / -type f | drweb-ctl scan --stdin
```

11. Scan files created not later than May 1, 2017 in all directories, starting with the root directory:

```
$ find / -type f -newermt 2017-05-01 | drweb-ctl scan --stdin
```

1.3. Scanning of Additional Objects

1. Scanning of objects located in the directory `/tmp` on the remote server `192.168.0.1` by connecting to it via SSH as a user `user` with the password `passw`:

```
$ drweb-ctl remotescan 192.168.0.1 /tmp --Login user --Password passw
```

2. Configuration management

1. Display information on a current program package, including information about running processes:

```
$ drweb-ctl appinfo
```

2. Output all parameters from the [Root] [section](#) of the active configuration:

```
$ drweb-ctl cfshow Root
```

3. Set 'No' as the value of the **Start** parameter in the [ClamD] section of the active configuration (this will disable the [Dr.Web ClamD](#) component):

```
# drweb-ctl cfset ClamD.Start No
```



Note that superuser privileges are required to perform this action. To elevate the privileges, you can use the **sudo** command, as shown in the following example:

```
$ sudo drweb-ctl cfset ClamD.Start No
```

4. Forbid the update component to perform an update of files 123.vdb and 567.dws:

```
# drweb-ctl cfset Update.ExcludedFiles -a 123.vdb
# drweb-ctl cfset Update.ExcludedFiles -a 567.dws
```

Note that in this case an option **-a** is used to add a new value to the already existing list of values of the parameter **Update.ExcludedFiles**.

5. Remove a file 123.vdb from the list of files, the update of which is prohibited by the update component:

```
# drweb-ctl cfset Update.ExcludedFiles -r 123.vdb
```

6. Reset the list of files, the update of which is prohibited for the update component, to the default value of:

```
# drweb-ctl cfset Update.ExcludedFiles -e
```

7. Perform force update of anti-virus components of the product:

```
$ drweb-ctl update
```

8. Restart the configuration of components of the installed Dr.Web program package:

```
# drweb-ctl reload
```

Note that superuser privileges are required to perform this action. To elevate the privileges, you can use the **sudo** command, as shown in the following example:

```
$ sudo drweb-ctl reload
```

9. Connect the product to the server of [central protection](#), operating on the 192.168.0.1 host under the condition that a public key of the server is located in file :cskey.pub

Configuration Parameters

The Dr.Web Ctl tool for managing the product from the command line does not have its own section with its parameters in the integrated [configuration file](#) of Dr.Web for UNIX File Servers. It uses the parameters specified in the [Root] [section](#) of the configuration file.



Dr.Web Web Management Interface

The web interface of Dr.Web for UNIX File Servers allows you to:

1. View the current state of the program's components, start or stop some of the components.
2. View the status of updates and start an updating process manually, if required.
3. View the status of the product's license and load a license key, if required.
4. View the list of detected threats and manage quarantined objects (threats detected in local file system via the [Dr.Web File Checker](#) component are displayed only).
5. Edit the settings of the components included in Dr.Web for UNIX File Servers.
6. Connect the program to a central protection server or switch the program's operation into standalone mode.
7. Start an on-demand scanning of local files (including a capability to do it by dragging and dropping files onto the page opened in your browser).

System Requirements of the Web Interface

Correct operation of the web interface is guaranteed for the following web browsers:

- **Internet Explorer**—version 11 and later.
- **Mozilla Firefox**—version 25 and later.
- **Google Chrome**—version 30 and later.

Accessing the Web Interface

To access the web interface, in the browser's address bar type in an address that looks like:

```
https://<host_with_drweb>:<port>/
```

where *<host_with_drweb>* is the IP address or the name of the host where the product containing the web-interface-server component—Dr.Web HTTPD—is running, and *<port>* is the port (on this host) which Dr.Web HTTPD is listening on. To access a product component which operates on the local host, use IP address 127.0.0.1 or the name `localhost`. By [default](#), the *<port>* is 4443.

Thus, to access the web interface on the local host by default, enter the following URL in the browser's address bar:

```
https://127.0.0.1:4443/
```

After connection to the managing server is established, a startup page opens and displays an authentication form. To access management functions, fill in the authentication form by specifying the login and password of a user who has administrative privileges on the host where the product operates.



Main Menu

In the left pane of the web interface, which appears once you have successfully passed authentication, there is a main menu, the items in which allow you to do the following:




- **Main**—opens the [main page](#) which displays the full list of installed components of Dr.Web for UNIX File Servers and their status.
- **Threats**—opens a page which [displays all the threats](#) detected on the server. In this section, you can manage these detected threats (for example, move infected objects to quarantine, rescan, cure or delete detected malicious objects).
- **Settings**—opens a page with the [component settings](#) of Dr.Web for UNIX File Servers installed on the server.
- **Information**—opens a page that shows brief information about the version of this web interface and about the state of virus databases.
- **Help**—opens a new browser tab with help information about the installed components of the product.
- **Scan file**—displays a panel for quick [file scanning](#), which will stay available on top of any opened page of the web interface until you close this panel.
- **Sign out**—ends the current web interface session.


Managing the Components

You can view the list of components included in Dr.Web for UNIX File Servers and manage their operation on the **Main** page.

The listed components of the product are divided into two groups: main components, which monitor threats, and service components, which are responsible for the overall correct operation of the product. The list of main components is displayed as a table in the upper part of the page (the list of components depends on the scope of supply for your product). For each component the following information is specified:

1. **Name of the component.** Click the name to open the [settings page](#) containing the settings for this component;
2. **State of the component.** The state of a component is indicated by a switch icon and by a note about the component's current state. To start a component or to suspend its operation, you only need to click its switch. The possible states of the switch are:


	—the component is disabled and is not used;
	—the component is enabled and works correctly;
	—the component is enabled but is not working because of an error.

If an error occurred in the operation of a component, instead of a note about the component's state an error message is displayed. If you click the  icon, a window will pop



up with detailed information about the error that occurred and with recommendations for resolving this error.

3. **Average load.** The average numbers of files processed by the component per second within the last minute, 5 minutes, 15 minutes respectively are specified (displayed as three numbers separated with forward slashes "/").
4. **Errors.** The average numbers of errors encountered by the component per second within the last minute, 5 minutes, 15 minutes respectively are specified (displayed as three numbers separated with forward slashes "/").

To display a tooltip, place the cursor over the  icon.

Below the table, which provides information about main components, you will find service components (such as the [the scanning engine](#), [the file scanning component](#), etc.) listed as a set of tiles. For each service component, its state and operational statistics are also displayed. To open the settings page of any of these components, click the name of a required component. As a rule, these components are started and stopped automatically when needed. If any of them may be started and stopped manually by the user, then, besides the name and the operational statistics, a switch for starting and stopping the component will be displayed on the tile of any such service component.

The bottom of the page displays whether the virus databases are up to date and [license](#) information. To force a virus database update, click **Update**. By clicking on the **Renew** button (or on the **Activate** button, depending on the current state of your license) you can renew or activate a license by uploading a valid key file that is appropriate for your product to the licensing server.

Threats Management

You can view the list of detected threats and manage the reaction to them on the **Threats** page.

This page contains the full list of threats detected by the components of Dr.Web for UNIX File Servers that monitor and scan the file system. In the upper part of the page, you can see a menu which allows filtering the threats by their category:

- **All**—show all detected threats (including both active and quarantined threats).
- **Active**—show only active threats; i.e. detected but not neutralized yet.
- **Blocked**—show all blocked threats, that is, threats that were not neutralized, but for which the infected objects containing them were blocked (only for file storages monitored by SpIDer Guard for SMB).
- **Quarantined**—show threats that were moved to quarantine.
- **Errors**—show threats that were not processed because of an error.

Just next to each name of a threat category (to its right) in the upper menu, the quantity of detected threats that fall into this category is displayed. The currently selected category, for which the threats belonging to it are currently displayed, is emphasized in a darker font. To display threats of a required category, click the name of the category in the menu.



For each threat, the following information is listed:




- **File**—name of the file that contains a malicious object (file path is not specified).
- **Owner**—name of the user who owns the infected file.
- **Component**—name of the component of Dr.Web for UNIX File Servers that detected the threat.
- **Threat**—name of the threat that was detected in the file (according to the classification used by the Doctor Web company).

For any object selected in the list, the following information is displayed:

- Name of the threat (displayed as a link that opens a page of the Dr.Web virus information library with the threat's description).
- File size, in bytes.
- Name of the component that detected the threat.
- Date and time when the threat was detected.
- Date and time when the threat was last modified.
- Name of the user who owns the infected file.
- Name of the group that includes the file owner.
- Name of the user who uploaded the file to the file server (only for file storages monitored by SpIDer Guard for SMB).
- Identifier that was assigned to the quarantined file containing a threat (if the file was quarantined).
- Full path that points to the original location of the file (where the file was located at the moment of threat detection).

You can select any object in the list by clicking on it. To select multiple objects, set the checkboxes for the corresponding objects. To select all objects or cancel the selection, select the check box in the **File** field in the threat list's header.

To apply actions to objects selected in the list, click the corresponding button on the toolbar, which is located directly above the threat list. The toolbar contains the following buttons (note that some of them can be unavailable depending on the type of selected threats):

	—instructs to remove (i.e. to permanently delete) selected files.
	—instructs to restore selected files from quarantine to their original location.
	<p>—instructs to apply an additional action to selected files (available actions are specified in the drop-down list):</p> <ul style="list-style-type: none">• Quarantine—instructs to put the selected files that contain threats to quarantine• Cure—will attempt to cure the threats



- **Ignore**—instructs to ignore the threats detected in selected files and to remove the threats from the list



Note that managing of threats detected on NSS volumes requires SpIDer Guard for NSS to be installed and started.

If in the settings of SpIDer Guard for NSS—which monitors NSS volumes—*Quarantine* is specified as an action that must be automatically applied to some threat type, the object containing a threat of this type will be placed to quarantine again on attempt to restore this object from quarantine to an NSS volume. For example, the default settings of this monitor move all incurable objects to quarantine. This is why, when any incurable object is restored from quarantine to an NSS volume, this object is automatically returned to quarantine.

You can also filter displayed threats based on a search query. To filter unnecessary threats out and display only those that correspond to the query, use the search box. The box is displayed on the right side of the toolbar and is marked with . To filter the threat list, enter a word in the search box. All threats that do not have the entered word in their name or description, will be hidden (this filtering is not case-sensitive). To clear search results and display the unfiltered list, click in the search box or erase the word.

Managing the Settings

You can view and change current [configuration parameters](#) of the components included in Dr.Web for UNIX File Servers and listed on the [main page](#). For that, open the **Settings** page. On this page you will also be able to switch the program into the *central protection* mode or into the *standalone* mode (for further information about these modes please refer to [Operation Modes](#)).

On the left side of the page, a menu is displayed, which contains the names of all the program's components whose settings can be viewed and adjusted. To view and adjust the settings of any component, first click on the name of a desired in this menu. The name of the component whose settings you are currently viewing and editing will be highlighted in this menu on the left.

- The **Central protection** item in the menu will take you to the [page for managing](#) the central protection mode.
- The **Common settings** item in the menu corresponds to the [settings](#) of the Dr.Web ConfigD component, which is responsible for the overall functioning of the product.

If a component has sections with additional settings apart from the section with its main settings (for example, such sections are available for the Dr.Web ClamD component, which emulates the interface of the **ClamAV®** anti-virus and uses these additional sections to hold individual scanning parameters for different clients that use different connection addresses), then an icon indicating that you can expand/collapse additional sections is displayed to the left of the



component's name. If the icon looks like ▶, additional sections are hidden. If the icon looks like ▼, additional sections are displayed on the menu, one per line. To expand/collapse the list of additional sections, click this expand/collapse icon next to the name of the required component.

- The additional sections with settings are displayed as indented lines. To view or edit parameters of an additional section, click its name in the menu.
- To add an additional subordinate section with settings for a component, if it is allowed, click + to the right of the component's name. In the window that will appear, specify a unique name (tag) for the new subsection and click **OK**. To close the window without creating a subsection, click **Cancel**.
- To delete a subsection for a component, if allowed, click ✖ to the right of the subsection's name (tag). In the window that will appear, confirm that you want to delete the subsection and click **OK**. To close the window without deleting the subsection, click **No**.

At the top of the settings page, you can see a menu that allows you to change the viewing mode. The following modes are available:

- **All**—show the table with all the component's configuration parameters that can be viewed and adjusted.
- **Changed**—show the table with the component's configuration parameters that have values different from the default ones.
- **Ini Editor**—show a text editor with this component's configuration parameters that have values different from the default ones. The displayed text has the same format as the [configuration file](#) (contains `parameter = value` pairs).

You can also filter displayed parameters based on a search query. To filter unnecessary parameters out and display only those that correspond to the query, use the search box. The box is displayed on the right side of the viewing mode menu and is marked with 🔍. To filter the parameter list, enter any word in the search box. All parameters that do not have the entered word in their description, will be hidden (this filtering is not case-sensitive). To clear the search results and display the unfiltered list, click ✖ in the search box or erase the word in it.

Parameters can be filtered out only when they are displayed in tabular form (i.e. in the **All** and **Changed** viewing modes).

Viewing and Editing Component Settings in Tabular Form

When viewing parameters in tabular form (the **All** and **Changed** viewing modes), each table row contains a description of a parameter (on the left) and its current value (on the right). For Boolean parameters (those that have only two available values: "Yes" and "No"), a checkbox is displayed instead of a value (checked means "Yes", unchecked means "No").



When you select to view all parameters (not only those that were changed), the modified (non-default) values are indicated in bold.



The complete parameter list is split into groups (such as **Main**, **Advanced**, etc.). To collapse or expand a group, click on its heading (its name). When a group is collapsed and its parameters are not displayed in the table, the following icon appears to the left of the group's name: ➤. When a group is expanded and the parameters are displayed in the table, the following icon appears to the left of the group's name: ▼.

To adjust a parameter, click its current value in the table (for a Boolean parameter—set or remove a check mark in the corresponding checkbox). If a parameter has a set of predefined values, they will all appear as a drop-down list after you click the current value. If a parameter has a numeric value, an editing box will appear after you click the current value. Specify a required value and press ENTER. The figure below shows examples of how to change parameter values (note that the set of components shown in the figure can differ from the one supplied to you). All changes made to parameter values are immediately applied to the configuration of the corresponding component.

Scanning Engine [ScanEngine]	
All Changed Ini Editor	
▼ General	
MaxForks Maximum number of scanning processes	10
LogLevel Logging level	Info ▼
Log Logging method	Auto
▼ Advanced	
FixedSocketPath Socket path	Not specified
MaxForksPerFile Maximum number of scan forks per one file	5

Figure 3. Component's settings in tabular form

If the parameter expects a string as its value or accepts a list of arbitrary values, a pop-up window will appear once you click on the parameter's current value to edit it. If the parameter accepts a list of values, they will be shown in a multi-line editing box (one value per line) as shown in the figure below. To edit the listed values, you need to change, delete or add any required lines in the editing box.

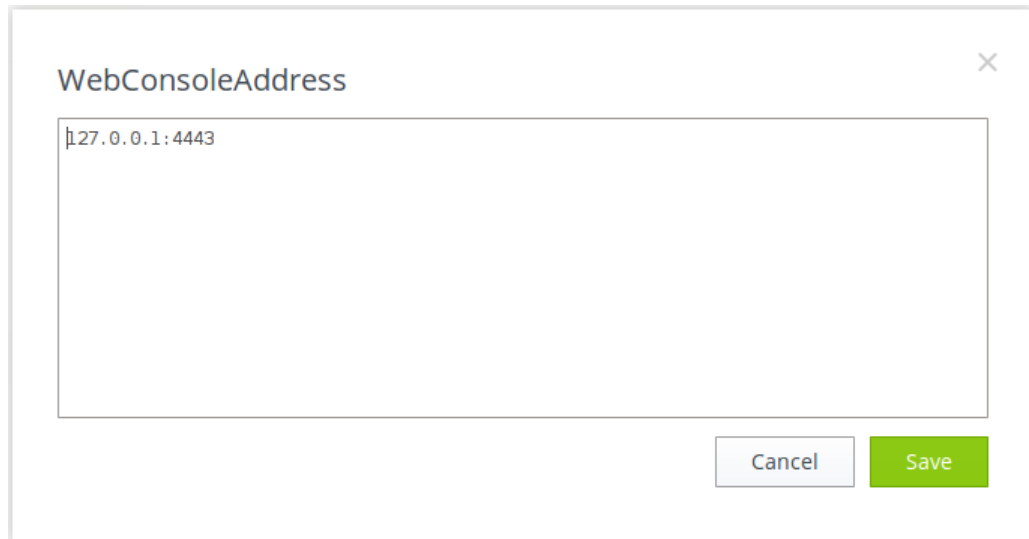


Figure 4. Editing a list of values

After editing the value of a parameter, click **Save** to apply your changes and to close the window. To close the window without applying the changes click **Cancel** or click the **X** icon in the upper right corner of the pop-up window.

Viewing and Editing Components' Settings in a Text Editor

When viewing [parameters](#) in the **Ini Editor** mode, they are displayed in the same format as in the [configuration file](#) of the product (as `parameter = value` pairs), where parameter is a parameter's name that is written directly into the configuration file (into the settings section of the corresponding component). In this mode, only those parameters are displayed whose values differ from the default ones (that is, parameters whose values are emphasized in bold font in the **All** viewing mode). The figure below shows how parameters are displayed in this simple-view textual editor.

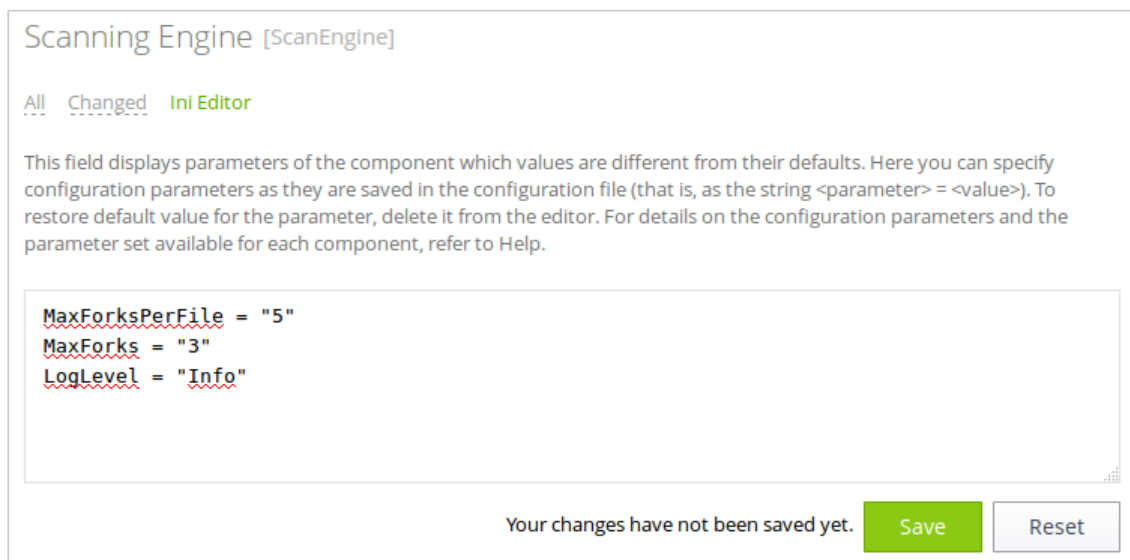


Figure 5. Simple textual settings editor



To make any desired changes, edit the text in this text editor according to the same rules as described for editing the configuration file (this will modify only the section that contains the settings of the component highlighted on the left). If necessary, you can specify a new value for any parameter available for the component. In this case, the value of this parameter changes from its default setting to the value you enter in the editor. If you want to reset the parameter back to its default value, just erase the line containing this parameter in this text editor. If you do so, then, once you save the changes, the parameter will be restored to its default value.

Once you have finished editing parameters' values, click **Save** to apply the changes or click **Cancel** to discard them.



If you click **Save**, the text is validated: the program checks whether all parameters are existent and their set values are valid. In case of an error, the appropriate message will be displayed.

For details on the configuration file, its, and its features that are important for specifying parameter values, refer to [Appendix D. Configuration File](#) section.

Additional Information

- [Configuration parameters](#) of Dr.Web ConfigD (Common settings).
- [Configuration parameters](#) of SpIDer Guard.
- [Configuration parameters](#) of SpIDer Guard for NSS.
- [Configuration parameters](#) of SpIDer Guard for SMB.
- [Configuration parameters](#) of Dr.Web ES Agent.
- [Configuration parameters](#) of Dr.Web Updater.
- [Configuration parameters](#) of Dr.Web ClamD.
- [Configuration parameters](#) of Dr.Web File Checker.
- [Configuration parameters](#) of Dr.Web Scanning Engine.
- [Configuration parameters](#) of Dr.Web Network Checker.
- [Configuration parameters](#) of Dr.Web SNMPD.
- [Configuration parameters](#) of Dr.Web CloudD.
- [Managing the Central Protection](#).

Managing the Central Protection

You can connect Dr.Web for UNIX File Servers to a central protection server or switch back to the standalone mode, thereby disconnecting the product from the central protection server. To open the page where you can manage central protection, chose the item called **Central protection** from the settings menu on the **Settings** page.



To connect Dr.Web for UNIX File Servers to a central protection server or to disconnect from it, use the corresponding checkbox on this page.

Connecting to an Anti-Virus Network

At an attempt to connect to a central protection server a pop-up window will appear on the screen; in this window you need to specify the parameters for connecting to the central protection server.

The screenshot shows a dialog box titled "Set manually" with a close button (X) in the top right corner. Inside the dialog, there are several input fields and buttons:

- A label "Server address and port:" followed by a text input field.
- A label "Server public key file:" followed by a text input field and a "Browse..." button.
- A section header "Authentication (optional)" with a downward arrow.
- A label "Workstation ID:" followed by a text input field.
- A label "Password:" followed by a text input field.
- A checkbox labeled "Connect the workstation as 'newbie'".
- At the bottom, there are two buttons: "Connect" (highlighted in green) and "Cancel".

Figure 6. Connection to the central protection server

In the drop-down list located at the top of the window chose one of the methods for connecting to a central protection server. Three methods are available:

- *Load from file*
- *Set manually*
- *Detect automatically*

If you select the *Load from file* option, then in the corresponding field of this window you will also need to specify a path to a file that contains connection settings. The file is provided by the anti-virus network administrator. If you select the *Set manually* option, you will need to specify the address and the port of the central protection server. For the *Set manually* or *Detect automatically* options, you can also specify the path to the file containing the server's public key (provided by your network administrator or Internet service provider).

If these fields are filled in, then your connection to the central protection server will succeed only if a correct identifier/password pair was entered. If you leave these fields empty, then connection



to the central protection server will succeed only if this connection is approved at the central protection server (either automatically or by the anti-virus network administrator, depending on that server's settings). Additionally, in the **Authentication** section you can specify your login (workstation identifier) and password for authentication on the central protection server, if you know them.

Moreover, you can use the **Connect as a "newbie"** option (to connect as a new user). In this case, if the Newbie mode is allowed on the central protection server for connections from workstations, then the central protection server, after approving this connection, automatically generates a unique identifier/password pair, which will be from this time on used for connecting your computer to the server. Note that in this mode the central protection server generates a new account for your workstation even if your workstation already has another account on the server.



Connection parameters must be specified in strict accordance with the instructions provided by the administrator of your anti-virus network or service provider.

To connect to the server after having specified all the parameters, press the **Connect** button and wait until the connection procedure completes. To close the settings window without connecting to the server, press the **Cancel** button.



Once you have connected Dr.Web for UNIX File Servers to a central protection server, its operation will be managed by the central protection server, until you switch back to the standalone mode. Connection to the central protection server will be established automatically every time when Dr.Web for UNIX File Servers is started.

Scanning local files

The web interface provides a capability to scan any files stored on your local computer (from which you are currently accessing the web interface) to determine whether the files have any malicious content, the scanning is done with the help of the scanning engine that is part of Dr.Web for UNIX File Servers. The files selected for scanning will be uploaded (via the HTTP protocol) to your server on which Dr.Web for UNIX File Servers is running, but after the scanning, even if any threats are found, the files will not be stored on the server, neither will they be moved to quarantine there. The user who sent the files to scan them will only be informed about the result of the scanning.

Opening a panel to scan local files and setting parameters for the scanning

You can select and upload the files that you want to scan via the scanning panel for local files which is displayed when you choose the **Scan file** item in the main menu of the web interface. The launched panel is displayed in the bottom right corner of the web interface. The figure below shows what the scanning panel for local files looks like.

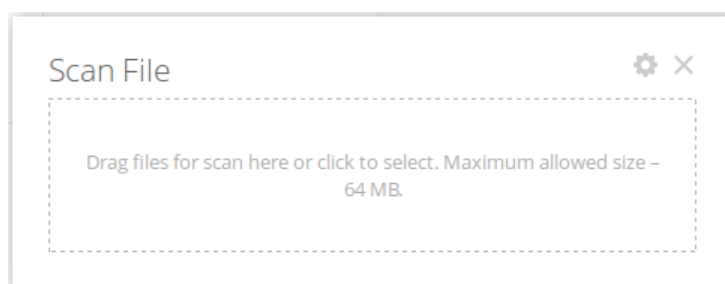




Figure 7. The scanning panel for local files

To close this panel, click the  icon on the panel's top right-hand corner. By clicking the  icon you can display the settings for the scanning of local files: the maximum time to scan a file (which does not include the time it takes to upload the file to your server from your local computer), using the heuristic analysis during the scan, and also the maximum compression ratio for compressed objects and the maximum nesting level for objects packed into containers (such as archives).

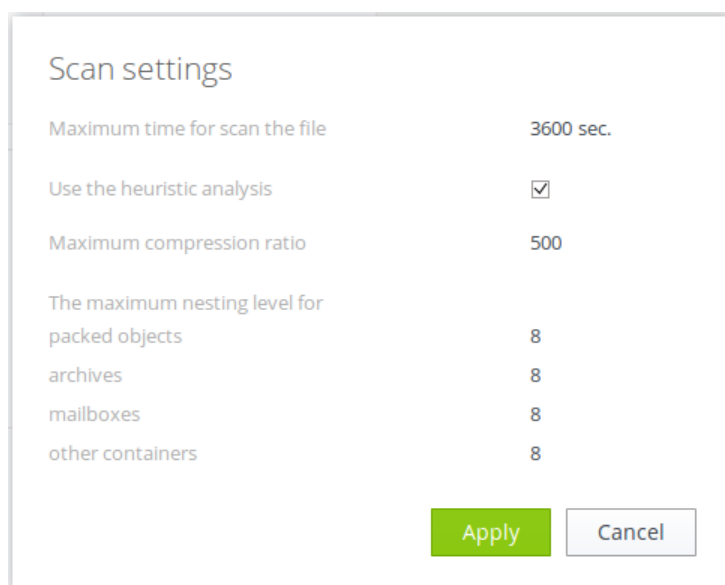


Figure 8. Setting the parameters for the scanning of local files

To apply the changed settings and to return to the file selection mode where you can choose the files to scan, press the **Apply** button. To go back to file selection without applying your changes to the settings, press the **Cancel** button.

Launching the scanning of local files

To select files for scanning and to start their scanning, left-click on the target area that says **Drag files for scan here or click to select**. Upon your click there, a standard file selection window of your operation system's file manager will open. You can choose multiple files at once for scanning. Please, note that you are not allowed to choose directories for scanning. You can also drag selected files with your mouse directly onto the target area of the file scanning panel from the file manager window. Once the files to be scanned have been specified, they will start being uploaded to your server where Dr.Web for UNIX File Servers is installed; and once a file is



uploaded, its scanning starts. During the uploading and scanning of the files the file scanning panel displays the overall progress of the scanning procedure.

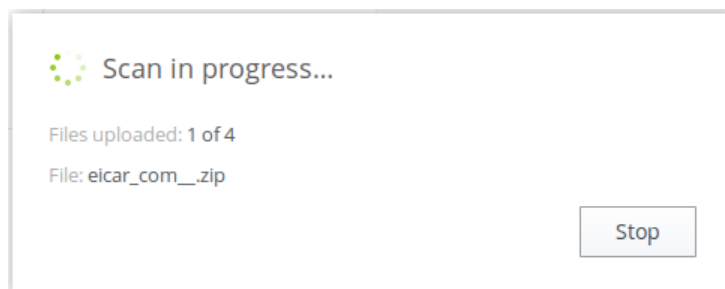


Figure 9. Current progress for the scanning of local files

If necessary, you can abort the scanning by pressing the **Stop** button. Once the scanning is completed, a report about the scanning of the uploaded files will be displayed on the file scanning panel.

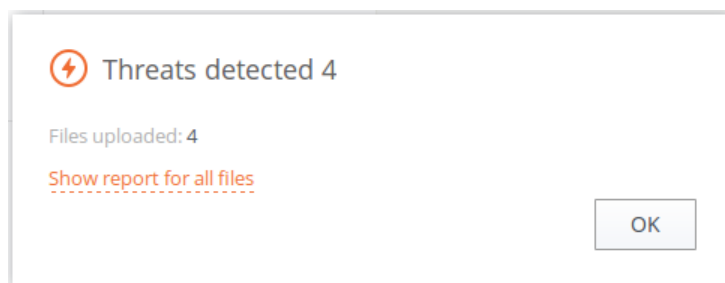


Figure 10. Results for the scanned local files

If multiple files were uploaded, an extended report about the scanning will be available. To see the extended report, click the link that says **Show report for all files**.

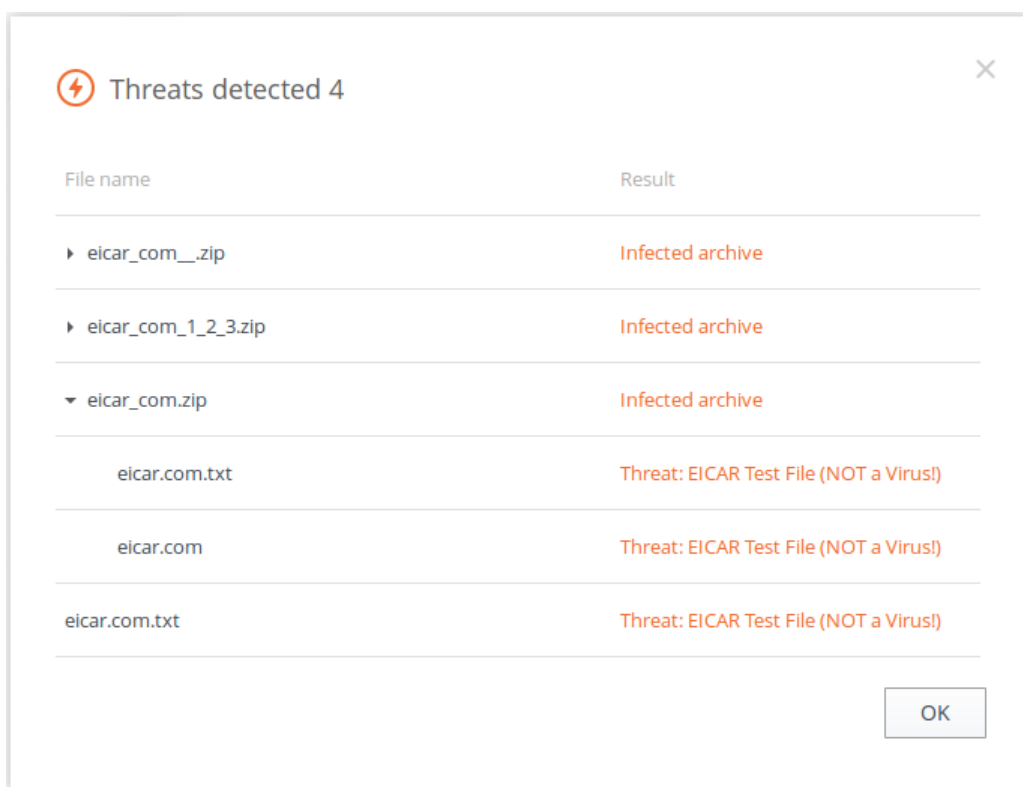


Figure 11. Extended report about the scanned local files

To close the report and to return to the state where the panel allows selecting new files for scanning, press **OK**.



It is possible to start scanning local files (using the current settings for the scanning) even when the file scanning panel is closed. To start uploading and scanning local files, just drag and drop them from the file manager window onto a page of the web interface opened in your browser.



SpIDer Guard



The component is included only in the distributions for **GNU/Linux** OS.

The Linux file system monitor SpIDer Guard is designed for monitoring file activity on **GNU/Linux** file system volumes. The component operates as a resident monitor and controls main file system events related to modification (file creation, opening, closing). When such event is intercepted, the monitor checks whether the file was modified and, if so, the module generates a task for the [Dr.Web File Checker](#) file checker component to scan the modified file by the [Dr.Web Scanning Engine](#) scanning engine.

Moreover, the file system monitor SpIDer Guard detects attempts to run programs from their executables files. If a program in an executable file is detected malicious during scanning, all processes started from this executable file will be forcibly terminated.

Operating Principles

The file system monitor SpIDer Guard operates in user space (*user mode*) using either **fanotify** mechanism or a special Linux *kernel module* (**LKM**—*Linux kernel module*) developed by Doctor Web. It is recommended that you use the *automatic mode* (**Auto**), which will allow the component to define the best operation mode on startup, as not all **Linux** kernel versions support **fanotify** used by the monitor. If the component cannot support the specified integration mode, the components exits after startup. If the auto mode is selected, the component attempts to use the **fanotify** mode and then the **LKM** mode. If neither of these two modes can be used, the component exits.



For some operating systems an already compiled kernel module is supplied together with SpIDer Guard. If a kernel module is not compiled for the operating system that uses SpIDer Guard, use the source code files, provided by Doctor Web to build and install the kernel module manually (for instructions, refer to the [Building kernel module for SpIDer Guard](#) section).

When new or modified files are detected, the monitor sends to the [Dr.Web File Checker](#) file checker component a task to scan these files. The component then initiates their scanning by the [Dr.Web Scanning Engine](#) scanning engine. The operation scheme is shown in the picture below. The scheme of the monitor operation is shown in the Figure below.

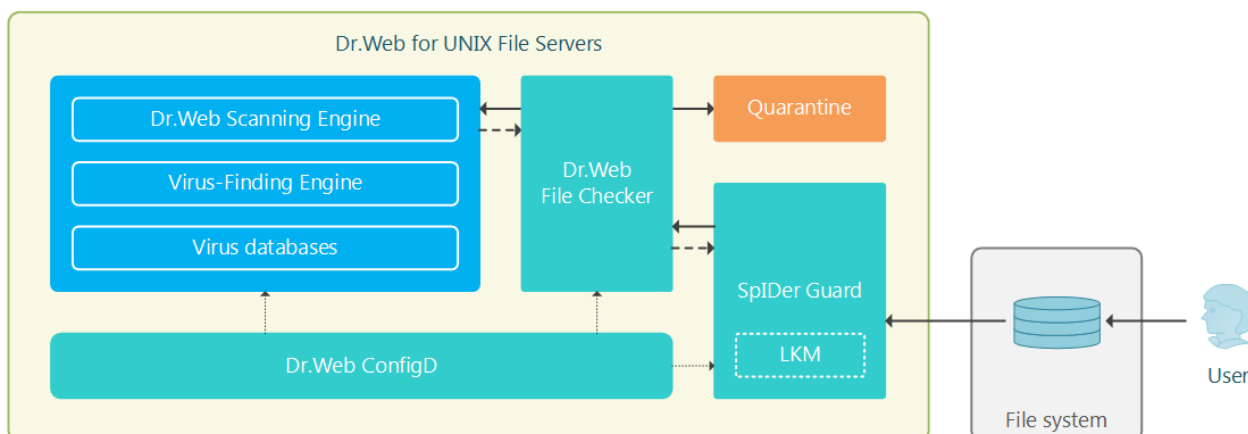


Figure 12. Diagram of the components' operation

To define file system objects that must be monitored, SpIDer Guard uses two configuration parameters:

- **IncludedPath**—paths to be monitored (*monitoring scope*).
- **ExcludedPath**—paths to be excluded from monitoring (*exclusion scope*).

Thus, only those files are monitored which paths are specified in the **IncludedPath** parameter and not specified in the **ExcludedPath** parameter. If a path is specified in both parameters, the **IncludedPath** parameter has higher priority than the other parameter: the objects are monitored by Samba file system monitor SpIDer Guard (for details on the algorithm to define objects to be included or excluded from monitoring, see [below](#)).

Example:

Let us assume that the lists contain the following paths:

```
IncludedPath = /a, /b/c, /d/file1  
ExcludedPath = /b, /d/file1, /b/c/file2
```

Then the file system monitor SpIDer Guard will monitor access to:

- All files in the `/a` directory
- All files in the `/b/c` directory except for `file2`.
- To the `/d/file1` file.

With these settings, the monitor does not control operations with other files.



Note that specifying `/b` path in the **ExcludedPath** list is syntactically valid, but has no effect: files in this directory are not monitored anyway as this path is not within the monitoring scope specified in the **IncludedPath** list.

Specifying exclusion in the **ExcludedPath** list can be useful when, for example, some files are frequently modified, which results in constant repeated scanning of these files and, thus, can increase system load. If it is known with certainty that frequent modification of files in a directory is not caused by a malicious program but is due to operation of a trusted program, you can add



the path to this directory or these files to the list of exclusions. In this case, the file system monitor SpIDer Guard stops responding to modification of these objects, even if they are within the monitoring scope. Moreover, you can add a program to the list of trusted programs (the **ExcludedProc** configuration parameter). In this case, file operations of this program will not cause scanning even if the modified or created files are monitored.

The file system monitor SpIDer Guard automatically detects mounting and demounting of new file system volumes (for example, on USB flash drives, CD/DVD, RAID arrays) and adjusts the list of monitored objects, if required.

Defining Whether an Object is Monitored

To define whether a file system object is to be monitored, the file system monitor SpIDer Guard, when detecting a file operation, does the following:

1. Gets information on the process that performed the file operation. If the executable path of this process (name of the executable file and its full path) is specified in the **ExcludedProc** list, the modified object is not within the monitoring scope and will not be scanned; the procedure ends.
2. Otherwise the monitor gets the full path to the modified object.
3. This path is checked if it is specified in the **IncludedPath** or **ExcludedPath** lists.
4. If the path coincides with one of the items in the **IncludedPath** list, the object is scanned; the procedure ends.
5. If the path coincides with one of the items in the **ExcludedPath** list, the operation is ignored and the object is not scanned; the procedure ends.
6. If the path is not specified in any of the lists, the path is changed to another path which is one level up.
7. If this result path is empty, the procedure ends. Otherwise, the procedure goes to step 3.

The procedure continues until the result path of an iteration coincides with an item of either **IncludedPath** or **ExcludedPath**, or until the system root directory is reached.

Command-Line Arguments

To run SpIDer Guard, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-spider [<parameters>]
```

SpIDer Guard can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h



	Arguments: None.
<code>--version</code>	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: <code>-v</code> Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-spider --help
```

This command outputs short help information on SpIDer Guard.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon at the startup of the operating system. To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-spider**


Configuration Parameters

The component uses configuration parameters which are specified in the `[LinuxSpider]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	Logging level of the component. If the parameter value is not specified, the DefaultLogLevel parameter value from the <code>[Root]</code> section is used. Default value: Notice
Log <i>{log type}</i>	Logging method
ExePath <i>{path to file}</i>	Path to the executable file of the component. Default value: <code><opt_dir>/bin/drweb-spider</code> <ul style="list-style-type: none">• For Linux: <code>/opt/drweb.com/bin/drweb-spider</code>



Start {Boolean}	<p>The component must be launched by the Dr.Web ConfigD configuration daemon.</p> <p>When you specify the <code>Yes</code> value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the <code>No</code> value, it instructs the configuration daemon to terminate the component immediately.</p> <p>Default value: <i>Depends on the product in which the component is supplied and operates.</i></p>
Mode {LKM FANOTIFY AUTO}	<p>Operation mode of the file system monitor SpIDer Guard.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <code>LKM</code>—using the Dr.Web LKM module installed in the operating system kernel (<i>LKM—Linux kernel module</i>).• <code>FANOTIFY</code>—using the fanotify monitoring interface.• <code>AUTO</code>—The best operation mode is set automatically. <div><p>Changing of this parameter value should be done with the extreme caution as various GNU/Linux OS kernels support both operating modes in a different way. It is strongly recommended that you set this parameter value to <code>AUTO</code>, as in this case the best mode will be selected for integration with the file system manager on startup. At that, the component will attempt to enable <code>FANOTIFY</code> mode and, on failure—<code>LKM</code>. If none of the modes can be set, the component exits.</p><hr/><p>If necessary, you can build a Dr.Web LKM module from the source codes and install it, following the instructions in the Building kernel module for SpIDer Guard section.</p></div> <p>Default value: <code>AUTO</code></p>
DebugAccess {Boolean}	<p>Indicates whether detailed messages about access to files are included in the log file on debug level (i.e. when LogLevel = <code>DEBUG</code>).</p> <p>Default value: <code>No</code></p>
ExcludedProc {path to file}	<p>List of processes that are excluded from monitoring. If a file operation was initiated by one of the processes specified here, the modified or created file will not be scanned.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The</i></p>



	<p><i>parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list of processes wget and curl.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[LinuxSpider] ExcludedProc = "/usr/bin/wget", "/usr/bin/curl"</pre>Two strings (one value per a string)<pre>[LinuxSpider] ExcludedProc = /usr/bin/wget ExcludedProc = /usr/bin/curl</pre>Adding values via the command drweb-ctl cfset.<pre># drweb-ctl cfset LinuxSpider.ExcludedProc - a /usr/bin/wget # drweb-ctl cfset LinuxSpider.ExcludedProc - a /usr/bin/curl</pre> <p>Default value: (not set)</p>
<p>ExcludedPath</p> <p><i>{path to file or directory}</i></p>	<p>Path to the object which must be excluded from monitoring. You can specified a directory or file path. If a directory is specified, all directory content will be excluded.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the files <code>/etc/file1</code> and directory <code>/usr/bin</code>.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[LinuxSpider] ExcludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[LinuxSpider] ExcludedPath = /etc/file1 ExcludedPath = /usr/bin</pre>Adding values via the command drweb-ctl cfset.



	<pre># drweb-ctl cfset LinuxSpider.ExcludedPath - a /etc/file1 # drweb-ctl cfset LinuxSpider.ExcludedPath - a /usr/bin</pre> <p><i>Note that symbolic links here have no effect as only the direct path to a file is analyzed when scanning.</i></p> <p>Default value: /proc, /sys</p>
IncludedPath <i>{path to file or directory}</i>	<p>Path to the object which must be monitored and scanned upon any file event. You can specified a directory or file path. If a directory is specified, all directory content will be scanned, if the paths are not specified in the ExcludedPath list.</p> <p>Note that this parameter has higher priority than the ExcludedPath parameter of the same section; that is, if the same object (file or directory) is specified in both parameter values, this object will be scanned upon any file event.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the files /etc/file1 and directory /usr/bin.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[LinuxSpider] IncludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[LinuxSpider] IncludedPath = /etc/file1 IncludedPath = /usr/bin</pre>Adding values via the command drweb-ctl cfset.<pre># drweb-ctl cfset LinuxSpider.IncludedPath - a /etc/file1 # drweb-ctl cfset LinuxSpider.IncludedPath - a /usr/bin</pre><p><i>Note that symbolic links here have no effect as only the direct path to a file is analyzed when scanning.</i></p><p>Default value: /</p>
OnKnownVirus	Action applied by Dr.Web for UNIX File Servers to a known threat (virus, etc.) detected by using signature analysis during the scanning



{action}	initiated by SpIDer Guard. Possible values: <i>Cure, Quarantine, Delete</i> Default value: <i>Cure</i>
OnIncurable {action}	Action applied by Dr.Web for UNIX File Servers to an incurable threat (that is, an attempt to apply <i>Cure</i> failed) detected during the scanning initiated by SpIDer Guard. Possible values: <i>Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnSuspicious {action}	Action applied by Dr.Web for UNIX File Servers to an unknown threat (or suspicious objects) detected by using heuristic analysis during the scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnAdware {action}	Action applied by Dr.Web for UNIX File Servers to adware detected during the scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnDialers {action}	Action applied by Dr.Web for UNIX File Servers to a dialer detected during the scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnJokes {action}	Action applied by Dr.Web for UNIX File Servers to joke detected during the scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnRiskware {action}	Action applied by Dr.Web for UNIX File Servers to riskware detected during the scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnHacktools {action}	Action applied by Dr.Web for UNIX File Servers to a hacktool (tool for remote administration, Trojan, etc.) detected during scanning initiated by SpIDer Guard. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>



ScanTimeout <i>{time interval}</i>	<p>Timeout for scanning one file initiated by SpIDer Guard.</p> <p><i>A value in the range from 1s to 1h can be specified</i></p> <p>Default value: 30s</p>
HeuristicAnalysis <i>{On Off}</i>	<p>Indicates whether heuristic analysis is used for detection of unknown threats during file scanning initiated by SpIDer Guard. Heuristic analysis provides higher detection reliability but, at the same time, it increases time of virus scanning.</p> <p><i>Action applied to threats detected by heuristic analyzer is specified as the OnSuspicious parameter value.</i></p> <p>Allowed values:</p> <ul style="list-style-type: none">• On—instructs to use heuristic analysis when scanning.• Off—instructs not to use heuristic analysis. <p>Default value: On</p>
PackerMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning packed objects. All objects at a deeper nesting level are skipped during file scanning initiated by SpIDer Guard.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>
ArchiveMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning archives. All objects at a deeper nesting level are skipped during file scanning initiated by SpIDer Guard.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 0</p>
MailMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning email messages and mailboxes. All objects at a deeper nesting level are skipped during file scanning initiated by SpIDer Guard.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 0</p>
ContainerMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning other containers (for example, HTML pages). All objects at a deeper nesting level are skipped during file scanning initiated by SpIDer Guard.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>



MaxCompressionRatio <i>{integer}</i>	<p>Maximum compression ratio of scanned objects (ratio between the uncompressed size and compressed size). If the ratio of an object exceeds the limit, this object is skipped during file scanning initiated by SpIDer Guard.</p> <p><i>The compression ratio must not be smaller than 2.</i></p> <p>Default value: 500</p>
--	---

File System Monitoring Setting

For instruction for configuring the file system monitoring, refer to the [File System Monitoring Setting](#).

Use kernel module for SpIDer Guard

If the operating system does not support the **fanotify** mechanism used by SpIDer Guard for monitoring actions on file system objects, it can use a special loadable module in kernel space (additionally, the kernel module can be used when the **fanotify** mechanism is implemented with restrictions on access to the file system as, for example, in systems with [mandatory access model](#)).

By default, SpIDer Guard is supplied with a completely built loadable kernel module for all operating systems listed in the [System Requirements](#) section. In addition, you can build a loadable kernel module manually using the source codes supplied with SpIDer Guard in a `tar.bz2` archive.



The loadable kernel module, used by SpIDer Guard, is intended for operation with **GNU/Linux** kernels 2.6.* and newer.

The archive with source codes is located in the `share/drweb-spider-kmod/src/` subdirectory of the Dr.Web for UNIX File Servers base directory `<opt_dir>` (for **Linux**: `/opt/drweb.com`). The archive's name is as follows: `drweb-spider-kmod-<version>-<date>.tar.bz2`.

The `drweb-spider-kmod` directory also contains the `check-kmod-install.sh` script. Run the script to check whether the used OS supports kernel versions included in the product. If not, a message prompting to manually build the module is displayed on the screen.

If the `drweb-spider-kmod` directory is missing at the specified path, install the `drweb-spider-kmod` package (from [repository](#) or [using custom installation](#) from universal package, depending on the [method](#) you selected to install the product).



To build the loadable kernel module manually from the source code files, administrative (*root*) privileges are required. For that purpose, you can use the **su** command to switch to another user or the **sudo** command to build the module as a different user.

Building the Kernel Module

1. Unpack the archive with source codes to any directory. For example, the following command

```
# tar -xf drweb-spider-kmod-<version>-<date>.tar.bz2
```

unpacks the source codes to the created directory. This directory has the archive's name and is created in the same location where the archive resides.

2. Go to the created directory and execute the following command:

```
# make
```

If an error occurs during the **make** command execution, resolve the issue (see [below](#)) and restart compilation.

3. After successful execution of the **make** command, enter the following commands:

```
# make install  
# depmod
```

4. After the kernel module is successfully compiled and registered on the system, perform additional configuration of SpIDer Guard. Set the component to operate with the kernel module by executing the following command:

```
# drweb-ctl cfset LinuxSpider.Mode LKM
```

It is also possible to specify **AUTO** instead of **LKM**. In this case, SpIDer Guard will automatically try to use either the kernel module or the **fanotify** monitoring interface. For details, type the following command:

```
$ man 1 drweb-spider
```

Possible Build Errors

While the **make** command is being executed, errors may occur. If so, check the following:

- To ensure successful building of the module, **Perl** and **GCC** are required. If they are missing on the system, install them.
- On certain OSes, you may need to install the `kernel-devel` package before starting the procedure.
- On certain operating systems, the procedure can fail because the path to the directory with source codes was incorrectly defined. If so, specify the **make** command with the `KDIR=<path to kernel source codes>` parameter. Typically, the source codes are located in the `/usr/src/kernels/<kernel version>` directory.



The kernel version returned by the **uname -r** command can differ from the directory name *<kernel version>*.



SpIDer Guard for SMB

SpIDer Guard for SMB is a monitor of shared file system directories used by SMB file server **Samba**. This component is designed to monitor actions applied to files in **Samba** shared directories. It operates as a resident monitor and controls basic actions in the protected file system (creation, opening, closing, and read or write operations). Once the component intercepts such operation, it checks whether the file was modified and if so, a task to scan the file is created and sent to the [Dr.Web File Checker](#) file scanner. If the file requires scanning, Dr.Web File Checker initiates the scanning by the [Dr.Web Scanning Engine](#) scanning engine. In case the file contains a threat, it is blocked for access for a period specified in settings until the threat is neutralized. Also the component settings can indicate file blocking in case of a scanning error (including cases when there is no valid license).



To avoid conflicts between SpIDer Guard and SpIDer Guard for SMB, which can occur in the process of scanning the files located in the shared directories of **Samba**, it is recommended that you additionally [configure](#) SpIDer Guard by performing one of the following actions:

- Add **Samba** shared directories to the exclusion scope (specify these directories in the **ExcludedPath** parameter).
- Add the **Samba** process (**smbd**) to the list of ignored processes (specify **smbd** in the **ExcludedProc** parameter).



The SpIDer Guard for SMB monitor uses a special VFS SMB module for the integration with the **Samba** server. With SpIDer Guard for SMB, several versions of this module are supplied. They are built for various versions of **Samba**. However, the supplied versions of the VFS SMB module may be incompatible with the version of **Samba** installed on your file server. It may occur, for example, if your **Samba** server uses the `CLUSTER_SUPPORT` option.

If VFS SMB modules are incompatible with the **Samba** server, build the VFS SMB module for your **Samba** server from the supplied source codes manually (including the compatibility with the `CLUSTER_SUPPORT` option if necessary). The procedure of building the VFS SMB module from the supplied source code files is described in the [Building the VFS SMB Module](#) section.

Operating Principles

SpIDer Guard for SMB operates in daemon mode (usually it is started by the [Dr.Web ConfigD](#) configuration daemon on system startup). After startup, the component operates as a server to which special plug-ins are connected (VFS SMB modules) that operate on the **Samba** server side and monitors user activity in shared directories. When a new or modified file is found on a volume, the monitor instructs the [Dr.Web File Checker](#) file checker to scan the file. Monitor operation scheme is shown in the figure below.

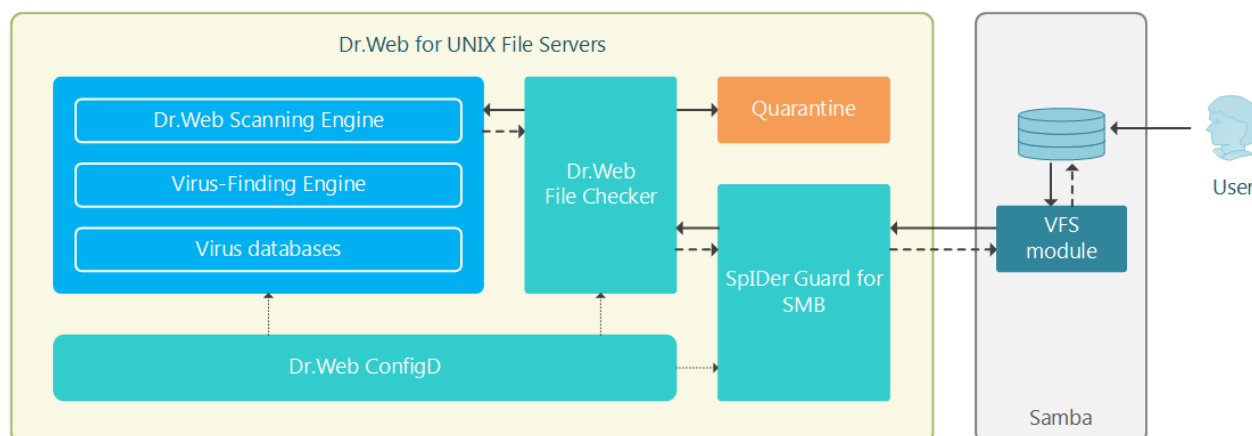


Figure 13. Diagram of the components' operation

If a file scanned at request of the monitor is infected with an incurable threat or with a threat for which the *"Block"* action is specified, the monitor instructs the VFS SMB module controlling the corresponding shared directory to block this file (that is, to prevent users from reading, editing, and running the file). A text file is also created next to the blocked object, if this setting is not disabled. The created text file describes the reason why the object was blocked. It is necessary to avoid the *"unexpected disappearance"* of the file to which the [action](#) *"Delete"* or *"Move to quarantine"* was applied. Thus, it prevents users from multiple attempts to recreate the moved or deleted file. Moreover, this text file also notifies the user that the computer may be infected with a malicious program. If informed on this, the user can start anti-virus scanning of the computer and neutralize local detected threats. Additionally file (depending on the value of the corresponding configuration parameter) can be blocked upon the scanning error, including the case when there is no valid license, which provides operation of SpIDer Guard for SMB.

You can disable monitoring of the specified files and directories stored in controlled shared directories of the **Samba** server. It can be useful when, for example, some files are frequently modified, which results in constant repeated scanning of these files and, thus, can increase system load. If it is known with certainty that frequent modification is typical for these files in the file server's storage, it is recommended that you add them to the list of exclusions. In this case, the monitor stops responding to modification of these objects and their scanning is not initiated.

To distinguish between directories that are to be monitored and the exclusions, the file storage monitor for **Samba**—SpIDer Guard for SMB—uses two configuration parameters:

- **IncludedPath**—paths to be monitored (*"monitoring scope"*).
- **ExcludedPath**—paths to be excluded from monitoring (*"exclusion scope"*).

Normally, as the monitoring scope, the monitor uses the entire shared directory. If you specify different monitoring and exclusion scopes, only those files in shared directory are monitored whose paths are not specified in the **ExcludedPath** parameter or are specified in the **IncludedPath** parameter. If a path is specified in both parameters, the **IncludedPath** parameter has higher priority than the other one: the objects in the included path will be monitored by the Samba shared directories monitor—AppendixesSpIDer Guard for SMB. Thus, use the **IncludedPath** parameter to add some files and directories for monitoring if they are located in the exclusion scope.



You can specify different protection parameters for different **Samba** shared directories monitored by SpIDer Guard for SMB, including different monitoring and exclusion scope as well as reaction to detected threats. For that purpose, in the configuration section of SpIDer Guard for SMB, specify individual settings for VFS SMB modules that control shared directories.

Command-Line Arguments

To launch SpIDer Guard for SMB from the command line of the operating system, the following command is used:

```
$ <opt_dir>/bin/drweb-smbspider-daemon [<parameters>]
```

SpIDer Guard for SMB can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-smbspider-daemon --help
```

This command outputs short help information on SpIDer Guard for SMB.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon at the startup of the operating system. To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the drweb-ctl [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-smbspider-daemon**



Configuration Parameters

The component uses configuration parameters which are specified in the [SMBSpider] section of the integrated [Appendix D. Configuration File](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-smb spider-daemon</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-smb spider-daemon• For FreeBSD: /usr/local/libexec/drweb.com/bin/drweb-smb spider-daemon
Start <i>{Boolean}</i>	<p>The component must be launched by the Dr.Web ConfigD configuration daemon.</p> <p><i>When you specify the Yes value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the No value, it instructs the configuration daemon to terminate the component immediately.</i></p> <p>Default value: No</p>
SambaChrootDir <i>{path to directory}</i>	<p>Defines the path to the root directory of the SMB file storage (can be redefined by the file server with the help of the chroot restriction).</p> <p>Used as a prefix inserted at the beginning of all paths to files and directories residing in the file server's storage and describes the path relative to the root of the local file system.</p> <p><i>If not specified, the path to the file system root is used /.</i></p> <p>Default value: (not specified)</p>
SmbSocketPath <i>{path to file}</i>	<p>Path to the socket file which enables interaction between SpIDer Guard for SMB and VFS SMB modules.</p> <p><i>The path is always relative and is a supplement for the path specified as the SambaChrootDir parameter value (if the SambaChrootDir parameter is empty, than the path to the file system root is supplemented /).</i></p>



	Default value: <code>var/run/.com.drweb.smb_spider_vfs</code>
ActionDelay <i>{time interval}</i>	<p>Delay time between the moment when a threat is detected and the moment when SpIDer Guard for SMB applies the action specified for this threat type. During this time period, the file is blocked.</p> <p>Default value: 24h</p>
MaxCacheSize <i>{size}</i>	<p>Size of cache used by VFS SMBmodules to store data on scanned files in monitored SMB directories.</p> <p><i>If 0 is specified, data is not cached.</i></p> <p>Default value: 10mb</p>
[*] ExcludedPath <i>{path to file or directory}</i>	<p>Path to the shared directory object which must be skipped during scanning. You can specify a directory or file path. It is also possible to use file masks (which contain the characters '?' and '*', as well as character classes '[]', '[!]', '[^]').</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the files <code>/etc/file1</code> and directory <code>/usr/bin</code>.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[SMBSpider] ExcludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[SMBSpider] ExcludedPath = /etc/file1 ExcludedPath = /usr/bin</pre>Adding values via the command drweb-ctl <code>cfset</code>.<pre># drweb-ctl cfset SMBSpider.ExcludedPath - a /etc/file1 # drweb-ctl cfset SMBSpider.ExcludedPath - a /usr/bin</pre> <p><i>If a directory is specified, all directory content will be skipped.</i></p> <p>Default value: (not specified)</p>
[*] IncludedPath <i>{path to file or directory}</i>	<p>Path to the shared directory object, which must be scanned. You can specify a directory or file path. It is also possible to use file masks (that contain the characters '?' and '*', as well as character classes '[]', '[!]', '[^]').</p>




	<p>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</p> <p>Example: Add to the list the files <code>/etc/file1</code> and directory <code>/usr/bin</code>.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[SMBSpider] IncludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[SMBSpider] IncludedPath = /etc/file1 IncludedPath = /usr/bin</pre>Adding values via the command <code>drweb-ctl cfset</code>.<pre># drweb-ctl cfset SMBSpider.IncludedPath - a /etc/file1 # drweb-ctl cfset SMBSpider.IncludedPath - a /usr/bin</pre> <p>If a directory is specified, all directory content will be scanned.</p> <p>Note that this parameter has higher priority than the ExcludedPath parameter (see above); that is, if the same object (file or directory) is specified in both parameter values, this object will be scanned.</p> <p>Default value: (not specified)</p>
<p>[*] AlertFiles <i>{Boolean}</i></p>	<p>Indicates whether a text file is created next to an object blocked by SMB directory monitor as malicious. The created text file describes the reason why the object was blocked. The created file will be named as <code><name of the blocked file>.drweb.alert.txt</code>.</p> <p>Allowed values:</p> <ul style="list-style-type: none">Yes—Files are created.No—Files are not created. <p>Default value: Yes</p>
<p>[*] OnKnownVirus <i>{action}</i></p>	<p>Action applied by Dr.Web for UNIX File Servers to a known threat (virus, etc.) detected by using signature analysis during the scanning initiated by SpIDer Guard for SMB.</p> <p>Allowed values: Block, Cure, Quarantine, Delete</p> <p>Default value: Cure</p>



<code>[*] OnIncurable</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to an incurable threat (that is, an attempt to apply Cure failed) detected during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Block, Quarantine, Delete</i> Default value: <i>Quarantine</i>
<code>[*] OnSuspicious</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to an unknown threat (or suspicious objects) detected by using heuristic analysis during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Quarantine</i>
<code>[*] OnAdware</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to adware detected during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Pass</i>
<code>[*] OnDialers</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to a dialer detected during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Pass</i>
<code>[*] OnJokes</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to joke detected during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Pass</i>
<code>[*] OnRiskware</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to riskware detected during the scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Pass</i>
<code>[*] OnHacktools</code> <code>{action}</code>	Action applied by Dr.Web for UNIX File Servers to a hacktool (tool for remote administration, Trojan, etc.) detected during scanning initiated by SpIDer Guard for SMB. Allowed values: <i>Pass, Block, Quarantine, Delete</i> Default value: <i>Pass</i>
<code>[*] BlockOnError</code> <code>{Boolean}</code>	Indicates whether SpIDer Guard for SMB should block access to a file if an attempt to cure it resulted in an error, or is there is no valid license, which allows the SpIDer Guard for SMB monitor to scan files.



	<div> When there is no valid license, if this parameter is set to Yes, SpIDer Guard for SMB will block all files moved to the shared directory it protects.</div> <p>Allowed values:</p> <ul style="list-style-type: none">• Yes—block access to a file.• No—access to a file is not blocked. <p>Default value: Yes</p>
<p>[*] ScanTimeout</p> <p><i>{time interval}</i></p>	<p>Timeout for scanning one file initiated by SpIDer Guard for SMB.</p> <p><i>A value in the range from 1s to 1h can be specified</i></p> <p>Default value: 30s</p>
<p>[*] HeuristicAnalysis</p> <p><i>{On Off}</i></p>	<p>Indicates whether heuristic analysis is used for detection of unknown threats during the scanning initiated by SpIDer Guard for SMB. Heuristic analysis provides higher detection reliability but, at the same time, it increases time of virus scanning.</p> <p><i>Action applied to threats detected by heuristic analyzer is specified as the OnSuspicious parameter value.</i></p> <p>Allowed values:</p> <ul style="list-style-type: none">• On—instructs to use heuristic analysis when scanning.• Off—instructs not to use heuristic analysis. <p>Default value: On</p>
<p>[*] PackerMaxLevel</p> <p><i>{integer}</i></p>	<p>Maximum nesting level when scanning packed objects. All objects at a deeper nesting level are skipped during the scanning initiated by SpIDer Guard for SMB.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>
<p>[*] ArchiveMaxLevel</p> <p><i>{integer}</i></p>	<p>Maximum nesting level when scanning archives. All objects at a deeper nesting level are skipped during the scanning initiated by SpIDer Guard for SMB.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 0</p>
<p>[*] MailMaxLevel</p> <p><i>{integer}</i></p>	<p>Maximum nesting level when scanning email messages and mailboxes. All objects at a deeper nesting level are skipped during the scanning initiated by SpIDer Guard for SMB.</p>



	<p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>
<p>[*] ContainerMaxLevel <i>{integer}</i></p>	<p>Maximum nesting level when scanning other containers (for example, HTML pages). All objects at a deeper nesting level are skipped during the scanning initiated by SpIDer Guard for SMB.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>
<p>[*] MaxCompressionRatio <i>{integer}</i></p>	<p>Maximum compression ratio of scanned objects (ratio between the compressed size and uncompressed size). If the ratio of an object exceeds the limit, this object is skipped during the scanning initiated by SpIDer Guard for SMB.</p> <p><i>The compression ratio must not be smaller than 2.</i></p> <p>Default value: 500</p>

Customizing Monitoring Settings

You can specify a different tag for each VFS SMB module which monitors each shared directory (file storage). You can do it in the configuration file of SMB server **Samba** (typically, this is `smb.conf` file). Unique tags for VFS SMB modules in `smb.conf` file are specified as follows:

```
smb_spider:tag = <share name>
```

where `<share name>` is a unique tag assigned to a VFS SMB module, which controls some shared directory, by the **Samba** server.

If a VFS SMB module has a unique tag `<share name>`, you can create a separate section in the configuration file of Dr.Web for UNIX File Servers in addition to `[SMBSpider]`. The created section will store all configuration parameters for scanning a particular file storage protected by this VFS SMB module. The name of this section should look as follows:

```
[SMBSpider.Share.<share name>].
```

Sections created for VFS SMB modules can contain parameters indicated with asterisk "[*]" in the abovementioned table. Other parameters cannot be specified in such individual sections as the parameter values configure operation of all VFS SMB modules operating with SMB directories monitor SpIDer Guard for SMB.

VFS SMB module uses parameter values from the general section

`[SMBSpider.Share.<share name>]` if these parameters are not specified in the individual section `[SMBSpider]`, created for this module. Thus, if no individual section, indicated with a tag, is created, all VFS SMB modules use the same parameters for monitoring shared directories. If you delete some parameter from the `[SMBSpider.Share.<share name>]` section, the



parameter value for this section (and for the corresponding shared directory with *<share name>*) will be taken from the “parent” parameter with the same name from the general [SMBSpider] section; the default parameter value is not used in this case.

To add new section for the shared **Samba** directory with a tag *<share name>* using the [Dr.Web Ctl](#) command-line tool for Dr.Web for UNIX File Servers management (it is run by `drweb-ctl` command-line tool for managing the solution from the command line Dr.Web Ctl (it is run by `drweb-ctl` command), use the command

```
# drweb-ctl cfset SmbSpider.Share -a <share name>
```

Example:

```
# drweb-ctl cfset SmbSpider.Share -a BuhFiles
# drweb-ctl cfset SmbSpider.Share.BuhFiles.OnAdware Quarantine
```

The first command adds the [SMBSpider.Share.BuhFiles] section to the configuration file; the second, changes **OnAdware** parameter value, so that the added section contains all parameters, marked with the “[*]” symbol in the abovementioned table, at that, values for all parameters specified in the command, except **OnAdware**, coincide with parameter values from the general [SMBSpider] section.

Integration with Samba File Server

For the instructions for integration with **Samba** file server, refer to the [Integration with Samba File Server](#) section.

Building the VFS SMB Module

If it is detected during the Dr.Web for UNIX File Servers installation that the **Samba** version used on your file server is incompatible with all supplied versions of the VFS SMB modules for SpIDer Guard for SMB, you need to build the VFS SMB module manually using the supplied source codes.

The source codes of the VFS SMB module for SpIDer Guard for SMB are supplied in the additional package named `drweb-smbspider-modules-src` and are packed in the `tar.gz` archive. The `drweb-smbspider-modules-src` package contains a `drweb-smbspider-11.0.src.tar.gz` archive with the source codes. This archive is placed into `/usr/src/` directory. If the archive is absent in this directory, install the package with the source codes manually (from [repository](#) or [by custom installation](#) from the universal package, depending on the [way](#), of the product installation).



Beside the source codes of the VFS SMB module used by SpIDer Guard for SMB, you will also need the source codes of the **Samba** version installed on your file server. If you do not have the source codes, download them from the developer's source, for example <https://www.samba.org/samba/download/>. To determine which **Samba** version is installed on your file server, use the following command:

```
$ smbd -V
```



To build the VFS SMB module for SpIDer Guard for SMB, the source codes of the actual version of **Samba** installed on your server must be used. Otherwise, correct operation of the SpIDer Guard for SMB is not guaranteed.

To build the module manually from the source codes, administrative (root) privileges are required. For that purpose, you can use the **su** command to switch to another user or the **sudo** command to build the module as a different user.

Building the VFS SMB Module

1. Unpack the archive with the VFS SMB module source codes to any directory. For example, the following command

```
# tar -xf drweb-smbspider-11.0.src.tar.gz
```

unpacks the source codes to the created directory. This directory has the archive's name and is created in the same location where the archive resides.

2. Determine the version of **Samba** installed on your server and download its source codes if necessary.
3. Determine if the version of **Samba** installed on your server uses the `CLUSTER_SUPPORT` option. To do that, use the following command:

```
$ smbd -b | grep CLUSTER_SUPPORT
```

If the output contains the `CLUSTER_SUPPORT` string, the version of **Samba** installed on your server uses the `CLUSTER_SUPPORT` option.

4. Go to the directory with the **Samba** source codes, perform configuration (`./configure`) and then building (**make**) of the server. When configuring, define the correct value of the option responsible for `CLUSTER_SUPPORT`. To learn to configure and build **Samba**, refer to developer's official documentation, for example at <https://www.samba.org/samba/docs/>.



Note that building **Samba** from the source code files is necessary only for the correct building of the VFS SMB module used by SpIDer Guard for SMB in the next step. You do not have to replace **Samba** already installed on your server with the new binaries built from the source codes.



5. When you have successfully built **Samba**, go to the directory with the VFS SMB module source codes and issue the following command:

```
# ./configure --with-samba-source=<path to dir with Samba src> && make
```

where *<path to dir with Samba src>* is the path to the directory where **Samba** was built in the previous step.

6. When the VFS SMB module is successfully built, copy the `libsmb_spider.so` binary file from the `.libs` subdirectory (created while building) to the directory for the VFS modules of installed **Samba** (by default, for **GNU/Linux**, it is `/usr/lib/samba/vfs`) and rename it to `smb_spider.so`. To do that, use the following command:

```
# cp ../libs/libsmb_spider.so /usr/lib/samba/vfs/smb_spider.so
```

7. After copying of the built VFS SMB module, you can delete the directories where the VFS SMB module and the **Samba** server were built.

After that, it is necessary to complete integration between Dr.Web for UNIX File Servers and the **Samba** server the way it is described in the [corresponding](#) section in Administrator Manual. Please note that in this case, in the first step of the integration, it is not necessary to create the `smb_spider.so` symbolic link in the directory of the VFS modules of the installed **Samba** server.



SpIDer Guard for NSS



The component is included only in the distributions designed for **GNU/Linux** OSes. It can operate only on **Novell Open Enterprise Server** SP2 based on **SUSE Linux Enterprise Server** 10 SP3 and newer.

NSS volumes monitor SpIDer Guard for NSS is designed for monitoring file activity on NSS (**Novell Storage Services**) file system volumes. The component operates in daemon mode and controls main file system events related to modification (creation, opening, closing). When such event is intercepted, the monitor checks whether the file content was modified and, if so, the monitor generates a task for [Dr.Web File Checker](#) to scan the modified content.

Operating Principles

The SpIDer Guard for NSS monitor operates as a daemon (usually it is started by the [Dr.Web ConfigD](#) configuration daemon at the startup of the operating system). This monitor controls only the volumes which are specified in its [settings](#) (parameters **NssVolumesMountDir** and **ProtectedVolumes**). The monitor does not detect automatically, when a new NSS file system volume is mounted or unmounted. When a new or modified file is found on the NSS volumes, the monitor instructs the [Dr.Web Scanning Engine](#) scanning engine to scan it. Another feature of the SpIDer Guard for NSS monitor is that it manages its own, separate, quarantine for threats detected on NSS volumes. A diagram of the monitor's operation is shown in the figure below.

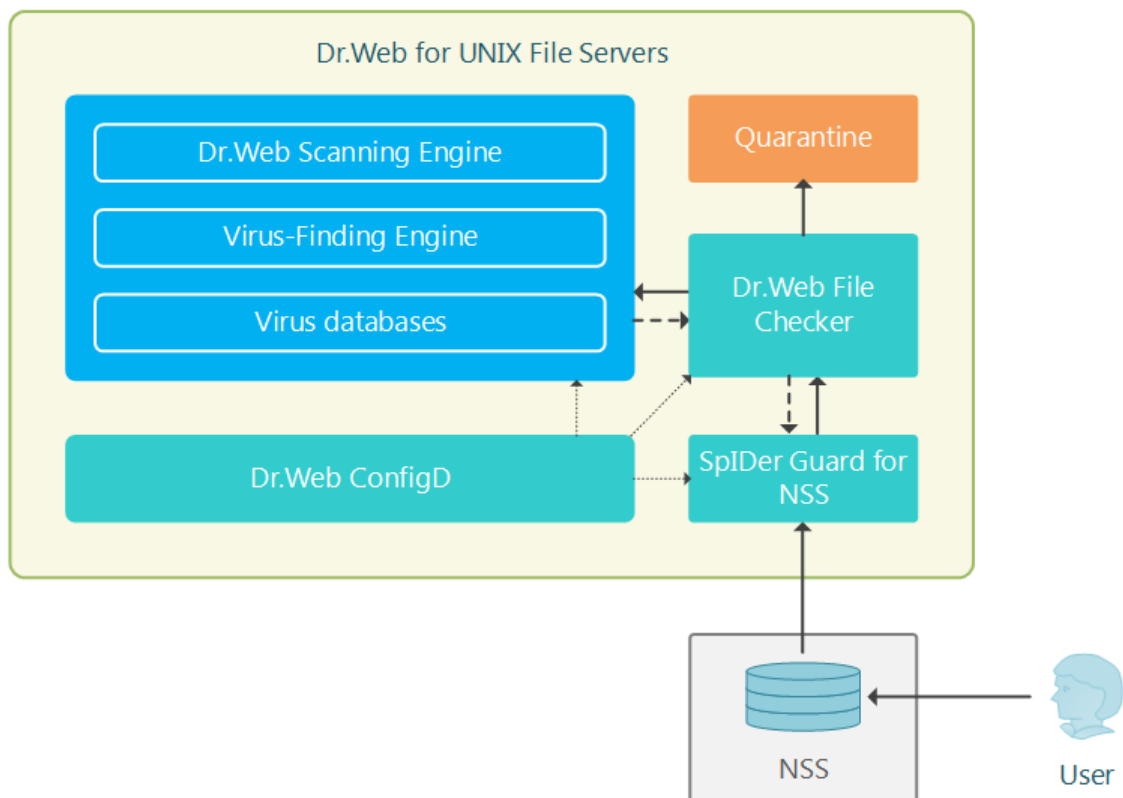


Figure 14. Diagram of the components' operation



NSS volumes monitor has the following feature: if a threat is detected in a file upon its copying (to a protected volume or within an NSS volume), SpIDer Guard for NSS marks only the copy of the infected file. The threat in the original file will not be detected. This file will be considered safe until an attempt to access this (original) file is performed or until it is modified if the file is located on an NSS volume.

If *Quarantine* action is specified for some threat type in NSS volumes monitor settings, the object containing a threat of this type will be placed to quarantine again on attempt to restore this object from quarantine to an NSS volume. For example, the following default [settings](#):

```
NSS.OnKnownVirus = Cure  
NSS.OnIncurable = Quarantine
```

move all incurable objects to quarantine. At that, when any incurable object is restored from quarantine to an NSS volume, this object is automatically returned to quarantine.

Specifying Paths to Scanning Objects

The monitor of NSS volumes SpIDer Guard for NSS checks only those objects that are located in protected volumes (the parameters **NssVolumesMountDir** and **ProtectedVolumes**) and the specified paths to which either do not coincide with those specified in the **ExcludedPath** parameter or matches the paths specified in the parameter **IncludedPath**. This parameter has priority over the parameter **ExcludedPath**—if the path to an object is specified in the both parameters, the object is scanned. It can be useful when, for example, files in some directory are frequently modified, which results in constant repeated scanning of these files and, thus, can increase system load. If it is known with certainty that frequent modification of files in a directory is not caused by a virus but is due to operation of a trusted program, you can add the path to this directory or these files to the list of exclusions. In this case, the NSS volume monitor SpIDer Guard for NSS stops responding to modification of these objects. The **IncludedPath** parameter is better be used if you want to allow scanning of some objects that are located inside the path specified in the **ExcludedPath** parameter.

Consider an example of the monitor configuration:

```
NssVolumesMountDir = /media/nss  
ProtectedVolumes =  
ExcludedPath = vol1/path1, vol1/path2, vol2/sys  
IncludedPath = vol1/path1, vol1/path2/incl, vol2/doc
```

According to the specified parameters, the monitor checks all files in the volume `vol3` (no limits on scanning), all files in the volume `vol2` (except files in the `/sys` directory and in all its subdirectories). In the volume `vol1`, files in the `/path2` directory are not checked; however, files in other directories of this volume, together with the content of the `/path2/incl` directory, are checked. Note that there is no sense to specify the same path (in this case, `vol1/path`) in the both lists because it means that there is no limits on the path scanning. Specifying the path



`vol2/doc` in the **IncludedPath** parameter is also useless because this directory is not located in the exclusion scope set for the volume `vol2` and contains only the `/sys` directory.

Command-Line Arguments

To run SpIDer Guard for NSS, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-nss [<parameters>]
```

SpIDer Guard for NSS can process the following options:

Parameter	Description
<code>--help</code>	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: <code>-h</code> Arguments: None.
<code>--version</code>	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: <code>-v</code> Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-nss --help
```

This command outputs short help information on SpIDer Guard for NSS.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon at the startup of the operating system. To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-nss**

Configuration Parameters

The component uses configuration parameters which are specified in the `[NSS]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.



The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
LogProtocol <i>{Boolean}</i>	<p>Indicates whether protocol messages are registered in the log file of NSS volume monitor SpIDer Guard for NSS.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Yes—messages are registered.• No—messages are not registered. <p>Default value: No</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-nss</p> <ul style="list-style-type: none">• For Linux: /opt/drweb.com/bin/drweb-nss
Start <i>{Boolean}</i>	<p>The component must be launched by the Dr.Web ConfigD configuration daemon.</p> <p><i>When you specify the Yes value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the No value, it instructs the configuration daemon to terminate the component immediately.</i></p> <p>Default value: No</p>
NssVolumesMountDir <i>{path to directory}</i>	<p>Path to the file system directory where NSS file system volumes are mounted.</p> <p>Default value: /media/nss</p>
ProtectedVolumes <i>{volume name}</i>	<p>Names of NSS file system volumes mounted on NssVolumesMountDir and protected by the suite. If no value is specified, all volumes in the volume mounting point of NssVolumesMountDir must be protected.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list of volumes vol1 and vol2.</p>



	<ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[<%NSSPIDER_SECTION%>] ProtectedVolumes = "vol1", "vol2"</pre>Two strings (one value per a string)<pre>[<%NSSPIDER_SECTION%>] ProtectedVolumes = vol1 ProtectedVolumes = vol2</pre>Adding values via the command drweb-ctl cfset.<pre># drweb-ctl cfset <%NSSPIDER_SECTION%>.ProtectedVolumes -a vol1 # drweb-ctl cfset <%NSSPIDER_SECTION%>.ProtectedVolumes -a vol2</pre> <p>Default value: <i>(not set)</i></p>
<p>ExcludedPath</p> <p><i>{path to file or directory}</i></p>	<p>Path to the object which must be skipped during scanning. You can specify a directory or file path. If a directory is specified, all directory content including subdirectories will be skipped. The exception is objects paths to which are specified in the parameter IncludedPath—such objects <i>will be scanned</i>.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the files /etc/file1 and directory /usr/bin.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[<%NSSPIDER_SECTION%>] ExcludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[<%NSSPIDER_SECTION%>] ExcludedPath = /etc/file1 ExcludedPath = /usr/bin</pre>Adding values via the command drweb-ctl cfset.<pre># drweb-ctl cfset <%NSSPIDER_SECTION%>.ExcludedPath -a /etc/file1 # drweb-ctl cfset <%NSSPIDER_SECTION%>.ExcludedPath -a /usr/bin</pre>



	<p><i>Paths in the list must be relative to a path indicated in NssVolumesMountDir.</i></p> <p>Default value: <i>(not set)</i></p>
IncludedPath <i>{path to file or directory}</i>	<p>Path to the object which must be scanned. You can specify a directory or file path. If a directory is specified, all directory content will be scanned.</p> <p>This parameter can be used only if you want to allow scanning of separate objects (files and subdirectories) paths to which is specified in the parameter ExcludedPath. In addition, this parameter has priority over the parameter ExcludedPath: if a path to an object is specified in the both parameters, this object will be scanned.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the files <code>/etc/file1</code> and directory <code>/usr/bin</code>.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[<%NSSPIDER_SECTION%>] IncludedPath = "/etc/file1", "/usr/bin"</pre>Two strings (one value per a string)<pre>[<%NSSPIDER_SECTION%>] IncludedPath = /etc/file1 IncludedPath = /usr/bin</pre>Adding values via the command <code>drweb-ctl cfset</code>.<pre># drweb-ctl cfset <%NSSPIDER_SECTION%>.IncludedPath -a /etc/file1 # drweb-ctl cfset <%NSSPIDER_SECTION%>.IncludedPath -a /usr/bin</pre> <p><i>Paths in the list must be relative to a path indicated in NssVolumesMountDir.</i></p> <p>Default value: <i>(not set)</i></p>
OnKnownVirus <i>{action}</i>	<p>Action applied by Dr.Web for UNIX File Servers to a known threat (virus, etc.) detected with signature analysis during the scanning initiated by NSS volume monitor.</p> <p>Possible values: <i>Cure, Quarantine, Delete</i></p> <p>Default value: <i>Cure</i></p>



OnIncurable <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to an incurable threat (that is, an attempt to apply <i>Cure</i> failed) detected during the scanning initiated by NSS volume monitor. Possible values: <i>Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnSuspicious <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to an unknown threat (or suspicious objects) detected by using heuristic analysis during the scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Quarantine</i>
OnAdware <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to adware detected during the scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnDialers <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to a dialer detected during the scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnJokes <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to joke programs detected during the scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnRiskware <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to riskware detected during the scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnHacktools <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to a hacktool (tool for remote administration, Trojan, etc.) detected during scanning initiated by NSS volume monitor. Possible values: <i>Report, Quarantine, Delete</i> Default value: <i>Report</i>
OnError <i>{action}</i>	Action applied by Dr.Web for UNIX File Servers to files that caused an error during the scanning initiated by NSS volume monitor.



	<p>Possible values: <i>Report, Quarantine, Delete</i></p> <p>Default value: <i>Report</i></p>
ScanTimeout <i>{time interval}</i>	<p>Timeout for scanning one file initiated by NSS volume monitor.</p> <p><i>A value in the range from 1s to 1h can be specified</i></p> <p>Default value: <i>30s</i></p>
HeuristicAnalysis <i>{On Off}</i>	<p>Indicates whether heuristic analysis is used for detection of unknown threats during the scanning initiated by NSS volume monitor. Heuristic analysis provides higher detection reliability but, at the same time, it increases time of virus scanning.</p> <p><i>Action applied to threats detected by heuristic analyzer is specified as the OnSuspicious parameter value.</i></p> <p>Allowed values:</p> <ul style="list-style-type: none">• <i>On</i>—instructs to use heuristic analysis when scanning.• <i>Off</i>—instructs not to use heuristic analysis. <p>Default value: <i>On</i></p>
PackerMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning packed objects. All objects at a deeper nesting level are skipped during the scanning initiated by NSS volume monitor.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: <i>8</i></p>
ArchiveMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning archives. All objects at a deeper nesting level are skipped during the scanning initiated by NSS volume monitor.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: <i>0</i></p>
MailMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning email messages and mailboxes. All objects at a deeper nesting level are skipped during the scanning initiated by NSS volume monitor.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: <i>8</i></p>
ContainerMaxLevel <i>{integer}</i>	<p>Maximum nesting level when scanning containers of other types. All objects at a deeper nesting level are skipped during the scanning initiated by NSS volume monitor.</p>



	<p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p>Default value: 8</p>
MaxCompressionRatio <i>{integer}</i>	<p>Maximum compression ratio of scanned objects (ratio between the compressed size and uncompressed size). If the ratio of an object exceeds the limit, this object is skipped during the scanning initiated by NSS volume monitor.</p> <p><i>The compression ratio must not be smaller than 2.</i></p> <p>Default value: 500</p>



If *Quarantine* action is specified for some threat type in NSS volumes monitor settings, the object containing a threat of this type will be placed to quarantine again on attempt to restore this object from quarantine to an NSS volume. For example, the following default settings:

```
NSS.OnKnownVirus = Cure
NSS.OnIncurable = Quarantine
```

move all incurable objects to quarantine. At that, when any incurable object is restored from quarantine to an NSS volume, this object is automatically returned to quarantine.

Integration with NSS Volumes

For the instructions for integration with NSS volumes, refer to the [Integration with NSS Volumes](#) section.



Dr.Web ClamD

The Dr.Web ClamD component performs emulation using the Dr.Web for UNIX File Servers interface of the **clamd** anti-virus daemon, which is a core component of the anti-virus product **Clam AntiVirus (ClamAV®)** from Sourcefire, Inc. This interface allows external applications that are able to interact with **ClamAV®** to use Dr.Web for UNIX File Servers for anti-virus scanning.

Operating Principles

The component is designed to check both the content of files in the local file system and the streams of data transmitted by an external application via a socket. Such checks are performed by the component at the request of an external application. Moreover, the component can check the content of those files for which an external application passed an open file descriptor via a socket.



File checks based on a passed file descriptor can be performed only if the descriptor was passed via a local UNIX socket.

If an external application has provided a path to a file in the local file system, the component sends the scanning task to the [Dr.Web File Checker](#) file checker component; otherwise, the component transmits data, received via the socket, to the [Dr.Web Network Checker](#) distributed scanning agent, as shown in the figure below.

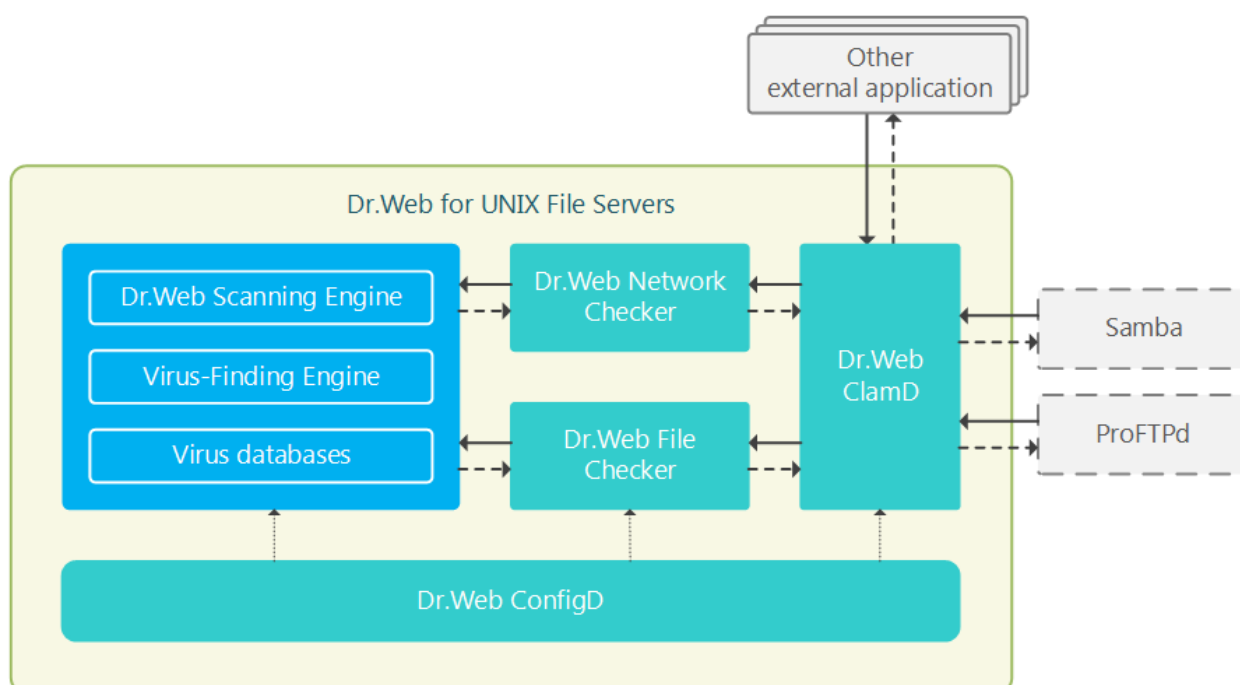


Figure 15. Diagram of the components' operation

By default, the component is not automatically launched upon the startup of Dr.Web for UNIX File Servers. To enable starting of the component, it is necessary [to set](#) the Yes value for the



The Figure above shows that external applications could be represented as file service servers (such as **Samba** and **ProFTPD**), if they are equipped with the integration module with **clamd**. For details, see section [Integration with External Applications](#).



Detected threats *cannot* be neutralized by Dr.Web for UNIX File Servers; the external application receives only the results of the scanning. Thus, any detected threats should be neutralized by the external application.

Command-Line Arguments

To run Dr.Web ClamD, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-clamd [<parameters>]
```

Dr.Web ClamD can process the following parameters:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-clamd --help
```

This command outputs short help information on Dr.Web ClamD.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when needed (as a rule, at the startup of the operating system). To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-clamd**

Configuration Parameters

The component uses configuration parameters which are specified in the [ClamD] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	Logging level of the component. If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used. Default value: Notice
Log <i>{log type}</i>	Logging method
ExePath <i>{path to file}</i>	Path to the executable file of the component. Default value: <opt_dir>/bin/drweb-clamd <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-clamd• For FreeBSD : /usr/local/libexec/drweb.com/bin/drweb-clamd
Start <i>{Boolean}</i>	The component must be launched by the Dr.Web ConfigD configuration daemon. When you specify the Yes value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the No value, it instructs the configuration daemon to terminate the component immediately. Default value: No
Endpoint.<tag>.ClamdSocket <i>{IP address UNIX socket}</i>	Defines a new connection point naming it <tag> and allocates a socket (IPv4 address or address of a UNIX socket) for clients that need to check files for threats. <i>Only one socket can be specified for one <tag> point.</i>



	Default value: <i>(not specified)</i>
<code>[Endpoint.<tag>.]DetectSuspicious</code> <code>{Boolean}</code>	<p>Inform about suspicious files detected by the heuristic analyzer.</p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: Yes</p>
<code>[Endpoint.<tag>.]DetectAdware</code> <code>{Boolean}</code>	<p>Inform about files containing adware.</p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: Yes</p>
<code>[Endpoint.<tag>.]DetectDialers</code> <code>{Boolean}</code>	<p>Inform about files containing dialers.</p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: Yes</p>
<code>[Endpoint.<tag>.]DetectJokes</code> <code>{Boolean}</code>	<p>Inform about files containing jokes.</p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: No</p>
<code>[Endpoint.<tag>.]DetectRiskware</code> <code>{Boolean}</code>	<p>Inform about files containing riskware.</p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: No</p>
<code>[Endpoint.<tag>.]DetectHacktools</code> <code>{Boolean}</code>	<p>Inform about files containing hacktools.</p>



	<p><i>If the <code>Endpoint.<tag></code> prefix is specified, it means that the parameter's value is set only for the <code><tag></code> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: No</p>
<p><code>[Endpoint.<tag>.]ReadTimeout</code> <i>{time interval}</i></p>	<p>Sets the maximum time to wait for data from a client.</p> <p><i>If the <code>Endpoint.<tag></code> prefix is specified, it means that the parameter's value is set only for the <code><tag></code> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 5s</p>
<p><code>[Endpoint.<tag>.]StreamMaxLength</code> <i>{size}</i></p>	<p>Sets the maximum size of data that can be received from a client (for transmitting data to scan as a stream of bytes).</p> <p><i>If the <code>Endpoint.<tag></code> prefix is specified, it means that the parameter's value is set only for the <code><tag></code> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 25mb</p>
<p><code>[Endpoint.<tag>.]ScanTimeout</code> <i>{time interval}</i></p>	<p>Sets the maximum time to scan one file (or one portion of data) received from a client.</p> <p><i>A value in the range from 1s to 1h can be specified</i></p> <p><i>If the <code>Endpoint.<tag></code> prefix is specified, it means that the parameter's value is set only for the <code><tag></code> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 3m</p>
<p><code>[Endpoint.<tag>.]HeuristicAnalysis</code> <i>{On Off}</i></p>	<p>Indicates whether heuristic analysis is used for scanning.</p> <p><i>If the <code>Endpoint.<tag></code> prefix is specified, it means that the parameter's value is set only for the <code><tag></code> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: On</p>



<code>[Endpoint.<tag>.] PackerMaxLevel</code> <code>{integer}</code>	<p>Sets the maximum nesting level of packed objects that can be scanned.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 8</p>
<code>[Endpoint.<tag>.] ArchiveMaxLevel</code> <code>{integer}</code>	<p>Sets the maximum nesting level of archives that can be scanned.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 8</p>
<code>[Endpoint.<tag>.] MailMaxLevel</code> <code>{integer}</code>	<p>Sets the maximum nesting level of mail files that can be scanned.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 8</p>
<code>[Endpoint.<tag>.] ContainerMaxLevel</code> <code>{integer}</code>	<p>Sets the maximum nesting level of objects in containers that can be scanned.</p> <p><i>A value in the range from 0 to 60 can be specified. If the value is set to 0, nested objects are not scanned.</i></p> <p><i>If the Endpoint.<tag> prefix is specified, it means that the parameter's value is set only for the <tag> connection point; otherwise, it is set for all points which do not have another value of this parameter specified for them.</i></p> <p>Default value: 8</p>



<code>[Endpoint.<tag>.]MaxCompressionRatio</code> <code>{integer}</code>	<p>Sets the maximum allowed compression ratio of compressed/packed objects (ratio between the uncompressed size and the compressed size). If the ratio of an object exceeds the limit, this object will be skipped during the scanning.</p> <p><i>The compression ratio must not be smaller than 2.</i></p> <p>Default value: 500</p>
---	--

Special Aspects of Component Configuration

Parameters marked with an optional `Endpoint.<tag>` prefix can be grouped. Each group defines a unique connection *point* (*endpoint*) that can be used by clients to connect to the component and has a unique `<tag>` identifier assigned to it. All the scanning parameters belonging to the same group define the settings that are applicable only when data is scanned for the clients connected to the corresponding connection point. If a parameter is specified without an `Endpoint.<tag>` prefix, this sets the value for all connection points. If you delete some parameter from some connection point, then instead of reverting to the program's hard-coded default value for this parameter, the program will use the current value of the corresponding "parent" parameter of the same name (set without the `Endpoint.<tag>` prefix).



The **ClamdSocket** parameter must always be specified with an `Endpoint.<tag>` prefix, as it defines both a listening socket and a group (connection point) to which this socket corresponds.

Example:

Let us assume that we need to set up two connection points for two groups of external applications (servers) — let the groups be called *servers1* and *servers2*. And the servers from the *servers1* group can connect through a UNIX socket, whereas the servers from the *servers2* group can connect via a network connection. Moreover, let us assume that heuristic analysis must be disabled by default, but must be used for servers from the *servers2* group. The following example shows how to configure this:

- 1) In the [configuration file](#):

```
[ClamD]
HeuristicAnalysis = Off

[ClamD.Endpoint.servers1]
ClamSocket = /tmp/srv1.socket

[ClamD.Endpoint.servers2]
ClamSocket = 127.0.0.1:1234
HeuristicAnalysis = On
```

- 2) For command-line-based management tool [Dr.Web Ctl](#):



```
# drweb-ctl cfset ClamD.HeuristicAnalysis Off
# drweb-ctl cfset ClamD.Endpoint -a servers1
# drweb-ctl cfset ClamD.Endpoint -a servers2
# drweb-ctl cfset ClamD.Endpoint.servers1.ClamdSocket /tmp/srv1.socket
# drweb-ctl cfset ClamD.Endpoint.servers2.ClamdSocket 127.0.0.1:1234
# drweb-ctl cfset ClamD.Endpoint.servers2.HeuristicAnalysis On
```



Both ways have an equal effect but if you edit the configuration file, you will also need to apply the changed settings by sending a `SIGHUP` signal to the **drweb-configd** component (to do that, you can issue the `drweb-ctl reload command`).

Integration with External Applications

The interface—that emulates the one of the **clamd** anti-virus daemon (included in **ClamAV**)—allows Dr.Web ClamD to communicate with any external application that is capable of connecting to this anti-virus daemon.

The table below shows examples of applications that can use **clamd** for anti-virus scans:

Product	Integration
File services	
FTP server ProFTPd	<p>Use of clamd:</p> <p>Scanning of files uploaded to the server via FTP.</p> <p>Integration requirements:</p> <p>Build ProFTPd with an additional module—<code>mod_clamav</code>.</p> <p>Links to documentation:</p> <p>ProFTPd documentation: http://www.proftpd.org/docs/</p> <p>Description and source code files of <code>mod_clamav</code>: https://github.com/jbenden/mod_clamav</p>

In the settings of the external software component that communicates directly with Dr.Web ClamD as with the **clamd** anti-virus daemon, specify an address for connecting to **clamd** as a path to a UNIX socket or as a TCP socket listened to by Dr.Web ClamD at one of its connection points (*endpoint*) set up in its configuration.

Example of how to connect **ProFTPd** to Dr.Web ClamD:

1. Configuring Dr.Web ClamD:

```
[ClamD]
Start = yes

[ClamD.Endpoint.ftps]
ClamdSocket = 127.0.0.1:3310
```



2. Configuring **ProFTPD**:

```
ClamAV on  
ClamServer 127.0.0.1  
ClamPort 3310
```




Dr.Web File Checker

The file checking component—Dr.Web File Checker—is designed for checking files and directories in the file system. It is used by other components of Dr.Web for UNIX File Servers to check file system objects. Moreover, this component also functions as a quarantine manager, as it manages the contents of the directories where isolated (quarantined) files are kept.

Operating Principles

This component is used to access any file system objects (files, directories, boot records). It is started with superuser (*root*) privileges.

It indexes all checked files and directories and saves all the data about the objects that have been checked to a special cache to avoid repeated checking of objects that have been already checked and have not been modified since that (in this case, if a request to check such an object is received, the previous check result, retrieved from cache, is returned). A diagram showing how the component works is given in the figure below.

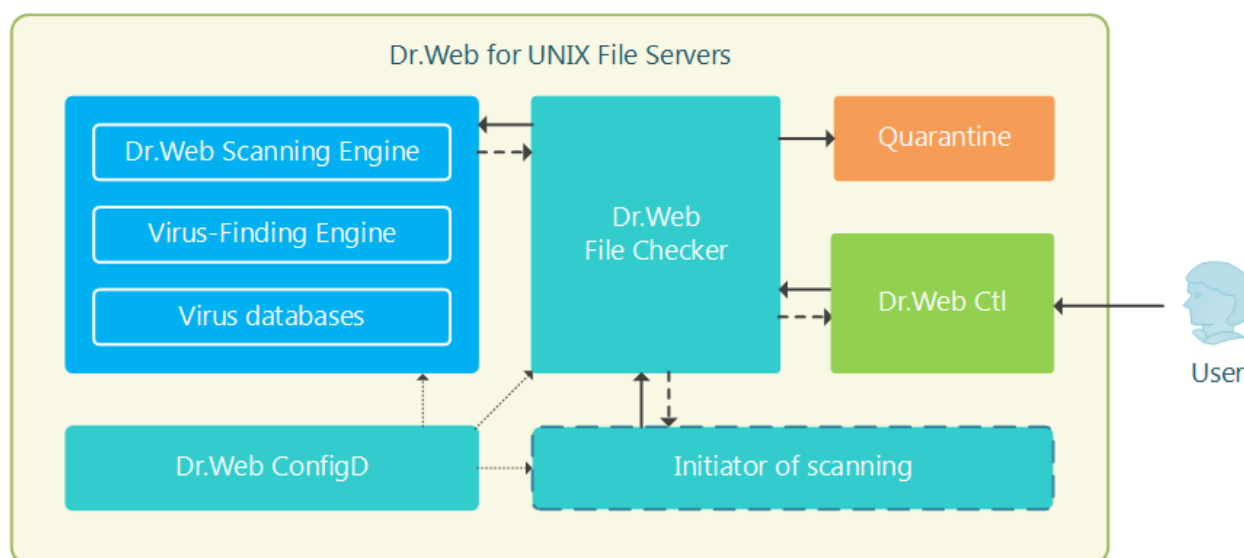


Figure 16. Diagram of the components' operation

When a request to check a file system object is received from Dr.Web for UNIX File Servers's components, it checks whether this object requires scanning. If so, a scanning task is generated for [Dr.Web Scanning Engine](#). If the scanned object contains a threat, Dr.Web File Checker neutralizes it (deletes or quarantines) if this action has been specified by the client component that initiated the scanning. Scanning can be initiated by various components of the product (for example, by the [SpIDer Guard for SMB](#) monitor).

During the scanning, the file-checking component generates and sends to the client component a report detailing the results of the scanning and the applied actions, if any.



Apart from the standard scanning method, the following special methods are available for internal use:

- *The “flow” scanning method.* A client component that uses this scanning method initializes detection and neutralization parameters only once. These parameters will be applied to all future requests to check a file coming from this client component. This method is used by the SpIDer Guard for SMB [monitor](#).
- *The “proxy” scanning method.* When this method is used, the file-checking component scans files without applying any actions to detected threats and without keeping any records about the detected threats to permit future action. Any necessary actions must be applied by the component that initiated the scanning process. This method is used by the [SpIDer Guard for SMB](#) monitor and by the [Dr.Web ClamD](#) component.

Files can be scanned with the “flow” and “proxy” scanning methods using the `flowscan` and `proxyscan` [commands of the Command-Line Call Format](#) utility (launched by the **drweb-ctl** command). However, for a normal on-demand scanning, it is recommended that you use the `scan` command.

The component collects statistics on scanned files averaging the number of files scanned per second in the last minute, 5 minutes, 15 minutes.

Command-Line Arguments

To launch Dr.Web File Checker, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-filecheck [<parameters>]
```

Dr.Web File Checker can process the following parameters:

Parameter	Description
<code>--help</code>	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: <code>-h</code> Arguments: None.
<code>--version</code>	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: <code>-v</code> Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-filecheck --help
```

This command outputs short help information on Dr.Web File Checker.



Startup Notes

The component cannot be launched directly from line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when receiving requests on file system scanning from other components of Dr.Web for UNIX File Servers. To manage the operation of the component, as well as to scan files when needed, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is launched by using the `drweb-ctl` [command](#)).

To scan an arbitrary file or directory using Dr.Web File Checker you can use `scan` command of Dr.Web Ctl:

```
$ drweb-ctl scan <path to file or directory>
```



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-filecheck**

Configuration Parameters

The component uses configuration parameters which are specified in the `[FileCheck]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

This section stores the following parameters:

LogLevel <i>{logging level}</i>	Logging level of the component. If the parameter value is not specified, the DefaultLogLevel parameter value from the <code>[Root]</code> section is used. Default value: Notice
Log <i>{log type}</i>	Logging method
ExePath <i>{path to file}</i>	Path to the executable file of the component. Default value: <code><opt_dir>/bin/drweb-filecheck</code> <ul style="list-style-type: none">• For Linux, Solaris: <code>/opt/drweb.com/bin/drweb-filecheck</code>• For FreeBSD: <code>/usr/local/libexec/drweb.com/bin/drweb-filecheck</code>
DebugClientIpc <i>{Boolean}</i>	Indicates whether detailed IPC messages should be included into the log file on the debug level (i.e. when LogLevel = <code>DEBUG</code>). Default value: No



DebugScan <i>{Boolean}</i>	Indicates whether detailed messages received during file scanning should be included into the log file on the debug level (i.e. when LogLevel = DEBUG). Default value: No
DebugFlowScan <i>{Boolean}</i>	Indicates whether detailed messages about file scanning by the “flow” method should be included into the log file on the debug level (i.e. when LogLevel = DEBUG). Normally this scanning method is used by the SpIDer Guard monitor. Default value: No
DebugProxyScan <i>{Boolean}</i>	Indicates whether detailed messages about file scanning by the “proxy” method should be included into the log file on the debug level (i.e. when LogLevel = DEBUG). Normally this scanning method is used by the SpIDer Guard for SMB monitor and by the Dr.Web ClamD component. Default value: No
DebugCache <i>{Boolean}</i>	Indicates whether detailed messages about the cached results of scanning should be included into the log file on the debug level (i.e. when LogLevel = DEBUG). Default value: No
MaxCacheSize <i>{size}</i>	Maximum allowed size of cache to store data about scanned files. <i>If 0 is specified, caching is disabled.</i> Default value: 50mb
RescanInterval <i>{time interval}</i>	Period of time during which a file will not be rescanned if the results of its previous scan are available in the cache (the period during which the stored information is considered up-to-date). <i>The parameter can have a value from 0s to 1m (inclusive). If the set interval is less than 1s—there will be no delay, the file will be scanned upon any request.</i> Default value: 1s
IdleTimeLimit <i>{time interval}</i>	Maximum time that the component can remain idle. If the specified value is exceeded, the component shuts down. <i>The parameter can have a value from 10s to 30d (inclusive).</i> Default value: 30s



Dr.Web Network Checker

Network checker agent Dr.Web Network Checker is designed for scanning the data received through the network in the checking engine, as well as distributed file scanning for threats. The component allows to arrange a connection between network hosts with installed Dr.Web for UNIX File Servers for receiving and transmitting data (for example, file content) via the network hosts to perform its scanning. The component organizes automatic distribution of scanning tasks (by transmitting and receiving them over the network) to all available network hosts to which it is configured. The component balances the load between the hosts caused by scanning tasks. If there are no configured connections with remote hosts, the component transmits all the data to the local Dr.Web Scanning Engine only.

Note that the component is always used to scan the data received via network connections. Thus, if the component is missing or unavailable, the performance of the components that transmit data for scanning via the network connection will be incorrect (SpIDer Gate, Dr.Web ClamD).



In case of high intensity of scanning of data transferred via the network, there is a possibility of having problems with scanning due to depletion of the number of available file descriptors. In this case, it is necessary to [increase the limit](#) of the number of file descriptors available to Dr.Web for UNIX File Servers.

During scanning, data can be shared either over an open channel or over a protected one, applying SSL/TLS. To use a secure HTTPS connection it is required to provide an appropriate SSL server certificate and private key for hosts that share files. If you need to generate SSL keys and certificates, you can use the **openssl** utility. An example of how to use the **openssl** utility to generate a certificate and a private key is given in the section [Appendix E. Generating SSL certificates](#).

Operating Principles

The Dr.Web Network Checker component allows to arrange connection between Dr.Web for UNIX File Servers and a set of other nodes which have Dr.Web for UNIX File Servers (or other Dr.Web for UNIX solution version 10.1 or above) installed on them. This will organize a distributed data scanning for threats (for example, file content). With the component, you can create and configure a “*scanning cluster*”, specifying the set of connections between cluster nodes (an instance of the distributed scanning agent Dr.Web Network Checkers should be launched at each node).

On each node within the cluster, Dr.Web Network Checker agent constitutes the automatic distribution of scanning jobs by transmitting data for scanning to all available nodes. At that, the agent sets up the load balancing on nodes, caused by file scanning, depending on resources available on remote nodes (the number of child scanning processes of Dr.Web Scanning Engine on each node acts as an indicator for the number of nodes available). The agent also considers the queue of files waiting for scanning on each host. Data received for scanning over the network is transmitted to the [Dr.Web Scanning Engine](#) scanning engine, as shown on the figure below.

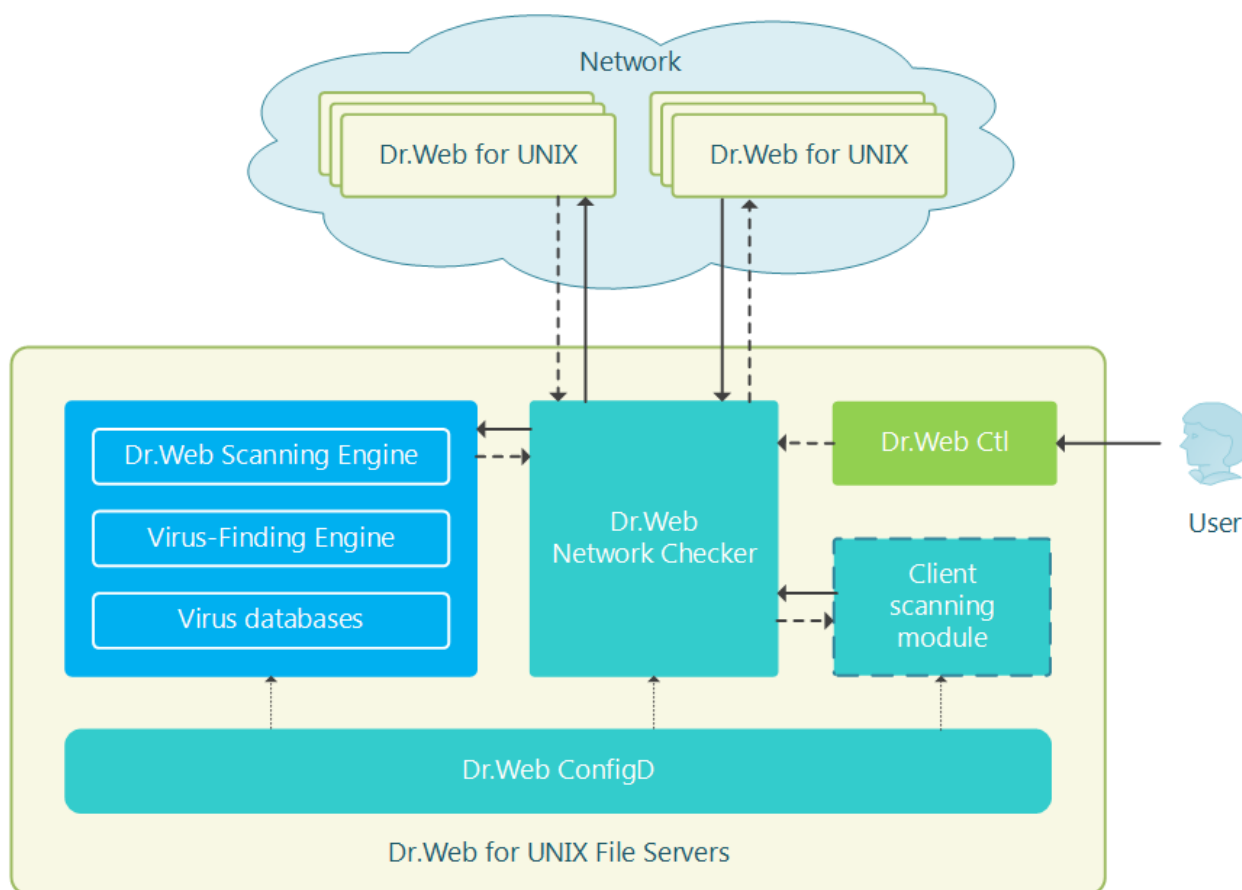


Figure 17. Diagram of the components' operation

In this case, any network node included in the scanning cluster can act as a scanning client that transmits data to a remote scan as well as a scanning server that receives data from the specified network nodes for verification. If necessary, the distributed scanning agent can be configured so that the node acts only as a scanning server or only as a scanning client.

On a local host, sending data for scanning via Dr.Web Network Checker can be started both at user's command specified via the [Dr.Web Ctl](#) command-line management tool and at requests received from some product components, for example, the [Dr.Web ClamD](#) component, which provides the interface of the **clamd** daemon included in **ClamAV®**. That is why the scheme contains an abstract "Client scanning module".

Note that components marked as "Client scanning module" always use the Dr.Web Network Checker for transmitting files to be scanned by Dr.Web Scanning Engine, even if Dr.Web Scanning Engine is located on the local host. Thus, if Dr.Web Network Checker is unavailable, these components will not work correctly.



It is possible to create your own component (external application) which will use Dr.Web Network Checker to check the files (including distributing the scanning jobs to the nodes of the scanning cluster). For this, the Dr.Web Network Checker component provides a custom API based on the **Google Protobuf** technology. The Dr.Web Network Checker API, as well as client application sample code that uses Dr.Web Network Checker, are supplied as part of `drweb-netcheck` package.



Command-Line Arguments

To run Dr.Web Network Checker, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-netcheck [<parameters>]
```

Dr.Web Network Checker can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-netcheck --help
```

This command outputs short help information on Dr.Web Network Checker.

Startup Notes

The component cannot be run directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is run automatically by the configuration daemon [Dr.Web ConfigD](#) configuration daemon when required (usually on operating system startup). If a value of the **FixedSocketPath** parameter in the [configuration](#) is specified, the agent is always running and available for clients via the specified UNIX socket. To start scanning via network, you can use the [Dr.Web Ctl](#) command-line tool for Dr.Web for UNIX File Servers management (it is started by the [command drweb-ctl](#)). If there are no configured connections to remote hosts, the local scanning will be started.

To scan an arbitrary file or directory using Dr.Web Network Checker you can use `netscan` command of Dr.Web Ctl tool:

```
$ drweb-ctl netscan <path to file or directory>
```



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-netcheck**



Configuration Parameters

The component uses configuration parameters which are specified in the [NetCheck] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel {logging level}	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log {log type}	<p>Logging method</p>
ExePath {path to file}	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/share/drweb-netcheck/linkchecker</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-netcheck• For FreeBSD : /usr/local/libexec/drweb.com/bin/drweb-netcheck
FixedSocketPath {path to file}	<p>Path to the UNIX socket of the fixed Dr.Web Network Checker agent instance.</p> <p>If this parameter is specified, the Dr.Web ConfigD configuration daemon checks that there is always a running component copy of the distributed scanning agent that is available to the clients via this socket.</p> <p>Default value: (not specified)</p>
RunAsUser {UID user name}	<p>The parameter determines under which user name the component should be run. The user name can be specified either as the user's number UID or as the user's login. If the user name consists of numbers (i.e. similar to number UID), it is specified with the "name:" prefix, for example: RunAsUser = name:123456.</p> <p><i>When a user name is not specified, the component operation terminates with an error after the startup.</i></p> <p>Default value: drweb</p>
IdleTimeLimit {time interval}	<p>Maximum time that the component can remain idle. If the specified value is exceeded, the component shuts down.</p>



	<p>Minimum value—10s.</p> <p><i>If the LoadBalanceAllowFrom or FixedSocketPath parameter is set, this setting is ignored (the component does not finish its operation after the time interval expires).</i></p> <p>Default value: 30s</p>
LoadBalanceUseSsl <i>{Boolean}</i>	<p>The indicator which determines whether a secure SSL/TLS connection is used for connection to other hosts.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Yes—instructs to use SSL/TLS• No—instructs not to use SSL/TLS <p><i>If the parameter is set to Yes, a certificate and the corresponding private key should be specified for this host and for hosts with which it interacts (the parameters LoadBalanceSslCertificate and LoadBalanceSslKey).</i></p> <p>Default value: No</p>
LoadBalanceSslCertificate <i>{path to file}</i>	<p>Path to the SSL certificate used by Dr.Web Network Checker for communication with other hosts via a secure SSL/TLS connection.</p> <p><i>Please note that the certificate file and the private key file (which is specified by a parameter described below) must form a matching pair.</i></p> <p>Default value: (not specified)</p>
LoadBalanceSslKey <i>{path to file}</i>	<p>Path to the private key used by Dr.Web Network Checker for communication with other hosts via a secure SSL/TLS connection.</p> <p><i>Please note that the certificate file and the private key file (which is specified by the mentioned parameter) must form a matching pair.</i></p> <p>Default value: (not specified)</p>
LoadBalanceSslCa <i>{path}</i>	<p>The path to the directory or file that contains the list of root certificates that are trusted. Among these certificates, there must be a certificate that certifies the authenticity of the certificates used by agents within the scanning cluster when exchanging data over SSL/TLS protocols.</p> <p><i>If the parameter value is empty, Dr.Web Network Checker working on this host does not authenticate certificates of interacting agents; however, depending on the settings, these agents can authenticate the certificate used by the agent operating on the host.</i></p> <p>Default value: (not specified)</p>
LoadBalanceServerSocket <i>{address}</i>	<p>Network socket (IP address and port) which is listened on this host by Dr.Web Network Checker for receiving files sent by</p>



	<p>remote hosts for scanning (if it can operate as a scanning server).</p> <p>Default value: <i>(not specified)</i></p>
LoadBalanceAllowFrom <i>{IP address}</i>	<p>IP address of a remote network host from which the Dr.Web Network Checker can receive files for scanning (as a scanning server).</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list of host addresses 192.168.0.1 and 10.20.30.45.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string <pre>Section [NetCheck] LoadBalanceAllowFrom = "192.168.0.1", "10.20.30.45"</pre> <ul style="list-style-type: none">Two strings (one value per a string) <pre>[NetCheck] LoadBalanceAllowFrom = 192.168.0.1 LoadBalanceAllowFrom = 10.20.30.45</pre> <ol style="list-style-type: none">Adding values via the command <code>drweb-ctl cfset</code>. <pre># drweb-ctl cfset NetCheck.LoadBalanceAllowFrom -a 192.168.0.1 # drweb-ctl cfset NetCheck.LoadBalanceAllowFrom -a 10.20.30.45</pre> <p><i>If the parameter is empty, removed files cannot be received for scanning (the host does not operate as a scanning server).</i></p> <p>Default value: <i>(not specified)</i></p>
LoadBalanceSourceAddress <i>{IP address}</i>	<p>IP address of a network interface used by Dr.Web Network Checker on the host for transferring files for their remote scanning (if the host operates as a scanning server and has several network interfaces).</p> <p><i>If an empty value is specified, the network interface automatically selected by the OS kernel is used.</i></p> <p>Default value: <i>(not specified)</i></p>
LoadBalanceTo	<p>Socket (IP address or port) of a remote host to which Dr.Web Network Checker on the host can send files for their remote</p>



<code>{address}</code>	<p>scanning (as a network scanning client).</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add sockets 192.168.0.1:1234 and 10.20.30.45:5678 to the list.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[NetCheck] LoadBalanceTo = "192.168.0.1:1234", "10.20.30.45:5678"</pre>Two strings (one value per a string)<pre>[NetCheck] LoadBalanceTo = 192.168.0.1:1234 LoadBalanceTo = 10.20.30.45:5678</pre>Adding values via the command drweb-ctl <code>cfset</code>.<pre># drweb-ctl cfset NetCheck.LoadBalanceTo -a 192.168.0.1:1234 # drweb-ctl cfset NetCheck.LoadBalanceTo -a 10.20.30.45:5678</pre> <p><i>If the parameter value is empty, local files cannot be transferred for a remote scanning (the host does not operate as a network scanning client).</i></p> <p>Default value: (not specified)</p>
LoadBalanceStatusInterval <code>{time interval}</code>	<p>Time interval considered by the host to send the next message containing information about its workload to all scanning clients (specified in the LoadBalanceAllowFrom parameter).</p> <p>Default value: 1s</p>
SpoolDir <code>{path to directory}</code>	<p>Local file system directory used to store files sent over the network for scanning and received by Dr.Web Network Checker.</p> <p>Default value: /tmp/netcheck</p>
LocalScanPreference <code>{fractional number}</code>	<p>Relative weight (priority) of this host which is considered when a scanning server is selected to scan a file (a local file or a file received over the network). If the relative weight of the local station is greater than the weights of all hosts available as scanning servers, files are scanned locally.</p> <p>Minimum value—1.</p>



	Default value: 1
--	------------------



Dr.Web Scanning Engine

Dr.Web Scanning Engine scanning engine is designed to search for viruses and other malicious objects in files and boot records (*MBR – Master Boot Record, VBR – Volume Boot Record*) of disk devices. The component loads the anti-virus engine Dr.Web Virus-Finding Engine into memory and starts it as well as loads Dr.Web virus databases used by the engine for threat detection.

The scanning engine operates in the daemon mode, as a service which receives scanning requests from other Dr.Web for UNIX File Servers components. *If the Dr.Web Scanning Engine and Dr.Web Virus-Finding Engine components are missing or unavailable, no anti-virus scanning is performed.*

Operating Principles

The component operates as a service which receives requests to scan file system objects (files and boot disk records) from Dr.Web for UNIX File Servers components. It also queues scanning tasks and scans requested objects by using Dr.Web Virus-Finding Engine. If a threat is detected and the scanning task instructs to cure threats, the scanning engine attempts to cure it if this action can be applied to the scanned object. The figure below shows the operation scheme of Dr.Web Scanning Engine scanning engine.

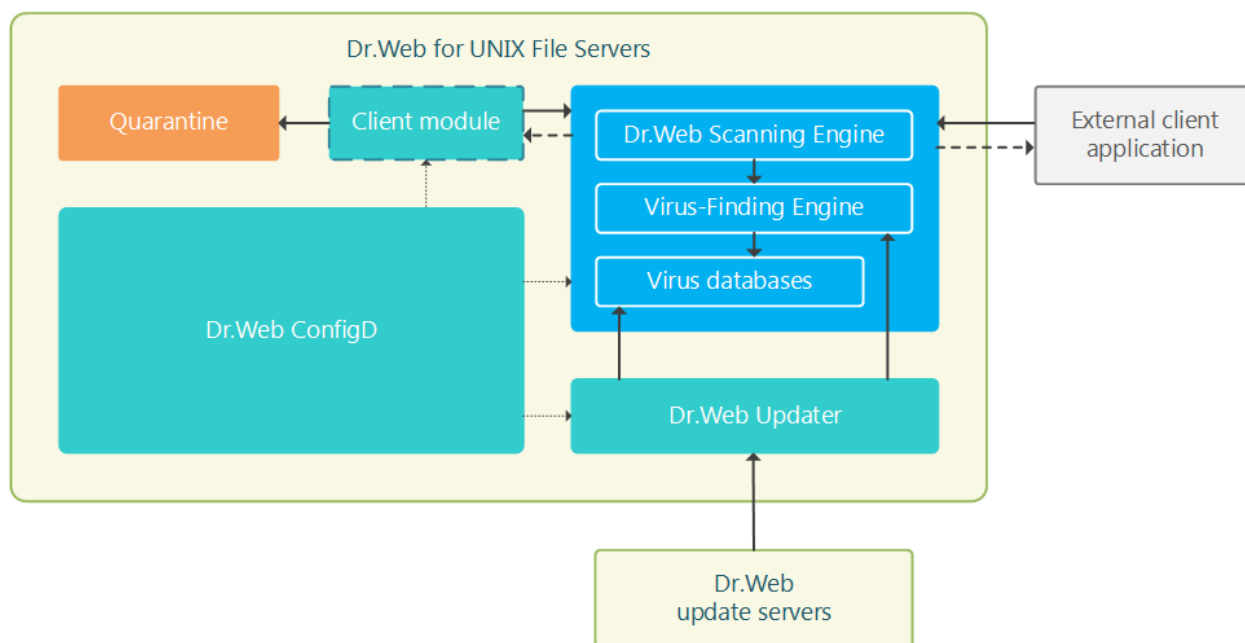


Figure 18. Diagram of the components' operation

The scanning engine, the anti-virus engine Dr.Web Virus-Finding Engine, and the virus databases form one unit and cannot be separated: the scanning engine downloads virus databases and provides the operation environment for the cross-platform anti-virus engine Dr.Web Virus-Finding Engine. The virus databases and the anti-virus engine are updated by the [Dr.Web Updater](#) update component that is included in the product, but this component is not a part of the scanning engine. The update component is run by the [Dr.Web ConfigD](#) configuration



daemon periodically or forcefully, if the corresponding command is sent by the user. Moreover, if Dr.Web for UNIX File Servers operates in central protection mode, updating of virus databases and anti-virus engine is performed by the [Dr.Web ES Agent](#) (it is not shown in the above-mentioned scheme). The latter component interacts with the central protection server and receives the updates.

The scanning engine can operate both under management of the configuration daemon Dr.Web ConfigD and in an autonomous mode. In the former case, the daemon runs the engine and ensures that virus databases are up to date. In the latter case, engine startup and updating of virus databases is performed by an external application that uses the engine. Dr.Web for UNIX File Servers's components that issue requests to the scanning engine asking it to scan files for them (indicated as "Client modules" in the diagram) use the same interface as other external applications would.



Users are provided with the opportunity to create own component (external application) using Dr.Web ASE for files checks. For this, Dr.Web Scanning Engine contains a special API, based on **Google Protobuf**. To obtain Dr.Web Scanning Engine API guide and examples of client application using Dr.Web Scanning Engine, contact Doctor Web partner care department (<https://partners.drweb.com/>).

Received tasks are automatically distributed into queues with different priorities: high, normal and low. Selection of the queue depends on the component that created a task: for example, tasks created by a file system monitor receive high priority as response time is important for monitoring. The scanning engine computes statistics of its operations, including the number of all tasks received for scanning and the queue length. As the average load rate, the scanning engine uses the average length of queues per second. This rate is averaged for the last minute, last 5 minutes and last 15 minutes.

Dr.Web Virus-Finding Engine supports signature analysis (signature-based threat detection) and other [methods](#) of heuristic and behavioral analysis designed for detection of potentially dangerous objects based on machine instructions and other attributes of executable code.



Heuristic analysis cannot guarantee highly reliable results and may commit the following errors:

- *Errors of the first type*. These errors occur when a safe object is detected as malicious (false positive detections).
- *Errors of the second type*. These errors occur when a malicious object is detected as safe.

Thus, objects detected by the heuristics analyzer are treated as *Suspicious*.

It is recommended that you choose to move suspicious objects to quarantine. After virus databases are updated, such files can be scanned using signature analysis. Keep the virus databases up to date in order to avoid errors of the second type.



Dr.Web Virus-Finding Engine allows to scan and cure both files and packed objects or objects in different containers (such as archives, email messages, etc.).

Command-Line Arguments

To run the scanning engine Dr.Web Scanning Engine from the command line, type the following command:

```
$ <opt_dir>/bin/drweb-se <socket> [<parameters>]
```

where the mandatory `<socket>` argument indicates the address of the socket used by Dr.Web Scanning Engine for processing requests of the client components. It can be set only as a file path (UNIX socket).

Dr.Web Scanning Engine can process the following options:

Parameter	Description
<code>--help</code>	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: <code>-h</code> Arguments: None.
<code>--version</code>	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: <code>-v</code> Arguments: None.
<i>Additional launch parameters (they are the same as configuration file parameters and substitute them when required):</i>	
<code>--EnginePath</code>	Function: Specify a path to the library of Dr.Web Virus-Finding Engine anti-virus engine. Short form: No. Arguments: <i><path to the file></i> —a full path to the library that you want to use.
<code>--VirusBaseDir</code>	Function: Specify a path to the directory with virus database files. Short form: No. Arguments: <i><path to the catalog></i> —path to the virus database directory.
<code>--TempDir</code>	Function: Specify a path to the directory with temporary files. Short form: No. Arguments: <i><path to the catalog></i> —full path to the directory with temporary files.
<code>--Key</code>	Function: Specify a path to the key file. Short form: No. Arguments: <i><path to the file></i> —a full path to the key file that you want to use.



<code>--MaxForks</code>	<p>Function: Determine the maximum allowed number of child processes which can be started by Dr.Web Scanning Engine during scanning.</p> <p>Short form: No.</p> <p>Arguments: <i><number></i>—the maximum allowed number of child processes.</p>
<code>--MaxForksPerFile</code>	<p>Function: Determine the maximum allowed number of child processes which can be used by Dr.Web Scanning Engine during scanning one compound file (an archive, a container, etc.).</p> <p>Short form: No.</p> <p>Arguments: <i><number></i>—the maximum allowed number of child processes. Cannot be less than 1 and greater than the <code>--MaxForks</code> parameter value.</p>
<code>--WatchdogInterval</code>	<p>Description: Determine frequency with which Dr.Web Scanning Engine checks whether child processes are operable and stops those processes that stopped responding.</p> <p>Short form: No.</p> <p>Arguments: <i><time interval></i>—frequency of checking child processes.</p>
<code>--Shelltrace</code>	<p>Function: turn on the shell tracing (log detailed information on file scanning performed by Dr.Web Virus-Finding Engine).</p> <p>Short form: No.</p> <p>Arguments: None.</p>
<code>--LogLevel</code>	<p>Description: Set the level of logging executed by Dr.Web Scanning Engine during the operation.</p> <p>Short form: No.</p> <p>Arguments: <i><logging level></i>. Allowed values:</p> <ul style="list-style-type: none">• <code>DEBUG</code>—the most detailed logging level. All messages and debug information are registered.• <code>INFO</code>—all messages are registered.• <code>NOTICE</code>—all error messages, warnings, and notifications are registered.• <code>WARNING</code>—all error messages and warnings are registered.• <code>ERROR</code>—only error messages are registered.
<code>--Log</code>	<p>Description: Specify the method for logging component messages.</p> <p>Short form: No.</p> <p>Arguments: <i><log type></i>. Allowed values:</p> <ul style="list-style-type: none">• <code>Stderr[:ShowTimestamp]</code>—messages are output to a standard error stream <code>stderr</code>. Additional option <code>ShowTimestamp</code> instructs to add a time stamp to every message.• <code>Syslog[:<facility>]</code>—messages are transmitted to the system logging service syslog. Additional option <i><facility></i> is used to specify a level at which syslog registers messages. The following values are possible:<ul style="list-style-type: none">◦ <code>DAEMON</code>—messages of daemons.◦ <code>USER</code>—messages of user processes.



- MAIL—messages of mail programs.
- LOCAL0—messages of local processes 0.
- ...
- LOCAL7—messages of local processes 7.
- *<path>*—path to the file where all messages are registered.

Examples:

```
--Log /var/opt/drweb.com/log/se.log
--Log Stderr:ShowTimestamp
--Log Syslog:DAEMON
```

Example:

```
$ /opt/drweb.com/bin/drweb-se /tmp/drweb.ipc/.se --MaxForks=5
```

This command starts an instance of Dr.Web Scanning Engine scanning engine, instructs it to create the `/tmp/drweb.ipc/.se` UNIX socket for an interaction with the client components and to start no more than 5 child scanning processes while scanning a file.

Startup Notes

When necessary, any number of scanning engine Dr.Web Scanning Engine instances can be started. The instances provide the scanning service for client applications (not only for Dr.Web for UNIX File Servers components). At that, if a value of the **FixedSocketPath** parameter is specified in the component's [configuration](#), one instance of the scanning engine is always running by the [Dr.Web ConfigD](#) configuration daemon and is always available for the clients via this UNIX socket. The instances of the scanning engine started directly from the command line, will operate in an autonomous mode without establishing connection to the configuration daemon, even if it is running. To manage the operation of the component, as well as to scan files when needed, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is launched by using the **drweb-ctl** [command](#)).

To scan an arbitrary file or directory using Dr.Web Scanning Engine you can use `rawscan` command of Dr.Web Ctl tool:

```
$ drweb-ctl rawscan <path to file or directory>
```



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-se**

Configuration Parameters

The component uses configuration parameters which are specified in the `[ScanEngine]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.



This section stores the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-se</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-se• For FreeBSD: /usr/local/libexec/drweb.com/bin/drweb-se
FixedSocketPath <i>{path to file}</i>	<p>Path to the UNIX socket of the fixed Dr.Web Scanning Engine scanning engine instance.</p> <p>If this parameter is specified, the Dr.Web ConfigD configuration daemon checks that there is always a running component copy of scanning engine that is available to the clients via this socket.</p> <p>Default value: (not specified)</p>
IdleTimeLimit <i>{time interval}</i>	<p>Maximum time that the component can remain idle. If the specified value is exceeded, the component shuts down.</p> <p>Minimum value—10s.</p> <p><i>If the FixedSocketPath parameter is set, this setting is ignored (the component does not finish its operation after the time interval expires).</i></p> <p>Default value: 30s</p>
MaxForks <i>{integer}</i>	<p>Maximum allowed number of child processes run by Dr.Web Scanning Engine, which can be run simultaneously.</p> <p>Default value: Automatically determined as twice the number of available CPU cores; or 4, if the resulting number is less than 4.</p>
MaxForksPerFile <i>{integer}</i>	<p>Maximum allowed number of Dr.Web Scanning Engine child scanning processes, which can be used simultaneously for scanning container files (such as archive).</p> <p>Cannot be less than 1 and greater than the MaxForks parameter value.</p> <p>Default value: Automatically determined as $\text{MaxForks}/2$.</p>
BufferedIo <i>{On Off}</i>	<p>Use buffered input/output (I/O) when checking files.</p>



	<p><i>Using buffered I/O in the FreeBSD and Linux OSes can increase scanning speed of the files on slow disks.</i></p> <p>Default value: Off</p>
WatchdogInterval <i>{time interval}</i>	<p>Rate at which Dr.Web Scanning Engine checks whether child processes are operable in order to detect processes that stopped responding ("watchdog").</p> <p>Default value: 1.5s</p>



Dr.Web Updater

The update component Dr.Web Updater is designed for receiving all available updates for virus databases and anti-virus engine Dr.Web Virus-Finding Engine from Doctor Web update servers component.

If Dr.Web for UNIX File Servers operates in [central protection mode](#), the updates are received from the central protection server (for example, from Dr.Web Enterprise Server); at that, all updates are received from the server via [Dr.Web ES Agent](#), and Dr.Web Updater is not used for downloading updates.

Operating Principles

The component is designed to establish connections to Doctor Web update servers to check for updates for virus databases and anti-virus engine Dr.Web Virus-Finding Engine. The lists of servers which constitute an available update zone are stored in a special file (the file is signed to prevent modification).

If the product is not connected to the central protection server or it is connected to the server in mobile mode, Dr.Web Updater is automatically started by the Dr.Web ConfigD configuration daemon. Startup is performed at periods specified in the [settings](#). The component can be also started by the configuration daemon if the appropriate [command](#) is received from a user (unscheduled update). The component operation scheme is shown in the figure below.

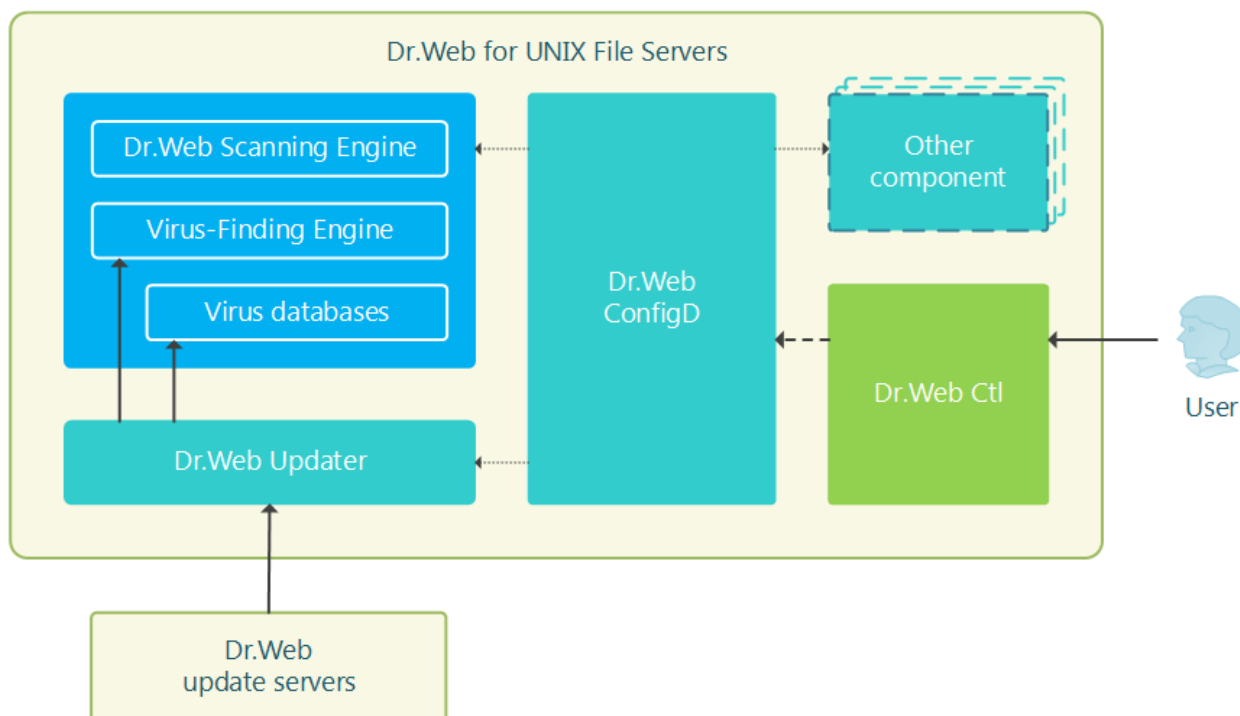


Figure 19. Diagram of the components' operation



When updates become available on the servers, they are downloaded to the `<var_dir>/cache` directory (for **Linux**—`var/opt/drweb.com/cache`), after that they are moved to the working directories of Dr.Web for UNIX File Servers.

By default, all updates are performed from the updating zone which is common for all Dr.Web products. The list of the servers used by default, which are included to the updating zone, is specified in the files which are located in directories, defined in `*Dr1Dir`

Command-Line Arguments

To run Dr.Web Updater, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-update [<parameters>]
```

Dr.Web Updater can process the following options:

Parameter	Description
<code>--help</code>	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: <code>-h</code> Arguments: None.
<code>--version</code>	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: <code>-v</code> Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-update --help
```

This command outputs short help information on Dr.Web Updater.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when needed. To manage the operation of the component, as well as to update virus databases and scanning engine, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-update**



Configuration Parameters

The component uses configuration parameters which are specified in the [Update] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-update</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-update• For FreeBSD: /usr/local/libexec/drweb.com/bin/drweb-update
RunAsUser <i>{UID user name}</i>	<p>The parameter determines under which user name the component should be run. The user name can be specified either as the user's number UID or as the user's login. If the user name consists of numbers (i.e. similar to number UID), it is specified with the "name:" prefix, for example: RunAsUser = name:123456.</p> <p><i>When a user name is not specified, the component operation terminates with an error after the startup.</i></p> <p>Default value: drweb</p>
UpdateInterval <i>{time interval}</i>	<p>The frequency to check for updates on Dr.Web update servers. This is a time period between a previous successful attempt to connect to the update servers (initiated automatically or manually) and the next attempt to perform an update.</p> <p>Default value: 30m</p>
RetryInterval <i>{time interval}</i>	<p>Frequency of repeated attempts to perform an update using the update servers if the previous attempt failed.</p> <p>The parameter can have a value of 1m to 30m inclusively.</p> <p>Default value: 3m</p>
MaxRetries <i>{integer}</i>	<p>Number of repeated attempts to perform an update using the update servers (at the rate specified in RetryInterval) if the previous attempt failed.</p>



	<p><i>If the value is set to 0, repeated attempts are not made (the next update will be performed after the time period specified in UpdateInterval).</i></p> <p>Default value: 3</p>
Proxy {connection string}	<p>Stores the parameters for connecting to a proxy server that is used by the updater component (Dr.Web Updater) when it is connecting to Dr.Web updates servers (for example, if direct connections to external servers are prohibited by your network's security policies).</p> <p>If the parameter value is not specified, the proxy server is not used.</p> <p>Allowed values:</p> <p><connection string>—Proxy server connection string. The string has the following format (URL):</p> <p>[<protocol>://][<user>:<password>@]<proxyhost>:<port></p> <p>where</p> <ul style="list-style-type: none">• <protocol>—Type of the used protocol (in the current version, only http is available).• <user>—Name of the user for connection to proxy.• <password>—Password for connection to proxy.• <proxyhost>—Address of the host where the proxy operates (IP address or domain name, i.e. FQDN).• <port>—Used port. <p>The <protocol> and <user>:<password> parameters can be absent. The address of proxy <proxyhost>:<port> is obligatory.</p> <p><i>If the user name (<user>) or password (<password>) contains the following characters: '@', ',', '%', or ':', these characters must be changed to the following codes: "%40", "%25" and "%3A" respectively.</i></p> <p>Examples:</p> <ol style="list-style-type: none">1. In the configuration file:<ul style="list-style-type: none">• Connection to a proxy on the host <i>proxyhost.company.org</i> using the port 123: Proxy = proxyhost.company.org:123• Connection to the proxy on the node <i>10.26.127.0</i> using the port 3336 over HTTP protocol as user "<i>legaluser</i>" with the password "<i>passw</i>": Proxy = http://legaluser:passw@10.26.127.0:3336• Connection to the proxy on the node <i>10.26.127.0</i> using the port 3336 as a user "<i>user@company.com</i>" with the password "<i>passw%123</i>": Proxy = user%40company.com:passw%25123%3A@10.26.127.0:33362. Specifying the same parameter value via command drweb-ctl cfset:



	<pre># drweb-ctl cfset Update.Proxy proxyhost.company.org:123 # drweb-ctl cfset Update.Proxy http://legaluser:passw@10.26.127.0:3336 # drweb-ctl cfset Update.Proxy user% 40company.com:passw%25123%3A@10.26.127.0:3336</pre> <p>Default value: <i>(not specified)</i></p>
ExcludedFiles <i>{file name}</i>	<p>Defines the name of the file that will not be updated by the Dr.Web Updater component.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add to the list the following files: 123.vdb and 456.dws.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<pre>[Update] ExcludedFiles = "123.vdb", "456.dws"</pre><ul style="list-style-type: none">Two strings (one value per a string)<pre>[Update] ExcludedFiles = 123.vdb ExcludedFiles = 456.dws</pre>Adding values via the command drweb-ctl cfset. <pre># drweb-ctl cfset Update.ExcludedFiles -a 123.vdb # drweb-ctl cfset Update.ExcludedFiles -a 456.dws</pre> <p>Default value: drweb32.lst</p>
NetworkTimeout <i>{time interval}</i>	<p>A time-out period imposed on the network-related operations of the updater component during the updating process.</p> <p>This parameter is used when a connection is temporarily lost: if the connection is established again before the timeout expires, the interrupted updating process will be continued.</p> <p><i>Specifying the time out value larger than 75s has no effect as the connection is closed by TCP timeout. The minimum allowed value is 5s.</i></p> <p>Default value: 60s</p>
BaseDr1Path <i>{path to file}</i>	<p>Path to the signed file that contains the list of update servers of a standard update zone, which are used by the update component for updating virus databases and anti-virus engine.</p>



	<p>Default value: <var_dir>/drl/bases/update.drl</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/drl/bases/update.drl• For FreeBSD: /var/drweb.com/drl/bases/update.drl
BaseCustomDrlPath <i>{path to file}</i>	<p>Path to the signed file that contains the list of update servers of a special update zone, which are used by the update components for updating virus databases and anti-virus engine.</p> <p><i>If the parameter is not empty, and the specified file exists, only servers are used for the update. The main file of the list (see above) is ignored. If the file identified by the parameter is empty, the update attempt will fail.</i></p> <p>Default value: <var_dir>/drl/bases/update.drl</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/drl/bases/custom.drl• For FreeBSD: /var/drweb.com/drl/bases/custom.drl
BaseUpdateEnabled <i>{Boolean}</i>	<p>Indicator that shows whether or not updating of virus databases and anti-virus engine is allowed.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Yes—updating is allowed and will be performed.• No—updating is not allowed and will not be performed. <p>Default value: Yes</p>
VersionDrlPath <i>{path to file}</i>	<p>Defines a path to the signed file that contains the list of the servers of an update zone, which are used by the update component for updating Dr.Web for UNIX File Servers components.</p> <p>Default value: <var_dir>/drl/version/update.drl</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/drl/version/update.drl• For FreeBSD: /var/drweb.com/drl/version/update.drl
VersionUpdateEnabled <i>{Boolean}</i>	<p>Indicator that shows whether or not updating of Dr.Web for UNIX File Servers component's version is allowed.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• Yes—updating is allowed and will be performed.• No—updating is not allowed and will not be performed. <p>Default value: Yes</p>
DwsDrlPath <i>{path to file}</i>	<p>The parameter is not used.</p> <p>Default value: <var_dir>/drl/dws/update.drl</p> <ul style="list-style-type: none">• For Linux, Solaris: /var/opt/drweb.com/drl/dws/update.drl• For FreeBSD: /var/drweb.com/drl/dws/update.drl



DwsCustomDrlPath <i>{path to file}</i>	<p>The parameter is not used.</p> <p>Default value: <code><var_dir>/drl/dws/custom.drl</code></p> <ul style="list-style-type: none">• For Linux, Solaris: <code>/var/opt/drweb.com/drl/dws/custom.drl</code>• For FreeBSD: <code>/var/drweb.com/drl/dws/custom.drl</code>
DwsUpdateEnabled <i>{Boolean}</i>	<p>The parameter is not used.</p> <p>Default value: <code>Yes</code></p>
AntispamDrlPath <i>{path to file}</i>	<p>The parameter is not used.</p> <p>Default value: <code><var_dir>/drl/antispam/update.drl</code></p> <ul style="list-style-type: none">• For Linux, Solaris: <code>/var/opt/drweb.com/drl/antispam/update.drl</code>• For FreeBSD: <code>/var/drweb.com/drl/antispam/update.drl</code>
AntispamCustomDrlPath <i>{path to file}</i>	<p>The parameter is not used.</p> <p>Default value: <code><var_dir>/drl/antispam/custom.drl</code></p> <ul style="list-style-type: none">• For Linux, Solaris: <code>/var/opt/drweb.com/drl/antispam/custom.drl</code>• For FreeBSD: <code>/var/drweb.com/drl/antispam/custom.drl</code>
AntispamUpdateEnabled <i>{Boolean}</i>	<p>The parameter is not used.</p> <p>Default value: <code>No</code></p>
BackupDir <i>{path to directory}</i>	<p>Path to the directory, where the previous versions of updated files are saved for possible rollback. Upon every update only updated files are backed up.</p> <p>Default value: <code>/tmp/update-backup</code></p>
MaxBackups <i>{integer}</i>	<p>The maximum number of the previous versions of updated files, which are saved. If this number is exceeded the oldest copy is removed upon the next update.</p> <p><i>If the parameter value is zero, the previous versions of the files are not saved.</i></p> <p>Default value: <code>0</code></p>



Dr.Web ES Agent

Central anti-virus protection agent Dr.Web ES Agent is designed for connecting Dr.Web for UNIX File Servers to the [central protection](#) server (for example, to Dr.Web Enterprise Server).

When Dr.Web for UNIX File Servers is connected to the central protection server Dr.Web ES Agent, the license [key file](#) is synchronized according to

Operating Principles

Dr.Web ES Agent establishes connection to the central protection server (for example, to Dr.Web Enterprise Server), which allows the network administrator to implement common security policy within the network, in particular, configure the same scanning settings and reaction on threat detection for all network stations and servers. Moreover, the central protection server also performs a role of an internal update server on the network, as it stores up-to-date virus databases, components (in this case, updating is performed via Dr.Web ES Agent, [Dr.Web Updater](#) is not used).

When connecting Dr.Web ES Agent to the central protection server, the agent ensures receipt of up-to-date settings for the program components and the license key file, which are then transmitted to the [Dr.Web ConfigD](#) configuration daemon for applying them to managed components. Moreover, the component also receives tasks to scan file system objects on the station (including scheduled tasks).



Note that the current version of Dr.Web for UNIX File Servers *does not* fully implement the central protection mode: the central protection server cannot manage operation settings of the program components.

Dr.Web ES Agent collects and sends the server statistics on detected threats and applied actions. The operation scheme is shown in the figure below.

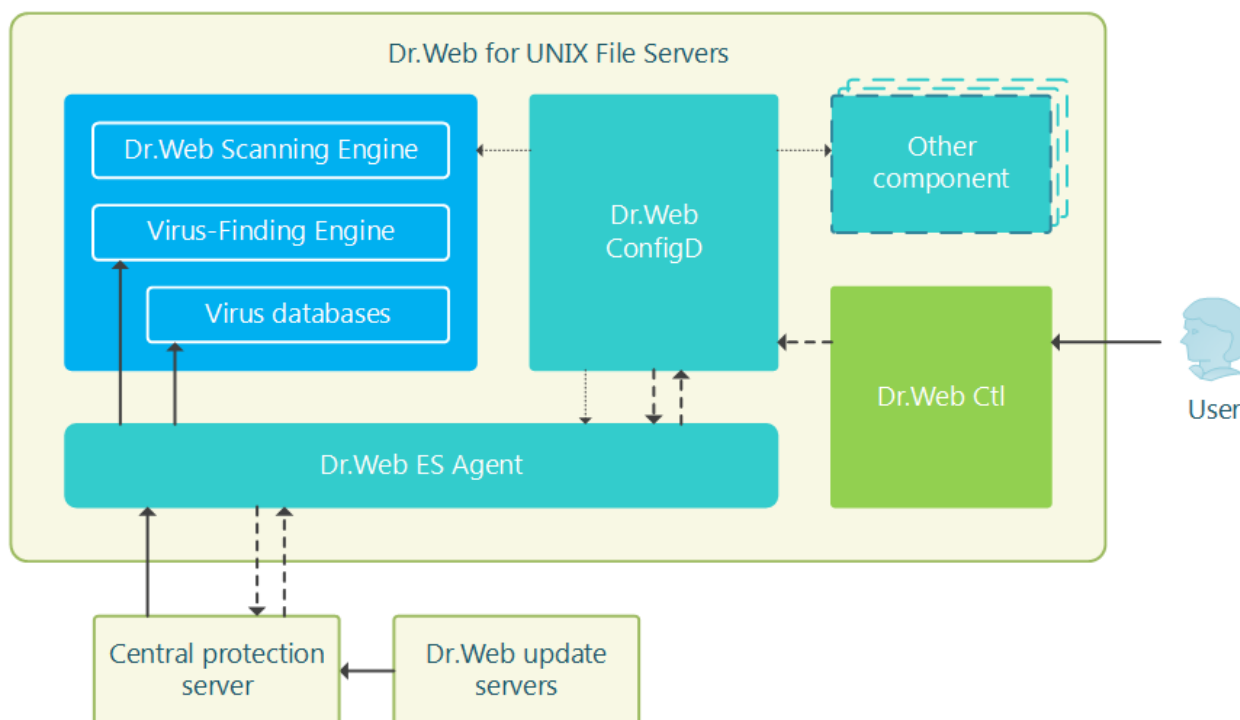


Figure 20. Diagram of the components' operation

To connect Dr.Web ES Agent to the central protection server, the password and identifier of the host ("station" in terms of the Central protection server) are required, as well as the public encryption key file, which is used by the server for authentication. Instead of the station identifier, you can specify the identifier of the main and tariff groups where the station is to be included. For required identifiers and public key file, contact the administrator of your anti-virus network.

Moreover, if this option is allowed on the central protection server, you can connect your host with the protected server ("workstation") as a "newbie". In this case, after the administrator confirms the request to connect, the central protection server automatically generates an identifier and a password, and sends them to the agent for future connections.

Command-Line Arguments

To run Dr.Web ES Agent, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-esagent [<parameters>]
```

Dr.Web ES Agent can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.



<code>--version</code>	<p>Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion.</p> <p>Short form: <code>-v</code></p> <p>Arguments: None.</p>
------------------------	---

Example:

```
$ /opt/drweb.com/bin/drweb-esagent --help
```

This command outputs short help information on Dr.Web ES Agent.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon at the startup of the operating system. To manage the operation of the component, as well as to connect Dr.Web for UNIX File Servers to central protection server, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the `drweb-ctl` [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-esagent**

Configuration Parameters

The component uses configuration parameters which are specified in the `[ESAgent]` section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <code><opt_dir>/bin/drweb-esagent</code></p> <ul style="list-style-type: none">• For Linux, Solaris: <code>/opt/drweb.com/bin/drweb-esagent</code>• For FreeBSD: <code>/usr/local/libexec/drweb.com/bin/drweb-esagent</code>



DebugIpc <i>{Boolean}</i>	<p>Indicates whether detailed IPC messages are included in the log at debug level (if LogLevel = DEBUG) (interaction of Dr.Web ES Agent and the Dr.Web ConfigD configuration daemon).</p> <p>Default value: No</p>
MobileMode <i>{On Off Auto}</i>	<p>Indicates whether the program can operate in mobile mode when connected to central protection server.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• On—instructs to use mobile mode if it is allowed by central protection server (that is, perform updates from update servers of Doctor Web via Dr.Web Updater).• Off—instructs not to use mobile mode and continue operation in central protection mode (updates are always received from the central protection server).• Auto—instructs to use mobile mode, if allowed by central protection server, and perform updates both from update servers of Doctor Web via Dr.Web Updater and from central protection server, depending on which connection is available and which connection quality is higher. <p><i>Note that behavior of this parameter depends on server permissions: if mobile mode is not allowed on the used server, this parameter has no effect.</i></p> <p>Default value: Auto</p>
Discovery <i>{On Off}</i>	<p>Indicates whether the agent is allowed to receive <i>discovery</i> requests from the network inspector built in the central protection server (<i>discovery</i> requests are used by the inspector to check the structure and state of the anti-virus network).</p> <p>Allowed values:</p> <ul style="list-style-type: none">• On—allow the agent to receive and process <i>discovery</i> requests.• Off—prohibit the agent to receive and process <i>discovery</i> requests. <p><i>Note that this parameter has higher priority than settings of the central protection server: if the parameter value is set to Off, the agent does not receive discovery requests even if this option is enabled on the server.</i></p> <p>Default value: On</p>
UpdatePlatform <i>{platform name}</i>	<p>Indicates the agent to receive updates for the anti-virus engine from the central protection server. The anti-virus engine was developed for the indicated platform, where the <i>platform name</i> is a string, which contains the platform name.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• GNU/Linux: unix-linux-32, unix-linux-64, unix-linux-mips• FreeBSD: unix-freebsd-32, unix-freebsd-64• Solaris: unix-solaris-32, unix-solaris-64• Darwin: unix-darwin-32, unix-darwin-64



It is strongly recommended to change the parameter value only if you are sure it is required.

Default value: *Depends on the currently used platform*



Dr.Web HTTPD

Dr.Web HTTPD provides infrastructure for local and remote interaction with Dr.Web for UNIX File Servers via HTTP (for example, via a web browser). The component provides two interfaces: an interface to manage the product (in addition to installing the Dr.Web HTTPD, a separate package must also be installed containing the files of the web interface for managing the product via a web browser) and a service interface used by the Dr.Web Link Checker component, which is an extension for **Mozilla Firefox** and **Google Chrome** browsers (installed separately).

Besides managing Dr.Web for UNIX File Servers through the product's web interface, it is also possible to use the command interface (HTTP API) of Dr.Web HTTPD directly to interact with the components of Dr.Web for UNIX File Servers via HTTPS. This capability allows you to create your own interface to manage Dr.Web for UNIX File Servers.

To use a secure HTTPS connection it is required to provide an appropriate SSL server certificate and private key for Dr.Web HTTPD. By default, an SSL server certificate and an SSL private key are generated for Dr.Web HTTPD automatically during the installation procedure, but, if necessary, you can generate your own certificate and key. If you need to generate SSL keys and certificates, you can use the **openssl** utility. An example of how to use the **openssl** utility to generate a certificate and a private key is given in the section [Appendix E. Generating SSL certificates](#).

Operating Principles

Dr.Web HTTPD is a web server for managing the operation of Dr.Web for UNIX File Servers. With Dr.Web HTTPD, it is possible not to use external web servers (for example, **Apache HTTP Server** or **Nginx**) and management services like **Webmin**. Moreover, the component can function simultaneously with such servers and services on the same host and not impede their operation at that.

The Dr.Web HTTPD server processes requests received via HTTP and HTTPS protocols to the sockets specified in the settings. For this reason, the server does not have any conflicts with web servers when they operate on the same host. The operation scheme is shown in the figure below.

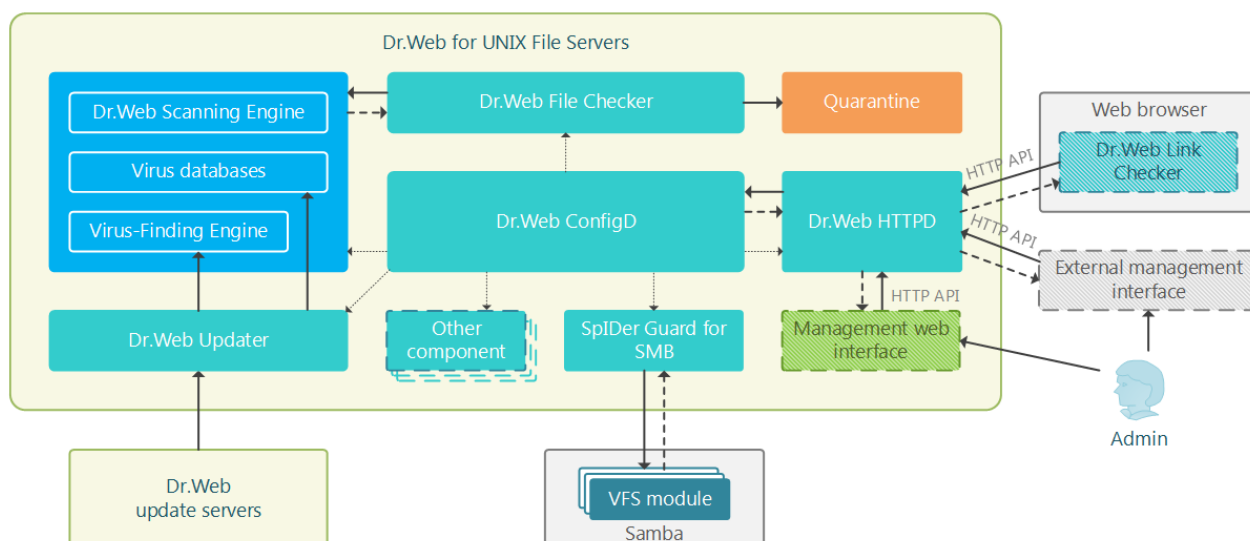


Figure 21. Diagram of the components' operation

HTTPS is used for product management; and HTTP, for processing requests from Dr.Web Link Checker—a web browser extension—it is installed to the browser separately).



It is not mandatory to install Dr.Web management web interface and Dr.Web Link Checker extension for the proper functioning of the product. They can be missing. This is why the corresponding blocks are circled with a dashed line.

The Dr.Web HTTPD component issues commands to the Dr.Web for UNIX File Servers [Dr.Web ConfigD](#) configuration daemon, as well as to the [Dr.Web File Checker](#) component for file checking, and to [SpIDer Guard for NSS](#) monitor. These commands are based on those that were received through the provided HTTP API (including those that were made via the management web interface or those made as requests from the Dr.Web Link Checker browser extension).

If the management web interface of Dr.Web for UNIX File Servers, which uses Dr.Web HTTPD, is included in the product, it is described in the corresponding [section](#).

If the Dr.Web's management web interface is not included in the product, it is possible to connect any external management interface to the product. For interaction with the product's components, this interface should use the HTTP API provided by Dr.Web HTTPD (not described in this user manual).



To obtain the HTTP API Dr.Web HTTPD manual, refer to Doctor Web partners care department (<https://partners.drweb.com/>).



Command-Line Arguments

To launch Dr.Web HTTPD from the command line of the operating system, the following command is used:

```
$ <opt_dir>/bin/drweb-httpd [<parameters>]
```

Dr.Web HTTPD can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-httpd --help
```

This command outputs short help information on Dr.Web HTTPD.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when required (usually at the startup of the operating system). If the component is running and the web interface is installed, then to manage the components of Dr.Web for UNIX File Servers, you can simply use any standard web-browser to access, via HTTPS, any of the addresses at which the web-interface is served. To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the drweb-ctl [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-httpd**

Configuration Parameters

The component uses configuration parameters which are specified in the [HTTPD] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.



The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-httpd</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-httpd• For FreeBSD: /usr/local/libexec/drweb.com/bin/drweb-httpd
Start <i>{Boolean}</i>	<p>The component must be launched by the Dr.Web ConfigD configuration daemon.</p> <p>When you specify the Yes value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the No value, it instructs the configuration daemon to terminate the component immediately.</p> <p>It depends on whether product management interface is installed.</p>
WebConsoleAddress <i>{address, ...}</i>	<p>List of network sockets (every network socket specified as <IP address>:<port>) on which Dr.Web HTTPD is listening for connections to the web interface provided for managing the product, if the web interface is installed. Access to the web-interface at these Addresses is provided via HTTPS.</p> <p><i>If no value is specified, it is impossible to use the web interface.</i></p> <p>Default value: 127.0.0.1</p>
LinkCheckerAddress <i>{address, ...}</i>	<p>List of network sockets (every network socket consists of <IP address>:<port>) on which Dr.Web HTTPD is listening for connections (via HTTP) from the Dr.Web Link Checker browser extension that checks webpages loaded by the browser for malicious objects.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add sockets 192.168.0.1:1234 and 10.20.30.45:5678 to the list.</p> <ol style="list-style-type: none">1. Adding of values to the configuration file.<ul style="list-style-type: none">• Two values in one string



	<div>[<%HTTPD_SECTION%>] LinkCheckerAddress = "192.168.0.1:1234", "10.20.30.45:5678"</div> <ul style="list-style-type: none">• Two strings (one value per a string) <div>[<%HTTPD_SECTION%> LinkCheckerAddress = 192.168.0.1:1234 LinkCheckerAddress = 10.20.30.45:5678</div> <p>2. Adding values via the command drweb-ctl cfset.</p> <div># drweb-ctl cfset <%HTTPD_SECTION%> >.LinkCheckerAddress -a 192.168.0.1:1234 # drweb-ctl cfset <%HTTPD_SECTION%> >.LinkCheckerAddress -a 10.20.30.45:5678</div> <p><i>If no value is specified, it is impossible to use the Dr.Web Link Checker browser extension. Note that at these addresses (sockets) you cannot access the web-interface provided for managing the product.</i></p> <p>Default value: (not specified)</p>
ServerSslCertificate {path to file}	<p>Path to the file with the server certificate used by the web interface server for communication with other hosts via HTTPS.</p> <p><i>This file is generated automatically during the installation of the component.</i></p> <p><i>Please note that the certificate file and the private key file (which is specified by a parameter described below) must form a matching pair.</i></p> <p>Default value: <etc_dir>/certs/serv.crt</p> <ul style="list-style-type: none">• For Linux, Solaris: /etc/opt/drweb.com/certs/serv.crt• For FreeBSD: /usr/local/etc/drweb.com/certs/serv.crt
ServerSslKey {path to file}	<p>Path to the private key file used by the web interface server for communication with other hosts via HTTPS.</p> <p><i>This file is generated automatically during the installation of the component.</i></p> <p><i>Please note that the certificate file (which is specified by the previous discussed parameter) and the private key file must form a matching pair.</i></p> <p>Default value: <etc_dir>/certs/serv.key</p> <ul style="list-style-type: none">• For Linux, Solaris: /etc/opt/drweb.com/certs/serv.key• For FreeBSD: /usr/local/etc/drweb.com/certs/serv.key
WebconsoleRoot {path to directory}	<p>Path to the directory which stores the files used by the management web interface if it is installed (similar to the htdocs directory of an Apache HTTP Server).</p> <p>Default value: <opt_dir>/share/drweb-httpd/webconsole</p>



	<ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/share/drweb-httpd/webconsole• For FreeBSD: /usr/local/libexec/drweb.com/share/drweb-httpd/webconsole
LinkcheckerRoot <i>{path to directory}</i>	<p>Path to the directory which stores the files used by Dr.Web Link Checker web browser extension.</p> <p>Default value: <opt_dir>/share/drweb-httpd/linkchecker</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/share/drweb-httpd/linkchecker• For FreeBSD: /usr/local/libexec/drweb.com/share/drweb-httpd/linkchecker
AccessLogPath <i>{path to file}</i>	<p>Path to the file where all HTTP/HTTPS requests from clients to the web interface server are registered.</p> <p><i>If not specified, HTTP/HTTPS requests are not logged to a file.</i></p> <p>Default value: (not specified)</p>



Dr.Web SNMPD

Dr.Web SNMP agent (Dr.Web SNMPD) is designed for integration of Dr.Web for UNIX File Servers suite with monitoring systems via SNMP. Such integration will allow to control operational status of Dr.Web for UNIX File Servers as well as collect statistics on detected and neutralized threats. The agent provides the following information to monitoring systems and SNMP managers:

- State of a program component
- Number of detected threats of various types (according to the Dr.Web classification)

Moreover, the agent sends SNMP trap notifications upon detection of a threat and upon failures in neutralization of detected threats. The agent supports SNMP protocol of version 2c and 3.

Description of the information which can be sent by the agent is stored in a special section of MIB (*Management Information Base*) created by Doctor Web. In the MIB section, defined by Dr.Web for UNIX-like operating systems, the following information is specified:

1. Formats of SNMP trap notifications about detection and neutralizing of threats and about errors related to the program components.
2. Operation statistics of the program components:
3. Current state of the program components

For more details about information that can be obtained over the SNMP protocol, refer to the corresponding [section](#).

Operating Principles

By default, the component is run automatically upon Dr.Web for UNIX File Servers startup. When run, the component structures data according to the structure described in MIB Dr.Web and waits for requests to receive data from external SNMP managers. The component receives information on the status of the program components and notifications on detected threats directly from the [Dr.Web ConfigD](#) configuration daemon, as shown in the figure below.

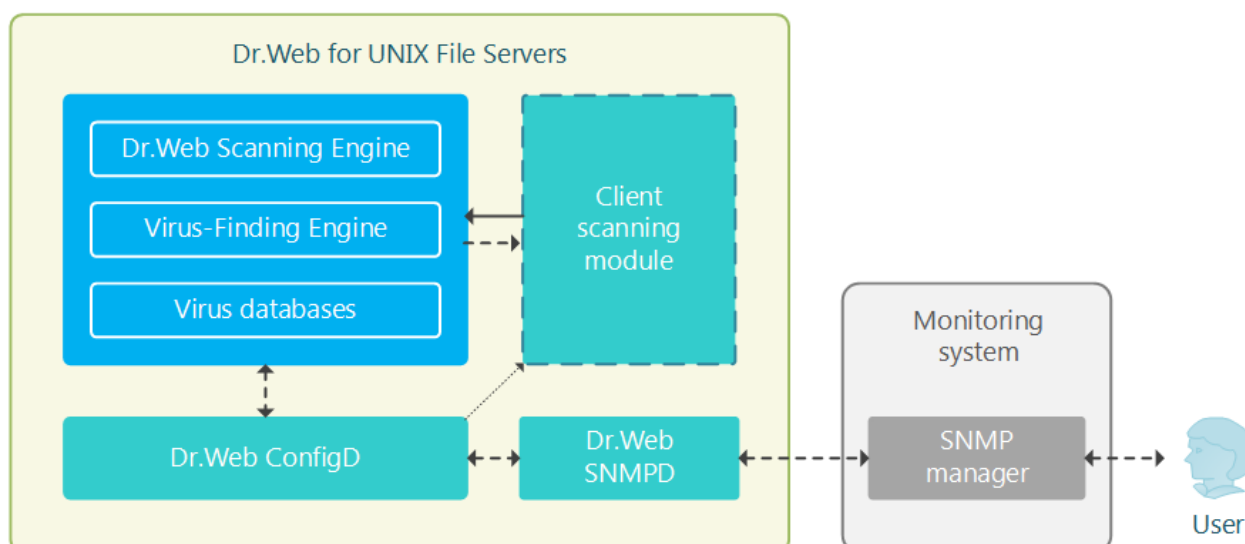


Figure 22. Diagram of the components' operation

Threats can be detected by the scanning engine during scanning initiated by Dr.Web for UNIX File Servers components; thus, the scheme contains an abstract "client scanning module". On threat detection, the appropriate count (of this threat type) is incremented by one and all SNMP managers that can receive notifications get an SNMP trap notifying on the detected threat.



Collected values of counters (for example, counters of detected threats) are not saved between launches of Dr.Web SNMPD. Thus, once Dr.Web SNMPD is relaunched for any reason (including general restart of Dr.Web for UNIX File Servers), the collected values of counters are reset to 0.

Integration with the System SNMP Agent

To enable correct operation of Dr.Web SNMP agent if the main system SNMP agent **snmpd** (**net-snmp**), already operates on the server, configure transmission of SNMP requests through the Dr.Web MIB branch from **snmpd** to Dr.Web SNMPD. For that purpose, edit the **snmpd** configuration file (usually for **Linux** the file is as follows: `/etc/snmp/snmpd.conf`), by adding the following line in it:

```
proxy -v <version> -c <community> <address>:<port> <MIB branch>
```

Where:

- **<version>** – SNMP version in use (2c, 3).
- **<community>**—"community string" used by Dr.Web SNMPD.
- **<address>:<port>**—network socket which is listened by Dr.Web SNMPD.
- **<MIB branch>**—OID of the MIB branch containing [descriptions](#) of variables and SNMP notifications (*trap*) used by Dr.Web (the OID equals `.1.3.6.1.4.1.29690`).



If you are using the default settings of Dr.Web SNMP agent, then the added line should look like this:

```
proxy -v 2c -c public localhost:50000 .1.3.6.1.4.1.29690
```

Note that since port 161 in this case will be used by the system's standard **snmpd**, it is required to specify another port for Dr.Web SNMPD in the ListenAddress [parameter](#) (in this example, 50000).

Command-Line Arguments

To launch Dr.Web SNMPD from the command line of the operating system, the following command is used:

```
$ <opt_dir>/bin/drweb-snmpd [<parameters>]
```

Dr.Web SNMPD can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-snmpd --help
```

This command outputs short help information on Dr.Web SNMPD.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when needed (as a rule, at the startup of the operating system). To manage the operation of the component, you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the drweb-ctl [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-snmpd**

Configuration Parameters

The component uses configuration parameters which are specified in the [SNMPD] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	Logging level of the component. If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used. Default value: Notice
Log <i>{log type}</i>	Logging method
ExePath <i>{path to file}</i>	Path to the executable file of the component. Default value: <opt_dir>/bin/drweb-snmpd <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-snmpd• For FreeBSD : /usr/local/libexec/drweb.com/bin/drweb-snmpd
Start <i>{Boolean}</i>	The component must be launched by the Dr.Web ConfigD configuration daemon. When you specify the Yes value for this parameter, it instructs the configuration daemon to start the component immediately; and when you specify the No value, it instructs the configuration daemon to terminate the component immediately. Default value: No
RunAsUser <i>{UID user name}</i>	The parameter determines under which user name the component should be run. The user name can be specified either as the user's number UID or as the user's login. If the user name consists of numbers (i.e. similar to number UID), it is specified with the "name:" prefix, for example: RunAsUser = name:123456.



	<p>When a user name is not specified, the component operation terminates with an error after the startup.</p> <p>Default value: <code>drweb</code></p>
ListenAddress {address}	<p>Address (IP address and port) listened by Dr.Web SNMPPD, which is waiting for client connections (SNMP managers).</p> <p><i>Note that interaction with snmpd requires a specified port, different from the standard port (161), and snmpd must be configured for proxying.</i></p> <p>Default value: <code>127.0.0.1:161</code></p>
SnmpVersion {V2c V3}	<p>The used SNMP protocol version (SNMPv2c or SNMPv3).</p> <p>Default value: <code>V2c</code></p>
V3EngineId {string}	<p>Identifier (string) of <i>Engine ID</i> for SNMPv3 (according to RFC 3411).</p> <p>Default value: <code>800073FA044452574542</code></p>
TrapReceiver {address list}	<p>List of addresses (IP address and port) where Dr.Web SNMPPD sends <i>SNMP trap</i> notifications after Dr.Web for UNIX File Servers components detected a threat.</p> <p><i>You can specify a list as the parameter value. The values in the list must be separated with commas (each value in the quotation marks). The parameter can be specified more than once in the section (in this case, all its values are combined into one list).</i></p> <p>Example: Add sockets 192.168.0.1:1234 and 10.20.30.45:5678 to the list.</p> <ol style="list-style-type: none">Adding of values to the configuration file.<ul style="list-style-type: none">Two values in one string<div><pre>[SNMPPD] TrapReceiver = "192.168.0.1:1234", "10.20.30.45:5678"</pre></div>Two strings (one value per a string)<div><pre>[SNMPPD] TrapReceiver = 192.168.0.1:1234 TrapReceiver = 10.20.30.45:5678</pre></div>Adding values via the command drweb-ctl <code>cfset</code>.<div><pre># drweb-ctl cfset SNMPPD.TrapReceiver - a 192.168.0.1:1234 # drweb-ctl cfset SNMPPD.TrapReceiver - a 10.20.30.45:5678</pre></div>



	Default value: <i>(not set)</i>
V2cCommunity <i>{string}</i>	<p>The string "SNMP read community" for authentication of SNMP managers (<i>SNMPv2c</i> protocol) when Dr.Web MIB variables are accessed for reading.</p> <p><i>The parameter is used if SnmVersion = V2c.</i></p> <p>Default value: <code>public</code></p>
V3UserName <i>{string}</i>	<p>The user name for authentication of SNMP managers (<i>SNMPv3</i> protocol) when Dr.Web MIB variables are accessed for reading.</p> <p><i>The parameter is used if SnmVersion = V3.</i></p> <p>Default value: <code>noAuthUser</code></p>
V3Auth <i>{SHA(<pwd>) MD5(<pwd>) None}</i>	<p>Method to authenticate SNMP managers (<i>SNMPv3</i> protocol) when Dr.Web MIB variables are accessed for reading.</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <code>SHA (<PWD>)</code> —SHA hash of the password is used (<PWD> strings).• <code>MD5 (<PWD>)</code> —MD5 hash of the password is used (<PWD> strings).• <code>None</code>—authentication is disabled. <p>where <PWD> is a plain text password.</p> <p>When specifying the parameter value from the command line, you may need to escape the brackets by using the slash mark \ in some shells.</p> <p>Examples:</p> <ol style="list-style-type: none">1. Parameter value in the configuration file: V3Auth = MD5 (123456)2. Specifying the same parameter value from the command line via command drweb-ctl <code>cfset</code>: drweb-ctl <code>cfset SNMPD.V3Auth MD5\ (123456\)</code> <p><i>The parameter is used if SnmVersion = V3.</i></p> <p>Default value: <code>None</code></p>
V3Privacy <i>{DES(<secret>) AES128(<secret>) None}</i>	<p>Method to encrypt SNMP messages (<i>SNMPv3</i> protocol).</p> <p>Allowed values:</p> <ul style="list-style-type: none">• <code>DES (<secret>)</code> —DES encryption algorithm is used.



- **AES128 (<secret>)**—AES128 encryption algorithm is used.
- **None**—SNMP-messages are not encrypted.

where <secret> is a secret key shared by the manager and the agent (*plain text*).

When specifying the parameter value from the command line, you may need to escape the brackets by using the slash mark \ in some shells.

Examples:

1. Parameter value in the configuration file:
V3Privacy = AES128(supersecret)
2. Specifying the same parameter value from the command line via **command drweb-ctl** cfset:
drweb-ctl cfset SNMPD.V3Privacy
AES128\ (supersecret\)

*The parameter is used if **SnmpVersion** = V3.*

Default value: None

Integration with SNMP Monitoring Systems

Dr.Web SNMP agent can perform functions of a data provider for any monitoring system that uses SNMP protocol version 2c or 3. The list of available data and their structure are [provided](#) in a Dr.Web MIB description file called `DrWeb-Snmpd.mib`, supplied with the product. This file is located in the `<opt_dir>/share/drweb-snmpd/mibs` directory.

For easy configuration, the component is supplied with templates of settings for popular monitoring systems:

- [Munin](#)
- [Nagios](#)
- [Zabbix](#)

Customization templates for monitoring systems are located in the `<opt_dir>/share/drweb-snmpd/connectors` directory.

Integration with Munin Monitoring System

The **Munin** monitoring system includes the central server (master) **munin**, which collects statistics from clients **munin-node** residing locally on the monitored hosts. At request of the server, each monitoring client collects data about monitored host operation by starting *plug-ins* that provide data transferred to the server.



To enable connection between Dr.Web SNMPD and the **Munin** monitoring system, a ready-to-use plug-in Dr.Web used by **munin-node** is supplied. The plug-in resides in the `<opt_dir>/share/drweb-snmpd/connectors/munin/plugins` directory. This plug-in collects data required for construction of the following two graphs:

- Number of detected threats
- File scan statistics
- Email message scanning statistics.

These plug-ins support SNMP protocols versions 1, 2c and 3. Based on these template plug-ins, you can create any other plug-ins to poll the status of Dr.Web for UNIX File Servers components via Dr.Web SNMPD.

In the `<opt_dir>/share/drweb-snmpd/connectors/munin` directory, the following files are located.

File	Description
<code>plugins/snmp__drweb_malware</code>	The munin-node plug-in for polling Dr.Web SNMPD via SNMP to gather the number of threats detected by Dr.Web on the host.
<code>plugins/snmp__drweb_filecheck</code>	The munin-node plug-in for polling Dr.Web SNMPD via SNMP to gather the statistics of files scanned by Dr.Web on the host.
<code>plugins/snmp__drweb_maild_multi</code>	<p>The munin-node plug-in used for polling Dr.Web SNMPD via SNMP to gather the statistics of email messages scanned by Dr.Web on the host.</p> <p>Note that this plug-in uses the <i>multigraph</i>, a feature available in Munin version above 1.4.</p>
<code>plugin-conf.d/drweb.cfg</code>	An example of the munin-node configuration for the environment variables of the Dr.Web plug-ins.

Connecting a host to Munin

In the present instruction, it is assumed that the **Munin** monitoring system is already deployed on the monitoring server and the monitored host features an installed and functioning Dr.Web SNMPD (it is possible for the component to function in [proxy](#) mode together with **snmpd**) and **munin-node**.

1. Monitored host configuration

- Copy the `snmp__drweb_*` files to the directory with plug-in libraries **munin-node** (the directory depends on the OS). For example, in **Debian/Ubuntu** operating systems, the path is `/usr/share/munin/plugins`.



- Configure **munin-node** by connecting to it the supplied Dr.Web plug-ins. To do this, use the **munin-node-configure** utility that is distributed with **munin-node**.

For example, the following command

```
$ munin-node-configure --shell --snmp localhost
```

will display on a terminal screen a list of commands for creation of required symbolic links to plug-ins. Copy and execute them in the command line. Note that the specified command presumes that:

- 1) **munin-node** is installed at the same host where Dr.Web SNMPD is installed. If it is not the case, please specify the correct FQDN or an IP address of the monitored host instead of a `localhost` value;
 - 2) Dr.Web SNMPD uses SNMP version 2c. If it is not the case, specify the correct SNMP version in **munin-node-configure** command. The command has several arguments for flexible configuration of plug-ins, e.g., you can specify the SNMP protocol version, port that is listened by SNMP agent at the monitored host, an actual value of the *community string*, and so on. If required, refer to the manual on **munin-node-configure** command.
- If necessary, define (or redefine) parameter values of the environment, where installed Dr.Web plug-ins for **munin-node** must be executed. As the environment parameters, the value *community string* is used. It is the port utilized by the SNMP agent, etc. These parameters must be defined in the file `/etc/munin/plugin-conf.d/drweb` (create it if required). As an example of this file, use the supplied file `drweb.cfg`.
 - In the **munin-node** configuration file (`munin-node.conf`), specify a regular expression to include all IP addresses of hosts that are allowed to connect **munin** servers (masters) to **munin-node** for receiving the values of the monitored parameters, for example:

```
allow ^10\.20\.30\.40$
```

In this case, only the IP address `10.20.30.40` is allowed to receive host parameters.

- Restart **munin-node**, for example, by using the following command:

```
# service munin-node restart
```

2. Munin server (master) configuration

Add the address and identifier of the monitored host to the **Munin** configuration file `munin.conf`, which is located, by default, in `/etc` directory (in **Debian/Ubuntu** operating systems it is `/etc/munin/munin.conf`):

```
[ <ID>; <hostname> . <domain> ]  
address <host IP address>  
use_node_name yes
```

where `<ID>` is the displayed host's identifier, `<hostname>` is the name of the host, `<domain>` is the name of the domain, `<host IP address>` is the IP address of the host.

For official documentation on configuration of the **Munin** monitoring system, refer to <http://munin.readthedocs.io>.



Integration with Zabbix Monitoring System

File templates, required for establishing connection between Dr.Web SNMPD and the **Zabbix** monitoring system, are located in the `<opt_dir>/share/drweb-snmpd/connectors/zabbix` directory.

File	Description
<code>zbx_drweb.xml</code>	Template for description of the monitored host that features installed Dr.Web.
<code>snmptt.drweb.zabbix.conf</code>	Configuring the snmptt utility—which is an <i>SNMP trap</i> handler

Template for description of the monitored host features:

- Description of counters ("*items*", according to the terminology of **Zabbix**). By default, the template is set to be used with SNMP v2.
- The set of predefined graphs: number of scanned files and distribution of detected threats by their type.

Connecting a host to Zabbix

In the present instruction, it is assumed that the **Zabbix** monitoring system is already deployed on the monitoring server and the monitored host features an installed and functioning Dr.Web SNMPD (it is possible for the component to function in [proxy](#) mode together with **snmpd**). Moreover, if you want to receive *SNMP trap* notifications from the monitored host (including notification on threats detected by Dr.Web for UNIX File Servers on a protected server), install the `net-snmp` package on the monitoring server (standard tools **snmptt** and **snmptrapd** are used).

1. In the **Zabbix** web interface, on the **Configuration** → **Templates** tab import the template of the monitored host from the `<opt_dir>/share/drweb-snmpd/connectors/zabbix/zbx_drweb.xml` file.
2. Add the monitored host to the appropriate list (at **Hosts** → **Create host**). Specify correct parameters of the host and settings of the SNMP interface (they must match the settings of **drweb-snmpd** and **snmpd** on the host):
 - The **Host** tab:
 - Host name:** `drweb-host`
 - Visible name:** `DRWEB_HOST`
 - Groups:** select *Linux servers*
 - Snmp interfaces:** Click **add** specify the IP address and port are used by Dr.Web SNMPD (it is considered that Dr.Web SNMPD operates on the local host, so the address `127.0.0.1` and the port `161` are specified by default).
 - The **Templates** tab:



Click **Add**, check *DRWEB*, click **select**.

- The **Macros** tab:

Macro: `{{SNMP_COMMUNITY}}`

Value: specify "read community" for SNMP V2c (by default, *public*).

Click **Save**.

Note: The `{{SNMP_COMMUNITY}}` macro can be specified directly in the host template.



By default, the imported *DRWEB* template is configured for SNMP v2. If you need to use another version of SNMP, edit the template accordingly on the appropriate page.

3. After the template is bound to the monitored host, if SNMP settings are specified correctly, the **Zabbix** monitoring system will start to collect data for counters (*items*) of the template; the collected data will be displayed on the **Monitoring** → **Latest Data** and **Monitoring** → **Graphs tabs**.
4. A special *item drweb-traps* is used for collecting *SNMP trap* notifications from Dr.Web SNMPD. The log of received *SNMP trap* notifications is available on the **Monitoring** → **Latest Data** → **drweb-traps** → **history** page. To collect notifications, **Zabbix** uses standard tools **snmptt** and **snmptrapd** from the `net-snmp` package. For details on how to configure the tools for receiving *SNMP trap* notifications from Dr.Web SNMPD, see below.
5. If necessary, you can configure a trigger that will change its state upon receiving an *SNMP trap* notification from Dr.Web SNMPD. Changing of its state can be used as an event source for generation appropriate notifications. The example below shows an expression for configuration of a trigger; the expression is specified in the **trigger expression** field:

- For **Zabbix 2.x**:

```
{TRIGGER.VALUE}=0 &  
{DRWEB:snmptrap[.*\1\3\6\1\4\1\29690\..*].nodata(60)}=1 ) |  
{TRIGGER.VALUE}=1 &  
{DRWEB:snmptrap[.*\1\3\6\1\4\1\29690\..*].nodata(60)}=0 )
```

- For **Zabbix 3.x**:

```
{TRIGGER.VALUE}=0 and {drweb-host:snmptrap["29690."].nodata(60)}=1 ) or  
{TRIGGER.VALUE}=1 and {drweb-host:snmptrap["29690."].nodata(60)}=0 )
```

An event is triggered (the value is set to 1) if the log of *SNMP trap* notifications from Dr.Web SNMPD was updated within a minute. If the log was not updated within the next minute, the value of the trigger is set to 0 again.

It is recommended to set in the **Severity** field for this trigger a notification type that is differ from *Not classified* value, for example, *Warning*.

Configuring Receipt of SNMP trap notifications for Zabbix

1. On the monitored host, in Dr.Web SNMPD settings (the **TrapReceiver** parameter), you should specify an address to be listened by **snmptrapd** on the host where **Zabbix** operates, for example:



```
SNMPD.TrapReceiver = 10.20.30.40:162
```

2. In the configuration file of **snmptrapd** (`snmptrapd.conf`), specify the same address and an application for processing received *SNMP trap* notifications (in this example, **snmptthandler**, **snmptt** component):

```
snmpTrapdAddr 10.20.30.40:162
traphandle default /usr/sbin/snmptthandler
```

Add the following string to the file, so that **snmptt** does not discard *SNMP trap* sent by Dr.Web SNMPD as unknown:

```
outputOption n
```

3. The **snmptthandler** component saves received *SNMP trap* notifications to the file on the disk in accordance with the specified format, which corresponds to the regular expression set in the host template for **Zabbix** (*drweb-traps* item). The format of the saved notification is specified in the `<opt_dir>/share/drweb-snmppd/connectors/zabbix/snmptt.drweb.zabbix.conf` file. The file must be copied to `/etc/snmp`.
4. Moreover, the path to the format files must be specified in the `snmptt.ini`:

```
[TrapFiles]
# A list of snmptt.conf files (this is NOT the snmptrapd.conf file).
# The COMPLETE path and filename. Ex: '/etc/snmp/snmptt.conf'
snmptt_conf_files = <<END
/etc/snmp/snmptt.conf
/etc/snmp/snmptt.drweb.zabbix.conf
END
```

After that, restart **snmptt** if it was started in daemon mode.

5. In the configuration file of the **Zabbix** server (`zabbix-server.conf`), specify (or check if they are already specified) the following settings:

```
SNMPTrapperFile=/var/log/snmptt/snmptt.log
StartSNMPTrapper=1
```

where `/var/log/snmptt/snmptt.log` is a log file used by **snmptt** to register information on received SNMP trap notifications.

For official documentation on **Zabbix**, refer to <https://www.zabbix.com/documentation/>.



Integration with Nagios Monitoring System

Files with configuration examples, required for establishing connection between Dr.Web SNMPD and the **Nagios** monitoring system, are located in the `<opt_dir>/share/drweb-snmpd/connectors/nagios` directory.

File	Description
<code>nagiosgraph/rrdopts.conf-sample</code>	Example of the RRD configuration file
<code>objects/drweb.cfg</code>	Configuration file describing <i>drweb</i> objects
<code>objects/nagiosgraph.cfg</code>	The configuration file of the component for graph plotting used by Nagiosgraph used by Nagios
<code>plugins/check_drweb</code>	The script for collecting data from the host on which Dr.Web is installed
<code>plugins/eventhandlers/submit_check_result</code>	The script for handling <i>SNMP trap</i> notifications
<code>snmp/snmptt.drweb.nagios.conf</code>	Configuring the snmptt utility—which is an <i>SNMP trap</i> handler

Connecting a host to Nagios

In the present instruction, it is assumed that the **Nagios** monitoring system is already deployed on the monitoring server, including configuration of the web server and the graphical tool **Nagiosgraph**, and the monitored host features an installed and functioning Dr.Web SNMPD (it is possible for the component to function in [proxy](#) mode together with **snmpd**). Moreover, if you want to receive *SNMP trap* notifications from the monitored host (including notification on threats detected by Dr.Web for UNIX File Servers on a protected server), install the `net-snmp` package on the monitoring server (standard tools **snmptt** and **snmptrapd** are used).

In the current manual, the following path conventions are used (real paths depend on the operating system and **Nagios** installation):

- `<NAGIOS_PLUGINS_DIR>`—directory with **Nagios** plug-ins, for example: `/usr/lib64/nagios/plugins`.
- `<NAGIOS_ETC_DIR>`—directory with **Nagios** settings, for example: `/etc/nagios`.
- `<NAGIOS_OBJECTS_DIR>`—directory with **Nagios** objects, for example: `/etc/nagios/objects`.
- `<NAGIOSGRAPH_DIR>`—**Nagiosgraph** directory, for example: `/usr/local/nagiosgraph`.
- `<NAGIOS_PERFDATA_LOG>`—file where **Nagios** records results of service check (must be the same as the `perflog` file from `<NAGIOSGRAPH_DIR>/etc/nagiosgraph.conf`). Records



from this file are read by the `<NAGIOSGRAPH_DIR>/bin/insert.pl` script and are recorded to the corresponding RRA archives **RRD Tool**.

Configuring Nagios:

1. Copy the `check_drweb` file to the `<NAGIOS_PLUGINS_DIR>` directory and the `drweb.cfg` file to the `<NAGIOS_OBJECTS_DIR>` directory.
2. Add hosts with Dr.Web that are to be monitored to the `drweb` group. On the hosts Dr.Web SNMPD must be running. By default, only `localhost` is added to this group.
3. If required, edit the `check_drweb` command which contains instruction to contact Dr.Web SNMPD on `drweb` hosts via the **snmplwalk** tool:

```
snmplwalk -c public -v 2c $HOSTADDRESS$:161
```

specify the correct version of SNMP protocol and parameters (such as "*community string*" or authentication parameters) as well as the port. The `$HOSTADDRESS$` variable must be included in the command (as this variable is later automatically substituted by **Nagios** to the correct host address when the command is invoked). OID is not required in the command. It is also recommended that you specify the command together with the full path to the executable file (usually `/usr/local/bin/snmpwalk`).

4. Connect *DrWeb* objects in the `<NAGIOS_ETC_DIR>/nagios.cfg` configuration file by adding the following string to the file:

```
cfg_file= <NAGIOS_OBJECTS_DIR>/drweb.cfg
```

5. Add **RRD Tool** settings for *DrWeb* graphics from the `rrdopts.conf-sample` file to the `<NAGIOSGRAPH_DIR>/etc/rrdopts.conf` file.
6. If **Nagiosgraph** is yet to be configured, do the following steps for its configuration:
 - Copy the `nagiosgraph.cfg` file to the `<NAGIOS_OBJECTS_DIR>` directory and edit the path to the `insert.pl` script in the **process-service-perfdata-for-nagiosgraph** command; for example, as follows:

```
$ awk ' $1 == "command_line" { $2 = "<NAGIOSGRAPH_DIR>/bin/insert.pl" }  
{ print }' ./objects/nagiosgraph.cfg > <NAGIOS_OBJECTS_DIR>/nagiosgraph.cfg
```

- Connect this file in the `<NAGIOS_ETC_DIR>/nagios.cfg` configuration file by adding the following line to it:

```
cfg_file=<NAGIOS_OBJECTS_DIR>/nagiosgraph.cfg
```

7. Check values of **Nagios** parameters in the `<NAGIOS_ETC_DIR>/nagios.cfg` configuration file:



```
check_external_commands=1
execute_host_checks=1
accept_passive_host_checks=1
enable_notifications=1
enable_event_handlers=1

process_performance_data=1
service_perfdata_file=/usr/nagiosgraph/var/rrd/perfdata.log
service_perfdata_file_template=$LASTSERVICECHECK$||$HOSTNAME$||$SERVICEDE
SC$||$SERVICEOUTPUT$||$SERVICEPERFDATA$
service_perfdata_file_mode=a
service_perfdata_file_processing_interval=30
service_perfdata_file_processing_command=process-service-perfdata-for-
nagiosgraph

check_service_freshness=1
enable_flap_detection=1
enable_embedded_perl=1
enable_environment_macros=1
```

Configuring Receipt of SNMP trap notifications for Nagios

1. On the monitored host in Dr.Web SNMPD settings (the **TrapReceiver** parameter), specify an address to be listened by **snmptrapd** on the host where **Nagios** operates, for example:

```
SNMPD.TrapReceiver = 10.20.30.40:162
```

2. Check for existing the `<NAGIOS_PLUGINS_DIR>/eventhandlers/submit_check_result` script which will be invoked when *SNMP trap* is received. If the script is missing, copy the `submit_check_result` file to this location from the `<opt_dir>/share/drweb-snmppd/connectors/nagios/plugins/eventhandlers/` directory. In this file, change the path specified in the `CommandFile` parameter. It must have the same value as the `command_file` parameter in the `<NAGIOS_ETC_DIR>/nagios.cfg` file.
3. Copy the `snmptt.drweb.nagios.conf` file to the `/etc/snmp/snmp/` directory. In this file, change the path to the `submit_check_result`—for example, by using the following command:

```
$ awk '$1 == "EXEC" { $2 =
<NAGIOS_PLUGINS_DIR>/eventhandlers/submit_check_result }{ print}'
./snmp/snmptt.drweb.nagios.conf > /etc/snmp/snmp/snmptt.drweb.nagios.conf
```

4. Add the `" /etc/snmp/snmptt.drweb.nagios.conf"` string to the `/etc/snmp/snmptt.drweb.nagios.conf` file. After that, restart **snmptt** if it was started in daemon mode.

After all required configuration files of **Nagios** are added and edited, run **Nagios** in debug mode by using the following command:

```
# nagios -v <NAGIOS_ETC_DIR>/nagios.cfg
```



Upon receipt of this command, **Nagios** will check for configuration errors. If no error is found, **Nagios** can be restarted as usual (for example, by using the **service** `nagios restart` command).

For official documentation on **Nagios**, refer to <http://www.nagios.org/documentation/>.

Dr.Web SNMP MIB

The list of operating parameters of Dr.Web for UNIX File Servers that can be fetched by external monitoring systems over the SNMP protocol is provided in the table below.

Parameter name	OID of the parameter	Type and description of the parameter
Common prefix for all names: <code>.iso.org.dod.internet.private.enterprises.drweb.drwebSnmpd</code> Common prefix for all OIDs: <code>.1.3.6.1.4.1.29690.2</code>		
alert	Asynchronous notifications about events (SNMP traps)	
<code>threatAlert</code>	<code>.1.1</code>	Notification about detecting a threat
<code>threatAlertFile</code>	<code>.1.1.1</code>	Name of the infected file (string)
<code>threatAlertType</code>	<code>.1.1.2</code>	Threat type (integer *)
<code>threatAlertName</code>	<code>.1.1.3</code>	Name of the threat (string)
<code>threatAlertOrigin</code>	<code>.1.1.4</code>	Identifier of the component that detected the threat (integer***)
<code>threatActionErrorAlert</code>	<code>.1.2</code>	Notification about an error occurred when trying to neutralize the threat
<code>threatActionErrorAlertFile</code>	<code>.1.2.1</code>	Name of the infected file (string)
<code>threatActionErrorAlertType</code>	<code>.1.2.2</code>	Threat type (integer *)
<code>threatActionErrorAlertName</code>	<code>.1.2.3</code>	Name of the threat (string)
<code>threatActionErrorAlertOrigin</code>	<code>.1.2.4</code>	Identifier of the component that detected the threat (integer***)
<code>threatActionErrorAlertError</code>	<code>.1.2.5</code>	Description of an error (string)
<code>threatActionErrorAlertErrorCode</code>	<code>.1.2.6</code>	Error code (integer corresponding to code from error catalogue)



Parameter name	OID of the parameter	Type and description of the parameter
<i>threatActionErrorAlertAction</i>	.1.2.7	Failed action (1—cure; 2—move to quarantine; 3—delete; 4—report; 5—ignore)
<i>componentFailureAlert</i>	.1.3	Notification about a component failure
<i>componentFailureAlertName</i>	.1.3.1	Component identifier (integer***)
<i>componentFailureAlertExitCodeDescription</i>	.1.3.2	Component exit code description (string)
<i>componentFailureAlertExitCode</i>	.1.3.3	Error code (integer corresponding to code from error catalogue)
<i>infectedUrlAlert</i>	.1.4	Notification about blocking a malicious URL (for HTTP/HTTPS connections)
<i>infectedUrlAlertUrl</i>	.1.4.1	The URL that was blocked (string)
<i>infectedUrlAlertDirection</i>	.1.4.2	HTTP message direction (integer: 1—request, 2—response)
<i>infectedUrlAlertType</i>	.1.4.3	Threat type (integer *)
<i>infectedUrlAlertName</i>	.1.4.4	Name of the threat (string)
<i>infectedUrlAlertOrigin</i>	.1.4.5	Identifier of the component that detected the threat (integer***)
<i>infectedUrlAlertSrcIp</i>	.1.4.6	IP address of connection source (string)
<i>infectedUrlAlertSrcPort</i>	.1.4.7	Port of connection source (integer)
<i>infectedUrlAlertDstIp</i>	.1.4.8	IP address of connection destination point (string)
<i>infectedUrlAlertDstPort</i>	.1.4.9	Port of connection destination point (integer)
<i>infectedUrlAlertSniHost</i>	.1.4.10	SNI of connection destination point (for SSL connections) (string)



Parameter name	OID of the parameter	Type and description of the parameter
<i>infectedUrlAlertExePath</i>	.1.4.11	Executable path of the program that establish the connection (string)
<i>infectedUrlAlertUserName</i>	.1.4.12	Name of the user with whose privileges is executing the program that establish the connection (string)
<i>infectedAttachmentAlert</i>	.1.5	Notification about detecting an infected email attachment
<i>infectedAttachmentAlertType</i>	.1.5.1	Threat type (integer *)
<i>infectedAttachmentAlertName</i>	.1.5.2	Name of the threat (string)
<i>infectedAttachmentAlertOrigin</i>	.1.5.3	Identifier of the component that detected the threat (integer***)
<i>infectedEmailAttachmentAlertSocket</i>	.1.5.4	IP address of the source of the email message (string)
<i>infectedEmailAttachmentAlertMailFrom</i>	.1.5.5	Sender of the email message (string)
<i>infectedEmailAttachmentAlertRcptTo</i>	.1.5.6	Recipients of the email message (string)
<i>infectedEmailAttachmentAlertMessageId</i>	.1.5.7	Value of <code>Message-ID</code> header of the email message (string)
<i>infectedEmailAttachmentAlertAction</i>	.1.5.8	Action that was applied to the whole email message or infected attachment (integer: 1—repack; 2—reject; 3—discard; 4—cure; 5—move to quarantine; 6—delete)
<i>infectedEmailAttachmentAlertDivert</i>	.1.5.9	Direction of the email message (integer: 1—incoming; 2—outgoing)
<i>infectedEmailAttachmentAlertSrcIp</i>	.1.5.10	IP address of connection source (string)
<i>infectedEmailAttachmentAlertSrcPort</i>	.1.5.11	Port of connection source (integer)



Parameter name	OID of the parameter	Type and description of the parameter
<i>infectedEmailAttachmentAlertDstIp</i>	.1.5.12	IP address of connection destination point (string)
<i>infectedEmailAttachmentAlertDstPort</i>	.1.5.13	Port of connection destination point (integer)
<i>infectedEmailAttachmentAlertSniHost</i>	.1.5.14	SNI of connection destination point (for SSL connections) (string)
<i>infectedEmailAttachmentAlertProtocol</i>	.1.5.15	Protocol type (integer: 1—SMTP; 2—POP3; 3—IMAP; 4—HTTP)
<i>infectedEmailAttachmentAlertExePath</i>	.1.5.16	Executable path of the program that establish the connection (string)
<i>infectedEmailAttachmentAlertUserName</i>	.1.5.17	Name of the user with whose privileges is executing the program that establish the connection (string)
<i>categoryUrlAlert</i>	.1.6	Notification about blocking a URL belonging to the unwanted category
<i>categoryUrlAlertUrl</i>	.1.6.1	The URL that was blocked (string)
<i>categoryUrlAlertCategory</i>	.1.6.2	Web resource category to which the URL belongs (integer**)
<i>categoryUrlAlertOrigin</i>	.1.6.3	Identifier of the component that detected the threat (integer***)
<i>categoryUrlAlertSrcIp</i>	.1.6.4	IP address of connection source (string)
<i>categoryUrlAlertSrcPort</i>	.1.6.5	Port of connection source (integer)
<i>categoryUrlAlertDstIp</i>	.1.6.6	IP address of connection destination point (string)
<i>categoryUrlAlertDstPort</i>	.1.6.7	Port of connection destination point (integer)
<i>categoryUrlAlertSniHost</i>	.1.6.8	SNI of connection destination point (for SSL connections)



Parameter name	OID of the parameter	Type and description of the parameter
		(string)
<i>categoryUrlAlertExePath</i>	.1.6.9	Executable path of the program that establish the connection (string)
<i>categoryUrlAlertUserName</i>	.1.6.10	Name of the user with whose privileges is executing the program that establish the connection (string)
<i>categoryUrlEmailAttachmentAlert</i>	.1.7	Notification about detecting an unwanted URL in the email message
<i>categoryUrlEmailAttachmentAlertType</i>	.1.7.1	Web resource category to which the URL belongs (integer**)
<i>categoryUrlEmailAttachmentAlertOrigin</i>	.1.7.2	Identifier of the component that detected the threat (integer***)
<i>categoryUrlEmailAttachmentAlertSocket</i>	.1.7.3	IP address of the source of the email message (string)
<i>categoryUrlEmailAttachmentAlertMailFrom</i>	.1.7.4	Sender of the email message (string)
<i>categoryUrlEmailAttachmentAlertRcptTo</i>	.1.7.5	Recipients of the email message (string)
<i>categoryUrlEmailAttachmentAlertMessageId</i>	.1.7.6	Value of <code>Message-ID</code> header of the email message (string)
<i>categoryUrlEmailAttachmentAlertAction</i>	.1.7.7	Action that was applied to the whole email message or an attachment (integer: 1—repack; 2—reject; 3—discard; 4—cure; 5—move to quarantine; 6—delete)
<i>categoryUrlEmailAttachmentAlertDivert</i>	.1.7.8	Direction of the email message (integer: 1—incoming; 2—outgoing)
<i>categoryUrlEmailAttachmentAlertSrcIp</i>	.1.7.9	IP address of connection source (string)
<i>categoryUrlEmailAttachmentAlertSrcPort</i>	.1.7.10	Port of connection source (integer)



Parameter name	OID of the parameter	Type and description of the parameter
<i>categoryUrlEmailAttachmentAlertDstIp</i>	.1.7.11	IP address of connection destination point (string)
<i>categoryUrlEmailAttachmentAlertDstPort</i>	.1.7.12	Port of connection destination point (integer)
<i>categoryUrlEmailAttachmentAlertSniHost</i>	.1.7.13	SNI of connection destination point (for SSL connections) (string)
<i>categoryUrlEmailAttachmentAlertProtocol</i>	.1.7.14	Protocol type (integer: 1—SMTP; 2—POP3; 3—IMAP; 4—HTTP)
<i>categoryUrlEmailAttachmentAlertExePath</i>	.1.7.15	Executable path of the program that establish the connection (string)
<i>categoryUrlEmailAttachmentAlertUserName</i>	.1.7.16	Name of the user with whose privileges is executing the program that establish the connection (string)
<i>spamEmailAlert</i>	.1.8	Notification about recognizing an email message as spam
<i>spamEmailAlertOrigin</i>	.1.8.1	Identifier of the component that detected the threat (integer***)
<i>spamEmailAlertSocket</i>	.1.8.2	IP address of the source of the email message (string)
<i>spamEmailAlertMailFrom</i>	.1.8.3	Sender of the email message (string)
<i>spamEmailAlertRcptTo</i>	.1.8.4	Recipients of the email message (string)
<i>spamEmailAlertMessageId</i>	.1.8.5	Value of <code>Message-ID</code> header of the email message (string)
<i>spamEmailAlertAction</i>	.1.8.6	Action that was applied to the whole email message or an attachment (integer: 1—repack; 2—reject; 3—discard; 4—cure; 5—move to quarantine; 6—delete)
<i>spamEmailAlertDivert</i>	.1.8.7	Direction of the email message (integer: 1—incoming; 2—



Parameter name	OID of the parameter	Type and description of the parameter
		outgoing)
<i>spamEmailAlertSrcIp</i>	.1.8.8	IP address of connection source (string)
<i>spamEmailAlertSrcPort</i>	.1.8.9	Port of connection source (integer)
<i>spamEmailAlertDstIp</i>	.1.8.10	IP address of connection destination point (string)
<i>spamEmailAlertDstPort</i>	.1.8.11	Port of connection destination point (integer)
<i>spamEmailAlertSniHost</i>	.1.8.12	SNI of connection destination point (for SSL connections) (string)
<i>spamEmailAlertProtocol</i>	.1.8.13	Protocol type (integer: 1—SMTP; 2—POP3; 3—IMAP; 4—HTTP)
<i>spamEmailAlertExePath</i>	.1.8.14	Executable path of the program that establish the connection (string)
<i>spamEmailAlertUserName</i>	.1.8.15	Name of the user with whose privileges is executing the program that establish the connection (string)
<i>blockedConnectionAlert</i>	.1.9	Notification about blocking a network connection
<i>blockedConnectionAlertOrigin</i>	.1.9.1	Identifier of the component that detected the threat (integer***)
<i>blockedConnectionAlertDivert</i>	.1.9.2	Direction of the connection (integer: 1—incoming; 2—outgoing)
<i>blockedConnectionAlertSrcIp</i>	.1.9.3	IP address of connection source (string)
<i>blockedConnectionAlertSrcPort</i>	.1.9.4	Port of connection source (integer)
<i>blockedConnectionAlertDstIp</i>	.1.9.5	IP address of connection destination point (string)



Parameter name	OID of the parameter	Type and description of the parameter
<i>blockedConnectionAlertDstPort</i>	.1.9.6	Port of connection destination point (integer)
<i>blockedConnectionAlertSniHost</i>	.1.9.7	SNI of connection destination point (for SSL connections) (string)
<i>blockedConnectionAlertProtocol</i>	.1.9.8	Protocol type (integer: 1—SMTP; 2—POP3; 3—IMAP; 4—HTTP)
<i>blockedConnectionAlertExePath</i>	.1.9.9	Executable path of the program that establish the connection (string)
<i>blockedConnectionAlertUserName</i>	.1.9.10	Name of the user with whose privileges is executing the program that establish the connection (string)
stat	Statistics on the operation of the software product	
<i>threatCounters</i>	.2.1	Counters of detected threats
<i>knownVirus</i>	.2.1.1	Number of detected known viruses (counter; integer)
<i>suspicious</i>	.2.1.2	Number of detected suspicious objects (counter; integer)
<i>adware</i>	.2.1.3	Number of detected adware (counter; integer)
<i>dialers</i>	.2.1.4	Number of detected dialers (counter; integer)
<i>joke</i>	.2.1.5	Number of detected joke programs (counter; integer)
<i>riskware</i>	.2.1.6	Number of detected riskware (counter; integer)
<i>hacktool</i>	.2.1.7	Number of detected hacktools (counter; integer)
<i>scanErrors</i>	.2.2	Counters of the errors that occurred while files were scanned



Parameter name	OID of the parameter	Type and description of the parameter
<i>sePathNotAbsolute</i>	.2.2.1	Number of occurrences of the "Path is not absolute" error (counter, integer)
<i>seFileNotFound</i>	.2.2.2	Number of occurrences of the "File not found" error (counter, integer)
<i>seFileNotRegular</i>	.2.2.3	Number of occurrences of the "File is not a regular file" error (counter, integer)
<i>seFileNotBlockDevice</i>	.2.2.4	Number of occurrences of the "File is not a block device" error (counter, integer)
<i>seNameTooLong</i>	.2.2.5	Number of occurrences of the "Path or file name is too long" error (counter, integer)
<i>seNoAccess</i>	.2.2.6	Number of occurrences of the "Permission denied" error (counter, integer)
<i>seReadError</i>	.2.2.7	Number of occurrences of the "Read error" (counter, integer)
<i>seWriteError</i>	.2.2.8	Number of occurrences of the "Write error" (counter, integer)
<i>seFileTooLarge</i>	.2.2.9	Number of occurrences of the "File size too big" error (counter, integer)
<i>seFileBusy</i>	.2.2.10	Number of occurrences of the "File is busy" error (counter, integer)
<i>seUnpackingError</i>	.2.2.20	Number of occurrences of the "Unpacking error" (counter, integer)
<i>sePasswordProtectd</i>	.2.2.21	Number of occurrences of the "Password protected" error (counter, integer)
<i>seArchCrcError</i>	.2.2.22	Number of occurrences of the "Archive Cyclic Redundancy



Parameter name	OID of the parameter	Type and description of the parameter
		Check error" (counter, integer)
<i>seArchInvalidHeader</i>	.2.2.23	Number of occurrences of the "Invalid archive header" error (counter, integer)
<i>seArchNoMemory</i>	.2.2.24	Number of occurrences of the "Not enough memory to process the archive" error (counter, integer)
<i>seArchIncomplete</i>	.2.2.25	Number of occurrences of the "Incomplete archive" error (counter, integer)
<i>seCanNotBeCured</i>	.2.2.26	Number of occurrences of the "Object cannot be cured" error (counter, integer)
<i>sePackerLevelLimit</i>	.2.2.30	Number of occurrences of the error that states that the maximum nesting level of packed objects was exceeded (counter, integer)
<i>seArchiveLevelLimit</i>	.2.2.31	Number of occurrences of the error that states that the maximum nesting level of archives was exceeded (counter, integer)
<i>seMailLevelLimit</i>	.2.2.32	Number of occurrences of the error that states that the maximum nesting level of email files was exceeded (counter, integer)
<i>seContainerLevelLimit</i>	.2.2.33	Number of occurrences of the error that states that the maximum nesting level of container files was exceeded (counter, integer)
<i>seCompressionLimit</i>	.2.2.34	Number of occurrences of the "Exceeded the maximum compression ratio" error (counter, integer)



Parameter name	OID of the parameter	Type and description of the parameter
<i>seReportSizeLimit</i>	.2.2.35	Number of occurrences of the "Exceeded the maximum size of the scanning results report" error (counter, integer)
<i>seScanTimeout</i>	.2.2.40	Number of occurrences of the "Scan timeout expired" error (counter, integer)
<i>seEngineCrash</i>	.2.2.41	Number of occurrences of the "Scanning Engine crash was detected" error (counter, integer)
<i>seEngineHangup</i>	.2.2.42	Number of occurrences of the "Scanning Engine stopped responding" error (counter, integer)
<i>seEngineError</i>	.2.2.44	Number of occurrences of the "Internal error of the Scanning Engine" (counter, integer)
<i>seNoLicense</i>	.2.2.45	Number of occurrences of the "No valid license found" error (counter, integer)
<i>seNonSupportedDisk</i>	.2.2.50	Number of Occurrences of the "Unsupported disk" error (counter, integer)
<i>seUnexpectedError</i>	.2.2.100	Number of occurrences of the "Unexpected error" (counter, integer)
<i>scanLoadAverage</i>	.2.3	Metrics of the file scanning load
<i>filesScannedTable</i>	.2.3.1	Average numbers of files scanned at the request of other components
<i>filesScannedEntry</i>	.2.3.1.1	Component of the product (entire table row, record)
<i>filesScannedIndex</i>	.2.3.1.1.1	Index of the component (identifier, integer***)
<i>filesScannedOrigin</i>	.2.3.1.1.2	Name of the component



Parameter name	OID of the parameter	Type and description of the parameter
filesScanned1min	.2.3.1.1.3	The average (averaged over one minute) number of files checked per second (string)
filesScanned5min	.2.3.1.1.4	The average (averaged over 5 minutes) number of files checked per second (string)
filesScanned15min	.2.3.1.1.5	The average (averaged over 15 minutes) number of files checked per second (string)
<i>bytesScannedTable</i>	.2.3.2	Average speed (in bytes per second) of scanning performed at the request of other components
bytesScannedEntry	.2.3.2.1	Component of the product (entire table row, record)
bytesScannedIndex	.2.3.2.1.1	Index of the component (identifier, integer***)
bytesScannedOrigin	.2.3.2.1.2	Name of the component
bytesScanned1min	.2.3.2.1.3	The average (averaged over one minute) number of bytes scanned per second (string)
bytesScanned5min	.2.3.2.1.4	The average (averaged over 5 minutes) number of bytes scanned per second (string)
bytesScanned15min	.2.3.2.1.5	The average (averaged over 15 minutes) number of bytes scanned per second (string)
<i>cacheHitFilesTable</i>	.2.3.3	Average numbers of scanning reports retrieved from the cache at the request of the components
cacheHitFilesEntry	.2.3.3.1	Component of the product (entire table row, record)
cacheHitFilesIndex	.2.3.3.1.1	Index of the component (identifier, integer***)
cacheHitFilesOrigin	.2.3.3.1.2	Name of the component



Parameter name	OID of the parameter	Type and description of the parameter
cacheHitFiles1min	.2.3.3.1.3	The average (averaged over one minute) number of reports retrieved from the cache per second (string)
cacheHitFiles5min	.2.3.3.1.4	The average (averaged over 5 minutes) number of reports retrieved from the cache per second (string)
cacheHitFiles15min	.2.3.3.1.5	The average (averaged over 15 minutes) number of reports retrieved from the cache per second (string)
<i>errorsTable</i>	.2.3.4	Average numbers of errors during the scanning that was performed at the request of the components
errorsEntry	.2.3.4.1	Component of the product (entire table row, record)
errorsIndex	.2.3.4.1.1	Index of the component (identifier, integer***)
errorsOrigin	.2.3.4.1.2	Name of the component
errors1min	.2.3.4.1.3	The average (averaged over one minute) number of scanning errors per second (string)
errors5min	.2.3.4.1.4	The average (averaged over 5 minutes) number of scanning errors per second (string)
errors15min	.2.3.4.1.5	The average (averaged over 15 minutes) number of scanning errors per second (string)
net	.2.4	Statistics on network activity
<i>markedAsSpam</i>	.2.4.1	Number of email messages marked as spam (counter, integer)
<i>blockedInfectionSource</i>	.2.4.101	Number of blocked URLs belonging to the "Infection"



Parameter name	OID of the parameter	Type and description of the parameter
		Source" category (counter, integer)
<i>blockedNotRecommended</i>	.2.4.102	Number of blocked URLs belonging to the "Not Recommended" category (counter, integer)
<i>blockedAdultContent</i>	.2.4.103	Number of blocked URLs belonging to the "Adult Content" category (counter, integer)
<i>blockedViolence</i>	.2.4.104	Number of blocked URLs belonging to the "Violence" category (counter, integer)
<i>blockedWeapons</i>	.2.4.105	Number of blocked URLs belonging to the "Weapons" category (counter, integer)
<i>blockedGambling</i>	.2.4.106	Number of blocked URLs belonging to the "Gambling" category (counter, integer)
<i>blockedDrugs</i>	.2.4.107	Number of blocked URLs belonging to the "Drugs" category (counter, integer)
<i>blockedObsceneLanguage</i>	.2.4.108	Number of blocked URLs belonging to the "Obscene Language" category (counter, integer)
<i>blockedChats</i>	.2.4.109	Number of blocked URLs belonging to the "Chats" category (counter, integer)
<i>blockedTerrorism</i>	.2.4.110	Number of blocked URLs belonging to the "Terrorism" category (counter, integer)
<i>blockedFreeEmail</i>	.2.4.111	Number of blocked URLs belonging to the "Free Email Services" category (counter, integer)
<i>blockedSocialNetworks</i>	.2.4.112	Number of blocked URLs belonging to the "Social



Parameter name	OID of the parameter	Type and description of the parameter
		Networks" category (counter, integer)
<i>blockedOwnersNotice</i>	.2.4.113	Number of blocked URLs belonging to the "Copyright Owner's Notice" category (counter, integer)
<i>blockedBlackList</i>	.2.4.120	Number of blocked URLs from the user's black list (counter, integer)
info	Information about the current state of the program	
components	.3.1	Current state of the program's components
<i>configd</i>	.3.1.1	drweb-configd component data
configdState	.3.1.1.1	Current state of the component (integer****)
configdExitCode	.3.1.1.2	Last exit code (integer corresponding to code from error catalogue)
configdExitTime	.3.1.1.3	Time of the last termination (<i>UNIX time</i>)
configdInstalledApps	.3.1.1.101	List of installed components
configdAppEntry	.3.1.1.101.1	Information about the installed component (entire table row, record)
configdAppIndex	.3.1.1.101.1.1	Index (ordinal number) of the installed component (integer)
configdAppName	.3.1.1.101.1.2	Name of the installed component (string)
configdAppExePath	.3.1.1.101.1.3	Path to the executable file of the component (string)
configdAppInstallTime	.3.1.1.101.1.4	Time when the component was installed (<i>UNIX time</i>)



Parameter name	OID of the parameter	Type and description of the parameter
configdAppIniSection	.3.1.1.101.1.5	Name of the section with the component's parameters in the configuration file
<i>scanEngine</i>	.3.1.2	drweb-se component data
scanEngineState	.3.1.2.1	Current state of the component (integer****)
scanEngineExitCode	.3.1.2.2	Last exit code (integer corresponding to code from error catalogue)
scanEngineExitTime	.3.1.2.3	Time of the last termination (UNIX time)
scanEngineStatus	.3.1.2.101	Current state of the Dr.Web Virus-Finding Engine (integer)
scanEngineVersion	.3.1.2.102	Version of the Dr.Web Virus-Finding Engine (string)
scanEngineVirusRecords	.3.1.2.103	Number of virus records (integer)
scanEngineMaxForks	.3.1.2.104	Maximum number of child processes for scanning (integer)
scanEngineQueues	.3.1.2.105	Scan task queues
scanEngineQueuesLow	.3.1.2.105.1	The queue of low-priority tasks
scanEngineQueueLowOut	.3.1.2.105.1.1	Number of low-priority tasks popped from the queue and transferred to processing (counter, integer)
scanEngineQueueLowSize	.3.1.2.105.1.2	Number of low-priority tasks in the queue waiting to be processed (counter, integer)
scanEngineQueuesMedium	.3.1.2.105.2	The queue of normal-priority tasks
scanEngineQueueMediumOut	.3.1.2.105.2.1	The number of normal-priority tasks popped from the queue and transferred to processing (counter, integer)



Parameter name	OID of the parameter	Type and description of the parameter
scanEngineQueueMediumSize	.3.1.2.105.2.2	Number of normal-priority tasks in the queue waiting to be processed (counter, integer)
scanEngineQueuesHigh	.3.1.2.105.3	The queue of high-priority tasks
scanEngineQueueHighOut	.3.1.2.105.3.1	The number of high-priority tasks popped from the queue and transferred to processing (counter, integer)
scanEngineQueueHighSize	.3.1.2.105.3.2	Number of high-priority tasks in the queue waiting to be processed (counter, integer)
scanEngineVirusBasesTable	.3.1.2.106	The list of virus databases.
scanEngineVirusBasesEntry	.3.1.2.106.1	Information about the virus database (entire table row; record)
scanEngineVirusBaseIndex	.3.1.2.106.1.1	Index of the virus database (integer)
scanEngineVirusBasePath	.3.1.2.106.1.2	Path to the virus database file (string)
scanEngineVirusBaseRecords	.3.1.2.106.1.3	Number of records in the virus database (integer)
scanEngineVirusBaseVersion	.3.1.2.106.1.4	Version of the virus database (integer)
scanEngineVirusBaseTimestamp	.3.1.2.106.1.5	Timestamp of the virus database (<i>UNIX time</i>)
scanEngineVirusBaseMD5	.3.1.2.106.1.6	MD5 checksum (string)
scanEngineVirusBaseLoadResult	.3.1.2.106.1.7	Result of the downloading of this virus database (string)
scanEngineQueuesTab	.3.1.2.107	The list of scan task queues
scanEngineQueueEntry	.3.1.2.107.1	Information about the queue (entire table row, record)
scanEngineQueueIndex	.3.1.2.107.1.1	Index (ordinal number) of the queue (integer)



Parameter name	OID of the parameter	Type and description of the parameter
scanEngineQueueName	.3.1.2.107.1.2	Name of the queue (string)
scanEngineQueueOut	.3.1.2.107.1.3	The number of tasks popped from the queue and transferred to processing (counter, integer)
scanEngineQueueSize	.3.1.2.107.1.4	Number of tasks in the queue waiting to be processed (counter, integer)
<i>fileCheck</i>	.3.1.3	drweb-filecheck component data
fileCheckState	.3.1.3.1	Current state of the component (integer****)
fileCheckExitCode	.3.1.3.2	Last exit code (integer corresponding to code from error catalogue)
fileCheckExitTime	.3.1.3.3	Time of the last termination (<i>UNIX time</i>)
fileCheckScannedFiles	.3.1.3.101	Number of scanned files (counter, integer)
fileCheckScannedBytes	.3.1.3.102	Number of scanned bytes (counter, integer)
fileCheckCacheHitFiles	.3.1.3.103	Number of scan reports retrieved from the cache (counter, integer)
fileCheckScanErrors	.3.1.3.104	Number of error occurrences in the Scanning Engine (counter, integer)
fileCheckScanStat	.3.1.3.105	List of clients
fileCheckClientEntry	.3.1.3.105.1	Information about the client (entire table row; record)
fileCheckClientIndex	.3.1.3.105.1.1	Index (ordinal number) of the client (integer)
fileCheckClientName	.3.1.3.105.1.2	Name of the client component (string)



Parameter name	OID of the parameter	Type and description of the parameter
fileCheckClientScannedFiles	.3.1.3.105.1.3	The number of files scanned for this client (counter, integer)
fileCheckClientScannedBytes	.3.1.3.105.1.4	The number of bytes scanned for this client (counter, integer)
fileCheckClientCacheHitFiles	.3.1.3.105.1.5	The number of scan reports retrieved from the cache for this client (counter, integer)
fileCheckClientScanErrors	.3.1.3.105.1.6	Number of error occurrences in the Scanning Engine when working for this client (counter, integer)
<i>update</i>	.3.1.4	drweb-update component data
updateState	.3.1.4.1	Current state of the component (integer****)
updateExitCode	.3.1.4.2	Last exit code (integer corresponding to code from error catalogue)
updateExitTime	.3.1.4.3	Time of the last termination (<i>UNIX time</i>)
updateBytesSent	.3.1.4.101	Number of bytes sent (counter, integer)
updateBytesReceived	.3.1.4.102	Number of bytes received (counter, integer)
<i>esagent</i>	.3.1.5	drweb-esagent component data
esagentState	.3.1.5.1	Current state of the component (integer****)
esagentExitCode	.3.1.5.2	Last exit code (integer corresponding to code from error catalogue)
esagentExitTime	.3.1.5.3	Time of the last termination (<i>UNIX time</i>)
esagentWorkStatus	.3.1.5.101	Component's current mode of operation (integer: 1—standalone mode, 2—is connecting, 3—is



Parameter name	OID of the parameter	Type and description of the parameter
		awaiting connection, 4—connection has been approved)
esagentIsConnected	.3.1.5.102	Is connected to the server (integer, 0—no, 1—yes)
esagentServer	.3.1.5.103	Address of the central protection server that is used (string)
<i>netcheck</i>	.3.1.6	drweb-netcheck component data
netcheckState	.3.1.6.1	Current state of the component (integer****)
netcheckExitCode	.3.1.6.2	Last exit code (integer corresponding to code from error catalogue)
netcheckExitTime	.3.1.6.3	Time of the last termination (<i>UNIX time</i>)
netcheckLocalSeForks	.3.1.6.101	The number of Scanning Engine processes available locally (integer)
netcheckRemoteSeForks	.3.1.6.102	Number of Scanning Engine processes available remotely (integer)
netcheckLocalFilesScanned	.3.1.6.103	The number of files that have been scanned locally (counter, integer)
netcheckNetworkFilesScanned	.3.1.6.104	The number of files that have been scanned via remote scanning (counter, integer)
netcheckLocalBytesScanned	.3.1.6.105	The number of bytes that have been scanned locally (counter, integer)
netcheckNetworkBytesScanned	.3.1.6.106	The number of bytes that have been scanned via remote scanning (counter, integer)
netcheckLocalBytesIn	.3.1.6.107	The number of bytes received from local clients (counter, integer)



Parameter name	OID of the parameter	Type and description of the parameter
netcheckLocalBytesOut	.3.1.6.108	The number of bytes sent back to local clients (counter, integer)
netcheckNetworkBytesIn	.3.1.6.109	The number of bytes received from remote hosts (counter, integer)
netcheckNetworkBytesOut	.3.1.6.110	The number of bytes sent to remote hosts (counter, integer)
netcheckLocalScanErrors	.3.1.6.111	Number of error occurrences in local Scanning Engine processes (counter, integer)
netcheckNetworkScanErrors	.3.1.6.112	Number of error occurrences in remote Scanning Engine processes (counter, integer)
<i>httpd</i>	.3.1.7	drweb-httpd component data
httpdState	.3.1.7.1	Current state of the component (integer****)
httpdExitCode	.3.1.7.2	Last exit code (integer corresponding to code from error catalogue)
httpdExitTime	.3.1.7.3	Time of the last termination (<i>UNIX time</i>)
<i>snmpd</i>	.3.1.8	drweb-snmpd component data
snmpdState	.3.1.8.1	Current state of the component (integer****)
snmpdExitCode	.3.1.8.2	Last exit code (integer corresponding to code from error catalogue)
snmpdExitTime	.3.1.8.3	Time of the last termination (<i>UNIX time</i>)
<i>clamd</i>	.3.1.20	drweb-clamd component data
clamdState	.3.1.20.1	Current state of the component (integer****)



Parameter name	OID of the parameter	Type and description of the parameter
clamdExitCode	.3.1.20.2	Last exit code (integer corresponding to code from error catalogue)
clamdExitTime	.3.1.20.3	Time of the last termination (<i>UNIX time</i>)
<i>icapd</i>	.3.1.21	drweb-icapd component data
icapdState	.3.1.21.1	Current state of the component (integer****)
icapdExitCode	.3.1.21.2	Last exit code (integer corresponding to code from error catalogue)
icapdExitTime	.3.1.21.3	Time of the last termination (<i>UNIX time</i>)
icapdConnectionsIn	.3.1.21.101	Number of accepted incoming connections (counter, integer)
icapdConnectionsCount	.3.1.21.102	Number of currently opened connections (counter, integer)
icapdOptions	.3.1.21.103	Number of <i>OPTIONS</i> requests (counter, integer)
icapdReqmod	.3.1.21.104	Number of <i>REQMOD</i> requests (counter, integer)
icapdRespmod	.3.1.21.105	Number of <i>RESPMOD</i> requests (counter, integer)
icapdBad	.3.1.21.106	Number of invalid requests (counter, integer)
<i>smb spider</i>	.3.1.40	drweb-smb spider-daemon component data
smb spiderState	.3.1.40.1	Current state of the component (integer****)
smb spiderExitCode	.3.1.40.2	Last exit code (integer corresponding to code from error catalogue)



Parameter name	OID of the parameter	Type and description of the parameter
smbspiderExitTime	.3.1.40.3	Time of the last termination (<i>UNIX time</i>)
smbspiderConnectionsIn	.3.1.40.101	Total number of opened connections (counter, integer)
smbspiderConnectionsCount	.3.1.40.102	Number of currently opened connections (counter, integer)
smbspiderShareTable	.3.1.40.103	Statistics on the protected Samba shared resources
smbspiderShareEntry	.3.1.40.103.1	Information about the protected Samba shared resource (entire table row; record)
smbspiderShareIndex	.3.1.40.103.1.1	Index (ordinal number) of the protected Samba shared resource (integer)
smbspiderSharePath	.3.1.40.103.1.2	Path to the protected Samba shared resource (string)
smbspiderShareConnectionsIn	.3.1.40.103.1.3	Total number of opened connections (counter, integer)
smbspiderShareConnectionsCount	.3.1.40.103.1.4	Number of currently opened connections (counter, integer)
<i>gated</i>	.3.1.41	drweb-gated component data
gatedState	.3.1.41.1	Current state of the component (integer****)
gatedExitCode	.3.1.41.2	Last exit code (integer corresponding to code from error catalogue)
gatedExitTime	.3.1.41.3	Time of the last termination (<i>UNIX time</i>)
gatedInterceptedIn	.3.1.41.101	Number of intercepted connections (counter, integer)
gatedInterceptedCount	.3.1.41.102	Number of currently monitored connections (counter, integer)
<i>maild</i>	.3.1.42	drweb-maild component data



Parameter name	OID of the parameter	Type and description of the parameter
maildState	.3.1.42.1	Current state of the component (integer****)
maildExitCode	.3.1.42.2	Last exit code (integer corresponding to code from error catalogue)
maildExitTime	.3.1.42.3	Time of the last termination (<i>UNIX time</i>)
maildStat	.3.1.42.4	Statistics of the drweb-maild component operation
maildStatNative	.3.1.42.4.1	Email scanning statistics via the component's internal interface drweb-maild (messages received by SpIDer Gate during the scan of intercepted SMTP, POP3, IMAP connections)
maildStatNativePassed	.3.1.42.4.1.1	Number of missed messages (counter, integer)
maildStatNativeRepacked	.3.1.42.4.1.2	Number of repackaged messages (counter, integer)
maildStatNativeRejected	.3.1.42.4.1.3	Number of rejected messages (counter, integer)
maildStatNativeFailed	.3.1.42.4.1.4	Number of message scanning errors (counter, integer)
maildStatNativeQueueSize	.3.1.42.4.1.5	The queue line, that is the number of files waiting to be scanned via the interface (integer)
maildStatMilter	.3.1.42.4.2	Email scanning statistics via the component's interface <i>Milter</i> of the drweb-maild component
maildStatMilterPassed	.3.1.42.4.2.1	Number of missed messages (counter, integer)
maildStatMilterRepacked	.3.1.42.4.2.2	Number of repackaged messages (counter, integer)
maildStatMilterRejected	.3.1.42.4.2.3	Number of rejected messages (counter, integer)



Parameter name	OID of the parameter	Type and description of the parameter
maildStatMilterFailed	.3.1.42.4.2.4	Number of message scanning errors (counter, integer)
maildStatMilterQueueSize	.3.1.42.4.2.5	The queue line, that is the number of files waiting to be scanned via the interface (integer)
maildStatSpamc	.3.1.42.4.3	Email scanning statistics via the component's interface <i>Spamd</i> of the drweb-maild component
maildStatSpamcPassed	.3.1.42.4.3.1	Number of missed messages (counter, integer)
maildStatSpamcRepacked	.3.1.42.4.3.2	Number of repackaged messages (counter, integer)
maildStatSpamcRejected	.3.1.42.4.3.3	Number of rejected messages (counter, integer)
maildStatSpamcFailed	.3.1.42.4.3.4	Number of message scanning errors (counter, integer)
maildStatSpamcQueueSize	.3.1.42.4.3.5	The queue line, that is the number of files waiting to be scanned via the interface (integer)
maildStatRspamc	.3.1.42.4.4	Email scanning statistics via the component's interface <i>Rspamd</i> of the drweb-maild component
maildStatRspamcPassed	.3.1.42.4.4.1	Number of missed messages (counter, integer)
maildStatRspamcRepacked	.3.1.42.4.4.2	Number of repackaged messages (counter, integer)
maildStatRspamcRejected	.3.1.42.4.4.3	Number of rejected messages (counter, integer)
maildStatRspamcFailed	.3.1.42.4.4.4	Number of message scanning errors (counter, integer)
maildStatRspamcQueueSize	.3.1.42.4.4.5	The queue line, that is the number of files waiting to be scanned via the interface (integer)
<i>lookupd</i>	.3.1.43	drweb-lookupd component data



Parameter name	OID of the parameter	Type and description of the parameter
lookupdState	.3.1.43.1	Current state of the component (integer****)
lookupdExitCode	.3.1.43.2	Last exit code (integer corresponding to code from error catalogue)
lookupdExitTime	.3.1.43.3	Time of the last termination (UNIX time)
<i>cloudd</i>	.3.1.50	drweb-cloudd component data
clouddState	.3.1.50.1	Current state of the component (integer****)
clouddExitCode	.3.1.50.2	Last exit code (integer corresponding to code from error catalogue)
clouddExitTime	.3.1.50.3	Time of the last termination (UNIX time)
<i>vpnd</i>	.3.1.51	drweb-vpnd component data
vpndState	.3.1.51.1	Current state of the component (integer****)
vpndExitCode	.3.1.51.2	Last exit code (integer corresponding to code from error catalogue)
vpndExitTime	.3.1.51.3	Time of the last termination (UNIX time)
vpndWorkStatus	.3.1.51.101	Component's current mode of operation (integer: 0—turned off, 1—server, 2—client)
vpndConnectionState	.3.1.51.102	Status of the established connection (integer: 0—status not set, 1—connecting, 2—connected, 3—error, 4—setting up NAT, 5—creating a protected tunnel)
vpndNetworkName	.3.1.51.103	Name of the created personal network (string)



Parameter name	OID of the parameter	Type and description of the parameter
<i>meshd</i>	.3.1.52	drweb-meshd component data
meshdState	.3.1.52.1	Current state of the component (integer****)
meshdExitCode	.3.1.52.2	Last exit code (integer corresponding to code from error catalogue)
meshdExitTime	.3.1.52.3	Time of the last termination (UNIX time)
<i>lotus</i>	.3.1.60	drweb-lotus component data
lotusState	.3.1.60.1	Current state of the component (integer****)
lotusExitCode	.3.1.60.2	Last exit code (integer corresponding to code from error catalogue)
lotusExitTime	.3.1.60.3	Time of the last termination (UNIX time)
<i>macgui</i>	.3.1.100	drweb-gui (for macOS) component data
macguiState	.3.1.100.1	Current state of the component (integer****)
macguiExitCode	.3.1.100.2	Last exit code (integer corresponding to code from error catalogue)
macguiExitTime	.3.1.100.3	Time of the last termination (UNIX time)
<i>macspider</i>	.3.1.102	drweb-spider (for macOS) component data
macspiderState	.3.1.102.1	Current state of the component (integer****)
macspiderExitCode	.3.1.102.2	Last exit code (integer corresponding to code from error catalogue)



Parameter name	OID of the parameter	Type and description of the parameter
macspiderExitTime	.3.1.102.3	Time of the last termination (<i>UNIX time</i>)
macspiderWorkStatus	.3.1.102.101	Component's current mode of operation (integer: 0—not set, 1—loading, 2—is running)
<i>macfirewall</i>	.3.1.103	drweb-firewall (for macOS) component data
macfirewallState	.3.1.103.1	Current state of the component (integer****)
macfirewallExitCode	.3.1.103.2	Last exit code (integer corresponding to code from error catalogue)
macfirewallExitTime	.3.1.103.3	Time of the last termination (<i>UNIX time</i>)
<i>linuxgui</i>	.3.1.200	drweb-gui (for Linux) component data
linuxguiState	.3.1.200.1	Current state of the component (integer****)
linuxguiExitCode	.3.1.200.2	Last exit code (integer corresponding to code from error catalogue)
linuxguiExitTime	.3.1.200.3	Time of the last termination (<i>UNIX time</i>)
<i>linuxspider</i>	.3.1.201	drweb-spider (for Linux) component data
linuxspiderState	.3.1.201.1	Current state of the component (integer****)
linuxspiderExitCode	.3.1.201.2	Last exit code (integer corresponding to code from error catalogue)
linuxspiderExitTime	.3.1.201.3	Time of the last termination (<i>UNIX time</i>)
linuxspiderWorkStatus	.3.1.201.101	Component's current mode of operation (integer: 0—not set, 1



Parameter name	OID of the parameter	Type and description of the parameter
		—loading, 2—running via fanotify , 3—running via LKM)
<i>linuxnss</i>	.3.1.202	drweb-nss (for Linux) component data
linuxnssState	.3.1.202.1	Current state of the component (integer****)
linuxnssExitCode	.3.1.202.2	Last exit code (integer corresponding to code from error catalogue)
linuxnssExitTime	.3.1.202.3	Time of the last termination (<i>UNIX time</i>)
linuxnssScannedFiles	.3.1.202.101	Number of scanned files (counter, integer)
linuxnssScannedBytes	.3.1.202.102	Number of scanned bytes (counter, integer)
linuxnssScanErrors	.3.1.202.103	Number of scanning error occurrences (counter, integer)
<i>linuxfirewall</i>	.3.1.203	drweb-firewall (for Linux) component data
linuxfirewallState	.3.1.203.1	Current state of the component (integer****)
linuxfirewallExitCode	.3.1.203.2	Last exit code (integer corresponding to code from error catalogue)
linuxfirewallExitTime	.3.1.203.3	Time of the last termination (<i>UNIX time</i>)
<i>ctl</i>	.3.1.300	drweb-ctl component data
ctlState	.3.1.300.1	Current state of the component (integer****)
ctlExitCode	.3.1.300.2	Last exit code (integer corresponding to code from error catalogue)



Parameter name	OID of the parameter	Type and description of the parameter
ctlExitTime	.3.1.300.3	Time of the last termination (<i>UNIX time</i>)
license	.3.2	License status
<i>licenseEsMode</i>	.3.2.1	The license has been granted by the central protection server (integer: 0—no, 1—yes)
<i>licenseNumber</i>	.3.2.2	License number (integer)
<i>licenseOwner</i>	.3.2.3	License owner (string)
<i>licenseActivated</i>	.3.2.4	License activation date (<i>UNIX time</i>)
<i>licenseExpires</i>	.3.2.5	License expiration date (<i>UNIX time</i>)

*) Threat types:

Code	Threat Type
1	Known virus
2	Suspicious object
3	Adware
4	Dialer
5	Joke program
6	Riskware
7	Hacktool

**) Categories of URL:

Code	Threat Type
1	Infection source
2	Not recommended
3	Adult content



Code	Threat Type
4	Violence
5	Weapons
6	Gambling
7	Drugs
8	Obscene language
9	Chats
10	Terrorism
11	Free email
12	Social networks
13	URL added due to a notice from copyright owner
14	Added to black list

***) Codes of Dr.Web components:

Code	Component
1	Dr.Web ConfigD (drweb-configd)
2	Dr.Web Scanning Engine (drweb-se)
3	Dr.Web File Checker (drweb-filecheck)
4	Dr.Web Updater (drweb-update)
5	Dr.Web ES Agent (drweb-esagent)
6	Dr.Web Network Checker (drweb-netcheck)
7	Dr.Web HTTPD (drweb-httpd)
8	Dr.Web SNMPD (drweb-snmpd)
20	Dr.Web ClamD (drweb-clamd)
21	Dr.Web ICAPD (drweb-icapd)
40	SpIDer Guard for SMB (drweb-smbspider-daemon)
41	SpIDer Gate (drweb-gated)



Code	Component
42	Dr.Web MailD (drweb-maild)
43	Dr.Web LookupD (drweb-lookupd)
50	Dr.Web CloudD (drweb-cloudd)
51	Dr.Web VPND (drweb-vpnd)
52	Dr.Web MeshD (drweb-meshd)
60	Dr.Web for Lotus
100	drweb-gui for macOS
102	SpIDer Guard for macOC
103	Dr.Web Firewall for Linux for macOS
200	drweb-gui for Linux
201	SpIDer Guard (drweb-spider)
202	SpIDer Guard for NSS (drweb-nss)
203	Dr.Web Firewall for Linux (drweb-firewall) for Linux
300	Dr.Web Ctl (drweb-ctl)
400	Enterprise scanner (this is not a real component of the product)

****) Possible states of the components:

Code	Status
0	Not installed
1	Installed but not started
2	Is starting
3	Is running
4	Is exiting

To get the values of the variables directly, you can use the **snmpwalk** utility:

```
$ snmpwalk -Os -c <community> -v <SNMP version> <host address> <OID>
```



For example, to get statistics about the threats detected on the local machine, use the following command (if the settings of Dr.Web SNMPD are set to their default values):

```
$ snmpwalk -Os -c public -v 2c 127.0.0.1 .1.3.6.1.4.1.29690.2.2.1
```



Dr.Web CloudD

The Dr.Web CloudD component refers to Dr.Web Cloud (a cloud service of Doctor Web). Dr.Web Cloud service collects up-to-date information from all Dr.Web anti-virus solutions about detected threats to prevent users from visiting unwanted websites and to protect operating systems from infected files containing brand-new threats that do not have any description in Dr.Web virus databases. Moreover, the use of Dr.Web Cloud service reduces the probability of false positives of the [Dr.Web Scanning Engine](#).

Operating Principles

The component is designed to refer to the Dr.Web Cloud service to scan contents of the specified file for threats unknown to the local [Dr.Web Scanning Engine](#), and to check whether the specified URL belongs to any of Doctor Web's predefined categories of web resources.

Dr.Web CloudD is automatically run by the configuration daemon. The component is run upon receiving a command from the user or one of the Dr.Web for UNIX File Servers components. The operation scheme is shown in the figure below.

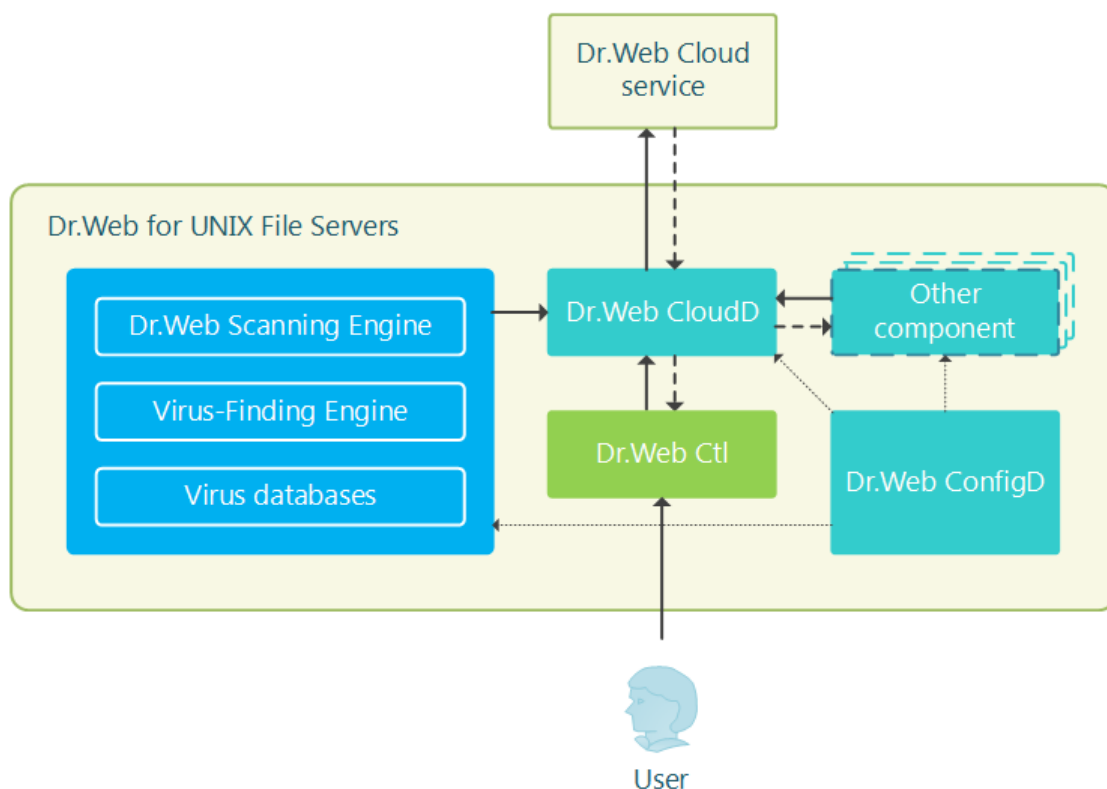


Figure 23. Diagram of the components' operation

component. Requests to the Dr.Web Cloud service (to scan URL or files) via this component can be sent by different components of Dr.Web for UNIX File Servers, marked in the scheme as "Other component" (depending on the product solution).



Besides that, the component is used during the scanning of files on the command from the Dr.Web for UNIX File Servers product management utility from the command line [Dr.Web Ctl](#) (it is started by the **drweb-ctl** command): upon detection of threats, the [Dr.Web Scanning Engine](#) scanning engine sends a report about the file to Dr.Web Cloud.

Command-Line Arguments

To run Dr.Web CloudD, type the following command in the command line:

```
$ <opt_dir>/bin/drweb-cloudd [<parameters>]
```

Dr.Web CloudD can process the following options:

Parameter	Description
--help	Function: Instructs to output short help information about command-line parameters to the console or to the terminal emulator and to exit upon completion. Short form: -h Arguments: None.
--version	Function: Instructs to output information about the version of this component to the console or to the terminal emulator and to exit after completion. Short form: -v Arguments: None.

Example:

```
$ /opt/drweb.com/bin/drweb-clamd --help
```

This command outputs short help information on Dr.Web CloudD.

Startup Notes

The component cannot be launched directly from the command line of the operating system in an autonomous mode (autonomously from other components). It is launched automatically by the [Dr.Web ConfigD](#) configuration daemon when needed. To manage the operation of the component you can use the [Dr.Web Ctl](#) command-line-based management tool for Dr.Web for UNIX File Servers (it is called by using the **drweb-ctl** [command](#)).



To request documentation about this component of the product from the command line, use the following command **man 1 drweb-cloudd**



Configuration Parameters

The component uses configuration parameters which are specified in the [CloudD] section of the integrated [configuration file](#) of Dr.Web for UNIX File Servers.

The section contains the following parameters:

LogLevel <i>{logging level}</i>	<p>Logging level of the component.</p> <p>If the parameter value is not specified, the DefaultLogLevel parameter value from the [Root] section is used.</p> <p>Default value: Notice</p>
Log <i>{log type}</i>	<p>Logging method</p>
ExePath <i>{path to file}</i>	<p>Path to the executable file of the component.</p> <p>Default value: <opt_dir>/bin/drweb-cloudd</p> <ul style="list-style-type: none">• For Linux, Solaris: /opt/drweb.com/bin/drweb-cloudd• For FreeBSD: /usr/local/libexec/drweb.com/bin/drweb-cloudd
RunAsUser <i>{UID user name}</i>	<p>The parameter determines under which user name the component should be run. The user name can be specified either as the user's number UID or as the user's login. If the user name consists of numbers (i.e. similar to number UID), it is specified with the "name:" prefix, for example: RunAsUser = name:123456.</p> <p><i>When a user name is not specified, the component operation terminates with an error after the startup.</i></p> <p>Default value: drweb</p>
IdleTimeLimit <i>{time interval}</i>	<p>Maximum time that the component can remain idle. If the specified value is exceeded, the component shuts down.</p> <p>Minimum value—10s.</p> <p>Default value: 30s</p>
PersistentCache <i>{On Off}</i>	<p>Enable or disable saving of the cache of Dr.Web Cloud replies to the disk.</p> <p>Default value: Off</p>
DebugSdk <i>{Boolean}</i>	<p>Indicates whether detailed Dr.Web Cloud messages should be included into the log file on the debug level (LogLevel = DEBUG).</p> <p>Default value: No</p>



Appendices

Appendix A. Types of Computer Threats

Herein, the term “*threat*” is defined as any kind of software potentially or directly capable of inflicting damage to a computer or network and compromising the user’s information or rights (that is, malicious and other unwanted software). In a wider sense, the term “*threat*” may be used to indicate any type of potential danger to the security of the computer or network (that is, vulnerabilities that can result in hacker attacks).

All of the program types stated below have the ability to endanger user data or confidentiality. Programs that do not conceal their presence in the system (e.g. spam distribution software and various traffic analyzers) are usually not considered as computer threats, although they can become threats under certain circumstances.

Computer Viruses

This type of computer threats is characterized by the ability to embed its code into other programs. Such implementation is called infection. In most cases, an infected file becomes a virus carrier and the embedded code does not necessarily match the original one. Most viruses are intended to damage or destroy data in the system.

In Doctor Web classification, viruses are divided by the type of objects they infect:

- *File viruses* infect files of the operating system (usually executable files and dynamic libraries) and are activated when the infected file is launched.
- *macro-viruses* are viruses that infect documents used by **Microsoft® Office** and some other applications supporting macro commands (for example, written in Visual Basic). *Macro commands* are a type of implemented programs (macros) written in a fully functional programming language. For instance, in **Microsoft® Word**, macros can be automatically Initiated upon opening (closing, saving, etc.) a document.
- *Script viruses* are created using script languages and usually infect other scripts (e.g. service files of an operating system). They are also able to infect other file formats that allow execution of scripts and thus take advantage of scripting vulnerabilities in web applications.
- *boot viruses* infect boot records of disks and partitions or master boot records of hard drives. They do not require much memory and remain ready to continue performing their tasks until a system roll-out, restart or shut-down is performed.

Most viruses have some kind of protection against detection. Protection methods are being constantly improved, and ways to overcome them are constantly being developed. All viruses may also be classified according to protection type they use:



- *Encrypted viruses* cipher their code upon every infection to hamper their detection in a file, boot sector or memory. All copies of such viruses contain only a small common code fragment (the decryption procedure) that can be used as a virus signature.
- *Polymorphic viruses* also encrypt their code, but besides that they also generate a special decryption procedure that is different in every copy of the virus. This means that such viruses do not have byte signatures.
- *Stealth viruses* perform certain actions to disguise their activity and thus conceal their presence in an infected object. Such viruses gather the characteristics of an object before infecting it and then plant these “dummy” characteristics that mislead the scanner searching for modified files.

Viruses can also be classified according to the programming language in which they are written (in most cases, it is Assembler, high-level programming languages, script languages, and others) or according to affected operating systems.

Computer Worms

Recently, malicious programs of the “computer worm” type have become much more common than viruses and other types of malware. Just like viruses, such programs can make copies of themselves, however they do not infect other objects. A worm gets into a computer from a network (most frequently as an attachment to an email or from the Internet) and sends the functioning copies of itself to other computers. To start their spread, worms can either rely on the computer user’s actions or can select and attack computers in an automatic mode.

Worms do not necessarily consist of only one file (the worm’s body). Many of them have an infectious part (the shellcode) that loads into the main memory (RAM) and then downloads the worm’s body as an executable file via the network. If only the shellcode is present in the system, the worm can be deleted by simply restarting the system (at which the RAM is erased and reset). However, if the worm’s body infiltrates the computer, then only an anti-virus program can cope with it.

Worms have the ability to cripple entire networks even if they do not bear any payload (i.e. do not cause any direct damage) due to their intensive distribution.

In Doctor Web classification, worms are divided by distribution method:

- *Net worms* distribute their copies via various network and file-sharing protocols.
- *Mail worms* spread themselves using email protocols (POP3, SMTP, etc.).
- *Chat worms* use protocols of popular messengers and chat programs (ICQ, IM, IRC, etc.).

Trojan Programs (Trojans)

This type of threats cannot reproduce itself. A Trojan substitutes a frequently-used program and performs its functions (or imitates its operation). Meanwhile, it performs some malicious actions in the system (damages or deletes data, sends confidential information, etc.) or makes it possible for hackers to access the computer without permission, for example, to harm the computer of a third party.



A Trojan's masking and malicious facilities are similar to those of a virus. A Trojan may even be a component of a virus. However, most Trojans are distributed as separate executable files (through file exchange servers, removable data carriers or email attachments) that are launched by users or system tasks.

It is very hard to classify Trojans due to the fact that they are often distributed by viruses or worms and also because many malicious actions that can be performed by other types of threats are attributed to Trojans only. Here are some Trojan types which are distinguished as separate classes in Doctor Web:

- *Backdoors* are Trojans that make it possible for an intruder to log on into the system or obtain privileged functions bypassing any existing access and security measures. Backdoors do not infect files, but they write themselves into the registry modifying the registry keys.
- *rootkits* are used to intercept system functions of an operating system in order to conceal themselves. Besides, a rootkit can conceal processes of other programs (e.g. other threats), registry keys, folders and files. It can be distributed either as an independent program or as a component of another malicious program. There are two kinds of rootkits according to the mode of operation: *User Mode Rootkits (UMR)* that operate in user mode (intercept functions of the user mode libraries) and *Kernel Mode Rootkits (KMR)* that operate in kernel mode (intercept functions on the level of the system kernel, which makes it harder to detect).
- *Keyloggers* are used to log data that users enter by means of a keyboard. The aim of this is to steal personal information (i.e. network passwords, logins, credit card data, etc.).
- *Clickers* redirect hyperlinks to certain addresses (sometimes malicious) in order to increase traffic of websites or perform DDoS attacks.
- *Proxy Trojans* provide anonymous Internet access through a victim's computer.

In addition, Trojans can also change the start page in a web browser or delete certain files. However, these actions can also be performed by other types of threats (viruses and worms).

Hacktools

Hacktools are programs designed to assist the intruder with hacking. The most common among them are port scanners that detect vulnerabilities in firewalls and other components of computer protection system. Besides hackers, such tools are used by administrators to check security of their networks. Occasionally, common software that can be used for hacking and various programs that use social engineering techniques are designated as among hacktools as well.

Adware

Usually, this term refers to a program code implemented into freeware programs that force display of advertisements to users. However, sometimes such codes can be distributed via other malicious programs and show advertisements in web browsers. Many adware programs operate with data collected by spyware.



Jokes

Like adware, this type of minor threats can not be used to inflict any direct damage to the system. Joke programs usually just generate messages about errors that never occurred and threaten to perform actions that will lead to data loss. Their purpose is to frighten or annoy users.

Dialers

These are special programs that are designed to scan a range of telephone numbers and find those where a modem answers. These numbers are then used to mark up the price of telephoning facilities or to connect the user to expensive telephone services.

Riskware

These software applications were not created for malicious purposes, but due to their characteristics can pose a threat to the computer's security. Riskware programs can not only damage or delete data, but they are also used by crackers (i.e. malevolent hackers) or by some malicious programs to harm the system. Among such programs, there are various remote chat and administrative tools, FTP-servers, etc.

Suspicious objects

These are possible computer threats detected by the heuristic analyzer. Such objects can potentially be any type of threat (even unknown to IT security specialists) or turn out to be safe in case of false detection. It is recommended that you choose to move the files containing suspicious objects to the quarantine, they also should be sent to Doctor Web anti-virus laboratory for analysis.



Appendix B. Neutralizing Computer Threats

All Doctor Web anti-virus solutions use several malicious software detection methods simultaneously, and that allows them to perform thorough checks on suspicious files and control software behavior.

Detection Methods

Signature Analysis

Signature analysis is the first stage of detection procedure and is used to check file code segments for the presence of known virus signatures. A signature is a finite continuous sequence of bytes necessary and sufficient to identify a specific virus. To reduce the size of the signature dictionary, Dr.Web anti-virus solutions use signature checksums instead of complete signature sequences. Checksums uniquely identify signatures, which preserves correctness of virus detection and neutralization. The Dr.Web virus databases are composed so that some entries can be used to detect not just specific viruses, but whole classes of threats.

Origins Tracing™

On completion of signature analysis, Dr.Web anti-virus solutions use the unique Origins Tracing™ method to detect new and modified viruses which use the known infection mechanisms. Thus, Dr.Web users are protected against such threats as the notorious **Trojan.Encoder.18** ransomware (also known as **gpcode**). In addition to detection of new and modified viruses, the Origins Tracing™ mechanism allows to considerably reduce the number of false positives of the heuristics analyzer. Objects detected using the Origins Tracing™ algorithm are indicated with the `.Origin` extension added to their names.

Execution Emulation

The technology of program code emulation is used for detection of polymorphic and encrypted viruses when a search by checksums cannot be applied directly, or is very difficult to be performed (due to the impossibility of building secure signatures). The method implies simulating the execution of an analyzed code by an *emulator*—a programming model of the processor and runtime environment. An emulator operates with protected memory area (*emulation buffer*), in which execution of the analyzed program is modelled instruction by instruction. However, none of these instructions is actually executed by the CPU. When the emulator receives a file infected with a polymorphic virus, the result of the emulation is a decrypted virus code, which is then easily determined by searching against signature checksums.



Heuristic Analysis

The detection method used by the heuristics analyzer is based on certain knowledge (*heuristics*) about certain features (attributes) than might be typical for the virus code itself, and vice versa, that are extremely rare in viruses. Each attribute has a *weight* coefficient which determines the level of its severity and reliability. The weight coefficient can be positive if the corresponding attribute is indicative of a malicious code or negative if the attribute is uncharacteristic of a computer threat. Depending on the sum weight of a file, the heuristics analyzer calculates the probability of unknown virus infection. If the threshold is exceeded, the heuristic analyzer generates the conclusion that the analyzed object is probably infected with an unknown virus.

The heuristics analyzer also uses the FLY-CODE™ technology, which is a versatile algorithm to extract packed files. The technology allows making heuristic assumptions about the presence of malicious objects in files compressed not only by packers that Dr.Web is aware of, but by also new, previously unexplored programs. While checking packed objects, Dr.Web Anti-virus solutions also use structural entropy analysis. The technology detects threats by the characteristic way in which pieces of code are arranged inside a file; thus, one virus-database entry allows identification of a substantial portion of threats packed with the same polymorphous packer.

As any system of hypothesis testing under uncertainty, the heuristics analyzer may commit type I or type II errors (omit viruses or raise false positives). Thus, objects detected by the heuristics analyzer are treated as “suspicious”.

While performing any of the checks previously mentioned, Dr.Web anti-virus solutions use the most recent information about known malicious software. As soon as experts of Doctor Web anti-virus laboratory discover new threats, an update for virus signatures, behavior characteristics and attributes is issued. In some cases updates can be issued several times per hour. Therefore even if a brand new malicious program passes through the Dr.Web resident guards and penetrates the system, then after an update the malicious program is detected in the list of processes and neutralized.

Actions

To avert computer threats, Dr.Web products use a number of actions that can be applied to malicious objects. A user can leave the default settings, configure which actions to apply automatically, or choose actions manually upon every detection. Below, you can see a list of available actions:

- *Ignore*—instructs to skip the detected threat without performing any other action.
- *Report*—instructs to inform on the detected threat without performing any other action.
- *Block*—instructs to block all attempts to access the infected file (the action might not be available for all components).
- *Cure*—instructs to cure the infected object by removing only malicious content from its body. Note that this action cannot be applied to all types of threats.



- *Quarantine (Move to Quarantine, Isolate)*—instructs to move the detected threat to a special directory and isolate it from the rest of the system.
- *Delete*—instructs to remove the infected object permanently.



If threat is detected in a file located in a container (an archive, email message, etc.), its removal is replaced with moving of a container to quarantine.



Appendix C. Contacting 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.



Appendix D. Configuration File

Configuration parameters of all Dr.Web for UNIX File Servers components are managed by a special coordinating daemon Dr.Web ConfigD. These parameters are stored in the `drweb.ini` file, which default directory is `etc_dir` (for **GNU/Linux** `/etc/opt/drweb.com`).



The text configuration file stores only those parameters which values differ from the defaults. If a parameter is absent in the configuration file, its default value is used.

For details on conventions for `<opt_dir>`, `<etc_dir>`, and `<var_dir>`, refer to the [Introduction](#).

You can view the list of all available parameters, including those that are absent in the configuration file and have default values, by using the following command:

```
$ drweb-ctl cfshow
```

You can change any parameter value in one of the two following ways:

1. Specify the parameter in the configuration file (by editing the file in any text editor) and send SIGHUP signal to the configuration daemon (the **drweb-configd** component) in order to apply the changes (to do that, you can issue the **drweb-ctl reload** [command](#)).
2. Type this command in the command line

```
# drweb-ctl cfset <section>.<parameter> <new value>
```



Note, that this command can be executed only if the management tool Dr.Web Ctl is run with superuser privileges. To gain superuser privileges, use **su** or **sudo** command.

For further information about the `cfshow` and `cfset` command syntax of the command-line management tool Dr.Web Ctl (the **drweb-ctl** module), refer to the section [Dr.Web Ctl](#).

File Structure

The configuration file has the following structure:

- File content is divided into named sections. Possible names of these sections are strictly predefined and cannot be changed. The section name is specified in square brackets and is similar to the component name Dr.Web for UNIX File Servers, which uses the section parameters (except for `[Root]` [section](#) , which stores all parameters of the configuration daemon Dr.Web ConfigD).
- The `' ; '` or `' # '` characters in the configuration file indicate the beginning of a comment—all text following the characters is skipped by components Dr.Web for UNIX File Servers while reading configuration parameters.



- One line of the file can contain only one parameter value. The general format of specifying the value is as follows (white spaces before and after the character '=' are ignored):

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

- All parameter names are strictly predefined and cannot be changed.
- All section and parameter names are case-insensitive. Parameter values, except for names of directories and files in paths (for **UNIX**-like OS) are also case-insensitive.
- Order in which sections are specified in the file and order in which parameters are specified in the section are of no importance.
- Parameter values in the configuration file can be enclosed in quotation marks, and must be enclosed in quotation marks if they have white spaces
- Some parameters can have a list of values. If so, the values are either separated with commas or specified several times in different lines of the configuration file. In the former case, white spaces around a comma are ignored. If a white space character is a part of a parameter value, the character must be enclosed in quotation marks.

Example of how to specify several values for one parameter:

- 1) As a comma-separated list:

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

- 2) In different lines of the configuration file:

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

Note that the order in which parameter values are specified is of no importance.



If parameter values are paths, then each listed parameter value must be put between quotation marks if the listing of components via commas is used. For example, if the parameter **ExcludedPaths** requires two paths `/etc/file1` and `/etc/file2`, then this parameter must be written into a configuration file either as a string

```
ExcludedPaths = "/etc/file1", "/etc/file2"
```

or as two strings

```
ExcludedPaths = /etc/file1  
ExcludedPaths = /etc/file2
```

Otherwise, the component that uses this parameter can interpret the string `'/etc/file1, /etc/file2'` as one path.

- If a parameter can have more than one values, it is designated explicitly. So, if this is not explicitly designated in the current manual or within the comments in the configuration file, the parameter can have only one value.



For description of the configuration file sections, see description of Dr.Web for UNIX File Servers components.

Parameter Types

Configuration parameters can be of the following types:

- *Address*—network connection address specified as `<IP address>:<port>` pair of values. In some cases, the port value can be omitted (if so, it is specified in the parameter description).
- *Boolean*—flag used as an indicator. Such parameters can have either `Yes` or `No` as values.
- *Integer*—parameter value can be a non-negative integer as a value.
- *Fractional number*—parameter value can be a non-negative number with a fractional part.
- *Time interval*—parameter value can be a time interval, consisting of a non-negative integer and a suffix (letter), which stands for a time unit. The following suffixes can be used:
 - `w`—weeks (`1w = 7d`);
 - `d`—days (`1d = 24h`);
 - `h`—hours (`1h = 60m`);
 - `m`—minutes (`1m = 60s`);
 - `s`—seconds.

If the suffix is omitted, the interval is considered as in seconds. For the time interval, expressed in seconds, it is allowed to specify milliseconds after a point (but no more than three digits after the separator, for example, `0.5s` – 500 milliseconds). It is possible to specify several time intervals in different time units. In this case, the resulting interval is counted as a sum of intervals (in fact, a time interval is always converted to milliseconds before the value is written to configuration).

In general terms, any time an interval can be represented as an expression of this form: $N_1wN_2dN_3hN_4mN_5[N_6]s$, where $N_1, ..., N_6$ is a number of corresponding time unites included in this interval. For example, a year (365 days) can be represented as follows (all records are equal): `365d`, `52w1d`, `52w24h`, `51w7d24h`, `51w7d23h60m`, `8760h`, `525600m`, `31536000s`.

The examples below show you how intervals of 30 minutes, 2 seconds, 500 milliseconds can be specified:

1. In the configuration file:

```
UpdateInterval = 30m2.5s
```

2. Using the [command](#) `drweb-ctl` `cfset`:

```
# drweb-ctl cfset Update.UpdateInterval 1802.5s
```

3. Via a command-line parameter (for example, for the [Command Line Arguments](#)):

```
$ drweb-se --WatchdogInterval 1802.5
```



- **Size**—parameter value can be the size of an object (file, buffer, cash, and so on), consisting of a non-negative integer and a suffix, which stands for a unit. The following suffixes can be used:
 - **mb**—megabytes (1mb = 1024kb);
 - **kb**—kilobytes (1kb = 1024b);
 - **b**—bytes.

If the suffix is omitted, the size is considered as in bytes. It is possible to specify several sizes in different units. In this case, the resulting size is counted as their sum (in fact, a size value is always converted to bytes).

- **path to a directory (file)**—parameter value can be a string, which is a path to a directory (file). Note that the file path must be ended with the file name.



In UNIX-like systems, names of catalogs and files are case sensitive. If it is not explicitly designated in a parameter description, paths cannot contain masks with special characters (?, *).

- **Logging level**—the level at which Dr.Web for UNIX File Servers events are logged. The parameter of this type can have the following values:
 - **DEBUG**—the most detailed logging level. All messages and debug information are registered.
 - **INFO**—all messages are registered.
 - **NOTICE**—all error messages, warnings, and notifications are registered.
 - **WARNING**—all error messages and warnings are registered.
 - **ERROR**—only error messages are registered.
- **Log type**—parameter value defines how Dr.Web for UNIX File Servers performs logging (its logging method). The parameter of this type can have the following values:
 - **Stderr[:ShowTimestamp]**—Messages are displayed in the *stderr*—standard error stream. This value can be used *only* in the settings of configuration daemon. At that, if it works in background mode ("*daemonized*"), i.e. it is launched with the parameter *-d* specified, this value *cannot* be used because components operating in the background mode cannot access I/O streams of the terminal). The additional parameter *ShowTimestamp* instructs to add a time stamp to every message.
 - **Auto**—messages for logging are sent to the configuration daemon Dr.Web ConfigD, which saves them to one location according to its configuration (the parameter **Log** in the [Root] section). This value is specified for all components *except for the configuration daemon* and is used as a default value.
 - **Syslog[:<facility>]**—messages are transmitted to the system logging service **syslog**.
 - Additional option *<facility>* is used to specify a level at which **syslog** registers messages. The following values are possible:
 - **DAEMON**—messages of daemons.
 - **USER**—messages of user processes.
 - **MAIL**—messages of mail programs.



- LOCAL0—messages of local processes 0.

...

- LOCAL7—messages of local processes 7.

- *<path>*—Messages are to be saved directly to the specified log.

Example of how to specify the parameter value:

1. In the configuration file:

```
Log = Stderr:ShowTimestamp
```

2. Using the [command drweb-ctl](#) cfset:

```
# drweb-ctl cfset Root.Log /var/opt/drweb.com/log/general.log
```

3. Via a command-line parameter (for example, for the [Command Line Arguments](#)):

```
$ drweb-se --Log Syslog:DAEMON
```

- *action*—action performed by Dr.Web for UNIX File Servers upon detection of certain threats or upon another event. The following values are possible:
 - Report—instructs only to notify on threat detection without performing any other action.
 - Block—instructs to block all attempts to access the infected file without modifying it (the action might not be available for all components).
 - Cure—instructs to attempt to cure the threat (that is, remove only malicious content).
 - Quarantine—instructs to move the infected file to quarantine.
 - Delete — instructs to delete the infected file.



Some of the actions can be applied only upon certain events (for example, a “scanning error” event cannot trigger the *Cure* action). Allowed actions are always listed in the parameter description of the *action* type.

Other parameter types and their possible values are specified in the description of these parameters.



Appendix E. Generating SSL certificates

For the Dr.Web for UNIX File Servers components that use a secure SSL/TLS data channel and application protocols, such as HTTPS, LDAPS, SMTPS, and so on, it is necessary to provide private SSL keys and the corresponding certificates. Keys and certificates for some components are generated automatically; and for others—they should be provided by the user. All the components use certificates in the PEN format.

To generate private keys and certificates used for connections via SSL/TLS, including verification certificates of Certification Authority (CA) and signed certificates, you can use the command-line utility **openssl** (included in an **OpenSSL** cryptographic package).

Consider sequence of actions required for generating a private key and the corresponding SSL certificate together with a SSL certificate signed by the CA verification certificate.

Generating a private SSL key and a certificate

The generation procedure consists of two steps:

1. Generating a private key (the RSA algorithm, the key's length is 2048 bits):

```
$ openssl genrsa -out keyfile.key 2048
```

If you want to password-protect the key, use the `-des3` option. The generated key is in the file `keyfile.key` located in the current directory. To view the key, use the command

```
$ openssl rsa -noout -text -in keyfile.key
```

2. Generating a certificate for the specified time period, based on the existing private key (in this case, for 365 days)

```
$ openssl req -new -x509 -days 365 -key keyfile.key -out certificate.crt
```

Note that this command will request data (name, organization, and so on) that should identify the certifying object. The generated certificate will be located into the file `certificate.crt`.

To check the contents of the generated certificate, use the command

```
$ openssl x509 -noout -text -in certificate.crt
```

Registering a certificate as a trusted CA certificate

If you want to register a certificate in the system list of trusted CA certificates (for instance, such a certificate could be generated during the previous step), do the following:

1. Move or copy the certificate file to the system's trusted certificate directory (`/etc/ssl/certs/` in **Debian/Ubuntu**).



2. In the trusted certificate directory, create a symbolic link to the certificate, where the name of the link is the hash value of the certificate.
3. Reindex the contents of the system's directory containing certificates.

The example given below performs all these three actions. This example assumes that the certificate that is registered as a trusted one is located in the file `/home/user/ca.crt`:

```
# cp /home/user/ca.crt ./
# ln -s ca.crt `openssl x509 -hash -noout -in ca.crt`.0
# c_rehash /etc/ssl/certs/
```

Creating a signed certificate

To create a signed certificate, do the following:

1. Generate a request for signing a certificate (*Certificate Signing Request – CSR*) based on the existing private key. If the key is absent, it should be generated. The request for signing is created with the following command:

```
$ openssl req -new -key keyfile.key -out request.csr
```

This command, as well as the command responsible for certificate creation, requests data that should identify the certified object. `keyfile.key` here is the existing file of the private key. The received request will be saved to the file `request.csr`.

To check the result of request creation, use the command

```
$ openssl req -noout -text -in request.csr
```

2. Create a signed certificated, based on the request and the existing CA certificate, by using the following command:

```
$ openssl x509 -req -days 365 -CA ca.crt -CAkey ca.key -set_serial 01 -in request.csr -out sigcert.crt
```

Note that, to create a signed certificate, you should have the following three files: the file of the root certificate `ca.crt` and its private key `ca.key` and the request for signing `request.csr`. The created signed certificate will be saved to the file `sigcert.crt`.

Use the following command to check the result:

```
$ openssl x509 -noout -text -in sigcert.crt
```

You may repeat these procedures as many times as unique certificates you want to create. For example, every agent for distributed file scanning rDr.Web Network Checker within a scanning cluster should has its own key and certificate.



Appendix F. Known Errors



If the occurred error is not present in this section, it is recommended that you contact [technical support](#). Be ready to name the error code and describe steps to reproduce the issue.

To identify the error, we recommend you to configure logging to a separate file and enable output of extended information to the log. For that, execute the following [commands](#):

```
# drweb-ctl cfset Root.Log <path to log file>
# drweb-ctl cfset Root.DefaultLogLevel DEBUG
```

To return to the default logging method and verbosity level, execute the following commands:

```
# drweb-ctl cfset Root.Log -r
# drweb-ctl cfset Root.DefaultLogLevel -r
```

Errors Determined by Code

If instead of receiving a textual error message or a numeric error code you received an internal error code that looks like `EC_XXX` (for instance, `EC_APP_TERMINATED`), then you can find out the numeric error code and the corresponding description of the error given in this section by using the table of the internal [catalog of errors](#).

Error message: *Error on monitor channel.*

Error code: `x1`

Description: One of the components cannot connect with the [Dr.Web ConfigD](#) configuration daemon.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restart the configuration daemon by executing the command

```
# service drweb-configd restart
```

2. Check whether the authentication mechanism for **PAM** is installed, configured and operates correctly. If not so, install and configure it (for details refer to administration guides and manuals for your OS distribution).
3. If **PAM** is configured correctly and restart of the configuration daemon does not help, restore program settings to the defaults.



To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the [configuration file](#)), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the configuration daemon after clearing the contents of the configuration file.

4. If it is not possible to start the configuration daemon, reinstall the `drweb-configd` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Operation is already in progress.*

Error code: x2

Description: Operation requested by the user is already in progress.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Wait until operation is finished. If necessary, repeat the required action after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Operation is in pending state.*

Error code: x3

Description: An operation requested by the user is in pending state (possibly, a network connection is currently establishing or one of the program components is loading or initializing, which takes a long time).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Wait for the operation to start. If necessary, repeat the required action after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Interrupted by user.*

Error code: x4

Description: The action is terminated by the user (possibly, it takes a long time).



For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Operation canceled.*

Error code: x5

Description: The action is canceled (possibly, it takes a long time).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action again.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *IPC connection terminated.*

Error code: x6

Description: An inter-process communication (IPC) connection with one of the components is terminated (most likely, the component shuts down because of the user command or being idle).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. If the operation is not finished, start it again. Otherwise, the termination is not an error.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid IPC message size.*

Error code: x7

Description: A message of invalid size is received during component inter-process communication (IPC).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restart the program by executing the following command:



```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid IPC message format.*

Error code: x8

Description: A message of invalid format is received during component inter-process communication (IPC).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restart the program by executing the following command:

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not ready.*

Error code: x9

Description: The required action cannot be performed because the necessary component or device is not initialized yet.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Component is not installed.*

Error code: x10

Description: One of the components which is necessary to execute a function is not installed.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Install or reinstall the necessary component. If you do not know the component name, try to determine it reviewing the log file.



2. If installation or reinstallation of the necessary component does not help, reinstall the program. For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Unexpected IPC message.*

Error code: x11

Description: An unexpected message is received during component inter-process communication (IPC).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restart the program by executing the following command:

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *IPC protocol violation.*

Error code: x12

Description: Protocol violation happens during component inter-process communication (IPC).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restart the program by executing the following command:

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Subsystem state is unknown.*

Error code: x13

Description: It was discovered that the current state is not known for a certain subsystem that is part of this software and is needed for carrying out the requested operation.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. Repeat the operation.
2. If the error persists, restart the program by executing the command

```
# service drweb-configd restart
```

and then repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Path must be absolute.*

Error code: x20

Description: Absolute path to file or directory is required (beginning with the root directory of the file system). Relative path is used now.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Change the path to the file or the directory so as to make the path absolute.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not enough memory.*

Error code: x21

Description: Not enough memory to complete the required operation (for example, an attempt to open a large file).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Increase size of available memory for program processes (for example, by changing the limits with the **ulimit** command), restart the program and repeat the operation.

Note that in some cases the system service **systemd** can ignore the specified limit changes. In this case, edit (or create if it does not exist) a file `/etc/systemd/system/drweb-configd.service.d/limits.conf` and specify the changed limit value, for example:

```
[Service]
LimitDATA=32767
```

The list of available limits of **systemd** can be viewed in the documentation **man** `systemd.exec`.
Uninstall Dr.Web for UNIX File Servers by entering the following command:



```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *I/O error.*

Error code: x22

Description: An input/output (I/O) error occurs (for example, the drive is not initialized yet or the partition of the file system is not available anymore).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the required I/O device or the partition of the file system is available. If necessary, mount it and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *No such file or directory.*

Error code: x23

Description: A specified object of the file system (file or directory) is missing. Possibly, it is removed.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path. If necessary, change it and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Permission denied.*

Error code: x24

Description: There are not enough permissions to access the specified object of the file system (file or directory).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the path is correct and whether the component has required permissions. If it is necessary to access the object, change access permissions or elevate component's permissions. Repeat the operation.



If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not a directory.*

Error code: x25

Description: A specified object of the file system is not a directory. Enter the path to the directory.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path. Change it and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Data file corrupted.*

Error code: x26

Description: Requested data is corrupted.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the operation.
2. If the error persists, restart the program by executing the command

```
# service drweb-configd restart
```

and then repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *File already exists.*

Error code: x27

Description: On attempt to create a file, another file with the same name is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path. Change it and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Read-only file system.*

Error code: x28

Description: On attempt to create or change an object of the file system (directory, file or socket), it is detected that the file system is read-only.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path. Change it so that the path indicates the writable partition of the file system and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Network error.*

Error code: x29

Description: A network error occurs (possibly, a remote node stops responding unexpectedly or the required connection fails).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the network is available and network settings are correct. If necessary, change network settings and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not a drive.*

Error code: x30

Description: An accessed input/output (I/O) device is not a drive.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the device name. Change the path so that it indicates to the drive and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Unexpected EOF.*

Error code: x31



Description: During data reading, the end of the file is reached unexpectedly.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the name of the file. If necessary, change the path so that it indicates the correct file and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *File was changed.*

Error code: x32

Description: During scanning the file, it is detected that the file was changed.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat scanning.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not a regular file.*

Error code: x33

Description: During accessing an object of the file system. it is detected that it is not a regular file (that is, it is a directory, socket or other object of the file system).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the name of the file. If necessary, change the path so that it indicates the regular file and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Name already in use.*

Error code: x34

Description: On attempt to create an object of the file system (directory, file or socket), another object with the same name is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in



the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path. Change it and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Host is offline.*

Error code: `x35`

Description: A remote node is not available through the network.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the required node is available. If necessary, change the node address and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Resource limit reached.*

Error code: `x36`

Description: The limit defined for the use of a certain resource has been reached.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the availability of the required resource. If necessary, raise the limit on the use of this resource and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Different mount points.*

Error code: `x37`

Description: Attempt to restore a file which requires its movement between the file system directories, which belong to different mounting points.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Choose another path for the file restoration and repeat the operation.



If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Unpacking error.*

Error code: x38

Description: Archive unpacking unsuccessful (it is possibly password protected or corrupted).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Make sure that file is not corrupted. If the archive is protected with password, remove the protection by entering the correct password and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Virus database corrupted.*

Error code: x40

Description: Virus databases are corrupted.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the virus database directory. Change the path, if necessary (the **VirusBaseDir** parameter in the [Root] [section](#) of the [configuration file](#)).

To view and change the path, go to the **Common Settings** page of the [web interface](#) (if it is installed).

You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Root.VirusBaseDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)



```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Non-supported virus database version.*

Error code: x41

Description: Current virus databases are meant for earlier program version.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the virus database directory. Change the path, if necessary (the **VirusBaseDir** parameter in the [Root] [section](#) of the [configuration file](#)).

To view and change the path, go to the **Common Settings** page of the [web interface](#) (if it is installed).

You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Root.VirusBaseDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Empty virus database.*

Error code: x42

Description: Virus databases are empty.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. Check the path to the virus database directory. Change the path, if necessary (the **VirusBaseDir** parameter in the [Root] [section](#) of the [configuration file](#)).
 - To view and change the path, go to the **Common Settings** page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Root.VirusBaseDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. Update virus databases:
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Object cannot be cured.*

Error code: x43

Description: An attempt to apply the Cure action to an incurable object during threat neutralization.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the /var/log/syslog file or the /var/log/messages file, depending on OS).

Resolving the error:

1. Select an action that can be applied to the object and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Non-supported virus database combination.*

Error code: x44

Description: The current virus database combination cannot be supported.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the /var/log/syslog file or the /var/log/messages file, depending on OS).

**Resolving the error:**

1. Check the path to the virus database directory. Change the path, if necessary (the **VirusBaseDir** parameter in the [Root] [section](#) of the [configuration file](#)).

- To view and change the path, go to the **Common Settings** page of the [web interface](#) (if it is installed).
- You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Root.VirusBaseDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Scan limit reached.*

Error code: x45

Description: When scanning an object, the specified limits have been reached (for example, the limit on the size of an unpacked file, on the nesting depth and others).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Change limits for scanning (in the component settings) by any of the following methods:
 - On the page with the component settings on the [web interface](#) (if it is installed)
 - Use the **drweb-ctl cfshow** and **drweb-ctl cfset** [commands](#).
2. After changing the settings, repeat the previously attempted operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Authentication failed.*

Error code: x47



Description: Invalid user credentials are used for authentication.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Enter valid credentials of the user with the necessary privileges. Try to complete authentication again.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Authorization failed.*

Error code: x48

Description: A user whose credentials are used for authorization does not have enough privileges.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Enter valid credentials of the user with the necessary privileges. Try to complete authentication again.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Access token is invalid.*

Error code: x49

Description: One of the program components provides invalid authorization token on attempt to access the operation, requiring elevated privileges.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Enter valid credentials of the user with the necessary privileges. Try to complete authentication again.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid argument.*

Error code: x60

Description: An invalid argument is used on attempt to run a command.

For details on the place and the reason of the error, refer to the program log (by default, it is located in



the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action again using valid argument.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid operation.*

Error code: x61

Description: An attempt to run an invalid command is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action again using valid command.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Superuser privileges required.*

Error code: x62

Description: Only a user with superuser privileges can perform this action.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Elevate your privileges to root privileges and repeat the required action. To elevate privileges, you can use the commands **su** and **sudo**.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Not allowed in central protection mode.*

Error code: x63

Description: The required action can be performed only if the program operates in standalone [mode](#).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Change product's operation mode to standalone mode and repeat the operation.



2. To change the mode

- Clear the **Enable the central protection mode** check box on the **Central protection** of [web interface](#) management (if installed).
- Or execute the [command](#)

```
# drweb-ctl esdisconnect
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Non-supported OS.*

Error code: x64

Description: The program does not support operating system installed on the host.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Install the operating system from the list mentioned in [system requirements](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Feature not implemented.*

Error code: x65

Description: Required features of one of the components are not implemented in the current version of the program.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Restore program settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the [configuration file](#)), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save  
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the program after clearing the contents of the configuration file by executing the command

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Unknown option.*

Error code: x66

Description: The [configuration file](#) contains parameters unknown or non-supported in the current version of the program.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Open the `<etc_dir>/drweb.ini` file in any text editor, remove the line, containing invalid parameter. Save the file and restart the [Dr.Web ConfigD](#) configuration daemon by executing the command:

```
# service drweb-configd restart
```

2. If it does not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the configuration daemon after clearing the contents of the configuration file.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Unknown section.*

Error code: x67

Description: The [configuration file](#) contains sections unknown or non-supported in the current version of the program.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Open the `<etc_dir>/drweb.ini` file in any text editor, remove the unknown (non-supported) section. Save the file and restart the [Dr.Web ConfigD](#) configuration daemon by executing the command:

```
# service drweb-configd restart
```

2. If it does not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```



Restart the configuration daemon after clearing the contents of the configuration file.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid option value.*

Error code: x68

Description: One of the parameters in the [configuration file](#) contains invalid value for the parameter.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Set the valid parameter value by any of the following methods:

- On the page with the component settings on the [web interface](#) (if it is installed)
- Use the **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` [commands](#).

If you do not know which value is valid for the parameter, refer to the help file of the component which uses this parameter. You may also restore parameter value to the default.

2. You may also directly edit the configuration file `<etc_dir>/drweb.ini`. To do this, open the configuration file in any text editor, find the line containing invalid parameter value, set valid value, then save the file and restart the [Dr.Web ConfigD](#) configuration daemon by executing the command

```
# service drweb-configd restart
```

3. If the previous steps do not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the configuration daemon after clearing the contents of the configuration file.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid state.*

Error code: x69

Description: The program or one of the components is in invalid state to complete the required operation.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the required action after some time.
2. If the error persists, restart the program by executing the command



```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Only one value allowed.*

Error code: x70

Description: One of the parameters in the [configuration file](#) contains a list of values; while it is allowed to contain only a single value.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Set the valid parameter value by any of the following methods:

- On the page with the component settings on the [web interface](#) (if it is installed)
- Use the **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` [commands](#).

If you do not know which value is valid for the parameter, refer to the help file of the component which uses this parameter. You may also restore parameter value to the default.

2. You may also directly edit the configuration file `<etc_dir>/drweb.ini`. To do this, open the configuration file in any text editor, find the line containing invalid parameter value, set valid value, then save the file and restart the [Dr.Web ConfigD](#) configuration daemon by executing the command

```
# service drweb-configd restart
```

3. If the previous steps do not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the configuration daemon after clearing the contents of the configuration file.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Tag value is invalid.*

Error code: x71

Description: One of the sections in the [configuration file](#) with a name containing a unique tag identifier has an invalid tag identifier.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. If the error occurs on attempt to create a section in the [web interface](#) or while using the [command](#)

```
# drweb-ctl cfset <section>.<parameter> <new value>
```

set valid value for the tag and save the section again.

2. If the section is saved directly in the configuration file `<etc_dir>/drweb.ini`, edit the file. To do this, open the configuration file in any text editor, find the section name containing invalid tag value and set valid value for the tag. Save the file and restart the [Dr.Web ConfigD](#) configuration daemon by executing the command

```
# service drweb-configd restart
```

3. If the previous steps do not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save  
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the configuration daemon after clearing the contents of the configuration file.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Record not found.*

Error code: x80

Description: On attempt to access a threat record, it is found out that the record is missing (possibly, another program component processed the threat).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Update the threat list after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Record is in process now.*

Error code: x81

Description: On attempt to access a threat record, it is found out that another program component is processing the record now.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. Update the threat list after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *File has already been quarantined.*

Error code: x82

Description: On attempt to move the file with the detected threat to quarantine, it is found out that the file is already in quarantine (most likely, another program component processed the threat).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Update the threat list after some time.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Cannot backup before update.*

Error code: x89

Description: Prior to downloading the updates from the updates server, an attempt to make a backup copy of the files to be updated failed.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the directory that stores backup copies of the files that are updated. Change the path, if necessary (the `BackupDir` parameter in the [Update] [section](#) of the [configuration file](#)).
 - To view and change the path, go to the **Updater** page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Update.BackupDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Update.BackupDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Update.BackupDir -r
```

2. Update virus databases:



- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

3. If the error persists, check whether the user under whose account the Dr.Web Updater component is running has a write permission to the directory specified in the **BackupDir**. The name of this user is specified in the **RunAsUser** parameter. If necessary, change the user specified in the **RunAsUser** parameter or grant the missing permissions in the directory's properties.
4. If the error persists, reinstall the `drweb-update` package.
For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid DRL file*

Error code: x90

Description: An integrity violation of one of the files with the list of update servers is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the file with the list of servers and change the path if necessary (parameters with ***DrlPath** in [секции](#) [Update] [of configuration file](#)).
 - To view and change the path, go to the **Updater** page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view the current parameter value use the command (`<*DrlPath>` needs to be substituted with a specified parameter name. If parameter name is unclear, refer to parameters values in section, skipping the command part in brackets):

```
$ drweb-ctl cfshow Update[.<*DrlPath>]
```

To set new parameter value, execute the command (`<*DrlPath>` needs to be substituted with a specified parameter name):

```
# drweb-ctl cfset Update.<*DrlPath> <new path>
```

To restore parameter value to the default, execute the command (`<*DrlPath>` needs to be substituted with a specified parameter name):

```
# drweb-ctl cfset Update.<*DrlPath> -r
```

2. Update virus databases:
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)



```
$ drweb-ctl update
```

3. If the error persists, reinstall the `drweb-update` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid LST file.*

Error code: `x91`

Description: An integrity violation of the file containing the list of updated virus databases is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Update virus databases again after some time.
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)

```
$ drweb-ctl update
```

2. If the error persists, reinstall the `drweb-update` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid compressed file.*

Error code: `x92`

Description: An integrity violation of the downloaded file containing updates is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Update virus databases again after some time.
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Proxy authentication error.*

Error code: x93

Description: The program fails to connect to update servers using the proxy server specified in the settings.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the parameters used to connect to a proxy server (they are set in the **Proxy** parameter in the [Update] [section](#) of the [configuration file](#)).

- To view and set the connection parameters, go to the **Updater** page of the [web interface](#) (if it is installed).
- You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Update.Proxy
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Update.Proxy <new parameters>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Update.Proxy -r
```

2. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *No update servers available.*

Error code: x94

Description: The program fails to connect to any of the update servers.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the network is available. Change network settings, if necessary.
2. If you can access the network only using a proxy server, set parameters to connect to the proxy server (you can set them in the **Proxy** parameter in the [Update] [section](#) of the [configuration file](#)).



- To view and set the connection parameters, go to the **Updater** page of the [web interface](#) (if it is installed).
- You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Update.Proxy
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Update.Proxy <new parameters>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Update.Proxy -r
```

3. If network connection parameters (including parameters of proxy server) are correct, but the error occurs, make sure you use the available list of update servers. The list of update servers used is displayed in parameters ***DrlPath** in [Update] section of configuration file. Note that, if parameters ***CustomDrlPath** indicate the existing correct file of servers list, the servers specified there will be used instead of the servers of the standard update zone (the value specified in the corresponding parameter ***DrlPath**, is ignored).

- To view and set the connection parameters, go to the **Updater** page of the [web interface](#) (if it is installed).
- You also may use the [commands](#) of the command-line management tool.

To view the current parameter value use the command (<**DrlPath*> needs to be substituted with a specified parameter name. If parameter name is unclear, refer to parameters values in section, skipping the command part in brackets):

```
$ drweb-ctl cfshow Update[.<*DrlPath>]
```

To set new parameter value, execute the command (<**DrlPath*> needs to be substituted with a specified parameter name:

```
# drweb-ctl cfset Update.<*DrlPath> <new path>
```

To restore parameter value to the default, execute the command (<**DrlPath*> needs to be substituted with a specified parameter name:

```
# drweb-ctl cfset Update.<*DrlPath> -r
```

4. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Invalid key file format.*

Error code: x95

Description: The key file format is violated.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether you have the key file and the path to it. You can specify the path to the key file in the **KeyPath** parameter in the [Root] [section](#) of the [configuration file](#).
 - To view and set the path to the key file, go to the **Common Settings** page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Root.KeyPath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.KeyPath <path to file>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.KeyPath -r
```

2. If you do not have the key file or the used key file is corrupted, purchase and install it. For more details on the key file, purchase and installation refer to section [Licensing](#).
3. To install the key file, you may use the license activation form which is located at the bottom of the **Main** page of the [web interface](#) (if it is installed).
4. You can view current license options in user's webpage **My Dr.Web** at <https://support.drweb.com/get+cabinet+link/>.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *License is already expired.*

Error code: x96

Description: The used license is expired.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Purchase a new license and install a key file that you will receive. For more details on ways to purchase the license and installation of the key file refer to section [Licensing](#).
2. To install the received key file, you may use the license activation form which is located at the



bottom of the **Main** page of the [web interface](#) (if it is installed).

3. You can view current license options in user's webpage **My Dr.Web** at <https://support.drweb.com/get+cabinet+link/>.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Network operation timed out.*

Error code: x97

Description: Network operation timed out (possibly, a remote node stops responding unexpectedly or the required connection fails).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check whether the network is available and network settings are correct. If necessary, change network settings and repeat the operation.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid checksum*

Error code: x98

Description: A checksum of the downloaded file containing updates is detected.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Update virus databases again after some time.
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid demo key file.*

Error code: x99

Description: The used demo key file is invalid (for example, it was received from another computer).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. Send a request for a new demo period for this computer or purchase a new license and install a key file that you will receive. For more details on ways to purchase the license and installation of the key file refer to section [Licensing](#).
2. To install the received key file, you may use the license activation form which is located at the bottom of the **Main** page of the [web interface](#) (if it is installed).
3. You can view current license options in user's webpage **My Dr.Web** at <https://support.drweb.com/get+cabinet+link/>.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Key file is blocked.*

Error code: x100

Description: The used license is blocked (probably, the license agreement conditions on using the Dr.Web program are broken).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Purchase a new license and install a key file that you will receive. For more details on ways to purchase the license and installation of the key file refer to section [Licensing](#).
2. To install the received key file, you may use the license activation form which is located at the bottom of the **Main** page of the [web interface](#) (if it is installed).
3. You can view current license options in user's webpage **My Dr.Web** at <https://support.drweb.com/get+cabinet+link/>.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid license.*

Error code: x101

Description: The used license is meant for other product or does not allow operation of the installed product components.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Purchase a new license and install a key file that you will receive. For more details on ways to purchase the license and installation of the key file refer to section [Licensing](#).
2. To install the received key file, you may use the license activation form which is located at the bottom of the **Main** page of the [web interface](#) (if it is installed).
3. You can view current license options in user's webpage **My Dr.Web** at



<https://support.drweb.com/get+cabinet+link/>.

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid configuration.*

Error code: x102

Description: One of the program components cannot be in operation because of incorrect configuration settings.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. If you do not know the name of the component which causes the error, try to determine it by reviewing the log file.
2. If the error is produced by the SpIDer Guard component, most likely that the mode which is selected for the component operation is not supported by OS. Check the selected mode and change it, if necessary. You can do it by setting the value `AUTO` (the **Mode** parameter in the [\[LinuxSpider\] section configuration file](#)).
 - To view and change the mode, go to the **SpIDer Guard** page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To set the value to `AUTO`, execute the command

```
# Dr.Web Ctl cfset LinuxSpider.Mode AUTO
```

To restore the parameter value to the default, execute the command

```
# Dr.Web Ctl cfset LinuxSpider.Mode -r
```

- If the error persists, [manually build and install](#) the loadable kernel module for SpIDer Guard.



Note that operation of SpIDer Guard and of the loadable kernel module is guaranteed only on the tested **UNIX** distributions (see section [System Requirements](#) in Administrator Manual).

If the error is produced by another component, restore component settings to the defaults by any of the following methods:

- On the page with the component settings on the [web interface](#) (if it is installed)
 - Use the **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` [commands](#).
 - Edit the [configuration file](#) manually (delete all parameters from the component section).
3. If the previous steps do not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:



```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save  
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the program after clearing the contents of the configuration file by executing the command

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Invalid executable file*

Error code: x104

Description: One of the program components cannot run, because of the incorrect path or corrupted file contents.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. If you do not know the name of the component which causes the error, try to determine it reviewing the log file.
2. Check the path to the executable file of the component in the program configuration file (the **ExePath** parameter in the component section), by executing the following [command](#) (change `<component section>` for the name of the corresponding section of the [configuration file](#))

```
$ drweb-ctl cfshow <component section>.ExePath
```

3. Restore the path to the default by executing the following command (change `<component section>` for the name of the corresponding section of the configuration file):

```
# drweb-ctl cfset <component section>.ExePath -r
```

4. If the previous steps do not help, reinstall the package of the corresponding component.
For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Virus-Finding Engine is not available.*

Error code: x105

Description: The file of Dr.Web Virus-Finding Engine is missing or unavailable (required for threat detection).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

**Resolving the error:**

1. Check the path to the **drweb32.dll** anti-virus engine file. Change the path, if necessary (the **CoreEnginePath** parameter in the [Root] [section](#) of the [configuration file](#)).
 - To view and change the path, go to the **Common Settings page of the [web interface](#)** (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the command

```
$ drweb-ctl cfshow Root.CoreEnginePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.CoreEnginePath <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.CoreEnginePath -r
```

2. Update virus databases:
 - Click **Update** on the **Main** page of the [web interface](#), if it is installed.
 - Or execute the [command](#)

```
$ drweb-ctl update
```

3. If the path is correct and the error persists after updating virus databases, reinstall the drweb-bases package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *No virus databases.*

Error code: x106

Description: Virus databases are not found.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the virus database directory. Change the path, if necessary (the **VirusBaseDir** parameter in the [Root] [section](#) of the [configuration file](#)).
 - To view and change the path, go to the **Common Settings page of the [web interface](#)** (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the command



```
$ drweb-ctl cfshow Root.VirusBaseDir
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Root.VirusBaseDir -r
```

2. Update virus databases:

- Click **Update** on the **Main** page of the [web interface](#), if it is installed.
- Or execute the [command](#)

```
$ drweb-ctl update
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Process terminated by signal.*

Error code: x107

Description: A component shuts down (possibly, because of the user command or being idle).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. If the operation is not finished, start it again. Otherwise, the shutdown is not an error.
2. If a component shuts down constantly, restore its settings to the defaults by any of the following methods:
 - On the page with the component settings on the [web interface](#) (if it is installed)
 - Use the **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` [commands](#).
 - Edit the [configuration file](#) manually (delete all parameters from the component section).
3. If it does not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save  
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the program after clearing the contents of the configuration file by executing the command

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Unexpected process termination.*

Error code: x108

Description: A component unexpectedly shuts down because of a failure.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Repeat the terminated operation.
2. If the component constantly shuts down abnormally, restore its settings to the defaults by any of the following methods:
 - On the page with the component settings on the [web interface](#) (if it is installed)
 - Use the **drweb-ctl** `cfshow` and **drweb-ctl** `cfset` [commands](#).
 - Edit the [configuration file](#) manually (delete all parameters from the component section).
3. If it does not help, restore program's settings to the defaults.

To do it, clear the contents of the `<etc_dir>/drweb.ini` file (it is recommended that you make a backup of the configuration file), for example, by executing the following commands:

```
# cp /etc/opt/drweb.com/drweb.ini /etc/opt/drweb.com/drweb.ini.save
# echo "" > /etc/opt/drweb.com/drweb.ini
```

Restart the program after clearing the contents of the configuration file by executing the command

```
# service drweb-configd restart
```

4. If the error persists after restoring program settings, reinstall the component package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Incompatible software detected.*

Error code: x109

Description: A program component cannot be in operation because an incompatible software is detected. This software interrupts correct component operation.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Disable or reconfigure the incompatible software so as to prevent any interference with the Dr.Web for UNIX File Servers operation.

If the error persists, contact [technical support](#) and be ready to name the error code.



Error message: *Linux kernel module for SpIDer Guard is not available.*

Error code: x113

Description: SpIDer Guard> requires a **Linux** kernel module which is missing.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check which operating mode of the component was selected and change it—if necessary—by setting the value to `AUTO` (for the **Mode** parameter in the [LinuxSpider] [section](#) of the [configuration file](#)).
 - To view and change the mode, go to the SpIDer Guard page of the [web interface](#) (if it is installed).
 - You also may use the [commands](#) of the command-line management tool.

To set the value to `AUTO`, execute the command

```
# drweb-ctl cfset LinuxSpider.Mode AUTO
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset LinuxSpider.Mode -r
```

2. If the error persists, [manually build and install](#) the loadable kernel module for SpIDer Guard.



Operation of SpIDer Guard and of the loadable kernel module is guaranteed only on the tested **UNIX** distributions (see section [System Requirements](#) in Administrator Manual).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Scanning Engine is not available.*

Error code: x119

Description: Dr.Web Scanning Engine component is missing or failed to start (required for threat detection).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the **drweb-se** executable file. Change the path, if necessary (the **ExePath** parameter in the [ScanEngine] [section](#) of the [configuration file](#)).

You may use the [commands](#) of the command-line management tool.



To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow ScanEngine.ExePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset ScanEngine.ExePath <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset ScanEngine.ExePath -r
```

2. If the error persists after entering the correct path

- Execute the command

```
$ drweb-ctl rawscan /
```

If the line `Error: No valid license provided`, is output, a valid key file is missing. Register the program and receive a license. After receiving the license, check whether the [key file](#) is available and install it, if necessary.

- If you use 64-bit version of the operating system, make sure that 32-bit application support libraries are installed (see section [System Requirements](#)) and, if necessary, install them. After installing the libraries, restart Dr.Web for UNIX File Servers by executing the command

```
# service drweb-configd restart
```

- If your operating system uses **SELinux**, configure the security policy for the **drweb-se** module (see section [Configuring SELinux Security Policies](#)).

3. If the configuration does not contain the Dr.Web Scanning Engine component settings or if the steps previously mentioned do not help, install or reinstall the `drweb-se` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *File Checker is not available.*

Error code: x120

Description: Dr.Web File Checker component is missing or failed to start (required for threat detection).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the **drweb-filecheck** executable file. Change the path, if necessary (the **ExePath** parameter in the [FileCheck] [section](#) of the [configuration file](#)).

You may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:



```
$ drweb-ctl cfshow FileCheck.ExePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset FileCheck.ExePath <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset FileCheck.ExePath -r
```

2. If the error persists after entering the correct path

- If you use 64-bit version of the operating system, make sure that 32-bit application support libraries are installed (see section [System Requirements](#)) and, if necessary, install them. After installing the libraries, restart Dr.Web for UNIX File Servers by executing the command

```
# service drweb-configd restart
```

- If your operating system uses **SELinux**, configure the security policy for the **drweb-filecheck** module (see section [Configuring SELinux Security Policies](#)).

3. If the configuration does not contain the Dr.Web File Checker component settings or if the steps previously mentioned do not help, install or reinstall the `drweb-filecheck` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *ES Agent is not available.*

Error code: x121

Description: Dr.Web ES Agent component is missing (it is necessary to connect to the central protection server).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the **drweb-esagent** executable file. Change the path, if necessary (the **ExePath** parameter in the [ESAgent] [section](#) of the [configuration file](#)).

You may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow ESAgent.ExePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset ESAgent.ExePath <new path>
```



To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset ESAgent.ExePath -r
```

2. If the configuration does not contain settings for the Dr.Web ES Agent component or if the error persists after entering the correct path, install or reinstall the `drweb-esagent` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Network Checker is not available.*

Error code: x123

Description: Dr.Web Network Checker component is missing (required for check of downloaded files).

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the **drweb-netcheck** executable file. Change the path, if necessary (the **ExePath** parameter in the [Netcheck] [section](#) of the [configuration file](#)).

You may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow Netcheck.ExePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset Netcheck.ExePath <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset Netcheck.ExePath -r
```

2. If the configuration does not contain settings for the Dr.Web Network Checker component or if the error persists after entering the correct path, install or reinstall the `drweb-netcheck` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *CloudD is not available.*

Error code: x124

Description: Dr.Web CloudD (required for requests to the Dr.Web Cloud service) is missing.



For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Check the path to the **drweb-cloudd** executable file. Change the path, if necessary (the **ExePath** parameter in the [CloudD] [section](#) of the [configuration file](#)).

You may use the [commands](#) of the command-line management tool.

To view current parameter value, execute the following command:

```
$ drweb-ctl cfshow CloudD.ExePath
```

To set a new parameter value, execute the command

```
# drweb-ctl cfset CloudD.ExePath <new path>
```

To restore the parameter value to the default, execute the command

```
# drweb-ctl cfset CloudD.ExePath -r
```

2. If the configuration does not contain settings for the Dr.Web CloudD component or if the error persists after entering the correct path, install or reinstall the `drweb-cloudd` package.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#) and be ready to name the error code.

Error message: *Unexpected error.*

Error code: x125

Description: Unexpected error occurs in operation of one of the components.

For details on the place and the reason of the error, refer to the program log (by default, it is located in the `/var/log/syslog` file or the `/var/log/messages` file, depending on OS).

Resolving the error:

1. Try restart the program by executing the command

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#) and be ready to name the error code.

Errors Without Codes



Symptoms	After installation of the SpIDer Guard kernel module, operating system abnormally shuts down with a kernel error « <i>Kernel panic</i> »
Description	SpIDer Guard kernel module cannot operate in the operating system kernel environment (for example, when OS operates in the Xen hypervisor environment).
Resolving the error:	
1. Cancel the load of the SpIDer Guard kernel module (kernel module name is <code>drweb</code>) by adding to the grub loader the following string	
<pre>drweb.blacklist=yes</pre>	
to the load settings string of the operating system kernel.	
2. When the OS is loaded, remove the <code>drweb PSB.Ko</code> module from the <code>/lib/'</code> <code>uname-r</code> <code>'/extra</code> directory of additional modules.	
3. Set operation mode for SpIDer Guard to <i>AUTO</i> by executing the following commands:	
<pre># drweb-ctl cfset LinuxSpider.Mode AUTO # drweb-ctl reload</pre>	
4. Do not use the Xen hypervisor, in case your operating system does not support the fanotify interface or this mode does not allow using SpIDer Guard for the full file system monitoring (relevant for GNU/Linux with mandatory access models, for example, for Astra Linux), making the use of the <i>LKM</i> mode mandatory for the file system monitoring.	
If the error persists, contact technical support .	

Symptoms:	Dr.Web MailD, SpIDer Gate, Dr.Web ICAPD (list of the indicated components depends on the installed product) do not scan messages; in the Dr.Web for UNIX File Servers log there are messages <code>Too many open files</code> .
Description:	Due to a large data scanning load, Dr.Web Network Checker has reached the ceiling of the number of available file descriptors.
Resolving the error:	
1. Raise the limit of the number of open file descriptors available to the application via the command ulimit -n (default limit of the descriptor number for Dr.Web for UNIX File Servers is 16384). Note that in some cases the system service systemd can ignore the specified limit changes. In this case, edit (or create if it does not exist) a file <code>/etc/systemd/system/drweb-configd.service.d/limits.conf</code> and specify the changed limit value:	
<pre>[Service] LimitNOFILE=16384</pre>	
The list of available limits of systemd can be viewed in the documentation man systemd.exec .	



2. Once the limit is changed, restart Dr.Web for UNIX File Servers by executing the following command:

```
# service drweb-configd restart
```

If the error persists, contact [technical support](#).

Symptoms: Web browser cannot establish connection to Dr.Web management web interface; components of Dr.Web anti-virus solutions are not in the list of running processes (`ps ax | grep drweb`); attempt to execute any `drweb-ctl <command>`, except for `drweb-ctl rawscan`, results in one of the following errors:

```
Error: connect: No such file or directory: "<path>/  
.com.drweb.public"
```

or

```
Error: connect: Connection refused: "<path>/com.drweb.public".
```

Description: Dr.Web for UNIX File Servers cannot start as the configuration daemon Dr.Web ConfigD is not available.

Resolving the error:

1. Execute the command

```
# service drweb-configd restart
```

to restart Dr.Web ConfigD and Dr.Web for UNIX File Servers.

2. If this command returns error message, or has no any effect, install `drweb-configd` component (package) separately.

Also note that this may mean that **PAM** authentication is not used in the system. If so, please install and configure PAM (the product cannot operate correctly without **PAM**).

3. If the error persists, remove Dr.Web for UNIX File Servers and then install it again.

For details on how to install and uninstall the product or product components, refer to sections [Installing the Product](#) and [Uninstalling the Product](#).

If the error persists, contact [technical support](#).

The Internal Catalog of Errors

Error code	Symbolic representation	Internal message about the error	Description
0	EC_OK	Success	This is not an error
1	EC_MONITOR_IPC_ERROR	Error on monitor channel	Error x1



Error code	Symbolic representation	Internal message about the error	Description
2	EC_ALREADY_IN_PROGRESS	<i>Operation is already in progress</i>	Error x2
3	EC_IN_PENDING_STATE	<i>Operation is in pending state</i>	Error x3
4	EC_INTERRUPTED_BY_USER	<i>Interrupted by user</i>	Error x4
5	EC_CANCELED	<i>Operation canceled</i>	Error x5
6	EC_LINK_DISCONNECTED	<i>Link disconnected</i>	Error x6
7	EC_BAD_MESSAGE_SIZE	<i>Invalid IPC message size</i>	Error x7
8	EC_BAD_MESSAGE_FORMAT	<i>Invalid IPC message format</i>	Error x8
9	EC_NOT_READY	<i>Not ready</i>	Error x9
10	EC_NOT_INSTALLED	<i>Component is not installed</i>	Error x10
11	EC_UNEXPECTED_MESSAGE	<i>Unexpected IPC message</i>	Error x11
12	EC_PROTOCOL_VIOLATION	<i>Protocol violation</i>	Error x12
13	EC_UNKNOWN_STATE	<i>Subsystem state is unknown</i>	Error x13
20	EC_NOT_ABSOLUTE_PATH	<i>Path must be absolute</i>	Error x20
21	EC_NO_MEMORY	<i>Not enough memory</i>	Error x21
22	EC_IO_ERROR	<i>IO error</i>	Error x22
23	EC_NO_SUCH_ENTRY	<i>No such file or directory</i>	Error x23
24	EC_PERMISSION_DENIED	<i>Permission denied</i>	Error x24
25	EC_NOT_A_DIRECTORY	<i>Not a directory</i>	Error x25
26	EC_DATA_CORRUPTED	<i>Data file corrupted</i>	Error x26
27	EC_FILE_EXISTS	<i>File already exists</i>	Error x27
28	EC_READ_ONLY_FS	<i>Read-only file system</i>	Error x28
29	EC_NETWORK_ERROR	<i>Network error</i>	Error x29
30	EC_NOT_A_DRIVE	<i>Not a drive</i>	Error x30
31	EC_UNEXPECTED_EOF	<i>Unexpected EOF</i>	Error x31
32	EC_FILE_WAS_CHANGED	<i>File was changed</i>	Error x32



Error code	Symbolic representation	Internal message about the error	Description
33	EC_NOT_A_REGULAR_FILE	<i>Not a regular file</i>	Error x33
34	EC_NAME_ALREADY_IN_USE	<i>Name already in use</i>	Error x34
35	EC_HOST_OFFLINE	<i>Host is offline</i>	Error x35
36	EC_LIMIT_REACHED	<i>Resource limit reached</i>	Error x36
37	EC_CROSS_DEVICE_LINK	<i>Mounting points are different</i>	Error x37
38	EC_UNPACKING_ERROR	<i>Unpacking error</i>	Error x38
40	EC_BASE_CORRUPTED	<i>Virus base corrupted</i>	Error x40
41	EC_OLD_BASE_VERSION	<i>Non-supported virus database version</i>	Error x41
42	EC_EMPTY_BASE	<i>Empty virus database</i>	Error x42
43	EC_CAN_NOT_BE_CURED	<i>Object cannot be cured</i>	Error x43
44	EC_INVALID_BASE_SET	<i>Non-supported virus database combination</i>	Error x44
45	EC_SCAN_LIMIT_REACHED	<i>Scan limit reached</i>	Error x45
47	EC_AUTH_FAILED	<i>Authentication failed</i>	Error x47
48	EC_NOT_AUTHORIZED	<i>Authorization failed</i>	Error x48
49	EC_INVALID_TOKEN	<i>Access token is invalid</i>	Error x49
60	EC_INVALID_ARGUMENT	<i>Invalid argument</i>	Error x60
61	EC_INVALID_OPERATION	<i>Invalid operation</i>	Error x61
62	EC_ROOT_ONLY	<i>Superuser privileges required</i>	Error x62
63	EC_STANDALONE_MODE_ONLY	<i>Not allowed in central protection mode</i>	Error x63
64	EC_NON_SUPPORTED_OS	<i>Non-supported OS</i>	Error x64
65	EC_NOT_IMPLEMENTED	<i>Feature not implemented</i>	Error x65
66	EC_UNKNOWN_OPTION	<i>Unknown option</i>	Error x66
67	EC_UNKNOWN_SECTION	<i>Unknown section</i>	Error x67
68	EC_INVALID_OPTION_VALUE	<i>Invalid option value</i>	Error x68



Error code	Symbolic representation	Internal message about the error	Description
69	EC_INVALID_STATE	<i>Invalid state</i>	Error x69
70	EC_NOT_LIST_OPTION	<i>Only one value allowed</i>	Error x70
71	EC_INVALID_TAG	<i>Tag value is invalid</i>	Error x71
80	EC_RECORD_NOT_FOUND	<i>Record not found</i>	Error x80
81	EC_RECORD_BUSY	<i>Record is in process now</i>	Error x81
82	EC_QUARANTINED_FILE	<i>File has already been quarantined</i>	Error x82
89	EC_BACKUP_FAILED	<i>Cannot backup before update</i>	Error x89
90	EC_BAD_DRL_FILE	<i>Invalid DRL file</i>	Error x90
91	EC_BAD_LST_FILE	<i>Invalid LST file</i>	Error x91
92	EC_BAD_LZMA_FILE	<i>Invalid compressed file</i>	Error x92
93	EC_PROXY_AUTH_ERROR	<i>Proxy authentication error</i>	Error x93
94	EC_NO_UPDATE_SERVERS	<i>No update servers available</i>	Error x94
95	EC_BAD_KEY_FORMAT	<i>Invalid key file format</i>	Error x95
96	EC_EXPIRED_KEY	<i>License is already expired</i>	Error x96
97	EC_NETWORK_TIMEOUT	<i>Network operation timed out</i>	Error x97
98	EC_BAD_CHECKSUM	<i>Invalid checksum</i>	Error x98
99	EC_BAD_TRIAL_KEY	<i>Invalid trial license</i>	Error x99
100	EC_BLOCKED_LICENSE	<i>Blocked license key</i>	Error x100
101	EC_BAD_LICENSE	<i>Invalid license</i>	Error x101
102	EC_BAD_CONFIG	<i>Invalid configuration</i>	Error x102
104	EC_BAD_EXECUTABLE	<i>Invalid executable file</i>	Error x104
105	EC_NO_CORE_ENGINE	<i>Core engine is not available</i>	Error x105
106	EC_NO_VIRUS_BASES	<i>No virus databases</i>	Error x106
107	EC_APP_TERMINATED	<i>Process terminated by signal</i>	Error x107
108	EC_APP_CRASHED	<i>Unexpected process termination</i>	Error x108



Error code	Symbolic representation	Internal message about the error	Description
109	EC_INCOMPATIBLE	<i>Incompatible software detected</i>	Error x109
113	EC_NO_KERNEL_MODULE	<i>Kernel module is not available</i>	Error x113
119	EC_NO_SCAN_ENGINE	<i>ScanEngine is not available</i>	Error x119
120	EC_NO_FILE_CHECK	<i>FileCheck is not available</i>	Error x120
121	EC_NO_ESAGENT	<i>ESAgent is not available</i>	Error x121
123	EC_NO_NET_CHECK	<i>NetCheck is not available</i>	Error x123
124	EC_NO_CLOUDD	<i>CloudD is not available</i>	Error x124
125	EC_UNEXPECTED_ERROR	<i>Unexpected error</i>	Error x125

Appendix G. List of Abbreviations

In this manual the following terms will be used without explanation hereinafter:

Convention	Complete form
<i>EPM</i>	ESP Package Manager (package manager)
<i>FQDN</i>	Fully Qualified Domain Name
<i>GID</i>	Group ID (system user group identifier)
<i>GNU</i>	GNU project (GNU is Not Unix)
<i>HTML</i>	HyperText Markup Language
<i>HTTP</i>	HyperText Transfer Protocol
<i>HTTPS</i>	HyperText Transfer Protocol Secure (via SSL/TLS)
<i>ID</i>	Identifier
<i>IP</i>	Internet Protocol
<i>LKM</i>	Linux Kernel Module
<i>MBR</i>	Master Boot Record
<i>NSS</i>	Novell Storage Services



Convention	Complete form
<i>OID</i>	(SNMP) Object ID
<i>PID</i>	Process ID (system process identifier)
<i>PAM</i>	Pluggable Authentication Modules
<i>RPM</i>	Red Hat Package Manager (package manager)
<i>RRA</i>	Round-Robin Archive
<i>RRD</i>	Round-Robin Database
<i>SMB</i>	Server Message Block (file access protocol)
<i>SNMP</i>	Simple Network Management Protocol
<i>SP</i>	Service Pack
<i>SSH</i>	Secure Shell
<i>SSL</i>	Secure Sockets Layer
<i>TCP</i>	Transmission Control Protocol
<i>TLS</i>	Transport Layer Security
<i>UID</i>	User ID (system user identifier)
<i>URL</i>	Unified Resource Locator
<i>VBR</i>	Volume Boot Record
<i>OS</i>	Operating System
<i>FS</i>	File System



Index

[

[SNMPD] section 201

A

Abbreviations 309

About the anti-virus 10

Activation 56

Appendices 249

appendix

computer threat types 249

fighting computer threats 253

B

Brief Instructions 65

Building the VFS SMB Module 140

C

Central protection 20

ClamD: configuration 154

CloudD: configuration 248

Command-line management 76

Components 12

computer threats 249

ConfigD: configuration 73

Configuration file 257

Configuration parameters 257

Configuring PARSEC 54

Configuring Security Subsystems 50

Configuring SELinux 51

Console uninstaller 39

Console-based installer 30

Conventions 8

Custom Installation of the Components 45

D

Dr.Web ClamD 152

Dr.Web CloudD 246

Dr.Web ConfigD 70

Dr.Web Ctl 76

Dr.Web ES Agent 187

Dr.Web File Checker 161

Dr.Web HTTPD 192

Dr.Web License 26

Dr.Web Network Checker 165

Dr.Web Scanning Engine 173

Dr.Web SNMP MIB 213

Dr.Web SNMPD 198

Dr.Web Updater 180

drweb-clamd 152

drweb-cloudd 246

drweb-configd 70

drweb-ctl 76

drweb-ctl usage examples 102

drweb-esagent 187

drweb-filecheck 161

drweb-httpd 192

drweb-netcheck 165

drweb-nss 143

drweb-se 173

drweb-smbspider-daemon 131

drweb-snmpd 198

drweb-spider 119

drweb-update 180

E

EICAR 59

ESAgent: configuration 189

F

fighting computer threats 253

File access rights 19

File permissions 19

FileCheck: configuration 163

Functions 10

G

General configuration settings 73

Generating certificates 262

Getting Started 56

H

How to change settings 65

How to Connect to the Central Protection Server 65

How to install a key 65

How to protect a server 65

How to upgrade the product 65

How to view log 65

How to view settings 65

HTTPD: configuration 194



Index

I

Installation from .run package 28
Installation from distribution 28
Installation from native packages 31
Installation from universal packages 28
Installing from the Repository 31
Installing the anti-virus 27
Installing the Product 27
Integration with clamd clients intended for ClamAV 159
Integration with NSS 63
Integration with Samba 61
Integration with SNMP Monitoring Systems 204
Introduction 9
Isolation 18

K

Key File 56
Known errors 264

L

License key file 56
Licensing 26
Linux file system monitor 119
Linux file system monitoring 60
LinuxSpider: configuration 122

M

Mobile mode 20
Modules 12
Monitoring systems 204
Monitoring via SNMP 204

N

NetCheck: configuration 168
Novell Storage Services (NSS) volumes monitor 143
NSS: configuration 145

O

Operating systems 23
Operation modes 20

P

Problems of SELinux 51
Product files 42
Product packages 42

Q

Quarantine 18
Quarantine directory 18

R

Registration 56
Removing the Anti-virus 27
Removing the Product 38

S

ScanEngine: configuration 177
Section [ClamD] 154
Section [CloudD] 248
Section [ESAgent] 189
Section [FileCheck] 163
Section [HTTPD] 194
Section [LinuxSpider] 122
Section [NetCheck] 168
Section [NSS] 145
Section [Root] 73
Section [ScanEngine] 177
Section [SMBSpider] 134
Section [Update] 182
SELinux Security 51
SMB shared directories monitor 131
SMBSpider: configuration 134
SNMPD: configuration 201
SpIDer Guard 119
SpIDer Guard for NSS 143
SpIDer Guard for SMB 131
SSL CA 262
SSL certificates 262
SSL private keys 262
Standalone mode 20
Starting command-line tool 77
Starting uninstaller 38
Structure of the product 12
System Requirements 23

T

Tasks 10
Technical support 256
Testing the Anti-virus 59



Index

U

- Uninstalling distribution 38
- Uninstalling from repository 39
- Uninstalling native packages 39
- Uninstalling the Product 38
- Update: configuration 182
- Upgrading components 34
- Upgrading the Product 34
- Upgrading to a Newer Version 35
- Use of LKM for SpIDer Guard 128

W

- Ways to install 27
- Web interface 105

