kaspersky

Kaspersky Vulnerability Management 1.0

© 2026 AO Kaspersky Lab

Contents

- Introduction to Kaspersky Vulnerability Management
  - Fixed issues
  - Hardware and software requirements
  - Solution architecture
  - Distribution kit
  - User interface of the solution
  - Supported operating systems
  - Supported software
  - Supported compliance profiles
- Getting started
  - Preparing the deployment
    - Installing the Kaspersky Security Center Linux Administration Server
    - Installing the Kaspersky Security Center Web Console
    - Installing the management web plug-in for the solution
    - Creating user accounts and configuring role-based access
    - Creating and configuring a database in PostgreSQL
    - Installing the Kaspersky Security Center Network Agent
    - Installing the jq tool
    - Installing OpenSSL for Linux
    - Installing other necessary tools
    - Applying the kaspersky_astra_pub_key.gpg key of the solution
  - Deployment
    - Installing and configuring the components of Kaspersky Vulnerability Management
    - Restarting the Kaspersky Security Center Web Console Plugin service
    - Enabling the solution on the Administration Server
    - Enabling the solution in Network Agents of assets
- Managing assets
  - Asset inventory
  - Changing asset importance
  - Managing asset groups
  - Software inventory
- Configuring and running a scan
  - About scan types
  - Creating and configuring scan tasks
  - Agentless white-box scanning
  - Agentless asset black-box scanning
  - Scanning for vulnerabilities in system and application software
  - Scanning assets for profile compliance
  - Importing custom compliance profiles

Analysis and prioritization of vulnerabilities
Viewing detected vulnerabilities
Managing reports
Creating and editing report templates
Generating reports from templates
Exporting reports
Deleting reports and report templates
Managing report data
Vulnerability elimination
Assessing vulnerability elimination
Administration
Known limitations
Upgrading from version 1.0 to version 1.0.1
Changing the database update period
Advanced configuration of the solution using configuration files
worker.json
scanner.json
privateapi.json
publicapi.json
vms.json
updater-scanner.json
updater-compliance.json
Using the setup assistant
Event logging
System audit event log
Event logs of solution components
Removing the solution
Disabling Kaspersky Vulnerability Management in Network Agents of assets
Disabling Kaspersky Vulnerability Management on the Administration Server
Uninstalling the components of Kaspersky Vulnerability Management
Uninstalling the management web plug-in for the solution
Using the public API of the solution
Legal information
Licensing of the solution
Data provision
Information about third-party code
Trademark notices
Glossary
Administration group
Administration Server
Compliance profile
CVSS
EPSS percentile
EPSS probability
Group task
OVAL
SBOM

Severity of the vulnerability

Vulnerability

Vulnerability elimination priority

XCCDF

Introduction to Kaspersky Vulnerability Management

The intended purpose of Kaspersky Vulnerability Management (hereinafter also referred to as the "solution") is vulnerability management in the IT infrastructure of an organization.

Kaspersky Vulnerability Management:

- Provides information about the assets of the organization's IT infrastructure
- Performs vulnerability scanning on assets
- Analyzes asset compliance with internal, industry and regulatory security requirements
- Provides the means to analyze and prioritize detected vulnerabilities
- Provides a public API that allows third-party systems to receive data about assets and detected vulnerabilities

Regular database updates help keep the solution up to date.

You can remotely manage the solution using the Kaspersky Security Center Web Console.

In the territory of the USA, the update functionality and KSN functionality may not be available in the solution.

Fixed issues

The following issues were found in Kaspersky Vulnerability Management version 1.0 and fixed in version 1.0:

- Fixed an issue that prevented dragging table lines in the New report template wizard and in the **Data presentation** section of report template properties.
- Fixed an issue that prevented removing entries in the asset card on the **Detected vulnerabilities** tab for vulnerabilities that had been detected using agentless black-box scanning and then fixed and rescanned.

Hardware and software requirements

Minimum hardware requirements

The minimum hardware requirements for the device on which you plan to install Kaspersky Vulnerability Management components are listed in the table below.

Minimum hardware requirements

Number of assets	Installation option	
	All on one device **Kaspersky Vulnerability Management components, including the DBMS, and Kaspersky Security Center components are installed on the same device.**	**Kaspersky Vulnerability Management installed separately** **Kaspersky Vulnerability Management components, including the DBMS, are installed separately from the Kaspersky Security Center components.**
1–1000	• CPU: 16 cores, 2.5 GHz • RAM: 24 GB • Free disk space: 300 GB • Disk performance, IOPS: 1000	• CPU: 12 cores, 2.5 GHz • RAM: 16 GB • Free disk space: 200 GB • Disk performance, IOPS: 1000
1001–10000	• CPU: 20 cores, 2.5 GHz • RAM: 48 GB • Free disk space: 600 GB • Disk performance, IOPS: 1000	• CPU: 16 cores, 2.5 GHz • RAM: 32 GB • Free disk space: 400 GB • Disk performance, IOPS: 1000

Software requirements

Kaspersky Vulnerability Management components can be installed on a device running one of the following operating systems:

- ALT Server 10 64-bit
- RED OS 8.0 64-bit
- Astra Linux Common Edition (operational update 2.12) 64-bit

  This operating system is not supported in the closed software environment mode.

- Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.5) 64-bit, also in closed software environment mode
- Debian 11.0 64-bit
- Platform V SberLinux OS Server (SLO) 8.10.1 64-bit
- Platform V SberLinux OS Server (SLO) 9.5.1 64-bit
- Ubuntu Server 22.04 LTS 64-bit

Kaspersky Security Center Network Agent supports the Kaspersky Vulnerability Management functionality in the following operating systems:

- ALT Server 10 64-bit
- RED OS 8.0 64-bit
- Astra Linux Common Edition (operational update 2.12) 64-bit
- Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.5) in closed software environment mode 64-bit
- Red Hat Enterprise Linux Server 10.0 64-bit
- Rocky Linux 9.6 64-bit
- Ubuntu Server 22.04 LTS 64-bit
- Ubuntu Server 24.04 LTS 64-bit
- Windows 10 32-bit/64-bit
- Windows 11 25H2 64-bit
- Windows Server 2025 64-bit

One of the following database management systems can be used for the solution's database:

- PostgreSQL 15.0, 16.0, 17.0, 18.4
- Postgres Pro 18.4
- Postgres Pro Certified 18.4

Supported versions of Kaspersky Security Center components

The current version of Kaspersky Vulnerability Management supports version 16.3 of the following components: Kaspersky Security Center Linux Administration Server, Kaspersky Security Center Web Console, and Network Agent.

> The current version of Kaspersky Vulnerability Management does not support the Kaspersky Security Center Linux Administration Server installed as a high-availability cluster.

For hardware and software requirements of the Kaspersky Security Center Administration Server, Kaspersky Security Center Web Console, and the Network Agent, see the Kaspersky Security Center Help.

Solution architecture

The Kaspersky Vulnerability Management solution comprises the Kaspersky Vulnerability Management components that interact with the components of Kaspersky Security Center Linux (hereinafter also referred to as "Kaspersky Security Center").

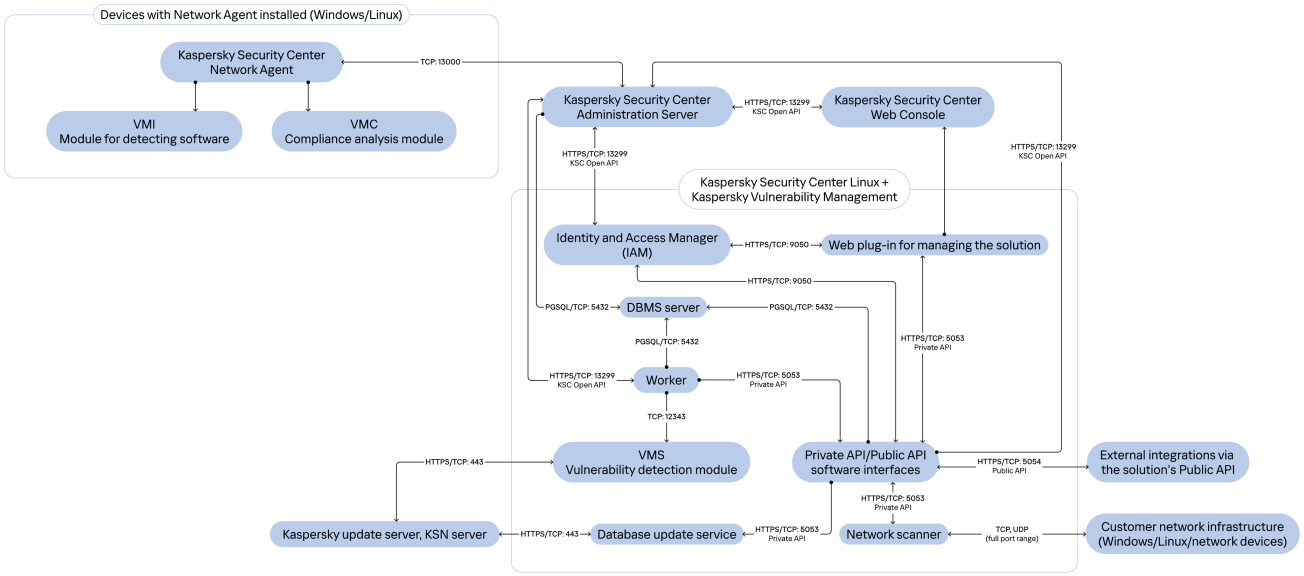
Kaspersky Vulnerability Management components interact with the following Kaspersky Security Center components:

- **Kaspersky Security Center Administration Server**. Serves as a centralized store of information about the applications installed on the organization's network and manages these applications.
- **Kaspersky Security Center Web Console**. Provides a web interface for creating and managing the client organization's network protection system managed by Kaspersky Security Center.
- **Kaspersky Security Center Network Agent**. Enables interaction between the devices on which it is installed, the Kaspersky Vulnerability Management components, and the Kaspersky Security Center Administration Server (hereinafter also referred to as the "Administration Server"); participates in software inventory and the scanning of assets for compliance with profiles.
- **Identity and Access Manager (IAM)**. Administration Server component that allows using domain authentication (Single Sign-On). The Single Sign-On function allows the user to enable secure authentication in Kaspersky Security Center Web Console without re-entering the password on the corporate network.
- **Web plug-in for managing the solution via Kaspersky Security Center Web Console**. After installing the plug-in, the **Vulnerability management** section appears in the main menu of Kaspersky Security Center Web Console, and the Kaspersky Vulnerability Management functionality built into some other sections of Kaspersky Security Center Web Console becomes available.

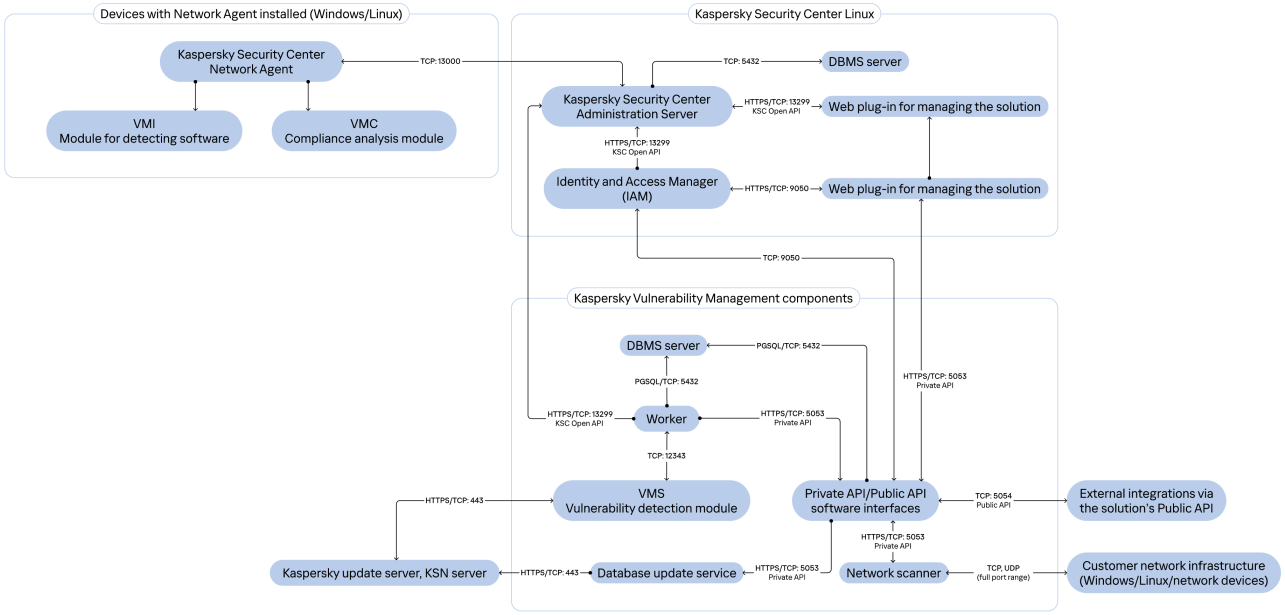
Kaspersky Vulnerability Management includes the following components:

- **Database.** Stores solution data.
- **Worker**. Performs service tasks and tasks created for Kaspersky Vulnerability Management in Kaspersky Security Center.
- **Network scanner**. Involved in *Agentless white-box scanning* and *Agentless black-box scanning* tasks.
- **Private API**. Web service for interacting with the solution's management web plug-in for the Kaspersky Security Center Web Console, Network Scanner, and Worker.
- **Public API**. Web service for interaction of external systems with Kaspersky Vulnerability Management.
- **VMS**. Vulnerability detection module. Involved in the *Vulnerability scanning* and *Agentless white-box scanning* tasks.
- **VMI**. Module for detecting software installed on devices. Carries out an inventory of assets.
- **VMC**. Compliance analysis module. Performs the *Scanning assets for profile compliance* task.
- **Solution database update service**. Web application that downloads databases for the Network scanner, VMS, and VMC modules.

The diagrams below show the architecture of the solution depending on the installation option.



Components of Kaspersky Vulnerability Management and Kaspersky Security Center Administration Server are installed on the same device



Components of Kaspersky Vulnerability Management and Kaspersky Security Center Administration Server are installed on different devices

Distribution kit

The Kaspersky Vulnerability Management distribution kit includes the following files:

kaspersky-vm_<version number>_amd64.deb	Installation package for operating systems that use the `.deb` package format.
kaspersky-vm-<version number>-1.x86_64.rpm	Installation package for operating systems that use the `.rpm` package format.
ksn.<language ID>	File with the text of the KSN Statement.
legal_notices.txt	File with information about third-party code.
license.<language ID>	File with the text of the End User License Agreement.
privacy.<language ID>	File with the text of the Privacy Policy.
webconsole-plugin.zip	ZIP archive for installing the Kaspersky Vulnerability Management web plug-in via Kaspersky Security Center Web Console.

User interface of the solution

The solution is managed through the Kaspersky Security Center Web Console interface (hereinafter also referred to as the "Web Console") using the following items:

- The main menu in the left part of the Web Console window. The section of the primary menu that contains nested sections for managing the solution is called **Vulnerability management**.
- The workspace in the right part of the Web Console window. The workspace displays information and controls for the selected section.

For information about logging in to Kaspersky Security Center Web Console, refer to the Kaspersky Security Center Help.

Data in all sections of the solution are presented in the form of tables.

The Kaspersky Vulnerability Management sections in the Web Console interface contain controls that you can use to do the following:

- Update data in tables. Clicking the **Refresh** button in the upper-left corner of the table starts an update of the data in the table.
- Export data to a CSV file. In the tables with vulnerabilities and assets, click the **Export to CSV** button.

  Export does not work when the number of records exceeds one million.

- Delete data from the solution. In the tables, using the check boxes to the left of the names of custom compliance profiles, report templates, and generated reports, select the required ones and click the **Delete** button.
- Search data in tables. In the search bar above the table, enter your search text and click the icon. The columns that can be searched are indicated in the search bar.
- Filter data in tables. Click the icon in the upper-right corner of the table to open a window, and in that window, specify and apply a filtering criterion for data in the table.

- Display data in tables. Click the ⚙ icon in the upper-right corner of the table to open a window, and in that window, select the columns that you want to display.
- Sort data in tables. Click the header of the table column and in the drop-down menu select the ascending or descending sorting order. If a sorting order is currently applied to a column, it is indicated by an icon to the right of the column name.

Supported operating systems

Kaspersky Vulnerability Management supports vulnerability scans in operating systems listed below.

Vulnerability scanning

- ALT Server 10
- RED OS 8.0
- Astra Linux Common Edition (operational update 2.12)
- Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.5) in closed software environment mode
- CentOS Stream 10.0
- Rocky Linux 9.6
- Ubuntu Server 22.04 LTS
- Ubuntu Server 24.04 LTS
- Windows 10 versions 1507, 1607, 1809, 21H2, 22H2
- Windows 11 25H2
- Windows Server 2025

All operating systems listed above are 64-bit.

Scanning assets for profile compliance

- RED OS
- The ALT family of operating systems
- Astra Linux
- Debian
- Ubuntu

- Windows
- Windows Server

Supported software

Kaspersky Vulnerability Management supports vulnerability scans in software listed below.

Software supported in the vulnerability database for the Agentless white-box scanning and Vulnerability scanning tasks

Software supported in the vulnerability database for the Agentless black-box scanning task

Supported compliance profiles

Kaspersky Vulnerability Management supports scanning assets for compliance with the following profiles:

- Scanning the Astra Linux configuration for compliance with the requirements of regulatory, methodological and organizational documents:
  - FSTEC of Russia profile No. 117 for Astra Linux
  - FSTEC of Russia profile No. 21 for Astra Linux
  - FSTEC of Russia profile No. 31 for Astra Linux
  - FSTEC of Russia profile No. 239 for Astra Linux
  - Kaspersky C Basic profile for Astra Linux
  - Kaspersky D Basic profile for Astra Linux
  - KSC profile for Astra Linux
- Scanning the Debian OS configuration for compliance with the requirements of regulatory, methodological and organizational documents:
  - FSTEC of Russia profile No. 117 for Debian
  - FSTEC of Russia profile No. 21 for Debian
  - FSTEC of Russia profile No. 31 for Debian
  - FSTEC of Russia profile No. 239 for Debian
  - Kaspersky C Basic profile for Debian

    - Kaspersky D Basic profile for Debian
    - KSC profile for Debian
- Scanning the Ubuntu OS configuration for compliance with the requirements of regulatory, methodological and organizational documents:
    - FSTEC of Russia profile No. 117 for Ubuntu
    - FSTEC of Russia profile No. 21 for Ubuntu
    - FSTEC of Russia profile No. 31 for Ubuntu
    - FSTEC of Russia profile No. 239 for Ubuntu
    - Kaspersky C Basic profile for Ubuntu
    - Kaspersky D Basic profile for Ubuntu
    - KSC profile for Ubuntu
- Scanning the ALT Linux configuration for compliance with the requirements of regulatory, methodological and organizational documents:
    - FSTEC of Russia profile No. 117 for ALT Linux
    - FSTEC of Russia profile No. 21 for ALT Linux
    - FSTEC of Russia profile No. 31 for ALT Linux
    - FSTEC of Russia profile No. 239 for ALT Linux
    - Kaspersky C Basic profile for ALT Linux
    - Kaspersky D Basic profile for ALT Linux
    - KSC profile for ALT Linux
- Scanning the RedOS configuration for compliance with the requirements of regulatory, methodological and organizational documents:
    - FSTEC of Russia profile No. 117 for RedOS
    - FSTEC of Russia profile No. 21 for RedOS
    - FSTEC of Russia profile No. 31 for RedOS
    - FSTEC of Russia profile No. 239 for RedOS
    - Kaspersky C Basic profile for RedOS
    - Kaspersky D Basic profile for RedOS
    - KSC profile for RedOS

- Scanning the Windows configuration for compliance with the requirements of regulatory, methodological and organizational documents:
  - FSTEC of Russia profile No. 117 for Windows
  - FSTEC of Russia profile No. 21 for Windows
  - FSTEC of Russia profile No. 31 for Windows
  - FSTEC of Russia profile No. 239 for Windows
  - Kaspersky C Basic profile for Windows
  - Kaspersky D Basic profile for Windows
  - KSC profile for Windows

D Basic profiles take into account international data security benchmarks for payment systems.
C Basic profiles take into account international recommendations and best practices for cybersecurity and protection of IT systems.
KSC profiles take into account the Hardening Guide for Kaspersky Security Center.

Getting started

This section describes the steps for preparing the components and deploying Kaspersky Vulnerability Management.

Preparing the deployment

You can deploy Kaspersky Vulnerability Management on the same device as the Kaspersky Security Center Administration Server or on a separate device.

> For supported versions of applications, refer to the *Hardware and software requirements* article.

Before deploying Kaspersky Vulnerability Management, make sure the following prerequisites are met:

- On the device for deployment of Kaspersky Vulnerability Management or on a separate device:
  - The Administration Server of Kaspersky Security Center Linux is installed.
  - The Kaspersky Security Center Web Console application is installed on the device with the Administration Server or separately and uses the Identity and Access Manager (IAM).
  - The solution's management web plug-in for the Kaspersky Security Center Web Console is installed on the device with the Kaspersky Security Center Web Console.
  - Roles and user accounts for the solution have been created on the device with the Kaspersky Security Center Web Console.
- On the device for deployment of Kaspersky Vulnerability Management:
  - The solution database has been created and configured in PostgreSQL or Postgres Pro.
  - The Kaspersky Security Center Network Agent is installed.
  - The jq tool is installed.
  - OpenSSL for Linux is installed.
  - The curl tool is installed for making requests to Kaspersky Security Center; coreutils, bash, sed, grep, less, findutils are installed.
  - The kaspersky_astra_pub_key.gpg has key of the solution has been uploaded and applied if you intend to deploy Kaspersky Vulnerability Management on a device running the Astra Linux Special Edition operating system in the closed software environment mode.
- Kaspersky Security Center Network Agent is installed on the managed assets.

Installing the Kaspersky Security Center Linux Administration Server

The Kaspersky Security Center Linux Administration Server as part of the Kaspersky Vulnerability Management solution must be installed on the device on which you intend to deploy the solution or on a separate device. For installation details, refer to the Kaspersky Security Center Help.

For supported versions of Kaspersky Security Center Linux, refer to the *Hardware and software requirements* article.

Installing the Kaspersky Security Center Web Console

Kaspersky Security Center Web Console that uses Identity and Access Manager (IAM) as part of the Kaspersky Vulnerability Management solution must be installed on the same device as the Administration Server or on a separate device. For installation details, refer to the Kaspersky Security Center Help.

For supported versions of the Kaspersky Security Center Web Console, refer to the *Hardware and software requirements* article.

Installing the management web plug-in for the solution

You need to install the management web plug-in for the solution on the device where Kaspersky Security Center Web Console is installed.

To install the management web plug-in for the solution via Kaspersky Security Center Web Console:

1. In the main window of the Web Console, select **Settings** → **Web plug-ins**.

   A list of installed web plug-ins opens.

2. Click **Add from file**.

3. This opens a window; in that window, upload the archive with the web plug-in distribution kit, vulnerability_management.zip, and the signature file, signature.txt. You can find these files in the webconsole-plugin.zip archive.

   The webconsole-plugin.zip archive is included in the overall distribution kit.

4. Click **Add**.

5. Wait for the installation to complete and click **OK** in the information window.

The new plug-in is displayed in the list of installed web plug-ins (**Settings** → **Web plug-ins**). The **Vulnerability management** section appears in the Web Console.

Creating user accounts and configuring role-based access

Before deploying Kaspersky Vulnerability Management, in Kaspersky Security Center Web Console, you must create basic roles and internal user accounts and assign the created roles to the user accounts in accordance with the table below.

For details about creating a role, refer to the Kaspersky Security Center Help.

For details about creating an internal user account, see the Kaspersky Security Center Help.

User accounts for Kaspersky Vulnerability Management must be excluded from two-factor authentication. For more details, refer to the Kaspersky Security Center Help.

Deleting the basic roles or changing the permissions assigned to the roles will cause errors of the solution.

The rights assigned to Kaspersky Vulnerability Management roles apply to all devices.

User account name	Role the user account must have	Scopes to which the role must have access	Permissions to be granted for a scope

VM Admin	VM Admin is a custom role for the main administrator of Kaspersky Vulnerability Management.	Kaspersky Security Center Administration Server:
		• General features:
		  • Basic functionality
		• System management:
		  • Connectivity
		  • Execute scripts remotely
		  • Hardware inventory
		  • Remote installation
		  • Software inventory
		  • Vulnerability and patch management
			• Read
			• Write
			• Execute
			• Perform operations with device selections

		Kaspersky Vulnerability Management: • Assets • Asset groups • Compliance • Network scanning API	• Read • Write

		Kaspersky Vulnerability Management:
		• Public API:
		  • Assets
		  • Software
		  • Vulnerabilities
		• Software
			• Read

		Kaspersky Vulnerability Management: • Reports	• Read • Write • Execute
		Kaspersky Vulnerability Management: • Vulnerabilities	• Read • Execute
VM Worker	VM Worker is the service role for interaction between the Worker, VM API, VMS, and Kaspersky Security Center Administration Server components.	Kaspersky Security Center Administration Server: • General features: • Basic functionality	• Read • Write • Execute
		Kaspersky Security Center Administration Server: • System management: • Connectivity Kaspersky Vulnerability Management: • Vulnerabilities	• Read • Execute
		Kaspersky Vulnerability Management: • Compliance	• Read • Write
		Kaspersky Vulnerability Management: • Network scanning API • Software	• Read

User account name	Role the user account must have	Scopes to which the role must have access	Permissions to be granted for a scope
VM Scanner	Service role for API calls specific to the Network scanner	Kaspersky Security Center Administration Server: • General features: • Basic functionality	• Read
		Kaspersky Vulnerability Management: • Network scanning API	• Read • Write

Creating and configuring a database in PostgreSQL

To create and configure a Kaspersky Vulnerability Management database in PostgreSQL:

1. Using any available tool for working with the PostgreSQL database server (for example, psql), run the command to create the database:

```
CREATE DATABASE "<Database name>" WITH OWNER=postgres ENCODING='UTF8' TABLESPACE=pg_default CONNECTION
LIMIT=-1;
```

2. Create a user role for Kaspersky Vulnerability Management:

```
CREATE ROLE <Role name> WITH PASSWORD '<Password>' LOGIN NOSUPERUSER INHERIT NOCREATEDB NOCREATEROLE
NOREPLICATION;
```

3. Grant the created user role the right to connect to the created database by running the following command:

```
GRANT CONNECT ON DATABASE "<Database name>" TO "<Role name>";
```

4. Go to the created database:

```
\c "<Database name>";
```

5. Create a schema with the name of the created user role in the created database by running the following command:

```
CREATE SCHEMA "<Role name>" AUTHORIZATION "<Role name>";
```

6. If necessary, configure an encrypted TLS connection with the created database. For more information about this, refer to the Secure TCP/IP Connections with SSL article in the PostgreSQL documentation.

7. For the created user role, enable password authentication in the created database:

Kaspersky Vulnerability Management does not support other authentication methods (for example, with a certificate or GSSAPI).

a. Find out the location of the pg_hba.conf configuration file:

```
psql -U postgres -d postgres -c "SHOW hba_file;"
```

b. Open the pg_hba.conf configuration file.

c. Add a line with password authentication settings for the created user role in the created database in the following format:

`<connection type> "<database name>" "<role name>" <connection source> <authentication method>`.

Specify the IP address of the device on which the Kaspersky Vulnerability Management components will be installed as the connection source.

For more information about password authentication in PostgreSQL databases, refer to the Password Authentication article in the PostgreSQL documentation.

For more information about working with the pg_hba.conf configuration file, see the PostgreSQL documentation in The pg_hba.conf file article.

d. Save the pg_hba.conf file.

e. Restart the PostgreSQL service:

```
systemctl reload postgresql;
```

Installing the Kaspersky Security Center Network Agent

Supported version of Network Agent must be installed on the Kaspersky Vulnerability Management device.

On assets that you intend to scan, you need to install a Network Agent that supports the functionality of the solution.

For details on how to install Network Agent, refer to the Kaspersky Security Center Help. During the installation of the Network Agents, specify the credentials for connecting to the Administration Server that is used as part of the solution.

For information about supported versions of Network Agent and the operating systems in which Network Agent supports the functionality of the solution, see the *Hardware and software requirements* article.

Installing the jq tool

On the Kaspersky Vulnerability Management device, you need to install the jq tool for parsing JSON data. To install this tool, use the package manager of your operating system.

For example, you can use one of the following commands:

- `sudo apt install jq`
- `sudo yum install jq`

Installing OpenSSL for Linux

OpenSSL for Linux must be installed on the Kaspersky Vulnerability Management device. To install this tool, use the package manager of your operating system.

For example, you can use one of the following commands:

- `sudo apt install openssl`
- `sudo yum install openssl`

Installing other necessary tools

On the Kaspersky Vulnerability Management device, you need to install the curl tool for making requests to Kaspersky Security Center, and also check that the coreutils, bash, sed, grep, less, and findutils operating system tools are installed. To install these tools, use the package manager of your operating system.

For example, you can use one of the following commands:

- `sudo apt install <tool name>`
- `sudo yum install <tool name>`

Applying the kaspersky_astra_pub_key.gpg key of the solution

Before deploying Kaspersky Vulnerability Management on a device running the Astra Linux Special Edition operating system in closed software environment mode, you need to apply the key of the solution on the device.

To apply the solution key on the device:

1. Download the kaspersky_astra_pub_key.gpg solution key.
2. In the `/etc/digsig/digsig_initramfs.conf` configuration file, set `DIGSIG_ELF_MODE=1`.
3. Install the compatibility package:

```
apt install astra-digsig-oldkeys
```

4. Create a directory for the solution key:

```
mkdir -p /etc/digsig/keys/legacy/kaspersky/
```

5. Place the solution key in the directory created at the previous step:

```
cp kaspersky_astra_pub_key.gpg /etc/digsig/keys/legacy/kaspersky/
```

6. Update the RAM file system image for all operating system kernels:

```
update-initramfs -u -k all
```

7. Restart the operating system.

Deployment

Deployment of Kaspersky Vulnerability Management involves the following steps:

1. Installing and configuring the components of Kaspersky Vulnerability Management
2. Restarting the Kaspersky Security Center Web Console Plugin service
3. Enabling the solution on the Kaspersky Security Center Administration Server
4. Enabling the solution in Network Agents on assets

Installing and configuring the components of Kaspersky Vulnerability Management

Kaspersky Vulnerability Management components provide the basic functionality of the solution.

To install the Kaspersky Vulnerability Management components:

1. Open a terminal.
2. Use the `cd` command to change to the directory that contains the Kaspersky Vulnerability Management components installation package:
   - kaspersky-next-vm_<version number>_amd64.deb for installation on operating systems that use the `.deb` package format
   - kaspersky-vm-<version number> -1.x86_64.rpm for installation on operating systems that use the `.rpm` package format

3. Install the package using one of the following commands:

   - `sudo apt install ./kaspersky-vm_<version number>_amd64.deb` for operating systems that use the `.deb` package format
   - `sudo yum localinstall ./kaspersky-vm-<version number> -1.x86_64.rpm` for operating systems that use the `.rpm` package format
   - `sudo apt-get install ./kaspersky-vm-<version number>-1.x86_64.rpm` for the ALT family of operating systems

   Upon completion of the installation, a message is displayed telling you how to configure Kaspersky Vulnerability Management.

4. Start the Kaspersky Vulnerability Management setup assistant:

   ```
   sudo /opt/kaspersky/vulnerability-management/helper/helper.sh configure
   ```

   The Kaspersky Vulnerability Management setup assistant starts: Follow the steps it suggests.

5. Select the language of the solution interface and system messages: `ru` for Russian or `en` for English.
6. Read the text of the End User License Agreement.
7. Accept the terms and conditions of the End User License Agreement by typing `y`.
8. Read the text of the Privacy Policy.
9. Accept the terms of the Privacy Policy by typing `y`.
10. Read the text of the KSN Statement.
11. Indicate whether you accept the terms and conditions of the KSN Statement by typing `y` or `n`.

    The KSN functionality may not be available in the solution in the territory of the USA.

12. Enter the credentials for connecting to the previously created database of the solution:

    a. FQDN or IP address of the device hosting the created database.

    Using localhost as the device address is not allowed.

    b. Port number for connecting to the device with the created database. The default value is 5432.

    c. Database name.

    d. Name and password of the solution's user role in the database.

e. Select an SSL mode for connecting to the database: `disable`, `allow`, `prefer`, `require`, `verify-ca`, `verify-full`. For more information about each mode, refer to the PostgreSQL documentation.

f. If you selected the `require`, `verify-ca`, or `verify-full` mode at the previous step, specify the path to the database SSL certificate.

13. For access to updated databases, enter proxy server settings:

    a. Whether the use of the proxy server is enabled: `true` or `false`.

    b. If you entered `true` at the previous step, to connect to the proxy server, enter the proxy server address in the `proxy.example.com` format, the proxy server port (by default, 8080), and the user name and password for authentication on the proxy server. You can look up the values of these settings in Kaspersky Security Center Web Console in the **Settings** → **Integration** section on the **Web Console proxy server settings** tab.

14. Enter the address for accessing the Kaspersky Security Center API in the `https://ksc.example.com:13299` format.

15. Enter the address for access to the Kaspersky Security Center IAM Proxy in the `https://ksc-iam.example.com:9050` format.

16. In the line offering you to automatically create TLS certificates for Kaspersky Vulnerability Management services, enter `y` or `n`.

17. If you entered `n` at the previous step, enter the necessary information about TLS certificates and private keys for Kaspersky Vulnerability Management services:

    > TLS certificates for services must be issued specifically for Kaspersky Vulnerability Management. Private keys must be non-encrypted.

    a. Specify the type of certificates used:

    - `1` – certificates issued by a Certificate Authority
    - `2` - self-signed certificates

    b. Enter the paths to the TLS certificate and private key for the Private API service.

    c. Enter the paths to the TLS certificate and private key for the Public API service.

    d. Enter the paths to the TLS certificate and private key for the VMS service.

18. Enter the name of the user account with the VM Scanner role that you created before installing the Kaspersky Vulnerability Management components.

19. Enter the password for the VM Scanner account.

20. Enter the name of the user account with the VM Worker role that you created before installing the Kaspersky Vulnerability Management components.

21. Enter the password for the VM Scanner account.

The setup assistant performs the necessary checks and installs the components of Kaspersky Vulnerability

Management.

If any checks are unsuccessful, for example, if there is no connectivity with the Kaspersky Security Center API or user credentials turn out to be incorrect, the setup assistant will prompt you to re-enter the data.

You can also run non-interactive setup of Kaspersky Vulnerability Management using the following command:

```
sudo KVMANSWERSFILE=/path/to/file.json /opt/kaspersky/vulnerability-management/helper/helper.sh
unattendedconfigure
```

Where `/path/to/file.json` is the path to a JSON file with **Kaspersky Vulnerability Management settings**.

```
{
        "eulaAccepted":true,
        "privacyPolicyAccepted":true,
        "ksnAggreementAccepted":true|false,
        "database":
        {
                "host":"",
                "port":,
                "dbname":"",
                "dbschema":"",
                "dbuser":"",
                "dbpassword":"",
                "dbssl":
                {
                        "mode":"verify-full|disable|allow|prefer|require|verify-ca",
                        "certpath":""
                }
        },

        "proxy":
        {
                "enabled":true|false,
                "host":"",
                "port":,
                "login":"",
                "password":""
        },

        "ksc":
        {
                "apiurl":"",
                "iamproxyurl":"",
                "users":
                {
                        "worker":
                        {
                                "username":"",
                                "password":""
                        },
                        "scanner":
                        {
                                "username":"",
                                "password":""
                        }
                }
        },
        "tls":
        {
                "certificates":"generate|existing",
                "existing":
                {
                        "type":"ca|selfsigned",
                        "services":
                        {
                                "ca":
                                {
                                        "certpath":""
                                },
                                "privateapi":
                                {
                                        "certpath":"",
```

```
                        "keypath":""
                },
                "publicapi":
                {
                        "certpath":"",
                        "keypath":""
                },
                "vms":
                {
                        "certpath":"",
                        "keypath":""
                }
        }
    }
  }
}
```

Settings in the `file.json` file define the following:

`.eulaAccepted` (boolean) indicates acceptance of the End User License Agreement. Must be `true` for the configuration process to continue.

`.privacyPolicyAccepted` (boolean) indicates acceptance of the Privacy Policy. Must be `true` for the configuration process to continue.

`.ksnAggreementAccepted` (boolean) indicates acceptance of the KSN Statement; this affects the vulnerability detection module. Possible values: `true` or `false`.

> The KSN functionality may not be available in the solution in the territory of the USA.

`.database.host` (string) is the FQDN or IP address for access to the device with the created solution database.

`.database.port` (int) is the port number for access to the device with the created solution database.

`.database.dbname` (string) is the name of the solution database.

`.database.dbschema` (string) is the schema in the database that the solution will use.

`.database.dbuser` (string) is the name of the user role for the solution in the database.

`.database.dbpassword` (string) is the password of the solution's user role in the database.

`.database.dbssl.mode` (string) sets the SSL mode for connecting to the database. Possible values: `verify-full|disable|allow|prefer|require|verify-ca`. For more information about each mode, refer to the PostgreSQL documentation.

`.database.dbssl.certpath` (string) is the path to the SSL certificate of the database. Required if `.database.dbssl.mode` is set to `verify-full`, `require`, or `verify-ca`.

`.proxy.enabled` (boolean) indicates whether a proxy server must be used for access to updated databases. If `true`, all other values in the `.proxy` section must be specified. You can look up the values of these settings in Kaspersky Security Center Web Console in the **Settings** → **Integration** section on the **Web Console proxy server settings** tab.

`.proxy.host` (string) is the address of the proxy server in the `proxy.example.com` format.

`.proxy.port` (int) is the port of the proxy server (by default, 8080).

`.proxy.login` (string) is the user name for authentication on the proxy server. To connect to the proxy server anonymously, do not specify a value.

`.proxy.password` (string) is the user password for authentication on the proxy server. To connect to the proxy server anonymously, do not specify a value.

`.ksc.apiurl` (string) is the address for access to the Kaspersky Security Center API in the `https://ksc.example.com:13299` format.

`.ksc.iamproxyurl` (string) is the address for access to the Kaspersky Security Center IAM Proxy in the `https://ksc-iam.example.com:9050` format.

`.ksc.users.worker.username` (string) is the name of the VM Worker user account that you created before installing the Kaspersky Vulnerability Management components.

`.ksc.users.worker.password` (string) is the password for the VM Worker account.

`.ksc.users.scanner.username` (string) is the name of the user account with the VM Scanner role that you created before installing the Kaspersky Vulnerability Management components.

`.ksc.users.scanner.password` (string) is the password for the VM Scanner account.

`.tls.certificates` (string) is the method of specifying TLS certificates. Possible values:

- `generate`. The setup assistant creates the TLS certificates for you; you do not need to fill in the rest of the fields in the `.tls` section.
- `existing`. You must enter the paths to previously created TLS certificates and fill in the rest of the fields in the `.tls` section.

`.tls.existing.type` (string) is the type of existing TLS certificates. Possible values:

- `ca`. Certificates issued by the corporate certificate authority. All certificates must be issued by the same root and intermediate (if any) certificate authorities.
- `selfsigned`. Self-signed certificates.

`.tls.existing.services.ca.certpath` (string) is the path to the file with the certificate chain of the corporate certificate authority. Must contain certificates of the root as well as intermediate (if any) certificate authorities. Do not fill this field if `.tls.existing.type` is set to `selfsigned`.

`.tls.existing.services.privateapi.certpath` (string) is the path to the certificate file for the Private API. The file must not be encrypted.

`.tls.existing.services.privateapi.keypath` (string) is the path to the file with the Private API certificate key. The file must not be encrypted.

`.tls.existing.services.publicapi.certpath` (string) is path to the file with the certificate for the Public API. The file must not be encrypted.

`.tls.existing.services.publicapi.keypath` (string) is the path to the file with the Public API certificate key. The file must not be encrypted.

`.tls.existing.services.vms.certpath` (string) is the path to the file with the VMS certificate key. The file must not be encrypted.

`.tls.existing.services.vms.keypath` (string) is the path to the file with the VMS certificate key. The file must not be encrypted.

Immediately after the installation of components, vulnerability databases may be out of date. Internet access is required to update them automatically. If there is no internet access, you can update the vulnerability databases using the Kaspersky Update Utility.

The solution databases must be available at startup of Kaspersky Vulnerability Management components. Otherwise, the components of the solution cannot start. To solve this issue, you need to wait until the solution database is ready to accept connections and then start the solution components using the `sudo /opt/kaspersky/vulnerability-management/helper/helper.sh start` command.

If you encounter any problems while installing or configuring the solution, you can use the setup assistant.

Restarting the Kaspersky Security Center Web Console Plugin service

To update the data in the solution's management web plug-in after installing the Kaspersky Vulnerability Management components, you need to restart the Kaspersky Security Center Web Console Plugin service.

To restart the Kaspersky Security Center Web Console Plugin service, run the following command:

```
systemctl restart KSCWebConsolePlugin.service
```

Enabling the solution on the Administration Server

To enable Kaspersky Vulnerability Management on the Kaspersky Security Center Administration Server:

1. On the device where the Administration Server is installed, run the following command:

   ```
   /opt/kaspersky/ksc64/sbin/klscflag -fset -pv ".core/.independent" -s klcsff -n KLCS_FF_VMGMT_SUPPORT -t d -v 1
   ```

2. Restart the Administration Server service using the following command:

   ```
   systemctl restart kladminserver_srv.service
   ```

Enabling the solution in Network Agents of assets

The Kaspersky Security Center Network Agent is involved in taking an asset inventory and scanning of assets for compliance with profiles. The relevant functionality of Kaspersky Vulnerability Management is supported in Network Agent, but is disabled by default.

To enable Kaspersky Vulnerability Management in Network Agents:

1. In the main window of the Web Console, select **Asset management** → **Policies**.
2. Create a new policy with settings for Kaspersky Security Center Network Agent or open an existing policy. For more information about creating and configuring policies, see the *Kaspersky Security Center Linux Help*.
3. Open the **Application settings** tab of the policy.
4. In the **Software updates and vulnerabilities** section, enable the **Enable Kaspersky Vulnerability Management** toggle switch.
5. Click **Save**.
6. Set a lock for the modified policy and wait for it to be enforced.

After the created or modified policy is enforced, the Kaspersky Vulnerability Management functionality becomes enabled in Network Agents installed on devices that belong to Kaspersky Security Center's **Managed Devices** group.

Managing assets

An *asset* is a workstation or server in the organization's IT infrastructure. The solution searches for vulnerabilities and performs vulnerability management on assets.

This section describes the mechanism for obtaining asset data and the actions that you can perform with the assets.

Asset inventory

Asset inventory is the process of obtaining information about assets.

Every 60 minutes, Kaspersky Vulnerability Management contacts the Kaspersky Security Center Administration Server to import data for all managed and unallocated assets registered in Kaspersky Security Center.

Asset data is enriched by the results of software inventory, which produces information about installed applications, and scanning, which produces information about detected vulnerabilities, found open ports and services, and compliance with profiles.

Asset data is displayed in the following sections:

- In the **Vulnerability management** → **Assets** section, a list of assets registered in Kaspersky Vulnerability Management is displayed. Clicking the link on the name of the asset opens an asset card containing the following information about the asset:
  - **General information**. Importance of the asset, which is not set by default; description and other related information; network and system parameters.
  - **Detected vulnerabilities**. Information about vulnerabilities detected on the asset.
  - **Installed software**. List of applications installed on the asset.
  - **Network scanning**. Network scan results, that is, open ports and services available through open ports.
  - **Compliance analysis**. Results of scanning assets for profile compliance.
- In the **Vulnerability management** → **Vulnerabilities** section, the table and vulnerability cards display the assets on which vulnerabilities were detected.
- In the **Vulnerability management** → **Software** section, an application card on the **Installed on assets** tab displays a list of assets on which the application is installed.

Assets that have already been deleted from the Administration Server are deleted from the database of the solution within 30 minutes after the next import of information about all managed and unallocated assets registered in Kaspersky Security Center.

Information about deletion of assets from the Kaspersky Vulnerability Management database is not retained.

Changing asset importance

Determining the importance of assets is necessary for asset prioritization. For example, suppose an organization has assets that belong to non-technical employees with a minimal set of permissions and tasks. Such assets are low-priority in terms of potential vulnerabilities and impact on the IT infrastructure. The importance of such assets can be considered low.

The importance of assets figures in the calculation of vulnerability elimination priority using the FSTEC method.

By default, the importance of assets is not set. You can set and change the importance of an asset.

How to set or change the importance of an asset in the table of assets

1. In the main window of the Web Console, select **Vulnerability management** → **Assets**.
2. Select check boxes to the left of the asset names to select one or more assets whose importance you want to set or change.
3. Click the **Change asset importance** button.

   This opens the **Change asset importance** window.
4. In the list of values, select the importance:
   - **Critical**
   - **High**
   - **Medium**
   - **Low**
   - **Undefined**
5. Click **Save**.

The new importance of the asset is displayed in the **Importance** column of the table of assets next to the names of the assets.

How to set or change the importance of an asset in the asset card

1. In the main window of the Web Console, select **Vulnerability management** → **Assets**.
2. Click the link with the name of the asset whose importance you want to set or change.

   This opens the asset card.
3. On the **General information** tab, in the **Importance** drop-down list, select a value:
   - **Critical**
   - **High**
   - **Medium**

- **Low**
- **Undefined**

4. Click **Save**.

The new importance of the asset is displayed in the **Importance** column of the table of assets next to the name of the asset.

Managing asset groups

You can combine assets into groups for the convenience of configuring Kaspersky Vulnerability Management tasks. You can select groups of assets at the step of selecting assets for scanning.

Kaspersky Vulnerability Management automatically adds assets to a group if they meet the conditions for inclusion in the group. In this way, addition of assets is dynamically reflected in the asset group.

You can add, edit, and delete groups of assets.

How to add a group of assets

1. In the main window of the Web Console, select **Vulnerability management** → **Groups of assets**.
2. Click **Add**.

   This opens the **Adding an asset group** window.
3. Enter the name of the group in the relevant field.
4. Select an operator for the inclusion conditions:
   - **AND**. If you add multiple conditions, they are joined by a logical AND. Selected by default.
   - **OR**. If you add multiple conditions, they are joined by a logical OR.
5. Create one or more conditions for inclusion in the asset group:
   a. For each condition, select an asset parameter and an operator, then enter a value of the parameter.
   b. Click the **Add a condition** button to add as many additional conditions as you need.
6. Click the **Check conditions** button to get a list of assets that satisfy the conditions.
7. If necessary, adjust the inclusion conditions for the asset group. You can delete all conditions by clicking the **Clear all** button.
8. Click **Save**.

The asset group is displayed in the **Groups of assets** table.

How to edit a group of assets

1. In the main window of the Web Console, select **Vulnerability management** → **Groups of assets**.
2. Click the link with the name of the group of assets that you want to delete.
   This opens the asset group settings window.
3. If necessary, edit the name of the group in the corresponding field.
4. If necessary, change the operator for the inclusion conditions:
   - **AND**. If you add multiple conditions, they are joined by a logical AND.
   - **OR**. If you add multiple conditions, they are joined by a logical OR.
5. If necessary, change the inclusion conditions for the asset group:
   - You can delete all conditions by clicking the **Clear all** button.
   - For each condition, select an asset parameter and an operator, then enter a value of the parameter.
   - Click the **Add a condition** button to add as many additional conditions as you need.
6. Click the **Check conditions** button to get a list of assets that satisfy the conditions.
7. If necessary, adjust the inclusion conditions for the asset group.
8. Click **Save**.

How to delete groups of assets

1. In the main window of the Web Console, select **Vulnerability management** → **Groups of assets**.
2. Select check boxes next to the names of asset groups that you want to delete.
3. Click the **Delete** button.

Groups of assets are deleted.

If all asset groups specified in the scope of the Kaspersky Vulnerability Management task have been deleted by the time the task is run, the task ends with an error.

If some of the asset groups specified in the scope of the Kaspersky Vulnerability Management task have been deleted by the time the task is run, the task is performed only for the asset groups that still exist.

Software inventory

Software inventory is the process of obtaining data about the software that is installed on the assets registered in the Kaspersky Vulnerability Management database.

Kaspersky Vulnerability Management uses the software detection module (VMI) to obtain information about applications installed on assets from the following sources:

- Administration Server. To obtain information about software installed on assets, the Kaspersky Vulnerability Management functionality must be enabled on Kaspersky Security Center Network Agents installed on the assets.
- Results of agentless white-box scanning.

Information about the software is displayed in the following section:

- The **Vulnerability management** → **Software** section displays a list of applications installed on assets registered in Kaspersky Vulnerability Management. Clicking the link on the name of the application opens an application card that contains a description of the application, information about the software vulnerabilities found in the application, and the assets on which the solution is installed.
- In the **Vulnerability management** → **Vulnerabilities** section, the table and vulnerability cards display applications in which vulnerabilities were detected.
- In the **Vulnerability management** → **Assets** section, the device cards on the **Installed software** tab display the applications installed on the assets.

Kaspersky Vulnerability Management can get information about the following **software**:

Kaspersky Vulnerability Management supports getting an inventory of software listed above installed on assets running the following **operating systems**:

- ALT Server 10
- RED OS 8.0
- Astra Linux Common Edition (operational update 2.12)
- Astra Linux Special Edition RUSB.10015-01 (operational update 1.8.5) in closed software environment mode
- CentOS Stream 10.0
- Rocky Linux 9.6
- Ubuntu Server 22.04 LTS
- Ubuntu Server 24.04 LTS
- Windows 10 versions 1507, 1607, 1809, 21H2, 22H2
- Windows 11 25H2
- Windows Server 2025

All operating systems listed above are 64-bit.

Configuring and running a scan

This section describes the configuring and running of scan tasks, and the results of scan tasks.

About scan types

Kaspersky Vulnerability Management has the following tasks for scanning assets for vulnerabilities and analyzing the compliance of assets with benchmarks:

- *Agentless white-box scanning*. Kaspersky Vulnerability Management uses this task to obtain information about the software installed on assets and looks for vulnerabilities in it. To run the task, you need to configure access to data of assets inside their operating systems using a user account; you do not need to install a scanning agent on the assets.

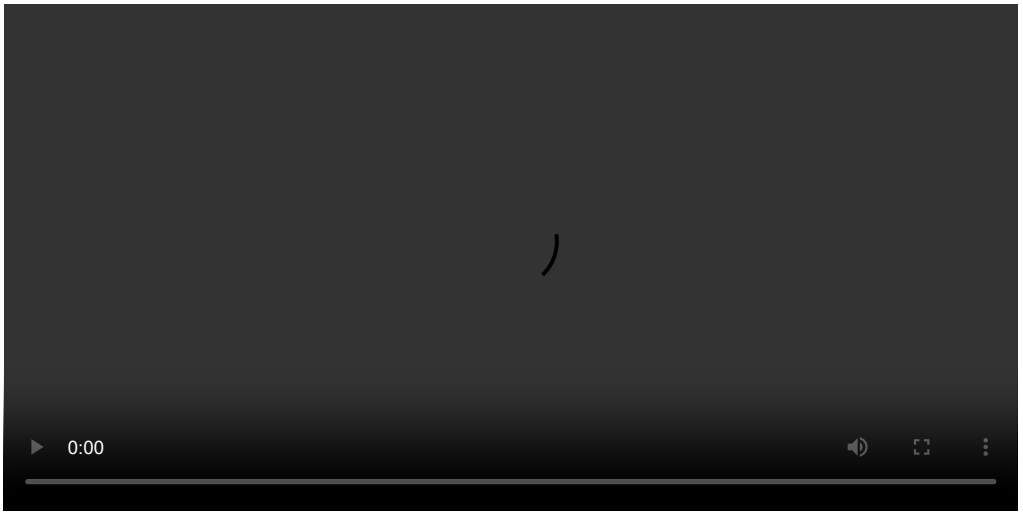
- *Agentless black-box scanning*. Kaspersky Vulnerability Management uses this task to scan for open ports, services accessible on the open ports, and service vulnerabilities. To run this task, you do not need to install the scanning agent on assets or use user credentials on assets.

- *Vulnerability scanning* Kaspersky Vulnerability Management uses this task to perform a vulnerability scan of the software registered in the solution databases.

- *Scanning assets for profile compliance* Kaspersky Vulnerability Management uses this task to check assets for compliance with internal, industry and regulatory security requirements.

Creating and configuring scan tasks

For an overview of configuring general scan task settings, you can watch the video below.



To create and configure a scan task:

1. In the main window of Web Console, select **Tasks**.

2. Click **Add**.

   This starts the New task wizard. Click the **Next** button to proceed to the next step.

3. In the **Application** drop-down list, select **Kaspersky Vulnerability Management**.

4. In the **Task type** drop-down list, select one of the tasks:

   - *Agentless white-box scanning*.
   - *Agentless black-box scanning*.
   - *Vulnerability scanning*.
   - *Scanning assets for profile compliance*.

5. Specify the name for the task that you are creating. The task name cannot exceed 100 characters and may not contain special characters ("*<>?\:|).

6. In the **Devices to which the task will be assigned** section, select **Specify device addresses manually or import addresses from a list**.

7. At the **Task scope** step, click the **Add devices** button.

8. In the displayed window, specify the device on which Kaspersky Vulnerability Management components are installed in any convenient way.

9. Click **Add**.

10. At the **Assets to scan** step, specify the assets on which you want the solution to perform the vulnerability scan or compliance analysis. The following options are available depending on the task:

    - **Selected assets**. A vulnerability scan or compliance analysis is performed on assets added to the list using the **Add assets** button.
    - **Groups of assets**. A vulnerability scan or compliance analysis is performed on assets belonging to groups of assets added to the list by clicking the **Add groups** button.
    - **All assets**. The vulnerability scan will be performed on all assets registered in the database of the solution at the time the task is started.

11. Depending on the task being created, specify the following settings:

- **For the Agentless white-box scanning task**

  For an overview of configuring the *Agentless white-box scanning* task, you can watch the video below.



  a. At the **SSH connection settings** step, in the **Username** field, enter the name of the user account that the Network scanner will use to connect to the assets selected for scanning at the **Assets to scan** step.

  b. In the **SSH port** field, enter the port number for the SSH connection to the assets to be scanned. The default setting is 22.

  c. Download the public key by clicking the **Download** button.

  d. On the assets to be scanned, create a user account with the name specified at step 9a. For example, you can use the `adduser` or `useradd` command.

  This user account must have sufficient privileges to run the `rpm` or `dpkg` commands.

  Authentication with a public key must be enabled in the SSH server settings: the `sshd_config` configuration file in the `/etc/ssh` or `/etc/ssh/sshd_config.d` directory must contain the `PubkeyAuthentication yes` line. If you have edited the configuration file, you must restart the SSH service to apply the new settings. For example, you can run the `sudo systemctl restart ssh` command.

  e. On the assets to be scanned, place the public key downloaded at step 9c of these instructions in the home directory of the created user account (for example, `/home/<user name>/.ssh/authorized_keys`).

  f. At the **Additional settings** step, if necessary, modify the **Number of connection attempts** setting to make the network scanner attempt to connect to the asset the specified number of times to scan it. Possible values: from 1 to 20 The default setting is one attempt.

  g. If necessary, edit the **Interval between attempts (sec.)** setting, which sets how long the Network scanner waits before trying again to connect to and scan an asset after an unsuccessful attempt to connect to that asset. Possible values: from 1 to 100000 The default setting is one second.

  h. If necessary, edit the **Maximum number of concurrent requests** setting, which sets the number of SSH requests that the Network scanner makes simultaneously to assets on the port specified at step 9b of these instructions. Possible values: from 1 to 100000 The default value is 10000.

  > This setting affects the performance of the solution. To keep the solution operating normally, we recommend keeping in mind the hardware of the device on which the Kaspersky Vulnerability Management components are installed.

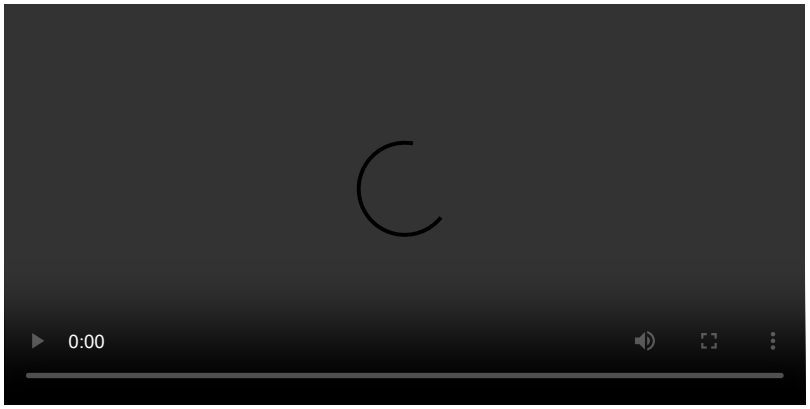
- **For the Agentless black-box scanning task**

  For an overview of configuring the *Agentless black-box scanning* task, you can watch the video below.

- **For the Scanning assets for profile compliance task**

  For an overview of configuring the *Scanning assets for profile compliance* task, you can watch the video below.

  Under **Compliance profiles**, select check boxes next to the names of profiles that you want to use to analyze assets.

  By clicking the **Refresh** button, you can force a refresh of the list of compliance profiles displayed in the task. Without user intervention, this list is synchronized with the list under **Vulnerability management** → **Compliance analysis** → **New profiles only** using a background synchronization task.

  Using the **Show selected profiles only** toggle switch, you can display all available compliance profiles or only selected profiles.

12. At the **Configure task schedule** step, select a schedule option.

13. Configure advanced settings of the task and its launch schedule.

    For more information about these settings of the group task, refer to the Kaspersky Security Center Help.

14. In the **Finish task creation** window, click the **Finish** button.

The created task is displayed in the **Tasks** section in the **Tasks** list and is passed to the Kaspersky Vulnerability Management background task service after synchronization with the Kaspersky Security Center Administration Server.

The background task service will start the created task on schedule or when a user runs it manually.

The default values for the task launch settings are specified in the configuration file of the background task service, `/etc/kaspersky/vulnerability-management/configs/worker.json`.

You can start and stop tasks in Kaspersky Security Center Web Console from the list of tasks in the **Tasks** section.

Copying, moving, pausing, and resuming Kaspersky Vulnerability Management tasks is not supported.

Agentless white-box scanning

Agentless white-box scanning can be performed only on Linux assets.

Using the *Agentless white-box scanning* Kaspersky Security Center group task, Kaspersky Vulnerability Management obtains information about software installed on assets and scans it for vulnerabilities. To run the task, you need to configure access to asset data using a user account; you do not need to install a scanning agent on the assets.

Kaspersky Vulnerability Management connects to the assets over SSH using a user account previously created on the assets. The public and private keys are generated during the installation of the solution components.

If your organization requires SSH key rotation, or if the SSH keys have been compromised, you can update the keys.

How to update SSH keys

1. Use any available means to generate a public/private key pair. The certificate files with the public and private key must be named whitebox.pub and whitebox.key, respectively.

2. Place the generated keys on the asset with the installed solution components in the `/etc/kaspersky/vulnerability-management/secrets/` directory.

3. Add the generated public key for the assets to be scanned to the `authorized_keys` file in the home directory of the user account that was created while configuring the task. For example, the `/home/<user name>/.ssh` directory.

While the *Agentless white-box scanning* task is running, the background task service may consume a lot of RAM if the assets being scanned are infected and respond to attempts to connect to them in an excessive way. To reduce the likelihood of exhausting memory, in the task settings, we recommend not including assets that you can scan for vulnerabilities using other types of scans, or increasing the period of the task.

Results of the software inventory data collection is displayed in the following sections:

- The **Vulnerability management** → **Software** section displays a list of applications installed on assets registered in Kaspersky Vulnerability Management. Clicking the link on the name of the application opens an application card that contains a description of the application, information about the software vulnerabilities found in the application, and the assets on which the solution is installed.

- In the **Vulnerability management** → **Vulnerabilities** section, in vulnerability cards on the **Affected software** tab, the list of applications with detected vulnerabilities is displayed. In the **Vulnerability management** → **Assets** section, the device cards in the **Installed software** section display applications installed on the assets.

Vulnerability scan results are displayed in the following sections:

- The **Vulnerability management** → **Vulnerabilities** section displays a list of detected vulnerabilities. Clicking the link in the vulnerability name opens the vulnerability card. The **Detection method** field in vulnerability cards displays **White box**.
- In the **Vulnerability management** → **Assets** section, on asset cards on the **Detected vulnerabilities** tab. The **Detection method** field in asset cards displays **White box**.
- In the **Vulnerability management** → **Software** section, in application cards on the **Detected vulnerabilities** tab. The **Source** field of application cards displays **White box**.

Agentless asset black-box scanning

Hackers can use open ports to penetrate the network and intercept traffic or launch attacks. Services that have open access ports may be vulnerable.

Using the *Agentless black-box scanning* group task of Kaspersky Security Center, Kaspersky Vulnerability Management can perform the following actions without installing agents and without using credentials on the assets being scanned (in black box mode):

- Scan assets for open ports and services that are accessible through open ports.
- Scan services accessible on open ports for vulnerabilities.

The results of the *Agentless black-box scanning* task are displayed after the background tasks have been processed by the service, which takes some time. You can speed up the processing of results by enabling the **Process results immediately after task completion** setting in the advanced task settings.

The application responsible for the security of an asset may interpret black-box scanning of the asset as a network attack. In this case, in the settings of the security application, add the IP address of the device on which the Kaspersky Vulnerability Management components are installed to exclusions. For example, in Kaspersky Endpoint Security for Linux, you can do this in Network Threat Protection settings.

The task results are displayed in the following sections:

- The **Vulnerability management** → **Vulnerabilities** section displays a list of detected vulnerabilities. Clicking the link in the vulnerability name opens the vulnerability card. The **Detection method** field in vulnerability cards displays **Black box**.
- In the **Vulnerability management** → **Assets** section, in the **Open ports** and **Services** columns (not displayed by default), on asset cards on the **Detected vulnerabilities** and **Network scanning** tabs. The **Detection method** field in asset cards displays **Black box**.
- In the **Vulnerability management** → **Software** section, in application cards on the **Detected vulnerabilities** tab. The **Source** field of application cards displays **Black box**.

Scanning for vulnerabilities in system and application software

Kaspersky Vulnerability Management performs a vulnerability scan on the software registered in the database using the *Vulnerability scanning* group task of Kaspersky Security Center.

As part of the *Vulnerability scanning* task, Kaspersky Vulnerability Management performs a scan for vulnerabilities in the following types of software: operating systems, Windows components, applications, and software libraries.

After the task is completed, Kaspersky Vulnerability Management recalculates the elimination priority for all vulnerabilities registered in the solution database.

Vulnerability scan results are displayed in the following sections:

- The **Vulnerability management** → **Vulnerabilities** section displays a list of detected vulnerabilities. Clicking the link in the vulnerability name opens the vulnerability card. The **Detection method** field in vulnerability cards displays **Agent**.
- In the **Vulnerability management** → **Assets** section, on asset cards on the **Detected vulnerabilities** tab. The **Detection method** field in asset cards displays **Agent**.
- In the **Vulnerability management** → **Software** section, on software cards on the **Detected vulnerabilities** tab. The **Source** field of application cards displays **Agent**.

Scanning assets for profile compliance

Kaspersky Vulnerability Management analyzes the compliance of assets with internal, industry and regulatory security requirements using the Kaspersky Security Center group task *Scanning assets for profile compliance*.

To run the *Scanning assets for profile compliance* task, the Kaspersky Vulnerability Management functionality must be enabled on Kaspersky Security Center Network Agents installed on the assets.

The **Vulnerability management** → **Compliance analysis** → **Profiles** section displays information about compliance profiles supplied with the databases of the Compliance analysis module or imported by the user.

In the settings of the *Scanning assets for profile compliance* task, you can select compliance profiles for Linux operating systems, even if the task scope contains Windows assets. It works the same the other way round as well.

The results of the analysis of assets for compliance with profiles are displayed in the following sections:

- In the **Vulnerability management** → **Compliance analysis** → **Compliance control** section.
- In the **Vulnerability management** → **Assets** section, on asset cards on the **Compliance analysis** tab.

Importing custom compliance profiles

Compliance profiles are supplied with Kaspersky Vulnerability Management and, by default, are updated once a day. You can import compliance profiles prepared in any other source.

To import compliance profiles:

1. In the main window of Web Console, select **Vulnerability management** → **Compliance analysis** → **Profiles**.
2. Click **Import**.
3. In the **Import profiles** window, download the ZIP archive with XML files of compliance profiles.

   The ZIP archive must contain a number of * -oval.xml files and a * -xccdf.xml file. The content of the *-xccdf.xml file must follow the XCCDF format.

4. If the uploaded archive contains compliance profiles that already have been uploaded, select an import option:
   - **All profiles**. New compliance profiles are added to the list, and previously added compliance profiles are updated.
   - **New profiles only**. New compliance profiles are added to the list, but previously added compliance profiles are not updated.
5. Click **Import**.

The imported compliance profiles appear in the table. Clicking the link in the name of a compliance profile opens a card with compliance profile information.

Analysis and prioritization of vulnerabilities

This section provides information about vulnerability cards and how to manage reports that you can use to analyze and prioritize vulnerabilities.

Viewing detected vulnerabilities

Vulnerabilities detected as a result of a scan are displayed in the following sections:

- The **Vulnerability management** → **Vulnerabilities** section displays a list of detected vulnerabilities. Clicking the link in the vulnerability name opens the vulnerability card. You can customize the display of the list of vulnerabilities using the buttons above the list of vulnerabilities:
  - **Grouped by vulnerabilities**. Unique vulnerabilities are listed. For each vulnerability, all assets and software affected by the vulnerability are displayed.
  - **No grouping**. The list contains unique combinations of vulnerability, affected asset, and affected software. In such a list, the same vulnerability may appear more than once.
- In the **Vulnerability management** → **Assets** section, on asset cards on the **Detected vulnerabilities** tab.
- In the **Vulnerability management** → **Software** section, on software cards on the **Detected vulnerabilities** tab.

Clicking the link with the name of the vulnerability opens the vulnerability card. Depending on how the list of vulnerabilities is displayed (**Grouped by vulnerabilities** or **No grouping**), the following data can be displayed on the vulnerability card:

- Name of vulnerability.
- Elimination priority calculated in accordance with the FSTEC method; information about the asset and software affected by the vulnerability (when displaying the list of vulnerabilities without grouping). Information about assets and software that affected by the vulnerability (when displaying the list of vulnerabilities grouped by vulnerability).
- ID of the vulnerability in the AO Kaspersky Lab's threat and vulnerability database, AO Kaspersky Lab's description of the vulnerability, and a description of the vulnerability from the vulnerability information provider (if any).

  AO Kaspersky Lab reproduces the description of the vulnerability from the vulnerability information provider without checking for accuracy and completeness, and without changes or redactions.
- External IDs of the vulnerability.
- The date of publication of the vulnerability description in the external source, the date and time of the first detection of the vulnerability, the date and time of the last detection of the vulnerability, and the method(s) used to detect the vulnerability (when displaying the list of vulnerabilities without grouping). The date of publication and method(s) used to detect the vulnerability (when displaying the list of vulnerabilities grouped by vulnerability).
- Severity and score in accordance with CVSS v2, CVSS v3, and CVSS v4 standards.

- Information about vulnerability exploitation risk, including EPSS probability and EPSS percentile.
- Recommended actions to eliminate the vulnerability.
- Links to public sources that describe the vulnerability or vulnerabilities of this type.

Managing reports

Reports present the data obtained by Kaspersky Vulnerability Management in a structured and easy-to-understand form. Reports can help you analyze data and assess the protection of your organization's IT infrastructure and inform managerial decisions.

All reports are generated based on report templates. Report templates can be either predefined or custom.

You can manage reports and report templates in the **Vulnerability management** → **Reports** section, on the **Reports** and **Templates** tabs.

Creating and editing report templates

You can perform the following actions with report templates:

How to create a report template

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.
2. Select the **Templates** tab.
3. Click **Add**.
   This opens the New report template wizard. Click the **Next** button to proceed to the next step.
4. Specify the basic settings of the report template:
   a. Select the type of report template:
      - **Vulnerabilities report**
      - **Assets report**
   b. Enter a name for the report template.
   c. If necessary, select the **Generate report immediately after template creation** check box to enable the generation of a report after creating a template.

5. Set the data collection scope for reports that will be generated based on the template in one of the following ways:

   - Under **Scan**, select **Selected assets** and create a list of assets whose data you want to be included in reports based on the template you are creating:

     a. Click **Add assets**.

     b. This opens the **Adding assets** window; in that window, select assets as necessary.

     c. Click **Save**.

     The selected assets are displayed in the table at the **Data collection scope** step.

   - Under **Scan**, select **All assets** to have the data of all assets registered in the solution database at the time of the report generation included in the template that you are creating.

6. Specify settings for the presentation of data in reports that will be generated based on the report template:

   a. Add or remove columns in the report using buttons.

   b. Enable or disable sorting by columns and filtering data in columns by clicking the **Edit** button.

7. Click **Save**.

The created report template is displayed on the **Templates** tab.

How to create a report template by copying an existing report template

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.
2. Select the **Templates** tab.
3. Select a report template using the check box next to the name of the template.
4. Click the **Copy** button.

   This opens the New report template wizard with the settings of the copied report template. Click the **Next** button to proceed to the next step.

5. If necessary, you can change the basic settings of the report template (except for the type of report template).
6. If necessary, change the data collection scope for reports that will be generated from the template.
7. If necessary, edit the settings for the presentation of data in reports that will be generated based on the template.
8. Click **Save**.

The created report template is displayed on the **Templates** tab.

How to edit a report template

1. Open the properties of a report template in any way.

   - On the **Templates** tab:

     a. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

     b. Select the **Templates** tab.

     c. Follow the link in the name of the report template.

   - On the **Reports** tab:

     a. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

     b. Select the **Reports** tab.

     c. Select a report using the check box to the left of its name.

     d. Click the **Go to template** button in the upper part of the table.

     > If the report template has been deleted, the **Go to template** button is unavailable.

   - From a report:

     a. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

     b. Select the **Reports** tab.

     c. Follow the link in the title of the report.

     d. This opens the report; click the **Go to template** button in the upper-left corner.

     > If the report template has been deleted, the **Go to template** button is unavailable.

2. In report template settings window, make your changes as necessary.

3. In the window with the report template settings, click the **Save** button.

Generating reports from templates

You can use report templates to generate reports in the following ways.

How to generate a report immediately after creating a report template

1. When creating or editing a report template, at the **General information** step, select the **Generate report immediately after template creation** check box.

2. Follow the steps of the wizard to configure the report template.

3. When creating or editing a report template, at the **Data presentation** step, click the **Save** button.

   A report is generated based on the saved report template.

How to start report generation on the Templates tab

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.
2. Select the **Templates** tab.
3. Select one or more report templates using the check boxes to the left of their names.
4. Click the **Generate** button in the upper part of the table.

Reports are generated from the selected report templates.

How to start report generation in the properties of a report template

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.
2. Select the **Templates** tab.
3. Click the link with the name of the report template from which you want to generate a report.
4. On the **General information** tab, click the **Generate** button.

The generated reports are displayed on the **Reports** tab and in the properties of the corresponding report templates on the **General information** tab under **Generated reports**.

> The **Vulnerability name**, **Description**, and **Recommendations to fix vulnerabilities** fields are displayed in the language of the Kaspersky Security Center Web Console user interface that was used when creating the report. Changing the UI language does not change the language of these fields.

By default, generated reports are retained in the database of the solution for 30 days. The time remaining until the report is deleted is displayed on the **Reports** tab in the table of reports in the **Will be deleted in** column. You can change the report retention period using the configuration file. You may need the saved reports at the vulnerability elimination assessment step.

If necessary, you can stop the report generation on the **Reports** tab by clicking the **Stop** button in the upper-left corner of the table of reports.

Exporting reports

You may need to export reports to files to provide them to your management or other interested parties that may not have access to Kaspersky Vulnerability Management, as well as to assess the elimination of vulnerabilities.

How to export reports on the Reports tab

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.
2. Select the **Reports** tab.
3. Select check boxes to the left of report names to select one or more reports.

4. In the drop-down menu of the **Export** button, select the file format to which you want to export the reports: XLSX or JSON.

   A window is opened in which you can save the report or an archive, if you are exporting multiple reports.

5. If necessary, edit the name of the report or archive.

6. Specify the path where you want to save the report file or an archive.

7. Click **Save**.

How to export a report from the report properties

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

2. Select the **Reports** tab.

3. Follow the link in the title of the report that you need.

   This opens the selected report.

4. In the drop-down menu of the **Export** button, select the format of the file to which you want to export the report: XLSX or JSON.

   This opens the window for saving the report.

5. If necessary, edit the name of the report.

6. Specify the path where you want to save the report file.

7. Click **Save**.

Report export does not work when the number of records exceeds one million. To resolve this issue, we recommend reducing the number of fields (columns) in a report.

Deleting reports and report templates

How to delete reports on the Reports tab

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

2. Select the **Reports** tab.

3. Select one or more reports using the check boxes to the left of their names.

4. Click the **Delete** button in the upper part of the table.

How to delete reports in the properties of a report template

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

2. Select the **Templates** tab.

3. Click the link with the name of the template for which you want to delete reports.

4. On the **General information** tab, under **Generated reports**, select check boxes next to the names of the reports that you want to delete.

5. Click the **Delete** button.

How to delete templates on the Templates tab

1. In the main window of the Web Console, select **Vulnerability management** → **Reports**.

2. Select the **Templates** tab.

3. Select one or more report templates using the check boxes to the left of their names.

4. Click the **Delete** button in the upper part of the table.

Managing report data

The data presented in a report depends on the type of the report.

A *report on vulnerabilities* contains a list of vulnerabilities detected on the assets specified in the report template, and information on how common the vulnerabilities are in the organization's IT infrastructure. A report on vulnerabilities also contains the following histograms:

- The number of vulnerabilities for each severity level.
- The most severe vulnerabilities according to CVSS v3.
- The most common vulnerabilities.

An *assets report* contains a list of assets on which vulnerabilities were found, with details of the detected vulnerabilities. An assets report also contains the following histograms:

- The number of assets with detected vulnerabilities for each asset importance level.
- Assets with the largest number of detected vulnerabilities.

You can perform the following actions with report data:

- Export one or more reports to files.
- Search for data in the report. In the search bar above the report table, enter your search text and click the icon.
- Sort report data. Click the header of a column in the report table to sort the report data by that column.

Vulnerability elimination

At this stage, you should take measures to eliminate the vulnerabilities or prevent their exploitation.

At this step, you need to:

1. Coordinate the installation of updates of vulnerable applications with your IT department.
2. Draw up an update plan.
3. Test the updates.
4. Installing updates using Kaspersky Security Center:
    a. For each of the vulnerable applications, in Kaspersky Security Center Web Console, create an *Install application remotely* group task with the installation package of the updated application version with the vulnerability fixed.

    For more information about creating an *Install application remotely* task, refer to the Kaspersky Security Center Help.

    b. Run the created *Install application remotely* tasks.
5. Develop security measures to compensate.

Assessing vulnerability elimination

At this step, you analyze information about the elimination of vulnerabilities.

At this step, you need to:

1. Generate reports for assets and/or vulnerabilities and export the reports.
2. Update previously generated reports after the time allotted for fixing vulnerabilities and export the reports.
3. Compare exported reports and assess the elimination of vulnerabilities.

Administration

This section describes known limitations and provides instructions for advanced configuration and uninstallation of the solution.

Known limitations

Kaspersky Vulnerability Management has some limitations that are non-critical for the operation of the solution:

- [8330927] For some vulnerabilities detected as a result of the *Agentless black-box scanning* task, only limited information is displayed.
- [8609559] Copying, moving, pausing, and resuming Kaspersky Vulnerability Management tasks is not supported.
- [9512558] In the cards of software installed on Linux assets, on the **Installed on assets** tab, the **Path to file** field is not filled.
- [10736272] Asset groups are not supported in data collection scopes for reports.
- [9180441] The scope of a role for Kaspersky Vulnerability Management is not limited, even if combinations of administration groups and users have been added in the scope of the role.
- [9180441] If a Kaspersky Vulnerability Management session is active and the set of permission for that session's user account is changed, the changes take effect with a lag of up to five minutes.
- [9344429] In the properties of Kaspersky Vulnerability Management tasks, on the **Results** tab, for events for the *Completed with warning* task status, the **Description** field is not filled.
- [9380569] White-box scanning can be performed only on Linux assets.
- [9320676] When scanning RED OS assets for compliance with profiles containing a rule that verifies that the automatic file system mounting service (autofs) is disabled (for example, FSTEC of Russia Profile No. 239 for RED OS and ALT Linux), error messages may appear or incorrect scan results may be generated.
- [10504067] When you call up Help from sections of Kaspersky Next Vulnerability Management, the start page of the help opens instead of the page relevant to the context.
- [9565353] The **Vulnerability name**, **Description**, and **Recommendations to fix vulnerabilities** fields are displayed in the language of the Kaspersky Security Center Web Console user interface that was used when creating the report. Changing the UI language does not change the language of these fields.

Upgrading from version 1.0 to version 1.0.1

Upgrading Kaspersky Vulnerability Management from version 1.0 to version 1.0.1 involves the following steps:

1. **Backing up the SSH keys that are used for agentless white-box scanning**

   Back up the whitebox.key private key and the whitebox.pub public key that you can find in the `/etc/kaspersky/vulnerability-management/secrets` directory.

2. Uninstalling the components of Kaspersky Vulnerability Management 1.0

3. Uninstalling the management web plug-in for the solution version 1.0

4. Installing and configuring the components of Kaspersky Vulnerability Management 1.0.1

   During the installation process, enter the credentials for connecting to the database that you had created for Kaspersky Vulnerability Management 1.0.

5. Installing the management web plug-in for the solution version 1.0.1

6. **Restoring SSH keys from the backup copies that you created at the first step of the instructions**

   Place the whitebox.key private key and the whitebox.pub public key that you backed up at the first step of the script, into the `/etc/kaspersky/vulnerability-management/secrets/` directory.

Changing the database update period

Kaspersky Vulnerability Management uses the following database update sources:

- Database updates for the software detection module are delivered by the Kaspersky Security Center Administration Server.
- Database updates for the vulnerability detection module are, by default, delivered once a day by the database update service of the solution.

  You can change the update period for VMS module databases in the `vms.json` configuration file in the `/etc/kaspersky/vulnerability-management/configs/` directory.
- Database updates for the compliance analysis module are, by default, delivered once a day by the database update service of the solution.

  You can change the update period for VMC module databases in the `privateapi.json` configuration file in the `/etc/kaspersky/vulnerability-management/configs/` directory.
- Database updates for the Network scanner are, by default, delivered once a day by the database update service of the solution.

  You can change the update period for the Network scanner databases in the `privateapi.json` configuration file in the `/etc/kaspersky/vulnerability-management/configs/` directory.

In the territory of the USA, the update functionality and KSN functionality may not be available in the solution.

Advanced configuration of the solution using configuration files

In configuration files in the `/etc/kaspersky/vulnerability-management/configs/` directory, you can manage settings that are not available in the user interface of the solution. You can edit or add settings described in the articles below.

Viewing the configuration files requires membership in the operating system's `root` or `kvm-group` groups. Editing the configuration files requires membership in the `root` group.

To apply changes to configuration files, you need to restart the services of the solution:

```
sudo /opt/kaspersky/vulnerability-management/helper/helper.sh restart
```

worker.json

The worker.json configuration file contains settings for the Worker service

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the worker.json configuration file

Setting	Description
`ksc.insecureSkipVerify`	Setting in the `tls.Config` structure that enables or disables the validation of server SSL/TLS certificates. Possible values: `true` or `false`. If set to `true`, the HTTP connection between the background task service (Worker) and Kaspersky Security Center for import of asset information is established without a trusted certificate.
`worker.ksc.username.file`	Path to the file with the name of the user account for the connection to Kaspersky Security Center.
`worker.ksc.password.file`	Path to the file with the password of the user account for the connection to Kaspersky Security Center.
`kscsync.enabled`	This setting enables the synchronization of tasks between Kaspersky Security Center and Kaspersky Vulnerability Management. Possible values: `true` or `false`. If set to `true`, task synchronization is enabled.

Setting	Description
`kscsync.ksc.insecureskipverify`	Setting in the `tls.Config` structure that enables or disables the validation of server SSL/TLS certificates. Possible values: `true` or `false`. If set to `true`, the HTTP connection between the background task service (Worker) and Kaspersky Security Center for task synchronization is established without a trusted certificate.
`kscsync.ksc.auth.secretid`	Prefix with the path to the settings of secrets that are used for access to Kaspersky Security Center. We do not recommend changing this value because this may prevent synchronization of tasks between Kaspersky Vulnerability Management and the Administration Server.
`kscsync.ksc.baseurl`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format.
`vmdclient.insecureskipverify`	Setting in the `tls.Config` structure that enables or disables the validation of server SSL/TLS certificates. Possible values: `true` or `false`. If set to `true`, the HTTP connection between the Worker and VMS services is established without a trusted certificate.
`vmdclient.auth.secretid`	Prefix with the path to the settings of secrets that are used for access to Kaspersky Security Center.
`vmdclient.auth.ksc.url`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format for authorizing the worker service before connecting to Kaspersky Security Center.
`vmdclient.baseurl`	URL of the device on which the VMS component is installed.
`vmdclient.username.file`	Path to the file with the name of the user account for the connection to Kaspersky Security Center.
`vmdclient.password.file`	Path to the file with the password of the user account for the connection to Kaspersky Security Center.
`taskConfigs`	This section contains settings for background and system tasks.
`detect.source.priority`	This setting sets the priority of the source of the detected vulnerabilities. Relevant at the stage of vulnerability data processing after *Vulnerability scanning* (\"agent\": 0), *Agentless black-box scanning* (\"blackbox\":1), or *Agentless white-box scanning* (\"whitebox\":2).
`detect.write.worker.count`	The number of parallel processes for saving vulnerability information. The recommended value is 10.
`software.inventory.source.priority`	Priority of the software information source that is taken into account in the data merge process: results of the *Vulnerability scanning* (\"agent\": 0) or *Agentless white-box scanning* (\"whitebox\":2) tasks.
`build.source.priority`	Priority of data sources for the asset data generation background task.
`compliance.ksc.insecureSkipVerify`	Setting in the `tls.Config` structure that enables or disables the validation of server SSL/TLS certificates. Possible values: `true` or `false`. If set to `true`, the HTTP connection between the background task service (Worker) and Kaspersky Security Center for the *Scanning assets for profile compliance* task is established without a trusted certificate.
`compliance.ksc.auth.secretid`	Prefix with the path to the settings of secrets that are used for access to Kaspersky Security Center.
`compliance.ksc.baseurl`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format for authorizing the compliance analysis module before connecting to Kaspersky Security Center.
`compliance.unsqueeze.updateable.bases`	This setting enables or disables the decompression of compliance analysis module (VMC) databases before sending to Network Agent. Possible values: `true` or `false`.
`vmd.detect.asset.cache.chunk.size`	Size of the batch of assets whose SBOMs are scanned for vulnerabilities, in megabytes. The default value is 50 MB. To change the default value, you can add this setting to the configuration file.

Setting	Description
`vmd.detect.scan.chunk.size`	Size of the batch of SBOMs to be scanned, in megabytes. The default value is 100 MB. To change the default value, you can add this setting to the configuration file.
`vmd.detect.sbom.read.worker.count`	Number of parallel processes for getting an SBOM to be scanned.
`vmd.detect.save.worker.count`	The number of parallel processes for saving vulnerability information.
`scheduler.config.refresh.interval`	How often configuration file updates are applied, in minutes. The default value is one minute. To change the default value, you can add this setting to the configuration file.
`scheduler.taskconfig.sources`	Source of information about the solution's tasks. Possible values: solution database (postgres) or file (file).
`log.level`	Level of detail for events related to background and system tasks. Possible values for this setting are `debug`, `info`, `warning`, `error`. The default level is `info`. To change the default value, you can add this setting to the configuration file.
`log.writer.file.maxagedays`	Retention period of the event log for background and system tasks. The default value is 30 days. To change the default value, you can add this setting to the configuration file.
`whitebox.private.key.path`	Path to the whitebox.key private key for the *Agentless white-box scanning* task.
`tracing.enabled`	This parameter enables or disables tracing of the Worker service. Possible values: `true` or `false`.
`vmapi.client.url`	URL of the server on which the Private API component is installed.
`vmapi.client.insecureskipverify`	Setting in the `tls.Config` structure that enables or disables the validation of server SSL/TLS certificates. Possible values: `true` or `false`. If set to `true`, the HTTP connection to the server on which the Private API component is installed is established without a trusted certificate.
`vmapi.client.auth.ksc.url`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format.
`vmapi.client.username.file`	Path to the file with the name of the user account for connecting the worker service to the Private API.
`vmapi.client.password.file`	Path to the file with the password of the user account for connecting the worker service to the Private API.
`vmapi.client.auth.secretid`	Prefix that reflects the path to the settings of secrets that are used by the worker service for access to the Private API.
`db.driver`	Name of the database management system holding the database of the solution.
`db.host`	FQDN or IP address for access to the device with the created solution database.
`db.port`	Port number for access to the device with the created database of the solution.
`db.user.file`	Path to the file with the name of the user role for the solution in the database.
`db.pass.file`	Path to the file with the password of the user role for the solution in the database.
`db.name`	Solution database name.
`db.sslmode`	SSL mode for connecting to the solution database. Possible values: `disable`, `allow`, `prefer`, `require`, `verify-ca`, `verify-full`.

scanner.json

You can use the scanner.json configuration file to configure the Network scanner.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the scanner.json configuration file

Setting	Description
`scan.poller`	Settings for getting the data of the *Agentless black-box scanning* task.
`job.template`	Settings of the task for getting Nuclei templates.
`log.level`	Level of detail of Network scanner events. Possible values for this setting are `debug`, `info`, `warning`, `error`. The default level is `info`. To change the default value, you can add this setting to the configuration file.
`taskConfigs`	This section contains a list of periodic tasks of the Network scanner and their settings. You can add or delete settings for the *Agentless white-box scanning* and *Agentless black-box scanning* tasks.
`scan.workerpool.count`	Number of worker threads processing Network scanner tasks.
`scheduler.sources`	This setting selects the source of tasks for the Network scanner: `file` or solution database (`postgres`).
`vmapi.client.url`	URL of the device with the Private API component installed.
`vmapi.client.insecureskipverify`	If set to `true`, the HTTP connection between the Network scanner and Kaspersky Security Center is established without a trusted certificate.
`vmapi.client.auth.ksc.url`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format for authorizing the Network scanner before connecting to Kaspersky Security Center.
`vmapi.client.username.file`	Path to the file with the name of the service user account for connecting the Network scanner to the Private API.
`vmapi.client.password.file`	Path to the file with the password of the service user account for connecting the Network scanner to the Private API.
`vmapi.client.auth.secretid`	Prefix that reflects the path to the settings of secrets that are used by the Network scanner for access to the Private API.
`scan.udp.timeout`	UDP port timeout during execution of the *Agentless black-box scanning* task.

privateapi.json

You can use the privateapi.json configuration file to configure the Private API service.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the privateapi.json configuration file

Setting	Description
`http.server.allowedorigins`	Sets the domains from which requests can be made to the Private API.
`http.server.host`	URL of the server on which the Private API component is installed.
`http.server.port`	Port number for access to the server on which the Private API component is installed.
`http.server.tls.enable`	This setting enables SSL for the connection to the Private API. Possible values: `true` or `false`.
`http.server.tls.servercertfile`	Path to the TLS certificate file for the Private API.
`http.server.tls.serverkey`	Path to the TLS certificate key file for the Private API.
`http.auth.methods`	Authentication type for access to the Private API.
`build.source.priority`	Priority of data sources for the asset data generation background task.
`log.level`	Level of detail for Private API request events. Possible values for this setting are `debug`, `info`, `warning`, `error`. The default level is `info`. To change the default value, you can add this setting to the configuration file.
`api.ksc.insecureskipverify`	If set to `true`, the HTTP connection between the Private API service and Kaspersky Security Center for authorization is established without a trusted certificate.
`api.ksc.baseurl`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format.
`api.ksc.auth.secretid`	Prefix with the path to the settings of secrets that are used for access to Kaspersky Security Center API.
`db.driver`	Name of the database management system holding the database of the solution.
`db.host`	FQDN or IP address for access to the device with the created solution database.
`db.port`	Port number for access to the device with the created solution database.
`db.user.file`	Path to the file with the name of the user role for the solution in the database.
`db.pass.file`	Path to the file with the password of the user role for the solution in the database.
`db.name`	Solution database name.
`db.sslmode`	SSL mode for connecting to the solution database. Possible values: `disable`, `allow`, `prefer`, `require`, `verify-ca`, `verify-full`.
`whitebox.public.key.path`	Path to public key file for the *Agentless white-box scanning* task.
`report.ttl`	Retention period for reports in hours.

Setting	Description
`updater.enabled`	This setting enables or disables database updates for the compliance analysis module (VMC) and the Network scanner. Possible values: `true` or `false`.
`updater.interval`	Database update interval in hours for the compliance analysis module and Network scanner.

publicapi.json

You can use the publicapi.json configuration file to configure the Public API service.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the publicapi.json configuration file

Setting	Description
`http.server.allowedorigins`	Sets the domains from which requests can be made to the Public API.
`http.server.host`	URL of the server on which the Public API component is installed.
`http.server.port`	Port number for access to the server on which the Public API component is installed.
`http.server.tls.enable`	This setting enables SSL for the connection to the Public API. Possible values: `true` or `false`.
`http.server.tls.servercertfile`	Path to the TLS certificate file for Public API access.
`http.server.tls.serverkey`	Path to the TLS certificate key file for the Public API.
`log.level`	Level of detail for Public API request events. Possible values for this setting are `debug`, `info`, `warning`, `error`. The default level is `info`. To change the default value, you can add this setting to the configuration file.
`api.ksc.insecureskipverify`	If set to `true`, the HTTP connection between the Public API service and Kaspersky Security Center for authorization is established without a trusted certificate.
`db.sslmode`	SSL mode for connecting to the solution database via the Public API. Possible values: `disable`, `allow`, `prefer`, `require`, `verify-ca`, `verify-full`.

vms.json

You can use the vms.json configuration file to configure the VMS module.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the vms.json configuration file

Setting	Description
`updater.enable`	This setting enables or disables database updates for the vulnerability detection module (VMS). Possible values: `true` or `false`. By default, this setting is set to `true`, which means that database updates are enabled.
`updater.intervalInHours`	Database update interval in hours for the vulnerability detection module (VMS).
`ksc.url`	URL of the Kaspersky Security Center Administration Server in the `https://ksc.example.com:13299` format.
`ksc.userPath`	Path to the file with the name of the user account for the connection to Kaspersky Security Center.
`ksc.pwdPath`	Path to the file with the password of the user account for the connection to Kaspersky Security Center.
`ksc.caFilePath`	Path to the SSL certificate file for the secure connection to Kaspersky Security Center.
`ksn.enable`	This setting enables or disables interaction with KSN. Possible values: `true` or `false`. The default setting is `false`, meaning that a proxy server is not used.
`ksn.discoveryUrl`	URL of the KSN server in the `https://ksn.example.com:2443` format.
`ksn.acceptedAgreements`	Specifying the "KSN" abbreviation as the value of this setting expresses acceptance of the terms and conditions of the KSN Statement. You can read the terms and conditions of the KSN Statement in the ksn.<language ID> file, which is included in the solution distribution kit and, after installation of the solution components, can be found in the directory `/opt/kaspersky/vulnerability-management/doc/`. This setting is mandatory if interaction with KSN is enabled (`ksn.enable` is set to `true`)

updater-scanner.json

You can use the updater-scanner.json configuration file to configure updates of the Network scanner vulnerability database that is used by the *Agentless black-box scanning* task.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the updater-scanner.json configuration file

Setting	Description
`app_name`	The name of the solution for filtering files related to the solution.
`self_update.enable`	Enables or disables updates for the utility that updates the vulnerability databases. Possible values: `true` or `false`. The default setting is `true`, meaning that the vulnerability database update utility is allowed to update itself.
`type`	Specifies the type of update source. The default setting is `KasperskyLabSecure`, and it cannot be changed.
`components`	Indicates the component of the solution for which the vulnerability databases are updated. The default setting is `VMN`, meaning that the updated vulnerability databases are intended for the Network scanner.
`mandatory_components`	The update is not performed if the vulnerability database for any of the components specified here is not available. The default setting is `VMN`, meaning that if the database of vulnerabilities for the Network scanner is not available, the update is not performed.
`index`	Relative path to the index file on the remote server.
`updater_files_dir`	Directory in which the update files for the updater utility are saved. The default setting is `/var/lib/kaspersky/vulnerability-management/updater`.
`download_dir`	Directory in which vulnerability databases are saved. The default setting is `/var/lib/kaspersky/vulnerability-management/updater/bases`.
`update_bases`	Enables/disables vulnerability database updates. Possible values: `true` or `false`. The default setting is `true`, meaning that updates of the vulnerability databases are enabled. You can configure the update interval for the Network scanner vulnerability database using the `updater.interval` setting in the privateapi.json configuration file.
`proxy.enable`	Enables or disables using a proxy server for downloading vulnerability databases. Possible values: `true` or `false`. The default setting is `false`, meaning that a proxy server is not used.
`proxy.host`	Proxy server IP address.
`proxy.port`	Port number of the proxy server.
`proxy.u_path`	Name of the file that stores the user name for authentication at the proxy server.
`proxy.p_path`	Name of the file that stores the password for authentication at the proxy server.

updater-compliance.json

You can use the updater-compliance.json configuration file to configure updates of the Compliance analysis module vulnerability database that is used by the *Scanning assets for profile compliance* task.

We do not recommend changing any settings that are not described in the table below. This can lead to Kaspersky Vulnerability Management errors.

Parameter paths must be converted to JSON format. For example, the value of the `detect.source.priority` parameter in the configuration file must be written as follows:

```
"detect": {
    "source.priority":""
  }
```

Settings of the updater-compliance.json configuration file

Setting	Description
`app_name`	The name of the solution for filtering files related to the solution.
`self_update.enable`	Enables or disables updates for the utility that updates the vulnerability databases. Possible values: `true` or `false`. The default setting is `true`, meaning that the vulnerability database update utility is allowed to update itself.
`type`	Specifies the type of update source. The default setting is `KasperskyLabSecure`, and it cannot be changed.
`components`	Indicates the component of the solution for which the vulnerability databases are updated. The default setting is `VMC`, meaning that the updated vulnerability databases belong to the compliance analysis module.
`mandatory_components`	The update is not performed if the vulnerability database for any of the components specified here is not available. The default setting is `VMC`, meaning that if the database of vulnerabilities for the compliance analysis module is not available, the update is not performed.
`index`	Relative path to the index file on the remote server.
`updater_files_dir`	Directory in which the update files for the updater utility are saved. The default setting is `/var/lib/kaspersky/vulnerability-management/updater`.
`download_dir`	Directory in which vulnerability databases are saved. The default setting is `/var/lib/kaspersky/vulnerability-management/updater/bases`.
`update_bases`	Enables/disables vulnerability database updates. Possible values: `true` or `false`. The default setting is `true`, meaning that updates of the vulnerability databases are enabled. You can configure the update interval for vulnerability databases for the compliance analysis module and the Network scanner using the `updater.interval` setting in the privateapi.json configuration file.
`proxy.enable`	Enables or disables using a proxy server for downloading vulnerability databases. Possible values: `true` or `false`. The default setting is `false`, meaning that a proxy server is not used.
`proxy.host`	Proxy server IP address.
`proxy.port`	Port number of the proxy server.
`proxy.u_path`	Name of the file that stores the user name for authentication at the proxy server.
`proxy.p_path`	Name of the file that stores the password for authentication at the proxy server.

Using the setup assistant

You can run the setup assistant to configure the solution and perform some auxiliary actions:

```
sudo /opt/kaspersky/vulnerability-management/helper/helper.sh
[configure|unattendedconfigure|start|stop|restart|status|collect-logs|ksn-participating|setplugin|help]
```

Where:

- `configure`. The setup assistant runs interactively to let you fully configure the solution.
- `unattendedconfigure`. The setup assistant runs in non-interactive mode to let you fully configure the solution. Along with this option, you need to add `KVMANSWERSFILE=/path/to/file.json`, where `/path/to/file.json` is the path to a JSON file with **Kaspersky Vulnerability Management settings**.

```
{
        "eulaAccepted":true,
        "privacyPolicyAccepted":true,
        "ksnAggreementAccepted":true|false,
        "database":
        {
                "host":"",
                "port":,
                "dbname":"",
                "dbschema":"",
                "dbuser":"",
                "dbpassword":"",
                "dbssl":
                {
                        "mode":"verify-full|disable|allow|prefer|require|verify-ca",
                        "certpath":""
                }
        },

        "proxy":
        {
                "enabled":true|false,
                "host":"",
                "port":,
                "login":"",
                "password":""
        },

        "ksc":
        {
                "apiurl":"",
                "iamproxyurl":"",
                "users":
                {
                        "worker":
                        {
                                "username":"",
                                "password":""
                        },
                        "scanner":
                        {
                                "username":"",
                                "password":""
                        }
                }
        },
        "tls":
        {
                "certificates":"generate|existing",
                "existing":
                {
                        "type":"ca|selfsigned",
                        "services":
                        {
                                "ca":
                                {
                                        "certpath":""
                                },
                                "privateapi":
                                {
                                        "certpath":"",
                                        "keypath":""
                                },
                                "publicapi":
                                {
                                        "certpath":"",
                                        "keypath":""
                                },
                                "vms":
                                {
```

```
                                "certpath":"",
                                "keypath":""
                            }
                        }
                    }
                }
            }
```

Settings in the `file.json` file define the following:

`.eulaAccepted` (boolean) indicates acceptance of the End User License Agreement. Must be `true` for the configuration process to continue.

`.privacyPolicyAccepted` (boolean) indicates acceptance of the Privacy Policy. Must be `true` for the configuration process to continue.

`.ksnAggreementAccepted` (boolean) indicates acceptance of the KSN Statement; this affects the vulnerability detection module. Possible values: `true` or `false`.

> The KSN functionality may not be available in the solution in the territory of the USA.

`.database.host` (string) is the FQDN or IP address for access to the device with the created solution database.

`.database.port` (int) is the port number for access to the device with the created solution database.

`.database.dbname` (string) is the name of the solution database.

`.database.dbschema` (string) is the schema in the database that the solution will use.

`.database.dbuser` (string) is the name of the user role for the solution in the database.

`.database.dbpassword` (string) is the password of the solution's user role in the database.

`.database.dbssl.mode` (string) sets the SSL mode for connecting to the database. Possible values: `verify-full|disable|allow|prefer|require|verify-ca`. For more information about each mode, refer to the PostgreSQL documentation.

`.database.dbssl.certpath` (string) is the path to the SSL certificate of the database. Required if `.database.dbssl.mode` is set to `verify-full`, `require`, or `verify-ca`.

`.proxy.enabled` (boolean) indicates whether a proxy server must be used for access to updated databases. If `true`, all other values in the `.proxy` section must be specified. You can look up the values of these settings in Kaspersky Security Center Web Console in the **Settings** → **Integration** section on the **Web Console proxy server settings** tab.

`.proxy.host` (string) is the address of the proxy server in the `proxy.example.com` format.

`.proxy.port` (int) is the port of the proxy server (by default, 8080).

`.proxy.login` (string) is the user name for authentication on the proxy server. To connect to the proxy server anonymously, do not specify a value.

`.proxy.password` (string) is the user password for authentication on the proxy server. To connect to the proxy server anonymously, do not specify a value.

`.ksc.apiurl` (string) is the address for access to the Kaspersky Security Center API in the `https://ksc.example.com:13299` format.

`.ksc.iamproxyurl` (string) is the address for access to the Kaspersky Security Center IAM Proxy in the `https://ksc-iam.example.com:9050` format.

`.ksc.users.worker.username` (string) is the name of the VM Worker user account that you created before installing the Kaspersky Vulnerability Management components.

`.ksc.users.worker.password` (string) is the password for the VM Worker account.

`.ksc.users.scanner.username` (string) is the name of the user account with the VM Scanner role that you created before installing the Kaspersky Vulnerability Management components.

`.ksc.users.scanner.password` (string) is the password for the VM Scanner account.

`.tls.certificates` (string) is the method of specifying TLS certificates. Possible values:

- `generate`. The setup assistant creates the TLS certificates for you; you do not need to fill in the rest of the fields in the `.tls` section.
- `existing`. You must enter the paths to previously created TLS certificates and fill in the rest of the fields in the `.tls` section.

`.tls.existing.type` (string) is the type of existing TLS certificates. Possible values:

- `ca`. Certificates issued by the corporate certificate authority. All certificates must be issued by the same root and intermediate (if any) certificate authorities.
- `selfsigned`. Self-signed certificates.

`.tls.existing.services.ca.certpath` (string) is the path to the file with the certificate chain of the corporate certificate authority. Must contain certificates of the root as well as intermediate (if any) certificate authorities. Do not fill this field if `.tls.existing.type` is set to `selfsigned`.

`.tls.existing.services.privateapi.certpath` (string) is the path to the certificate file for the Private API. The file must not be encrypted.

`.tls.existing.services.privateapi.keypath` (string) is the path to the file with the Private API certificate key. The file must not be encrypted.

`.tls.existing.services.publicapi.certpath` (string) is path to the file with the certificate for the Public API. The file must not be encrypted.

`.tls.existing.services.publicapi.keypath` (string) is the path to the file with the Public API certificate key. The file must not be encrypted.

`.tls.existing.services.vms.certpath` (string) is the path to the file with the VMS certificate key. The file must not be encrypted.

`.tls.existing.services.vms.keypath` (string) is the path to the file with the VMS certificate key. The file must not be encrypted.

For example, `sudo KVMANSWERSFILE=/path/to/file.json /opt/kaspersky/vulnerability-management/helper/helper.sh unattendedconfigure`

- `start`. The setup assistant starts the processes of solution components.
- `stop`. The setup assistant stops the processes of solution components.
- `restart`. The setup assistant stops the processes of solution components and starts them again.
- `status`. The setup assistant outputs the status of the solution components.
- `collect-logs`. The setup assistant collects event logs and editable configuration files, archives these, and displays the path to the resulting archive.
- `ksn-participating`. The setup assistant lets you change the decision to accept the terms and conditions of the KSN Statement.

  You can read the terms and conditions of the KSN Statement in the ksn.<language ID> file, which is included in the solution distribution kit and, after installation of the solution components, can be found in the directory `/opt/kaspersky/vulnerability-management/doc/`.
- `setplugin`. The setup assistant installs and configures the plugin data storage. It may also be necessary if you need to replace the Private API certificate, change the address of the device for installing the Private API, or if the plugin data storage could not be installed or verified during installation.
- `help`. Outputs the help of the setup assistant.

Event logging

Kaspersky Vulnerability Management uses the following methods of logging events:

- System audit log. Information about user actions involving Kaspersky Vulnerability Management components and tasks.
- Logs of solution components. Information about events occuring during the operation of Kaspersky Vulnerability Management components.

System audit event log

By default, the system audit log is located at the following path: `/var/log/kaspersky/vulnerability-management/<component name>/*.log`.

System audit log records can contain the following fields:

- `ts`. UTC date and time when the action was performed.
- `msg`. **Description of the performed action with the [Audit] prefix**.
  - [Audit] Export assets to CSV
  - [Audit] Set assets importance
  - [Audit] Import compliance profile

    - [Audit] Delete compliance profiles
    - [Audit] Run task
    - Add Nuclei templates via the API ([Audit] Add templates)
    - Delete Nuclei templates via the API ([Audit] Delete templates)
    - Add agentless black-box scan results ([Audit] Add scan results)
    - [Audit] Export vulnerabilities to CSV
    - [Audit] Create report template
    - [Audit] Update report template
    - [Audit] Delete report templates
    - [Audit] Delete reports
- `component` and `version`. ID and version of the component affected by the user's actions.
- `userID`. ID of the user that performed the action.
- `actionType`. **Numeric ID of the performed action**.
    - 1. [Audit] Export assets to CSV
    - 2. [Audit] Set assets importance
    - 3. [Audit] Import compliance profile
    - 4. [Audit] Delete compliance profiles
    - 5. [Audit] Run task
    - 6. [Audit] Add templates
    - 7. [Audit] Delete templates
    - 8. ([Audit] Add scan results)
    - 9. [Audit] Export vulnerabilities to CSV
    - 11. [Audit] Create report template
    - 12. [Audit] Update report template
    - 13. [Audit] Delete report templates
    - 14. [Audit] Delete reports

- **Fields specific to the performed action**.
  - `userID`. User ID.
  - `assetIDs`. Comma-separated list of internal asset IDs.
  - `importance`. Asset importance: `0` – undefined, `1` – low, `2` – medium, `3` – high, `4` – critical.
  - `strategy`. Method for specifying the importance of assets: `include` sets importance for assets with IDs in `assetIDs`; exclude sets importance for all assets not in `assetIDs`.
  - `benchmarkID`. Value of the id attribute of the Benchmark element in the XCCDF file.
  - `benchmarkDbID`. Unique internal ID of the loaded benchmark.
  - `profileIDs`. Comma-separated unique IDs for deleted compliance profiles.
  - `executionID`. ID of the task execution session. Stored in the Kaspersky Vulnerability Management database; you can use it to find more information about the task.
  - `templateIDs`. Comma-delimited Nuclei template IDs.
  - `assetID`. Asset ID.
  - `taskID`. Task ID for the asset.
  - `parentTaskID`. Scanning task ID.

Example of a system audit log record:


```
{"level":"info","ts":"2025-11-21T16:47:53.862+0300","msg":"[Audit] Set assets
importance","component":"vm_api","version":"0.9.0.0","userID":"123","actionType":2,"importance":2,"strategy":"include
["bd2a7e69-a6d6-4194-86ab-98f79f9b283f","96d9b5a7-874c-4af0-bcfd-389e66954abf"]}
```


By default, Kaspersky Vulnerability Management retains system audit log records for 30 days.

Event logs of solution components

Events of the Network scanner, Private API, and Worker components are written to a file located by default in the `/var/log/kaspersky/vulnerability-management/<component name>` directory, where "component name" can stand for `scanner`, `privateapi`, or `worker`.

The default values of the event logging settings for each solution component are encoded and are not specified in the configuration files:

- `log.level`: event detail level. Possible values for this setting are `debug`, `info`, `warning`, `error`. The default value is `info`.
- `log.filename`: full path to the component event log.
- `log.writer.file.maxsizemb`: maximum size of the event log file in megabytes before rotation. The default value is 1000 MB.

- `log.writer.file.maxbackups`: maximum number of backup copies of the component work log retained after rotation. The default value is three.
- `log.writer.file.maxagedays`: storage period for the component log. The default value is 30.
- `log.writer.file.uncompressed`: indicates whether rotated component log files should remain uncompressed. By default, the value is set to `false`, files are compressed.
- `log.writer.file.localtime`: determines whether the system time (UTC) is used to format the timestamps in the backup files of the component operation logs. This option is enabled by default.

You can add the above parameters with different values in the component configuration files (`scanner.json`, `privateapi.json`, and `worker.json`) located in `/etc/kaspersky/vulnerability-management/configs/` directory. Parameter paths must be converted to JSON format. For example, the value of the `log.writer.file.maxsizemb` and `log.writer.file.maxbackups` settings in the configuration file must be specified as follows:

```
"log": {
  "writer.file": {
    "maxsizemb": 1000,
    "maxbackups": 3
    }
}
```

Uninstalling the solution

Uninstalling Kaspersky Vulnerability Management involves the following steps:

1. Disabling the solution in Network Agents of assets
2. Disabling the solution on the Kaspersky Security Center Administration Server
3. Uninstalling the components of Kaspersky Vulnerability Management
4. **Deleting roles and user accounts** created for the solution

   To learn more about deleting roles and user accounts, refer to the Kaspersky Security Center Help.
5. Deleting the management web plug-in for the solution via Kaspersky Security Center Web Console

Disabling Kaspersky Vulnerability Management in Network Agents of assets

Kaspersky Security Center Network Agent is involved in the asset inventory and the analysis of the compliance of assets with internal, industry, and regulatory security requirements. The necessary functionality is enabled in Network Agents of assets during the deployment of the solution.

To disable the Kaspersky Vulnerability Management functionality in Network Agents:

1. In the main window of the Web Console, select **Asset management** → **Policies**.
2. Open the policy with settings for the Kaspersky Security Center Network Agent, in which the Kaspersky Vulnerability Management functionality is enabled.
3. Select the **Application settings** tab.
4. In the **Software updates and vulnerabilities** section, disable the **Enable Kaspersky Vulnerability Management** toggle switch.
5. Click **Save**.

After the modified policy is applied, the Kaspersky Vulnerability Management functionality becomes disabled in Network Agents installed on devices that belong to Kaspersky Security Center's **Managed Devices** group.

Disabling Kaspersky Vulnerability Management on the Administration Server

To disable the functionality of Kaspersky Vulnerability Management on the Kaspersky Security Center Administration Server:

1. On the device where the Administration Server is installed, run the following command:

```
/opt/kaspersky/ksc64/sbin/klscflag -fset -pv ".core/.independent" -s klcsff -n KLCS_FF_VMGMT_SUPPORT -t d -v 0
```

2. On the device where the Kaspersky Security Center Administration Server is installed, restart the Administration Server service using the following command:

```
systemctl restart kladminserver_srv.service
```

Uninstalling the components of Kaspersky Vulnerability Management

Kaspersky Vulnerability Management components provide the basic functionality of the solution.

To uninstall the Kaspersky Vulnerability Management components:

1. On a device with Kaspersky Vulnerability Management installed, open a terminal.
2. Run one of the following commands:
   - `sudo apt remove --purge kaspersky-vm` in operating systems that use the `.deb` package format
   - `sudo dnf remove kaspersky-vm` in operating systems that use the `.rpm` package format
   - `sudo apt-get remove --purge kaspersky-vm` in the ALT family of operating systems

Uninstalling the management web plug-in for the solution

To uninstall the management web plug-in for the solution via Kaspersky Security Center Web Console:

1. In the main window of the Web Console, select **Settings** → **Web plug-ins**.

   A list of installed web plug-ins opens.
2. In the list of web plug-ins, select the check box next to Kaspersky Vulnerability Management.
3. Click the **Delete** button.
4. This opens a window asking you to save a backup copy of the plug-in before uninstalling; click **OK** to confirm or **Skip backup** if you do not want to retain a backup copy.

The Kaspersky Vulnerability Management web plug-in is uninstalled. The **Vulnerability management** section in the main menu of the Web Console is hidden.

Using the public API of the solution

To get data from Kaspersky Vulnerability Management, other solutions can use VM REST API, the public API of Kaspersky Vulnerability Management. Among solutions by AO Kaspersky Lab, Kaspersky SIEM uses the VM REST API to fetch assets and vulnerabilities data from Kaspersky Vulnerability Management.

VM REST API is an HTTP interface that allows a client application to query data from Kaspersky Vulnerability Management using requests and responses. The requests and responses are transmitted over HTTP in JSON format.

VM REST API requests are generated using methods. To earn about VM REST API methods, you can refer to this reference.

Each request to the VM REST API must include authorization using a session token and be accompanied by the `X-KSC-Session: <session_token>` header. To create a session token, you need to call the Session.StartSession method of the KSC Open API.

Requests to the VM REST API must be sent to `https://<VM API FQDN>:<port number>/<version of VM REST API>/<method name>`, where VM API FQDN is the fully qualified domain name of the server on which the Kaspersky Vulnerability Management components are installed. For example: `https://example.com:5054/v1/assets`.

An example of a request using the curl utility:

```
curl -k --location --request POST 'https://example.com:5054/v1/assets' \
--header 'Content-Type: application/json' \
--header 'X-KSC-Session: <session_token>' \
--data ''
```

Legal information

This section contains the legal information.

Licensing of the solution

Kaspersky Vulnerability Management is distributed with a limited term of use. Information on granting the user the right to use the solution is contained in the End User License Agreement.

About the End User License Agreement

The *End User License Agreement* is a binding agreement between you and AO Kaspersky Lab, stipulating the terms on which you may use the application.

Read through the terms of the License Agreement carefully before you start using the application.

You can view the terms of the License Agreement:

- By reading the text in the license.<language ID> file. This file is included in the solution distribution kit.
- During installation of Kaspersky Vulnerability Management components.

  By confirming that you agree with the End User License Agreement when installing the Kaspersky Vulnerability Management components, you signify your acceptance of the terms of the End User License Agreement. If you do not accept the terms of the End User License Agreement, the installation of Kaspersky Vulnerability Management components is terminated.
- After installing the Kaspersky Vulnerability Management solution.

  After installing the solution, you can find the file with the text of the End User License Agreement of the solution on the device in the `/opt/kaspersky/vulnerability-management/doc/license.<language ID>` directory.

About the license

The *license* is a time-limited right to use the solution, granted under the End User License Agreement.

The scope of services provided and the term of use depend on the type of license under which you are using the solution.

About the License certificate

A *license certificate* is a document that contains the following information about the license being granted:

- License ID or order number
- Information about the license user
- Information about the solution that can be activated under the provided license
- Limitation on the number of licensing units (for example, the number of events per second that can be processed)
- License validity start date
- License expiration date or validity period
- License type

Data provision

Data transferred to AO Kaspersky Lab

Data for updating the vulnerability databases. Kaspersky protects any information thus received in accordance with law and the applicable rules of Kaspersky. Data is transmitted over encrypted channels.

Data processed locally

To perform its main functions, Kaspersky Vulnerability Management can receive, store, and process the following data:

- Asset information. Kaspersky Vulnerability Management receives information about assets from the Kaspersky Security Center Administration Server. The user can see the description of assets in the interface of Kaspersky Security Center Web Console:
  - Asset parameters required to identify it: ID in the database, name and description, IPv4 or IPv6 address, DNS name and DNS domain, ID and version of Network Agent, NetBIOS name.
  - Other information about the asset: type, last-seen time on the network, date when the asset was added in the system, operating system of the asset and its architecture, operating system build number, asset importance, Windows domain, virtual machine type, motherboard vendor, motherboard firmware version.
- Information about open ports and services on the asset. Kaspersky Vulnerability Management receives the information using the Network scanner installed on the network. The user can view the scan results in the Kaspersky Security Center Web Console interface: a list of open ports, protocols for which the ports are open; services accessible on the open ports; date and time of the last asset scan.

- Results of compliance analysis of assets. The analysis is performed by the Compliance analysis module. The analysis results are transmitted by the Network Agent to Kaspersky Vulnerability Management. The user can view the results of compliance analysis in the Kaspersky Security Center Web Console interface. The following information is provided: IDs of rules and compliance profiles, names and descriptions of rules and compliance profiles, names of benchmarks, how a compliance profile was added to the database, links to regulatory documents of compliance profiles, date and time when a compliance profile was added to Kaspersky Vulnerability Management, list of covered measures, date and time of the last asset scan.

- Information about vulnerabilities detected on assets. Kaspersky Vulnerability Management obtains information about vulnerabilities using the Vulnerability Detection Module and the Network Scanner. The user can view the detected vulnerabilities in the Kaspersky Security Center Web Console interface: names and descriptions of vulnerabilities; internal and external IDs of vulnerabilities; scores and severities of vulnerabilities; elimination priorities of vulnerabilities; sources of vulnerabilities and recommendations for their elimination; existence of an exploit, PoC, and network attack vectors in vulnerabilities; trends of vulnerabilities; CWE; EPSS; date of publication; indicator of possibility of exploitation by attackers; detection method; date of first and last detection; affected software and assets on which it is installed.

- Information about software installed on assets. Kaspersky Vulnerability Management obtains software information using the software detection module. This data is transferred by the Network Agent to Kaspersky Vulnerability Management. The solution also gets software inventory data by scanning assets in white-box mode. The user can view information about detected software in the interface of Kaspersky Security Center Web Console: name and description, version, vendor, type, first-seen date and time, size, installation path, detected vulnerabilities, assets on which the software is installed.

- Log files of Kaspersky Vulnerability Management components. Log files are not automatically transferred to AO Kaspersky Lab, but the administrator can transfer these files to AO Kaspersky Lab manually upon request as part of troubleshooting Kaspersky Vulnerability Management.

- Data required for the installation of Kaspersky Vulnerability Management and integration with the Administration Server of Kaspersky Security Center.

- Data required for authorization of requests to the public API.

- Data required to run an asset scan in white-box mode.

- Information about the user's acceptance of the terms and conditions of the binding agreement with AO Kaspersky Lab.

- Authentication credentials of Kaspersky Vulnerability Management users.

- Any data that the user enters in the interface of the web plug-in for managing the Kaspersky Vulnerability Management solution through the Kaspersky Security Center Web Console.

The data listed above is stored in the database to which the user connects Kaspersky Vulnerability Management, or in configuration files. The retention period in the database is set by the user.

For more detailed information about the processing, storage, and destruction of information obtained during the use of the solution and transmitted to Kaspersky, refer to the Privacy Policy on the Kaspersky website.

Information about third-party code

Information about third-party code is provided in the `legal_notices.txt` file in the `/opt/kaspersky/vulnerability-management/doc/` directory.

Trademark notices

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

Adobe, Acrobat, ColdFusion, Photoshop, Reader are either registered trademarks or trademarks of Adobe in the United States and/or other countries.

Apache, Apache Flink, Apache HTTP Server, Apache Solr, Apache Spark, Apache Struts, Apache Tomcat, and OFBIZ are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries.

Atlassian, Atlassian Confluence are registered trademarks of ATLASSIAN PTY LTD.

CentOS, Red Hat, and Red Hat Enterprise Linux are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Chrome, Chromium, Google, Google Chrome, and Vite are trademarks of Google LLC.

Cisco, Cisco Jabber, and Jabber are registered trademarks or trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Debian is a registered trademark of Software in the Public Interest, Inc.

Docker and the Docker logo are trademarks or registered trademarks of Docker, Inc. in the United States and/or other countries. Docker, Inc. and other parties may also have trademark rights in other terms used herein.

Dropbox is a trademark of Dropbox, Inc.

EaseUS Todo Backup is a registered trademark of Chengdu Yiwo Tech Development Co., Ltd.

Electrum is a registered trademark of Electrum Technologies GmbH.

Evernote is a registered trademark of the Evernote Corporation.

Firefox, Mozilla, and Thunderbird are trademarks of the Mozilla Foundation in the U.S. and other countries.

FortiClient is a trademark of Fortinet, Inc.

Foxit is a registered trademark of Foxit Corporation.

The Grafana Word Mark and Grafana Logo are either registered trademarks/service marks or trademarks/service marks of Coding Instinct AB, in the United States and other countries and are used with Coding Instinct's permission. We are not affiliated with, endorsed or sponsored by Coding Instinct, or the Grafana community.

GITHUB is a trademark of GitHub, Inc., registered in the United States and other countries.

IrfanView is a registered trademark of Skiljan, Irfan.

iTunes is a trademark of Apple Inc.

JetBrains is a trademark of JetBrains s.r.o.

Joomla! is a registered trademark of Open Source Matters, Inc.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

MariaDB is a registered trademark of the MariaDB Corporation.

MongoDB is a registered trademark of MONGODB, INC.

MINIO is a trademark of the MinIO Corporation.

Nginx is a registered trademark of NGINX, Inc.

Excel, Internet Explorer, Microsoft, Microsoft Edge, Microsoft Teams, OneNote, Outlook, PowerPoint, Project, SharePoint, Visual Studio, Windows Server, and Word are trademarks of the Microsoft group of companies.

OpenSSL is a trademark owned by the OpenSSL Software Foundation.

openSUSE is a registered trademark of SUSE LLC.

Oracle is a registered trademark of Oracle and/or its affiliates.

PostgreSQL is a registered trademark of the PostgreSQL Global Development Group.

Radmin is a registered trademark of Famatech.

Redis is a registered trademark of Redis Ltd.

Speccy is a registered trademark of PIRIFORM SOFTWARE LIMITED.

Rocky Linux is a trademark of The Rocky Enterprise Software Foundation.

TeamViewer is a registered trademark of Teamviewer GmbH.

Telegram is a registered trademark of Telegram Messenger LLP.

Ubuntu is a registered trademark of Canonical Ltd.

WooCommerce is a trademark of Automattic Inc.

Zabbix is a registered trademark of Zabbix SIA.

2GIS is a registered trademark of DoubleGIS LLC.

Alt is a registered trademark of Alt-Invest LLC.

RED OS is a registered trademark of Red Soft LLC.

Glossary

Administration group

A set of devices combined in Kaspersky Security Center in accordance with the functions they perform and the set of Kaspersky applications installed on them. Devices are grouped to simplify administration as the group of devices can be managed as a single entity. An administration group can include other groups. For each application installed in the administration group, group policies and group tasks can be created.

Administration Server

A component of Kaspersky Security Center that centrally stores information about all Kaspersky applications installed within the corporate network. It can also be used to manage these applications.

Compliance profile

A set of rules written in accordance with the OVAL and XCCDF specifications that describe the conditions of asset compliance with internal, industry, and regulatory security requirements. Rules of a compliance profile are united by a common theme and placed in a single XML file.

CVSS

Common Vulnerability Scoring System is an open standard for measuring the vulnerabilities. CVSS defines a set of metrics and formulas for evaluating the severity of a vulnerability, with values ranging from 0 (minimum) to 10 (maximum). CVSS helps prioritize response to vulnerabilities depending on their severity.

In different versions of the standard, the evaluation methodology and the set of metrics are different.

EPSS percentile

Percentage of vulnerabilities that have the same or lower probability of exploitation.

EPSS probability

Exploitation probability for the vulnerability in the wild in the next 30 days.

Group task

A task assigned to an administration group and performed on all managed devices included in this administration group.

OVAL

Open Vulnerability and Assessment Language is an open language for describing and evaluating vulnerabilities.

SBOM

An SBOM (Software Bill of Materials) is a list of all components of an object with a description of the dependencies of the components, methods of scanning the components and authenticating their source.

Severity of the vulnerability

Indicates the severity of the threat posed by the vulnerability. Determined by the numerical CVSS score. If scores are available for multiple versions of CVSS, the score in accordance with the newest version of the standard is used.

Possible vulnerability severities:

- *Low*, corresponding to numerical CVSS scores from 0.0 to 3.9, inclusive
- *Medium*, corresponding to numerical CVSS scores from 4.0 to 6.9, inclusive
- *High*, corresponding to numerical CVSS scores from 7.0 to 9.9, inclusive
- *Critical*, corresponding to a CVSS score of 10.0

The higher the severity, the sooner it is recommended to eliminate the vulnerability.

Vulnerability

A flaw in the operating system or application that can be exploited by intruders to infiltrate an operating system or application and compromise its integrity. A large number of vulnerabilities in an operating system makes it unreliable because malware that have infiltrated the operating system can cause malfunctions of the operating system itself as well as the installed applications.

Vulnerability elimination priority

The vulnerability elimination priority according to the FSTEC methodology is an estimate that indicates how urgently a particular vulnerability must be eliminated in your IT infrastructure specifically.

The FSTEC methodology supplements the technical evaluation of the CVSS with the context of the corporate IT infrastructure. The current FSTEC methodology considers the following parameters when determining the elimination priority for a vulnerability:

- CVSS vulnerability score
- Type and number of affected assets
- Impact of the affected assets on the security of the corporate IT infrastructure
- Information about the exploitation of the vulnerability and the existence of an exploit
- Possible impact of exploitation

Possible values of the vulnerability elimination priority:

- Critical
- High
- Medium
- Low

The higher the elimination priority, the sooner it is recommended to eliminate the vulnerability.

XCCDF

Extensible Configuration Checklist Description Format is an XML-based format for describing security checklists.