Micro Focus
Fortify ScanCentral

Software Version: 20.1.0

Installation, Configuration, and Usage Guide

Document Release Date: May 2020

Software Release Date: May 2020

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

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 May 11, 2020. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

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

Contents

Preface

Contacting Micro Focus Fortify Customer Support

You can contact Micro Focus Fortify Customer Support, manage your Support cases, acquire licenses, and manage your account on the following website:

https://softwaresupport.softwaregrp.com

For More Information

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

About the Documentation Set

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

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

Change Log

The following table lists changes made to this document.

Software Release / Document Version	**Changes**
20.1.0	All references to CloudScan were replaced with ScanCentral. **New topics** - "What's New in Micro Focus Fortify ScanCentral 20.1.0" on page 12 - "Securing the Controller for Authorized Client Use Only" on page 27 - "Enabling and Disabling Auto-Updates of ScanCentral Clients" on page 40 - "Using the Package Scanner Tool" on page 46 **Modified topics** - "Configuring the ScanCentral Controller" on page 17 - "(Windows only) Configuring Sensors to Offload Translation For .NET Languages" on page 33
19.2.0	**New topics** - "Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 32 - "(Windows only) Configuring Sensors to Offload Translation For .NET Languages" on page 33 **Modified topics** - "Installing the ScanCentral Controller" on page 15 was modified to reflect the new installation procedure used for installation on both Linux and Windows. - "Creating ScanCentral Clients" on page 28was modified to reflect the introduction of the `CloudScan_Client_<version>.zip` file, which is used to create stand-alone clients that support translation on CloudScan sensors. - "Upgrading the ScanCentral Controller" on page 38 was modified to reflect file name changes.

Software Release / Document Version	Changes
	• The procedure described in "Upgrading Fortify ScanCentral Sensors" on page 39 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 42. • New argument command options were added to "Fortify ScanCentral Command Options" on page 58. **Removed topics** Installing the CloudScan Controller on a Linux System Installing the CloudScan Controller on a Windows System
19.1.0	**New topics** • What's New in Micro Focus Fortify CloudScan 19.1.0 • "Fortify ScanCentral Command Options" on page 58 **Modified topics** • "Fortify ScanCentral Components" on page 14 • "Creating ScanCentral Clients" on page 28 • "Accessing Help for Command-Line Options" on page 64 • "Submitting Scan Requests" on page 42
18.20	Minor changes, including version number and the font used to display content.

Chapter 1: Introduction

With Fortify ScanCentral (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 ScanCentral 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 ScanCentral sensors. For information about the languages supported for offloading translation, see "Creating ScanCentral Clients" on page 28. 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, Fortify Software Security Center can direct ScanCentral to output FPR files directly to the server.

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-and-services/documentation. All guides are available in both PDF and HTML formats. Product help is available within the Fortify WebInspect products.

All Products

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

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

Micro Focus Fortify 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 SSC_Guide_<*version*>.pdf	This document provides Fortify Software Security Center users with detailed information about how to deploy and use Software Security Center. It provides all of the

Document / File Name	Description
	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 Eclipse User Guide Eclipse_Plugins_Guide_<*version*>.pdf	This document provides information about how to install and use the Fortify Complete and the Fortify Remediation Plugins for Eclipse.

Document / File Name	**Description**
Micro Focus Fortify Plugins for JetBrains IDEs User Guide JetBrains_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 Fortify ScanCentral 20.1.0

Micro Focus Fortify ScanCentral 20.1.0 includes the changes described here.

Product Name Change

As of this release, the product named Fortify CloudScan in earlier releases is now called Fortify ScanCentral.

Automatic Client Updates

Now, after each startup, the ScanCentral Controller checks clients to determine whether updates are available (the client software version is earlier than the Controller version). The Controller places any available update files in a specific directory. Client updates begin after you next start the Controller. For more information, see "Enabling and Disabling Auto-Updates of ScanCentral Clients" on page 40.

Securing the Controller for Authorized Client Use Only

You can now use the new `client_auth_token` property to restrict the use of the ScanCentral Controller to authorized clients only. For details see "Securing the Controller for Authorized Client Use Only" on page 27. The `client_auth_token` property value can be stored in the `client.properties` file as plain text, or as encrypted keys.

Chapter 2: Fortify ScanCentral Components

A Fortify ScanCentral installation includes the following three components:

- **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.

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

- **ScanCentral Controller**: Server that receives the Fortify Static Code Analyzer mobile build sessions and scan instructions from the ScanCentral clients (or project packages with translation and scan instructions), routes the information to ScanCentral sensors, and (optionally) uploads scan results (FPR files) to Fortify Software Security Center.
- **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 the analysis. For information about the languages supported for performing translation, see "Creating ScanCentral Clients" on page 28.

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

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

Installing and Configuring the Fortify ScanCentral Components

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

- ScanCentral Controller
- ScanCentral clients

- ScanCentral sensors
- (Optional) Fortify Software Security Center

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

This section contains the following topics:

Installing the ScanCentral Controller

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

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

Jobs are deleted from the Controller after seven days, unless you change the `job_expiry_delay` variable value of 168 hours in the `config.properties` file. (You can find the `config.properties` file in the *`<sc_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, *`<sc_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, *`<sc_controller_dir>`* resembles the following:

```
bin/

tomcat/

readme.txt
```

See Next

"Configuring the ScanCentral Controller" on page 17

For information about how to update your Controller, see "About Upgrading Fortify ScanCentral Components" on page 38 and "Upgrading the ScanCentral Controller" on page 38.

See Also

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

Installing and Uninstalling the ScanCentral Controller as a Service

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

Installing the ScanCentral Controller as a Service

To install the ScanCentral 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 `Tomcat` directory.
4. Navigate to the `<sc_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 `Tomcat` directory.
3. Navigate to the `<sc_controller_dir>/tomcat/bin` directory, and then run the following:

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

The service name must not contain any spaces.

Uninstalling the ScanCentral Controller Service

To uninstall the Apache Tomcat 9.0 service:

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

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

See Also

"Configuring the ScanCentral Controller" below

Configuring the ScanCentral Controller

After you install the ScanCentral 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 ScanCentral Controller:

1. Navigate to `<sc_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**
`allow_insecured_clients_with_empty_token`	If set to `true`, this enables CloudScan version 19.2.0 or earlier clients that do not support client authentication to connect to the Controller. For details, see "Securing the Controller for Authorized Client Use Only" on page 27.
`client_auto_update`	If set to `true`, this enables the Controller to automatically update all outdated ScanCentral clients. For details, see "Enabling and Disabling Auto-Updates of ScanCentral Clients" on page 40.
`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 20.
`worker_auth_token`	A string that contains no spaces or backslashes. If you prefer not to use plain text, you can use an encrypted shared secret as the value for this property. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret on the Controller" on page 19.
`ssc_url`	URL for the Fortify Software Security Center server; all uploads are sent to

Option	Description
	this address. Example: `https://<ssc_host>:<port>/ssc`
`this_url`	URL for the ScanCentral Controller; used in emails to refer to this server for manual job result downloads. Example: `https://<controller_host>:8443/scancentral-ctrl`
`ssc_cloudctrl_secret`	Password that Fortify Software Security Center uses to request data from the ScanCentral Controller. Specify a string that contains no spaces or backslashes. (Optional) Use an encrypted shared secret. For instructions on how to encrypt a shared secret, see "Encrypting the Shared Secret" on the next page.
`pool_mapping_mode`	Used to configure different modes for mapping scan requests to sensor pools. For information about the valid values for `pool_mapping_mode`, see "About the pool_mapping_mode Property" on page 1.
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
`ssc_remote_ip_trusted_proxies_range`	Remote IP range (in CIDR format)
`ssc_remote_ip_header`	Remote IP HTTP header The default value is `X-Forwarded-For`.
`remote_ip_proxy_header`	Remote IP proxy header
`ssc_trusted_proxies_remote_ip`	If `remote_ip_proxy_header` is set, you must also specify a value for this property.

3. Save and close your `config.properties` file.

4. Start the ScanCentral Controller. (For instructions, see "Starting the Fortify ScanCentral Components" on page 35.)

See Also

"Installing the ScanCentral Controller" on page 15

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_cloudctrl_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, `<`*`sc_controller_dir`*`>\bin\pwtool.bat <`*`pwtool_key_filepath`*`>`
   - On a Linux system, `<`*`sc_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_cloudctrl_secret`

   > **Tip:** Fortify recommends that you assign separate, unique shared secrets for the `worker_auth_token`, `smtp_auth_pass`, and `ssc_cloudctrl_secret` properties.
4. 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 Controller" on page 17

"Creating ScanCentral Sensors" on page 30

About Upgrading Fortify ScanCentral Components

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

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

Caution! A version 20.1.0 ScanCentral sensor does not support packages generated by version 19.2.0 clients. If you want to offload translation for scan projects uploaded by CloudScan client 19.2.0, do not upgrade your sensors to version 20.1.0.

This section contains the following topics:

Securing the ScanCentral Controller

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

"Creating a Secure Connection Using a Certificate Signed by a Certificate Signing Authority" on page 24

Creating a Secure Connection Using Self-Signed Certificates

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

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

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

- On a Linux system:

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

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

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

Prompt	Value
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) `<sc_controller_dir>\tomcat\webapps\scancentral-ctrl\WEB-INF\classes`
   - (Linux) `<sc_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 ScanCentral clients and sensors. For information about how to set up the ScanCentral clients and sensors, see "Creating ScanCentral Clients" on page 28 and, "Creating ScanCentral Sensors" on page 30, respectively.

9. Add your self-signed certificate to the java keystore on all entities that communicate with the ScanCentral Controller (includes all ScanCentral clients, ScanCentral sensors, and Fortify Software Security Center installations) as follows:

   a. For ScanCentral clients and ScanCentral sensors, open a command prompt and type the following:

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

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

For a Fortify Software Security Center installation or for standalone ScanCentral 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.

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

3. Generate a Certificate Signing Request (CSR).

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

   On a Windows system:

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

- On a Linux system:

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

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

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

   - On a Linux system:

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

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

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

   - On a Windows system:

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

   - On a Linux system:

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

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

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

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

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

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

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

7. Restart Tomcat Server.
8. In the `config.properties` file, update the `this_url` property with your secure URL:

a. Navigate to the `config.properties` file and open it in a text editor.

   On a Windows system:

   ```
   <sc_controller_dir>\tomcat\webapps\scancentral-ctrl\WEB-
   INF\classes\config.properties
   ```

   On a Linux system:

   ```
   <sc_controller_dir>/tomcat/webapps/scancentral-ctrl/WEB-
   INF/classes/config.properties
   ```

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

   Example: `this_url=https://<controller_host>:8443/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 `<sc_controller_dir>/tomcat/webapps/cloud-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 19.
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/cloud-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 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 components. The security vulnerability summaries that Fortify products provide may mandate an even higher level of secure deployment.

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

Creating ScanCentral 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 clients. If you use a language that supports offloading the translation phase of analysis to your sensors, you can create standalone clients, independent of Fortify Static Code Analyzer.

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

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

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

Creating a Standalone Client

If you plan to offload both the translation and scanning phases of analysis to your ScanCentral 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.

> **Important!** Before you can use a client, you must create a file named `client.properties` in the `Core/config` directory on the client machine.

Creating a Client Using Static Code Analyzer 20.1.0

Use the following procedure to create a ScanCentral client if:

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

  and/or

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

To create a client:

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

> **Important!** Before you can use a client, you must create a file named `client.properties` in the `Core/config` directory on the client machine.

Updating a ScanCentral Client

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

To update a standalone ScanCentral 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 a ScanCentral 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 33

"Configuring Sensors to Use the Progress Command when Starting on Java 11" on page 32

Creating ScanCentral Sensors

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

Creating a ScanCentral Sensor Using Static Code Analyzer 20.1.0

The following procedure describes how to create a new sensor. For information about how to upgrade an existing sensor, see "Upgrading Fortify ScanCentral Sensors" on page 39.

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

To create a ScanCentral sensor:

1. Log in to the build machine using an account that is not an administrator or root.
2. Install Fortify Static Code Analyzer 20.1.0. (For instructions, see the *Micro Focus Fortify Static Code Analyzer User Guide*.)

3. Create a file named `worker.properties` in the `<sca_install_dir>\Core\config` directory.
4. Add the following property to the `worker.properties` file:

   ```
   worker_auth_token=<value_set_in_controller_configuration>
   ```

5. Specify either a clear text password, or an encrypted shared secret (password the Controller uses to communicate with the sensor) as the `worker.properties` value. For information about how to generate an encrypted shared secret, see "Encrypting the Shared Secret on a Sensor" on page 20.
6. Save and close your `worker.properties` file.

Updating a Sensor Based on a Fortify Static Code Analyzer Version Earlier than 20.1.0

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

To create a sensor using a Fortify Static Code Analyzer version earlier than 20.1.0:

1. Log in to the build machine using an account that is not an administrator or root.
2. Install Fortify Static Code Analyzer on the build machine if it does not already have Fortify Static Code Analyzer installed. For more information about how to install Fortify Static Code Analyzer, see the *Micro Focus Fortify Static Code Analyzer User Guide*.
3. Back up the following directories:

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

4. Extract the contents of the `scancentral.zip` file to the `<sca_install_dir>\Core\config` directory (`<sca_install_dir>/Core/config` on Linux).
5. Accept all overwrite requests.

   Note: Linux users may also need to run `chmod +x ScanCentral` in the `bin` directory.

6. In the `<sca_install_dir>\Core\config` directory (`<sca_install_dir>/Core/config` on Linux), create a file named `worker.properties`.
7. In the `worker.properties` file, create the following property:

   ```
   worker_auth_token=<shared_secret>
   ```

Creating a ScanCentral 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_sc_controller_url> <shared_secret>`
   - To use an encrypted password, run `setupworkerservice.bat <sca_version> <full_sc_controller_url> "<encrypted_shared_secret>" <path_to_pwtool.keys_file>`

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

     For information about how to encrypt a shared secret, see "Encrypting the Shared Secret on a Sensor" on page 20.
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 ScanCentral Sensor Auto-Start on Windows as a Service" on page 51

See Also

"Fortify ScanCentral Components" on page 14

"Creating ScanCentral Sensors" on page 30

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

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

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

   ```
   <user_role> readonly
   ```

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

   ```
   <user_role> <password> readonly
   ```

   where `<user_role>` is the value you specified in the JMX access file.
3. Run one of the following commands:
   - On Windows systems, run `cacls jmxremote.password /P <username>:R`
   - On Linux systems, run `chmod 600 jmxremote.password`

4. Open the `worker.properties` file in a text editor, and then add the following properties to it:

   ```
   sca_jmx_port=<port>
   sca_jxm_access_file=<path_to_access_file>
   sca_jmx_password_file=<path_to_password_file>
   sca_jmx_password=<password>
   sca_jxm_user=<user_role>
   sca_jmx_auth=true
   ```

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

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

Important! If you use `sca_jmx_auth`, you can start only one ScanCentral 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 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 sensors for remote translation of code written in a .NET language, make sure that the following requirements are met.

ScanCentral client machine requirements:

- Java 8
- 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 sensor machine requirements:

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

Beginning with (CloudScan) version 19.2.0, remote translation and scanning for .NET and ASP.NET projects were supported. ScanCentral supports the same MSBuild versions as Fortify Static Code Analyzer. (.NET packaging and scanning works only on Windows systems.)

The requirements for using this feature are as follows:

- Configure at least one sensor with the software required to support .NET capability.
- ScanCentral 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, 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 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 Integration

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

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

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

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

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

Excluding .NET Projects from Translation

The ScanCentral 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 `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 <sc_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 Fortify ScanCentral Components

Before you begin to use Fortify ScanCentral:

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

Starting the ScanCentral Controller

To start the ScanCentral Controller:

1. On the machine that hosts the ScanCentral Controller, navigate to the Tomcat `<bin>` directory:

   On a Windows system:

   ```
   cd <sc_controller_dir>\tomcat\bin
   ```

   On a Linux system:

   ```
   cd <sc_controller_dir>/tomcat/bin
   ```

2. Run one of the following commands:

   - On a Windows system, run `startup.bat`.

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

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

Starting ScanCentral Sensors

To start the ScanCentral sensors:

1. Start the Controller if it is not already running.
2. On each sensor, navigate to the <*sca_install_dir*> directory of the installation directory, as follows:

   - 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
   ```

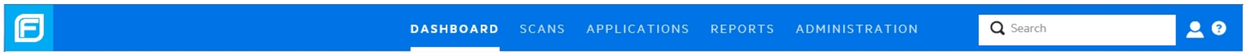
   On a Linux system:

   ```
   ./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 51.

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

Starting Fortify Software Security Center



Start Fortify Software Security Center. If Fortify ScanCentral is integrated with Fortify Software Security Center, after you log in to Fortify Software Security Center, notice that the Fortify header now includes the **SCANS** link. If you do not see the **SCANS** link in the header, log out, open a new browser window, and then log in again. If the **SCANS** link is still missing from the header, check to make sure that the connection between Fortify Software Security Center and Fortify ScanCentral is set up. (See "Configuring the Connection to Fortify Software Security Center" on page 48.)

Stopping the ScanCentral Controller

To stop the ScanCentral Controller:

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

   On a Windows system:

   ```
   cd <sc_controller_dir>\tomcat\bin
   ```

   On a Linux system:

   ```
   cd <sc_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 Fortify ScanCentral Components

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

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

> **Caution!** A version 20.1.0 ScanCentral sensor does not support packages generated by version 19.2.0 clients. If you want to offload translation for scan projects uploaded by CloudScan client 19.2.0, do not upgrade your sensors to version 20.1.0.

This section contains the following topics:

Upgrading the ScanCentral Controller

The following procedure described how to upgrade a ScanCentral Controller.

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

To upgrade your 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_sc_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 ScanCentral Controller" on page 15.)
6. If your existing `config.properties` file has been modified, you must merge it with the new `config.properties` file. (You cannot simply copy the existing `config.properties` file.)
7. Navigate to the `jobFiles` and `cloudCtrlDb` directories of the existing Controller, and then copy these to the new Controller.

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

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

See Also

"About Upgrading Fortify ScanCentral Components" on the previous page

"Upgrading the ScanCentral Controller" on the previous page

"Upgrading Fortify ScanCentral Sensors" below

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

Upgrading Fortify ScanCentral Sensors

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

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

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

   > **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 20.
8. Save the `worker.properties` file.
9. Start the sensors.

See Also

"About Upgrading Fortify ScanCentral Components" on page 38

"Upgrading the ScanCentral Controller" on page 38

"Creating ScanCentral Clients" on page 28

"Creating ScanCentral Sensors" on page 30

Enabling and Disabling Auto-Updates of ScanCentral Clients

By default, after each startup and after a Controller update, the Controller checks to determine whether updates are available for clients (the client software version is earlier than the Controller version). The Controller places any available update files in a specific directory. Client updates begin after you next start the Controller.

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

To enable or disable automatic updates of your ScanCentral clients:

1. Navigate to the `<sc_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 autoupdates, set `client_auto_update` to `true` (default). To disable auto-updates, set the value to `false`.
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 are installed in a location that requires that a user have administrator privileges to modify files that reside there (for example, program files), in order to update the client, you must start it with administrator privileges. If auto-update is enabled, the update process must finish successfully before the sensor can start.

See Also

"About Upgrading Fortify ScanCentral Components" on page 38

"Enabling and Disabling Auto-Updates of ScanCentral Clients" on the previous page

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 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 <sc_controller_url> start -b <my_build_id> -scan –Xmx2G
```

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

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

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

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

Note: Jobs submitted (and FPRs) can be no larger than 1GB. Before you start large scans, review "Optimizing Scan Performance" on page 57.

Targeting a Specific Sensor Pool for a Scan Request

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

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

To get the UUID for the sensor pool:

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

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

want to target for a scan request.

> **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 <sc_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 sensors. If your build tool is Apache Maven, Gradle, or MSBuild, include the `-bt` option.

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

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

Objective	**Command**
Start a job to scan a Gradle project	`scancentral.bat -url <controller_url> start -bt gradle`
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 Gradle project, get email notifications from the ScanCentral 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>`

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

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

To start ScanCentral in a virtual environment:

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

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

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

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

Starting ScanCentral in an Unactivated Virtual Environment

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

Starting ScanCentral Outside of a Virtual Environment

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

To start ScanCentral outside of a virtual environment:

1. Open a command line.
2. Start ScanCentral.
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 have Java 8 installed on your sensor. In addition, you must specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" that the CLS files are related to APEX, and not to Visual Basic 6.

To prepare for scanning, run the following:

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

Note: For information on using the `-sargs` and `-targs` options, see the "Arguments Command" section in "Submitting Scan Requests" on page 42.

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

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

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

Translating SQL Projects

To perform remote translation of a SQL project, you must have Java 8 installed on your sensor. In addition, you must specify an additional translation argument for the project so that Fortify Static Code Analyzer "knows" what type of SQL (TSQL or PL/SQL) is required. (By default, on Windows, Fortify Static Code Analyzer uses TSQL, but on UNIX, it uses PL/SQL.)

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

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

Note: For information on using the `-sargs` and `-targs` options, see the "Arguments Command" section in "Submitting Scan Requests" on page 42.

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, run:

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

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

See Also

"Fortify ScanCentral Command Options" on page 58

"Submitting Scan Requests" on page 42

"Submitting Scan Requests and Uploading Results to Fortify Software Security Center" on page 49

Using the Package Scanner Tool

The package scanner 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 `scancentral/bin` directory .

The command-line parameters used with the package scanner tool are described in the following table.

Parameter	Description
`-package`	(Required) Path to the package generated by the `packagescanner` command
`-fpr`	(Required) Path of saved FPR files
`-sca-path`	(Optional if started from Fortify Static Code Analyzer and apps) Path to the Fortify Static Code Analyzer executable
`-sca-log`	(Optional) Log for all clean, translation, and scan Fortify Static Code Analyzer commands. By default, the log file is created in a temp folder, which is removed after program execution.

Retrieving Scan Results from the ScanCentral Controller

To retrieve scan results, run the following command:

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

Chapter 4: Working with Fortify ScanCentral from Fortify Software Security Center

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

Scans View Page	Functionality

Scan Requests	View and export Fortify ScanCentral 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 ScanCentral 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 ScanCentral Controller can upload scan results directly to Fortify Software Security Center application versions.
- You can create and manage ScanCentral 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 and Fortify ScanCentral:

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**.

   The ScanCentral page opens.
3. To enable the polling of ScanCentral Controller to retrieve scan request status, select the **Enable ScanCentral** check box.
4. In the **ScanCentral Controller URL** box, type the URL for the ScanCentral Controller.
5. In the **ScanCentral poll period (seconds)** box, either select or type the number of seconds to elapse between ScanCentral 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 ScanCentral Controller. (If you use clear text, this string must match the value stored in the ScanCentral Controller `config.properties` file for the `ssc_cloudctrl_secret` key.)

7. Click **SAVE**.
8. Restart the Fortify Software Security Center server.

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

See Also

"Working with Fortify ScanCentral from Fortify Software Security Center" on page 47

"Starting the Fortify ScanCentral Components" on page 35

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 Fortify ScanCentral 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 ScanCentral 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 one of the following commands:

   ```
   scancentral.bat -sscurl <ssc_url> -ssctoken <ScanCentralCtrlToken> start -upload
   -versionid 10 -b <mybuildId> -uptoken <ScancentralCtrlToken> -scan –Xmx2G
   ```

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

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

Appendix A: Configuring Sensor Auto-Start

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

This section contains the following topics:

Enabling ScanCentral Sensor Auto-Start on Windows as a Service

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

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

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

   > **Note**: Sensors are dedicated machines that are meant only to run Fortify Static Code Analyzer on behalf of Fortify ScanCentral; 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 Sensor as a Service" on page 31

Troubleshooting

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

- Main ScanCentral 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`

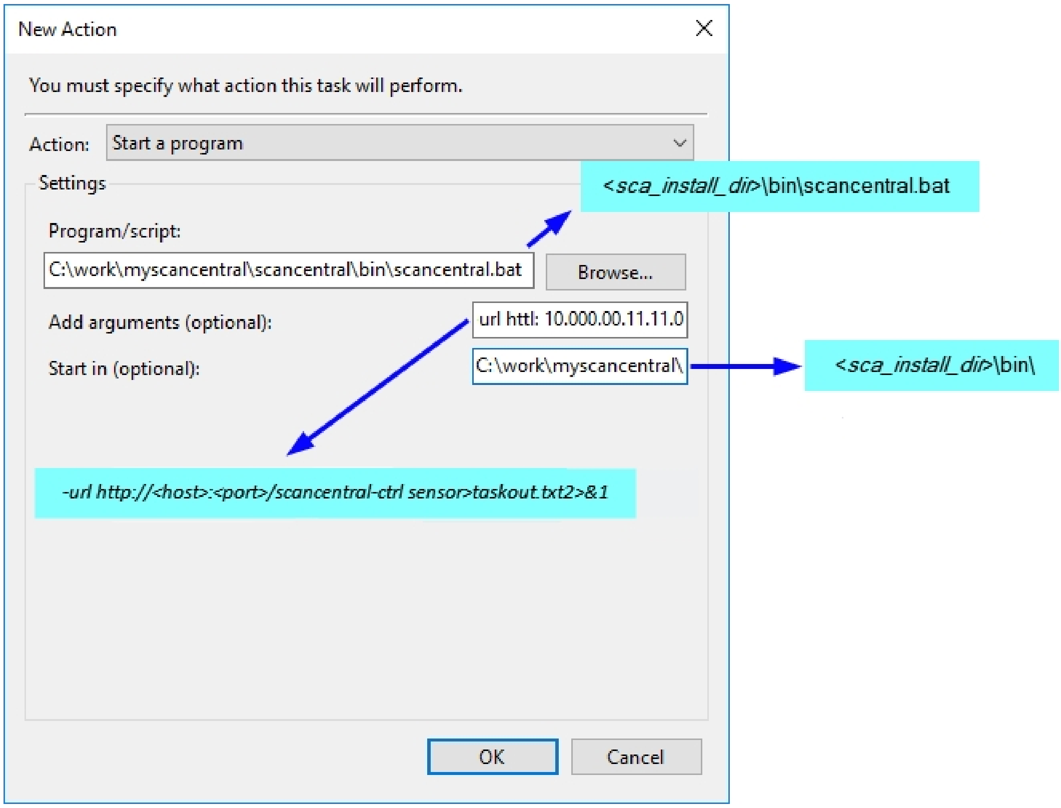
Enabling ScanCentral Sensor Auto-Start on Windows as a Scheduled Task

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; 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.



New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script: C:\work\myscancentral\scancentral\bin\scancentral.bat

Add arguments (optional): url httl: 10.000.00.11.11.0

Start in (optional): C:\work\myscancentral\

<sca_install_dir>\bin\scancentral.bat

<sca_install_dir>\bin\

-url http://<host>:<port>/scancentral-ctrl sensor>taskout.txt2>&1

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:

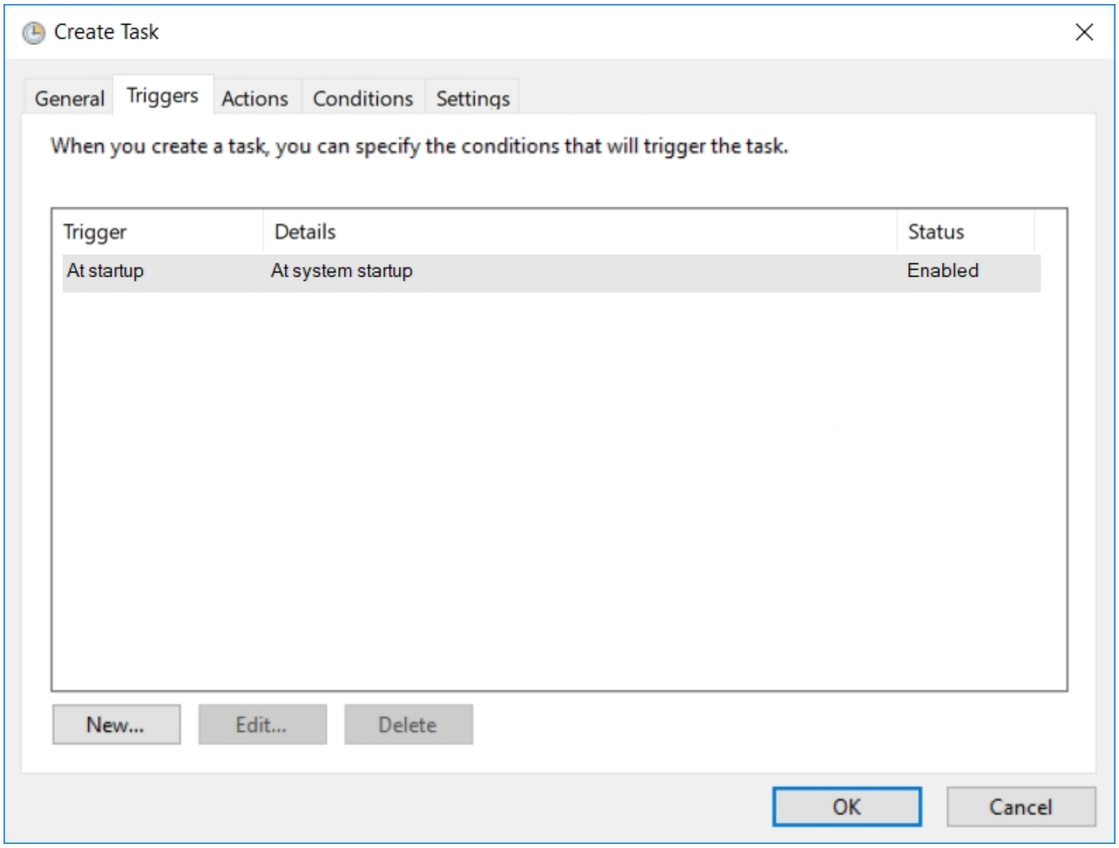
```
-url http://<host>:<port>/scancentral-ctrl worker >taskout.txt 2>&1
```

d. In the **Start in (optional)** box, type the path to the ScanCentral sensor `bin` directory.

   Example: `<`*`sca_install_dir`*`>\bin\`

e. Click **OK**.

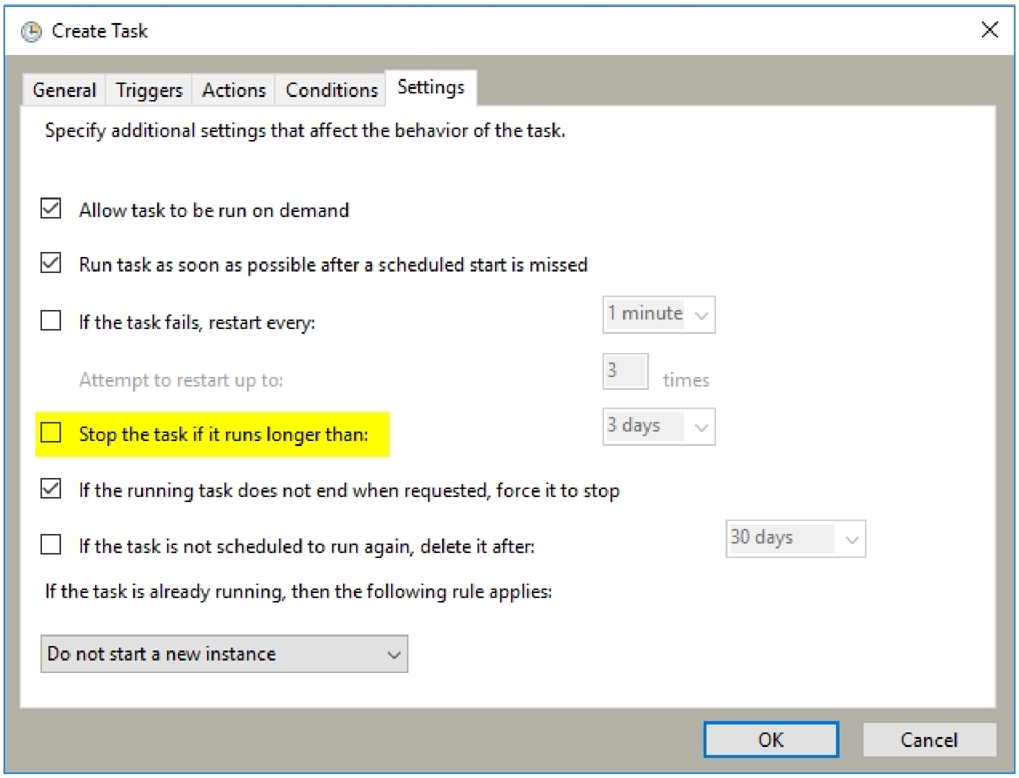
6. Return to the Task Scheduler and select the **Triggers** tab.



When you create a task, you can specify the conditions that will trigger the task.

Trigger	Details	Status
At startup	At system startup	Enabled

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



Specify additional settings that affect the behavior of the task.

☑ Allow task to be run on demand

☑ Run task as soon as possible after a scheduled start is missed

□ If the task fails, restart every: 1 minute

Attempt to restart up to: 3 times

□ Stop the task if it runs longer than: 3 days

☑ If the running task does not end when requested, force it to stop

□ If the task is not scheduled to run again, delete it after: 30 days

If the task is already running, then the following rule applies:

Do not start a new instance

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 ScanCentral 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 <sc_
   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 Command Options

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

Global Options

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

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

Status Command

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

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

Start Command

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

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

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

Option	Description
	installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral 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 ScanCentral sensor.

Option	Description
-hello	Sensor reporting for duty.

Package Command

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

Option	**Description**
-bf, --build-file <*file*>	Specify the build file if you are not using a default name such as `build.gradle` or `pom.xml`. You cannot use this option with the `-scan` option.
-bt, --build-tool <*name*>	Specify the build tool name used for the project. You cannot use this option with the `-scan` option.
-hv, --php-version <*version*>	Specify the PHP version.
-o, --output <*file*>	Specify the output file name. The file extension must be `*.zip`.
-pyr, --python-requirements <*file*>	Specify the Python project requirements file to install and collect dependencies.
-pyv, --python-virtual-env <*directory*>	Specify the Python virtual environment location.
-q, --quiet	Prevent the printing of stdout from the build execution.
-t, --include-test	Include the test source set (Gradle) or test scope (Maven) to scan (for Java projects only).
-yv, --python-version <*version*>	Specify the Python version to automatically find the installed Python. Allowed values: 2 or 3. This flag is ignored if the ScanCentral 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 11" on page 32.

Accessing Help for Command-Line Options

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

-h

-h start

-h worker

-h <*any_command_listed_with-help*>

For a complete list of all command-line options, see "Fortify ScanCentral Command Options" on page 58.

Send Documentation Feedback

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

Feedback on Installation, Configuration, and Usage Guide (Fortify ScanCentral 20.1.0)

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!