Micro Focus
Fortify CloudScan

Software Version: 19.2.0

Installation, Configuration, and Usage Guide

Document Release Date: November 2019
Software Release Date: November 2019

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated otherwise, a valid license from Micro Focus is required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

This document was produced on November 07, 2019. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

Contents

Preface

Contacting Micro Focus Fortify Customer Support

If you have questions or comments about using this product, contact Micro Focus Fortify Customer Support using one of the following options.

To Manage Your Support Cases, Acquire Licenses, and Manage Your Account

https://softwaresupport.softwaregrp.com

To Call Support

1.844.260.7219

For More Information

For more information about Fortify software products:
https://software.microfocus.com/solutions/application-security

About the Documentation Set

The Fortify Software documentation set contains installation, user, and deployment guides for all Fortify Software products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. You can access the latest versions of these documents from the following Micro Focus Product Documentation website:

https://www.microfocus.com/support-and-services/documentation

Change Log

The following table lists changes made to this document.

Software Release / Document Version	Changes
19.2.0	**New topics** • "What's New in Micro Focus Fortify CloudScan 19.2.0" on page 16 • "Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 37 • "(Windows only) Configuring Sensors to Offload Translation For .NET Languages" on page 38 **Modified topics** • "Installing the CloudScan Controller" on page 19 was modified to reflect the new installation procedure used for installation on both Linux and Windows. • "Creating CloudScan Clients" on page 33was modified to reflect the introduction of the `CloudScan_Client_<version>.zip` file, which is used to create stand-alone clients that support translation on CloudScan sensors. • "Upgrading the CloudScan Controller" on page 42 was modified to reflect file name changes. • The procedure described in "Upgrading Fortify CloudScan Sensors" on page 43 was modified to reflect the fact that the `Fortify_CloudScan_Update_<version>_Linux.zip` and `Fortify_CloudScan_Update_<version>_windows_x64.zip` file are no longer available (or used) and have been replaced by the single file `Cloudscan_Client_<version>.zip`. • Information about how to use CloudScan to scan Python projects was added to "Submitting Scan Requests" on page 45. • New argument command options were added to "CloudScan Command Options" on page 62. **Removed topics** • Installing the CloudScan Controller on a Linux System

Software Release / Document Version	Changes
	• Installing the CloudScan Controller on a Windows System
19.1.0	**New topics** • What's New in Micro Focus Fortify CloudScan 19.1.0 • "CloudScan Command Options" on page 62 **Modified topics** • "Fortify CloudScan Components" on page 18 • "Creating CloudScan Clients" on page 33 • "Accessing Help for Command-Line Options" on page 45 • "Submitting Scan Requests" on page 45
18.20	Minor changes, including version number and the font used to display content.

Chapter 1: Introduction

With Fortify CloudScan (CloudScan), Fortify Static Code Analyzer users can better manage their resources by offloading code analysis tasks from their build machines to a cloud of machines (sensors) provided for this purpose.

You can start a Fortify Static Code Analyzer analysis of your code from a CloudScan client in one of two ways:

- You can perform the translation phase on a local or build machine to generate a mobile build session (MBS). The CloudScan client then hands off the MBS to the CloudScan Controller, which distributes the MBS to the CloudScan sensors. The sensors then perform the scanning phase of the analysis.
- If your application version is written in a language supported for centralized translation, you can also offload the translation phase of the analysis to your CloudScan sensors. For information about the languages supported for offloading translation, see "Creating CloudScan Clients" on page 33. For information about the specific language versions supported, see the *Micro Focus Fortify Software System Requirements* document.

If your code is written using a language other than one supported for offloading project translation, the translation phase (less processor- and time-intensive than the scanning phase) is completed on the build machine. After translation is completed, CloudScan generates a project package, which it then moves to a distributed cloud of machines (sensors) for scanning. In addition to freeing up build machines, this process makes it easy to add more resources to the cloud and grow the system as needed, without having to interrupt your build process. And, Fortify Software Security Center can direct CloudScan to output FPR files directly to the server.

This content provides information on how to install, configure, and use CloudScan to streamline your static code analysis process.

Intended Audience

This content is written for anyone who intends to install, configure, or use CloudScan to offload the translation (for supported languages) and scanning phases of the Fortify Static Code Analyzer process to CloudScan sensors.

Related Documents

This topic describes documents that provide information about Micro Focus Fortify software products.

Note: You can find the Micro Focus Fortify Product Documentation at https://www.microfocus.com/support-and-services/documentation. All guides are available in both PDF and HTML formats. Product help is available within the Fortify WebInspect products.

All Products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website.

Document / File Name	Description
About Micro Focus Fortify Product Software Documentation About_Fortify_Docs_<*version*>.pdf	This paper provides information about how to access Micro Focus Fortify product documentation. **Note:** This document is included only with the product download.
Micro Focus Fortify Software System Requirements Fortify_Sys_Reqs_<*version*>.pdf	This document provides the details about the environments and products supported for this version of Fortify Software.
Micro Focus Fortify Software Release Notes FortifySW_RN_<*version*>.pdf	This document provides an overview of the changes made to Fortify Software for this release and important information not included elsewhere in the product documentation.
What's New in Micro Focus Fortify Software <version> Fortify_Whats_New_<*version*>.pdf	This document describes the new features in Fortify Software products.

Micro Focus Fortify CloudScan

The following document provides information about Fortify CloudScan. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / File Name	Description
Micro Focus Fortify CloudScan Installation, Configuration, and Usage Guide	This document provides information about how to install, configure, and use Fortify CloudScan to streamline the static code analysis process. It is written

Document / File Name	Description
CloudScan_Guide_<*version*>.pdf	for anyone who intends to install, configure, or use Fortify CloudScan to offload the resource-intensive translation and scanning phases of their Fortify Static Code Analyzer process.

Micro Focus Fortify Software Security Center

The following documents provide information about Fortify Software Security Center. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-software-security-center.

Document / File Name	Description
Micro Focus Fortify Software Security Center User Guide SSC_Guide_<*version*>.pdf	This document provides Fortify Software Security Center users with detailed information about how to deploy and use Software Security Center. It provides all of the information you need to acquire, install, configure, and use Software Security Center. It is intended for use by system and instance administrators, database administrators (DBAs), enterprise security leads, development team managers, and developers. Software Security Center provides security team leads with a high-level overview of the history and current status of a project.

Micro Focus Fortify Static Code Analyzer

The following documents provide information about Fortify Static Code Analyzer. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-static-code.

Document / File Name	Description
Micro Focus Fortify Static Code Analyzer User Guide SCA_Guide_<*version*>.pdf	This document describes how to install and use Fortify Static Code Analyzer to scan code on many of the major programming platforms. It is intended for people responsible for security audits and secure coding.
Micro Focus Fortify Static Code Analyzer Custom Rules Guide	This document provides the information that you need to

Document / File Name	Description
SCA_Cust_Rules_Guide_<*version*>.zip	create custom rules for Fortify Static Code Analyzer. This guide includes examples that apply rule-writing concepts to real-world security issues. **Note:** This document is included only with the product download.
Micro Focus Fortify Audit Workbench User Guide AWB_Guide_<*version*>.pdf	This document describes how to use Fortify Audit Workbench to scan software projects and audit analysis results. This guide also includes how to integrate with bug trackers, produce reports, and perform collaborative auditing.
Micro Focus Fortify Plugins for Eclipse Installation and Usage Guide Eclipse_Plugins_Guide_<*version*>.pdf	This document provides information about how to install and use the Fortify Complete and the Fortify Remediation Plugins for Eclipse.
Micro Focus Fortify Plugins for IntelliJ, WebStorm, and Android Studio User Guide IntelliJ_AndStud_Plugins_Guide_<*version*>.pdf	This document describes how to install and use both the Fortify Analysis Plugin for IntelliJ IDEA and Android Studio and the Fortify Remediation Plugin for IntelliJ IDEA, Android Studio, and WebStorm.
Micro Focus Fortify Jenkins Plugin User Guide Jenkins_Plugin_Guide_<*version*>.pdf	This document describes how to install, configure, and use the plugin. This documentation is available at https://www.microfocus.com/documentation/fortify-jenkins-plugin.
Micro Focus Fortify Security Assistant Plugin for Eclipse User Guide SecAssist_Eclipse_Guide_<*version*>.pdf	This document describes how to install and use Fortify Security Assistant plugin for Eclipse to provide alerts to security issues as you write your Java code.
Micro Focus Fortify Extension for Visual Studio User Guide VS_Ext_Guide_<*version*>.pdf	This document provides information about how to install and use the Fortify extension for Visual Studio to analyze, audit, and remediate your code to resolve security-related issues in solutions and projects.
Micro Focus Fortify Static Code Analyzer Tools Properties Reference	This document describes the properties used by Fortify Static Code Analyzer tools.

Document / File Name	Description
Guide SCA_Tools_Props_Ref_<*version*>.pdf	

Micro Focus Fortify WebInspect

The following documents provide information about Fortify WebInspect. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-webinspect.

Document / File Name	Description
Micro Focus Fortify WebInspect Installation Guide WI_Install_<*version*>.pdf	This document provides an overview of Fortify WebInspect and instructions for installing Fortify WebInspect and activating the product license.
Micro Focus Fortify WebInspect User Guide WI_Guide_<*version*>.pdf	This document describes how to configure and use Fortify WebInspect to scan and analyze Web applications and Web services. **Note:** This document is a PDF version of the Fortify WebInspect help. This PDF file is provided so you can easily print multiple topics from the help information or read the help in PDF format. Because this content was originally created to be viewed as help in a web browser, some topics may not be formatted properly. Additionally, some interactive topics and linked content may not be present in this PDF version.
Micro Focus Fortify WebInspect on Docker User Guide WI_Docker_Guide_<*version*>.pdf	This document describes how to download, configure, and use Fortify WebInspect that is available on the Docker container platform. This full version of the product is intended to be used in automated processes as a headless scanner configured by way of the command line interface (CLI) or the application programming interface (API).
Micro Focus Fortify WebInspect Tools Guide WI_Tools_Guide_<*version*>.pdf	This document describes how to use the Fortify WebInspect diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect

Document / File Name	Description
	and Fortify WebInspect Enterprise.
Micro Focus Fortify WebInspect Runtime Agent Installation Guide WI_RT_Agent_Install_<*version*>.pdf	This document describes how to install the Fortify WebInspect Runtime Agent for applications running under a supported Java Runtime Environment (JRE) on a supported application server or service and applications running under a supported .NET Framework on a supported version of IIS.
Micro Focus Fortify WebInspect Agent Rulepack Kit Guide WI_Agent_Rulepack_Guide_<*version*>.pdf	This document describes the detection capabilities of Fortify WebInspect Agent Rulepack Kit. Fortify WebInspect Agent Rulepack Kit runs atop the Fortify WebInspect Runtime Agent, allowing it to monitor your code for software security vulnerabilities as it runs. Fortify WebInspect Agent Rulepack Kit provides the runtime technology to help connect your dynamic results to your static ones.

Micro Focus Fortify WebInspect Enterprise

The following documents provide information about Fortify WebInspect Enterprise. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-webinspect-enterprise.

Document / File Name	Description
Micro Focus Fortify WebInspect Enterprise Installation and Implementation Guide WIE_Install_<*version*>.pdf	This document provides an overview of Fortify WebInspect Enterprise and instructions for installing Fortify WebInspect Enterprise, integrating it with Fortify Software Security Center and Fortify WebInspect, and troubleshooting the installation. It also describes how to configure the components of the Fortify WebInspect Enterprise system, which include the Fortify WebInspect Enterprise application, database, sensors, and users.
Micro Focus Fortify WebInspect Enterprise User Guide WIE_Guide_<*version*>.pdf	This document describes how to use Fortify WebInspect Enterprise to manage a distributed network of Fortify WebInspect sensors to scan and analyze Web applications and Web services.

Document / File Name	Description
	Note: This document is a PDF version of the Fortify WebInspect Enterprise help. This PDF file is provided so you can easily print multiple topics from the help information or read the help in PDF format. Because this content was originally created to be viewed as help in a web browser, some topics may not be formatted properly. Additionally, some interactive topics and linked content may not be present in this PDF version.
Micro Focus Fortify WebInspect Tools Guide WI_Tools_Guide_<*version*>.pdf	This document describes how to use the Fortify WebInspect diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect and Fortify WebInspect Enterprise.

What's New in Micro Focus Fortify CloudScan 19.2.0

Micro Focus Fortify CloudScan 19.2.0 includes the changes described here.

CloudScan Packaging Utility Language Support

Fortify CloudScan now supports remote translation and scanning for .NET and Asp.Net projects.

The Fortify CloudScan utility that packages source code, dependencies, and Fortify Static Code Analyzer translation instructions, which was introduced with version 19.1.0, now supports the following additional languages:

- ABAP (Advanced Business Application Programming)
- Apex (Salesforce)
- Classic ASP (ASP Classic)
- Adobe ColdFusion
- PL/SQL / T-SQL
- Microsoft TypeScript
- Microsoft VBScript
- (Windows only) .NET applications, including C#, VB.NET, .NET Core, and .NET Standard

  For information about how to configure CloudScan sensors to work with .NET languages, see " (Windows only) Configuring Sensors to Offload Translation For .NET Languages" on page 38.

Note: These languages are in addition to Python, Ruby, JavaScript, PHP, and Java (through the Gradle or Apache Maven build tool).

This CloudScan utility also supports auto-packaging using MSBuild, in addition to the Gradle and Maven build tools.

Changes to Fortify CloudScan Installation and Upgrade

Starting with this release:

- A single `cloudscan_<version>.zip` file is used to install Fortify CloudScan clients on both Windows and Linux operating systems.
- To upgrade your Fortify CloudScan sensors, you no longer use the `Fortify_CloudScan_Update_<version>_Linux.zip` or `Fortify_CloudScan_Update_<version>_windows_x64.zip`. Now, you either install the latest version of Fortify Static Code Analyzer, or unzip the `cloudscan_<version>.zip` file. For more information, see "Upgrading Fortify CloudScan Sensors" on page 43.

Security Content Updates for CloudScan Sensors

When security content updates are required, CloudScan now communicates this to its sensors.

CloudScan Sensor Configuration for Java 11

If you start your CloudScan sensors on Java 11, and you want to use the `progress` command to check the progress of your Fortify Static Code Analyzer scans, some minor sensor configuration is required. For instructions, see "Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 37.

Chapter 2: Fortify CloudScan Components

A Fortify CloudScan installation includes the following three components:

- **CloudScan client** or **CLI**: A build machine on which Fortify Static Code Analyzer translates your code and generates Fortify Static Code Analyzer mobile build sessions (MBS). The translated source code, along with optional and required data, such as custom rules and Fortify Static Code Analyzer command-line arguments, are uploaded to the CloudScan Controller.

  The interface for issuing Fortify CloudScan commands is installed on your clients. You can use this interface to create or identify a Fortify Static Code Analyzer mobile build session, set the parameters for the scan, and communicate your intentions to the CloudScan Controller.

  Note: A thin client that does not require that Fortify Static Code Analyzer be installed may pack the code with dependencies into a package to send to the Controller for further translation and scanning.

- **CloudScan Controller**: Server that receives the Fortify Static Code Analyzer mobile build sessions and scan instructions from the CloudScan clients (or project packages with translation and scan instructions), routes the information to CloudScan sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center.
- **CloudScan sensors:** Distributed network of computers set up to receive Fortify Static Code Analyzer mobile build sessions (MBSs) and scan code using Fortify Static Code Analyzer. If your applications are written in a supported language, the sensors can also perform the translation phase of the analysis. For information about the languages supported for performing translation, see "Creating CloudScan Clients" on page 33.

Note: The minimum installation requires three physical or virtual machines: a Fortify CloudScan client, a sensor, and a Controller. A Fortify Software Security Center server is optional.

Note: As you set up your CloudScan environment, you can use subnets to segment your build machines from the cloud infrastructure. The build machines need only communicate with the CloudScan Controller, which in turn communicates with the cloud (sensors).

Chapter 3: Installing and Configuring the CloudScan Components

The following table lists the components, which, in addition to Fortify Static Code Analyzer, you must install and configure for CloudScan deployment. Install these components in the following order:

- CloudScan Controller
- CloudScan clients
- CloudScan sensors
- (Optional) Fortify Software Security Center

For information about hardware and software requirements for these components, see the *Micro Focus Fortify Software System Requirements* document.

This section contains the following topics:

Installing the CloudScan Controller

The CloudScan Controller (Controller) is a standalone server that sits between the CloudScan clients, sensors, and optionally, Fortify Software Security Center. The Controller accepts scan requests issued by the clients and passes them on to an available sensor. A sensor returns scan results to the Controller, which stores them temporarily.

Caution! Before you install the Controller, you must first download and configure a Java Runtime Environment (JRE). For information about supported JRE versions, see the *Micro Focus Fortify Software System Requirements* guide. For information about how to download and configure JRE, see the documentation for the supported JRE version.

Jobs are deleted from the Controller after seven days, unless you change the `job_expiry_delay` variable value of 168 hours in the `config.properties` file. (You can find the `config.properties` file in the *<cs_controller_dir>*`/tomcat/webapps/cloud-ctrl/WEB-INF/classes` directory.)

Caution! The name of the directory into which you install the Controller must not include spaces.

To install the CloudScan Controller (on a Linux or Windows system):

- Extract the contents of the `Fortify_CloudScan_Controller_`*<version>*`_x64.zip` file to a directory that does not include either the *<sca_install_dir>* or the *<ssc_install_dir>*.

Note: In this document, *<cs_controller_dir>* refers to the CloudScan Controller installation directory, *<sca_install_dir>* refers to the Fortify Static Code Analyzer installation directory, and *<ssc_install_dir>* refers to the Fortify Software Security Center server installation directory.

After you install the CloudScan Controller, *<cs_controller_dir>* resembles the following:

```
bin/

tomcat/

cloudscan.zip

readme.txt
```

Note: The `cloudscan.zip` file includes the CloudScan CLI.

Save the `cloudscan.zip` file to an accessible directory or USB key to use later to configure CloudScan sensors and standalone clients.

See Next

"Configuring the CloudScan Controller" on page 22

For information about how to update your Controller, see "About Upgrading Fortify CloudScan Components" on page 42 and "Upgrading the CloudScan Controller" on page 42.

See Also

"Installing and Uninstalling the CloudScan Controller as a Service" below

Installing and Uninstalling the CloudScan Controller as a Service

If you use Windows, you can install the CloudScan controller as a Windows service.

Installing the CloudScan Controller as a Service

To install the CloudScan controller as a service on a machine without other Tomcat instances running:

1. Log on to Windows as a local user with administrator privileges.
2. Check to make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
3. Check to make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the CloudScan `Tomcat` directory.
4. Navigate to the `<cs_controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat install
```

This creates a service with the name "Tomcat9."

To install the controller as a service with a different name:

1. Check to make sure that the `JRE_HOME` and `JAVA_HOME` environment variables are correctly configured.
2. Check to make sure that the `CATALINA_HOME` environment variable is either empty or set up to point to the CloudScan `Tomcat` directory.
3. Navigate to the `<cs_controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat install <service_name>
```

The service name must not contain any spaces.

Uninstalling the CloudScan Controller Service

To uninstall the Apache Tomcat 9.0 service:

1. Stop the service.
2. Navigate to the `<cs_controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat remove
```

To uninstall the controller as a service with a name other than Apache Tomcat 9.0:

1. Stop the service.
2. Navigate to the `<cs_controller_dir>/tomcat/bin` directory, and then run the following:

```
service.bat remove <service_name>
```

See Also

"Configuring the CloudScan Controller" on the next page

Configuring the CloudScan Controller

After you install the CloudScan Controller, edit global properties such as the email address to be used, the shared secret for the Controller (password that Fortify Software Security Center uses when it requests data from the CloudScan Controller), the shared secret for the sensor, and the Fortify Software Security Center URL (if you plan to upload your FPRs to Fortify Software Security Center).

Caution! To avoid potential conflicts, Fortify recommends that you run the Controller on a Tomcat Server instance other than the instance that Fortify Software Security Center uses.

To configure the CloudScan Controller:

1. Navigate to `<cs_controller_dir>/tomcat/webapps/cloud-ctrl/WEB-INF/classes`.
2. Open the `config.properties` file in a text editor, and then configure the properties listed in the following table.

Option	Description
`worker_auth_token`	A string that contains no spaces or backslashes. If you prefer not to use plain text, you can use an encrypted shared secret as the value for this property. For instructions on how to encrypt a shared secret, "Encrypting the Shared Secret on the Controller" on the next page.
`ssc_url`	URL for the Fortify Software Security Center server; all uploads are sent to this address. Example: `https://<ssc_host>:<port>/ssc`
`this_url`	URL for the CloudScan Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/cloud-ctrl`
`ssc_cloudctrl_secret`	Password that Fortify Software Security Center uses to request data from the CloudScan Controller. Specify a string that contains no spaces or backslashes. (Optional) Use an encrypted shared secret. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret" on the next page.
`pool_mapping_mode`	Used to configure different modes for mapping scan requests to sensor pools. For information about the valid values for `pool_mapping_mode`, see "About the pool_mapping_mode Property" on page 25.
If your remote IP address is different than the configured Fortify Software Security Center URL,	

Option	Description
you can use one of the following properties to set up the remote IP address.	
`ssc_remote_ip`	Remote IP address
`ssc_remote_ip_trusted_proxies_range`	Remote IP range (in CIDR format)
`ssc_remote_ip_header`	Remote IP HTTP header The default value is `X-Forwarded-For`.
`remote_ip_proxy_header`	Remote IP proxy header
`ssc_trusted_proxies_remote_ip`	If `remote_ip_proxy_header` is set, you must also specify a value for this property.

3. Save and close your `config.properties` file.
4. Start the CloudScan Controller. (For instructions, see "Starting the Fortify CloudScan Components" on page 40.)

See Also

"Installing the CloudScan Controller" on page 19

Encrypting the Shared Secret

Passwords exist in the CloudScan Controller and sensor configuration files as plain text. If you prefer to encrypt your passwords, you can.

You can use encrypted keys as values for the `worker_auth_token`, `smtp_auth_pass` and `ssc_cloudctrl_secret` properties in the `config.properties` file on the Controller, and as the value for `worker_auth_token` in the `worker.properties` file on a sensor.

Note: For the sake of security, make sure that the `pwtools.key` file you use to encrypt secrets for sensors is different from the `pwtools.key` file you use to encrypt secrets on the Controller.

Encrypting the Shared Secret on the Controller

To encrypt a shared secret on the Controller:

1. Run one of the following:
   - On a Windows system, `<cs_controller_dir>\bin\pwtool.bat <path_to_pwtool.keys>`
   - On a Linux system, `<cs_controller_dir>/bin/pwtool <path_to_pwtool.keys>`
2. When prompted, type the password to encode, and then press **Enter**.

   The pwtool generates a new `pwtool.keys` file to `<path_to_pwtool.keys>` and prints a new encrypted secret to the console.
3. Copy the new encrypted secret, and paste it as the value for one of the following properties in the `config.properties` file:
   - `worker_auth_token`
   - `smtp_auth_pass`
   - `ssc_cloudctrl_secret`

   Tip: Fortify recommends that you assign separate, unique shared secrets for the `worker_auth_token`, `smtp_auth_pass`, and `ssc_cloudctrl_secret` properties.
4. When you create an encrypted secret with the `pwtool.keys` file, the output begins with `{fp0}`. For the `worker_auth_token` in the `config.properties` file, *omit* `{fp0}` from the encrypted value. For the `ssc_cloudctrl_secret` in the `config.properties` file, you must *include* `{fp0}` in the encrypted value.
5. Create two additional encrypted shared secrets (steps 1 and 2) and, in the `config.properties` file, paste these as values for the two properties to which you did not already assign an encrypted secret in step 3.
6. Uncomment the following line (property) in the `config.properties` file, and then save the file:
   `#pwtool_keys_file=${catalina.base}/pwtool.keys`

Encrypting the Shared Secret on a Sensor

To encrypt a shared secret on a sensor:

1. Run one of the following:
   - On a Windows system, `<sca_install_dir>\bin\pwtool.bat <path_to_pwtool.keys>`
   - On a Linux system, `<sca_install_dir>/bin/pwtool <path_to_pwtool.keys>`
2. When prompted, type the password to encode, and then press **Enter**.

   The pwtool generates a new `pwtool.keys` file to `<path_to_pwtool.keys>` and prints a new encrypted secret to the console.
3. Copy the encrypted secret, and paste it as the value for `worker_auth_token` property in the `worker.properties` file.
4. Add the following line (property) to the `worker.properties` file, and then save the file:

   `pwtool_keys_file=<path_to_pwtool.keys>`

See Also

"Configuring the CloudScan Controller" on page 22

"Creating CloudScan Sensors" on page 35

About the pool_mapping_mode Property

The `pool_mapping_mode` property in the `config.properties` file determines how the system maps scan requests to sensor pools. Valid values for the `pool_mapping_mode` property are as follows:

- `DISABLED`—This is the default value. It is compatible with Fortify Software Security Center 16.10 and earlier versions. In this mode, a CloudScan client *can* request a specific sensor pool when it submits a scan request. Otherwise, the default pool is used. The Controller behaves the same in disabled mode as it behaved in versions earlier than 16.20.
- `ENABLED`—You can use this mode only with Fortify Software Security Center 16.20 and later versions. In this mode, if a scan request is associated with an application version in Fortify Software Security Center, the Controller queries Fortify Software Security Center to determine the sensor pool assigned to the application version. Or, a CloudScan client can request a specific sensor pool when it submits a scan request. (A client request for a specific sensor pool takes precedence over a query from the Controller.)

  Note: Sensors in the default sensor pool run scan requests that are not associated with an application version (and no specific pool is requested on the CloudScan client command line).

- `ENFORCED`—You can use this mode only with Fortify Software Security Center 16.20 and later versions. As with the ENABLED mode, if a scan request is associated with an application version in Fortify Software Security Center, the Controller queries Fortify Software Security Center for the sensor pool to use for the application version. Otherwise, the default sensor pool is targeted for scan requests. A client cannot request a specific sensor pool in the ENFORCED mode.

The following table shows how the Fortify Software Security Center integration with Fortify CloudScan responds to different input when `pool_mapping_mode` is set to `DISABLED`, `ENABLED`, or `ENFORCED`.

Note: By default, in enabled and enforced modes, all application versions are assigned to the Default pool.

INPUT	DISABLED	ENABLED	ENFORCED
No pool or version specified	Default sensor pool	Default sensor pool	Default sensor pool
Specific sensor pool (only) specified	Requested sensor pool	Requested sensor pool	Denied
Application version (only) specified	Default sensor pool	SSC-assigned pool	SSC-assigned pool
Invalid sensor pool (only) specified	Denied	Denied	Denied
Invalid application version (only)	Default pool	Denied	Denied

INPUT	DISABLED	ENABLED	ENFORCED
specified			
Valid sensor pool and application version specified	Requested sensor pool	Requested sensor pool	Denied
Invalid sensor pool and valid application version specified	Denied	Denied	Denied
Valid sensor pool but invalid application version specified	Requested sensor pool	Requested sensor pool	Denied

See Also

"Configuring the CloudScan Controller" on page 22

Securing CloudScan Deployment

The Micro Focus Fortify family of products collects and displays information about an enterprise's applications. That information includes summaries of the potential security vulnerabilities uncovered in the source code.

Just as you apply security precautions to your applications, you must also secure access to the CloudScan components. The security vulnerability summaries that Fortify products provide may mandate an even higher level of secure deployment.

CloudScan works with your code base. Because this information offers various opportunities for mishandling or abuse, Fortify recommends that you deploy CloudScan in a secure operations facility and secure access to CloudScan installation directories.

Securing the CloudScan Controller

The following procedure describes how to create a secure connection (HTTPS) between the CloudScan Controller/Tomcat server and CloudScan CLI. This procedure requires either a self-signed certificate or a certificate signed by a certificate authority such as VeriSign.

To create a secure connection (HTTPS) between the CloudScan Controller/Tomcat server and CloudScan CLI, use one of the following procedures.

Note: The following sections show *examples* of how to create a connection. For the most current information, see your Apache Tomcat documentation.

"Creating a Secure Connection Using Self-Signed Certificates" on the next page

"Creating a Secure Connection Using a Certificate Signed by a Certificate Signing Authority" on page 29

Creating a Secure Connection Using Self-Signed Certificates

To enable SSL on Tomcat using a self-signed certificate:

1. To generate a keystore that contains a self-signed certificate, open a command prompt and run one of the following `Java keytool` commands:
   - On a Windows system:

     ```
     %JAVA_HOME%\bin\keytool -genkey -alias <alias_name> -keyalg RSA -keystore <mykeystore>
     ```

   - On a Linux system:

     ```
     $JAVA_HOME/bin/keytool -genkey -alias <alias_name> -keyalg RSA -keystore <mykeystore>
     ```

2. Provide values for the prompts listed in the following table.

Prompt	Value
Enter keystore password:	Type a secure password.
Re-enter new password:	Re-type your secure password.
What is your first and last name?	Type your hostname. You can use your fully-qualified domain name here. **Note:** If you plan to provide an IP address as the hostname, then you must also provide the `-ext san=ip:<ip_address>` parameter to keytool. Without the `-ext san=ip:<ip_address>` parameter, the SSL handshake fails.
What is the name of your organizational unit?	Name to identify the group that is to use the cert.
What is the name of your organization?	Name of your organization.
What is the name of your City or Locality?	City or locality in which your organization is located.
What is the name of your State or Province?	State or province in which your organization is located.

Prompt	Value
What is the two-letter country code for this unit?	If your server is located in the United States, type **US**.
Confirm your entries:	Type yes to confirm your entries.
Enter key password for <tomcat><Return if same as keystore password>:	Password for your Tomcat server key. Press **Return / Enter** to use the same password you established for your keystore. (Fortify recommends that you create a new key password.)
Re-enter new password:	Re-type your key password.

3. To export the certificate from the Tomcat keystore, open a command prompt and type one of the following:
   - On a Windows system:

     %JAVA_HOME%\bin\keytool -export -alias <*alias_name*> -keystore <mykeystore> -file YourCertFile.cer

   - On a Linux system:

     $JAVA_HOME/bin/keytool -export -alias <*alias_name*> -keystore <mykeystore> -file YourCertFile.cer

4. Add the following connector to the `server.xml` file in the `tomcat\config` directory:

```
<Connector port="8443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="<mykeystore>" "keystorePass="<mypassword>"
clientAuth="false" sslProtocol="TLS"/>
```

   Note: The default `server.xml` file installed with Tomcat includes an example `<connector>` element for an SSL connector.

5. Navigate to one of the following directories, and then open the `config.properties` file in a text editor:
   - (Windows) *<cs_controller_dir>*`\tomcat\webapps\cloud-ctrl\WEB-INF\classes`
   - (Linux) *<cs_controller_dir>*`/tomcat/webapps/cloud-ctrl/WEB-INF/classes`
6. Update the `this_url` property, with your https address and port.

   Example: `this_url=https://`*<controller_host>*`:8443/cloud-ctrl`

7. Restart your Tomcat server.
8. Set up your CloudScan clients and sensors. For information about how to set up the CloudScan clients and sensors, see "Creating CloudScan Clients" on page 33 and, "Creating CloudScan

Sensors" on page 35, respectively.

9. Add your self-signed certificate to the java keystore on all entities that communicate with the CloudScan Controller (includes all CloudScan clients, CloudScan sensors, and Fortify Software Security Center installations) as follows:
   a. For CloudScan clients and CloudScan sensors, open a command prompt and type the following:

      ```
      cd <sca_install_dir>\jre\bin
      ```

      Where *<sca_install_dir>* is the directory where the CloudScan sensor or CloudScan client is installed.

      For a Fortify Software Security Center installation or for standalone CloudScan clients, open a command prompt and type one of the following:

      - On Windows:

        ```
        cd %JAVA_HOME%\jre\bin
        ```

      - On Linux:

        ```
        cd $JAVA_HOME/jre/bin
        ```

   b. Run the following command:

      ```
      keytool -import -alias <aliasName> -keystore ..\lib\security\

      cacerts -file YourCertFile.cer –trustcacerts
      ```

      Where `YourCertFile.cer` is the same certificate file that you exported in step 1.

Creating a Secure Connection Using a Certificate Signed by a Certificate Signing Authority

To enable SSL on Tomcat using a certificate signed by a certificate signing authority:

1. Use the Java keytool to generate a new keystore containing a self-signed certificate, as follows:
   - On a Windows system:

     ```
     %JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA -keystore
     "<mykeystore>"
     ```

   - On a Linux system:

     ```
     $JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA -keystore
     "<mykeystore>"
     ```

2. The keytool prompts you for the information described in the following table.

Prompt	Data
Enter keystore password:	Type a secure password.
Re-enter new password:	Re-enter your secure password.
What is your first and last name?	Type your hostname. You can use your fully qualified domain name here. **Note:** If you plan to enter an IP address as the hostname, then you will also need to pass an additional parameter to keytool, `-ext san=ip:<ipaddress>`. Without this additional parameter, the SSL handshake fails.
What is the name of your organizational unit?	Type the name of the group that is to use the certificate. (This can be anything you want.)
What is the name of your organization?	Type the name of your organization (This can be anything you want.)
What is the name of your City or Locality?	Type the city or locality. (This can be anything you want.)
What is the name of your State or Province?	Type the state or province. (This can be anything you want.)
What is the two-letter country code for this unit?	If your server is located in the United States, type **US**.
Confirm your entries:	Type **yes** to confirm your entries.
Enter key password for `<tomcat><Return if same as keystore password>`:	Type a password for your Tomcat server key, or press **Return** to use the same password you established for your keystore. Fortify recommends that you create a new password.
Re-enter new password:	Re-type your key password.

3. Generate a Certificate Signing Request (CSR).

   To obtain a certificate from a certificate signing authority, you must generate a Certificate Signing Request (CSR). The certificate authority uses the CSR to create the certificate. Create the CSR as follows:

   On a Windows system:

```
%JAVA_HOME%\bin\keytool -certreq -alias <alias_name> -keyalg RSA -file
"yourCSRname.csr" -keystore "<mykeystore>"
```

- On a Linux system:

```
$JAVA_HOME/bin/keytool -certreq -alias <alias_name> -keyalg RSA -file
"yourCSRname.csr" -keystore "<mykeystore>"
```

4. Send the CSR file to the certificate signing authority you have chosen.
5. Once you receive your certificate from the certificate signing authority, import it into the keystore that you created, as follows:
   - On a Windows system:

```
%JAVA_HOME%\bin\keytool -import -alias  <alias_name> -trustcacerts -file
"YourVerisignCert.crt"
 -keystore "<mykeystore>"
```

   - On a Linux system:

```
$JAVA_HOME/bin/keytool -import -alias  <alias_name> -trustcacerts -file
"YourVerisignCert.crt"
 -keystore "<mykeystore>"
```

   The root CA already exists in the `cacerts` file of your JDK, so you are just installing the intermediate CA for your certificate signing authority.

   Note: If you purchased your certificate from VeriSign, you must first import the chain certificate. You can find the specific chain certificate on the VeriSign website or click the link for the chain certificate in the email you received from VeriSign with your certificate.

   - On a Windows system:

```
%JAVA_HOME%\bin\keytool -import -alias IntermediateCA -trustcacerts-
file "chainCert.crt" -keystore "<mykeystore>"
```

   - On a Linux system:

```
$JAVA_HOME/bin/keytool -import -alias IntermediateCA -trustcacerts -
file "chainCert.crt" -keystore "<mykeystore>"
```

6. Add the following connector to the `server.xml` file in the `tomcat\config` directory:

```
<Connector port="8443" maxThreads="200"

scheme="https" secure="true" SSLEnabled="true"

keystoreFile="<mykeystore>" keystorePass="<mypassword>"

clientAuth="false" sslProtocol="TLS"/>
```

Note: An example `<Connector>` element for an SSL connector is included in the default `server.xml` file installed with Tomcat.

7. Restart Tomcat Server.
8. In the `config.properties` file, update the `this_url` property with your secure URL:
   a. Navigate to the `config.properties` file and open it in a text editor.
      On a Windows system:

      ```
      <cs_controller_dir>\tomcat\webapps\cloud-ctrl\WEB-
      INF\classes\config.properties
      ```

      On a Linux system:

      ```
      <cs_controller_dir>/tomcat/webapps/cloud-ctrl/WEB-
      INF/classes/config.properties
      ```

   b. Update the `this_url` property with your https address and port.

      Example: `this_url=https://<controller_host>:8443/cloud-ctrl`

Support for Multiple Fortify Static Code Analyzer Versions

To support heterogeneous environments and facilitate phased Fortify Static Code Analyzer upgrades, the CloudScan Controller supports scan request routing based on Fortify Static Code Analyzer version. For example, you can configure two different client machines, each with a different Fortify Static Code Analyzer version, and configure the Fortify CloudScan sensors with compatible Fortify Static Code Analyzer versions. Jobs from each client are then routed to the sensor that has the same Fortify Static Code Analyzer version installed.

If you have an existing Fortify Static Code Analyzer installation (with an included `cloudscan.bat`) in your path and a mixed version environment, make sure that you are running the latest CloudScan executable when you run the CloudScan client and CloudScan sensor commands. (Use explicit paths.) Adding capacity (new clients or sensors) is simple—just clone the VMs you have already configured, or use sensor hosts with the same specifications and installation folder structure.

Important! If you clone VMs, you *must* remove the `worker_persist.properties` file from sensor work directory (current directory when starting sensor) after cloning.

Note: Use CloudScan sensor machines dedicated to CloudScan and run CloudScan sensors under a dedicated username. Run only one CloudScan sensor instance per machine, and do not run any other Java processes under the same username after you start the CloudScan.

If the Controller and Fortify Software Security Center run on different machines, you must check to make sure that `cloud-ctrl\WEB-INF\classes\config.properties (ssc_url, this_url)`

and the CloudScan Controller URL set on Fortify Software Security Center (select **Administration** > **Configuration** > **CloudScan**) resolve to the correct IP addresses.

Check to make sure that the following channels of communication are not blocked by a firewall or other tool:

- CloudScan Controller to Fortify Software Security Center port (for scan uploads)
- Fortify Software Security Center to the CloudScan Controller port (for Fortify CloudScan administration console functionality)
- CloudScan clients to the CloudScan Controller port
- CloudScan sensors to the CloudScan Controller port
- CloudScan clients to the Fortify Software Security Center port (required only if Fortify Software Security Center is in lock down mode, or if the `-sscurl` option is used)

Creating CloudScan Clients

Unless you use a language that supports offloading the translation phase of analysis to your sensors, you must have a licensed copy of Fortify Static Code Analyzer on each of the machines you plan to use as CloudScan clients. If you use a language that supports offloading the translation phase of analysis to your sensors, you can create standalone clients, independent of Fortify Static Code Analyzer.

The languages that support offloading the translation phase of analysis are:

- Python
- Ruby
- JavaScript
- PHP
- Java
- ABAP (Advanced Business Application Programming)
- Apex (Salesforce)
- Classic ASP (ASP Classic)
- Adobe ColdFusion
- PL/SQL / T-SQL
- Microsoft TypeScript
- Visual Basic 6.0
- .NET applications (C#, VB.NET, .NET Core, and .NET Standard)

Caution! As you specify an installation path, make sure that the path name contains no spaces.

Creating a Standalone Client

If you plan to offload both the translation and scanning phases of analysis to your CloudScan sensors, you can use standalone clients.

To create a standalone client (independent of Fortify Static Code Analyzer):

- Extract the contents of the `Fortify_CloudScan_Client_<version>_x64.zip` file to any directory on your machine.

Creating a Client Using Static Code Analyzer 19.2.0

Use the following procedure to create a CloudScan client if:

- You do *not* use a language that supports offloading translation.

  and/or
- You do *not* plan to offload project translation to your sensors.

To create a client:

1. Log on to a build machine using credentials for an account that is *not* an administrator or root account.
2. Use the instructions provided in the *Micro Focus Fortify Static Code Analyzer User Guide* to install Fortify Static Code Analyzer and applications on your build machine.

Updating a CloudScan Client

Important! If your CloudScan Controller version is later than your Fortify Static Code Analyzer installation version, Fortify recommends that you update the CloudScan client to the same version as the CloudScan Controller. This ensures you are running the most recent code.

To update a standalone CloudScan client (independent of Fortify Static Code Analyzer):

- Delete the client, and then extract the `Fortify_CloudScan_Client_<version>_x64.zip` file to any directory on the machine.

  Or,
- Extract the contents of the `Fortify_CloudScan_Client_<version>_x64.zip` file on top of the existing client.

To update a CloudScan client that resides on the same machine as Fortify Static Code Analyzer:

1. Log on to the build machine using credentials for an account that is *not* an administrator account or root.
2. Back up the following directories:

   On a Windows system:

   - `<sca_install_dir>\bin`
   - `<sca_install_dir>\Core\lib`
   - `<sca_install_dir>\Core\config`

   On a Linux system:

   - `<sca_install_dir>/bin`

- `<sca_install_dir>/Core/lib`
- `<sca_install_dir>/Core/config`

3. Upgrade Fortify Static Code Analyzer. For instructions on how to install and upgrade Fortify Static Code Analyzer, see the *Micro Focus Fortify Static Code Analyzer User Guide*.
4. Accept all overwrite requests.

Note: On a Linux system, you may also need to run `chmod +x cloudscan` (in the `<sca_install_dir>/bin/cloudscan` directory).

Tip: After you configure a client, you can copy the configuration files and use them to create other clients.

See Also

"(Windows only) Configuring Sensors to Offload Translation For .NET Languages" on page 38

"Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 37

Creating CloudScan Sensors

To make it convenient for network administrators to isolate traffic to CloudScan sensors, Fortify recommends that you install CloudScan sensors in a separate subnet. Use the sensors only as scan boxes. CloudScan supports only one sensor per machine.

Creating a CloudScan Sensor Using Static Code Analyzer 19.2.0

The following procedure describes how to create a new sensor. For information about how to upgrade an existing sensor, see "Upgrading Fortify CloudScan Sensors" on page 43.

Note: If you use Windows, you can install the sensor as a Windows service. For instructions, see "Creating a CloudScan Sensor as a Service" on the next page.

To create a CloudScan sensor:

1. Log in to the build machine using an account that is not an administrator or root.
2. Install Fortify Static Code Analyzer 19.2.0. (For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.)
3. Create a file named `worker.properties` in the `<sca_install_dir>\Core\config` directory.
4. Add the following property to the `worker.properties` file:

```
worker_auth_token=<value_set_in_controller_configuration>
```

5. Specify either a clear text password, or an encrypted shared secret (password the Controller uses to communicate with the sensor) as the `worker.properties` value. For information about how to generate an encrypted shared secret, see "Encrypting the Shared Secret on a Sensor" on

page 24.

6. Save and close your `worker.properties` file.

Updating a Sensor Based on a Fortify Static Code Analyzer Version Earlier than 19.2.0

If your CloudScan Controller version is later than your Fortify Static Code Analyzer installation version, Fortify recommends that you update the CloudScan sensor so that it is the same version as the CloudScan Controller. This ensures you are running the most recent code.

To create a sensor using a Fortify Static Code Analyzer version earlier than 19.2.0:

1. Log in to the build machine using an account that is not an administrator or root.
2. Install Fortify Static Code Analyzer on the build machine if it does not already have Fortify Static Code Analyzer installed. For more information about how to install Fortify Static Code Analyzer, see the *Micro Focus Fortify Static Code Analyzer User Guide*.
3. Back up the following directories:
   - `<sca_install_dir>\bin`
   - `<sca_install_dir>\Core\lib`
   - `<sca_install_dir>\Core\config`
4. Extract the contents of the `cloudscan.zip` file to the `<sca_install_dir>\Core\config` directory (`<sca_install_dir>/Core/config` on Linux).
5. Accept all overwrite requests.

   Note: Linux users may also need to run `chmod +x cloudscan` in the `bin` directory.
6. In the `<sca_install_dir>\Core\config` directory (`<sca_install_dir>/Core/config` on Linux), create a file named `worker.properties`.
7. In the `worker.properties` file, create the following property:

   `worker_auth_token=<shared_secret>`

Creating a CloudScan Sensor as a Service

If you use Windows services, you can install the sensor as a Windows service.

To install the sensor as a Windows service:

1. Navigate to the `<sca_install_dir>\bin\cloudscan-worker-service` directory, and then do one of the following:
   - To use a clear text password, run `setupworkerservice.bat <sca_version> <full_cs_controller_url> <shared_secret>`
   - To use an encrypted password, run `setupworkerservice.bat <sca_version> <full_cs_controller_url> "<encrypted_shared_secret>" <path_to_pwtool.keys_`

file>

Important! Make sure that you enclose <encrypted_ shared_secret> in quotation marks. This ensures that the encrypted shared secret does not get corrupted when the services installer creates the worker.properties file.

For information about how to encrypt a shared secret, see "Encrypting the Shared Secret on a Sensor" on page 24.

2. Start the service, as follows:

```
net start FortifyCloudscanWorkerService
```

The services installer creates the C:\CloudscanWorkdir\worker.properties file for you.

See Next

"Enabling CloudScan Sensor Auto-Start on Windows as a Service" on page 55

See Also

"Fortify CloudScan Components" on page 18

"Creating CloudScan Sensors" on page 35

Configuring Sensors to Use the Progress Command when Starting on Java 11

If you plan to start your CloudScan sensors on Java 11, and you want to use the progress command to check the progress of your Fortify Static Code Analyzer scans, the following sensor configuration is required:

1. Create a JMX access file, and add the following text to it:

   <*user_role*> readonly

   where <*user_role*> is text that represents something like a username.
2. Create a JMX password file, and add the following text to it:

   <*user_role*> <*password*> readonly

   where <*user_role*> is the value you specified in the JMX access file.
3. Run one of the following commands:
   - On Windows systems, run cacls jmxremote.password /P <*username*>:R
   - On Linux systems, run chmod 600 jmxremote.password
4. Open the worker.properties file in a text editor, and then add the following properties to it:

   sca_jmx_port=<*port*>
   sca_jxm_access_file=<*path_to_access_file*>
   sca_jmx_password_file=<*path_to_password_file*>
   sca_jmx_password=<*password*>
   sca_jxm_user=<*user_role*>

```
sca_jmx_auth=true
```

5. Save and close the `worker.properties` file.

After you complete this configuration, CloudScan clients starts on the specified port using JMX password authentication. Make sure that the port is not already bound.

> **Important!** If you use `sca_jmx_auth`, you can start only one CloudScan sensor. Any attempt to open a new Fortify Static Code Analyzer instance results in a bind port error. To have multiple sensors on a machine, you must have several CloudScan instances, each with its own `worker.properties` file.

(Windows only) Configuring Sensors to Offload Translation For .NET Languages

If you plan to use your CloudScan sensors for remote translation of code written in a .NET language, make sure that the following requirements are met.

CloudScan client machine requirements:

- Java 8
- MSBuild (version that corresponds to the version released with Visual Studio 2017, or earlier)
- NuGet (optional)
- .NET Framework, .NET Core, or .NET Standard, depending on project configuration
- Windows operating system

CloudScan sensor machine requirements:

- Java 8
- .NET Framework supported for Fortify Static Code Analyzer
- Windows operating system

Beginning with version 19.20, CloudScan supports remote translation and scanning for .NET and Asp.Net projects. CloudScan supports the same MSBuild versions as Fortify Static Code Analyzer. (.NET packaging and scanning works only on Windows systems.)

The requirements for using this feature are as follows:

- Configure at least one sensor with the software required to support .NET capability.
- CloudScan clients must have the software required to build and pack .NET projects installed .

Enabling .NET Translation Capability on Sensors

To enable remote translation of .NET, do the following:

- Install the .NET Framework version that Fortify Static Code Analyzer supports. (See the *Micro Focus Fortify Software System Requirements* document.)

After you start CloudScan, it automatically detects the .NET Framework version installed and displays a message that .NET capability is enabled for the detected .NET Framework version. This indicates that

the sensor can now translate .NET projects built with same or earlier .NET Framework version. The rule is not applied to .NET Core or .NET Standard because any .NET Framework version can scan this kind of project.

Remote translation of .NET is disabled if:

- .NET Framework is not installed on the sensor.
- A .NET Framework version earlier than the supported version (for Fortify Static Code Analyzer) is installed on the sensor.

Important! To avoid Windows errors caused by too long a path during .NET translation, Fortify strongly recommends that you start CloudScan sensors from a folder with a short name and path. For more information, see https://docs.microsoft.com/en-us/windows/win32/fileio/naming-a-file.

Using the MSBuild - CloudScan Integration

To use MSBuild CloudScan integration, the required MSBuild version must be on the PATH. To make sure the project is built correctly, Fortify recommends that you start CloudScan from the Visual Studio command prompt, which sets the required .NET variables automatically.

Some projects also require that NuGet be started to restore some dependencies. If any dependencies are unresolved, the MSBuild would fail and the scan results might be incomplete. For these kinds of projects, you need to install NuGet manually on the machine and make sure it is available on the PATH. If NuGet is found, CloudScan will run it automatically.

To translate and scan a .NET project on CloudScan, run the following:

```
cloudscan -url <cloudscan_url> start --build-tool msbuild --build-file
<solution file name or path to solution file> [--save-package
```

Note that `--build-file` is required for .NET projects because the solution name is a custom-named file and CloudScan does not try to detect the `*.sln` file.

Alernatively, you can save the project package locally, as follows:

```
cloudscan package -o <path to package> --build-tool msbuild --build-file
<solution file>
```

To send the package to CloudScan, run:

```
cloudscan -url <cloudscan_url> start -package <package path>
```

CloudScan returns a job ID that you can use to track the scan.

Fortify Static Code Analyzer Mobile Build Session Version Compatibility

The Fortify Static Code Analyzer version on a CloudScan client must be compatible with the Fortify Static Code Analyzer version installed on the sensors. The version number format is

`major.minor+patch.buildnumber` (for example 19.20.0080). The major and minor portions of the Fortify Static Code Analyzer version numbers on both the CloudScan client and sensor must match. For example, 19.20 works with 19.2x.

To check the Fortify Static Code Analyzer version used, run the command `sourceanalyzer.exe -version`.

Starting the Fortify CloudScan Components

Before you begin to use Fortify CloudScan:

1. Wait until the CloudScan Controller is up and running.
2. (Optional) Wait until Fortify Software Security Center is up and running.
3. Check to make sure that the sensors and clients are up and running.

Starting the CloudScan Controller

To start the CloudScan Controller:

1. On the machine that hosts the CloudScan Controller, navigate to the Tomcat `<bin>` directory:
   On a Windows system:

   ```
   cd <cs_controller_dir>\tomcat\bin
   ```

   On a Linux system:

   ```
   cd <cs_controller_dir>/tomcat/bin
   ```

2. Run one of the following commands:
   - On a Windows system, run `startup.bat`.

     Note: If Tomcat is running as a service, rather than running `start.bat`, you can just start the service.

   - On a Linux system, run `./startup.sh`.

Starting CloudScan Sensors

To start the CloudScan sensors:

1. Start the Controller if it is not already running.
2. On each sensor, navigate to the `cs_worker_dir` directory of the installation directory, as follows:
   - On a Windows system, `cd <cs_worker_dir>\bin`
   - On a Linux system, `cd <cs_worker_dir>/bin`
3. Run one of the following commands:
   On a Windows system:

```
cloudscan.bat -url http://<controller_host>:8080/cloud-ctrl worker
```

On a Linux system:

```
./cloudscan -url http://<controller_host>:8080/cloud-ctrl worker
```

If the sensor starts successfully, it prints messages that signal its waiting status to the console. After you verify that the sensor is working, you can create a Startup Task in Windows Task Scheduler or add it to your startup scripts. For more information, see "Configuring Sensor Auto-Start" on page 55.

Note: Make sure that you run a given sensor consistently from the same directory. Otherwise, its UUID changes and, if Fortify CloudScan is connected to Fortify Software Security Center, Fortify Software Security Center identifies it as different sensor.

Starting Fortify Software Security Center

Start Fortify Software Security Center. If Fortify CloudScan is integrated with Fortify Software Security Center, after you log in to Fortify Software Security Center, notice that the Fortify header now includes the **SCANS** link. If you do not see the **SCANS** link in the header, log out, open a new browser window, and then log in again. If the **SCANS** link is still missing from the header, check to make sure that the connection between Fortify Software Security Center and Fortify CloudScan is set up. (See "Configuring the Connection to Fortify Software Security Center" on page 52.)

Stopping the CloudScan Controller

To stop the CloudScan Controller:

1. On the machine where the CloudScan Controller is installed, navigate to the Tomcat `bin` directory:
   On a Windows system:

   ```
   cd <cs_controller_dir>\tomcat\bin
   ```

   On a Linux system:

   ```
   cd <cs_controller_dir>/tomcat/bin
   ```

2. Type one of the following commands:
   On a Windows system:

   ```
   shutdown.bat
   ```

   On a Linux system:

   ```
   ./shutdown.sh
   ```

Chapter 4: About Upgrading Fortify CloudScan Components

Fortify CloudScan-related functionality in Fortify Software Security Center requires an updated CloudScan Controller and sensors. If you do not need sensor metrics, you can use sensor versions earlier than version 16.10. You can use existing Fortify CloudScan clients without limiting functionality (unless you want to specify that a scan request from a client target a specific sensor pool). If you need remote translation and scan functionality, use CloudScan client, sensor, and Controller version 19.1.0 or later.

Important! You must upgrade the Controller before you upgrade the Fortify CloudScan sensors and clients, *and* before you upgrade the Fortify Software Security Center server.

Caution! A version 19.2.0 CloudScan sensor does not support packages generated by version 19.1.0 clients. If you want to scan projects uploaded by CloudScan client 19.1.0, do not upgrade your sensors to version 19.2.0.

1. Copy data from the old Controller to the new Controller. Make sure that you merge your existing `config.properties` file with the new `config.properties` file.
2. Start the new Controller. (The database is automatically migrated.)

This section contains the following topics:

Upgrading the CloudScan Controller

The following procedure described how to upgrade a CloudScan Controller.

Caution! Before you upgrade the Controller, you must first download and configure a Java Runtime Environment (JRE). For information about supported JRE versions, see the *Micro Focus Fortify Software System Requirements* guide. For information about how to download and configure JRE, see the Oracle documentation for the supported JRE version.

To upgrade your CloudScan Controller:

1. Go to one of the following Software Licenses and Downloads Portal sites:
   - https://entitlement.microfocus.com
   - https://entitlement.mfgs.microfocus.com (for US Government solutions)
2. Download the `Fortify_CloudScan_Controller_<version>_x64.zip` file.

Note: For detailed instructions on how to download Micro Focus Software, see https://www.brainshark.com/mfLD/vu?pi=zFszsRA7ezW1H3z0&nodesktopflash=1.

3. (Recommended) Allow all jobs to finish.

   Note: If you do not allow all jobs to finish before you shut down the Controller, some jobs fail after the upgrade, and the failure may not be evident for some time. (See the `worker_inactive_delay` configuration parameter in the `<new_cs_controller_dir>/tomcat/webapps/cloud-ctrl/WEB-INF/classes/config.properties` file.)

4. Shut down the Controller.
5. Install the new Controller. (For information, see "Installing the CloudScan Controller" on page 19.)
6. If your existing `config.properties` file has been modified, you must merge it with the new `config.properties` file. (You cannot simply copy the existing `config.properties` file.)
7. Navigate to the `jobFiles` and `cloudCtrlDb` directories of the existing Controller, and then copy these to the new Controller.

   Note: To change these directories, edit the `config.properties` file.

8. Start the new Controller. (The database is automatically migrated.)

See Also

"About Upgrading Fortify CloudScan Components" on the previous page

"Upgrading the CloudScan Controller" on the previous page

"Upgrading Fortify CloudScan Sensors" below

Upgrading Fortify CloudScan Sensors

To upgrade your Fortify CloudScan sensors (on either Windows or Linux), you can either install the latest version of Fortify Static Code Analyzer, or unzip the `cloudscan.zip` file.

To upgrade sensors by installing or upgrading Fortify Static Code Analyzer:

1. Stop all sensors from running.
2. Go to one of the following Software Licenses and Downloads Portal sites:
   - https://entitlement.microfocus.com
   - https://entitlement.mfgs.microfocus.com (for US Government solutions)
3. Download the installer file for your operating system:
   - Windows: `Fortify_SCA_and_Apps_<version>_windows_x64.exe`
   - macOS: `Fortify_SCA_and_Apps_<version>_osx_x64.app.zip`
   - Linux: `Fortify_SCA_and_Apps_<version>_linux_x64.run`

- Solaris: `Fortify_SCA_<version>_solaris_x86.run` or `Fortify_SCA_<version>_solaris10_sparc.run`

Note: For detailed instructions on how to download Micro Focus Software, see https://www.brainshark.com/mfLD/vu?pi=zFszsRA7ezW1H3z0&nodesktopflash=1.

4. Install or upgrade Fortify Static Code Analyzer based on the instructions provided in the *Micro Focus Fortify Static Code Analyzer Software Version User Guide*.
5. Check the `<sca_install_dir>\Core\config` directory to make sure that the `worker.property` file exists.
6. Add the following property to the `worker.properties` file:
7. `worker_auth_token=<value_set_in_controller_configuration>`
8. Specify either a clear text password, or an encrypted shared secret (password the Controller uses to communicate with the sensor) as the `worker.properties` value. For information about how to generate an encrypted shared secret, see "Encrypting the Shared Secret on a Sensor" on page 24.
9. Save the `worker.properties` file.
10. Start the sensors.

See Also

"About Upgrading Fortify CloudScan Components" on page 42

"Upgrading the CloudScan Controller" on page 42

"Creating CloudScan Clients" on page 33

"Creating CloudScan Sensors" on page 35

Chapter 5: Managing Scan Requests

Scan requests are submitted from CloudScan clients. You can submit multiple scan requests, one after another, and the CloudScan sensors continues to run. If CloudScan is connected to a running Fortify Software Security Center server, you can do the following from the Scans view in Fortify Software Security Center:

- Cancel scan requests
- View and export scan request details

For details, see the *Micro Focus Fortify Software Security Center User Guide*.

This section contains the following topics:

Accessing Help for Command-Line Options

To access help for command-line options on a client or sensor, navigate to the `<sca_install_dir>bin`, and then run one of the following:

-h

-h start

-h worker

-h <*any_command_listed_with-help*>

For a complete list of all command-line options, see "CloudScan Command Options" on page 62.

Submitting Scan Requests

Depending on the language used to develop your source code, you can request a scan that offloads only the scanning phase of code analysis, or a scan that offloads both project translation and scanning to your CloudScan sensors.

Offloading Scanning Only

To submit a scan request that offloads only the scanning phase of code analysis, run the following command:

```
cloudscan.bat –url http://<controller_host>:8080/cloud-ctrl start -b <my_build_id> -
scan –Xmx2G
```

You can pass any relevant Fortify Static Code Analyzer scan tuning parameter (for example, `–Xmx` to specify the amount of memory for a scan) on the command line after the `–scan` keyword. If you use options such as `–build-label`, `-build-application`, or `–build-version`, make sure that you escape any quotes around the parameter. For example:

```
-scan –Xmx2G –build-label \"Application 5.4 – September 20, 2017\"
```

If the submission succeeds, you receive a token ID. The Fortify CloudScan sensor pulls the scan request from the Controller, processes it, and publishes the results to the Controller.

For information about the options to use for larger scans, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

Note: Jobs submitted (and FPRs) can be no larger than 1GB. Before you start large scans, review "Optimizing Scan Performance" on page 61.

Targeting a Specific Sensor Pool for a Scan Request

To target a specific sensor pool for a scan request, you must have:

- UUID for the sensor pool
- `pool_mapping_mode` property set to enabled or disabled

To get the UUID for the sensor pool:

1. Log on to Fortify Software Security Center.
2. On the Fortify header, select **SCANS**.
3. In the left panel, select **Sensor Pools**.

   The **Sensor Pools** table lists the existing sensor pools.
4. In the **Sensor Pools** table, copy the value shown in the **Pool UUID** column for the sensor pool you want to target for a scan request.

To specify a sensor pool to use for a scan request:

- From the command line on the client host, run the following:

  ```
  cloudscan.bat –url http://<controller_host>:8080/cloud-ctrl start -b
  <mybuildid> -pool <uuid> -scan
  ```

Offloading Both Translation and Scanning

If you use a supported language, you can offload both translation and scanning phases of code analysis to your CloudScan sensors. If your build tool is Apache Maven, Gradle, or MSBuild, include the `-bt` option.

Note: The -bt option is required for all technologies. For a projects without a build tool, -bt is set to none.

In the examples shown in the following table, CloudScan is integrated with Fortify Software Security Center, email is configured for CloudScan, and Fortify Software Security Center, the CloudScan Controller, and CloudScan sensors are up and running.

Objective	Command
Start a job to scan a Gradle project	`cloudscan.bat -url <controller_url> start -bt gradle`
Start a job to scan a Maven project with a non-default build file	`cloudscan.bat -url <controller_url> start -bt mvn -bf c:\myproj\myproj-pom.xml`
Start a job to scan a Gradle project, get email notifications from the CloudScan Controller, and upload the results to Fortify Software Security Center	`cloudscan.bat -url <controller_url> start -bt gradle -email username@domain.com -upload -uptoken <ssc_upload_token> -application "MyProject" -version "1.0"`

Translating Python Projects

Objective	Command
Start a job to scan a Python 2 project	`cloudscan.bat -url <controller_url> start -bt none –python-version 2 -python-requirements <path_to_requirements_file>`
Start a job to scan a Python project under an active virtual environment with dependencies already installed	`cloudscan.bat -url <controller_url> start -bt none`
Start a job to scan a Python project under an active virtual environment without project dependencies installed	`cloudscan.bat -url <controller_url> start -bt none --python-requirements <path_to_requirements_file>`

Objective	Command
Start a job to scan a Python project using an existing Python virtual environment and install project dependencies	`cloudscan.bat -url <controller_url> start -bt none --python-virtual-env <virtual_environment_location> --python-requirements <path_to_requirements_file>`

You can use CloudScan to work with Python in any of three ways. You can start CloudScan in a prepared virtual environment (see "Starting CloudScan in a Virtual Environment" below). You can use an existing virtual environment, without activating that virtual environment (see "Starting CloudScan in an Unactivated Virtual Environment" below). In this case, CloudScan activates the virtual environment itself. Finally, you can start the job outside of a virtual environment (see "Starting CloudScan Outside of a Virtual Environment" on the next page).

Starting CloudScan in a Virtual Environment

If you work in a virtual environment, all of your project dependencies are already installed. You do not need to invoke the pip package manager before you start CloudScan, or to specify the Python version (this is detected automatically).

To start CloudScan in a virtual environment:

1. Open a command line.
2. Activate the virtual environment.
3. Start CloudScan.

Example: `Ccloudscan.bat -url <controller_url> start -bt none`

If pip dependencies are not yet installed in the virtual environment used, CloudScan installs them automatically using the requirements file:

```
cloudscan.bat -url <controller_url> start -bt none --python-requirements <path_to_requirements_file>
```

Starting CloudScan in an Unactivated Virtual Environment

To start CloudScan in a virtual environment (with all dependencies installed) without activating that virtual environment:

1. Open a command line.
2. Start the Python project scan:

```
cloudscan -url <controller_url> start -bt none --python-virtual-env <venv_location>
```

or

```
cloudscan -url <controller_url> start -bt none --python-virtual-env <venv_location> --python-requirements <path_to_requirements_file>
```

CloudScan goes to the virtual environment, determines the Python version used, packages all required libraries, and then creates the package.

Starting CloudScan Outside of a Virtual Environment

If you plan to start CloudScan and there is no virtual environment on the client, you must have Python installed on the client, specify the Python version, and specify the Python requirements file. CloudScan locates the Python installation. In this case, CloudScan creates a temporary virtual environment, installs all dependencies from the requirements file, and then generates the package.

To start CloudScan outside of a virtual environment:

1. Open a command line.
2. Start CloudScan.
3. Run the following:

```
cloudscan –url <controller_url> start -bt none --python-requirements
<path> --python-version <version>
```

Translating Apex Projects

To perform remote translation of an APEX project, you must have Java 8 installed on your sensor. In addition, you must specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" that the CLS files are related to APEX, and not to Visual Basic 6.

To prepare for scanning, run the following:

```
cloudscan arguments -targs "-apex"
```

Note: For information on using the `-sargs` and `-targs` options, see the "Arguments Command" section in "Submitting Scan Requests" on page 45.

To scan the project using CloudScan, run the following:

```
cloudscan -url <controller_url> start -bt none
```

Alternatively, you can save the project package locally, as follows:

```
cloudscan package -o <path to package> -bt none
```

To send an existing package to CloudScan, run the following:

```
cloudscan -url <controller_url> start -package <package path>
```

CloudScan returns a job ID that you can use to track the scan.

Translating SQL Projects

To perform remote translation of a SQL project, you must have Java 8 installed on your sensor. In addition, you must specify an additional translation argument for the project so that Fortify Static Code

Analyzer "knows" what type of SQL (TSQL or PL/SQL) is required. (By default, on Windows, Fortify Static Code Analyzer uses TSQL, but on UNIX, it uses PL/SQL.)

To prepare a SQL project for scanning, run the following:

```
cloudscan arguments -targs "-sql-language <PL/SQL OR TSQL>"
```

Note: For information on using the `-sargs` and `-targs` options, see the "Arguments Command" section in "Submitting Scan Requests" on page 45.

To scan the project, run the following command:

```
cloudscan -url <controller_url> start -bt none
```

Alternatively, to save the package locally, run:

```
cloudscan package -o <path to package> -bt none
```

To send existing package to CloudScan, run:

```
cloudscan -url <controller_url> start -package <package path>
```

CloudScan returns a job ID that you can use to track the scan.

See Also

"CloudScan Command Options" on page 62

"Submitting Scan Requests" on page 45

"Submitting Scan Requests and Uploading Scan Results to a Fortify Software Security Center Application Version " on page 53

Viewing Scan Request Status

To view the status of a scan request, run the following command:

```
cloudscan.bat –url http://<Controller_Host>:8080/cloud-ctrl status -token <tokenid>
```

You can also view scan request status from the Fortify Software Security Center user interface. For instructions, see the *Micro Focus Fortify Software Security Center User Guide*.

If You use the Jenkins Master / Slave Architecture

If you use the Jenkins Master / Slave architecture and the Fortify Cloud Scan client (Jenkins Slave) performs the Fortify Static Code Analyzer translation, the Jenkins Slave must know whether the Fortify Static Code Analyzer scan is completed and the FPR is correctly uploaded to Fortify Software Security Center.

Canceling a Scan Request

To cancel a scan request, run the following command:

```
cloudscan.bat –url http://<controller_host>:8080/cloud-ctrl cancel -token <tokenid>
```

You can also cancel scan requests from the Scans view in Fortify Software Security Center. For instructions, see the *Micro Focus Fortify Software Security Center User Guide*.

Retrieving Scan Results from the CloudScan Controller

To retrieve scan results, run the following command:

```
cloudscan.bat -url http://<controller_host>:8080/cloud-ctrl retrieve -token <tokenid>
-f worker.fpr -log worker.log
```

Viewing Client and Sensor Logs

To view the CloudScan client and sensor logs on a Windows system:

- Navigate to `%FORTIFY_HOME%\cloudscan\log`, where `%FORTIFY_HOME% is ${win32.LocalAppdata}\Fortify`.

  On Windows 7, for example, the location is `C:\Users\<user>\AppData\Local\Fortify`.

  If you have separate installs, the log is located at: `<cs_client_dir>\bin\Fortify\log\cloudscan.log`

To view the CloudScan client and sensor logs on a Linux system, navigate to the following directories:

- To retrieve the CloudScan log, navigate to `~/.fortify/cloudscan/log/cloudscan.log`.
- To retrieve the CloudScan Controller log, navigate to `<cs_controller_dir>\tomcat\logs\cloudCtrl.log` on Windows and to `<cs_controller_dir>/tomcat/logs/cloudCtrl.log`on Linux.
- To retrieve the Fortify Software Security Center log, navigate to `<fortify.home>/<app_context>/logs`.

Chapter 6: Working with Fortify CloudScan from Fortify Software Security Center

While you can deploy the Controller in standalone mode, communication with Fortify Software Security Center provides additional benefits. If Fortify Software Security Center is integrated with Fortify CloudScan, then the Fortify Software Security Center Scans view includes the CloudScan pages, which are described in the following table.

Scans View Page	Functionality
Scan Requests	View and export Fortify CloudScan scan request details Cancel prepared scan requests
Controller	View Controller information
Sensors	View sensor information
Sensor Pools	Create and manage groups of sensors to which you can target scan requests.

For detailed information, see the *Micro Focus Fortify Software Security Center User Guide*.

See Also

"Configuring the Connection to Fortify Software Security Center" below

Configuring the Connection to Fortify Software Security Center

While the CloudScan Controller can be deployed in standalone mode, communication with Fortify Software Security Center provides additional benefits:

- The Fortify Software Security Center user interface includes a Scans view that makes it easy to view the status of recent scan requests.
- The CloudScan Controller can upload scan results directly to Fortify Software Security Center application versions.
- You can create and manage CloudScan sensor pools from Fortify Software Security Center. (For information about sensor pools, see the *Micro Focus Fortify Software Security Center User Guide*.)

Note: You must use the same or a later version of Fortify Software Security Center as the Fortify Static Code Analyzer version installed on your CloudScan clients.

To integrate Fortify Software Security Center and Fortify CloudScan:

1. Log in to Fortify Software Security Center as an administrator, and then, on the Fortify header, click **ADMINISTRATION**.
2. In the left panel, select **Configuration**, and then select **CloudScan**.

   The CloudScan page opens.
3. To enable the polling of CloudScan Controller to retrieve scan request status, select the **Enable CloudScan** check box.
4. In the **CloudScan Controller URL** box, type the URL for the CloudScan Controller.
5. In the **CloudScan poll period (seconds)** box, either select or type the number of seconds to elapse between CloudScan polls.
6. In the **SSC and CloudScan Controller shared secret** box, type the password for Fortify Software Security Center to use when it requests data from the CloudScan Controller. (If you use clear text, this string must match the value stored in the CloudScan Controller `config.properties` file for the `ssc_cloudctrl_secret` key.)
7. Click **SAVE**.
8. Restart the Fortify Software Security Center server.

Important! You must use the same or a later version of Fortify Software Security Center as the Fortify Static Code Analyzer version installed on your CloudScan clients.

See Also

"Working with Fortify CloudScan from Fortify Software Security Center" on the previous page

"Starting the Fortify CloudScan Components" on page 40

Submitting Scan Requests and Uploading Scan Results to a Fortify Software Security Center Application Version

To submit a scan request, the results of which you want to upload to an application version in Fortify Software Security Center, you can obtain the application version ID and access tokens from Fortify Software Security Center. Use the fortifyclient tool to obtain these items. You can reuse the token with future requests. For more information about the fortifyclient tool, see the *Micro Focus Fortify Software Security Center User Guide*.

Note: The Fortify Software Security Center user account must have permission to upload scan results for the application version. A user who submits a Fortify CloudScan job for upload to a Fortify Software Security Center application version must use a token that was obtained using an account that has permission to upload scan results. If a Fortify Software Security Center user is assigned to a target application version with a view-only role, and that user requests a token and uses it to submit the job, the upload fails.

To submit a job to be uploaded to an application version:

1. Open a command prompt, and then type the following command:

```
fortifyclient.bat  listApplicationVersions -url http://<ssc_host>:8180/ssc -user
<user> -password <pwd>
```

Sample Output

```
ID      Name                    Version

10     CloudScan   Test         1.0

12     CloudScan   Test         2.0

4      Bill Payment Processor   1.1

3      Logistics                2.5

2      Logistics                1.3

8      RWI                      2.0

5      RWI                      1.0
```

2. To generate a CloudScan Controller token, run the following command.

```
fortifyclient.bat token -gettoken CloudCtrlToken -url http://<ssc_host>:8180/ssc
-user <user> -password <pwd>

Authorization Token: <..cloudCtrlToken...>
```

3. To submit your job and upload your scan results to a Fortify Software Security Center application version, run one of the following commands:

```
cloudscan.bat -sscurl http://<ssc_host>:8180/ssc -ssctoken <CloudCtrlToken> start
-upload -versionid 10 -b <mybuildId> -uptoken <cloudCtrlToken> -scan –Xmx2G
```

Note: Instead of `–versionid <version id>`, you can pass `––application <application_name> --application-version <version_name>`. The `<application_name>` and `<version_name>` must match the values in Fortify Software Security Center. These values are case sensitive.

Typically, the steps above are combined into a scripted flow from a build server.

Appendix A: Configuring Sensor Auto-Start

The following procedures are designed to provide general guidance to enable sensor auto-start and may not be appropriate in all environments. Fortify strongly recommends that you review the instructions with your system administrator and make any changes required for your environment.

This section contains the following topics:

Enabling CloudScan Sensor Auto-Start on Windows as a Service

Check to make sure the Controller is running before you perform the following procedure.

To enable sensor auto-start on Windows as a service:

1. Log in to the sensor machine as a local admin user.

   Note: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of Fortify CloudScan; they are not shared with any other service. To avoid issues associated with insufficient privileges, use a fully-privileged administrative account for the auto-start setup.

2. Open a command prompt and navigate to the `<sca_install_dir>\bin\cloudscan-worker-service` directory.
3. Run the `setupworkerservice.bat` script with no arguments to see the usage help.
4. Re-run the batch script with the required arguments included.
5. Open Windows Services and check to make sure that the sensor service is present.
6. Right-click the listed sensor service, and then select **Start**.
7. Fortify recommends that you change the startup type setting to **Manual** until you verify that the sensor runs successfully. After verification, change the startup type setting to **Automatic (Delayed Start)** in Windows Services.
8. Check to make sure that the sensor communicates with the Controller.

See Also

"Creating a CloudScan Sensor as a Service" on page 36

Troubleshooting

Review the following logs to troubleshoot issues encountered during the configuration of sensor auto-start as a Windows service:

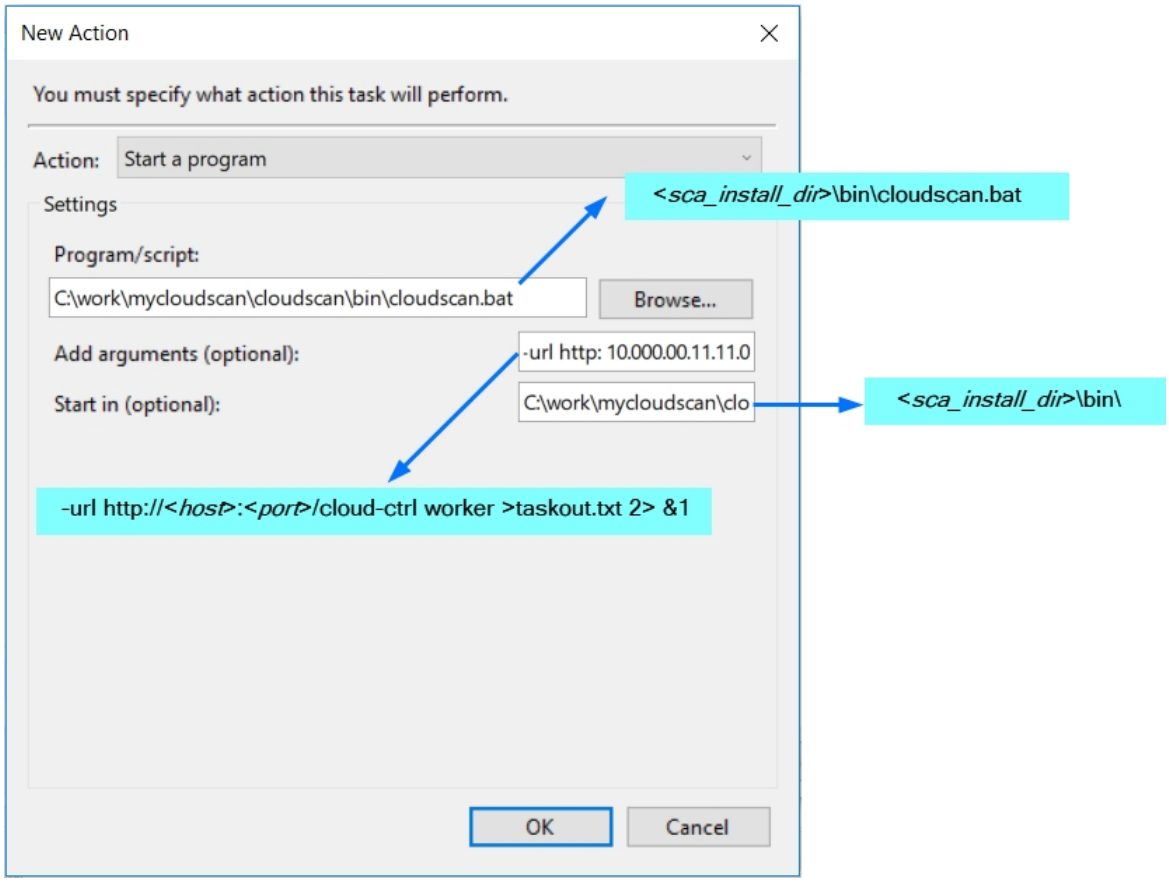
- Main CloudScan sensor log:
  `C:\Windows\System32\config\systemprofile\AppData\Local\Fortify\cloudscan\cloudscan.log`
- Sensor temporary folders that contain MBS files, Fortify Static Code Analyzer log files, and generated FPR files: `c:\CloudscanWorkdir\<job_token>`
- Sensor `stdOut` and `stdErr` logs: `c:\CloudscanWorkdir\workerout.log` and `c:\CloudscanWorkdir\workererr.log`

  Note: Before you start a sensor, check to make sure that the log files are not open in an application. Open log files prevent procrun from writing to the file.
- Commons-daemon log: `c:\CloudscanWorkdir\<year_month_day>.log`

Enabling CloudScan Sensor Auto-Start on Windows as a Scheduled Task

1. Log on to the sensor machine as the local admin user.

   Note: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of Fortify CloudScan; they are not shared with any other service. To avoid issues related to insufficient privileges, use a fully-privileged administrator account for the auto-start setup.
2. Start the Task Scheduler.
3. In the **Actions** panel, select **Create Task**.
   The Create Task window opens.
4. On the **General** tab, provide the following information:
   a. In the **Name** box, type a name for the task.
   b. Select the **Run whether user is logged on or not** option.
5. Select the **Actions** tab, and then click **New**.
   The New Action dialog box opens.

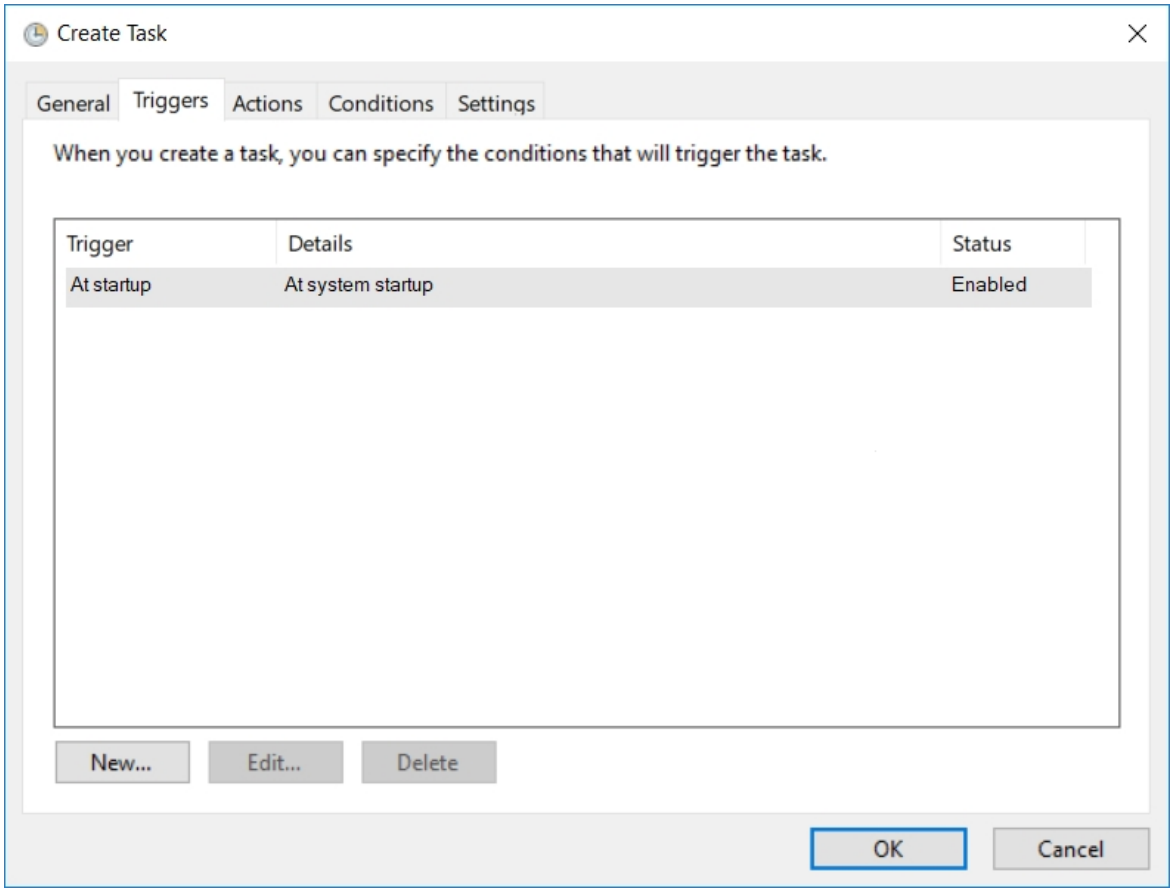


a. From the **Action** list, select a program to start.
b. In the **Program/script** box, type the directory path to your `cloudscan.bat` file.
   Example: *<sca_install_dir>*`\bin\cloudscan.bat`
c. In the **Add arguments (optional)** box, type the following:

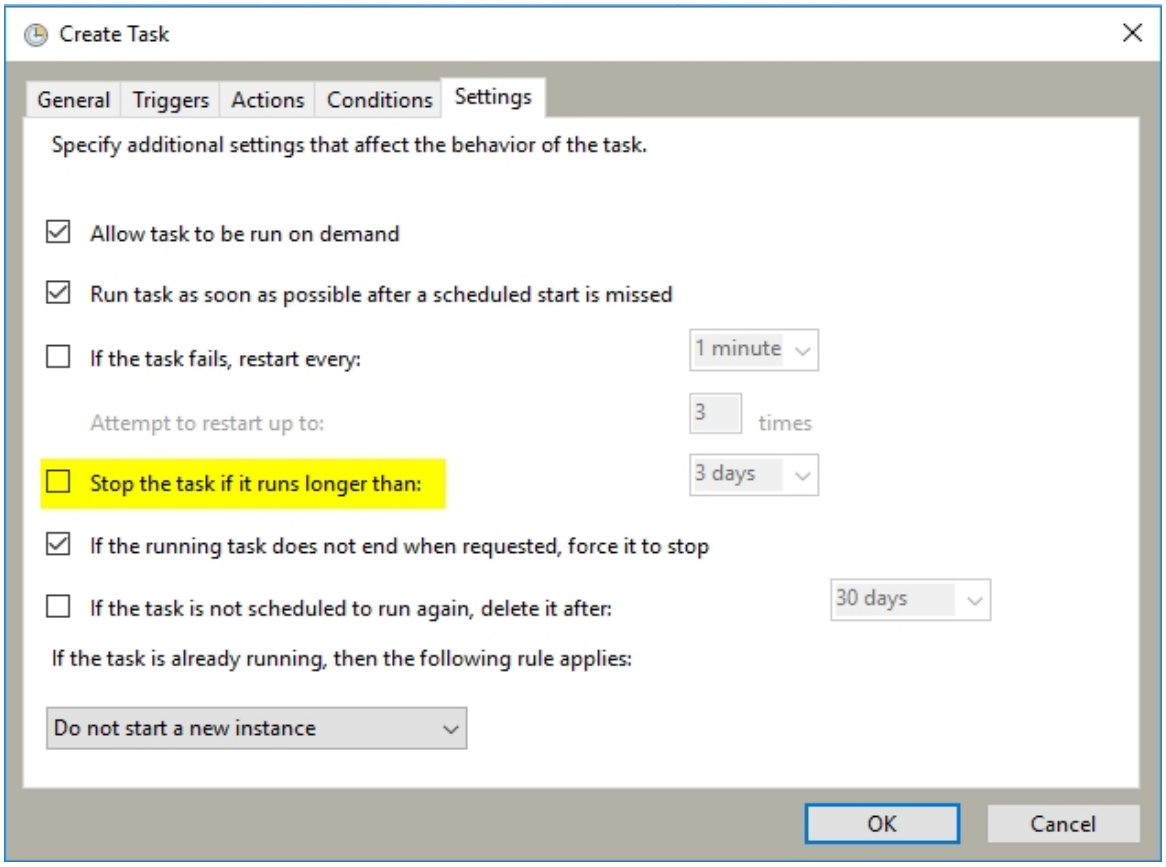
   `-url http://`*<host>*`:`*<port>*`/cloud-ctrl worker >taskout.txt 2>&1`

d. In the **Start in (optional)** box, type the path to the CloudScan sensor `bin` directory.
   Example: *<sca_install_dir>*`\bin\`
e. Click **OK**.

6. Return to the Task Scheduler and select the **Triggers** tab.



7. Check to make sure that the **At startup trigger** is enabled, and then click **OK**.
8. Select the **Settings** tab.



9. Make sure the **Stop the task if it runs longer than** check box is cleared, and then click **OK**.
10. Click **Save**.
11. Restart the machine.

The script output in the `taskout.txt` file indicates whether the CloudScan sensor started successfully.

You can also start and stop the scheduled task manually from the Task Scheduler interface when logged into the machine.

Enabling CloudScan Sensor Auto-Start on a Linux System

Note: The following procedure has been tested with Red Hat; there may be some variation for other Linux varieties. Please review these steps with your system administrator before you make any changes.

1. Log in to the machine as "root."
2. Run the `visudo` command to edit the `sudoers` file and disable requiretty.

```
Defaults   !requiretty
```

Note: You can also disable requiretty per user.

3. Set auto-start, as follows:
   a. Verify the command invocation from the console (modify according to your install directory).

      ```
      sudo -u <username> -- <sca_install_dir>/bin/cloudscan -url
      http://<host>:8080/cloud-ctrl worker > <sca_install_
      dir>/bin/workerout.txt 2>&1  &
      ```

      - Add the `sudo` command to the end of the file (add it before the line `exit 0` if it exists).
      - The ampersand (`&`) at the end enables the machine to boot up even if sensor startup fails or hangs.
      - The double-dash (`--`) is important to separate the options for sudo from the options for your service.

   b. Make the change to the startup file.

      Caution! Make sure that you do not change anything else in your bootup script.

      ```
      vi /etc/rc.d/rc.local
      ```

4. Check the setup:
   a. Reboot and log in to the machine as "root."
   b. To verify the processes under root, type:

      ```
      ps -x | grep java
      ```

   c. Verify that the output shows that the sensor is not started under root.
   d. To verify the processes under the user, type:

      ```
      sudo -u <username> ps x | grep java
      ```

   e. Verify that the output displays the sensor process.
   f. To verify the existence and contents of the script output file, type:

      ```
      tail -f/opt/<sca_install_dir>/bin/workerout.txt
      ```

      Example: `tail -f/Fortify/Fortify_SCA_and_Apps_<version>/bin/workerout.txt`

Appendix B: Optimizing Scan Performance

If you plan to regularly scan large applications, Fortify recommends that you run a manual test scan on hardware that is equivalent to the hardware on which your sensor is installed.

To optimize your scan:

1. To set the Fortify Static Code Analyzer scan parameters for optimal performance, adjust the memory settings to align with your hardware.

   For information about how to tune Fortify Static Code Analyzer, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

2. Run the scan.
3. Note the size of the resulting FPR file and scan log. To ensure that the CloudScan Controller and Fortify Software Security Center can accept FPR or log files larger than 1 GB, increase the following file size threshold:

   - Navigate to the `<cs_install_dir>\tomcat\webapps\cloud-ctrl` directory on Windows (`<cs_install_dir>/tomcat/webapps/cloud-ctrl` on Linux) , open the `config.properties` file, and then set the Controller threshold as follows:

     ```
     max_upload_size=<max_fpr_or_logfile_size_in_MB>
     ```

     The default value is 1024.

4. Check to make sure that your Fortify Software Security Center hardware and application startup parameters are set to process very large FPR files. For more information, see the *Micro Focus Fortify Static Code Analyzer User Guide*.

Appendix C: CloudScan Command Options

This appendix provides information about the command-line arguments that you can use with Fortify CloudScan.

Global Options

This section provides information about the command-line arguments that you can use with Fortify CloudScan.

Global Option	Use to:
-h <*command*>	Get help for the selected command. To see all command help, type `-h all`.
-ssctoken <*token*>	Specify the Fortify Software Security Center cloud authorization token.
-sscurl <*url*>	Specify the Fortify Software Security Center server URL.
-url <*url*>	Specify the CloudScan controller URL.
-version	Get the product version.

Status Command

Use the `status` command to check the status of the Controller or a job.

Option	Description
-ctrl	Verify that the Controller is running.
-token, --job-token <*token*>	Specify the job token to query.

Start Command

Use the `start` command to start a remote scan.

Option	Description
-application, --application <*name*>	Specify the Fortify Software Security Center application name.
-b, --build-id <*id*>	Specify the build ID of the session to export.
-bf, --build-file <*file*>	Specify the build file, unless it has a default name such as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
-block	Wait for the job to complete, and then download the result.
-bt, --build-tool <*name*>	Specify build tool name used for the project. You cannot use this option with the `-scan` option.
-email <*address*>	Specify the email address for job status notifications.
-f, --output-file <*file*>	Specify the name for the local FPR file output.
-filter <*file*>	Specify the filter file to use during a scan (repeatable).
-hv, --php-version <*version*>	Specify the PHP version.
-log, --log-file <*file*>	Specify the name for the local log file output.
-mbs <*file*>	Specify the mobile build session to upload.
-o, --overwrite	Overwrite the existing FPR or log with new data.
-p, --package <*file*>	Specify the project package file to upload.
-pool, --submit-to-pool <*uuid*>	Specify the sensor pool to which to submit the job.
-project, --project-name <*name*>	Specify the Fortify Software Security Center application name DEPRECATED: use short `-application` or long `--application`.
-projroot, --project-root <*dir*>	Specify the project directory for the mobile build

Option	Description
	session export.
-projtl, --project-template <*file*>	Specify the project template file to include.
-pyr, --python-requirements <*file*>	Specify the Python project requirements file to install and collect dependencies.
-pyv, --python-virtual-env <*directory*>	Specify the Python virtual environment location.
-q, --quiet	Prevent the printing of stdout from the build execution.
-rules <*file/dir*>	Specify custom rules file or directory to use during the scan (repeatable).
-scan	Set the point beyond which all arguments are for sourceanalyzer. You cannot use this option with the `--build-tool` or `--package option`.
-sp, --save-package <*file*>	Specify the package file to save after uploading. The file extension must be `*.zip`.
-t, --include-test	Include test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
-upload, --upload-to-ssc	Upload the FPR to Fortify Software Security Center upon completion.
-uptoken, --ssc-upload token <*token*>	Specify the Fortify Software Security Center file upload token.
-version, --application-version <*name*>	Specify the Fortify Software Security Center application version name.
-versionid, --project-version-id <*id*>	Specify the Fortify Software Security Center application version ID DEPRECATED long option: use long `--application-version-id`.
-versionname, --project-version-name <*name*>	Specify the Fortify Software Security Center application version name DEPRECATED: use short `-version` or long `--application-version`.
-yv, --python-version <*version*>	Specify the Python version to automatically find the

Option	Description
	installed Python. Allowed values: 2 or 3. This flag is ignored if the CloudScan client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Retrieve Command

Use the `retrieve` command to download the result of a remote scan job.

Option	Description
-block	Wait for the job to complete and download the result.
-f, --output-file <*file*>	Specify the file name for local FPR output.
-log, --log-file <*file*>	Specify the file name for local log output.
-o, --overwrite	Overwrite the existing FPR or log with new data.
-token, --job-token <*token*>	Specify the job token to query.

Cancel Command

Use the `cancel` command to cancel a remote scan job.

Option	Description
-token, --job-token <*token*>	Specify the job token to query.

Worker Command

Use the `worker` command to start or test a CloudScan sensor.

Option	Description
-hello	Sensor reporting for duty.

Package Command

Use the `package` command to create a zip package of the specified project.

Option	Description
-bf, --build-file <*file*>	Specify the build file if you are not using a default name such as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
-bt, --build-tool <*name*>	Specify the build tool name used for the project. You cannot use this option with the `-scan` option.
-hv, --php-version <*version*>	Specify the PHP version.
-o, --output <*file*>	Specify the output file name. The file extension must be `*.zip`.
-pyr, --python-requirements <*file*>	Specify the Python project requirements file to install and collect dependencies.
-pyv, --python-virtual-env <*directory*>	Specify the Python virtual environment location.
-q, --quiet	Prevent the printing of stdout from the build execution.
-t, --include-test	Include the test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
-yv, --python-version <*version*>	Specify the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the CloudScan client is started under a Python virtual environment or if `-python-virtual-env` is specified.

Arguments Command

Use the `arguments` command to generate a settings file for additional Fortify Static Code Analyzer command-line options.

Option	Description
-o, --overwrite	Overwrite the existing arguments file.
-p, --project-dir <*directory*>	Specify the project directory in which to create the Fortify Static Code Analyzer translation and scan additional arguments file.
-sargs, --scan-args	Fortify Static Code Analyzerscan arguments (repeatable)
-targs, --translation-args	Fortify Static Code Analyzer translation arguments (repeatable)

Important! The `-targs` and `-sargs` options take a single string argument. To specify multiple translation or scan arguments, use multiple `-targs` and (or) `-sargs` options. If the translation or scan option has a path parameter that includes a space, enclose the path in single quotes.

Example: The following generates a `fortify-sca.settings` file in the current directory.

```
cloudscan.bat arguments -o -targs "-Xmx4G" -targs "-cp 'myProject Dir/path
to/lib/*.jar'" -targs "-exclude 'myProject Dir/path to/src/*.js'" -sargs
"-Xms256M" -sargs "-analyzers controlflow,dataflow"
```

The resulting `fortify-sca.settings` file looks similar to the following:

```
{
  "translationArgs": [
  "-Xmx4G",
  "-cp",
  "myProject Dir/path to/lib/*.jar",
  "-exclude",
  "myProject Dir/path to/src/*.jar"
 ],
  "scanArgs": [
  "-Xms256M",
  "-analyzers",
  "controlflow,dataflow"
 ]
}
```

Progress Command

Use the `progress` command to get the progress of a Fortify Static Code Analyzer scan.

Important! If your projects are based on Java 11, and you want to use the progress command to check the progress of your scans, some minor sensor configuration is required. For instructions, see "Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 37.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Installation, Configuration, and Usage Guide (Fortify CloudScan 19.2.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to FortifyDocTeam@microfocus.com.

We appreciate your feedback!