Micro Focus
Fortify ScanCentral SAST

Software Version: 21.1.0

Installation, Configuration, and Usage Guide

Document Release Date: Revision 1 - September 14, 2021
Software Release Date: July 2021

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

All trademarks, service marks, product names, and logos included in this document are the property of their respective owners.

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 June 11, 2022. 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/documentation

Contents

Preface

Contacting Micro Focus Fortify Customer Support

Visit the Support website to:

- Manage licenses and entitlements
- Create and manage technical assistance requests
- Browse documentation and knowledge articles
- Download software
- Explore the Community

https://www.microfocus.com/support

For More Information

For more information about Fortify software products:

https://www.microfocus.com/cyberres/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/documentation

To be notified of documentation updates between releases, subscribe to Fortify Product Announcements on the Micro Focus Community:

https://community.microfocus.com/cyberres/fortify/w/fortify-product-announcements

Fortify Product Feature Videos

You can find videos that highlight Fortify products and features on the Fortify Unplugged YouTube channel:

https://www.youtube.com/c/FortifyUnplugged

Change Log

The following table lists changes made to this document.

Software Release / Document Version	Changes
21.1.0 / Revision 2 - January 26, 2022	The string `-Xmx#G` is normally used to specify the amount of memory to use for a scan, but does not work in this release. As a result, the string was removed from commands described in "Submitting Scan Requests" on page 57 and in "Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 70. **Note:** The commands that included the `-Xmx#G` string donot work in this release. Without it, the commands work as intended.
21.1.0 / Revision 1 - September 14, 2021	**Modified Topic:** "About Upgrading ScanCentral SAST Components" on page 52 now includes information about Fortify ScanCentral SAST and Fortify Software Security Center version compatibility.
21.1.0	**New topics** • "What's New in Micro Focus ScanCentral SAST 21.1.0" on page 17 • "Generating a ScanCentral SAST Package" on page 63 • "Configuring Job Cleanup Timing on Sensors" on page 66 • "Configuring the Logging Level on the Controller " on page 31 **Modified topics** • A description of the *Micro Focus Fortify Software Security Center with Elasticsearch Deployment Guide* was added to "Related Documents" on page 13. • A note and minor changes were added to "Fortify ScanCentral SAST Components" on page 18. • The table of properties in "Configuring the ScanCentral SAST Controller" on page 21 now includes several new properties. • A note about the use of the caret character (^) in worker_auth_token

Software Release / Document Version	Changes
	tokens was added to "Creating a ScanCentral SAST Sensor as a Service" on page 42. • Removed an unnecessary step (create a `worker.properties` file) from "Creating ScanCentral SAST Sensors" on page 41. • A note about the .NET framework version installed on sensor machines was added to "(Windows only) Configuring Sensors to Offload Translation for .NET Languages" on page 45. • "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 55 contains additional details about upgrade paths for clients and sensors. • New example commands were added to "Submitting Scan Requests" on page 57. • The `-debug` global option was added to "Fortify ScanCentral SAST Command Options" on page 79. **Removed topics** • What's New in Micro Focus ScanCentral SAST 20.2.0 • Installing and Configuring the Fortify ScanCentral SAST Components
20.2.0	**General** • The product name was changed from Fortify ScanCentral to Fortify ScanCentral SAST to distinguish it from the newly introduced Fortify ScanCentral DAST. **New topics** • What's New in Micro Focus ScanCentral SAST 20.2.0 • "Changing Sensor Expiration Time" on page 43 • "Support for Multiple Fortify Static Code Analyzer Versions" on page 43 • "Viewing Scan Request Status" on page 66 • "Viewing Client and Sensor Logs" on page 66 **Modified topics** • A note that describes the difference between standalone and

Software Release / Document Version	Changes
	embedded clients was added to "Fortify ScanCentral SAST Components" on page 18. • The section "Updating a Sensor Based on a Fortify Static Code Analyzer Version Earlier than 20.2.0" was removed from "Creating ScanCentral SAST Sensors" on page 41. • In "Configuring the ScanCentral SAST Controller" on page 21 `ssc_cloudctrl_secret` was replaced with `ssc_scancentral_ctrl_secret` and the `fail_job_if_uptoken_invalid` option was added. • In "Encrypting the Shared Secret" on page 27 `ssc_cloudctrl_secret` was replaced with `ssc_scancentral_ctrl_secret`. • The procedure used to start the Controller was modified in "Starting the ScanCentral SAST Components" on page 48. • A note about the compatibility rules that apply to remote translation was added to "Upgrading the ScanCentral SAST Controller" on page 52. • "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 55 was changed to reflect the change in the default value for the `client_auto_update` property from `true` to `false`. • Command-line options used with the packagecanner tool were changed in "Using the PackageScanner Tool" on page 64. • Various changes were made to command-line options in "Fortify ScanCentral SAST Command Options" on page 79. **Removed topics** • What's New in Micro Focus Fortify ScanCentral 20.1.0 • Accessing Help for Command-Line Options
20.1.0	All references to CloudScan were replaced with ScanCentral. **New topics** • What's New in Micro Focus Fortify ScanCentral 20.1.0 • "Securing the Controller for Authorized Client Use Only" on page 38 • "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 55

Software Release / Document Version	Changes
	• "Using the PackageScanner Tool" on page 64 **Modified topics** • "Configuring the ScanCentral SAST Controller" on page 21 • "(Windows only) Configuring Sensors to Offload Translation for .NET Languages" on page 45
19.2.0	**New topics** • "Configuring Sensors to Use the Progress Command when Starting on Java" on page 44 • Configuring Sensors to Use the Progress Command when Starting on Java 11 • "(Windows only) Configuring Sensors to Offload Translation for .NET Languages" on page 45 **Modified topics** • "Installing the CloudScan Controller" was modified to reflect the new installation procedure used for installation on both Linux and Windows. • "Creating CloudScan Clients" was modified to reflect the introduction of the `CloudScan_Client_<version>.zip` file, which is used to create standalone clients that support translation on CloudScan sensors. • "Upgrading the CloudScan Controller" was modified to reflect file name changes. • The procedure described in "Upgrading Fortify CloudScan Sensors" 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 57. • New argument command options were added to Appendix C: CloudScan Command Options.

Software Release / Document Version	Changes
	Removed topics Installing the CloudScan Controller on a Linux System Installing the CloudScan Controller on a Windows System

Chapter 1: Introduction

With Fortify ScanCentral SAST (ScanCentral), 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 ScanCentral 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 ScanCentral client then hands off the MBS to the ScanCentral Controller, which distributes the MBS to the 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 sensors. For information about the languages supported for offloading translation, see "Creating ScanCentral SAST Clients" on page 39. 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, ScanCentral 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, the ScanCentral Controller can direct the output FPR to Fortify Software Security Center.

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

Intended Audience

This content is written for anyone who intends to install, configure, or use ScanCentral to offload the translation (for supported languages) and scanning phases of the Fortify Static Code Analyzer process to ScanCentral 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/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 Software Security Center

The following document provides 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	This document provides Fortify Software Security Center users with detailed information about how to

Document / File Name	Description
SSC_Guide_<*version*>.pdf	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 SCA_Cust_Rules_Guide_<*version*>.zip	This document provides the information that you need to 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	This document provides information about how to

Document / File Name	Description
Eclipse User Guide Eclipse_Plugins_Guide_<*version*>.pdf	install and use the Fortify Complete and the Fortify Remediation Plugins for Eclipse.
Micro Focus Fortify Plugins for JetBrains IDEs and Android Studio User Guide JetBrains_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 other JetBrains IDEs.
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 Guide SCA_Tools_Props_Ref_<*version*>.pdf	This document describes the properties used by Fortify Static Code Analyzer tools.

What's New in Micro Focus ScanCentral SAST 21.1.0

Micro Focus ScanCentral SAST 21.1.0 includes the changes described here.

Improved Accuracy of Job Processing Messages

Previously, when a job was assigned to a sensor, the Controller sent the email message "ScanCentral job request accepted." After the job was completed, the Controller sent the email message "ScanCentral job completed."

Now, when the Controller accepts a job, it sends the email message "ScanCentral job request accepted." After the job is assigned to a sensor, the Controller sends the email message "ScanCentral job request assigned." Finally, after the job is completed, the Controller sends the email message "ScanCentral job completed."

New -debug Option

The `-debug` option, which enables debug logging on clients and sensors, was added in this release. (See "Fortify ScanCentral SAST Command Options" on page 79.)

-upload Option Required for Scans When Fortify Software Security Center is in Lockdown Mode

Previously, if Fortify Software Security Center was in lockdown mode, you could run a scan, even if you failed to specify the `-upload` option in the ScanCentral SAST command. The results shown for the scan on the **SCANCENTRAL** > **SAST** tab in Fortify Software Security Center left out the application version and the scan was not uploaded. Now, if Fortify Software Security Center is in lockdown mode, and you try to start a scan without using the `-upload` option, client execution fails with an error.

Note too, that if `ssc_lockdown_mode` is enabled, then the value set for `pool_mapping_mode` in the `config.properties` file is ignored and `pool_mapping_mode` is automatically set to `ENFORCED`. For information about the `pool_mapping_mode` property, see "About the pool_mapping_mode Property" on page 29.

New Properties Enable Control Over the Domains the Controller Uses to Send email Notifications

Two new properties in the `config.properties` file enable you to specify which outgoing email domains are allowed for outgoing emails and which domains are disallowed. For information about how to use the new `email_allow_list` and `email_deny_list` properties, see "Configuring the ScanCentral SAST Controller" on page 21.

Chapter 2: Fortify ScanCentral SAST Components

A Fortify ScanCentral SAST deployment includes the following three components:

- **ScanCentral Controller**: A standalone web application that receives the Fortify Static Code Analyzer mobile build sessions and scan instructions from ScanCentral SAST clients (or project packages with translation and scan instructions), routes the information to sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center.

- **ScanCentral client**: 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 ScanCentral Controller.

  Note: A client can be either an *embedded* client, which resides on the same machine as Fortify Static Code Analyzer, or a *standalone* client, which is independent of Fortify Static Code Analyzer.

  Within an SCA and Apps installation, the files used to create ScanCentral SAST sensors and embedded clients are the same. The only difference is how you invoke their functionality from the command line. To use ScanCentral SAST as a sensor, you run ScanCentral SAST using the `worker` command. To use ScanCentral SAST as a client to initiate a scan, you invoke it using the `start` command. Sensor functionality depends on Fortify Static Code Analyzer. So, you can have a standalone client, but not a standalone sensor.

  The interface for issuing Fortify ScanCentral SAST 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 ScanCentral Controller.

  Note: A standalone 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 translation and scanning.

- **ScanCentral 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 analyses. For information about the languages supported for performing translation, see "Creating ScanCentral SAST Clients" on page 39.

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

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

To successfully deploy Fortify ScanCentral SAST, in addition to installing Fortify Static Code Analyzer, you must complete the following tasks in the order you see listed here:

- Deploy (or connect to an existing) a Fortify Software Security Center instance
- Install the Fortify ScanCentral SAST Controller
- Create ScanCentral SAST sensors

Instructions for completing these tasks are provided in the following sections. For information about hardware and software requirements for these components, see the *Micro Focus Fortify Software System Requirements* document.

Installing the Controller

The ScanCentral SAST Controller (Controller) is a standalone server that sits between the ScanCentral SAST 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 a 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. (Controller cleanup removes the job directory, removes jobs from the database, and removes information about expired sensors from the database so that they are no longer displayed in Fortify Software Security Center.) You can find the `config.properties` file in the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory.

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

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

- Extract the contents of the `Fortify_ScanCentral_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, *<controller_dir>* refers to the ScanCentral 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 ScanCentral Controller, *<controller_dir>* resembles the following:

```
bin/

tomcat/

readme.txt
```

See Next

"Configuring the ScanCentral SAST Controller" on the next page

For information about how to update your Controller, see "About Upgrading ScanCentral SAST Components" on page 52 and "Upgrading the ScanCentral SAST Controller" on page 52.

See Also

"Installing the Controller as a Service" below

"Starting the ScanCentral SAST Components" on page 48

Installing the Controller as a Service

To install the 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 ScanCentral SAST `Tomcat` directory.
4. Navigate to the *<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 ScanCentral SAST `Tomcat` directory.

3. Navigate to the *<controller_dir>*/tomcat/bin directory, and then run the following:

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

The service name must not contain any spaces.

"Configuring the ScanCentral SAST Controller" below

Uninstalling the Controller Service

To uninstall the Apache Tomcat 9.0 service:

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

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

See Also

"Installing the Controller as a Service" on the previous page

"Configuring the ScanCentral SAST Controller" below

Configuring the ScanCentral SAST Controller

After you install the 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 ScanCentral 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 Controller:

1. Navigate to *<controller_dir>*/tomcat/webapps/scancentral-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
`pwtool_keys_file`	Path to a file with pwtool keys. If encoded passwords are used, this must point to a file with the pwtool keys used to encode the passwords. Otherwise you can comment it out. `pwtool_keys_file=${catalina.base}/pwtool.keys`
`db_dir`	ScanCentral SAST database home directory db_dir=${catalina.base}/cloudCtrlDb
`worker_auth_token`	A string that contains no spaces or backslashes used to secure the Controller for use by authorized sensors only. 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, see "Encrypting the Shared Secret on the Controller" on page 27.
`client_auth_token`	A string that contains no spaces or backslashes, used to secure the Controller for use by authorized clients only. 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, see "Encrypting the Shared Secret on a Client" on page 29.
`allow_insecure_clients_with_empty_token`	Used to support older clients for which the `client_auth_token` cannot be set. You can allow connections from clients that have no token set. Connections from clients with configured but incorrect secret keys are rejected. This property is available for old (v1) endpoints only. For details, see "Securing the Controller for Authorized Client Use Only" on page 38.
`job_file_dir`	Job storage directory.
`max_upload_size`	Maximum size (MB) of files that can be uploaded to the Controller from clients or sensors (for example, log files, result files, job files).
`smtp_host`	SMTP server host name.
`smtp_port`	SMTP server port number.
`smtp_ssl`	If set to `true`, the Controller uses SSL for connections to the SMTP

Option	Description
	server. Otherwise, it does not use SSL (default).
`smtp_ssl_check_trust`	If set to `false`, the SMTP serv er certificate is always trusted. Otherwise, the certificate trust is based on the certification path (the default).
`smtp_ssl_check_server_identity`	If set to `false`, STMP server identity is not checked. Otherwise, the Controller checks server identity, as specified by RFC 2595 (the default).
`smtp_auth_user` / `smtp_auth_pass`	If your SMTP server requires authentication, uncomment both the `smtp_auth_user` and `smtp_auth_pass` properties and set their values. Otherwise, leave both properties commented. You can use either a plain text password or a password encoded using pwtool for `smtp_auth_pass`.
`from_email`	Email address of the sender.
`job_expiry_delay`	The number of hours after a job finishes that the job becomes a candidate for cleanup. (The default is 168 hours, or 7 days.)
`worker_stale_delay`	Number of seconds after a sensor stops communicating that it becomes stale. Assign a value that is larger than the `worker_sleep_interval` and `worker_jobwatcher_interval`defined for any sensor.
`worker_inactive_delay`	Number of minutes after a sensor becomes inactive that all of its unfinished jobs are marked as faulted. Assign a value that is much larger than `worker_stale_delay`. Note that this option uses different time units than does `worker_stale_delay`.
`worker_expiry_delay`	Number of hours after a sensor stops communicating that it becomes a candidate for cleanup. (The default is 168 hours, or 7 days.)
`cleanup_period`	Frequency (in minutes) with which expired jobs and sensors are cleaned up. (The default is 60.)
`ssc_url`	URL for the Fortify Software Security Center server; all uploads are

Option	Description
	sent to this address. Example: `https://<ssc_host>:<port>/ssc`
`ssc_lockdown_mode`	If set to `true`, ScanCentral SAST clients are forced to work with the ScanCentral Controller throughFortify Software Security Center. Jobs must be uploaded to a Fortify Software Security Center application version (a job cannot be started without the upload). In SSC lockdown mode, users cannot assign scans to specific sensor pools manually. Instead , the mapping configured on Fortify Software Security Center for the selected application version is applied. In SSC lockdown mode, you: • Cannot use the client command `-url` option, but must use the `-ssc_url` option with the `-ssc_token` option instead • Must specify the application version, the application version id, and the `-upload` option when starting the scan • Cannot specify the `-pool` option, because the job is assigned to the pool configured for the specified application version
`ssc_scancentral_ctrl_secret`	Password that Fortify Software Security Center uses to request data from the 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 page 27. **Note:** The `ssc_cloudctrl_secret` option is supported for backward compatibility with Fortify CloudScan.
`ssc_lockdown_mode`	If set to true, ScanCentral SAST clients are forced to work with the ScanCentral Controller through Fortify Software Security Center.
`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

Option	Description
	page 29.
`this_url`	URL for the Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/scancentral-ctrl`
`ssc_remote_ip`	Remote IP address You can configure an allowed remote IP address for Fortify Software Security Center. Only requests with a matching remote IP address are allowed.
`ssc_remote_ip_header`	Remote IP HTTP header, where the Fortify Software Security Center remote IP is found if `ssc_remote_ip_trusted_proxies_range` is set. The default value is `X-FORWARDED-FOR`.
`ssc_remote_ip_trusted_proxies_range`	Remote IP range (in CIDR format) Set this if Fortify Software Security Centeraccesses the Controller via (reverse) proxy server. You can specify comma-separated IP addresses or CIDR network ranges. This is disabled by default, which means that `ssc_remote_ip_header` is never used to retrieve the remote IP address for Fortify Software Security Center.
`remote_ip_proxy_header`	Remote IP proxy header
`client_auto_update`	If set to `true`, enables the Controller to automatically update all outdated sensors and clients. For details, see "Enabling and Disabling Auto-Updates of Clients and Sensors" on page 55.
`email_allow_list`	Use this property to specify the list of email domains that the Controller can use to send notifications. Examples of valid values: `*@yourcompanyname.com`

Option	Description

	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name@yourcompanyname.com`
	To specify multiple values, you can use commas (s), colons (:), or semicolons (;) as delimiters.

`email_deny_list`	Use this property to specify the list of email domains that the Controller cannot use to send notifications.
	Examples of valid values:
	`*@yourcompanyname.com`
	`*@*yourcompanyname.com`
	`a*@yourcompanyname.com`
	`name@yourcompanyname.com`
	To specify multiple values, you can use commas (s), colons (:), or semicolons (;) as delimiters.

`fail_job_if_ssc_upload_data_invalid`	If ScanCentral SAST is configured to upload scan results to an application version in Fortify Software Security Center, and either the `ScanCentralCtrlToken` token has expired or the specified application version does not exist, scan jobs are run, but the upload to Fortify Software Security Center fails. (The default behavior.) If you set this option to `true`, before the Controller creates a job and assigns it to a sensor, it checks to make sure that the `ScanCentralCtrlToken` token has not expired, and that the application version exists in Fortify Software Security Center. The default value is `false`. If set to `true` and the `ScanCentralCtrlToken` token expires before a scan job is assigned to sensor, the scan does not run and the job fails.
If your remote IP address is different than the configured Fortify Software Security Center URL, you can use one of the following properties to set up the remote IP address.	
`ssc_remote_ip`	Remote IP address You can configure an allowed remote IP address for Fortify Software

Option	Description
	Security Center. Only requests with a matching remote IP address are allowed.
`ssc_remote_ip_trusted_proxies_range`	Remote IP range (in CIDR format)
`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 Controller. (For instructions, see "Starting the ScanCentral SAST Components" on page 48.)

See Also

"Installing the Controller" on page 19

"Configuring Job Cleanup Timing on Sensors" on page 66

Encrypting the Shared Secret

Passwords exist in the ScanCentral 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:

- `worker_auth_token`, `smtp_auth_pass` and `ssc_scancentral_ctrl_secret` properties in the `config.properties` file on the Controller
- `worker_auth_token` property in the `worker.properties` file on a sensor
- `client_auth_token` property in the `client.properties` file on a client

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, `<controller_dir>\bin\pwtool.bat <pwtool_key_filepath>`
   - On a Linux system, `<controller_dir>/bin/pwtool <pwtool_key_filepath>`
2. When prompted, type the password to encode, and then press **Enter**.

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

The pwtool generates a new key stored in the file on the path specified in step 1, or reuses an existing file on specified path.

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_scancentral_ctrl_secret`
   - client_auth_token

   Tip: Fortify recommends that you assign separate, unique shared secrets for the `worker_auth_token`, `smtp_auth_pass`, and `ssc_scancentral_ctrl_secret` properties.
4. 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.
5. Uncomment the following line (property) in the `config.properties` file:
   `Pwtools_keys_file=d:\SecretKeys\SecretKey.txt`
6. Save the `config.properties` file.

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 <pwtool_key_filepath>`
   - On a Linux system, `<sca_install_dir>/bin/pwtool <pwtool_key_filepath>`
2. When prompted, type the password to encode, and then press **Enter**.

   The pwtool generates a new `pwtool.keys` file to `<pwtool_key_filepath>` 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:

   `pwtool_keys_file=<pwtool_key_filepath>`

Encrypting the Shared Secret on a Client

To encrypt a shared secret on a client:

1. Run one of the following commands.
   - On a Windows system:
     - For a client used as part of Fortify Static Code Analyzer and applications, run `<sca_install_dir>\bin\pwtool.bat <pwtool_key_filepath>`
     - For a standalone client, run `<client_install_dir>\bin\pwtool.bat <pwtool_key_filepath>`
   - On a Linux system:
     - For a client used as part of Fortify Static Code Analyzer and applications, run `<sca_install_dir>/bin/pwtool <path_to_pwtool.keys>`
     - For a standalone client, run `<client_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 key in the file on the specified path, or reuses an existing file and prints the encrypted password.
3. Copy the new encrypted secret, and paste it as the value for the `client_auth_token` property in the `client.properties` file.
4. Add the following to the `client.properties` file:

   `pwtool_keys_file=<pwtool_key_filename>`

See Also

"Configuring the ScanCentral SAST Controller" on page 21

"Creating ScanCentral SAST Sensors" on page 41

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`— In this mode, a ScanCentral SAST client requests a specific sensor pool when it submits a scan request. Otherwise, the default pool is used.
- `ENABLED`— 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 ScanCentral 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 ScanCentral SAST client command line).

- `ENFORCED`—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.

  If `ssc_lockdown_mode` is enabled, then the value set for `pool_mapping_mode` in the `config.properties` file is ignored and `pool_mapping_mode` is automatically set to ENFORCED.

The following table shows how the Fortify Software Security Center integration with Fortify ScanCentral SAST 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) specified	Default pool	Denied	Denied
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

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

See Also

"Configuring the ScanCentral SAST Controller" on page 21

Configuring the Logging Level on the Controller

To configure the logging level on the Controller:

1. Navigate to `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`, and open the `log4j2.xml` file in a text editor.
2. Locate one of the following strings:
   - `<Logger name="com.fortify.cloud" level="info" additivity="false">`
   - `<Logger name="com.fortify.cloud.ctrl.service" level="info" additivity="false">`
3. For a more detailed level of logging, change the level, as follows:

   `<Logger name="com.fortify.cloud" level="debug" additivity="false">`

   Standard log levels supported by `log4j2.xml` are as follows.

Standard Level	intLevel
OFF	0
FATAL	100
ERROR	200
WARN	300
INFO	400
DEBUG	500
TRACE	600
ALL	Integer.MAX_VALUE

4. To apply the change, restart the Controller.

For more information about defining custom log levels, see the Apache Logging Services website (https://logging.apache.org/log4j/2.x/manual/customloglevels.html).

Securing the Controller

The following procedure describes how to create a secure connection (HTTPS) between the ScanCentral SAST Controller/Tomcat server and ScanCentral SAST 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 Controller/Tomcat server and ScanCentral 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

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

Prompt	Value
	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.
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\conf` 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) `<controller_dir>\tomcat\webapps\scancentral-ctrl\WEB-INF\classes`
   - (Linux) `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes`
6. Update the `this_url` property, with your https address and port.

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

7. Restart your Tomcat server.
8. Set up your clients and sensors. For information about how to set up the ScanCentral SAST clients and sensors, see "Creating ScanCentral SAST Clients" on page 39 and, "Creating ScanCentral SAST Sensors" on page 41, respectively.
9. Add your self-signed certificate to the java keystore on all entities that communicate with the Controller (includes all clients, sensors, and Fortify Software Security Center installations) as follows:
   a. For ScanCentral SASTclients and sensors, open a command prompt and type the following:

      `cd <sca_install_dir>\jre\bin`

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

      For a installation or for standalone ScanCentral SAST 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	Type the name of the group that is to use the certificate.

Prompt	Data
organizational unit?	(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. Navigate to one of the following directories, and then open the `config.properties` in a text editor:

   On a Windows system:

```
<controller_dir>\tomcat\webapps\scancentral-ctrl\WEB-INF\classes
```

   On a Linux system:

```
<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes
```

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

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

See Also

"Securing the Controller for Authorized Client Use Only" below

Securing the Controller for Authorized Client Use Only

You can restrict the use of the ScanCentral Controller to authorized clients only.

To secure the Controller for use by authorized clients only:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Set the `client_auth_token` property.
3. On the client machine, go to the `Core/config` directory, and open the `client.properties` in a text editor.

   The `client_auth_token` property can be stored in the `client.properties` file as plain text, or as an encrypted key. For information about how to generate an encrypted key for `client_auth_token`, see "Encrypting the Shared Secret" on page 27.
4. Add the `client_auth_token` property to the file, and then set the same value for it that you gave to the `client_auth_token` property in step 2.
5. Start the Controller.

Allowing CloudScan Clients that do not Support Client Authentication to Connect to the Controller

If you have CloudScan version 19.2.0 or earlier clients that do not support client authentication, you can enable them to connect to the Controller.

To enable CloudScan version 19.2.0 or earlier clients to connect to the Controller:

1. Navigate to the `ControllerTomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Set the `allow_insecured_clients_with_empty_token` property to `true`.

If you set the `allow_insecured_clients_with_empty_token` property to `false`, only clients that support the `client_auth_token` can connect to Controller.

Note: If a client (CloudScan 19.2.1 or ScanCentral 20.1.0 and later client) supports the `client_auth_token` property and that property value is left unspecified, the client cannot connect to the Controller even if the `allow_insecured_clients_with_empty_token` is

set to `true`, *unless* the `client_auth_token` value on the Controller is also left unspecified.

Securing ScanCentral SAST 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 ScanCentral SAST components. The security vulnerability summaries that Fortify products provide may mandate an even higher level of secure deployment.

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

Creating ScanCentral SAST 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 ScanCentral SAST 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 and container configurations that are supported for 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, ASP.NET, and .NET Standard)
- Dockerfiles

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 ScanCentral SAST sensors, you can use standalone clients.

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

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

Creating an Embedded Client Using Fortify Static Code Analyzer

Use the following procedure to create an embedded client (client included with SCA and Apps) 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 an embedded 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 Client

Important! Fortify recommends that your standalone ScanCentral SAST clients and yourFortify Static Code Analyzer installation be the same version.

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

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

  Or,

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

To update an embedded 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 ScanCentral` (in the `<sca_install_dir>/bin/ScanCentral` 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 45

"Configuring Sensors to Use the Progress Command when Starting on Java" on page 44

Creating ScanCentral SAST Sensors

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

Creating a Sensor Using Static Code Analyzer 21.1.0

The following procedure describes how to create a new sensor. For information about how to upgrade an existing sensor, see "Upgrading ScanCentral SAST Sensors" on page 53.

Note: If you use Windows, you can install the sensor as a Windows service. For instructions, see "Creating a ScanCentral SAST Sensor as a Service" below.

To create a sensor:

1. Log in to the build machine using an account that is not an administrator or root.
2. Install Fortify Static Code Analyzer 21.1.0. (For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.)
3. Navigate to the `<sca_install_dir>\Core\config` directory, and open the `worker.properties` file in a text editor.
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 28.
6. Save and close your `worker.properties` file.

Creating a ScanCentral SAST 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\ScanCentral-worker-service` directory, and then do one of the following:
   - To use a clear text password, run `setupworkerservice.bat <sca_version> <full_controller_url> <shared_secret>`
   - To use an encrypted password, run `setupworkerservice.bat <sca_version> <full_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.

     Caution! The `setupworkerservice` command does not correctly handle worker_auth_token tokens that contain the caret character (^). If you must use the caret character as a part of a worker_auth_token, use the following formula:

     ```
     saved_caret_count = carets_used_on_command_line / 8
     ```

     Examples:
     For a worker_auth_token that contains a single caret, such as this^that, run the following command:

```
setupworkerservice.bat 21.1 http://url.com this^^^^^^^^that
```

For a worker_auth_token that contains two caret characters, such as this^^that, run the following command:

```
setupworkerservice.bat 21.1 http://url.com
this^^^^^^^^^^^^^^^^that
```

For information about how to encrypt a shared secret, see "Encrypting the Shared Secret on a Sensor" on page 28.

2. Start the service, as follows:

```
net start FortifyScanCentralWorkerService
```

The services installer creates the `<sca_install_dir>\Core\config\worker.properties` file for you.

See Next

"Enabling Sensor Auto-Start on Windows as a Service" on page 72

See Also

"Fortify ScanCentral SAST Components" on page 18

"Creating ScanCentral SAST Sensors" on page 41

Changing Sensor Expiration Time

By default, sensors expire 168 hours after they become inactive. To reset this default value:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory, and open the `config.properties` file in a text editor.
2. Locate the `worker_expiry_delay` setting, and then change the number of hours to elapse after inactivity before sensors expire.

See Also

"Creating ScanCentral SAST Sensors" on page 41

Support for Multiple Fortify Static Code Analyzer Versions

To support heterogeneous environments and facilitate phased Fortify Static Code Analyzer upgrades, the ScanCentral Controller supports scan request routing based on the 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 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 `scancentral.bat`) in your path and a mixed version environment, make sure that you are running the latest ScanCentral SAST executable when you run the client and 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 sensor machines dedicated to ScanCentral SAST and run sensors under a dedicated username. Run only one sensor instance per machine, and do not run any other Java processes under the same username after you start the ScanCentral Controller.

If the Controller and Fortify Software Security Center run on different machines, you must check to make sure that `scancentral-ctrl\WEB-INF\classes\config.properties (ssc_url, this_url)` and the ScanCentral Controller URL set on Fortify Software Security Center (select **Administration** > **Configuration** > **ScanCentral SAST**) 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:

- Controller to Fortify Software Security Center port (for scan uploads)
- Fortify Software Security Center to the ScanCentral Controller port (for Fortify ScanCentral SAST administration console functionality)
- Clients to the ScanCentral Controller port
- Sensors to the ScanCentral Controller port
- 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)

Configuring Sensors to Use the Progress Command when Starting on Java

If you plan to start your ScanCentral SAST sensors on Java, 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, ScanCentral SAST clients start 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 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 ScanCentral SAST instances, each with its own `worker.properties` file.

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

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

ScanCentral SAST client machine requirements:

- MSBuild (version that corresponds to the version released with Visual Studio 2019, or earlier)
- NuGet (optional)
- .NET Framework, .NET Core, or .NET Standard, depending on project configuration
- Windows operating system

ScanCentral SAST sensor machine requirements:

- .NET Framework supported for Fortify Static Code Analyzer

  Note: The .NET Framework version installed on a sensor machine must be the same as or later than the version that the project to be translated requires. This means, for example, that you cannot run a translation of a project that uses .NET Framework 4.8 on a sensor that has .NET Framework 4.7.2 installed.

- Windows operating system

Tip: For information about specific version requirements, see the *Micro Focus Fortify Software System Requirements* document.

Beginning with (CloudScan) version 19.2.0, remote translation and scanning for .NET projects were supported. ScanCentral SAST supports the same MSBuild versions as Fortify Static Code Analyzer. (.NET packaging and scanning work 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.
- 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 ScanCentral SAST, 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 ScanCentral SAST 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 ScanCentral SAST Integration

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

Some projects also require that you start NuGet 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, ScanCentral SAST runs it automatically.

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

```
scancentral -url <scancentral_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 ScanCentral does not try to detect the `*.sln` file.

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

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

To send the package to ScanCentral SAST, run:

```
scancentral -url <scancentral_url> start -package <package path>
```

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

Excluding .NET Projects from Translation

The ScanCentral SAST 20.1.0 client does not support the Fortify Static Code Analyzer flag `-exclude-disabled-projects`. To exclude a .NET project from translation, you must use the ScanCentral SAST `arguments` command. Invoke the `arguments` action and specify the `-targs` flag with `-exclude Src/<excluded_project>` value, where `<excluded_project>` is the project directory.

The following example shows how to exclude SubprojectB from translation and scanning:

```
ProjectRoot
+- MySolution.sln
+- SubprojectA
   +- SubprojectA.csproj
   +- ...
+- SubprojectB
   +- SubprojectB.csproj
   +- ...
```

The following command (invoked from project root folder) creates a `fortify-sca.settings` file under `ProjectRoot`:

```
scancentral arguments -targs -exclude Src/SubprojectB
```

The following command (invoked from project root folder) starts the remote translation of the project, with SubprojectB excluded:

```
scancentral -url <controller_url> start -bt msbuild -bf <MySolution.sln>
```

Fortify Static Code Analyzer Mobile Build Session Version Compatibility

The Fortify Static Code Analyzer version on a ScanCentral 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 19.2.0.0080). The major and minor portions of the Fortify Static Code Analyzer version numbers on both the ScanCentral client and sensor must match. For example, 19.2.0 works with 19.2.x.

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

Starting the ScanCentral SAST Components

Before you begin to use ScanCentral SAST:

1. If you plan to upload your scan results to Fortify Software Security Center, make sure that Fortify Software Security Center is up and running.
2. Wait until the Controller is up and running.
3. Check to make sure that the sensors and clients are up and running.

Starting the Controller

To start the Controller:

1. If you plan to upload your scan results to Fortify Software Security Center, check to make sure that the Fortify Software Security Center instance is running.
2. On the machine that hosts the Controller, navigate to the Tomcat `<bin>` directory:
   On a Windows system:

   ```
   (cd <controller_dir>\tomcat\bin)
   ```

   On a Linux system:

   ```
   (cd <controller_dir>/tomcat/bin)
   ```
3. 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 ScanCentral SAST Sensors

To start the sensors:

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

   ```
   scancentral.bat -url <sc_controller_url> worker
   ```

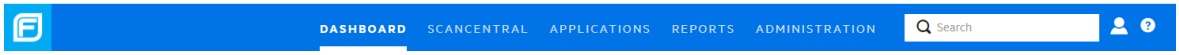
   On a Linux system:

   ```
   ./ScanCentral -url <sc_controller_url> 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 72.

Note: Make sure that you run a given sensor consistently from the same directory. Otherwise, its UUID changes and, if ScanCentral SAST 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 ScanCentral SAST is integrated with Fortify Software Security Center, after you log in to Fortify Software Security Center, notice that the Fortify header now includes the **SCANCENTRAL** link. If you do not see the **SCANCENTRAL** link in the header, log out, open a new browser window, and then log in again. If the **SCANCENTRAL** link is still missing from the header, check to make sure that the connection between Fortify Software Security Center and ScanCentral SAST is set up. (See "Configuring the Connection to Fortify Software Security Center" on page 68.)

Stopping the Controller

To stop the ScanCentral SAST Controller:

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

   ```
   cd <controller_dir>\tomcat\bin
   ```

   On a Linux system:

   ```
   cd <controller_dir>/tomcat/bin
   ```

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

   ```
   shutdown.bat
   ```

   On a Linux system:

   ```
   ./shutdown.sh
   ```

Chapter 3: About Upgrading ScanCentral SAST Components

ScanCentral SAST-related functionality in Fortify Software Security Center requires an updated ScanCentral SAST Controller and sensors. If you need remote translation and scan functionality, use client, sensor, and Controller 19.1.0 or later version.

Important! You must upgrade the Controller before you upgrade the ScanCentral SAST sensors and clients, *and* before you upgrade the Fortify Software Security Center server. Also, make sure that your Controller version is the same as your Fortify Software Security Center server version.

Caution! A version 21.1.0 sensor does not support packages generated by clients of an earlier version. If you want to offload translation for scan projects uploaded by CloudScan client 19.2.0, do not upgrade your sensors to ScanCentral SAST version 20.2.0 or 21.1.0.

This section contains the following topics:

Upgrading the ScanCentral SAST Controller

The following procedure describes how to upgrade the 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.

Note: The following compatibility rules apply to remote translation:
- The version 21.1.x Controller assigns jobs from ScanCentral versions 19.2.x and 20.1.x to Fortify Static Code Analyzer 20.1.x only.
- You cannot use Fortify Static Code Analyzer 19.2.x with a version 21.1.x Controller. This also applies to remote scans.

To upgrade your ScanCentral 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_ScanCentral_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_controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes/config.properties` file.)
4. Shut down the Controller.
5. Install the new Controller. (For information, see "Installing the 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 ScanCentral SAST Components" on the previous page

"Upgrading ScanCentral SAST Sensors" below

"Enabling and Disabling Auto-Updates of Clients and Sensors" on page 55

Upgrading ScanCentral SAST Sensors

Important! If Fortify Static Code Analyzer and applications is installed in a location that requires that you have administrator privileges to modify it (for example, program files), in order to update a sensor, you must start it with administrator privileges. Otherwise, the sensor cannot write files to disk. If auto-update is enabled, major updates on standalone clients must finish successfully before the sensor can start. With auto-update enabled, patch updates allow sensors and clients to start unless the upgrade fails.

To upgrade your ScanCentral SAST sensors (on Windows or Linux), you can either install the latest version of Fortify Static Code Analyzer, or unzip the `Fortify_ScanCentral_Client_<version>_x64.zip` file. You can use the client-only approach if you will plan only to use remote translation and analysis workflows. Local translation requires a local Fortify Static Code Analyzer installation. You can also find the ScanCentral SAST client inside the `Fortify_ScanCentral_Controller_<version>_x64.zip` file in the `Tomcat/client/scancentral.zip` directory.

Tip: You can configure automatic upgrades of both ScanCentral SAST sensors and clients. For details, see "Enabling and Disabling Auto-Updates of Clients and Sensors" on the next page.

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 agencies)
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`

   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 User Guide*.
5. Check the `<sca_install_dir>/Core/config` directory to make sure that the `worker.property` file resides there.
6. Add the following property to the `worker.properties` file:

   ```
   worker_auth_token=<value_set_in_controller_configuration>
   ```
7. 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 28.
8. Save the `worker.properties` file.
9. Start the sensors.

See Also

"About Upgrading ScanCentral SAST Components" on page 52

"Enabling and Disabling Auto-Updates of Clients and Sensors" on the next page

"Upgrading the ScanCentral SAST Controller" on page 52

"Creating ScanCentral SAST Clients" on page 39

"Creating ScanCentral SAST Sensors" on page 41

Enabling and Disabling Auto-Updates of Clients and Sensors

You can have all ScanCentral SAST clients and sensors check with the Controller after a manual update and following each startup to determine whether updates are available (the client or sensor version is earlier than the Controller version). Then, if an update is available, the Controller updates all sensors and clients.

The upgrade paths for clients and sensors as of the v 20.2.0 release of Fortify ScanCentral SAST are as follows:

- Standalone clients can be upgraded to a patch or major version (for example from 20.2.0 to 21.1.0, or 20.2.0 to 20.2.1).
- If auto-upgrade is enabled and a major upgrade of standalone clients fails, the clients do not start any jobs until they are upgraded.
- If auto-upgrade is enabled and a patch upgrade of standalone clients fails, the clients continue to work, but a warning is displayed.
- You can upgrade embedded clients and sensors to a patch version only (for example, from 20.2.0 to 20.2.1 or 20.2.2 , but not to 21.1.0). Major auto-upgrade is not available for embedded clients and sensors.
- If auto-upgrade is enabled and a patch upgrade of an embedded client fails, the clients and sensors continue to work but a warning is displayed.

To upgrade sensors and embedded clients to the next version, you must install the latest SCA and Apps version.

About Scan Assignment

Clients can assign scans to Fortify Static Code Analyzer instances that have the same major version and any patch of that version. For example, a 21.1.0 client can send scans to Fortify Static Code Analyzer versions 21.1.0, 21.1.1, 21.1.2, and so on. However, a client cannot assign scans to Fortify Static Code Analyzer of a different major version. For example, 20.2.0 clients cannot send scans to Fortify Static Code Analyzer version 21.1.0.

Important! ScanCentral SAST clients and sensors check for updates only if you use the `-url` or `-sscurl` options. The `package` command will not start the update process.

To enable or disable automatic updates of your clients and sensors:

1. Navigate to the `<controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-INF/classes` directory and open the `config.properties` file in a text editor.
2. Locate the `client_auto_update` property.
3. To enable auto-updates, set `client_auto_update` to `true`. To disable auto-updates, set the value to `false` (the default).
4. Save and close the file.

The update process (and its resulting success or failure status) is printed to the console.

Important! If Fortify Static Code Analyzer and applications is installed in a location that requires that you have administrator privileges to modify it (for example, program files), in order to update the sensor, you must start it with administrator privileges. Otherwise, the sensor cannot write files to disk. If auto-update is enabled, major updates on standalone clients must finish successfully before the sensor can start. With auto-update enabled, patch updates allow sensors and clients to start unless the upgrade fails.

See Also

"About Upgrading ScanCentral SAST Components" on page 52

"Upgrading the ScanCentral SAST Controller" on page 52

Chapter 4: 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 ScanCentral SAST sensors.

Offloading Scanning Only

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

```
scancentral.bat –url <controller_url> start -b <my_build_id> -scan
```

You can pass any relevant Fortify Static Code Analyzer scan tuning option 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  –build-label \"Application 5.4 – August 20, 2021\"
```

If the submission succeeds, you receive a token ID. The Fortify ScanCentral SAST 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 1 GB. Before you start large scans, review "Optimizing Scan Performance" on page 78.

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 **SCANCENTRAL**.

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.

   Note: All sensors that are unassigned and enabled are used, even they are not assigned to sensor pools.

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

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

```
scancentral.bat –url <controller_url> 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 ScanCentral SAST sensors. If your build tool is Apache Maven, Gradle, or MSBuild, make sure that you include the `-bt` option.

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

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

Objective	Command
Start a job to scan a MSBuild project	`scancentral.bat -url <controller_url> start -bt msbuild -bf mySolution.sln`
Start a job to scan a Maven project that includes the test scope	`scancentral.bat -url <controller_url> start -bt mvn -t`
Start a job to scan a Maven project with a non-default build file	`scancentral.bat -url <controller_url> start -bt mvn -bf c:\myproj\myproj-pom.xml`
Start a job to scan a JavaScript/TypeScript project	`scancentral.bat -url <controller_url> start -bt none`
Start a job to scan a PHP 7.1 project	`scancentral.bat -url <controller_`

Objective	Command
	url> start -bt none -hv 7.1
Start a job to scan an ABAP project	scancentral.bat -url <*controller_url*> start -bt none
Start a job to scan a Ruby project	scancentral.bat -url <*controller_url*> start -bt none
Start a job to scan a Gradle project	scancentral.bat -url <*controller_url*> start -bt gradle
Start a job to scan a Gradle project, get email notifications from the Controller, and upload the results to Fortify Software Security Center	scancentral.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	scancentral.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	scancentral.bat -url <*controller_url*> start -bt none
Start a job to scan a Python project under an active virtual environment without project dependencies installed	scancentral.bat -url <*controller_url*> start -bt none --python-requirements <*path_to_requirements_file*>
Start a job to scan a Python project using an existing Python virtual environment and install project dependencies	scancentral.bat -url <*controller_url*> start -bt none --python-virtual-env <*virtual_environment_location*> --python-requirements <*path_to_requirements_file*>

You can use ScanCentral SAST to work with Python in any of three ways. You can start ScanCentral SAST in a prepared virtual environment (see "Starting ScanCentral SAST in a

Virtual Environment" on the next page). You can use an existing virtual environment, without activating that virtual environment (see "Starting ScanCentral SAST in an Unactivated Virtual Environment" below). In this case, ScanCentral SAST activates the virtual environment itself. Finally, you can start the job outside of a virtual environment (see "Starting ScanCentral SAST Outside of a Virtual Environment" on the next page).

Starting ScanCentral SAST 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 ScanCentral SAST, or to specify the Python version (this is detected automatically).

To start ScanCentral SAST in a virtual environment:

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

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

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

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

Starting ScanCentral SAST in an Unactivated Virtual Environment

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

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

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

   or

   ```
   scancentral -url <controller_url> start -bt none --python-virtual-
   env <venv_location> --python-requirements <path_to_requirements_
   file>
   ```

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

Starting ScanCentral SAST Outside of a Virtual Environment

If you plan to start ScanCentral SAST 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. ScanCentral SAST locates the Python installation. In this case, ScanCentral SAST creates a temporary virtual environment, installs all dependencies from the requirements file, and then generates the package.

To start ScanCentral SAST outside of a virtual environment:

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

```
scancentral –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 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 an Apex project, run the following:

```
scancentral arguments -targs "-apex"
```

Note: For information on using the `-sargs` and `-targs` options, see the "Arguments Command" section in "Fortify ScanCentral SAST Command Options" on page 79.

To scan the project using ScanCentral SAST, run the following:

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

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

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

To send an existing package to ScanCentral SAST, run the following:

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

ScanCentral SAST 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 specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" what type of SQL (T-SQL or PL/SQL) is required. (By default, on Windows, Fortify Static Code Analyzer uses T-SQL, but on Linux, it uses PL/SQL.)

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

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

Note: For informationabout using the `-sargs` and `-targs` options, see "Fortify ScanCentral SAST Command Options" on page 79.

To scan the project, run the following command:

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

Alternatively, to save the package locally, run:

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

To send existing package to ScanCentral SAST, run:

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

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

See Also

"Fortify ScanCentral SAST Command Options" on page 79

"Submitting Scan Requests" on page 57

"Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 70

Generating a ScanCentral SAST Package

The examples listed in the following table illustrate various ways to generate a ScanCentral SAST package.

Objective	Command
Create a package from a Gradle project	`scancentral.bat package -bt gradle -o myPackage.zip`
Create a package from a Maven project with a `custom pom.xml`	`scancentral.bat package -bt mvn -bf myCustomPom.xml -o myPackage.zip`
Create a package from an MSBuild project	`scancentral.bat package -bt msbuild -bf mySolution.sln -o myPackage.zip`
Create a package from a JavaScript/TypeScript project	`scancentral.bat package -bt none -o myPackage.zip`
Create a package from a JavaScript/TypeScript project and include the node_modules **Caution!** This may greatly increase the package size as well as the scan time.	`scancentral.bat package -bt none --scan-node-modules -o myPackage.zip`
Create a package from a PHP 7.1 project	`scancentral.bat package -bt none -hv 7.1 -o myPackage.zip`
Create a package from an ABAP project	`scancentral.bat package -bt none -o myPackage.zip`
Create a package from a Ruby project	`scancentral.bat package -bt none -o myPackage.zip`
Create a package from a Python 2 project	`scancentral.bat package -bt none -yv 2 -pyr <path_to_requirements_file> -o myPackage.zip`
Create a package from a Python project under an active virtual environment with dependencies already installed	`scancentral.bat package -bt none -o myPackage.zip`

Objective	Command
Create a package from a Python project under an active virtual environment without project dependencies installed	`scancentral.bat package -bt none -pyr <path_to_requirements_file> -o myPackage.zip`
Create a package from a Python project using an existing Python virtual environment and install project dependencies	`scancentral.bat package -bt none -pyv <virtual_environment_location> -pyr <path_to_requirements_file> -o myPackage.zip`

Using the PackageScanner Tool

The packagescanner tool (`packagescanner.bat` on Windows and `packagescanner` on Linux) takes a package generated using the ScanCentral package command, generates Fortify Static Code Analyzer commands, and then performs a scan using a locally installed Fortify Static Code Analyzer instance. The tool is located in the `<sca_install_dir>/bin<sca_install_dir>/bin` directory.

The command-line options used with the packagescanner tool are described in the following table.

Option	Description
`-b, --build-id <id>`	(Optional) Specifies the build ID. Fortify Static Code Analyzer uses the build ID to track which files are compiled and combined as part of a build, and later, to scan those files. If you do not specify a build ID, ScanCentral SAST generates one based on language, number of projects, and so on.
`-debug`	Enables debug logging on ScanCentral SASTclients and sensors.
`-fpr`	(Required) Path of saved FPR files
`-package`	(Required) Path to the package file generated by the ScanCentral SAST command-line interface.

Option	Description
`-sargs, --scan-arguments`	(Optional) Additional Fortify Static Code Analyzer scan options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and parameter.
`-sca-path`	(Optional if started from Fortify Static Code Analyzer and apps) Path to the Fortify Static Code Analyzer executable. If ScanCentral SAST is part of SCA and Apps, the path is determined automatically.
`--sca-scan-log`	(Optional) Log for a scan command. By default, the log file is created in a temp folder, which is removed after program execution.
`--sca-translation-log`	(Optional) Log for all translation commands. By default, the log file is created in a temp folder, which is removed after program execution.
`-targs, --translation-arguments`	(Optional) Fortify Static Code Analyzer translation options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and parameter.
`-version`	(Optional) PackageScanner version.

Retrieving Scan Results from the Controller

To retrieve scan results, run the following command:

```
scancentral.bat -url <controller_url> retrieve -token <tokenid> -f worker.fpr
-log worker.log
```

Viewing Scan Request Status

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

```
scancentral.bat –url http://<Controller_Host>:8080/scancentral-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*.

Viewing Client and Sensor Logs

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

- Navigate to `%FORTIFY_HOME%\scancentral-<version>\log`, where `%FORTIFY_HOME%` is `${win32.LocalAppdata}\Fortify`.
  On Windows 10, for example, the location is
  `C:\Users\<user>\AppData\Local\Fortify\scancentral-<version>\log`

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

- To retrieve the ScanCentral SAST log, navigate to `~/.fortify/scancentral-<version>/log/scancentral.log`.
- To retrieve the ScanCentral Controller log, navigate to one of the following:
  - On Windows, `<controller_dir>\tomcat\logs\scancentralCtrl.log`
  - On Linux, `<controller_dir>/tomcat/logs/scancentralCtrl.log`
- To retrieve the Fortify Software Security Center log, navigate to `<fortify.home>/<app_context>/logs`.

Configuring Job Cleanup Timing on Sensors

To prevent the progressive loss of disc space as job files accumulate, Fortify ScanCentral SAST sensors automatically clean up internal job files (packages received from the Controller, FPRs, logs, and so on), and Fortify Static Code Analyzer build files related to cleaned ScanCentral jobs. Although you cannot disable this feature, you can configure its timing.

To configure the timing of job file cleanup on a sensor:

1. Navigate to the <*sca_install_dir*>./Core/config directory, and then open the worker.properties file in a text editor.
2. Configure the following properties based on your scheduling needs:

Property Name	Description	Default Value (hours)
worker_cleanup_age	Age (in hours) job files must be before they are removed from the sensor working directory	168 (or, one week)
worker_cleanup_ interval	Frequency with which the cleanup process runs	1

3. Save and close your worker.properties file.
4. Restart the sensor.

Chapter 5: Working with ScanCentral SAST 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 ScanCentral SAST, then the Fortify Software Security Center Scans view includes the ScanCentral SAST pages, which are described in the following table.

Scans View Page	Functionality
Scan Requests	View and export ScanCentral SAST 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 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 Controller can upload scan results directly to Fortify Software Security Center application versions.
- You can create and manage ScanCentral SAST sensor pools from Fortify Software Security Center. (For information about sensor pools, see the *Micro Focus Fortify Software Security Center User Guide*.)

To integrate Fortify Software Security Center with ScanCentral SAST:

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 **ScanCentral SAST**.

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

   Note: The `ssc_cloudctrl_secret` key is supported for backward compatibility with Fortify CloudScan.
7. Click **SAVE**.
8. Restart the Fortify Software Security Center server.

Important! You must use the same or a later version of ScanCentral SAST as the Fortify Static Code Analyzer version installed on your clients.

See Also

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

"Starting the ScanCentral SAST Components" on page 48

Chapter 6: Submitting Scan Requests and Uploading Results to Fortify Software Security Center

To submit a scan request, the results of which you want to upload to an application version in Fortify Software Security Center, use the fortifyclient tool to obtain the application version ID, and access tokens from Fortify Software Security Center. You can reuse the token for future requests. For information about how to use 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, and must have access to the application version on Fortify Software Security Center. A user who submits a ScanCentral SAST 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 <ssc_url> -user <user> -password
   <pwd>
   ```

   Sample Output

   ```
   ID      Name                    Version
   10      ScanCentral Test        1.0
   12      ScanCentral 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 Controller token, run the following command:

```
fortifyclient.bat token -gettoken ScanCentralCtrlToken -url <ssc_url> -user
<user> -password <pwd>

Authorization Token: <..scancentralCtrlToken...>
```

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

```
scancentral.bat -sscurl <ssc_url> -ssctoken <ScanCentralCtrlToken> start
-upload -versionid 10 -b <mybuildId> -uptoken <ScancentralCtrlToken> -scan
```

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 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 ScanCentral SAST; 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\ScanCentral-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 ScanCentral SAST Sensor as a Service" on page 42

Troubleshooting

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

- Main ScanCentral SAST sensor log:

  On Windows

  ```
  C:\Windows\System32\config\systemprofile\AppData\Local\Fortify\
  scanCentral\scancentral.log
  ```

  On Linux

  ```
  .fortify/scancentral/log/scancentral.log
  ```

- Sensor temporary folders that contain MBS files, Fortify Static Code Analyzer log files, and generated FPR files: `c:\ScanCentralWorkdir\<job_token>`
- Sensor `stdOut` and `stdErr` logs: `c:\ScanCentralWorkdir\workerout.log` and `c:\ScanCentralWorkdir\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:\ScanCentralWorkdir\<year_month_day>.log`

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

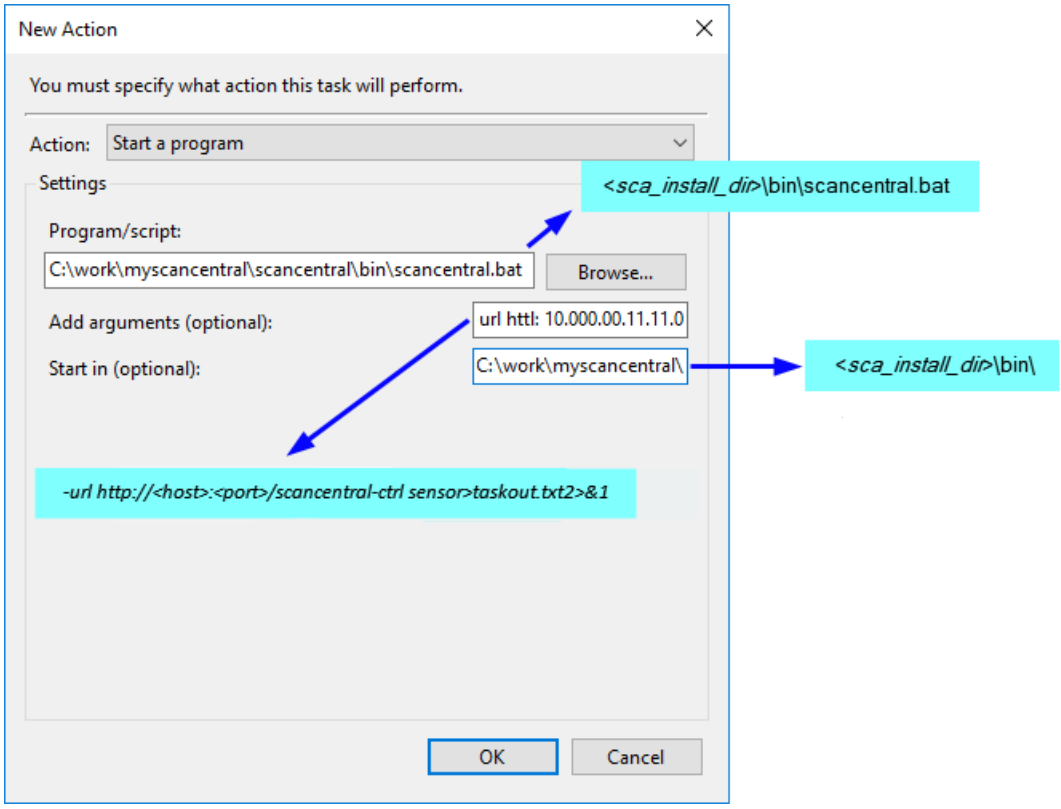
1. Log on to the sensor machine as the local admin user.

   Note: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of Fortify ScanCentral SAST; 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.

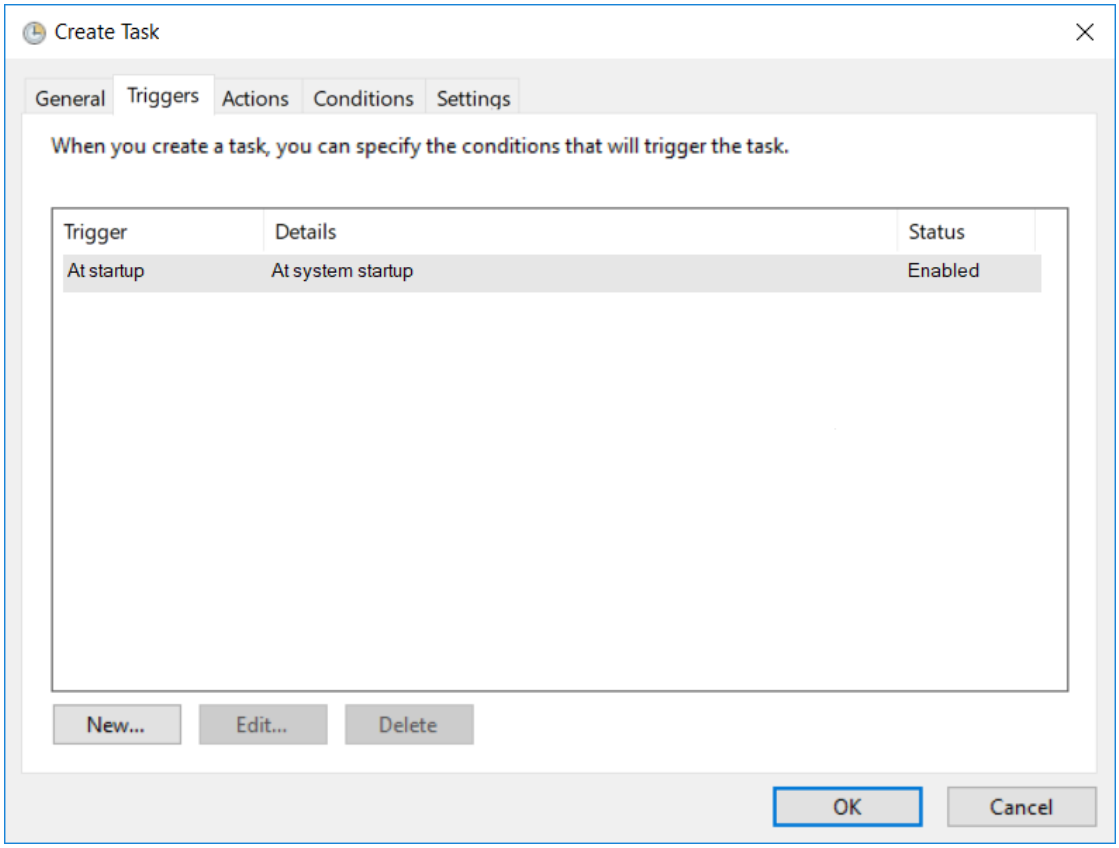
5. Select the **Actions** tab, and then click **New**.
   The New Action dialog box opens.



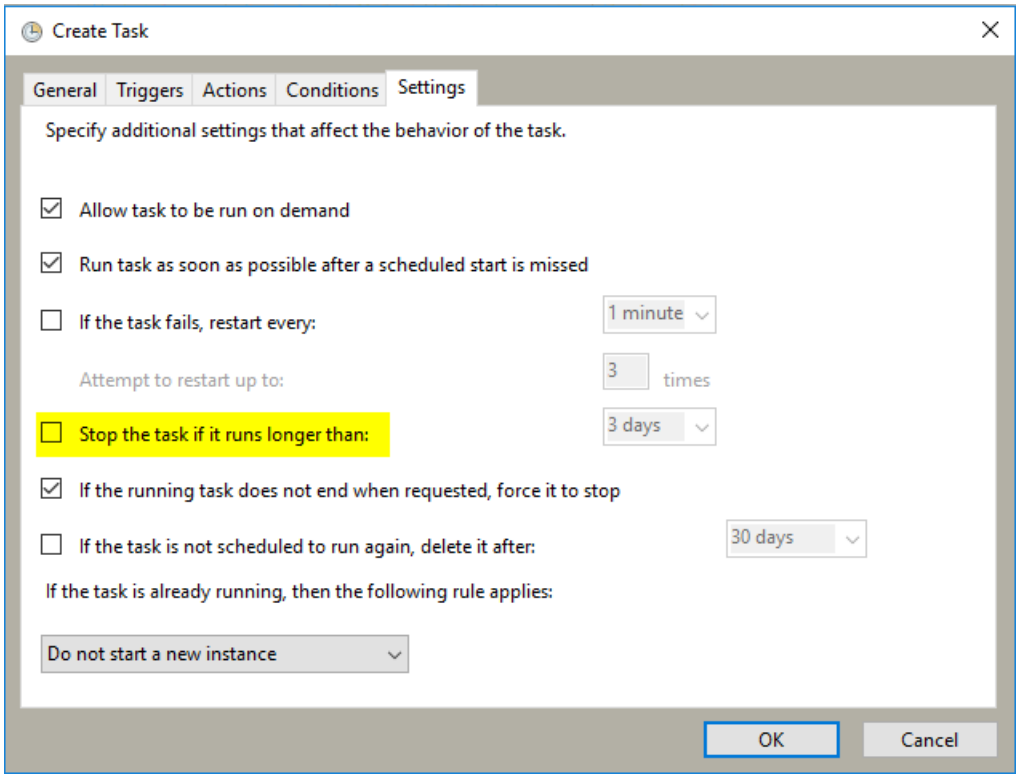
   a. From the **Action** list, select a program to start.
   b. In the **Program/script** box, type the directory path to your `scancentral.bat` file.
      Example: *<sca_install_dir>*`\bin\scancentral.bat`
   c. In the **Add arguments (optional)** box, type the following:

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

   d. In the **Start in (optional)** box, type the path to the ScanCentral 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 sensor started successfully.

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

Enabling ScanCentral 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/ScanCentral -url
      <controller_url> 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 ScanCentral 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 `<scancentral_install_dir>\tomcat\webapps\scancentral-ctrl` directory on Windows (`<scancentral_install_dir>/tomcat/webapps/scancentral-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: Fortify ScanCentral SAST Command Options

This appendix provides information about the command-line options that you can use with Fortify ScanCentral SAST.

Global Options

This section provides information about the command-line options that you can use with Fortify ScanCentral SAST.

Global Option	Use to:
`-debug`	Enables debug logging on ScanCentral SAST clients and sensors. For information on how to configure the logging level on the Controller, see "Configuring the Logging Level on the Controller " on page 31.

`-h <command>`
or
`--help <command>`
	Get help for the selected command. To see all command help, type `-h all`.

`-ssctoken <ScanCentralCtrlToken>`	Specify the Fortify Software Security Center authorization token.
`-sscurl <url>`	Specify the Fortify Software Security Center server URL.
`-url <url>`	Specify the ScanCentral SAST 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>`	Specifies the Fortify Software Security Center application name.
`--application-version-<id>`	Specifies the Fortify Software Security Center application version ID.
`-bc, --build-command <commands>`	For use with Maven, Gradle and MSBuild. Specifies custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral stops working on the build. (Gradle only) If you *do not* use `-bc`, the default command, default tasks and target are invoked. If the build fails, ScanCentral displays a warning, but continues to work and then displays a message to indicate that the build procedure failed and your results may be incomplete.
`-b, --build-id <id>`	Specifies the build ID of the session to export.
`-bf, --build-file <file>`	Specifies the build file, unless it has a default name such

Option	Description
	as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
`-block`	Waits for the job to complete, and then downloads the result.
`-bt, --build-tool <name>`	Specifies the build tool name used for the project. You cannot use this option with the `-scan` option. Example: `-bt mvn -bc "package --setting custom.xml"`
`-email <address>`	Specifies the email address for job status notifications.
`-exclude-disabled-projects`	A boolean flag. By default (without this option), all projects in the solution, disabled or enabled, are translated. **Note:** This option is for the Fortify Static Code Analyzer command line, not the MSBuild command line. No environment variable is associated with this flag, but you can get the same behavior from the MSBuild integration by setting the `ScaExcludeDisabledProjects` property to true on the MSBuild command line.
`-f, --output-file <file>`	Specifies the name for the local FPR file output.
`-filter <file>`	Specifies the filter file to use during a scan (repeatable).
`-hv, --php-version <version>`	Specifies the PHP version.
`-log, --log-file <file>`	Specifies the name for the local log file output.
`-mbs <file>`	Specifies the mobile build session to upload.
`-o, --overwrite`	Overwrites the existing FPR or log with new data.
`-p, --package <file>`	Specifies the project package file to upload.
`-pool, --submit-to-pool`	Specifies the sensor pool to which to submit the job.

Option	Description
<*uuid*>	
`-projroot, --project-root` <*dir*>	Specifies the project directory for the mobile build session export.
`-projtl, --project-template` <*file*>	Specifies the issue template file to include.
`-pyr, --python-requirements` <*file*>	Specifies the Python project requirements file to install and collect dependencies.
`-pyv, --python-virtual-env` <*directory*>	Specifies the Python virtual environment location.
`-q, --quiet`	Prevents the printing of stdout from the build execution.
`-rules` <*file/dir*>	Specifies custom rules file or directory to use during the scan (repeatable).
`-sargs, --scan-args`	Fortify Static Code Analyzer scan arguments (repeatable) Takes a single string argument. For multiple scan arguments, use multiple `-sargs` options. If the scan option has a path parameter that includes a space, enclose the path with single quotes.
`-scan`	Sets the point beyond which all arguments are for sourceanalyzer. You cannot use this option with the `--build-tool` or `--package` option.
`-snm, --scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not greatly improve type resolution or dataflow, and can result in an excessive number of false positives, Fortify recommends that you exclude them from scans. By default, node_modules are not applied to a package unless you apply the `--scan-node-modules` option from the command line.

Option	Description
`-skipBuild`	Disables the project preparation build step before packaging. If you use `-skipBuild` option, the `-bc` option (if used) is ignored.
`-sp, --save-package <file>`	Specifies the package file to save after uploading. The file extension must be `*.zip`.
`-t, --include-test`	Includes test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
`-targs, --translation-args`	Fortify Static Code Analyzer translation arguments (repeatable) Takes a single string argument. For multiple translation arguments, use multiple `-targs` options. If the translation option has a path parameter that includes a space, enclose the path with single quotes.
`-upload, --upload-to-ssc`	Uploads the FPR to Fortify Software Security Center upon completion.
`-uptoken, --ssc-upload-token <token>`	Specifies the Fortify Software Security Center file upload token.
`-version, --application-version <name>`	Specifies the Fortify Software Security Center application version name.
`-yv, --python-version <version>`	Specifies the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral SAST 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 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
`-bc, --build-command <commands>`	Specify custom build parameters for preparing and building a project. For example, to invoke a Gradle build before packaging: `-Prelease=true clean customTask build` If you use the `-bc` option, and the build fails, ScanCentral stops working on the build. (Gradle only) If you *do not* use `-bc`, the default tasks and targets are invoked. If the build fails, ScanCentral SAST displays a warning, but continues. You can use this option with Maven, Gradle and MSBuild.
`-bf, --build-file <file>`	Specify the build file if you are not using a default name such as `build.gradle` or `pom.xml`.
`-bt, --build-tool <name>`	Specify the build tool name used for the project. You cannot use this option with the project.
`-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.

Option	Description
`-q, --quiet`	Prevent the printing of stdout from the build execution.
`-snm, --scan-node-modules`	Specifies node_modules dependencies in the package. If you set `--scan-node-modules`, all third-party library scan results are added to the resulting FPR. **Tip:** Because including node_modules dependencies in a package does not improve type resolution or dataflow results, and because they degrade translation and scan speed, Fortify recommends that you exclude them from scans. By default, node_modules are not applied to a package unless you apply the `--scan-node-modules` option from the command line.
`-skipBuild`	Disables the project preparation build step before packaging.
`-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 ScanCentral SAST 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 Analyzer scan 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.

```
scancentral.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" on page 44.

Packagescanner Command

Use the help command (-h or --help) to get the information listed in the following table.

Option	Description
`-b, --build-id <id>`	(Optional) Specifies the build ID. Fortify Static Code Analyzer uses the build ID to track which files are compiled and combined as part of a build, and later, to scan those files. If you do not specify a build ID, ScanCentral SAST generates one based on language, number of projects, and so on.
`-debug`	Enables debug logging on ScanCentral SAST clients

Option	Description
	and sensors.
-fpr	(Required) Path of saved FPR file.
-package	(Required) Path to the package file generated by the ScanCentral SAST command-line interface.
-sargs, --scan-arguments	(Optional) Additional Fortify Static Code Analyzer scan options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and parameter.
-sca-path	(Optional if ScanCentral SAST is part of the SCA and apps installation) Path to the Fortify Static Code Analyzer executable file. If ScanCentral SAST is part of SCA and Apps, the path is determined automatically.
-sca-scan-log	(Optional) Log for a scan command. By default, the log file is created in a temp folder, which is removed after program execution.
-sca-translation-log	(Optional) Log for all translation commands. By default, the log file is created in a temp folder, which is removed after program execution.
-targs,--translation-arguments	(Optional) Fortify Static Code Analyzer translation options. Enclose multiple options in quotes separated by spaces, or repeat this option for each Fortify Static Code Analyzer option and parameter.
-version	(Optional) PackageScanner version.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email.

Note: If you are experiencing a technical issue with our product, do not email the documentation team. Instead, contact Micro Focus Fortify Customer Support at https://www.microfocus.com/support so they can assist you.

If an email client is configured on this computer, click the link above to contact the documentation team and an email window opens with the following information in the subject line:

Feedback on Installation, Configuration, and Usage Guide (Fortify ScanCentral SAST 21.1.0)

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!