Kaspersky Anti-Virus 5.5 for Proxy Server

Administrator Guide

APPLICATION VERSION: 5.5 PLANNED UPDATE 3

Dear User!

Thank you for choosing our product. We hope that this documentation will help you in your work and will provide answers regarding this software product.

Revision date: 5/30/12

http://www.kaspersky.com
http://support.kaspersky.com

CONTENTS

INTRODUCTION

Kaspersky Anti-Virus 5.5 for Proxy Server provides anti-virus protection for network traffic routed through proxy servers which support the Internet Content Adaptation Protocol (ICAP).

The program allows:

- Perform anti-virus scans on objects transferred through the proxy server.

  Kaspersky Anti-Virus does not scan the data transferred via HTTPS.

- Cure infected objects, or block access to infected objects if disinfection fails.
- Use group settings to define filtration parameters that are applied depending on the address of the user requesting an object, and the object's address (URL).
- Log activity statistics, including information about anti-virus scanning and its results, and application errors and warnings.
- Notify administrators about detection of malicious software.
- Update the anti-virus databases. By default the application uses Kaspersky Lab's update servers as the source of updates. But it can be configured to update the databases from a local directory;

  The anti-virus databases are used in the detection and disinfection of infected objects. The application uses database records to analyze every object, checking it for virus presence: its content is compared with code typical for specific viruses.

  Please be aware that new viruses appear every day, and therefore you are advised to maintain the anti-virus databases in an up-to-date state. New updates are available hourly on Kaspersky Lab's update servers.

IN THIS SECTION

WHAT'S NEW

The current version of Kaspersky Anti-Virus has the following improvements:

- Support for Squid 3.0 or higher has been added.
- New configuration options are available for user groups. In particular, groups support now the parameters (see page 30) for selection of maximum scan duration and the set of Kaspersky Anti-Virus databases to use.
- Support for the ICAP **preview** feature has been added (see page 39), which reduces traffic and filtration time. Using **preview** decreases the volume of data transferred through the network, and accelerates the sorting of scanned objects.
- Option of viewing detailed information on the license by traffic is added (see page 21).

- Kaspersky Anti-Virus performance has been improved.

HARDWARE AND SOFTWARE SYSTEM REQUIREMENTS

In order for Kaspersky Anti-Virus to operate, the system must meet the following hardware and software requirements:

- Minimum hardware requirements:
  - processor Intel Pentium® II 400 MHz or higher;
  - 1 GB RAM;
  - 150 MB of disk space for Kaspersky Anti-Virus setup.
  - 200 MB of available disk space for temporary files.
- Software requirements:
  - for 32-bit platforms, one of the following operating systems:
    - Red Hat Enterprise Linux Server 6.2;
    - Fedora 16;
    - CentOS 5.7, 6.2;
    - SUSE Linux Enterprise Server 11 SP1;
    - Novell Open Enterprise Server 2 SP3;
    - openSUSE Linux 12.1;
    - Debian GNU/Linux 6.0.4 Squeeze;
    - Mandriva Enterprise Server 5.2;
    - Ubuntu 10.04, 12.04 LTS;
    - FreeBSD 8.2, 9.0;
  - for 64-bit platforms, one of the following operating systems:
    - Red Hat Enterprise Linux Server 6.2;
    - Fedora 16;
    - CentOS 5.7, 6.2;
    - SUSE Linux Enterprise Server 11 SP1;
    - Novell Open Enterprise Server 2 SP3;
    - openSUSE Linux 12.1;
    - Debian GNU/Linux 6.0.4 Squeeze;
    - Ubuntu 10.04, 12.04 LTS;
    - FreeBSD 8.2, 9.0.

- Squid 3.0 proxy server or higher with ICAP support.

  Kaspersky Anti-Virus integration with Squid 3.1.6 is not supported. For details see http://bugs.squid-cache.org/show_bug.cgi?id=3011.

- Glibc 2.2.x or higher (for Linux distributions).

- A Perl interpreter, version 5.0 or higher (for details see http://www.perl.org).

OBTAINING INFORMATION ABOUT ANTI-VIRUS

Kaspersky Lab provides various information sources about Anti-Virus. Select the source that suits you best depending on the importance and urgency of your question.

You can refer to the sources to research on your own or contact the Sales Department. If you already purchased the Kaspersky Anti-Virus, contact the Technical Support service. If the question does not require an urgent answer, you can discuss it with Kaspersky Lab's specialists and other Kaspersky software users in our web forum.

SOURCES OF INFORMATION TO RESEARCH ON YOUR OWN

You have the following sources of information on Kaspersky Anti-Virus at your disposal:

- Documentation.
- Manual pages.

Documentation

Administrator Guide contains the following information:

- on the purpose of Kaspersky Anti-Virus;
- on the hardware and software requirements for Kaspersky Anti-Virus installation and operation;
- on the installation of Kaspersky Anti-Virus;
- on managing Anti-Virus from the command line.

This document in PDF format is included into the Kaspersky Anti-Virus distribution kit. Alternatively you can download file from the Kaspersky Anti-Virus page of the Kaspersky Lab website.

Manual pages

To view information about Kaspersky Anti-Virus, you can refer to its manual pages, available after product installation in the */opt/kaspersky/kav4proxy/share/man/* directory.

CONTACTING THE TECHNICAL SUPPORT SERVICE

If you already purchased the Kaspersky Anti-Virus, you can obtain information about it from the Technical Support service by phone or via Internet.

Before contacting the Technical Support service please read the Support rules for Kaspersky Lab's products (http://support.kaspersky.com/support/rules).

Technical Support by e-mail

You can ask your question to the Technical Support Service specialists by filling out a Helpdesk web form at http://support.kaspersky.com/helpdesk.html.

You can ask your question in Russian, English, German, French or Spanish.

In order to send an e-mail message with your question, you must indicate the **client number** obtained from the Technical Support website during registration along with your **password**.

If you are not yet a registered user of Kaspersky Lab applications, you can fill out a registration form (https://support.kaspersky.com/ru/personalcabinet/Registration/Form/?LANG=en). Specify the key filename during the registration.

The Technical Support service will respond to your request in your Personal Cabinet (https://support.kaspersky.com/en/PersonalCabinet) and to the e-mail address you specified in your request.

Describe the problem you have encountered in the request web form providing as much detail as possible. Specify the following information in the mandatory fields:

- **Request type**. Select the topic which is the closest to the problem encountered, for example, "Product Installation/Removal Problem" or "Anti-Virus scan/virus removal problem".
- **Kaspersky Anti-Virus name and version number**.
- **Request text**. Describe the problem encountered in detail.
- **Client number and password**. Enter the client number and the password you have received during the registration at the Technical Support service website.
- **E-mail address**. The Technical Support service will send their answer to this e-mail address.

Technical support by phone

If you have a problem which requires urgent help, you can call your nearest Technical Support office. When you apply to Russian-speaking (http://support.kaspersky.ru/support/support_local) or international (http://support.kaspersky.ru/support/international) Technical Support specialists, please do not forget to provide Kaspersky Anti-Virus information (http://support.kaspersky.ru/support/details), it will facilitate timely assistance.

DISCUSSION OF KASPERSKY LAB'S APPLICATIONS IN WEB FORUM

If your question does not require an immediate answer, you can discuss it with the Kaspersky Lab experts and other users in our forum at http://forum.kaspersky.com.

In this forum you can view existing topics, add your comments, create new topics and use the search engine.

OPERATION ALGORITHM AND TYPICAL SCHEMES OF PROGRAM DEPLOYMENT

This chapter explains the application's functionality, its configuration and integration with an existing network structure.

IN THIS SECTION

HOW THE KASPERSKY ANTI-VIRUS WORKS

Kaspersky Anti-Virus does not scan the data transferred via HTTPS.

Kaspersky Anti-Virus performs anti-virus scanning of HTTP traffic using two modes of proxy operation: **REQMOD** and **RESPMOD**.

In the **RESPMOD** mode, the application checks objects requested by users via a proxy server. In the **REQMOD** mode it scans objects transmitted by users through the proxy: for instance, for a web-based mail server interface. Kaspersky Anti-Virus scans message attachments transferred by users to mail servers.

In the **RESPMOD** mode, the application uses this algorithm to scan internet traffic (see fig. 1):

1. The user requests an object through a proxy via HTTP.
2. If the requested object is available within the proxy cache, it will be returned to the user. If the object is not found in the cache, the proxy accesses a remote server and downloads the requested object from it.
3. The proxy uses ICAP to transfer the received object to Kaspersky Anti-Virus for an anti-virus scan.
4. Kaspersky Anti-Virus looks for a correspondence between the request parameters (user IP address, URL of the requested object) and its groups (see page 29). If it finds a correspondence, it scans and processes the object in accordance with the rules specified for that group. If a request does not match any of the existing groups, the application uses the default group rules for anti-virus scanning and processing.
5. The application assigns a specific status to a scanned object on the basis of the anti-virus scan results. Access to objects with a specific status is granted or blocked according to the processing group parameters (see page 29).

6. If access to an object has been granted, Kaspersky Anti-Virus allows the proxy to cache the object and transmit it to users. If access to an object is blocked, Kaspersky Anti-Virus prevents the proxy from caching the object or delivering it to users. Instead of receiving the requested object, the user will be notified that access to the object has been blocked.

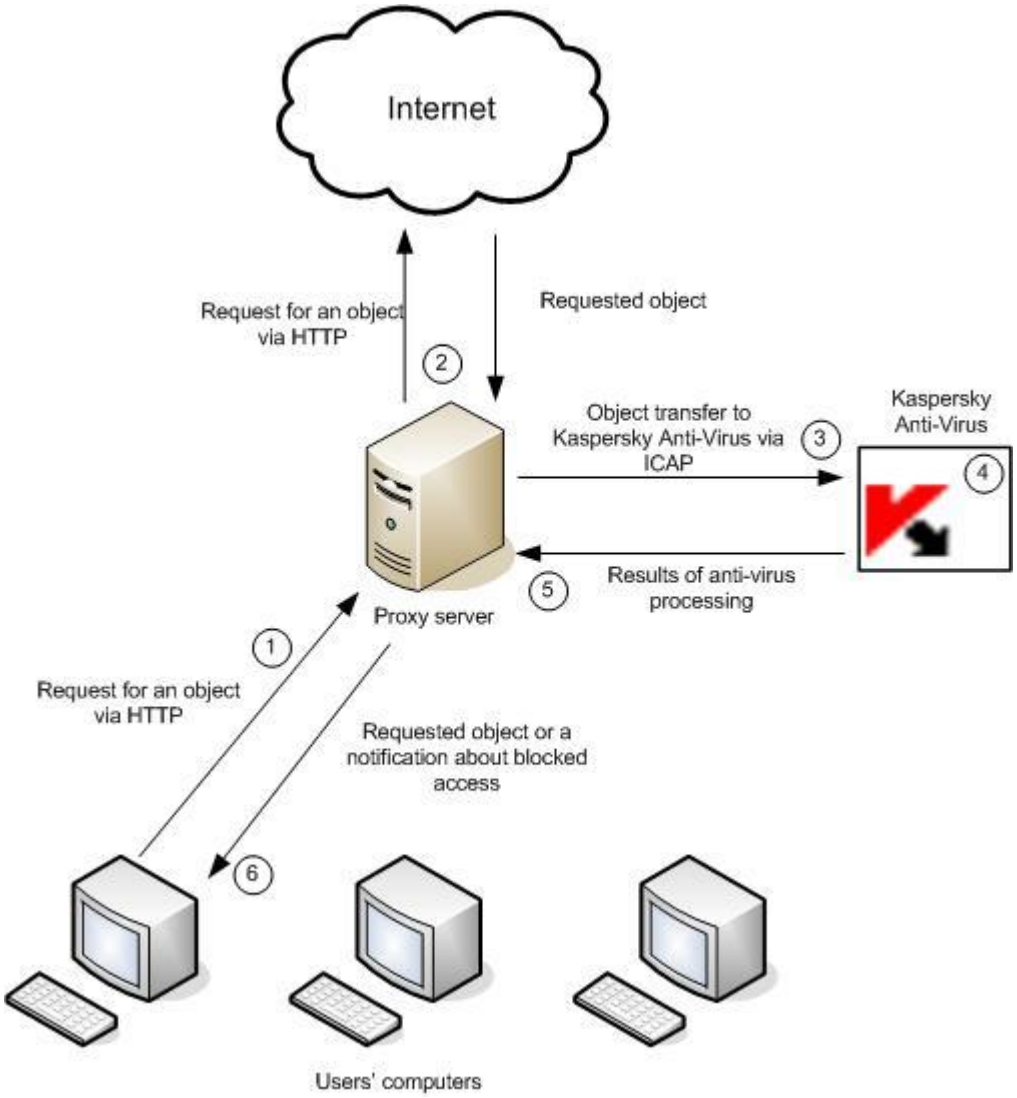

Figure 1. Anti-virus scanning of traffic in the RESPMOD mode

In the **REQMOD** mode, the application uses the following algorithm to scan internet traffic (see fig. 2):

1. The user sends an object using HTTP via a proxy.

2. The proxy uses ICAP to transfer the received object to Kaspersky Anti-Virus for an anti-virus scan.

3. Kaspersky Anti-Virus looks for a correspondence between the request parameters (user IP address, URL of the requested object) and its groups (see page 29). If it finds a correspondence, it scans and processes the object in accordance with the rules specified for that group. If a request does not match any of the existing groups, the application uses the default group rules for anti-virus scanning and processing.

4. After anti-virus check the product assigns a certain status to the scanned object; transfer of that object will be allowed or prohibited in accordance with the status. Access to objects with a specific status is granted or blocked according to the processing group parameters (see page 29).

5. If transfer is allowed, the proxy transmits the object sent by the user. If transfer is prohibited, the proxy does not transmit the object and instead notifies the user that the transfer has been blocked.

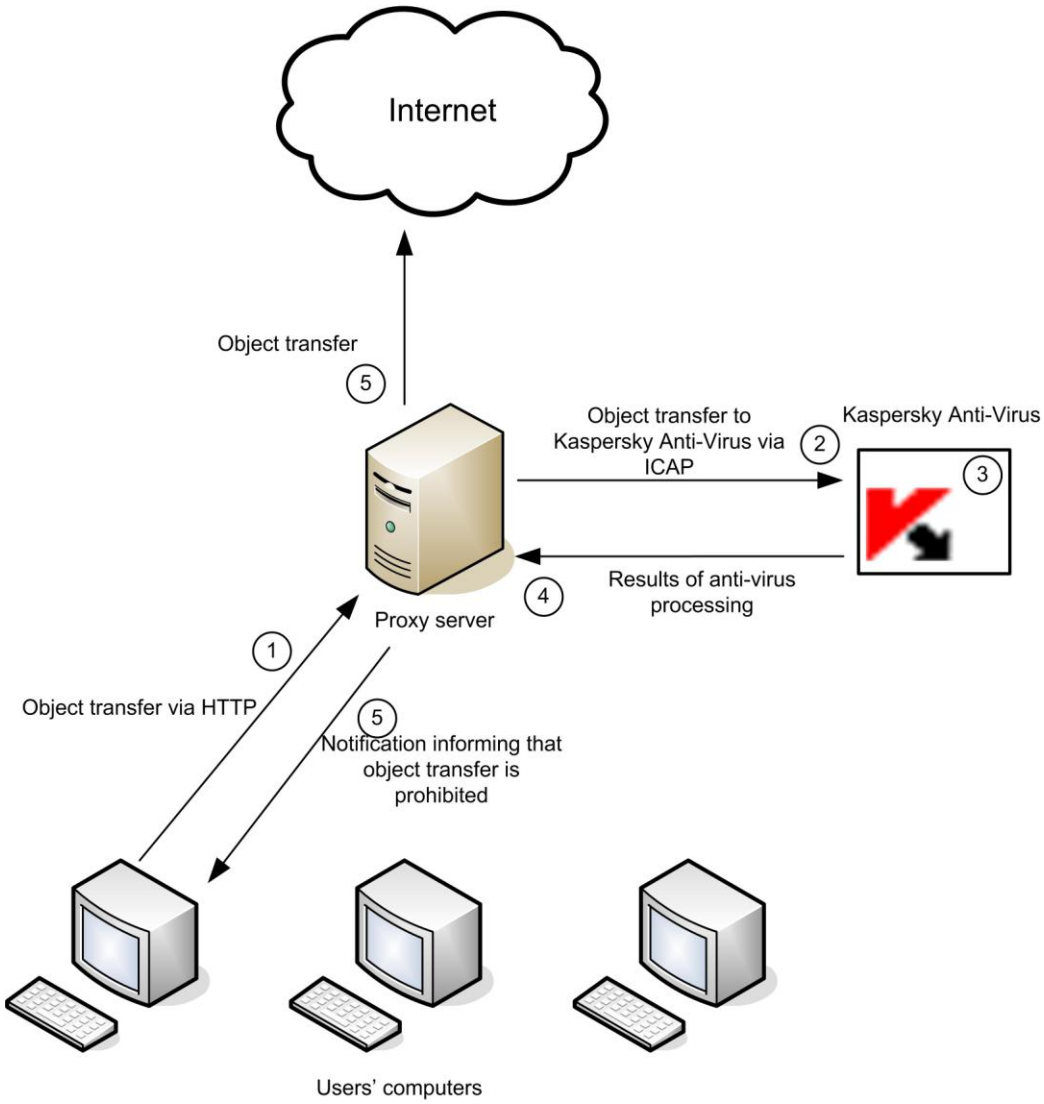

Figure 2. Anti-virus scanning of traffic in the REQMOD mode

ICAP REQUESTS PROCESSING ALGORITHM

During interaction with the proxy server, Kaspersky Anti-Virus acts as an ICAP server. The main ICAP server process controls child processes, which perform the following functions:

- receive and process requests from ICAP client (proxy server);
- interact with the anti-virus kernel: send requests for scanning and receive scan results;
- collect statistical information about scanning;
- transfer data from the anti-virus kernel to ICAP client.

Each child process starts several anti-virus kernels running as separate processes. Maximum number of anti-virus kernels that a single child process may use is defined by the **MaxEnginesPerChild** setting.

When the program starts, the main ICAP server process starts one child process. After start and until a request is intercepted, the child process remains in standby mode.

When ICAP client reports a connection, the child process intercepts that connection and switches to active mode. After that all requests arriving within that connection will be processed by that child process. When the child process completes processing of all requests, it switches back to standby mode.

If all child processes are active and their number does not exceed the **MaxChildren** value, the main ICAP server process spawns one more child process.

A child process handles requests until the number of processed requests reaches the **MaxReqsPerChild** value. After that the process stops receiving new connections from ICAP client, completes processing of all current requests and closes.

Forced child termination by the main process is another variant of its closing. That happens if the number of child process in standby state exceeds the **IdleChildren** setting value. Processes that have handled the maximum number of requests complete their work first in such case.

TYPICAL DEPLOYMENT SCENARIOS

This section contains descriptions of two main schemes used to deploy Kaspersky Anti-Virus:

- installation on the same server with the proxy;
- installation on a dedicated server.

General guidelines provided in the examples will help you configure Kaspersky Anti-Virus as your network structure may require.

INSTALLATION ON THE SAME SERVER WITH THE PROXY

Further in this document the operation and configuration of Kaspersky Anti-Virus will be described specifically for that scenario – on the same server with a proxy!

Installation on the same server with proxy helps achieve better processing performance because data between the proxy and Kaspersky Anti-Virus travel locally only involving no network transfers. This deployment scheme is efficient when the proxy server load is not too high. If a proxy is used to serve multiple user requests, installing the product on a dedicated server is recommended (see page 13) because anti-virus scanning and processing are quite resource-intensive procedures and thus they can negatively affect overall proxy performance.

The following configuration steps are performed automatically during product installation:

1. The installer configures Kaspersky Anti-Virus to start automatically at the OS startup and expect requests from the proxy on port 1344 via all network interfaces of the server.
2. The following lines will be added to the **ICAP OPTIONS** section of the proxy configuration file specified during program installation:

```
icap_enable on

icap_send_client_ip on

icap_service is_kav_resp respmod_precache 0 \

icap://localhost:1344/av/respmod

icap_service is_kav_req reqmod_precache 0 \

icap://localhost:1344/av/reqmod
```

```
icap_class ic_kav is_kav_req is_kav_resp

icap_access ic_kav allow all
```

- these settings make the proxy transfer all the requested objects to Kaspersky Anti-Virus via port 1344 of the local interface.

INSTALLATION ON A DEDICATED SERVER

Installing the program on a dedicated server is recommended when the proxy server is heavily loaded, and also when Kaspersky Anti-Virus is used to process the traffic from several proxy servers.

Since automatic configuration of the Anti-Virus and proxy is impossible in this deployment scenario, you will have to configure them manually.

CONFIGURING INTEGRATION WITH A SQUID PROXY

The following procedure is used to integrate Kaspersky Anti-Virus with a dedicated Squid server:

1. Once Kaspersky Anti-Virus is installed, use the **ListenAddress** parameter in the `[icapserver.network]` section of the kav4proxy.conf configuration file to specify the IP address of the network interface and the port that Kaspersky Anti-Virus will use to expect proxy requests for anti-virus scanning of accessed objects. By default, Kaspersky Anti-Virus expects requests at **localhost:1344**.

   Before changing the value of **ListenAddress** parameter, stop Kaspersky Anti-Virus Service using the following command:

   for Linux:

   ```
   # /etc/init.d/kav4proxy stop
   ```

   In FreeBSD:

   ```
   # /usr/local/etc/rc.d/kav4proxy stop
   ```

   Execute the following command to start the Kaspersky Anti-Virus service:

   for Linux:

   ```
   # /etc/init.d/kav4proxy start
   ```

   In FreeBSD:

   ```
   # /usr/local/etc/rc.d/kav4proxy start
   ```

2. Make the following changes in the proxy server configuration file:

   - for Squid 3.0:

     a. Add the following line to the `ACCESS CONTROLS` section:

     ```
     acl acl_kav_GET method GET
     ```

     b. Add the following lines to the `ICAP OPTIONS` section:

     ```
     icap_enable on
     icap_send_client_ip on
     icap_service is_kav_resp respmod_precache 0 \
     ```

```
icap://<ip_address>:<port>/av/respmod
icap_service is_kav_req reqmod_precache 0 \
icap://<ip_address>:<port>/av/reqmod
icap_class ic_kav_resp is_kav_resp
icap_class ic_kav_req is_kav_req
icap_access ic_kav_req allow all !acl_kav_GET
icap_access ic_kav_resp allow all
```

- for Squid 3.1:

```
icap_enable on
icap_send_client_ip on
icap_service is_kav_resp respmod_precache 0 \
icap://<ip_address>:<port>/av/respmod
icap_service is_kav_req reqmod_precache 0 \
icap://<ip_address>:<port>/av/reqmod
adaptation_access is_kav_req allow all
adaptation_access is_kav_resp allow all
```

`<ip_address>` stands here for the IP address of the server where Kaspersky Anti-Virus is installed; `<port>` is the port on which Kaspersky Anti-Virus expects the proxy requests for anti-virus scanning.

Kaspersky Anti-Virus integration with Squid 3.1.6 is not supported. For details see http://bugs.squid-cache.org/show_bug.cgi?id=3011.

3. Restart the proxy.

INSTALLATION OF THE APPLICATION

Before installing Kaspersky Anti-Virus, you are advised to:

1. Make sure that your system meets the hardware and software requirements (see page 6).
2. Log on to the system as **root**.

IN THIS SECTION

INSTALLATION ON A SERVER RUNNING LINUX

Kaspersky Anti-Virus for servers running the Linux operating system is distributed in two different installation packages:

- **.rpm** – for systems that support RPM Package Manager;
- **.deb** – for the distributions supporting a control system by packages dpkg.

To initiate installation of Kaspersky Anti-Virus from the rpm package, enter the following at the command line:

```
# rpm –i kav4proxy-<distribution package version>.i386.rpm
```

To initiate installation of Kaspersky Anti-Virus from the deb package, enter the following at the command line:

```
# dpkg –i kav4proxy-<distribution package version>.deb
```

To install Kaspersky Anti-Virus on a 64-bit operating system from the deb-package, execute the following command:

```
# dpkg –i --force-architecture kav4proxy-<distribution package version>.deb
```

During the setup process you will have to specify additional information (see page 16) regarding connection to the Internet, downloading of the anti-virus databases and settings for interaction with the proxy server.

INSTALLATION ON A SERVER RUNNING FREEBSD

The distribution file for installation of Kaspersky Anti-Virus on servers running the FreeBSD operating system is supplied as a .tgz package.

To initiate installation of Kaspersky Anti-Virus from the tgz package, enter the following at the command line:

```
# pkg_add kav4proxy-<distribution package version>.tgz
```

During the setup process you will have to specify additional information (see page 16) regarding connection to the Internet, downloading of the anti-virus databases and settings for interaction with the proxy server.

INSTALLATION PROCEDURE

Algorithms described in this section require that the target server has already Squid 3.0 proxy server or higher installed.

Kaspersky Anti-Virus integration with Squid 3.1.6 is not supported. For details see http://bugs.squid-cache.org/show_bug.cgi?id=3011.

Kaspersky Anti-Virus must be installed in two stages. The first stage will be performed automatically after execution of the commands described in Installation on a server running Linux (see page 15) and Installation on a server running FreeBSD (on page 15), and comprises the following steps:

1. The **klusers** group and the **kluser** account are created with the necessary privileges that Kaspersky Anti-Virus will use to start and operate.
2. Copying of the files from distribution package to computer.
3. Registration of the services necessary for Kaspersky Anti-Virus to function.

POST-INSTALLATION CONFIGURATION

Post-installation configuration is the second stage of installation, which includes configuration of the application and of the proxy server. To initiate the configuration process, use the postinstall.pl script located in the folder /opt/kaspersky/kav4proxy/lib/bin/setup/. When the script starts you will be asked to perform the following actions:

1. Specify the path to the license key file.
2. Configure the Internet proxy server using the following format:

   `http://<proxy server IP address>:<port>`

   or

   `http://<user_name>:<password>@<proxy server IP address>:<port>,`

   depending on whether the proxy requires authentication. The updater component (keepup2date) uses this setting to connect to Kaspersky Lab servers and download database updates.

   Set this option to **no** if you are not using a proxy server for connection to the Internet.
3. Download database updates from the servers of Kaspersky Lab. Enter **yes** or **no** depending on your intention to run the update procedure immediately. Once the updates are downloaded, you will see an offer to configure automatic updating. Automatic updates will be performed every hour by default.
4. Configure the product integration with Webmin.
5. Integrate Kaspersky Anti-Virus with proxy server. Specify one of the following values:
   - **No integration**. No integration will be performed then.
   - **Configure to work with remote proxy**. In that case you will be offered to enter the address of a remote proxy in the `<domain name|IP address>:<port>` format or **cancel** to cancel integration. The address suggested by default is 0.0.0.0:1344 (which means that the product will receive and send data using port 1344 of all network adapters).
   - **Configure Squid manually**. In that case you will be offered to perform the configuration procedure manually. Specify full path to the Squid configuration file, then the path to the Squid executable file. Then enter **yes** to confirm that the product should be integrated with the specified proxy server. To cancel integration, enter **no**.

- **Squid (<path to the squid.conf configuration file>)**. Then the post-install configuration script of Kaspersky Anti-Virus will perform the integration procedure automatically.

If you cancel proxy integration during this stage, you can run the /opt/kaspersky/kav4proxy/lib/bin/setup/proxy_setup.pl automatic integration script later.

Once the initial configuration procedure on a Linux server is complete, the installer starts the service of Kaspersky Anti-Virus. After that the service will be launched automatically when the operating system starts up.

A FreeBSD server requires starting the service of Kaspersky Anti-Virus and configuring its automatic launch manually.

To start the service of Kaspersky Anti-Virus and enable its automatic launch in FreeBSD, perform the following steps:

1. Add the `kav4proxy_enable="YES"` string to the /etc/rc.conf configuration file.

2. Execute the following command:

   ```
   /usr/local/etc/rc.d/kav4proxy.sh start
   ```

USING KASPERSKY ANTI-VIRUS

This chapter describes how to carry out tasks related to the basic features of Kaspersky Anti-Virus, including updating the application, management of license keys, anti-virus protection of HTTP traffic, and configuration of anti-virus scanning parameters for different user groups. The implementation of these tasks in a specific configuration will depend upon the particular organization of the network and the existing security policy.

IN THIS SECTION

UPDATING DATABASES

Kaspersky Anti-Virus uses the anti-virus databases while processing objects requested by users through the proxy server.

The anti-virus databases are employed while scanning for, and disinfecting, infected objects; they contain descriptions of all currently known viruses and the methods of disinfection for objects affected by those viruses.

The *keepup2date* component is included in the application to provide software updates. The updates are retrieved from the Kaspersky Lab's update servers, e.g.:

http://downloads1.kaspersky-labs.com/

http://downloads2.kaspersky-labs.com/

ftp://downloads1.kaspersky-labs.com/

and others.

The *updcfg.xml* file included in the installation package lists the URLs of all available update servers.

The keepup2date component supports basic authentication for connections through a proxy server.

To update the anti-virus databases, the keepup2date component selects an address from the list of update servers and tries to download updates from that server. If the first server is currently unavailable, the application attempts to connect to another server, and so on until updates are downloaded or the end of the list is reached.

Updates of anti-virus databases are available hourly on Kaspersky Lab's update servers.

After connection to an update server, keepup2date finds available updates for the product databases and downloads them.

You are strongly advised to set up the keepup2date component to update the databases every hour!

After a successful update the command, specified as the value of the **PostUpdateCmd** parameter in the `[updater.options]` section of the configuration file, will be executed. By default, this command automatically initiates the reloading of the anti-virus databases. Incorrectly modifying this parameter may prevent the application from using the updated databases, or cause it to function erroneously.

All settings of the keepup2date component are stored in the `[updater.*]` sections of the configuration file.

If your network has a complicated structure, and you are using multiple servers with Kaspersky Anti-Virus installed, you are advised to download updates from Kaspersky Lab's update servers every hour and place them in a network directory. To keep other networked servers constantly updated, configure them to copy the updates from that directory (see page 20).

The update task can be scheduled to run automatically using the **cron** (see page 19) or started manually from the command line (see page 19). Starting the keepup2date component requires **root** or **kluser** user privileges.

AUTOMATIC DATABASE UPDATES

You can schedule regular automatic updates for the databases using the **cron** service. You can configure cron either manually or using the keepup2date.sh script located in the /opt/kaspersky/kav4proxy/lib/bin/setup/ directory.

- *To create a cron task which updates the anti-virus databases hourly, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/lib/bin/setup/keepup2date.sh –install
```

- *To delete this cron task, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/lib/bin/setup/keepup2date.sh –uninstall
```

Example: configure the application to automatically update your anti-virus databases hourly. The system log should only record errors which occur in the component's operation. A general log should record all task starts. No information should be output to the console.

Solution: to perform the above task:

1. In the application configuration file, specify these parameter values:

```
[updater.report]

Append=true

ReportLevel=1
```

2. Edit the file that sets rules for the cron process (crontab -e) by adding the following line for the root or kluser user:

```
23 * * * * /opt/kaspersky/kav4proxy/bin/kav4proxy-keepup2date -q
```

The specified time setting for the cron task start is just an example. You are advised to specify your own settings for the start time to avoid overloading the updating servers.

MANUAL UPDATING OF THE DATABASES

You can start an update to your anti-virus databases from the command line at any time.

Example: start updating the databases, save the results of updating in the keepup2date.log file within the directory /var/log/kaspersky/kav4proxy/.

Solution: to accomplish the task, log in as root (or any other privileged user) and enter at the command line:

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-keepup2date -l \

/var/log/kaspersky/kav4proxy/keepup2date.log
```

If you need to update the databases on several servers, it is more convenient to download them once to a network directory and then mount the directory in the file system of each server running Kaspersky Anti-Virus instead of downloading the databases to each server individually. After that you will only need to run the updater script having specified first the mounted directory as the source of updates.

Example: initiate update of the databases using the local /home/kavuser/bases/ directory as the source. Output the results to the /tmp/updatesreport.log file.

Solution: to accomplish the task, log in as root (or any other privileged user) and enter at the command line:

1. Mount the shared directory containing updates to the anti-virus databases as the local /home/kavuser/bases/ directory.

2. Enter the following at the command line:

   ```
   # /opt/kaspersky/kav4proxy/bin/kav4proxy-keepup2date -l \

   /tmp/updatesreport.log -g /home/kavuser/bases
   ```

You can also update the application remotely using the appropriate Webmin plug-in.

CREATING A SHARED DIRECTORY FOR STORING AND SHARING DATABASE UPDATES

To correctly update the databases on local servers from a shared directory, that directory must have the same file system structure as Kaspersky Lab's update servers.

Example: create a shared local directory which local servers will use as the source of anti-virus database updates.

Solution: to accomplish the task, log in as root (or any other privileged user) and enter at the command line:

1. Create a local directory. The **kluser** account must have sufficient privileges to write to it.

2. Run the keepup2date component as follows:

   ```
   # /opt/kaspersky/kav4proxy/bin/kav4proxy-keepup2date -x <rdir>
   ```

   where `<rdir>` – full path to the created directory.

3. Provide reading access to that directory for local computers on your network.

LICENSES MANAGEMENT

The right to use Kaspersky Anti-Virus is determined by the license key. The key is included in the product's distribution kit, and entitles you to use the application as soon as you purchase it.

The application checks for the presence of an installed license key every time it starts or reloads its anti-virus databases.

If a license key is not installed, or an error has occurred while loading information about the current license, the application switches into a special unlicensed mode of operation. In that mode it does not perform anti-virus scanning of objects transferred through the proxy server; instead, all objects are treated using the action specified by the **LicenseErrorAction** parameter.

After the license expires, the functionality of the application will still be preserved except for the ability to update the anti-virus databases. You will still be able to perform anti-virus scanning and processing of objects, but you will be unable to use databases issued after the license expiration date. Therefore, you may not be protected against new viruses that have appeared after the license expired.

To protect your computer against new viruses, you are advised to renew the license.

A key file entitles you to use Kaspersky Anti-Virus and contains all the necessary information related to the license that you have purchased, such as the license type, its expiry date, information about distributors, etc.

In addition to the right to use the application during the license period, the license gives the following benefits:

- twenty-four-hour technical support;
- hourly updates of the anti-virus databases;
- timely notifications about new virus threats.

Therefore it is essential to extend your license to use Kaspersky Anti-Virus in a timely fashion. You can also install an additional key, which the application will start using as soon as the current active key expires.

VIEWING LICENSE INFORMATION

You can view information about installed license keys in the reports of the kavicapserver component. Each time the component starts kavicapserver loads the license key information and displays it in the report. The kavicapserver.log report file is stored in the /var/log/kaspersky /kav4proxy/ directory.

All information about license keys may be viewed either on the server's console, or remotely from any networked computer that has access to the Webmin module.

All information about license keys may be viewed either on the server's console, or remotely from any networked computer that has access to the Webmin module.

➧ *To view information about all installed license keys, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -s
```

In the server console, you will see information similar to the following:

```
Kaspersky license manager for Linux. Version 5.5.85/RELEASE #59

(C) 2012 Kaspersky Lab ZAO. All Rights Reserved.

Registered trademarks and service marks are the property of their respective owners.



License info:

Product name: Kaspersky Anti-Virus for xSP International Edition. 1000-1499 Mb of
traffic per day 1 year NFR Traffic Licence: Anti-Virus for Proxy Server

Invalid reason: Expired
```

```
Active key info:

Key file:        070C3064.key

Install date:    24-05-2012 UTC

Product name:    Kaspersky Anti-Virus for xSP International Edition. 1000-1499 Mb of
traffic per day 1 year NFR Traffic Licence: Anti-Virus for Proxy Server

Creation date:   03-11-2009 UTC

Expiration date: 03-11-2011 UTC

Serial:          0F92-0004AA-070C3064

Type:            Commercial

Count:           1024

Lifespan:        365

Objs:            3:1024
```

- *To view information about a license key, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -k 070C2FB1.key
```

where `070C2FB1.key` is the name of the license key file.

In the server console, you will see information similar to the following:

```
Kaspersky license manager for Linux. Version 5.5.85/RELEASE #59

(C) 2012 Kaspersky Lab ZAO. All Rights Reserved.

Registered trademarks and service marks are the property of their respective owners.

Product name:    Kaspersky Anti-Virus for xSP International Edition. 1000-1499 Mb of
traffic per day 1 year NFR Traffic Licence: Anti-Virus for Proxy Server

Creation date:   03-11-2009

Expiration date: 03-11-2011

Serial:          0F92-0004AA-070C2FB1

Type:            Commercial

Count:           250

Lifespan:        365

Objs:            3:250
```

- *To view the license status details, execute the following command:*

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -i
```

In the server console, you will see information depending on the licensing type. For example, in case of the traffic amount-based license:

```
Kaspersky license manager for Linux. Version 5.5.85/RELEASE #59

(C) 2012 Kaspersky Lab ZAO. All Rights Reserved.

Registered trademarks and service marks are the property of their respective owners.


Licensed traffic units: 250 (MB)

Traffic units used: 0 (MB)

Traffic units left: 250 (MB)
```

LICENSE RENEWAL

Renewing the Kaspersky Anti-Virus license will give you the right to re-enable full product functionality. Access to the additional services (see page 20) will be restored for the license duration.

The license term depends on the product you bought and the type of the license you purchased.

To renew the license for Kaspersky Anti-Virus:

- Contact the company that sold you the product, and renew your license for Kaspersky Anti-Virus.
- Purchase a license extension directly from Kaspersky Lab. Write a letter of request to our Sales Department at sales@kaspersky.com, or fill in the corresponding form on our website www.kaspersky.com in the section **E-Store** → **Renew Your License**. After your payment is received, we will send a license key to the email address indicated in the corresponding field of your license renewal form.

Kaspersky Lab regularly organizes promotional actions providing licenses for our products with considerable discounts. Please monitor the promotions announced at the web site of Kaspersky Lab (in the **Products** → **Special offers** section).

To install a new key file, enter the following at the command line:

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -a 00053E3D.key ,
```

where `00053E3D.key` is the name of the license key file.

If installation is successful, information similar to the following will be displayed on the server console:

```
Kaspersky license manager. Version 5.5.3/RELEASE

Copyright (C) Kaspersky Lab. 1997-2009.

Key file 00053E3D.key is successfully registered
```

We recommend that you update the anti-virus databases.

If you want to install a new license key before the current license key expires, you can add it as a backup license key. The backup key will be activated immediately the current one expires. The term of validity for the additional key starts from the activation date. You can install only one backup key.

If you have installed two keys (the current and an additional one), you can view information about both of them in the server console.

REMOVING A LICENSE KEY

- *To remove the current license key, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -da
```

If the component removes the license key successfully, information similar to the following will be displayed on the server console:

```
Kaspersky license manager. Version 5.5.3/RELEASE

Copyright (C) Kaspersky Lab. 1997-2009.

Active key was successfully removed
```

- *To remove the additional license key, enter the following at the command line:*

```
# /opt/kaspersky/kav4proxy/bin/kav4proxy-licensemanager -dr
```

If the component removes the license key successfully, information similar to the following will be displayed on the server console:

```
Kaspersky license manager. Version 5.5.3/RELEASE

Copyright (C) Kaspersky Lab. 1997-2009.

Additional key was successfully removed
```

USING A CONTROL SCRIPT

The kav4proxy application control script located in the /etc/init.d/ directory is used to start, stop and restart the application. The kav4proxy application control script uses the following command line parameters::

- **start** – command to check the configuration file and launch the application. If the program is running already, execution of the *kav4proxy* script will be stopped. If the program is not running, the script checks the configuration file and starts Kaspersky Anti-Virus. A return code of **0** indicates a successful start.
- **stop** – command to stop the application. Before stopping, the script checks whether the application is running (by the process ID). If the application is running, the script sends the SIGTERM signal. If the application does not stop within 30 seconds, the script sends the SIGKILL signal. A return code of **0** indicates a successful execution.
- **restart** – command to stop and restart the application, as provided by using the stop, and then start, keys.
- **reload** – command to reload the application configuration and the anti-virus databases using the SIGHUP signal.
- **reload_avbase** – command to reload only the anti-virus databases, and validate the license key.
- **stats** – command to write the results of statistics counters to a file (see page 35) and switch the report logging to another file (see page 36).

ENSURING ANTI-VIRUS PROTECTION OF HTTP TRAFFIC

Kaspersky Anti-Virus does not scan the data transferred via HTTPS.

Example: Provide anti-virus scanning of HTTP traffic transmitted by a proxy server installed on the same server as Kaspersky Anti-Virus, in accordance with the following requirements:

- General parameters of anti-virus scanning must be used for all requests;
- Disinfection of infected objects must be enabled.
- Scanning of e-mail databases must be disabled.
- Scanning of packed and archived objects must be enabled.
- Block access to infected, suspicious and damaged objects, and objects containing code that resembles a known virus.
- Use partial mode while processing proxy server requests.
- Disable anti-virus scanning of objects requested from the www.example.com web server.
- Store statistics on the results of anti-virus scanning in the /var/log/ kaspersky/kav4proxy/statistic file.

To accomplish the task, perform these steps:

1. Install Kaspersky Anti-Virus on the same server as the proxy server (see page 12) and perform its post-installation configuration (see page16).
2. Specify the following parameter values in the kav4proxy.conf configuration file (leave the values of parameters not mentioned in the example unchanged):

```
[icapserver.filter]

ExcludeURL=^www\.example\.com\/.*


[icapserver.engine.options]

ScanPacked=true

ScanArchives=true

ScanMailBases=false

ScanMailPlain=false

Cure=true


[icapserver.actions]

InfectedAction=deny

SuspiciousAction=deny

WarningAction=deny

ErrorAction=skip

ProtectedAction=skip

CorruptedAction=skip
```

```
[icapserver.protocol]

AnswerMode=partial


[icapserver.statistics]

AVStatisticsFile=/var/log/kaspersky/kav4proxy/statistic
```

3. Restart Kaspersky Anti-Virus using the following command:

```
# /etc/init.d/kav4proxy restart
```

CONFIGURING ANTI-VIRUS SCAN PARAMETERS FOR USER GROUPS

The example in section (see page 24) uses common settings for anti-virus processing of all user requests coming through the proxy server. Kaspersky Anti-Virus allows the definition of groups, to allow different parameters to be used for anti-virus protection of individual users.

Example: configure the application to perform anti-virus checks of HTTP traffic in accordance with the following requirements:

- These anti-virus scanning parameters must be specified for the **managers** group, which comprises computers using IP addresses on the 192.168.1.0/255.255.255.0 subnet:
  - Scanning of packed, archived files and e-mail databases must be disabled.
  - Disinfection of infected objects must be enabled.
  - Access should be granted to clean and disinfected objects only.
- These anti-virus scanning parameters must be specified for the **sales** group, which comprises computers using IP addresses on the 192.168.1.0/255.255.255.0 subnet:
  - Scan all objects.
  - Disinfection of infected objects must be enabled.
  - Block access to infected, suspicious and damaged objects, and objects containing code that resembles a known virus.
- These anti-virus scanning parameters must be specified for all other users:
  - Scanning of e-mail databases must be disabled.
  - Disinfection of infected objects must be disabled.
  - Access should only be granted to objects that have been assigned the *OK* status after a scan (see page 31).

➧ *To accomplish the task, perform these steps:*

1. In the kav4proxy.conf configuration file, create the following sections containing andfor a group **managers**:

```
[icapserver.groups:managers]

Priority=1

ClientIP=192.168.1.0/255.255.255.0

URL=.*


[icapserver.engine.options:managers]

ScanPacked=false

ScanArchives=false

ScanMailBases=false

ScanMailPlain=false

Cure=true


[icapserver.actions:managers]

InfectedAction=deny

SuspiciousAction=deny

WarningAction=deny

ErrorAction=deny

ProtectedAction=deny

CorruptedAction=deny
```

2. In the kav4proxy.conf configuration file, create the following sections containing anti-virus scanning parameters for the **sales** group:

```
[icapserver.groups:sales]

Priority=2

ClientIP=192.168.2.0/255.255.255.0

URL=.*


[icapserver.engine.options:sales]

ScanPacked=true

ScanArchives=true

ScanMailBases=true

ScanMailPlain=true

Cure=true
```

```
[icapserver.actions:sales]
InfectedAction=deny
SuspiciousAction=deny
WarningAction=deny
ErrorAction=skip
ProtectedAction=skip
CorruptedAction=deny
```

3. Specify the following parameter values for the default group:

```
[icapserver.engine.options]
ScanPacked=true
ScanArchives=true
ScanMailBases=false
ScanMailPlain=false
Cure=false

[icapserver.actions]
InfectedAction=deny
SuspiciousAction=deny
WarningAction=deny
ErrorAction=deny
ProtectedAction=deny
CorruptedAction=deny
```

4. Restart Kaspersky Anti-Virus using the following command:

```
# /etc/init.d/kav4proxy restart
```

DETAILED SETTINGS FOR KASPERSKY ANTI-VIRUS

This chapter contains a detailed explanation of basic parameters of Kaspersky Anti-Virus. Unlike the required settings essential for application functioning, which are specified during installation and post-installation configuration, additional configuration can be performed at the administrator's discretion. It is intended to extend the application's functionality, and its ability to enforce your corporate security policy.

IN THIS SECTION

CREATING GROUPS

The use of groups allows an administrator to specify different anti-virus processing for objects being requested or transferred through a proxy server by different user groups. A request is associated with a specific group depending on the IP address of the client computer requesting the object through a proxy server, and the URL of that object.

Ensure that the **icap_send_client_ip** parameter in Squid configuration file is set to **on**. This value means that Squid will transfer the client's IP address to Kaspersky Anti-Virus.

If a request's parameters do not match any existing group, the application will process the requested objects in accordance with the rules specified for the default group.

Each group's parameters are stored in the following five sections of the application's configuration file:

- `[icapserver.groups:<group name>]` – contains parameters that define the group applicability range (IP addresses of clients, object URLs) and the group's priority.
- `[icapserver.filter:<group name>]` – contains filtration rules for the <group name>;
- `[icapserver.engine.options:<group name>]` – contains anti-virus scanning parameters used to process objects associated with the group;

- `[icapserver.actions:<group name>]` – contains parameters that determine what actions are performed by the application on objects with a particular anti-virus scan status;

- `[icapserver.notify:<group name>]` – contains parameters that define the group applicability range (IP addresses of clients, object URLs) and the group's priority **deny**).

The default group parameters are specified in the `[icapserver.groups]`, `[icapserver.filter]`, `[icapserver.options]`, `[icapserver.actions]` and `[icapserver.notify]` sections.

You do not have to specify all group parameters while creating a new group. If some parameters are missing, the application uses the default settings.

Example: create **managers** the managers group to define rules for processing objects requested by client computers using the subnet 192.168.10.0/255.255.255.0. Prevent the group from accessing any objects that are not clean, disinfected and password-protected. Set the group priority to **2**. Use default values for all other parameters.

To accomplish the task, log in as the **root** (or any other privileged user) and create these sections in the kav4proxy.conf configuration file:

```
[icapserver.groups:managers]

Priority=2

ClientIP=192.168.10.0/255.255.255.0

URL=.*



[icapserver.engine.options:managers]

Cure=true



[icapserver.actions:managers]

ErrorAction=deny

ProtectedAction=skip
```

ANTI-VIRUS SCAN SETTINGS

The anti-virus engine parameters in the `[icapserver.engine.options:<group name>]` section define modes for scanning and disinfecting requested objects within a corresponding group, as follows:

- **ScanPacked=true|false** – enables/disables scanning of packed files. If the mode is disabled, all packed objects are considered to be clean.

- **ScanArchives=true|false** – enables/disables scanning of objects inside archives. If the mode is disabled, all archive files are considered to be clean.

- **ScanSFXArchives=true|false** – enables / disables the mode of scanning self-extracting archives (archives that contain an executable extraction module). If the mode is disabled, all self-extracting are considered to be clean.

- **ScanMailBases=true|false** – enables/disables scanning of email databases (either requested or transferred via a proxy server). If the mode is disabled, all email databases are considered to be clean.

- **ScanMailPlain=true|false** – enables/disables scanning of email databases in plain text format (requested or transferred through proxy server). If the mode is disabled, all email databases are considered to be clean.
- **UseAnalyzer=yes|no** – enables/disables heuristic analyzer used for anti-virus scanning.
- **HeuristicLevel=Recommended|Light|Deep|Medium** – sets the detail level for scan with the heuristic analyzer. The detail level provides the balance between the thoroughness of searches for new threats, the load on the operating system's resources and the time required for scanning. The higher the detail level, the more resources it will require and the longer it will take. Available values:
  - **Light** – least detailed scan, minimum system load;
  - **Medium** – medium scan, balanced system load;
  - **Deep** – most detailed scan, maximum system load;
  - **Recommended** – recommended value.
- **Cure=true|false** – enables/disables disinfection of infected objects. If the disinfection mode is disabled, the program makes no attempts to cure infected objects.
- **MaxScanTime** – maximum time to spend scanning a single object. If an object is not checked within the specified interval, it will be assigned the *ERROR* status.

CHOOSING ACTIONS FOR SCANNED OBJECTS

Actions performed by the application on scanned objects are defined by the status assigned to those objects following an anti-virus check.

Kaspersky Anti-Virus uses the following statuses:

- **OK** – clean object that has successfully passed the scanning procedure and is not infected;
- **INFECTED** – the object is infected; either it cannot be cured, or disinfection has not been attempted;
- **CURED** – the object was infected, but has been cured successfully;
- **WARNING** – the object contains code that resembles a known virus;
- **SUSPICIOUS** – the object is suspected of being infected with an unknown virus;
- **PROTECTED** – the object is password-protected and therefore cannot be scanned;
- **CORRUPTED** – the object is damaged;
- **ERROR** – object scanning resulted in an error.

Actions performed by Kaspersky Anti-Virus on objects with a specific status are determined by the parameters in the `[icapserver.actions]` section (for the default group) and `[icapserver.actions:<group name>]` section (for groups created by the administrator):

- **InfectedAction** – action taken on infected objects that have not been cured or cannot be cured.
- **SuspiciousAction** – action taken on objects suspected of being infected with an unknown virus.
- **WarningAction** – action taken on objects containing code that resembles a known virus.
- **ErrorAction** – action taken on objects that have been assigned the *ERROR* status.
- **ProtectedAction** – action taken on password-protected objects.

- **CorruptedAction** – action taken on damaged objects.
- **CuredAction** – action taken on disinfected objects.

The parameters defining these actions can take the following values:

- **skip** – allows object transfer;
- **deny** – prohibits object transfer, replacing the object with a corresponding notification file.

If **deny** is the action used on an object, then, depending upon the object's status, it will be replaced with one of the following placeholder files:

- **object_infected** – template containing a notification about detection of an infected object.
- **object_suspicious** – template containing a notification about detection of an object suspected of being infected with an unknown virus.
- **object_warning** – template containing a notification about detection of an object that resembles the code of a known virus.
- **object_protected** – template containing a notification about detection of a password-protected object.
- **object_error** – template containing a notification about detection of an object which caused a scanning error.
- **object_corrupted** – template containing a notification about detection of a damaged object.
- **object_cured** – template containing a notification about detection of an infected object that has been successfully cured.

Administrators can modify the text of these templates at their discretion, including addition of special macros (see page 51).

Example: Specify the following actions for scanned objects for the default group:

- allow transfer of the objects that have been assigned the *CURED* and *PROTECTED* status;
- prohibit transfer of all other objects.

Solution: to accomplish the task, log in as **root** (or any other privileged user) and specify the following parameter values in the `[icapserver.actions]` section:

```
[icapserver.actions]

CuredAction=skip

ProtectedAction=skip

InfectedAction=deny

SuspiciousAction=deny

WarningAction=deny

ErrorAction=deny

CorruptedAction=deny
```

ADMINISTRATOR NOTIFICATIONS

Every time the application performs the **deny** action on an object transferred through the proxy, it also runs a special script. Such script's example is located at: /opt/kaspersky/kav4proxy/share/examples/notify.sh. The **NotifyScript** parameter, in the `[icapserver.notify:<group name>]` section of the application configuration file, contains the script's filename.

Below you can examine a sample notification script and the steps necessary to configure the application to run the script.

Administrators can use SHELL syntax to create their own custom scripts, which will be executed every time the application blocks an object transfer via the proxy after scanning it. Every group created by the administrator can be assigned its own notification script (see page 29).

To configure the application to send notifications about blocked objects to admin@test.local, perform the following steps:

1. Create an executable script file with the following contents:

```
#!/bin/sh

recipients='admin@test.local'


action=%ACTION%

verdict=%VERDICT%


sendmail -t -i<<EOT

From: Kaspersky Anti-Virus For Linux Proxy Server <root@$HOSTNAME>

To: $recipients

Subject: $verdict object requested


Action applied:     $action

Verdict:      $verdict

Requested URL:      %URL%

Client IP:    %CLIENT_ADDR%


Found:

    Infected:  %VIRUS_LIST%

    Cured:     %CURED_LIST%

    Suspicious: %SUSP_LIST%

    Warnings:  %WARN_LIST%
```

```
This message generated by %PRODUCT% at %DATE% on $HOSTNAME

EOT
```

During script creation you can use special macros (see page 51), such as %URL%, %CLIENT_ADDR%, etc. to specify additional information.

2. Save the script file and make sure that the **kluser** user account has sufficient privileges for its execution.

3. Set the script filename as the value of the **NotifyScript** parameter. For instance, if the script has been saved as the file /usr/local/bin/notify.sh, and it should be executed whenever objects processed according to the default group rules are blocked, specify the following value for the **NotifyScript** parameter in the `[icapserver.notify]` section:

```
[icapserver.notify]

NotifyScript=/usr/local/bin/notify.sh
```

The application installation contains notification templates, which can be used when creating scripts. By default, these templates are located at /opt/kaspersky/kav4proxy/share/notify/.

OPERATION MODES

Depending on the status of the license (see the "License management section on page 20) and the databases, Kaspersky Anti-Virus can run in one of the following modes:

- **Basic mode** – fully functional mode of Kaspersky Anti-Virus operation. In this mode the application performs anti-virus scanning of proxy traffic, and disinfection of infected objects (if enabled).

- **Operation without updates** – the mode used by the application when the current license expires. In this mode the application performs anti-virus scanning of proxy traffic and, if enabled, disinfection of infected objects using the anti-virus databases current at the moment of license expiry.

- **Unlicensed operation** – the mode used by the application when the license key is not installed, or when an error has occurred while loading the information about the current license. In this situation the application does not perform anti-virus scanning of proxy traffic, and applies to all objects the action defined by the **LicenseErrorAction** parameter.

- **Operation without the anti-virus databases** – the mode used by the application if its anti-virus databases are not installed or if an error has occurred while loading them. In this mode the application does not perform anti-virus scanning of proxy traffic, and applies to all objects the action defined by the **BasesErrorAction** parameter.

MODES OF INTERACTION WITH PROXY VIA ICAP

The mode used by Kaspersky Anti-Virus to work with a proxy server is defined by the **AnswerMode** parameter in the `[icapserver.protocol]` section of the kav4proxy.conf configuration file, which can take the following values:

- **partial** – in this mode, Kaspersky Anti-Virus sends parts of the object being scanned to the proxy server, with the frequency determined by the **MaxSendDelayTime** parameter for their further transfer to the user. The last part of an object will only be sent to the user when the anti-virus scan of the object is complete, and only if the resulting status does not mean that the **deny** action should be applied to that object. If the **deny** action is applied to the object, the application does not send a template-based file to the user (see page 31), instead the application initiates disconnection..

This mode is convenient when large files are downloaded. In this case, users begin receiving objects before completion of an anti-virus check: otherwise, a user may terminate connection before he/she receives a response because of a long waiting period.

- **complete** – in this mode, Kaspersky Anti-Virus returns an object to the proxy server only after it is downloaded and tested completely, and provided that its resultant status does not require the **deny** action. If the **deny** action is applied to the object because of its status, the application will return a template-based file to the user, instead of the requested object (see page 31).

When **complete** mode is used, after clicking on an object in the browser window, the user will not see a dialog box allowing him/her to save the object or cancel scanning until the object is completely downloaded by the proxy server and scanned by Kaspersky Anti-Virus. The download can only be cancelled by closing the browser window, thus terminating the connection.

LOGGING APPLICATION STATISTICS

Kaspersky Anti-Virus provides two types of statistical information for administrators:

- statistics on the results of anti-virus scanning and processing;
- general statistics on the application's activity.

Statistics of anti-virus processing can be written to a local file or to a network socket. To log statistics to a local file, specify the path to this file as the value for the **AVStatisticsFile** parameter. The **AVStatisticsAddress** parameter is intended to specify a network socket.

Every line in the resulting statistics file will contain information about a single tested object, in the following format:

`<LEN><tab><RESULT><tab><METHOD><tab><ICAP_CLIENT_IP><tab>`
`<HTTP_USER_NAME><tab><HTTP_USER_IP><tab><URL>`, where `<tab>` stands for the tabulation character.

The values for all parameters are summarized in the table below.

Table 1. Statistics parameters

SYMBOLIC NAME	VALUE
<LEN>	Request size, bytes.
<RESULT>	The result of anti-virus scan.
<METHOD>	The mode of ICAP request processing (RESPMOD or REQMOD).
<ICAP_CLIENT_IP>	IP address of the ICAP client that has requested an object.
<HTTP_USER_NAME>	Name of the HTTP user that has requested an object.
<HTTP_USER_IP>	IP address of the HTTP user that has requested an object.
<URL>	Requested object URL.

If there are some reasons preventing output of a report on a processed object, information about that object will not be logged.

Besides the anti-virus scanning statistics, Kaspersky Anti-Virus also uses special counters providing statistical information about Kaspersky Anti-Virus activity. Logging of counter values into a file is regulated by the **CounterStatisticsFile** option of the program configuration file. The resulting file will contain a log of values returned by counters, as described in the table below.

Table 2. Counters of Kaspersky Anti-Virus activity

COUNTER	DESCRIPTION
Total_requests	Total number of processed scan requests.
Infected_requests	The number of requests which returned infected or suspicious objects, or objects resembling a known virus.
Protected_requests	The number of requests which returned protected objects.
Error_requests	The number of requests which returned objects causing processing errors.
Proccessed_traffic	The total volume of processed traffic, including service traffic (MB).
Clean_traffic	The total volume of clean traffic (MB).
Infected_traffic	The total volume of infected traffic (MB).
Traffic_per_min	Average MB per minute.
Request_per_min	Average number of ICAP requests processed per minute.
Engine_errors	Number of errors which occurred during the anti-virus kernel operation.
Total_connections	The number of active connections to the ICAP server.
Total_processes	The total number of running processes working on user requests.
Idle_processes	The number of idle processes waiting for requests.

APPLICATION REPORTING PARAMETERS

The results of operations performed by components of Kaspersky Anti-Virus are summarized in a log file in text format, specified by the **ReportFileName** parameter in the `[icapserver.report]`) section or in system log (**syslog**). If an empty string is set as the value of the **ReportFileName** parameter **ReportFileName=**, no information about application activity will be logged.

The amount of output information can be altered by changing the report detail level, set by the **ReportLevel** parameter in the `[icapserver.report]` section.

The level of detail is a number that sets the level of verboseness for information regarding the components' work. Each subsequent level includes information of the previous level together with some additional data.

Possible levels of report details are listed in the table below

Table 3. Levels of report details

LEVEL	LEVEL NAME	LEVEL LETTER SYMBOL	VALUE
0	Fatal Errors	F	Information about critical errors only (i.e. errors which cause program termination because some actions cannot be performed). For instance, virus infection of a component, or an error while initializing or loading databases and license keys.
1	Errors	E	Information about other errors which do not cause termination of components' activity; for example, information about an error encountered during file scanning.
2	Warning	W	Notifications about errors that may lead to the application shutdown (license key expiration warning, out-of-disk-space warning, etc.).
3	Info, Notice	I	Important informational messages, such as whether a component is running or inactive, the path to the configuration file, the scan scope, database updates, license keys, statistics summary.
4	Activity	A	Messages about scanning of files in accordance with the level of details defined for the report.
9	Debug	D	All debug messages.

Information about fatal errors is always displayed, regardless of the report detail level. The optimal level is level 4, which is also the default level.

Information messages may be subdivided into the following types:

- Messages pertaining to anti-virus checks.
- Messages pertaining to the operation of the application.

For example, information about the results of anti-virus scan for an object will be logged in the following format:

```
<DD-MM-YY HH:MM:SS> <REPORT_LEVEL> <METHOD> <ICAP_CLIENT_IP> <HTTP_USER_NAME>
<HTTP_USER_IP> <URL> <RESULT>
```

The values for all parameters are summarized in the table below.

Table 4. Logging settings

SYMBOLIC NAME	VALUE
<DD-MM-YY HH:MM:SS>	Date and time of record creation in the format defined by the **DateFormat** and **TimeFormat** settings.
<REPORT_LEVEL>	Letter indicating the amount of details in the report.
<METHOD>	The mode of ICAP request processing (RESPMOD or REQMOD).
<ICAP_CLIENT_IP>	IP address of the ICAP client that has requested an object.
<HTTP_USER_NAME>	Name of the HTTP user that has requested an object.
<HTTP_USER_IP>	IP address of the HTTP user that has requested an object.
<URL>	Requested object URL.
<RESULT>	The result of anti-virus scan.

CREATING A MEMORY DUMP TO DETECT ERRORS

Memory dump files or core files are created during an emergency shutdown of the application process; they can be used later by experts at Kaspersky Lab to identify the cause of problems in the operation of Kaspersky Anti-Virus. Memory dump files or core files are created during an emergency shutdown of the application process. The creation of core files is disabled by default.

To enable creation of memory dump files, specify the path /var/log/kaspersky/kav4proxy/core/ as the value of the **CorePath** parameter in the `[icapserver.path]` of the application configuration file.

Make sure that the partition where the /var/log/kaspersky/kav4proxy/core/ directory is located has sufficient free disk space for storage of core files.

In FreeBSD-based systems a modification of system kernel parameters may be necessary. To do that, enter the following command as the root user:

```
# sysctl -w kern.sugid_coredump=1
```

After that in the case of an emergency shutdown of the application, a file containing a dump of its memory will be created in the /var/log/kaspersky/kav4proxy/core/ directory.

For systems running FreeBSD, generation of dump files should be disabled once they are no longer necessary; all changes to the system kernel should also be reversed. To do that, enter the following at the command line:

```
# sysctl -w kern.sugid_coredump=0
```

WORK WITH INTERNET BROADCASTING STATIONS

Anti-virus scanning of the traffic generated by Internet broadcasting stations may interrupt the data stream or proxy operation. That complicates listening to the Internet radio broadcasts. In such cases you are advised to exclude such traffic from the scope of anti-virus scanning using the **ExcludeMimeType** parameter:

```
[icapserver.filter]


ExcludeMimeType=^audio/mpeg$

ExcludeMimeType=^application/vnd.ms.wms-hdr.asfv1$

ExcludeMimeType=^application/x-mms-framed$
```

These settings will exclude data streams in MPEG, ASF and Microsoft Windows Media formats from the scope of anti-virus scanning.

OPTIMIZING KASPERSKY ANTI-VIRUS

Kaspersky Anti-Virus can be optimized to reduce response time and traffic. Major reasons for reduced performance are:

- sending large amounts of data between Kaspersky Anti-Virus and the proxy server;
- scanning all objects, without distinction.

The Kaspersky Anti-Virus supports the **204 No Content** response. Using this feature helps to reduce traffic (see page 39).

Scanning all objects without distinction can be avoided by setting up exclusions (see page 39).

REDUCING TRAFFIC

In some cases, an object received from a proxy server is not modified by the application (for example, when the object is not infected). If the application is functioning in “complete” mode (see page 34), the entire object will be sent back to the proxy server.

If the application is interacting with a proxy server in “partial” mode (see page 34) and the checked object is small, the application can complete scan before the **MaxSendDelayTime** expires. In this situation also the entire object will be sent to the proxy server.

Use the **204 No Content** response to avoid unnecessary traffic. Assign the value **true** to the **Allow204** parameter in the `[icapserver.protocol]` section of the Kaspersky Anti-Virus configuration file. After that the **204 No Content** response is used instead of sending the entire object.

SETTING UP EXCLUSIONS

One way to improve Kaspersky Anti-Virus performance is to set up exclusions. There are three types of exclusion rules:

- exclusion by the object’s type;
- exclusion by the object’s URL;
- exclusion by the object’s size.

When excluding objects by their URLs, the application compares the object’s URL with the **ExcludeURL** parameter value in the `[icapserver.filter]` section of the application configuration file. If the comparison succeeds, no virus scan is performed and the **204 No Content** response is sent to the proxy server.

When excluding objects by the object type, the application analyzes the type of the received object (by the **Content-Type** value). If the Content-Type matches one of the **ExcludeMimeType** parameter values in the `[icapserver.filter]` section of the **kav4proxy.conf** file, no virus scan is performed and the **204 No Content** response is sent to the proxy server.

When excluding objects based on their size, the Kaspersky Anti-Virus checks the **Content-Length** field from the object’s HTTP header. If the object size exceeds the **MaxReqLength** parameter value from the `[icapserver.filter]` section of the kav4proxy.conf file, no virus scan is performed and the **204 No Content** response is sent to a client.

Enable the ICAP preview feature to use exclusions more effectively. The functionality allows Kaspersky Anti-Virus to receive just the object beginning instead of downloading the object completely. Kaspersky Anti-Virus can efficiently filter objects using HTTP headers in their initial part. If an object matches an existing filtration rule, Kaspersky Anti-Virus stops its download and returns the **204 No Content** response. That approach considerably decreases the traffic between proxy server and Kaspersky Anti-Virus improving its performance.

The size of the initial part of the received object is specified via the **PreviewSize** parameter in the `[icapserver.protocol]` section of the kav4proxy.conf file. The proxy server must be properly configured to enable Preview. For Squid proxy servers, the Preview feature is enabled via the icap_preview_enable parameter of the Squid configuration file.

UNINSTALLING THE APPLICATION

- *To uninstall Kaspersky Anti-Virus from a server running Linux will require one of these steps:*

  - To uninstall Kaspersky Anti-Virus installed from a .deb package, type the following at the command line:

    ```
    # rpm –e <distribution_package_name>
    ```

  - To uninstall the application installed from a .deb package, type the following at the command line:

    ```
    # dpkg –r <distribution_package_name>
    ```

- *To remove Kaspersky Anti-Virus from a server running FreeBSD, type the following at the command line:*

  ```
  # pkg_delete <distribution_package_name>
  ```

The procedure removing Kaspersky Anti-Virus runs automatically; it successively performs these operations:

1. Removes the cron task updating the anti-virus databases from the list of tasks for the **kluser** user.
2. Removes settings, made by the application in the Squid proxy server configuration file and restarts the proxy server.
3. Termination of application services.
4. Rolls-back the registration for automatic start-up of application services in the system.
5. Removes temporary files and directories created while Kaspersky Anti-Virus was running.
6. Removes application files: the procedure deletes all the Kaspersky Anti-Virus directories and files, including the anti-virus databases installed with the package. The only exceptions are reports, configuration files and the backup directory, which will not be deleted.

VALIDATING KASPERSKY ANTI-VIRUS SETTINGS

After Kaspersky Anti-Virus has been installed and configured, you can verify whether the application is configured correctly, using a test "virus" and its modifications.

IN THIS SECTION

TEST "VIRUS" EICAR AND ITS MODIFICATIONS

This test "virus" was specially developed by eicar (The European Institute for Computer Antivirus Research) for the testing of anti-virus products.

The test "virus" IS NOT A VIRUS, because it does not contain code that can harm your computer. However, most anti-virus products identify this file as a virus.

Never use real viruses for testing the operation of an anti-virus product!

You can download this test "virus" from the **EICAR**'s official website at http://www.eicar.org/anti_virus_test_file.htm.

Before you download the file, you must disable the computer's anti-virus protection, because otherwise the application would identify and process the file *anti_virus_test_file.htm* as an infected object transferred via the HTTP protocol.

Do not forget to enable the anti-virus protection immediately after you download the test "virus".

Do not forget to enable the anti-virus protection immediately after you download the test **EICAR** site as an infected object containing a virus that **cannot be disinfected** and performs the actions specified for this type of object.

You can also modify the standard test "virus" to verify the operation of the Kaspersky Anti-Virus. To modify the "virus", change the content of the standard "virus" by adding one of the prefixes to it (see table below). To modify test "virus", you can use any text or hypertext editor, such as **Microsoft Notepad**, **UltraEdit32**, etc.

You can test the correctness of the operation of the anti-virus application using the modified "virus" of EICAR only if your anti-virus bases were last updated on or after October 24, 2003 (October, 2003 cumulative updates).

In the table below, the first column contains the prefixes that must be added at the start of the standard test "virus" string. The second column lists all possible statuses that the Anti-Virus application can assign to the object, based on the results of the scan. The third column indicates how the application processes objects with the specified status. Please note that that actual actions performed on the objects are determined by the Kaspersky Anti-Virus settings.

After you have added a prefix to the test "virus", save the new file under a different name, for example: *eicar_dele.com*. Assign similar names to all modified "viruses".

Table 5. Modifications of the test "virus"

PREFIX	OBJECT STATUS	OBJECT PROCESSING INFORMATION
No prefix, standard test "virus".	**Infected**. Object contains code of a known virus. You cannot disinfect the object.	Kaspersky Anti-Virus recognizes such object as a virus that cannot be disinfected. An error occurs while attempting to disinfect the object; the action performed will be that specified for non-disinfectable objects.
CORR–	**Corrupted**.	Kaspersky Anti-Virus could access the object but could not scan it because it is corrupted (for example, the file structure is corrupted, or the file format is invalid).
WARN–	**Suspicious**. Object contains code of a known virus. You cannot disinfect the object.	The object has been found suspicious by the heuristic code analyzer. At the time of detection, the Anti-Virus threat signature databases contain no description of the procedure for treating this object.
	Suspicious. Object contains code of a known virus. You cannot disinfect the object.	Kaspersky Anti-Virus detected a partial correspondence of a section of object code with a section of code of a known virus. At the time of detection, the Anti-Virus threat signature databases contain no description of the procedure for treating this object.
ERRO–	**Scanning error**.	An error occurred during a scan of an object. Kaspersky Anti-Virus could not access the object, since the integrity of the object has been breached (for example, no end to a multivolume archive) or there is no connection to it (if the object is scanned on a network resource).
CURE–	**Infected**. Object contains code of a known virus. Disinfectable.	Object contains a virus that can be disinfected. Kaspersky Anti-Virus will disinfect the object; the text of the "virus" body will be replaced with the word CURE.
DELE–	**Infected**. Object contains code of a known virus. You cannot disinfect the object.	Kaspersky Anti-Virus recognizes such object as a virus that cannot be disinfected. An error occurs while attempting to disinfect the object; the action performed will be that specified for non-disinfectable objects.

TESTING THE ANTI-VIRUS SCANNING SETTINGS FOR HTTP TRAFFIC

The procedure for testing the settings of Kaspersky Anti-Virus described further requires installed wget utility.

In order to verify that the Kaspersky Anti-Virus configuration is correct:

1. Configure the anti-virus scanning settings (see page 24).
2. Specify the proxy server address in the wget configuration file (/etc/wgetrc (Linux), /usr/local/etc/wgetrc (FreeBSD)), for example:

   ```
   http_proxy = http://proxy.example.com:3128/
   ```

3. You can try to download this test "virus" from the **EICAR** official website at http://www.eicar.org/anti_virus_test_file.htm.

   As a result, file download will be blocked and the console will display the following or similar information:

```
$ wget http://www.eicar.org/download/eicar.com

--2010-01-13 11:38 AM:47--  http://www.eicar.org/download/eicar.com

Connecting to 172.16.0.1:8080... connected.

Proxy request sent, awaiting response... 403 Forbidden

2010-01-13 11:38 AM:47 ERROR 403: Forbidden.
```

KASPERSKY ANTI-VIRUS CONFIGURATION FILE

In this section gives a detailed description of the kav4proxy.conf file, which contains all settings for Kaspersky Anti-Virus. Immediately after installation, parameters are set to the application's default settings.

Table 6. Settings of Kaspersky Anti-Virus configuration file

SETTING	DESCRIPTION
The `[path]` section contains parameters defining paths to the directories essential for the Kaspersky Anti-Virus functioning:	
BasesPath=/var/opt/kaspersky/kav4proxy/bases	Full path to the directory where the anti-virus databases are stored.
LicensePath=/var/opt/kaspersky/kav4proxy/licenses	Full path to the directory where the license keys for the Kaspersky Anti-Virus are stored.
TempPath=/tmp	Full path to the directory where the Kaspersky Anti-Virus temporary files are stored.
KLPluginsPath=/opt/kaspersky/kav4proxy/lib/ppl	Full path to the directory where the Kaspersky Anti-Virus library files are stored.
The `[options]` section contains the parameters that define the user and the group used to run the Kaspersky Anti-Virus.	
User=kluser	Name of the user whose privileges the Kaspersky Anti-Virus uses to run.
Group=klusers	Name of the group whose privileges the Kaspersky Anti-Virus uses to run.
The `[locale]` section contains the parameters that define the date and time format in reports and Kaspersky Anti-Virus statistics.	
DateFormat=%d-%m-%Y	Date format in the application activity report.
TimeFormat=%H:%M:%S	Time format in the report.
The `[icapserver.network]` section contains network settings of the Kaspersky Anti-Virus.	

ListenAddress=localhost:1344	IP address and the port that Kaspersky Anti-Virus uses to wait for proxy requests sent via ICAP.
	Before changing the value of **ListenAddress** parameter, stop Kaspersky Anti-Virus Service using the following command:
	for Linux:
	`# /etc/init.d/kav4proxy stop`
	for FreeBSD:
	`# /usr/local/etc/rc.d/kav4proxy stop`
	Execute the following command to start the Kaspersky Anti-Virus service:
	for Linux:

SETTING	DESCRIPTION

	`# /etc/init.d/kav4proxy start`
	
	for FreeBSD:
	
	`# /usr/local/etc/rc.d/kav4proxy start`

Timeout=0	Network timeout for interaction via ICAP.
The `[icapserver.process]` section contains advanced settings for Kaspersky Anti-Virus processes:	
MaxChildren=3	The maximum number of running child processes working on requests sent via ICAP.
IdleChildren=1	The maximum number of running child processes waiting for requests sent via ICAP.
MaxReqsPerChild=0	The maximum number of requests to scan objects that a child process can serve. After processing the specified number of requests, a child process terminates and the application initiates a new child process.
MaxEnginesPerChild=2	The maximum number of scanning modules used simultaneously by child processes for anti-virus scanning of objects. A larger number of scanning modules allows faster anti-virus scanning of objects, at the expense of server's other performance (for instance, GUI interaction). Please take into account the hardware of your server when setting this parameter's value.
The `[icapserver.protocol]` section contains the settings for the interaction between Kaspersky Anti-Virus and the proxy server via ICAP.	
AnswerMode=partial|complete	The method of interaction with the proxy. The **partial** value means that Kaspersky Anti-Virus will allow transfer of the scanned object's parts to the client before the object is completely downloaded from the Internet and scanned. The **complete** value means that Kaspersky Anti-Virus will only allow transfer of a requested object to the client after it is downloaded completely and scanned. Default value: **partial**.
MaxSendDelayTime=10	Time interval (seconds) that determines the frequency used to send parts of a requested object to the client in **partial** mode.
ReqModeServiceUrl=av/reqmod	ICAP service URL for checking the stream of HTTP requests.
RespModeServiceUrl=av/respmod	ICAP service URL for checking the stream of HTTP responses.
PreviewSize=0	The size of preview request. If the parameter value is **0**, then the server refuses to receive preview requests.
MaxConnections=5000	The maximum number of connections allowed for the current ICAP server. This parameter's value is returned to the ICAP client via the OPTIONS method. If the parameter value is **0**, then the OPTIONS method does not return the number of connections.

SETTING	DESCRIPTION
Allow204	Allows/prohibits using of the standard ICAP response **204 No Content**. The default value is **true**.
HTTPClientIpICAPHeader=X-Client-IP	Name of the ICAP heading, which contains the IP address of the HTTP client.
HTTPUserNameICAPHeader=X-Client-Username	Name of the ICAP heading, which contains the HTTP client's user name.
SendAVScanResult=true|false	Notification mode for alerting about a detected threat. If the parameter value is **true**, the following information is added to the ICAP response: **X-Virus-ID** – name of detected threat **X-Response-Info** – request processing result (blocked, filtered, or passed). Default value: **false**.
The `[icapserver.statistics]` section contains the parameters pertaining to the generation of Kaspersky Anti-Virus statistics.	
CounterStatisticsFile	Path to the file where the values of statistics counters will be stored.
AVStatisticsFile	Path to the file where anti-virus scanning statistics will be stored.
AVStatisticsAddress	Network socket for logging anti-virus scanning statistics.
The `[icapserver.report]` section contains the parameters pertaining to report generation by the Kaspersky Anti-Virus.	
ReportFileName=/var/log/kaspersky/kav4proxy/kavicapserver.log	Filename for the report on Kaspersky Anti-Virus activity.
Buffered=true|false	Buffer mode for recording to the report file. To enable the mode, set **true** as the parameter value. Default value: **false**.
ReportLevel=0|1|2|3|4|9	Level of details in report. Default value: **4**.
ShowOk=true|false	The logging mode for information about objects where scanning revealed no malicious code. Default value: **true**.
Append=true|false	Report generation mode in which the report is created anew each time the Kaspersky Anti-Virus starts. If you wish to add new information to an existing report instead of overwriting it, set the parameter value to **true**. Default value: **true**.
AVReportFileName=/var/log/kaspersky/kav4proxy/av_server_log	Filename for the report on Kaspersky Anti-Virus engine.
AVReportLevel=0|1|2|3|4|9	Level of details in Anti-Virus report.
The `[icapserver.path]` section contains parameters that define paths to specific application files.	
PidFile=/var/run/kavicapserver.pid	Path to the Kaspersky Anti-Virus PID file.
CorePath	Path to the directory where dump files will be saved that are created if Kaspersky Anti-Virus processes crash. In order to enable the option of creating dump files, specify the value **/var/log/kaspersky/kav4proxy/core/**. By default, the value of this parameter is not defined (creation of dump files is disabled).
Sections described above contain the anti-virus processing parameters for the default group (see page 29).	

SETTING	DESCRIPTION
The `[icapserver.groups]` section contains parameters that define the paths to special application files.	
Priority	Group priority. If a request's parameters match several groups, the processing will use the rules of the group with the highest priority. Default value: **0** (the highest priority).
ClientIP	IP address of the client that has requested an object through the proxy server. Objects requested from a specified IP address and located at an address defined by the URL parameter will be processed using the rules of this group. Default value: **.***.
URL	URL of a requested object. Objects with a specified URL and requested from an IP address defined by the**ClientIP** parameter will be processed using the rules of this group. Default value: **.***.
The `[icapserver.filter]` section contains filtration parameters for the default group.	
ExcludeMimeType	Exception mask for filtering by MIME type (regular expressions can be used). The Kaspersky Anti-Virus will not perform anti-virus scanning of objects with a MIME type which matches the specified mask.
ExcludeURL	Exception mask for filtering by URL type (regular expressions can be used). The Kaspersky Anti-Virus will not perform anti-virus scanning of objects from an URL which matches the specified mask.
MaxReqLength=0	Maximum size of the objects to be scanned, bytes.
The `[icapserver.engine.options]` section contains the anti-virus scanning parameters for the default group	
ScanPacked=true|false	Instruction to scan packed files To disable the mode, set **false** as the parameter value. Default value: **true**.
ScanArchives=true|false	Instruction to check archived objects. To disable the mode, set **false** as the parameter value. Default value: **true**.
ScanSFXArchives=true|false	Instruction to check SFX archives. To disable the mode, set **false** as the parameter value. Default value: **true**.
ScanMailBases=true|false	Instruction to scan email databases (requested or transferred through the proxy server). To disable the mode, set **false** as the parameter value. Default value: **true**.
ScanMailPlain=true|false	Instruction to scan databases of email messages in plain text format (requested or transferred through proxy server). To disable the mode, set **false** as the parameter value. Default value: **true**.
UseAnalyzer=yes|no	Enables/disables heuristic analyzer used for anti-virus scanning. The Heuristic Analyzer scans the standard sequence of operations allowing the nature of the file to be determined with a reasonable degree of certainty. The advantage of using this method is that new threats are detected before virus analysts have encountered them. To disable the mode, set **no** as the

SETTING	DESCRIPTION
	parameter value. Default value: **yes**.
HeuristicLevel=Recommended|Light|Deep|Medium	The level of detail of the heuristic analysis. The detail level provides the balance between the thoroughness of searches for new threats, the load on the operating system's resources and the time required for scanning. The higher the detail level, the more resources it will require and the longer it will take. Available values: • **Light** – least detailed scan, minimum system load; • **Medium** – medium scan, balanced system load; • **Deep** – most detailed scan, maximum system load; • **Recommended** – recommended value. Default value: **Recommended**.
Cure=true|false	Instruction to cure infected objects. To disable the mode, set **false** as the parameter value. Default value: **false**.
MaxScanTime	Maximum time to spend scanning a single object, seconds. If an object is not checked within the specified interval, it will be assigned the *ERROR* status. Default value: **300**.
MaxNestingLevel=8	Max nesting level. Kaspersky Anti-Virus doesn't scan object with nested level more than this parameter's value.
The `[icapserver.actions]` section contains the settings specifying actions to be taken on scanned objects for the default group.	
CuredAction=skip|deny	Action on disinfected objects. Default value: **skip**.
InfectedAction=skip|deny	Action on infected objects. Default value: **deny**.
SuspiciousAction=skip|deny	Action on suspicious objects. Default value: **deny**.
WarningAction=skip|deny	Action on an object resembling a known virus. Default value: **deny**.
ErrorAction=skip|deny	Action on an object which has caused a scanning error. Default value: **skip**.
ProtectedAction=skip|deny	Action on password-protected objects. Default value: **skip**.
CorruptedAction=skip|deny	Action on damaged objects. Default value: **skip**.
LicenseErrorAction=skip|deny	Action on scanned objects if the Kaspersky Anti-Virus has failed to load the license key information. Default value: **skip**.
BasesErrorAction=skip|deny	Action performed on objects if the Kaspersky Anti-Virus fails to load the anti-virus databases. Default value: **deny**.
MaxReqLengthAction=skip|deny	Action with the object, which size exceeds the set maximum size (**MaxReqLength**). Default value: **skip**.

SETTING	DESCRIPTION
PartialResponseAction=check|deny	Action, when a server response containing a part of the object is detected. Set **check** to allow message checking. Set **deny** to deny response transfer and report an access error. Default value: **check**.
PartialRequestAction=check|deny|reset	Action, when a request for delivery of an object part is detected. Set **check** to allow checking of the request. Set **deny** to deny the request and report an access error. Set **reset** to delete information on the request of the object part from the request and to request the whole object. Default value: **check**.
The `[icapserver.filter]` section contains filtration parameters for the default group.	
NotifyTemplateDir	Directory where notification templates are stored.
NotifyScript	Script used by the application to notify the administrator about objects prohibited for transfer through the proxy server.
The `[updater.path]` section contains the paths of directories and files necessary for the functioning of the **keepup2date** component.	
BackUpPath	Full path to the directory where the anti-virus databases are stored. Default value: **/var/opt/kaspersky/kav4proxy/bases.backup**.
AVBasesTestPath	Full path to the avbasestest utility, which validates the anti-virus databases Default value: **/opt/kaspersky/kav4proxy/lib/bin/avbasestest**.
The `[updater.options]` section contains parameters used by the **keepup2date** component.	
KeepSilent=true|false	The mode which determines whether component messages are output to the console **keepup2date**. When the parameter is set to **true** the component does not output reports to the console. Default value: **false**.
ProxyAddress	Address of the proxy server used for connection. This parameter is defined as `http://username:password@url:port`. The Username and/or password parameters may be missing from the proxy address. If no address is specified, its value will be taken from the http_proxy environment variable.
UseProxy=true|false	The mode of proxy use during connection with Kaspersky Lab's update servers. When set to **false**, the proxy server will not be used. If the parameter is set to **true**, the component will use the proxy address defined by the **ProxyAddress** parameter. If the **ProxyAddress** parameter value is undefined, then the http_proxy environment variable will be used. If the environment variable is not defined, a proxy server will not be used. Default value: **false**.
UseUpdateServerUrl=true|false	Parameter which defines whether the updater will use the address defined by the **UpdateServerUrl** parameter. Default value: **false**.
UseUpdateServerUrlOnly=true|false	Parameter which defines whether the updater will use only the address defined by the **Update-ServerUrl** parameter. When set to **false**, a failed attempt to update databases using the

SETTING	DESCRIPTION
	UpdateServerUrl address as the source will be followed by an attempt to use another address from the list of update servers. Default value: **false**.
UpdateServerUrl=http://url/ **|** ftp://url/ **| /local_path/**	Address of the update source.
PostUpdateCmd=/etc/init.d/kav4proxy reload_avbase	Command performed immediately after an update of the anti-virus databases. The default value forces the application to reload the updated anti-virus databases automatically. Modification of this parameter is not recommended.
RegionSettings=Russia	The region where the user is located. It is used to select the most convenient Kaspersky Lab's update server from which to download updates to the anti-virus databases.
ConnectTimeout=30	Timeout (seconds) for network operations during updates of the anti-virus databases. If no data arrive within the specified interval while downloading database updates, the updater selects another server from the list of Kaspersky Lab update servers.
PassiveFtp=true|false	The parameter determines the use of passive FTP mode. Default value: **false**.
The `[updater.report]` section contains settings for output of reports by the **keepup2date** component.	
Append=true|false	Instruction to append a report to the end of an existing report file. When the parameter is set to **false** the component does not output reports to the console. Default value: **true**.
ReportFileName	Name of the file used for logging reports about the component's activity.
ReportLevel=0|1|2|3|4|9	Level of details in report. Default value: **4**.

MACROS

Kaspersky Anti-Virus allows the use of special macros, both in the template-based placeholder files which are sent to users instead of blocked objects (see page 31), and in the text of notification scripts (the **NotifyScript** parameter). Description of these macros contains in the table below.

Table 7. Macros

Macro syntax	Description
%VIRUS_LIST%	List of viruses that an object is infected with.
%WARN_LIST%	List of objects containing code that resembles a known virus.
%SUSP_LIST%	List of objects suspected of infection with an unknown virus.
%CURED_LIST%	List of removed viruses.
%CLIENT_ADDR%	IP address of the client computer that has requested an object.
%URL%	Requested object URL.
%ACTION%	Action performed on an object.
%VERDICT%	Object status
%PRODUCT%	Product description.
%DATE%	Time of message creation.

KAVICAPSERVER RETURN CODES

Table 8. kavicapserver return codes

RETURN CODE	VALUE
0	No errors detected at component start.
30	Fatal system error.
65	Error loading the configuration file (file not found).
66	Error in the configuration file or command line parameters.
70	The component executable file is corrupted.

COMMAND LINE OPTIONS FOR LICENSEMANAGER

Table 9. Command line options for licensemanager

HELP OPTIONS	
-h	Display on the console reference information about the component's command line options, and exit.
COMMAND LINE OPTIONS FOR MANAGING LICENSE KEYS	
-s	Display on the console information about all installed license keys.
-c(C) <path_to_file>	Use the alternative configuration file <path_to_file>.
-i	Display on the console information about the license key.
-k <path_to_file>	Display on the console the detailed information about license parameter.
-a <path_to_file>	Install a license key.
-d <a|r>	Delete the current/additional key.

LICENSEMANAGER RETURN CODES

Table 10. Licensemanager return codes

RETURN CODE	VALUE
0	The component has successfully completed its operation.
30	Fatal system error.
64	Licensing error
65	Error loading the configuration file (file not found).
66	Error in the configuration file or command line parameters.
70	The component executable file is corrupted.

COMMAND LINE FOR OPTIONS FOR KEEPUP2DATE

Table 11. Command line for options for keepup2date

HELP OPTIONS	
-h	Display on the console reference information about the component's command line options, and exit.
-v	Display the application version on the console and exit.
-s	Display a list of update servers with information about their respective regions.
UPDATE OPTIONS	
-c <path_to_file>	Use the alternative configuration file <path_to_file>.
-u <directory>	Copy the application update to the local <directory>. Within the specified directory, the utility will reproduce the internal structure of an update server, enabling local computers to update from that directory.
-b <path>	When updating, create in the <path> directory a backup copy of the anti-virus databases being updated.
-t <path>	Use the <path> directory to store temporary files.
-r	Cancel the last update. Updated databases will be replaced by their previous versions.
-k	Disable execution of the command defined by the PostUpdateCmd parameter.
-d <path_to_file>	Use the specified PID file.
-g <url>	Use the server with the specified URL as the source of updates.
-q	Disabling the output of information about component operation.
-e	Display only information about critical errors.
REPORT GENERATION OPTIONS	
-l <path_to_file>	Log work results in file <path_to_file>.

KEEPUP2DATE RETURN CODES

Table 12. Keepup2date return codes

RETURN CODE	VALUE
0	The anti-virus databases do not need an update.
1	The anti-virus databases were updated successfully.
10	A fatal error occurred; updating was interrupted.
12	An error while rolling back to the previous version of the anti-virus databases. Rollback has been interrupted.
30	The **PostUpdateCmd** command could not be executed after the databases were updated.
60	License information is missing, or no license key was found, using the path specified in the configuration file.
75	The configuration file cannot be loaded or contains errors.
128 + signal code	The application has exited upon a signal with the corresponding code.

LOCATIONS OF KASPERSKY ANTI-VIRUS FILES

In further examples we shall use names of the components that are installed on a server running Linux!

After Kaspersky Anti-Virus installation on a server running Linux, the program files will be located as follows (provided that the default paths are accepted):

/etc/opt/kaspersky/kav4proxy.conf – configuration file containing Kaspersky Anti-Virus parameters;

/opt/kaspersky/kav4proxy/bin/ – directory containing executable files of the application components:

kav4proxy-keepup2date – updater utility for the databases of Kaspersky Anti-Virus;

kav4proxy-licensemanager – utility for license keys management.

/opt/kaspersky/kav4proxy/lib/bin/avbasestest – utility validating the downloaded databases for the keepup2date component.

/etc/init.d/kav4proxy – Kaspersky Anti-Virus management script.

/opt/kaspersky/kav4proxy/lib/bin/setup/ – directory containing scripts for post-installation setup and removal of the Kaspersky Anti-Virus:

postinstall.pl – post-installation Kaspersky Anti-Virus setup script.

uninstall.pl – Kaspersky Anti-Virus removal script.

keepup2date.sh – script that configures the keepup2date component;

proxy_setup.pl – script configuring a Squid proxy for integration with Kaspersky Anti-Virus.

/opt/kaspersky/kav4proxy/sbin/kav4proxy-kavicapserver – executable file of the main Kaspersky Anti-Virus component.

/opt/kaspersky/kav4proxy/share/contrib/kav4proxy.wbm – Webmin plug-in module.

/opt/kaspersky/kav4proxy/share/doc/ – directory containing license information and deployment documentation:

LICENSE – license agreement;

README-SQUID.txt – instruction for Kaspersky Anti-Virus integration with Squid proxy server.

/opt/kaspersky/kav4proxy/share/man/ – directory containing man files.

To connect Kaspersky Anti-Virus help system (manual pages) running under Linux, use the following command:

```
# export MANPATH="$MANPATH:/opt/kaspersky/kav4proxy/share/man/:".
```

/opt/kaspersky/kav4proxy/share/notify/ – directory containing notification templates.

/opt/kaspersky/kav4proxy/share/examples/ – directory containing sample configurations of Kaspersky Anti-Virus:

kav4proxy-default.conf – default configuration file of Kaspersky Anti-Virus;

notify.sh – administrator notification script.

/var/log/kaspersky/kav4proxy/ – directory where log files of Kaspersky Anti-Virus are stored.

After Kaspersky Anti-Virus installation on a server running FreeBSD, the program files will be located as follows (provided that the default paths are accepted):

/usr/local/etc/kaspersky/kav4proxy.conf – configuration file containing Kaspersky Anti-Virus parameters;

/usr/local/bin/ – directory containing executable files of the application components:

kav4proxy-keepup2date – utility updating the anti-virus databases;

kav4proxy-licensemanager – utility for license keys management.

/usr/local/libexec/kaspersky/kav4proxy/avbasestest – utility validating the downloaded databases for the keepup2date component.

/usr/local/etc/rc.d/kav4proxy – Kaspersky Anti-Virus management script.

/usr/local/libexec/kaspersky/kav4proxy/setup/ – directory containing scripts for post-installation setup and removal of the Kaspersky Anti-Virus:

postinstall.pl – post-installation Kaspersky Anti-Virus setup script.

uninstall.pl – Kaspersky Anti-Virus removal script.

keepup2date.sh – script that configures the keepup2date component;

proxy_setup.pl – script configuring a Squid proxy for integration with Kaspersky Anti-Virus.

/usr/local/sbin/kav4proxy-kavicapserver – executable file of the main Kaspersky Anti-Virus component.

/usr/local/share/kav4proxy/contrib/kav4proxy.wbm – Webmin plug-in module.

/usr/local/share/doc/kav4proxy/ – directory containing license information and deployment documentation:

LICENSE – license agreement;

README-SQUID.txt – instruction for Kaspersky Anti-Virus integration with Squid proxy server.

/usr/local/man/ – directory containing man files.

To connect Kaspersky Anti-Virus help system (manual pages) running under FreeBSD, use the following command:

```
# setenv MANPATH /usr/local/man.
```

/usr/local/share/kav4proxy/notify/ – directory containing notification templates.

/usr/local/share/examples/kav4proxy/ – directory containing sample configurations of Kaspersky Anti-Virus:

kav4proxy-default.conf – default configuration file of Kaspersky Anti-Virus;

notify.sh – administrator notification script.

/var/log/kaspersky/kav4proxy/ – directory where log files of Kaspersky Anti-Virus are stored.

KASPERSKY LAB

Kaspersky Lab software is internationally renowned for its protection against viruses, malware, spam, network and hacker attacks, and other threats.

In 2008, Kaspersky Lab was rated among the world's top four leading vendors of information security software solutions for end users (IDC Worldwide Endpoint Security Revenue by Vendor). Kaspersky Lab is the preferred developer of computer protection systems among home users in Russia, according to the COMCON survey "TGI-Russia 2009".

Kaspersky Lab was founded in Russia in 1997. Today, it is an international group of companies headquartered in Moscow with five regional divisions that manage the company's activity in Russia, Western and Eastern Europe, the Middle East, Africa, North and South America, Japan, China, and other countries in the Asia-Pacific region. The company employs more than 2000 qualified specialists.

Products. Kaspersky Lab's products provide protection for all systems—from home computers to large corporate networks.

The personal product range includes anti-virus applications for desktop, laptop, and pocket computers, and for smartphones and other mobile devices.

Kaspersky Lab delivers applications and services to protect workstations, file and web servers, mail gateways, and firewalls. Used in conjunction with Kaspersky Lab's centralized management system, these solutions ensure effective automated protection for companies and organizations against computer threats. Kaspersky Lab's products are certified by the major test laboratories, are compatible with the software of many suppliers of computer applications, and are optimized to run on many hardware platforms.

Kaspersky Lab's virus analysts work around the clock. Every day they uncover hundreds of new computer threats, create tools to detect and disinfect them, and include them in the databases used by Kaspersky Lab applications. *Kaspersky Lab's Anti-Virus database is updated hourly*; and the *Anti-Spam database every five minutes*.

Technologies. Many technologies that are now part and parcel of modern anti-virus tools were originally developed by Kaspersky Lab. It is no coincidence that many other developers user the Kaspersky Anti-Virus kernel in their products, including: SafeNet (USA), Alt-N Technologies (USA), Blue Coat Systems (USA), Check Point Software Technologies (Israel), Clearswift (UK), CommuniGate Systems (USA), Critical Path (Ireland), D-Link (Taiwan), M86 Security (USA), GFI (Malta), IBM (USA), Juniper Networks (USA), LANDesk (USA), Microsoft (USA), NETASQ (France), NETGEAR (USA), Parallels (Russia), SonicWALL (USA), WatchGuard Technologies (USA), ZyXEL Communications (Taiwan). Many of the company's innovative technologies are patented.

Achievements. Over the years, Kaspersky Lab has won hundreds of awards for its services in combating computer threats. For example, in 2010 Kaspersky Anti-Virus was given several top Advanced+ awards after a series of tests held by AV-Comparatives, a renowned Austrian anti-virus lab. But Kaspersky Lab's main achievement is the loyalty of its users worldwide. The company's products and technologies protect more than 300 million users, and its corporate clients number more than 200,000.

Kaspersky Lab official site:	http://www.kaspersky.com
Virus Encyclopedia:	http://www.securelist.com
Anti-Virus Lab:	newvirus@kaspersky.com (only for sending archived files that seem to be infected)
	http://support.kaspersky.com/virlab/helpdesk.html
	(for queries to virus analysts)
Kaspersky Lab applications web forum:	http://forum.kaspersky.com

INFORMATION ABOUT THIRD-PARTY CODE

The legal_notices.txt file contains the information about third-party code, located in the application setup folder.