

Kaspersky Endpoint Security 12.0 for Linux

Quick Start Guide to installation and initial configuration using the command line

This guide will help you perform the installation and initial configuration of Kaspersky Endpoint Security 12.0 for Linux using the command line on an individual device in your infrastructure.

This guide does not cover the installation of Kaspersky Endpoint Security 12.0 for Linux in Light Agent mode, or the installation and initial configuration of the application using Kaspersky Security Center. For details of the usage modes of the application, see the Help: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/93769.htm>.

In that article, you will also find recommendations on how to optimize the application to minimize the impact on the performance of your infrastructure.

Preparing to install Kaspersky Endpoint Security 12.0 for Linux	2
Deployment and initial configuration of Kaspersky Endpoint Security 12.0 for Linux	4
Installing the application locally on the command line	4
Initial configuration of Kaspersky Endpoint Security 12.0 for Linux after installing the package	4
Optimizing application performance	7
Configuring the Web Threat Protection and Network Threat Protection tasks	7
Specifying File Threat Protection task settings and exclusions	8
Recommendations for typical server roles	9
PostgreSQL DB Server	9
Web server	9
Proxy server	9

Preparing to install Kaspersky Endpoint Security 12.0 for Linux

Preparation for installation involves the following steps:

1. Making sure the target device (the device on which you are installing the application) satisfies the minimum hardware requirements of Kaspersky Endpoint Security 12.0 for Linux:
<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/261258.htm>.

Please note that the amount of system resources required by Kaspersky Endpoint Security 12.0 for Linux depends on the functionality that you plan to use and the workload that the device must process. This means that the minimum system requirements are sufficient for a server not under load, while a high-load server may require significantly more resources.

2. Making sure the target device satisfies the minimum software requirements of Kaspersky Endpoint Security 12.0 for Linux. The Help contains a list of architectures (32-bit, 64-bit, and ARM 64-bit) and OS distributions that are supported for installing Kaspersky Endpoint Security 12.0 for Linux:
<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/261283.htm>.

If your OS is not on the list of supported operating systems, we recommend contacting Kaspersky Technical Support to find out if you can installing Kaspersky Endpoint Security 12.0 for Linux on your operating system.

If your operating system is not on the list, it may mean that Kaspersky Endpoint Security 12.0 for Linux had not been tested for compatibility with this OS at the time when the release version of the application was released. This may be the case if the OS is outdated and no longer supported by the OS vendor or, conversely, if the OS is too new.

3. Making sure that the device has all packages needed for installing Kaspersky Endpoint Security 12.0 for Linux and preparing the OS.

For instructions on how to check if the OS is suitably configured to install the application, refer to the Help: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/248502.htm>.

Please note:

- You must check if the perl interpreter 5.10 or later is installed. To do so, look at the output of the perl -v command. Example:

```
$ perl -v
This is perl 5, version 30, subversion 0 (v5.30.0) built for x86_64-linux-
gnu-thread-multi
...
```

- For RHEL-like systems, such as RED OS, you must also make sure the perl-Getopt-Long and perl-File-Copy packages are present. To do so, run the following commands:

- rpm -q perl-Getopt-Long
- rpm -q perl-File-Copy

Example output of the rpm -q perl-Getopt-Long command if the package is installed:

```
$ rpm -q perl-Getopt-Long
perl-Getopt-Long-2.51-1.el7.noarch
```

Example output of the rpm -q perl-Getopt-Long command if the package is not installed:

```
$ rpm -q perl-Getopt-Long
package perl-Getopt-Long is not installed
```

The initial configuration script, which runs after installing the Kaspersky Endpoint Security 12.0 for Linux package, requires perl and the above mentioned packages.

4. Making sure a license key is present to activate Kaspersky Endpoint Security 12.0 for Linux.

The set of available features of Kaspersky Endpoint Security depends on the license (see the table in the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/256558.htm>).

For details about Kaspersky Endpoint Security 12.0 for Linux licensing, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/69238.htm>.

You can add a license key using an activation code or a key file.

To activate with an activation code, the application needs internet access to connect to Kaspersky activation servers.

If necessary, you can get a key file based on an activation code: <https://support.kaspersky.com/common/buy/7180>.

When you first install Kaspersky Endpoint Security 12.0 for Linux, the initial configuration script prompts you to activate the application using a trial license with a term of 30 days.

Deployment and initial configuration of Kaspersky Endpoint Security 12.0 for Linux

Installing the application locally on the command line

Before installation, you must find out the architecture of your operating system (32-bit, 64-bit or ARM 64-bit) and the type of package manager that your OS uses.

Then you need to copy the `kesl` package (a required package that installs the application) and the `kesl-gui` package (an optional package that installs the graphical user interface) to the target device. You must choose packages to match your architecture, package manager type, and your OS bitness.

If you do not want to use a graphical user interface on the target device, you do not need to copy the `kesl-gui` package.

We recommend using the latest builds of the packages; these contain fixes for errors discovered after the official release date of the version of Kaspersky Endpoint Security that you are installing. To download the latest build with fixes, you can use the Kaspersky Technical Support request form: <https://support.kaspersky.com/corporate/faq-for-business-products#how-to-create-technical-support-request>.

For instructions on how to install the `kesl` and `kesl-gui` packages for various architectures and package managers for different Linux OS distributions, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/233694.htm>.

Note that installing the `kesl-gui` package without first installing the `kesl` package is not possible.

Initial configuration of Kaspersky Endpoint Security 12.0 for Linux after installing the package

After installation, you must run the initial configuration script for Kaspersky Endpoint Security 12.0 for Linux.

The following is a discussion of the initial configuration script in interactive mode (for details, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/197897.htm>). This article does not cover initial configuration in automatic mode using an `autoinstall.ini` file; for a description of this process, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/197909.htm>.

To start the Kaspersky Endpoint Security initial configuration script, run the following command:

```
# /opt/kaspersky/kesl/bin/kesl-setup.pl
```

You must run the initial configuration script as root.

Please note the following important initial configuration steps:

- At the first step of the script, when asked whether you want to use the application in Light Agent mode to protect virtual environments, answer `n` (No).
- At the next step, select the language. This setting determines the language in which the script displays the End User License Agreement and the Privacy Policy.
- You are then prompted to read and accept the End User License Agreement and the Privacy Policy.

If you fail to accept at least one of these agreements, the initial configuration script terminates, and you cannot use Kaspersky Endpoint Security 12.0 for Linux.

- Next, you are prompted to accept or reject the terms of use of the Kaspersky Security Network (for details, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/246794.htm>).

Use of Kaspersky Security Network is voluntary. You can enable or disable the use of Kaspersky Security Network at any time after installing the application.

- The next important step of the initial configuration script is to specify proxy server settings to make sure the application has internet access.

If the target device uses a proxy server for internet access, you must configure specify proxy server settings to allow the application to update its databases from Kaspersky public update servers. For details of proxy server configuration, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/197905.htm>.

- The next important step of the initial configuration script is the initial update of the application databases. The Kaspersky Endpoint Security 12.0 for Linux distribution kit does not include the anti-virus databases that the application needs to work.

The application is not activated until the initial database update task is successfully completed.

You can perform the database update while the initial configuration script is running or after the script has finished.

After completing the initial configuration, updating the databases, and adding the license key, Kaspersky Endpoint Security 12.0 for Linux becomes operational with default settings.

You can check if the application was installed and activated successfully with the following command:

```
# kectl-control --app-info
```

Example output of the kectl-control --app-info command:

```
# kectl-control --app-info
Name: Kaspersky Endpoint Security 12.0 for Linux
Version: 12.0.0.6805
Policy: Kaspersky Security Center

License information: The key is valid
License expiration date: 2024-09-09 00:00:00
MDR BLOB file status: Not loaded

Storage state: No objects in Storage
Storage space usage: Storage size is unlimited

Last run date of the Scan_My_Computer task: Never run

Last release date of databases: 2024-08-22 19:59:00
Application databases loaded: Yes

Kaspersky Security Network usage: Extended KSN mode
Kaspersky Security Network infrastructure: Kaspersky Security Network
Managed Detection and Response state: Inactive
File Threat Protection: Available and running
Container Monitoring: Unavailable due to license limitation
System Integrity Monitoring: Unavailable due to license limitation
Firewall Management: Available and stopped
Anti-Cryptor: Available and stopped
Web Threat Protection: Available and stopped
Device Control: Available and running
Removable Drives Scan: Available and stopped
Network Threat Protection: Available and running
Behavior Detection: Available and running
Application Control: Available and stopped
Kaspersky Endpoint Detection and Response (KATA) Integration: Available and stopped
Application update state: No application updates available
#
```

Some application tasks may not be available due to license restrictions. For example, in the screenshot above, the Container Monitoring and System Integrity Monitoring tasks are not available for this reason (see <https://support.kaspersky.com/help/KES4Linux/12.0.0/en-US/210891.htm> and <https://support.kaspersky.com/help/KES4Linux/12.0.0/en-US/197992.htm>).

The **Policy: Not Applied** line means that Kaspersky Endpoint Security 12.0 for Linux is running with local settings, and not with policy settings centrally distributed through Kaspersky Security Center.

Optimizing application performance

This section provides recommendations for configuring the application to minimize the impact on the performance of business applications running on the target device. This section also provides recommendations for configuring the application for several typical server roles.

Command line settings are described because when installed as described above, the application is not governed by a Kaspersky Security Center policy.

You must configure the application as root or as a user to which the Kaspersky Endpoint Security 12.0 for Linux administrator role was assigned during the installation process.

You can also configure the application centrally using Kaspersky Security Center policies and tasks in the Kaspersky Security Center Web Console (<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/245583.htm>) or Administration Console (<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/245658.htm>). This article does not cover the management the application using Kaspersky Security Center.

Configuring the Web Threat Protection and Network Threat Protection tasks

After completing the installation and initial configuration of Kaspersky Endpoint Security 12.0 for Linux, analyze the resource consumption on the device and the impact on the performance of installed applications using built-in tools of your operating system or monitoring system data. If the analysis reveals a significant degradation of application performance and/or levels of system resource usage that are critical for the operation of the OS or applications, we recommend performing additional configuration of the running Web Threat Protection and Network Threat Protection tasks.

By default, these tasks are disabled (the Web Threat Protection task is enabled at application startup only if an installed web browser is detected on the system).

Consider the following:

- If the system has its own network rules (or firewalld rules) with Drop policies, then for the application to work correctly, you need to add an accept rule to allow incoming connections from the local host:

```
iptables -A INPUT -i lo -j ACCEPT
```
- If you did not specify proxy server settings during the initial configuration of the application, and the device does not have direct access to the internet, then for the Web Threat Protection task to work correctly, you must set CertificateVerificationPolicy=LocalCheck in the encrypted connections scan settings (for details, see the Help article: <https://support.kaspersky.com/help/KES4Linux/12.0.0/en-US/198037.htm>).

You can specify this setting by running the following command:

```
# kesc-control --set-net-settings CertificateVerificationPolicy=LocalCheck
```

With this value, the application does not use the internet to check and download the missing chains that are required to validate a certificate.

You can also specify proxy server settings by running the following command:

- If your proxy server uses authentication:

```
# kesc-control --set-app-settings UseProxy=Yes ProxyServer=<user name>:<password>@<IP address of the proxy server>:<port>
```

- If your proxy server does not use authentication:

```
# kesl-control --set-app-settings UseProxy=Yes ProxyServer=<IP address of the proxy server>:<port>
```

- If you are using the Network Threat Protection task, setting MonitorNetworkPorts=All is guaranteed to cause resources to become unavailable via the SMB protocol.
- Using the Web Threat Protection and Network Threat Protection tasks together with Kubernetes requires additional meticulous and painstaking configuration of OS network rules, and in some cases it may be not possible at all, for example when using Kubernetes with CNI Cillium.

Specifying File Threat Protection task settings and exclusions

The File Threat Protection task has ID=1 and is enabled by default after the application is installed and activated. For details about this task, see the Help article:

<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/197961.htm>.

The File Threat Protection task intercepts file operation events in the system and, by default, blocks these operations until the application has finished processing the event. After processing is complete, the application either allows the operations and lifts the block from the file operation, or the application subjects the scanned file to actions described in the FirstAction, SecondAction settings of the File Threat Protection task (for details about these settings, see the Help article:

<https://support.kaspersky.com/KES4Linux/12.0.0/en-US/234812.htm>).

Enabling archive scanning (the ScanArchived setting of the File Threat Protection task) can have a significant impact on system performance. If archive scanning is enabled, access to the archived file is "delayed" until it is unpacked into a temporary location and scanned. The definition of archive here includes traditional archives as well as packaged objects such as .jar.

By default, archive scanning is disabled and it we do not recommend enabling it as long as performance is a priority.

Several types of exclusions exist for the File Threat Protection task:

- In the task settings, you can configure exclusions by file path (you can also specify the mount type), file mask, threat type, and path to the process performing file operations.

Exclusions of this type work as follows: the file operation is intercepted and blocked; then if the application determines that the operation is being performed in a scope excluded from scanning, the block from the file operation is lifted, and the object of the file operation is not scanned. Minimal delays in file operations do occur.

- In the general application settings, you can configure global exclusions.

Such exclusions can be specified for system mount points and system mount point types (Local, AllRemoteMounted, Mounted:NFS, Mounted:SMB, Mounted:Custom). Scopes excluded in this way are not tracked by the application at all, which means no delay whatsoever is introduced to file operations.

We recommend adding mounted remote resources with unstable or slow connections (for example, NFS, SMB) to global exclusions.

For information on configuring exclusions of both types for the File Threat Protection task, see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/248490.htm>.

Do exclusions make the device less secure? By adding exclusions, we strive to achieve a reasonable balance between the security of the system and its performance. For example, according to Kaspersky virus experts, you can safely add data files and index files of databases to exclusions. It is believed that no danger exists of such files becoming virus vectors.

For information on configuring exclusions to reduce the impact on the performance of the target device, see also the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/206054.htm>.

Recommendations for typical server roles

PostgreSQL DB Server

In case of the PostgreSQL database, it is critical to prevent even the slightest delays in file operations with data files and Write Ahead Log (WAL) files. We also recommend excluding file operations of PostgreSQL processes from scanning.

For a typical PostgreSQL installation, these files (including the PostgreSQL executable files) are located at the following paths:

- `/var/lib/postgresql/13/main` — directory with data
- `/var/lib/postgresql/13/main/pg_wal` — WAL files
- `/usr/lib/postgresql/13/bin/` — executable files

Ask your database administrators how the paths are configured on your device.

We recommend adding the first two paths to global exclusions from scanning (see the Help article: <https://support.kaspersky.com/KES4Linux/12.0.0/en-US/248490.htm>).

If the paths to the data directory and the WAL files are not separate mount points in the system (you can find the list of mount points in the output of the `mount` command), you must make them mount points. For example, to make a mount point from the data directory for a typical PostgreSQL installation (`/var/lib/postgresql/13/main`), run the following command:

```
mount --bind /var/lib/postgresql/13/main/ /var/lib/postgresql/13/main
```

In this case, there is no need to add a separate mount point for the WAL file directory because it is already under the `/var/lib/postgresql/13/main` mount point.

Next, you need to add a global exclusion:

```
# kesl-control --set-app-settings ExcludedMountPoint.item_0000=/var/lib/postgresql/13/main
```

To ensure that the mount point created with the `'mount --bind'` command remains in the system after restart, add the following line to `/etc/fstab`:

```
/var/lib/postgresql/13/main /var/lib/postgresql/13/main none defaults,bind 0 0
```

To apply the global exclusions configured above, restart the application:

```
# systemctl restart kesl.service
```

Web server

For a typical web server, in high user load conditions and when serving pages over HTTPS, the Web Threat Protection and Network Threat Protection tasks seriously impact system resource consumption due to the cost of decrypting HTTPS traffic. In this case, we recommend disabling the Web Threat Protection and Network Threat Protection tasks.

Proxy server

In case of a proxy server or a load balancer installed on the target device, we recommend disabling the Web Threat Protection and Network Threat Protection tasks.