

Positive Technologies Application Inspector Enterprise Edition

Version 3.6.0



Administrator Guide

POSITIVE TECHNOLOGIES

Copyright © 2020 JSC Positive Technologies. All Rights Reserved.

This document is the property of JSC Positive Technologies (hereinafter referred to as Positive Technologies) and protected by national copyright laws and international copyright treaties.

The document may not be copied or distributed in whole or in part in any form, including translation, or transmitted to third parties without the written permission of Positive Technologies.

This document may be amended without prior notice.

Positive Technologies, Positive Hack Days, PTSECURITY, MaxPatrol, XSpider, SurfPatrol, N-Scope, Positive Technologies Application Firewall, Positive Technologies Application Inspector, Positive Technologies MultiScanner, and Positive Technologies Reporting Portal are registered trademarks or trademarks of Positive Technologies.

Other trademarks used in the text are given for informational purposes only and are the exclusive property of their respective owners. Positive Technologies is not affiliated with such owners and does not make products bearing such trademarks.

Last edited: 30-Sep-20

Contents

1.	About this document.....	4
1.1.	Document conventions.....	4
1.2.	Other sources of information about PT AI Enterprise Edition.....	5
2.	About PT AI Enterprise Edition.....	6
2.1.	User roles.....	7
2.2.	General operating scenario of PT AI Enterprise Edition.....	8
3.	Hardware and software requirements.....	10
4.	Licensing.....	11
5.	Deployment of PT AI Enterprise Edition.....	12
6.	Installing the product.....	13
6.1.	Installing the PT AI Enterprise Server module.....	13
6.2.	Installing the PT AI Enterprise Viewer module.....	15
6.3.	Deploying the PT AI Enterprise Agent module.....	16
6.3.1.	Preparing for PT AI Enterprise Agent installation.....	16
6.3.2.	Installing the PT AI Enterprise Agent module.....	16
6.3.3.	Setting up the PT AI Enterprise Agent module.....	17
6.4.	Installing the AI.Shell light agent.....	18
6.4.1.	Installing AI.Shell from the package.....	18
6.4.2.	Installing AI.Shell with Windows Installer.....	19
6.4.3.	Configuring AI.Shell and starting scanning.....	20
7.	Configuring scan settings.....	22
7.1.	Configuring scanning in the configuration file.....	22
7.2.	Configuring a security policy.....	22
8.	PT AI Enterprise Edition integration into the CI process.....	26
8.1.	Configuring the TeamCity build agent.....	26
8.2.	Configuring the Jenkins build agent.....	31
8.2.1.	Jenkins basic configuration.....	32
8.2.2.	Setting up the Jenkins build agent using the plugin.....	36
9.	Contacting Technical Support.....	42
9.1.	Technical Support online.....	42
9.2.	Technical Support by phone.....	42
9.3.	Technical Support working hours.....	43
9.4.	How Technical Support processes requests.....	43
9.4.1.	Providing information for Technical Support.....	43
9.4.2.	Request types.....	43
9.4.3.	Response time and request prioritization.....	44
9.4.4.	Request processing.....	45
	Appendix A. Example configuration file.....	47
	Appendix B. Parameters for starting a scan from CLI.....	52
	Appendix C. Return codes.....	53

1. About this document

The Administrator Guide contains reference information and instructions for deployment, setup, and administration of Positive Technologies Application Inspector Enterprise Edition (PT AI Enterprise Edition). The guide does not provide instructions on how to use the main functions of the product.

The guide is intended for specialists who install, setup, and administer PT AI Enterprise Edition.

The PT AI Enterprise Edition documentation includes:

- This document
- User Guide—detailed information about product scenarios and configuring functions to solve specific tasks

In this section

[Document conventions \(see Section 1.1\)](#)

[Other sources of information about PT AI Enterprise Edition \(see Section 1.2\)](#)

1.1. Document conventions

This guide uses the following document conventions.

Table 1. Document conventions

Sample text	Description
Warning. Disabling the module decreases the level of network security	Warnings. Contain information about actions or events with potentially negative consequences
Note. You can create additional reports	Notes. Contain tips, descriptions of important special cases, and additional or reference information that might be useful
► To open the file:	The beginning of instructions is marked with a specific symbol
Click OK	Names of interface elements (for example, buttons, text boxes, and menu items) are highlighted in bold
Run the <code>Stop-Service</code> command	Command-line text and code examples that need to be entered using the keyboard are highlighted in a special font. File names and paths to files and folders are also indicated in a special font
CTRL+ALT+DELETE	Key combination. To activate the combination, the keys need to be pressed at the same time

Sample text	Description
<Application name>	Variables are enclosed in angle brackets

1.2. Other sources of information about PT AI Enterprise Edition

You can find additional information about PT AI Enterprise Edition at ptsecurity.com and on the Technical Support portal at support.ptsecurity.com.

The support.ptsecurity.com portal contains knowledge base articles, news about Positive Technologies product updates, and answers to FAQs. Create a user account on the portal to have access to the knowledge base and all news.

If you cannot find the information you require, please contact [Technical Support](#) (see [Section 9](#)).

2. About PT AI Enterprise Edition

PT AI Enterprise Edition is a distributed system that purports to automate search for vulnerabilities and undocumented features (UDF) within the secure development life cycle and the information security audit. PT AI Enterprise Edition embraces the advantages of static, dynamic, and interactive approaches to analysis.

PT AI Enterprise Edition can be integrated into the process of continuous integration (CI) on the build agents like TeamCity and Jenkins, which allows checking code for vulnerabilities at the build stage for the product under development.

PT AI Enterprise Edition supports the role-based access model (administrator, auditor, security manager). Each user has a corresponding level of data access and interacts with the product according to the assigned role.

With PT AI Enterprise Edition, software engineers can write secure code and promptly fix vulnerabilities while security managers can control the secure development process.

PT AI Enterprise Edition contributes to higher product quality and reduces time for software development and QA as well as labor intensity typical for manual analysis of vulnerabilities.

PT AI Enterprise Edition benefits:

- Role-based access (RBAC) model.
- High efficiency of vulnerability search with a low level of false positives.
- No need to deploy the application.
- Visual demonstration of vulnerabilities.
- Time saving when checking code due to incremental scanning that takes into consideration previous results.
- Excluding selected vulnerabilities from scan results by adding comments to source code.

The key features of PT AI Enterprise Edition include:

- Code analysis at the early stages of development.
- Automatic generation of an HTTP request (exploit). With the exploit, you can test a vulnerability detected on a deployed application.
- Flexible integration with an application firewall that allows generation of virtual patches for detected vulnerabilities.
- Scanning a running web application with a black-box method on a test bench. The scanner analyzes dynamic scripts, forms, parameters, headers, and other entry points through which malicious data penetrates and impacts the system.
- Support for custom search templates and custom rules to detect any constructions that contain specific business logic or any signs of UDF.
- Determining the best place in code to fix a vulnerability
- Collecting statistics on scan results and found vulnerabilities.
- Creating Jira issues to fix vulnerabilities found during scan.

In this section

[User roles \(see Section 2.1\)](#)

[General operating scenario of PT AI Enterprise Edition \(see Section 2.2\)](#)

2.1. User roles

In PT AI Enterprise Edition, each user has their own role (administrator, security manager, and auditor). The user is granted permissions according to their role.

Permissions can be global or project-specific. A scan project (hereinafter "project") is a named task that contains the results of code analysis of a particular application in a separate folder and all its subfolders. Users with global permissions can perform actions within their role across all projects. Users with project-specific permissions can perform actions only in the projects they participate in.

The administrator in PT AI Enterprise Edition has global permissions, the security manager has global and project-specific permissions, the auditor has only project-specific permissions.

The administrator role is assigned by the system administrator. There is no role for the system administrators because they are not project participants. The system administrator is responsible only for PT AI Enterprise Edition maintenance (sets up and maintains modules, integrates PT AI Enterprise Edition in the CI process).

Administrator

The system administrator assigns an administrator for all projects in PT AI Enterprise Edition.

The administrator can:

- Generate tokens for scan agents, AI.Shell light agent, and CI/CAT plugins.
- Configure the system in the web interface.
- Configure scan agents.
- Manage user permissions in the system
- Create and configure scan projects
- Manage scan results

Security manager

The administrator assigns a global security manager.

The global security manager can create new projects. In the created project, the global security manager automatically receives permissions of the project security manager. The project manager can, in turn, assign other project managers and auditors to his or her project.

The security manager is responsible for monitoring compliance of the developed software with project-specific security policy standards.

The security manager configures projects and runs scans in the PT AI Enterprise Viewer interface, manages scan results, generates reports on detected vulnerabilities.

Auditor

Administrators and security managers can assign auditors for each project.

The auditor uses the user interface to view project scan results, generates reports on detected vulnerabilities, changes their status (confirms or rejects a vulnerability).

The auditor role may also be assigned to a software engineer who runs project checks on the build agent after commits.

2.2. General operating scenario of PT AI Enterprise Edition

In general, operation of PT AI Enterprise Edition consists of the following steps:

1. Installing the system modules. The system administrator deploys the PT AI Enterprise Edition modules and additional components required for the PT AI Enterprise Agent module operation. During the PT AI Enterprise Server module installation, the system administrator assigns the administrator role.
2. Configuring system settings. The administrator configures PT AI Enterprise Edition in the web interface.
3. Configuring roles. The administrator in the web interface assigns other administrators and global managers.
4. Creating a project. The administrator or global manager creates a project in the PT AI Enterprise Viewer interface.

5. Integrating PT AI Enterprise Edition into the CI process. The system administrator configures the vulnerability search mechanism on the build agent.
6. Configuring a scan project. There are two ways of configuring a project:
 - In the configuration file. The administrator or security manager indicates scan settings in the configuration file.
 - In PT AI Enterprise Viewer. The administrator or security manager configures scan settings.
7. Running a project check for vulnerabilities in PT AI Enterprise Viewer or on the build agent.
8. If the project is checked for vulnerabilities on the build agent:
 - PT AI Enterprise Agent receives a task for code scanning on the build agent (for example, triggered by the software engineer's commit).
 - PT AI Enterprise Agent scans code for vulnerabilities and returns results to the build agent.
 - Test results are located in the log file in the build agent interface.
 - Depending on the response settings of the build agent to events received from the log file, the project build is stopped if the security policy is violated or the project build continues if the security policy is complied with.

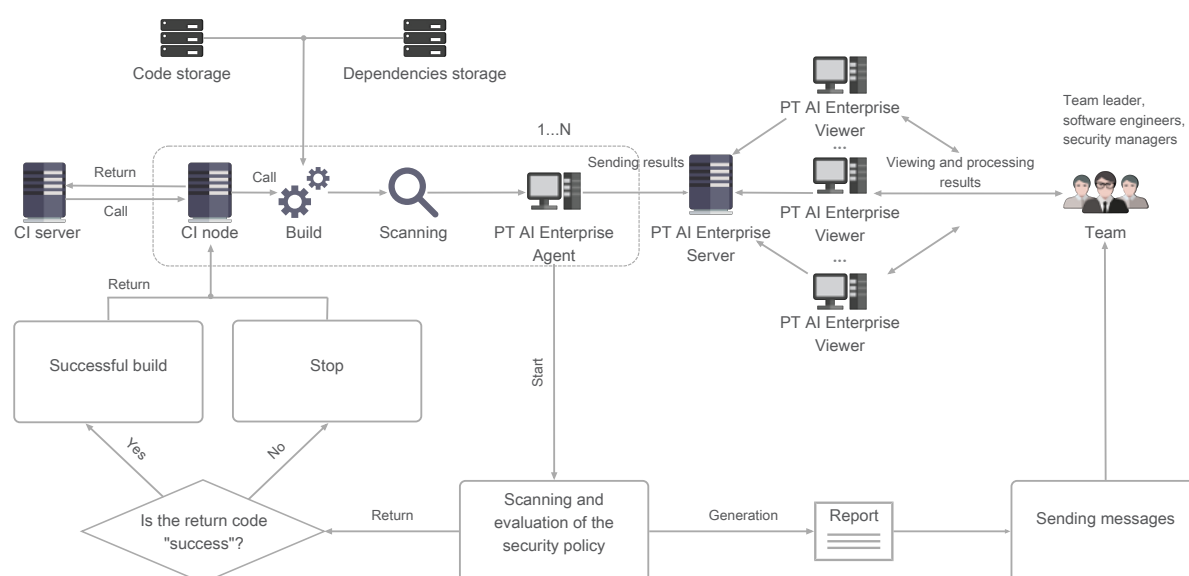


Figure 1. Using PT AI Enterprise Edition in continuous integration

9. Managing detected vulnerabilities in PT AI Enterprise Viewer. With the set of tools provided in the interface, the security manager or auditor checks and analyzes detected vulnerabilities.
10. Fixing vulnerabilities. The software developer fixes vulnerabilities in his or her development environment and sends code for rescanning (commit). Scanning continues until the project complies with the security policy.
11. Generation of the scan results report. The security manager or auditor generates a report on the number and types of vulnerabilities detected, and assesses the quality of security policy implementation in the project.

3. Hardware and software requirements

Minimum hardware and software requirements for a computer with the PT AI Enterprise Server module:

- Intel Core i7 3.2 GHz processor or similar
- 8 GB RAM
- 200 GB of free disk space
- 10 Mbps network adapter
- Screen resolution of 1366×768 pixels
- A 64-bit version of Windows Server 2012 R2 or later
- Automation tool for Windows PowerShell version 5.0 or later
- Browser: Microsoft Edge, Mozilla Firefox 46 or later, Google Chrome 50 or later

Minimum hardware and software requirements for a computer with the PT AI Enterprise Viewer module:

- Intel Core i5 2 GHz processor or similar
- 8 GB RAM
- 10 Mbps network adapter
- Screen resolution of 1366×768 pixels
- Browser: Microsoft Edge, Mozilla Firefox 46 or later, Google Chrome 50 or later

Minimum hardware and software requirements for a computer with the PT AI Enterprise Agent module:

- Intel Core i7 3.2 GHz processor or similar
- 8 GB RAM
- 10 Mbps network adapter
- Screen resolution of 1366×768 pixels
- Browser: Microsoft Edge, Mozilla Firefox 46 or later, Google Chrome 50 or later

4. Licensing

All PT AI Enterprise Edition modules are protected from unlicensed use by the Sentinel network license.

PT AI Enterprise Edition performs scanning only if a valid license is present. The validity period for the PT AI Enterprise Edition license is 1 year. When the license is expired, active scanning is terminated with return code 3. You have full access to the scan results received earlier.

The PT AI Enterprise Edition licenses differ by:

- Number of projects (10, 25, 50, 100, and limitless)
- Sets of programming languages available for project scan

License activation is performed by the administrator in the PT AI Enterprise Edition web interface.

5. Deployment of PT AI Enterprise Edition

PT AI Enterprise Edition consists of three modules that are installed separately:

- PT AI Enterprise Server. A management and connection module. It grants other modules access to PT AI Enterprise Edition data and manages it. The installation software for PT AI Enterprise Server consists of a group of services that manage the system operation, a service for the message broker server RabbitMQ, and the PostgreSQL database.
- PT AI Enterprise Agent. A module for scanning source code. It is a console application. PT AI Enterprise Agent scans source code for vulnerabilities and returns results to PT AI Enterprise Server.
- PT AI Enterprise Viewer. A module that provides an interface for user interaction with PT AI Enterprise Edition. PT AI Enterprise Viewer allows configuring and starting scanning and displays scan results.

PT AI Enterprise Server must be installed on a separate computer with the corresponding [hardware specifics \(see Section 3\)](#). It is recommended that you install PT AI Enterprise Viewer and PT AI Enterprise Agent on separate computers. PT AI Enterprise Agent conducts a resource-intensive code analysis. Due to this, if both modules are installed on the same computer, operating speed may suffer when you work in the user interface.

PT AI Enterprise Server can support operation of several instances of PT AI Enterprise Viewer and PT AI Enterprise Agent. Increasing the number of installed PT AI Enterprise Agent instances allows scanning several projects and perform their build on the CI agents at the same time.

If the company provides for continuous integration, PT AI Enterprise Agent must be installed on the same computer as the build agent. Also, if the build agents are running under a Linux operating system or docker containers are used, you need to install the AI.Shell component on them (supplied with PT AI Enterprise Edition).

The PT AI Enterprise Edition deployment is performed using the installation wizard. The file with the installation wizard must be downloaded to each computer where the system modules will be installed.

To ensure secure data transfer between modules, PT AI Enterprise Agent and PT AI Enterprise Server use the access token that must be generated in the web interface.

Note. PT AI Enterprise Edition still supports SSL certificates used in earlier versions of the product.

After PT AI Enterprise Edition is deployed, users connect PT AI Enterprise Viewer and PT AI Enterprise Agent to PT AI Enterprise Server:

- At the PT AI Enterprise Viewer start, they enter the address of the computer that hosts PT AI Enterprise Server. User authentication in the system is performed via Active Directory. After the user enters the address, a window opens with the projects that the user is granted access to by the administrator or security manager.
- Before starting PT AI Enterprise Agent, in the `aic.user.config` configuration file, specify the access token and address of the computer with the installed PT AI Enterprise Server module.

6. Installing the product

In general, installation of PT AI Enterprise Edition consists of the following steps:

1. Installing the PT AI Enterprise Server module
2. Installing the PT AI Enterprise Viewer module
3. Deploying the PT AI Enterprise Agent module:
 - Installing and configuring software required for the module operation
 - Installing the module
 - Configuring the module
4. Installing the AI.Shell light agent (if necessary)

In this section

[Installing the PT AI Enterprise Server module \(see Section 6.1\)](#)

[Installing the PT AI Enterprise Viewer module \(see Section 6.2\)](#)

[Deploying the PT AI Enterprise Agent module \(see Section 6.3\)](#)

[Installing the AI.Shell light agent \(see Section 6.4\)](#)

6.1. Installing the PT AI Enterprise Server module

► To install PT AI Enterprise Server on the computer:

1. Run the setup file.
2. In the window with the text of the End User License Agreement, read the terms and conditions of the End User License Agreement.
3. Select **I accept the agreement** if you agree with the License Agreement.
4. Click **Next**.
5. In the window **Select Destination Location**, indicate the folder for installing the module.
6. Click **Next**.

The window for general PT AI Enterprise Server configuration will open.

7. In the window, specify the following settings:
 - **Domain**. The domain name.
 - **Message queue port**. The port of the RabbitMQ server.
 - **Message queue user**. The name of a user of the RabbitMQ server.
 - **Message queue password**. The password for connecting to the RabbitMQ server.
 - **Database port**. The port of the PostgreSQL server.

- **Database name.** The name of the PostgreSQL database.
- **Database user.** The name of the PostgreSQL user.
- **Database password.** The password for connecting to the PostgreSQL database.

8. Click **Next**.

The service configuration window will open. PT AI Enterprise Edition assigns default ports for connecting to the services. If the default port is busy, PT AI Enterprise Edition will warn you about this and suggest assigning another port manually.

9. In the window, specify the following settings:

- **Host.** The IP address or DNS name of a computer that has PT AI Enterprise Server installed to access the services.
- **File sources path.** A path to a folder where files to be scanned will be loaded. Default: C : \ProgramData\Application Inspector\Sources.
- **Update service working folder.** A path to the folder for managing updates.
- **FUS server host.** The address of the JSC Positive Technologies update server.
- **License server host.** The address of the license server.
- **Gateway service https port.** A gateway port for HTTPS services.
- **Gateway service http port.** A gateway port for HTTP services.
- **Auth service port.** A port for connecting to the authentication and authorization service.
- **Project service port.** A port for connecting to the service with project data.
- **File service port.** A port for connecting to the file management service.
- **Settings service port.** A port for connecting to the settings service.
- **Agent auth gateway port.** A port for connecting to PT AI Enterprise Agent.
- **Descriptions service port.** A port for connecting to the description service.
- **Vault server port.** A port for connecting to the Vault data secure storage service.
- **Notify service port.** A port for connecting to the notification service.
- **Issue tracker service port.** A port for connecting to the issue tracker.
- **Consul service port.** A port for connecting to the Consul service.
- **Update service port.** A port for connecting to the update service.
- **UI service port.** A port for connecting to the PT AI Enterprise Edition web interface.
- **Change history service port.** A port for the event log service.
- **System management service port.** A port for connecting to the system management service.
- **Scan scheduler port.** A port for connecting to the scan queue management service.

10. Click **Next**.

The certificate configuration window will open.

11. In the box **Location of SSL certificate**, indicate the location of the server certificate.
12. In the box **SSL password**, enter the password for the server certificate.
13. If the company has a server for issuing and revoking certificates, select **Verify client certificate for revocation**.
14. Click **Next**.

The page for configuring the administrator with global permissions will open.

15. In the **Administrators** box, enter the name (sAMAccountName) of a user that will be assigned the administrator role.

Note. You can assign several administrators by indicating their names separated by a semicolon.

16. Click **Next**.
17. Verify that the installation options are correct, and click **Install**.

The module installation will start on the computer. Wait for the installation to finish.

Note. At PT AI Enterprise Server installation, [Microsoft .NET 4.7.2 Development Pack](#) will be installed as well (if not already). In this case, for correct operation, OS restart may be required after the installation.

18. Click **Finish**.

PT AI Enterprise Server is installed.

6.2. Installing the PT AI Enterprise Viewer module

- To install PT AI Enterprise Viewer on the computer:

1. Run the setup file.

The window for selecting the installation language will open.
2. Select a language and click **OK**.
3. In the window with the text of the End User License Agreement, read the terms and conditions of the End User License Agreement.
4. Select **I accept the agreement** if you accept all terms and conditions of the license agreement.
5. Click **Next**.

Preparation for installing the program will continue.
6. Select a destination folder for the program installation.
7. Click **Next**.
8. Select **Create a desktop icon** to create a shortcut on the desktop.

9. Verify that the installation options are correct, and click **Install**.

The module installation will start on the computer. Wait for the installation to finish.

Note. At PT AI Enterprise Viewer installation, [Microsoft .NET 4.7.2 Development Pack](#) will be installed as well (if not already). In this case, for correct operation, OS restart may be required after the installation.

10. Click **Finish**.

PT AI Enterprise Viewer is installed.

6.3. Deploying the PT AI Enterprise Agent module

This section provides instructions on how to prepare for deployment, installation, and setup of PT AI Enterprise Agent.

In this section

[Preparing for PT AI Enterprise Agent installation \(see Section 6.3.1\)](#)

[Installing the PT AI Enterprise Agent module \(see Section 6.3.2\)](#)

[Setting up the PT AI Enterprise Agent module \(see Section 6.3.3\)](#)

6.3.1. Preparing for PT AI Enterprise Agent installation

For correct PT AI Enterprise Agent operation, it is necessary that you install and set up the following programs:

- To compile C# projects: [Microsoft .NET 4.7.2 Development Pack](#) At the PT AI Enterprise Agent installation, Microsoft .NET 4.7.2 Development Pack will be loaded and installed as well (if not already). In this case, for correct operation, OS restart may be required after the installation.
- To load Java dependencies: [Apache Maven](#).
- To download PHP dependences: [Composer](#).

Note. To scan Xamarin projects, install Microsoft .NET Framework 3.5.1.

In addition, in the user account control (UAC) settings for the Windows OS, it is recommended that you disable notifications about software installation.

6.3.2. Installing the PT AI Enterprise Agent module

- To install PT AI Enterprise Agent on the computer:

1. Run the setup file.

The window for selecting the installation language will open.

2. Select a language and click **OK**.

The Setup Wizard window opens.

3. Click **Next**.
4. In the window with the text of the End User License Agreement, read the terms and conditions of the End User License Agreement.
5. Select **I accept the agreement** if you accept all terms and conditions of the license agreement.
6. Click **Next**.

Preparation for installing the program will continue.

7. Select a destination folder for the program installation.
8. Click **Next**.
9. In the open window, configure connection between the PT AI Enterprise Agent and PT AI Enterprise Server modules using the access token:
 - Specify the name of a computer with PT AI Enterprise Server installed.
 - Enter an access token.

Note. To create an access token, when the PT AI Enterprise Server module is installed, go to the PT AI Enterprise Edition web interface and follow the instructions from the "Creating an access token" section of the User Guide.

10. Click **Next**.
11. If you do not want to install the scan agent management service along with the PT AI Enterprise Agent module, clear the **Work with scan scheduler** check box in the open window.

Note. If the service is not installed, the scan agent can be started only in the console. Scanning in the PT AI Enterprise Viewer web interface will not be available.

12. Verify that the installation options are correct, and click **Install**.

The installation of the program on your computer will start. Wait for the installation to finish.

13. Click **Finish**.

PT AI Enterprise Agent is installed.

6.3.3. Setting up the PT AI Enterprise Agent module

At the PT AI Enterprise Agent installation, you set up a connection between it and PT AI Enterprise Server via an access token. If the PT AI Enterprise Server address or token changes after the installation, you can reflect these changes in the PT AI Enterprise Agent configuration in the `aic.user.config` configuration file.

Note. If you have the previous version of PT AI Enterprise Edition, you can continue using the client certificate and set up the PT AI Enterprise Agent connection with it.

► To set up PT AI Enterprise Agent:

1. Open the `aic.user.config` file in the folder that contains the installed PT AI Enterprise Agent module.
2. Add or change file lines as shown in the figure below.

```
<Agent.Properties.Settings>
  <setting name="AuthCertificatePassword" serializeAs="String">
    <value>Password</value> // Password to client certificate
  </setting>
  <setting name="AuthCertificatePath" serializeAs="String">
    <value>AI.Enterprise.Client.Development.pfx</value> // Path to the certificate file
  </setting>
  <setting name="SettingProviderUri" serializeAs="String">
    <value>https://ai-enterprise-server.domain.ru</value> // Server address
  </setting>
  <setting name="AccessToken" serializeAs="String">
    <value>Token</value> // Access token
  </setting>
</Agent.Properties.Settings>
```

Figure 2. PT AI Enterprise Agent setup

3. Save the file.

PT AI Enterprise Agent is now set up.

6.4. Installing the AI.Shell light agent

The PT AI Enterprise Agent module supports only Windows operating systems. If a company supports integration into the CI process on build agents under a Linux OS or you are working with Docker containers, you must install the AI.Shell component on the CI agents. AI.Shell is a light cross-platform agent that sends a scan task to the PT AI Enterprise Server module, and PT AI Enterprise Server queues the task to be performed by an available PT AI Enterprise Agent.

In this section

[Installing AI.Shell from the package \(see Section 6.4.1\)](#)

[Installing AI.Shell with Windows Installer \(see Section 6.4.2\)](#)

[Configuring AI.Shell and starting scanning \(see Section 6.4.3\)](#)

6.4.1. Installing AI.Shell from the package

Depending on an operating system, AI.Shell has several distribution options.

Table 2. AI.Shell installation packages

OS	Extension of the installation package
Microsoft Windows (x64)	A standard installer for Microsoft Windows (containing all files of a required application) or a ready-to-use package for Windows Docker that does not require installation
Latest Alpine (x64) or LTS	.tar.gz
Latest CentOS (x64) or LTS	.rpm
Latest Ubuntu (x64) or LTS	.deb
Latest Debian (x64) or LTS	.deb
Other Linux systems	.tar.gz

Installing AI.Shell from packages with .deb and .rpm extensions

Note. If the operating system does not contain the libssl library, which is required for secure data transfer via the Internet and is part of the implementation of SSL and TLS encryption protocols, it must be installed in advance.

- To install AI.Shell from packages with .deb and .rpm extensions,

run the installation command:

For .deb packages: `apt-get install <Package name>` or `dpkg -i <Package name>`

For .rpm packages: `rpm -i <Package name>`

Installing AI.Shell from a .tar.gz archive

- To install AI.Shell from a .tar.gz archive:

1. Download the provided archive and unpack its contents to the `/usr/share/aisa` directory.
2. Run the installation command:

`sh/<AI.Shell directory path>/scripts/install.sh`

Note. You can remove AI.Shell with `sh/<AI.Shell directory path>/scripts/remove.sh`.

6.4.2. Installing AI.Shell with Windows Installer

- To install AI.Shell on a computer:

1. Run the setup file.

The window for selecting the installation language will open.

2. Select a language and click **OK**.

The Setup Wizard window opens.

3. In the window with the text of the End User License Agreement, read the terms and conditions of the End User License Agreement.
4. Select **I accept the agreement** if you accept all terms and conditions of the license agreement.
5. Click **Next**.
6. Select a destination folder for the program installation.
7. Click **Next**.
8. In the open window, configure a connection between AI.Shell and the PT AI Enterprise Server module using the access token:
 - Specify the name of a computer with PT AI Enterprise Server installed.
 - Enter an access token.

Note. To create an access token, when the PT AI Enterprise Server module is installed, go to the PT AI Enterprise Edition web interface and follow the instructions from the "Creating an access token" section of the User Guide.

9. Click **Next**.
10. Verify that the installation options are correct, and click **Install**.

The installation of the program on your computer will start. Wait for the installation to finish.

11. Click **Finish**.

AI.Shell is now installed.

6.4.3. Configuring AI.Shell and starting scanning

You can start scanning using AI.Shell in the CLI or on the CI agent.

- To start a scan task using AI.Shell:

1. Set up the AI.Shell connection to the PT AI Enterprise Server module by running the following command:

```
aisa --set-settings -u <PT AI Enterprise Server address> -t <Access token>
```

Note. If a scan is started on the CI agent, instead of using a separate command to set up a connection to PT AI Enterprise Server, you can specify connection settings right before running a scan.

2. Run the scan command:

```
aisa --project-name "<Name of a scan project>" --scan-target "<Path of a scan target>" if you set up a connection between AI.Shell and PT AI Enterprise Server at the previous step
```

```
aisa --set-settings -u <PT AI Enterprise Server address> -t <Access token> --project-name "<Name of a scan project>" --scan-target "<Path of a scan target>" if you did not set up a connection at the previous step
```

Note. To start a scan using AI.Shell, use the same start settings as for PT AI Enterprise Agent, as well as some specific settings (see the table below).

Table 3. Scan start parameters for AI.Shell

Setting	Description	Note
-u	PT AI Enterprise Server server address	Only paired settings are taken into account. Specified settings are first validated and then scanned
-t	Access token	
--no-wait	Enabling subscription to scan task events	<p>If the setting is specified, after the scan is queued, AI.Shell finishes its work. If it is not specified, AI.Shell will receive from PT AI Enterprise Server all the scan task statuses and the scanning progress until the task is completed. Only then will AI.Shell finish its work.</p> <p>Note. If the --no-wait parameter is present, you cannot generate reports even if the --reports-folder and --reports report generation settings are specified</p>

After the scan is completed, in addition to standard return codes, AI.Shell-specific codes may appear:

- 27: the AI.Shell version is out of date and a new version is available.
- 28: no active scan agents.
- 29: no access token for AI.Shell.

7. Configuring scan settings

After all the PT AI Enterprise Edition modules are installed, you need to configure project scanning and a security policy. A security policy is a set of conditions to achieve a required level of software security. Breach of a security policy is set in a rule. During project building, the build agent validates scan results against the rule conditions and stops the build process if the policy is breached.

You can configure project scanning and the security policy in the following ways:

- During the build agent configuration. You indicate scan settings when you create a configuration file. You indicate a security policy rule when you create security policies.
- Before the build agent configuration. You create text files with scan settings and a security policy rule in advance. When you configure the build agent, you indicate a path to the files created. In this case, you do not need to configure scanning and the security policy on the build agent.
- In the PT AI Enterprise Edition interface. For details on how to configure a scan project and security policy in the PT AI Enterprise Edition interface, see User Guide.

This section contains instructions on how to create a configuration file with scan settings and a file with a security policy rule.

You may find details on scan configuration on the build agent in the section dedicated to PT AI Enterprise Edition integration in the CI process.

In this section

[Configuring scanning in the configuration file \(see Section 7.1\)](#)

[Configuring a security policy \(see Section 7.2\)](#)

7.1. Configuring scanning in the configuration file

► To configure scanning in the configuration file:

1. Create a JSON configuration file in a text editor.
2. Write scan settings in the configuration file.

You may find an example configuration file in the [appendix \(see Appendix A\)](#).

3. Save the configuration file.

7.2. Configuring a security policy

You can configure a security policy in the configuration file or in the PT AI Enterprise Edition interface. For the detailed information on how to configure a security policy, see User Guide.

► To create a security policy rule:

1. Create a JSON file in a text editor.
2. Write a rule in a file according to the format in the table below.
3. Save the file.

Table 4. Format of writing a security policy rule

Section name	Line name	Line description
Policy	ID	Arbitrary line notation (optional, unique)
	CountToActualize	Minimum number of policy triggerings by detected vulnerabilities (if 0 or not found, one triggering is enough)
	Scopes	Set of rules (the policy is triggered if at least one scope is relevant)
Scope	Rules	Set of rules within the scope (if all rules match, the scope is relevant)
Rule	Field	Name of the vulnerability field
	Value	Field value required for a rule to be triggered (case-insensitive if isRegex is not set)
	IsRegex	A flag for comparison by a regular expression

In the table below, there are descriptions of general "Field" line values. You may find a full list of values in a JSON report that is generated after scanning is finished. Report parameters are set with the arguments `--reports` and `--reports-folder` when configuring the build agent.

Table 5. General "Field" line values

Value	Data type	Description
ApprovalState	Number	Vulnerability confirmation status. Possible values: 0 for no status, 1 for confirmed, 2 for rejected, 3 for non-existing, 4 for automatically confirmed
EntryPointFile	String	Path of a file with an entry point
EntryPointValue	String	Function or entry point method in the vulnerability card
IsSecondOrder	Logical value	IsSecondOrder enabled: false or true
IsSuspected	Logical value	Vulnerability suspected: false or true

Value	Data type	Description
IsSuppressed	Logical value	Suppressed vulnerabilities: false or true. By default, suppressed vulnerabilities are taken into account in the security policy
VulnerabilityLevel	String	Vulnerability severity: high, medium, low, potential
SearchAlgorithm	Number	Vulnerability search algorithm. Possible values: 0—from entry points (FromEntryPoint), 1—from available public and protected methods (FromPublicProtected), 2—Taint, 999—unknown
VulnerabilityTitle	String	Vulnerability type
VulnerableFile	String	Path of a file with an entry point
VulnerableValue	String	Exit point function or method in the vulnerability card
Exploit	String	Vulnerability check request
VulnerableFunction	String	Vulnerable function
Payload	String	Attack vector parameters
IsNew	Logical value	Flag for a new detected vulnerability: false or true

Below is an example rule for searching high-severity vulnerabilities excluding the rejected and suspected vulnerabilities.

```
[
  {
    "CountToActualize": 1,
    "Scopes": [
      {
        "Rules": [
          {
            "Field": "VulnerabilityLevel", // field name
            "Value": "High", // field value, case insensitive
            "IsRegex": false // whether to use regular expressions
          },
          {
            "Field": "IsSuspected",
```



```
    "Value": "false",
    "IsRegex": false
  },
  {
    "Field": "ApprovalState",
    "Value": "[^2]",
    "IsRegex": true
  }
]
}
]
```

8. PT AI Enterprise Edition integration into the CI process

PT AI Enterprise Edition can be integrated into the CI process on various build agents. The integration allows scanning code for vulnerabilities during project build.

In this section

[Configuring the TeamCity build agent \(see Section 8.1\)](#)

[Configuring the Jenkins build agent \(see Section 8.2\)](#)

8.1. Configuring the TeamCity build agent

TeamCity configuration consists of five steps:

1. Configuring the version control system (VCS)
2. Creating a configuration file (optional)
3. Source code build
4. Creating security policies (optional)
5. Configuring scan start

► To configure TeamCity:

1. Configure the VCS to load source code to a folder:
 - Open the build project in TeamCity.
 - In the menu of the project configuration, select **Version Control settings**.
 - Click **Attach VCS Root**.
 - On the open page **Edit VCS Root**, indicate the type of a VCS where source code is stored, the repository address, and a full path to a local folder where the code is loaded to from the repository.

VCS Roots
In this section you can configure how project source code is retrieved from VCS. [?](#)

[+ Attach VCS root](#)

Name		Checkout Rules
(jetbrains.git) Frameworks belongs to WebEcoSystem :: PT AI Tests :: Tests Latest check for changes: 14:55 (periodical run by the schedule) Changes checking interval: 5m	Edit	Detach Edit checkout rules (0)

Checkout Options
Additional VCS checkout and display settings.

Checkout Settings

VCS checkout mode: [?](#) Prefer to checkout files on agent (recommended)

Checkout directory: [?](#) Custom path
C:\Frameworks\ [Reset](#)
A relative or an absolute path to the agent working directory. Restrictions apply. [?](#)
⚠ This directory might be cleaned by TeamCity before the build

Clean build: ☐ Clean all files in the checkout directory before the build

Display Settings

Display options: ☐ Show changes from snapshot dependencies [?](#)
This also affects treatment of pending changes in schedule trigger.

You can also configure VCS roots labeling with the help of [VCS labeling build feature](#).

[Save](#) [Cancel](#)

[🔧 Hide advanced options](#)

Figure 3. Configuring a VCS

- Click **Save**.
- If you want to write a configuration file on the build agent, add a step that will create a file:
In the project settings menu, select **Build Steps**.
 - Click **Add build step**.
 - In the drop-down list **Runner Type**, select **Create Text File**.
 - In the box **File Content**, enter the text of the configuration file.
 - In the box **Destination file**, indicate a full path to the folder on the computer where the configuration file will be stored.

Runner type: Create Text File
Creates text file with specified content

Step name: AI Settings File
Optional, specify to distinguish this build step from other steps.

Execute step: If all previous steps finished successfully
Specify the step execution policy.

File content:

```
{
// Main Settings
"ProjectName": "Test Proj",
"ProgrammingLanguage": "Java",
"ScanAppType": "Configuration, Fingerprint, Java, PM",
"ThreadCount": 1,
"RootFolder": "C:\\TestProj",
"ScanTarget": "C:\\TestProj",
"Site": "http://localhost",
"IsDownloadDependencies": true,
"
```

Destination file: C:\test\proj.aiproj

Hide advanced options

Save Cancel

Figure 4. Creating a configuration file

4. Click **Save**.
5. Set up source code build:
 - Click **Add build step**.
 - In the drop-down list **Runner Type**, select **Command Line**.
 - In the box **Step name**, set the name of the build step.
 - In the box **Execute step**, indicate the step execution scenario **If all previous steps finished successfully**.
 - In the box **Working directory**, indicate the full path to the root folder that contains the application source code.
 - In the drop-down list **Run**, select **Custom script**.
 - In the **Custom script** box, enter the `mvn clean package` command.

Build Step (1 of 3): Build App | ▾

Runner type: Command Line ▾
Simple command execution

Step name: Build App
Optional, specify to distinguish this build step from other steps.

Execute step: If all previous steps finished successfully ▾
Specify the step execution policy.

Working directory: C:\Frameworks\Java\struts 📁 🔗
Optional, set if differs from the checkout directory.

Run: Custom script ▾

Custom script: Enter build script content:

```
mvn clean package
```

📄
A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

Deploy Artifacts To Artifactory

Artifactory server URL: <Do not activate> ▾
Select an Artifactory server.

[🔧 Hide advanced options](#)

Save Cancel

Figure 5. Setting up source code build

6. Click **Save**.
7. Set up policy creation:
 - Click **Add build step**.
 - In the drop-down list **Runner Type**, select **Create Text File**.
 - In the box **Step name**, set the name of the build step.
 - In the box **Execute step**, indicate the step execution scenario **If all previous steps finished successfully**.
 - In the box **File Content**, enter the text of the policy file.
 - In the box **Destination file**, indicate a full path to the folder on the computer where the policy file will be stored.

Build Step (2 of 4): Create Policies |

Runner type: Create Text File
Creates text file with specified content

Step name: Create Policies
Optional, specify to distinguish this build step from other steps.

Execute step: If all previous steps finished successfully
Specify the step execution policy.

File content:

```

{
  "CountToActualize": 1,
  "Scopes": [
    {
      "Rules": [
        {
          "Field": "Level",
          "Value": "High",
          "IsRegex": false
        },
        {
          "Field": "Exploit",
          "Value": ".",
          "IsRegex": true
        },
        {
          "Field": "ApprovalState",
          "Value": "({?/2}) *$",
          "IsRegex": true
        }
      ]
    }
  ]
}

```

Destination file: C:\test_ci\policies.json

Figure 6. Setting up policy file creation

8. Click **Save**.
9. Set up scan start from a console:
 - Click **Add build step**.
 - In the drop-down list **Runner Type**, select **Command Line**.
 - In the box **Step name**, set the name of the build step.
 - In the box **Execute step**, indicate the step execution scenario **If all previous steps finished successfully**.
 - In the drop-down list **Run**, select **Custom script**.
 - If scanning configuration is specified in the PT AI Enterprise Edition interface, in the **Custom script** box, enter the command: `aic.exe --project-name "<Name of the scan project>" --scan-target "<Path of the scan target>" --policies-path "<Path of the file with policy description>" --reports "<Report format: HTML, JSON, PDF, WAF>" --reports-folder "<Path of a folder where reports are saved>"`
 - If scanning configuration is specified in the configuration file, in the **Custom script** box, enter the command: `aic.exe --project-settings-file "<Path of the configuration file>" --scan-target "<Path of the scan target>" --policies-path "<Path of the file with policy description>" --reports "<Report format: HTML, JSON, PDF, WAF>" --reports-folder "<Path of a folder where reports are saved>"`

Build Step (3 of 4): Run ai |

Runner type: Command Line
Simple command execution

Step name: Run ai
Optional, specify to distinguish this build step from other steps.

Execute step: If all previous steps finished successfully
Specify the step execution policy.

Working directory: C:\AI\ApplicationInspector
Optional, set if differs from the checkout directory.

Run: Custom script

Custom script: Enter build script content:

```
exe -console --project-name "test_ci" --project-folder "C:\Framework\Java\struts\tax
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

Deploy Artifacts To Artifactory

Artifactory server URL: <Do not activate>
Select an Artifactory server.

[Hide advanced options](#)

Save Cancel

Figure 7. Setting up scan start from a console

10. Click **Save**.

11. Run project build.

After the scan is completed, the page of the project build will display a return code and build result. You may find the list of possible return codes in the [appendix \(see Appendix C\)](#).

Overview Changes Build Log Parameters Artifacts

Result: Exit code 10 (new)

Time: 23 Aug 17 16:10 - 16:16 (5m:27s)

Investigation: Start investigation... of current problems in this build configuration (Test_ci)

Build problems (1 new)

Build failure condition (1)

Process exited with code 10

[16:16:04] Process exited with code 10

[Hide details](#)

Figure 8. Scan results

8.2. Configuring the Jenkins build agent

Depending on the number of scanned projects, there are two ways to configure Jenkins.

If you need to check a separate project for vulnerabilities, you can use the Jenkins basic configuration. It implies that you will have to configure the build agent separately for each project.

If you need to check several projects at once, we recommend that you configure Jenkins using a plugin. The plugin allows you to avoid configuring the build agent separately for each project.

In this section

Jenkins basic configuration (see Section 8.2.1)

Setting up the Jenkins build agent using the plugin (see Section 8.2.2)

8.2.1. Jenkins basic configuration

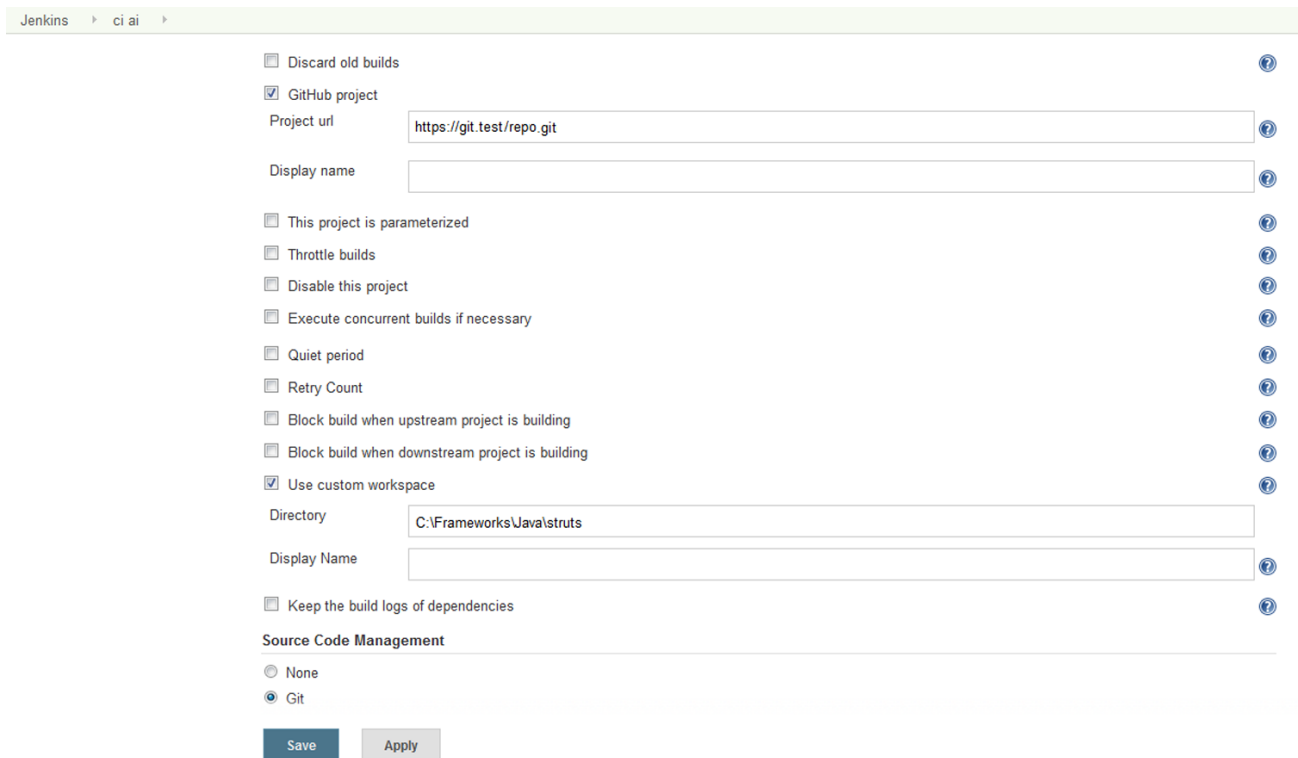
Note. The basic configuration is used if the scan agent management service is not installed.

The basic Jenkins configuration consists of five steps:

1. Configuring the version control system (VCS)
2. Creating a configuration file (optional)
3. Source code build
4. Creating security policies (optional)
5. Configuring scan start

► To configure Jenkins:

1. Configure the VCS to load source code to a folder:
 - Open the page for project configuration in Jenkins.
 - Under **General**, select **GitHub project**.
 - In the box **Project URL**, enter the address of the repository that stores the project source code.
 - Click **Advanced** under **General**.
 - Select **Use custom workspace**.
 - In the box **Directory**, indicate a full path to a local folder where the code is loaded to from the repository.



The screenshot shows the Jenkins configuration page for a project named 'ci ai'. The page is divided into several sections with various options and input fields. The 'Discard old builds' section has a checkbox that is unchecked. The 'GitHub project' section has a checkbox that is checked, and the 'Project url' field contains the text 'https://git.test/repo.git'. The 'Display name' field is empty. The 'This project is parameterized' section has a checkbox that is unchecked. The 'Throttle builds' section has a checkbox that is unchecked. The 'Disable this project' section has a checkbox that is unchecked. The 'Execute concurrent builds if necessary' section has a checkbox that is unchecked. The 'Quiet period' section has a checkbox that is unchecked. The 'Retry Count' section has a checkbox that is unchecked. The 'Block build when upstream project is building' section has a checkbox that is unchecked. The 'Block build when downstream project is building' section has a checkbox that is unchecked. The 'Use custom workspace' section has a checkbox that is checked, and the 'Directory' field contains the text 'C:\Frameworks\Java\struts'. The 'Display Name' field is empty. The 'Keep the build logs of dependencies' section has a checkbox that is unchecked. The 'Source Code Management' section has two radio buttons: 'None' and 'Git', with 'Git' selected. At the bottom of the page, there are two buttons: 'Save' and 'Apply'.

Jenkins > ci ai >

☐ Discard old builds

☒ GitHub project

Project url:

Display name:

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

☐ Quiet period

☐ Retry Count

☐ Block build when upstream project is building

☐ Block build when downstream project is building

☒ Use custom workspace

Directory:

Display Name:

☐ Keep the build logs of dependencies

Source Code Management

☐ None

☒ Git

Figure 9. Configuring a VCS

2. If you want to write a configuration file on the build agent, add a step that will create a file:
 - In the box **Build**, click **Add build step** and, in the drop-down menu, select **Execute Windows batch command**.
 - In the box **Command**, enter the text of the configuration file according to the syntax for BAT files.

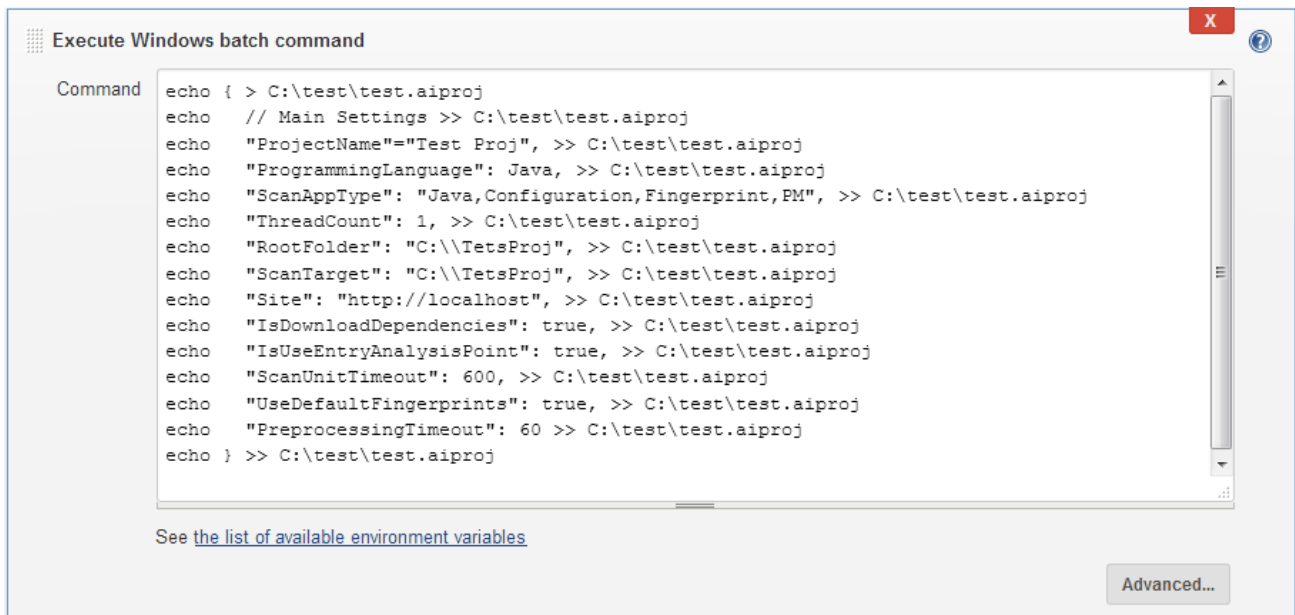


Figure 10. Creating a configuration file

3. Set up source code build:

- In the **Build** box, click **Add build step** and, in the drop-down menu, select **Invoke top-level Maven targets**.
- In the drop-down list **Maven Version**, select **maven**.
- In the box **Goals**, enter `clean package` as shown in the figure below.

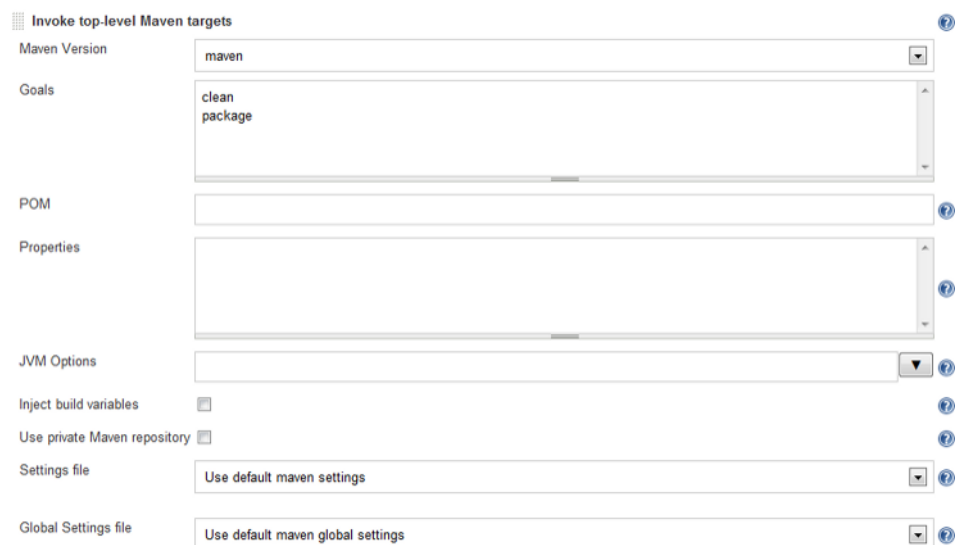


Figure 11. Setting up source code build

4. Set up policy creation:

- In the **Build** box, click **Add build step** and, in the drop-down menu, select **Execute Windows batch command**.
- In the box **Command**, enter the text of the policy file according to the syntax for BAT files.



Figure 12. Setting up creation of a policy file

5. Set up scan start from a console:

- In the **Build** box, click **Add build step** and, in the drop-down menu, select **Execute Windows batch command**.
- If scanning configuration is specified in the PT AI Enterprise Edition interface, in the box **Command**, enter the command: `aic.exe --project-name "<name of the scan project>" --scan-target "<path to the scan target>" --policies-path "<path to the file with policy description>" --reports "<report format: HTML, JSON, PDF, WAF>" --reports-folder "<path to a folder where reports are saved>"`
- If scanning configuration is specified in the configuration file, in the **Command** box, enter the command: `aic.exe --project-settings-file "<path to the configuration file>" --scan-target "<path to the scan target>" --policies-path "<path to the file with policy description>" --reports "<report format: HTML, JSON, PDF, WAF>" --reports-folder "<path to a folder where reports are saved>"`

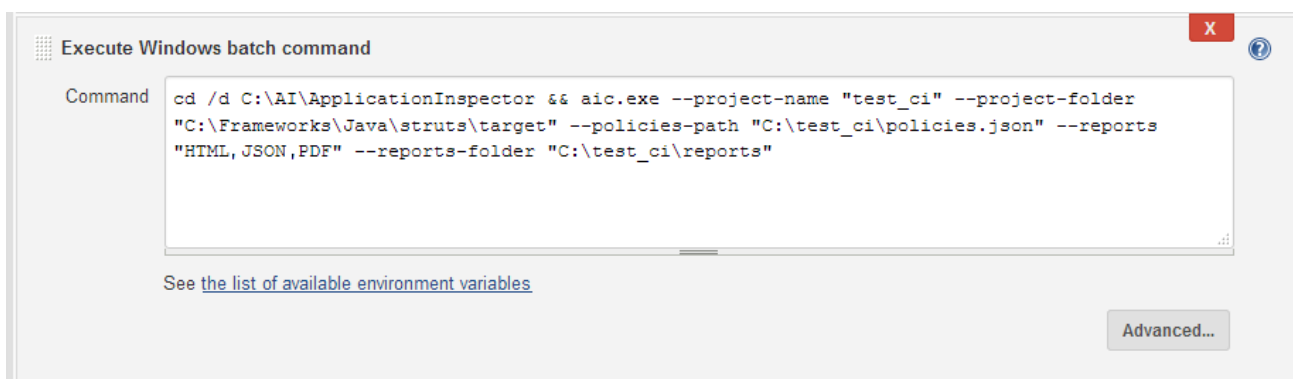


Figure 13. Setting up scan start from a console

6. Click **Save**.
7. Run project build.

After the scan is completed, the **Console Output** page will display a return code and build result. You may find the list of possible return codes in the [appendix \(see Appendix C\)](#).

```
Checking out Revision a08e800b023b8c5d28577f7be7f0275f106ac7e2b (refs/remotes/origin/master)
Commit message: "Master fix"
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f a08e800b023b8c5d28577f7be7f0275f106ac7e2b
> git.exe rev-list a08e800b023b8c5d28577f7be7f0275f106ac7e2b # timeout=10
[struts2] $ cmd.exe /C "C:\soft\mvn\bin\mvn.cmd clean package && exit %%ERRORLEVEL%%"
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building StrutsExample Maven Webapp 1.0-SNAPSHOT
[INFO] -----

C:\Frameworks\Java\struts>echo          } 1>>c:\test_ci\policies.json
C:\Frameworks\Java\struts>echo          ] 1>>c:\test_ci\policies.json
C:\Frameworks\Java\struts>echo          } 1>>c:\test_ci\policies.json
C:\Frameworks\Java\struts>echo          ] 1>>c:\test_ci\policies.json
C:\Frameworks\Java\struts>echo    } 1>>c:\test_ci\policies.json
C:\Frameworks\Java\struts>echo ] 1>>c:\test_ci\policies.json

C:\Frameworks\Java\struts>exit 0
[struts] $ cmd /c call C:\Users\user\AppData\Local\Temp\jenkins3895922381476295227.bat

C:\Frameworks\Java\struts>cd /d C:\AI\ApplicationInspector && aic.exe -console --project-name "test_ci"
--project-folder "C:\Frameworks\Java\struts\target" --policies-path "C:\test_ci\policies.json" --reports "HTML,JSON,PDF"
--reports-folder "C:\test_ci\reports"
Work started
Project test_ci running
Invalid license token. Try reconnect.
Attempt 1 of 5.
Loading project test_ci
Project test_ci loaded
Run scan service
Inspection result d8307b88-4d9d-454d-ba23-898991a86331 from 09.02.2018 15:53:29 loaded
Scan completed
Report generation started at C:\test_ci\report.json
Report generation finished
Load politic from C:\test_ci\policies.json
1 loaded
Items to check 50

C:\AI\ApplicationInspector>exit 10
Build step 'Execute Windows batch command' marked build as failure
Finished: FAILURE
```

Figure 14. Scan results

8.2.2. Setting up the Jenkins build agent using the plugin

Set up the Jenkins build agent using the plugin following these steps:

1. Upload the plugin to Jenkins.
2. Set up plugin connection to PT AI Enterprise Server.
3. Set up a project task. A project task is created separately for each project.

Upload the plugin to Jenkins

► To upload the plugin to Jenkins:

1. Download the plugin from the repository at github.com/PositiveTechnologies/ptaiPlugins.
2. Select Maven to build the plugin: `mvn clean install -DskipTests`.
Note. To build a plugin, first install and configure JDK version 8.
3. Go to the Jenkins web interface.
4. To the left, select the **Manage Jenkins** section.
5. On the open page, select **Manage plugins**.
6. Select the **Advanced** tab.
7. Under **Upload plugin**, click **Choose file**.
8. In the open window, select the `ptai-jenkins-plugin.hpi` file in the `ptai-ee-tools-java\ptai-jenkins-plugin\target` folder created after the build was completed, and click **Open**.
9. Click **Upload**.

The PT AI EE plugin is uploaded and is displayed on the **Plugin Manager** page on the **Installed** tab. You can proceed to setting up the plugin connection to PT AI Enterprise Server.

Setup of the plugin connection to PT AI Enterprise Server

► To set up the plugin connection to PT AI Enterprise Server:

1. To the left, select the **Manage Jenkins** section.
2. On the open page, select **Configure System**.
3. In the list of settings under **PT AI vulnerability analysis**, click **Add PT AI global configuration** to add a new connection to PT AI Enterprise Server.

PT AI vulnerability analysis

PT AI global configurations

PT AI configuration

Configuration name

PT AI server URL

Credentials

Figure 15. Setting up the plugin connection to PT AI Enterprise Server

4. In the **Configuration name** box, enter a connection display name in the Jenkins web interface.

5. In the **PT AI server URL** box, enter the PT AI Enterprise Server address.
6. Click **Add** next to the **Credentials** box and, in the open menu, select **Jenkins**.

The **Jenkins Credentials Provider** window will open: **Jenkins**.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain: Global credentials (unrestricted)

Kind: PT AI server authentication

PT AI client API token:

PT AI server CA certificates: pDAwpBHOzZhlfdVt9SenKgETsApHFTc8wJAYnZYhEd0DwWLPypkjBrYJW2PD3op
HLivJoOwjdIk5HQ8ythGzKQBY39Li6faIArxnnTWA42LmXi+8kmileWhQJAWbJa
9U1UczkMEBoprGKYGZ00jtrsUQ5Gz7Fm08nTtNn1noOJtzL1wxCW6WLOQrl0KkZf
WJX9br8fK3AUqEdRqQJBAIQ/cwsfro2YQS1ncv3IYPYJKYaZkeC/MQaTBdFp4qh1
DGB5HPi1hAG9s8OK7f+qKmHzioreNhK0PPApmh5JHLQ=

Test CA certificates

ID:

Description:

Add Cancel

Figure 16. Adding certificates

7. In the **Kind** drop-down list, select **PT AI server authentication**.
8. In the **PT AI client API token** box, enter an access token for the CI/CD plugins.
Note. To create an access token, go to the PT AI Enterprise Edition web interface and follow the instructions from the "Creating an access token" section of the User Guide.
9. In the **PT AI server CA certificates** box, specify the server certificate.
10. Click **Test CA certificates** to test the server certificate.
11. Click **Save**.

The plugin connection to PT AI Enterprise Server is now set up. You can proceed to setting up a project task.

Setting up a project task

- To set up a project task:

1. To the left, select the **New item** section.
2. On the open page, select **Freestyle project** and specify the name of the project.
3. Click **OK**.

4. On the open page under **Source code management** in the **Repository URL** box, enter the address of the repository where the source code of the scanned application is stored.
5. Under **Build** in the **Add build step** button menu, select the **PT AI vulnerability analysis** plugin.
6. In the **Scan settings type** drop-down list, select:
 - **PT AI Viewer UI-defined settings** if the scan is set up in PT AI Enterprise Viewer.
 - **JSON-defined settings** if you want to write to the configuration file and the security policy file on the build agent.
7. If you selected **PT AI Viewer UI-defined settings**, in the **Project name** box, enter the name of the project that was created and set up in PT AI Enterprise Viewer.

PT AI vulnerability analysis [X]

Scan settings type: PT AI Viewer UI-defined settings

Project name: Test project

Server config: Global scope defined PT AI server config

Configuration name: test

Test PT AI project presence

Work mode: Synchronous mode

Fail step if SAST failed: ☒

Fail step if SAST unstable: ☐

Reports to generate: Add report

Files to analyse

Files

Files to analyse: **/*

Remove prefix:

Advanced...

Add transfer set

Figure 17. Setting up a project task if the scan is set up in PT AI Enterprise Viewer

8. If you selected **JSON-defined settings**, in the **Scan settings** and **Policy** boxes, enter the text of the configuration file and the policy file (optional) in JSON.

Build

PT AI vulnerability analysis

Scan settings type: **JSON-defined settings**

Scan settings:

```
{
  "ProjectName": "test",
  "ProgrammingLanguage": "CSharp",
  "ScanAppType": "CSharp, Configuration, PmTaint, Fingerprint",
  "ThreadCount": 1,
}
```

Policy:

```
[
  {
    "CountToActualize": 1,
    "Scopes": [
      {

```

Server config: **Global scope defined PT AI server config**

Configuration name: **test**

Test PT AI project presence

Figure 18. Setting up a project task if the scan is set up on the build agent

9. In the **Server config** drop-down list, select:
 - **Global scope defined PT AI server config** to use the plugin global variables set during the setup of the plugin connection to PT AI Enterprise Server.
 - **Task scope defined PT AI server config** to use plugin settings configured for a specific project. For example, if you want to set up project connection to another PT AI Enterprise Server.
10. If you selected **Global scope defined PT AI server config**, in the **Configuration name** box, select the PT AI Enterprise Server name specified during the setup of the plugin connection to PT AI Enterprise Server.
11. If you selected **Task scope defined PT AI server config**, configure the plugin connection settings for a specific project similar to the plugin connection settings for PT AI Enterprise Server.
12. In the **Work mode** drop-down list, select:
 - **Asynchronous mode** for scanning in asynchronous mode. Jenkins will start a code check for vulnerabilities and continue building without waiting for the check to be completed.
 - **Synchronous mode** for scanning in synchronous mode. Jenkins will start a code check for vulnerabilities and wait for the check to finish.
13. If you selected **Synchronous mode**:

Select **Fail step if SAST failed** for Jenkins to stop the build if the scanned application does not comply with the set security policy.

Select **Fail step if SAST unstable** for Jenkins to stop the build if errors that are not related to vulnerability check are raised (for example, configuration errors, missing or invalid license).

14. In the **Files to analyze** box, indicate what files to exclude from and/or include to the scan by mask.
15. Click **Save**.
16. Run the project build.

After the build is completed, code check for vulnerabilities via plugin will start. The results of the check will be saved to the project report in the **Workspace** section.

9. Contacting Technical Support

Technical Support includes the following services:

- Help with queries about product usage and features
- Diagnostics, including pinpointing the causes of failures and informing the client of identified issues
- Resolution of product-related problems, and providing solutions or workarounds that maintain necessary performance
- Correcting product-related bugs (as part of product update releases)

You can obtain Technical Support at the online portal support.ptsecurity.com or by phone. When possible, we recommend using the online portal.

This section describes how to get Technical Support and the terms and conditions for using the service.

In this section

[Technical Support online \(see Section 9.1\)](#)

[Technical Support by phone \(see Section 9.2\)](#)

[Technical Support working hours \(see Section 9.3\)](#)

[How Technical Support processes requests \(see Section 9.4\)](#)

9.1. Technical Support online

You can request help on the Technical Support online portal at support.ptsecurity.com.

You can create a portal account using email addresses on your organization's official domain. You can specify other email addresses as secondary addresses for the account. For quicker response, specify the name of your organization and contact phone number in the account profile.

The support.ptsecurity.com portal contains knowledge base articles, news about Positive Technologies product updates, and answers to FAQs. Create a user account on the portal to have access to the knowledge base and all news.

Technical Support online is available in English and Russian.

9.2. Technical Support by phone

You can contact Technical Support at the following phone number: +7 495 744 01 44.

Technical Support by phone is available in English and Russian.

By phone you can receive a quick diagnosis, answers to simple questions, or a status update for a previous support request.

If the issue cannot be resolved within a reasonable amount of time (15–20 minutes), please create a request online at support.ptsecurity.com. To be guaranteed of receiving a response, be sure to follow the Technical Support recommendations when filling out your request online and provide any information requested.

9.3. Technical Support working hours

You can create and update support requests, read news, and access the knowledge base online 24/7. Technical Support processes requests and receives phone calls Monday–Friday, 09:00–19:00 UTC+3.

9.4. How Technical Support processes requests

When your request is received, Technical Support classifies it by type and severity in order to take further steps.

In this section

[Providing information for Technical Support \(see Section 9.4.1\)](#)

[Request types \(see Section 9.4.2\)](#)

[Response time and request prioritization \(see Section 9.4.3\)](#)

[Request processing \(see Section 9.4.4\)](#)

9.4.1. Providing information for Technical Support

When requested by a Positive Technologies support specialist, please provide:

- Product license number
- Log files and other diagnostic data stored in the product
- Screenshots
- Results of implementing Technical Support recommendations
- Remote access to the product (the particular access method best for diagnostics is decided by mutual agreement)

Positive Technologies has no obligation to provide Technical Support services if the above information is not provided.

If information needed for the request is not provided within a reasonable period of time (two weeks from the date of the most recent activity), Technical Support may close the request and notify you accordingly.

9.4.2. Request types

Technical Support assigns one of the following types to each request.

Queries regarding installation, reinstallation, and pre-start configuration of the product

Covers product setup and initial use. Technical Support of this type is available for 30 days following activation of the product.

Queries regarding product administration and configuration

Covers questions related to product use and recommendations for product optimization and configuration.

Restoring the product

In the event of a critical failure and/or unavailability of core functionality, a Positive Technologies specialist will assist with restoring the product. Restoration involves either reinstallation of the product (potentially causing loss of data) or rollback to a backup (if backups have been created prior to when the problem occurred). Positive Technologies is not responsible for data loss in case of faulty backups.

Updating the product

Positive Technologies provides product updates for the duration of the license period.

Positive Technologies is not responsible for problems caused by failure to follow proper update practices.

If a bug is found

If diagnostic analysis identifies a defect in the product, Positive Technologies shall make reasonable efforts to provide a workaround (if possible) and fix the defect in the earliest possible update.

9.4.3. Response time and request prioritization

Response time is defined as the time from receipt of a support request until Technical Support responds with a notification that work has been started on your request.

Processing time is defined as the time from when work is started on your request until Technical Support describes steps for resolving the problem, or until Technical Support classifies the issue as a software defect and refers it to the relevant development team.

Response time and processing time depend on the [severity level \(see Table 6\)](#) that you indicate in your request.

Technical Support may adjust the severity level of a request based on the criteria listed below. Every reasonable effort will be made to comply with the target deadline, but an extension may be required in exceptional circumstances.

Table 6. Response and processing time

Severity level	Severity criteria	Response time	Processing time
Critical	Emergencies that fully prevent the product from operating normally (not including initial installation) or have a critical impact on business activity	Within 4 hours	No limit
High	Failures partially affecting product functionality and arising in all operating conditions, or having a significant impact on business activity	Within 24 hours	No limit
Normal	Failures arising in specific operating conditions or not having a significant impact on business activity	Within 24 hours	No limit
Low	Questions of an informational nature or failures that do not impact product use	Within 24 hours	No limit

Response time and processing time are defined in terms of Technical Support working hours.

9.4.4. Request processing

As your request is processed, Technical Support will inform you of:

- Diagnostic analysis and results
- Solutions and ways to work around the causes of the problem
- Planning and release of product updates (if required to resolve the problem)

If changes to the product are required to resolve the problem, Positive Technologies shall include a patch in the earliest possible product update (depending on the complexity of changes required).

The request shall be considered closed if:

- A solution or workaround is delivered that does not impact the performance or a critically important function of the product.
- A bug in the product is diagnosed, technical information is collected about the bug and the conditions for reproducing it, and the bug is due to be fixed as part of a subsequent product update.
- The problem is identified as having been caused by third-party software or hardware not covered under the warranty.
- The problem is classified as an unsupported type.

Appendix A. Example configuration file

Below is an example configuration file where scan settings are indicated. You can write a configuration file in the build agent or in a text editor.

```
{
  // Configuring general settings
  "ProjectName": "Test_Proj", // Project name
  "ProgrammingLanguage": "Csharp", // Programming language: Java, Php, Csharp, VB,
  ObjectiveC, Cplusplus, Sql, Swift, Python, JavaScript, Kotlin, Go
  "ScanAppType": "CSharp , Configuration, Fingerprint, PmTaint ", // Vulnerability
  search modules: Php, Java, CSharp, Configuration, Fingerprint, PmTaint , Blackbox,
  JavaScript

  "ThreadCount": 1, // Number of threads
  "Site": "http://localhost", // Local host address
  "IsDownloadDependencies": true, // Download dependencies

  "IsUsePublicAnalysisMethod": false, // Use available public and protected methods
  for search
  "IsUseEntryAnalysisPoint": true, // Start search from entry points

  "ScanUnitTimeout": 600, // Maximum duration of a file scan in seconds
  "PreprocessingTimeout": 60, // Preprocessing timeout in minutes
  "CustomParameters": null, // Additional start settings

  "SkipFileFormats": ["*.gif"], // File formats excluded from scan
  "SkipFilesFolders": ["\\.git\\", "\\.gitignore", "\\.gitmodules", "\
\\.gitattributes", "\\$tf\\", "\\$BuildProcessTemplate\\", "\\.tfignore"], // Tree
  filter

  // Search for vulnerabilities
  "DisabledPatterns": ["145", "146", "148", "149"], // Search by templates, disabled
  templates
  "DisabledTypes": [], // Source-code checks, disabled source-code checks
  "ConsiderPreviousScan": true, // Take into account the previous scan
  "UseIssueTrackerIntegration": true, // Use integration with Jira

  // Java settings
  "IsUnpackUserPackages": false, // Unpack user packages
```

```
"JavaParameters": null, // JDK start settings
"JavaVersion": 0, // JDK version, 0 stands for version 1.8, 1 stands for version
1.11

// C# settings
"ProjectType": "Solution", // Project type: Solution, WebSite
"SolutionFile": "path_to_solution.sln", // Path of a solution or project file
"WebSiteFolder": "path_to_website", // Website folder

// JavaScript settings
"JavaScriptProjectFile": "path_to_file", // Path of the script file
"JavaScriptProjectFolder": "path_to_dir", // Path of the "javascript" project root
file

// PMTaint settings
"UseTaintAnalysis": false, // Use taint analysis
"UsePmAnalysis": true, // Use PM analysis only
"DisableInterpretCores": false, // Ignore interpretation cores (C#, Java, PHP)
during analysis

// YARA knowledge base settings
"UseDefaultFingerprints": true, // Use PT AI database of vulnerable components
"UseCustomYaraRules": false, // Use YARA user rules

// Black-box settings
"BlackBoxScanLevel": "None", // Search mode: Fast, Normal, Full
"CustomHeaders": [["", ""]], // Additional headers
"Authentication": {
  "auth_item": {
    "domain": null, // Address for authentication validation for the black-box
scanner
  "credentials": {
    "cookie": null, // Cookie value
    "type": 2, // Authentication type: 0 = Form, 1 = HTTP, 2 = None, 3 = Cookie
    "login": {
      "name": null, // Login name key
      "value": null, // Login name value
```



```
    "regex": null,
    "is_regex": false
  },
  "password": {
    "name": null, // Password key
    "value": null, // Password value
    "regex": null, // Example: "p[aA]ss(word)?"
    "is_regex": false
  }
},
"test_url": null, // Address for verifying successful authentication
"form_url": null, // Form address
"form_xpath": ".//form", // Form XPath path
"regex_of_success": null // Regex for verifying successful authentication
}
},
"ProxySettings": {
  "IsEnabled": false, // Enable proxy server settings
  "Host": null, // IP address
  "Port": null, // Port
  "Type": 0, // Proxy server type: 0 = HTTP, 1 = HTTPNOCONNECT, 2 = SOCKS4, 3 =
SOCKS5
  "Username": null, // Username
  "Password": null // Password
},

// Configuring automated vulnerability check
"RunAutocheckAfterScan": false, // Run autocheck after scanning
"AutocheckSite": "http://localhost", // Local host address for automated check. If
there is none, the value from the "Site" setting is used
"AutocheckCustomHeaders": [""], // Additional headers
"AutocheckAuthentication": {
  "auth_item": {
    "domain": null, // Check address
    "credentials": {
      "cookie": null, // Cookie value
```

```
"cookies": null,
"type": 2, // Authentication type: 0 = Form, 1 = HTTP, 2 = None, 3 = Cookie
"login": {
  "name": null, // Login name key
  "value": null, // Login name value
  "regexp": null,
  "is_regexp": false
},
"password": {
  "name": null, // Password key
  "value": null, // Password value
  "regexp": null, // Example: "p[aA]ss(word)?"
  "is_regexp": false
}
},
"test_url": null, // "Verification address" box
"form_url": null, // "Sign-in form URL" box
"form_xpath": ".//form", // Form XPath path
"regexp_of_success": null // Regex for verifying successful authentication
}
},
"AutocheckProxySettings": {
  "IsEnabled": false, // Enable proxy server settings
  "Host": null, // IP address
  "Port": null, // Port
  "Type": 0, // Proxy server type: 0 = HTTP, 1 = HTTPNOCONNECT, 2 = SOCKS4, 3 =
SOCKS5
  "Username": null, // Username
  "Password": null // Password
},

"SendEmailWithReportsAfterScan": true, // Email the report after the scan is
finished
"CompressReport": false, // Compress the report before emailing it

// Report emailing settings
```

```
"EmailSettings": {
  "SmtpServerAddress": "mail.ptsecurity.ru", // SMTP server address
  "UserName": "testagent_wes@ptsecurity.com", // Username
  "Password": "P@ssw0rd", // Password
  "EmailRecipients": "User@ptsecurity.ru", // Recipient's address, you can specify
several email addresses separated with a comma
  "EnableSsl": true, // Enable SSL
  "Subject": "Email Title", // Email subject
  "ConsiderCertificateError": true, // Take into account certificate errors
  "SenderEmail": "testagent_wes@ptsecurity.com" // Sender
},

// Configuring a report
"ReportParameters": {
  "SaveAsPath": null, // Folder for saving reports
  "UseFilters": false, // Use filters
  "CreatePdfPrintVersion": false, // Create a print version
  "IncludeDiscardedVulnerabilities": false, // Add discarded vulnerabilities to the
report
  "IncludeSuppressedVulnerabilities": false, // Add suppressed vulnerabilities to the
report
  "IncludeSuspectedVulnerabilities": true, // Add suspected vulnerabilities to the
report
  "IncludeGlossary": false, // Add reference information on vulnerabilities to the
report
  "IncludeDFD": false // Add a data flow diagram (DFD)
}
}
```

Appendix B. Parameters for starting a scan from CLI

You can start a scan in PT AI Enterprise Edition from the CLI. Below is the list of start parameters.

Table 7. The list of start parameters

Setting	Used in the AI.Shell light agent	Description
--project-name	Yes	Case-insensitive name of a scan project
--full-scan	Yes	Forced start of a full scan
--policies-path	Yes	A path to the file with policy description
--project-settings-file	Yes	Path to the configuration file
--reports-folder	Yes	A path to the folder for saving reports
--reports	Yes	Format for saving a report: HTML, PDF, JSON, WAF. You can indicate several formats separated by commas
--report-type	No	Type of a report created after scanning is completed. Possible values: PlainReport, AutoCheck, Nist, Oud4, Owaspm, Pcidss, Sans. If the type is not specified, the PlainReport report is generated
--sync	No	Synchronized start of several instances of the executable file aic.exe. At the synchronized start of aic.exe, PT AI Enterprise Agent performs scans one by one
--scan-target	Yes	A path to a folder or file of the target application. If a path is not indicated, PT AI Enterprise Agent scans the current folder
--restore-sources	No	Setting for downloading an application for scanning from PT AI Enterprise Server to the target folder --scan-target. Note that the content of the folder will be replaced with the data from PT AI Enterprise Server
--settings-export-file	No	Exporting project settings to a file from the specified path in JSON
--scan-off	Yes	Creating a project based on the configuration file without starting a scan task. Used together with the --project-settings-file parameter

Appendix C. Return codes

When scanning is completed on the build agent, a return code is displayed. Below is a list of possible return codes.

Table 8. Return codes

Code	Description
0	Successful scan
-1	Program started
-2	An error occurred when starting a child process
2	Target folder not found
3	License issue
4	Scan project not found
5	Project configuration error
6	Errors occurred during scanning
7	Report emailing error
8	Incorrect path to the configuration file
9	Incorrect path to the report folder
10	Security policy triggered (the build does not comply with the set security policy)
11	Incorrect report configuration
12	Client certificate not found
13	Scan deleted
14	Authentication error occurred during autocheck configuration
15	Incorrect proxy-server configuration (if an automated vulnerability check was used)
16	Incorrect address of the website for scanning (if an automated vulnerability check was used)
17	Security policy error
18	Critical error in the scan module
19	Scan module not found
20	Error occurred when loading source files from PT AI Enterprise Server
21	Timeout for PT AI Enterprise Server notifications on PT AI Enterprise Agent regular operation
22	Update error
23	Incorrect certificate password

Code	Description
24	Server certificate not found
26	The scheduler service is running. You cannot start a scan agent via the console
27	The AI.Shell version is out of date and a new version is available
28	No active scan agents available
29	No access tokens for AI.Shell
60	Joined codes 6 and 10 The security policy was triggered and non-critical errors were detected during scanning
100	Scan canceled (CTRL+C)
1000	Unknown error

About JSC Positive Technologies

For 18 years, Positive Technologies has been creating innovative solutions for information security. We develop products and services to detect, verify, and neutralize the real-world business risks associated with corporate IT infrastructure. Our technologies are backed by years of research experience and the expertise of world-class cybersecurity experts.

Over 2,000 companies in 30 countries trust us to keep them safe.

POSITIVE TECHNOLOGIES

info@ptsecurity.com

ptsecurity.com
