How To - CI/CD with Checkmarx

Checkmarx SAST

Checkmarx SAST (CxSAST) is an enterprise-grade flexible and accurate static analysis solution used to identify hundreds of security vulnerabilities in custom code. It is used by development, DevOps, and security teams to scan source code early in the SDLC, identify vulnerabilities and provide actionable insights to remediate them. Supporting over 22 coding and scripting languages and their frameworks with zero configuration to scan any language.

Ease of Automation

Seamlessly integrates with all IDEs, build management servers, bug tracking tools and source repositories to automatically enforce a security policy.

Manage Security at Scale

Empower teams to set and use policies to govern application security, enforce them through build-tool integrations and manage remediation efforts through IT workflow support.

Accelerate Time to Remediation

Allow developers to fix multiple vulnerabilities at a single point in the code using our unique “Best Fix Location” algorithm.

Find Vulnerabilities Sooner

Checkmarx SAST scans uncompelled code and doesn’t require complete build. No dependency configurations and no learning curve when switching languages!

Supported Major Programming Languages and Frameworks

Environment	Primary Languages	Secondary Languages	Frameworks
Java	•Java •J2SE •J2EE	•JSP •JavaScript •VBScript •PL\SQL •HTML5	•Struts •Spring MVC •Spring Dependency Injection •iBatis* •GWT •Hibernate •OWASP ESAPI •JSTL FMT Taglib •ATG DSP Taglib •Java Server Faces (JSF) •JSP •Google Guice •PrimeFaces
Microsoft C#.net	•C# •VB.NET	•ASP.NET •JavaScript •VBScript •PL\SQL •HTML5	•Enterprise Libraries •Telerik •ComponentArt •Infragistics •iBatis* •Hibernate.Net [*] •Entity framework •ASP.Net MVC framework •ASP.Net CORE Razor •ASP.NET Core
ASP Active Server Pages	ASP	•JavaScript [**] •VBScript •PL\SQL •HTML5	ASP.Net MVC framework
Microsoft Visual Basic	VB6		
C++	C/C++		•MISRA •Informix ESQL/C •MySQL
php	PHP		•Zend •Kohana •CakePHP •Symfony •Smarty •bWapp •OWASP ESAPI
Apex Develop On Demand	Apex	VisualForce	
Ruby	Ruby		Ruby on Rails
JS	•JavaScript •ES5 •ES6		•JQuery •Node.js •Ajax •Knockout •AngularJS •ExpressJS •Pug (Jade) •Handlebars •Cordova/PhoneGap •Hapi.JS •XS (SAP) •Backbone •Kony Visualizer •ReactJS* •SAPUI5
TypeScript	Typescript**		Angular
VBSCRIPT			
Perl	Perl		
	Android (Java)		Volley (Android)
Objective-C	Objective C Swift		
HTML 5			
PL SQL			
python	Python	•JavaScript •VB script •PL\SQL	Django
Groovy	Groovy	•JavaScript •VB script •PL\SQL	
Scala	Scala		Akka
GO	GO Language		Protobuf

Supported Vulnerabilities

HIGH RISK	MEDIUM THREAT	LOW VISIBILITY
CGI Reflected XSS	Access Control	Arithmetic Operation On Boolean
CGI Stored XSS	Buffer Overflow	Blind SQL Injections
Code Injection	CGI Reflected XSS All Clients	Client Side Only Validation
Command Injection	CGI Stored XSS	Cookie not Sent Over SSL
Connection String Injection	CGI XSS	Dangerous File Upload
LDAP Injection	Cookies Scoping	Dead Code
Process Control	Cross Site History Manipulation	Deprecated And Obsolete
Reflected XSS	DB Paramater Tampering	Deprecated CRT Functions VS2005
Reflected XSS All Clients	Dangerous Functions	DoS by Unreleased Resources
Resource Injection	Data Filter Injection	Equals without GetHashCode
SOQL SOSL Injection	DoS by Sleep	Escape False Warning
SQL injection	Double Free	Files Canonicalization Problems
Second Order SQL Injection	Environment Injection	Hardcoded Absolute Path
Stored XSS	Environment Manipulation	Hardcoded Password
UTF7 XSS	Files Manipulation	Password in Connection String
XPath Injection	Frame Spoofing	Impersonation Issue

Plugins available for :

CxPlugins page

CLI	Command Line Interface can be used from Windows or Linux OS *Cx Plugin Version: 8.90.2* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
Eclipse	Eclipse IDE Plugin *Cx Plugin Version: 89.0.0* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
IntelliJ	IntelliJ IDE Plugin *Cx Plugin Version: 8.90.0* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
Visual Studio	VS IDE Plugin *Cx Plugin Version: 8.50.2* | *CxSast Min Version: 8.5.0* | *Older Versions*	Download
Jenkins	Plugin for Jenkins build server *Cx Plugin Version: 8.90.4* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
SonarQube	Plugin for SonarQube (Sonar 6.3 - 6.7.1 LTS) *Cx Plugin Version: 8.90.0* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
SonarQube Widget	SonarQube Dashboard Widget (Sonar 4.5.4-6.1) *Cx Plugin Version: 8.42.0* | *CxSast Min Version: 8.4.1*	Download
Maven	Maven Plugin *Cx Plugin Version: 8.80.2* | *CxSast Min Version: 8.8.0* | *Older Versions*	Download
Bamboo	Bamboo Plugin *Cx Plugin Version: 8.90.0* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
TeamCity	TeamCity Plugin *Cx Plugin Version: 8.90.0* | *CxSast Min Version: 8.9.0* | *Older Versions*	Download
TFS	TFS Build server plugin *Cx Plugin Version: 1.4.0.2* | *CxSast Min Version: 7.1.2*	Download
CxAPI	CxAPI Examples *Cx Plugin Version: 7.2.3* | *CxSast Min Version: 7.2.3*	Download

To download above plugins go to https://www.checkmarx.com/plugins/

Installing CxSAST (v8.8.0 to v8.9.0)

Before installing CxSAST, make sure that you understand the System Architecture, that your server host(s) complies with the server host requirements, and that you have properly prepared the installation environment.

Prior to installing CxSAST, if not already installed on the server host, install the following prerequisites, which are included in the installation zip file (“third party” folder):

- **IIS (Windows 7 or greater) -** see the OS-specific instructions (IISInstallationProcess.rtf file)
- **MS SQL**
- **VC++ Runtime Redistributable**

For more information, see server host requirements.

If you are interested in configuring a High Availability solution please contact Checkmarx support.

If your portal is installed on a separate machine from manager, please perform the following procedure.

Installation Permissions

The user performing the installation must have administrative network permissions (user name and password) for the computer/server running CxSAST Services.

SQL Server Database

If the database uses **Windows domain authentication**, the machine with the product installed on it must be added to a Windows domain. In addition, the user account performing the installation (Centralized or CxManager) must have SA permission on the database server for the duration of the installation process. If SA permission is unavailable, certain prerequisites must be fulfilled prior to the installation:

- Build three SQL databases using the names; CxDB, CxActivity and CxARM.
- Create login for Windows User and associate it with DB_owner permission for CxDB, CxActivity and CxARM. This user should be a dedicated Service user and the same user must perform the installation, see Link for additional information.

If the database uses **SQL Server native authentication,** prepare an SQL Server user account. This account must have SA permissions for the duration of the installation process. If SA permission are unavailable, certain prerequisites must be fulfilled prior to the installation.

- Build three SQL databases using the names CxDB, CxActivity and CxARM.
- Create login for SQL User and associated it with the DB_owner permission for CxDB, CxActivity and CxARM. Define this user in the CxSAST installation.

For upgrades, all previously defined SQL connection parameters are loaded from the existing configuration. If Windows authentication is being used, run the installer with the same user that is defined for the CxServices or any other Windows authenticated user with DB owner permission on CxDB, CxActivity and CxARM.

AWS RDS

DBaaS is not supported natively. but AWS RDS can be used - To make RDS work you need to create three databases, CxDB, CxActivity and CxARM. Give the user that you created for Checkmarx dbo privileges to the newly created databases. Run the installer again and when the installation connects to the Database and you see a message about the three databases already existing, just click continue. Once the installation is complete the RDS should work.

Setting Up CxSAST

License Validation

It is recommended to obtain a license before you start your installation. This way you will be able to provide the license during the installation and be able to use the product immediately.

Your CxSAST license is tied to a specific machine (server); so all you have to do is to run the Cx HID Generator and a HID (hardware identification number) is provided. The HID Generator can be downloaded from the Cx Utilities page.

Please send the Hardware ID number to your technical contact or your sales manager. They will send you back your license. If you do not know who to send the Hardware ID to, please send it to support@checkmarx.com.

If you have already installed CxSAST and have not yet obtained a permanent CxSAST license, send your hardware ID (**Start** > **All Programs** > **C heckmarx** > **HardwareId**) to your Checkmarx sales representative or Checkmarx support to obtain a Production license file.

Installation Package

1. Download the CxSAST installation package.
2. On each server component host:
    a. Extract the downloaded ZIP archive, supplying the password provided by Checkmarx support.
    b. Run **CxSetup.exe** and begin the installation.

Installing CxSAST

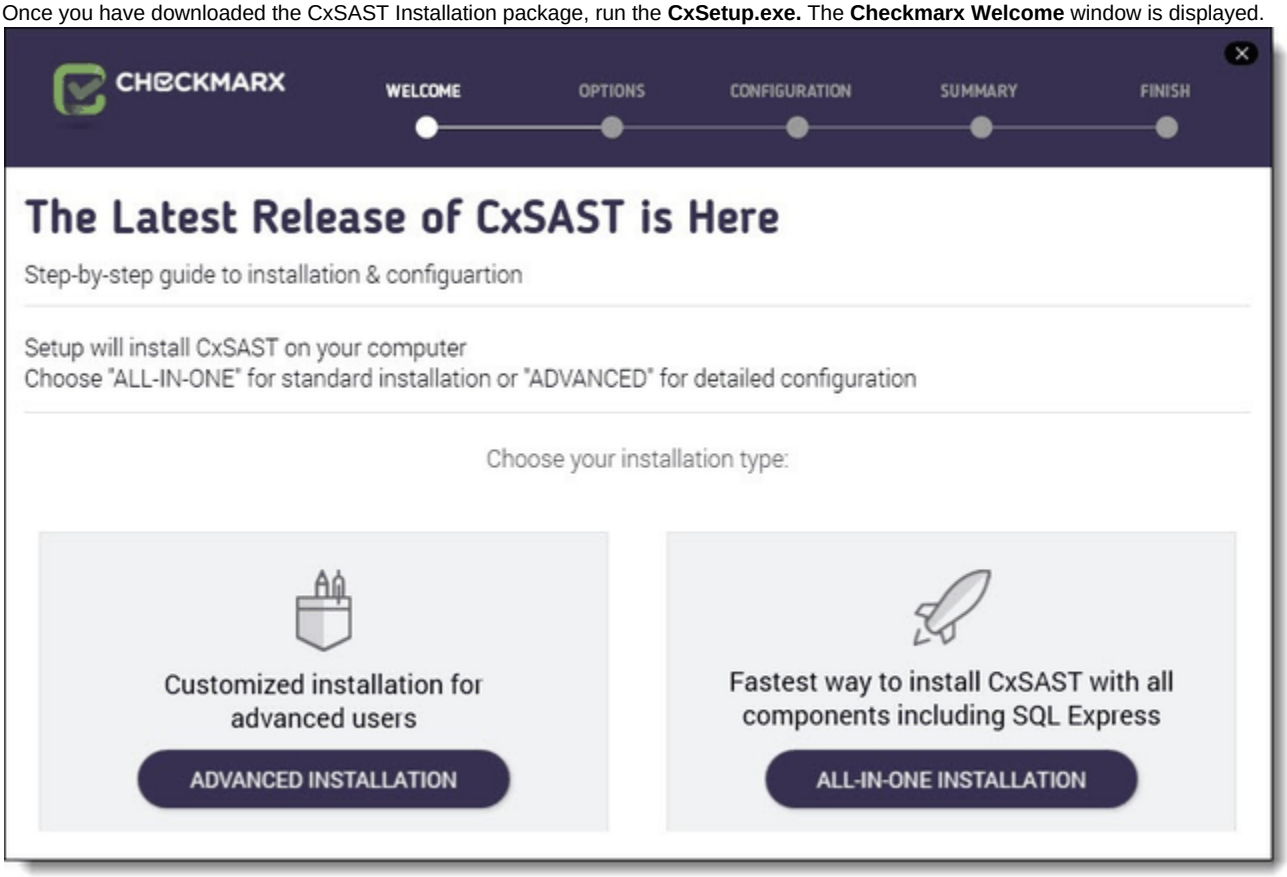
Prerequisites and Recommendations

- The installer requires **.Net**
    1. 7.1

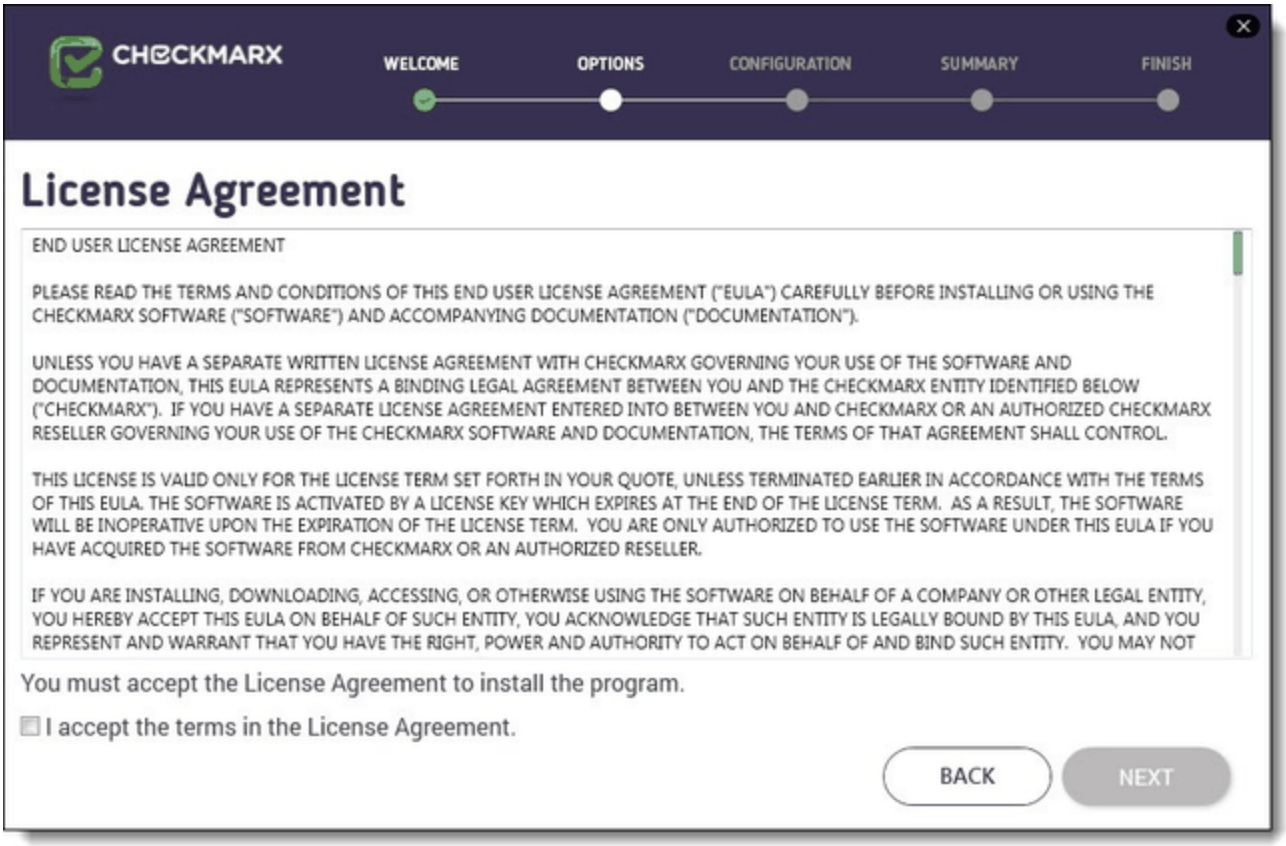
  Framework installed on your server (If missing, it will be installed by the CxSAST installer).
- The required Web Server for Checkmarx is IIS Server (if missing, it will be installed by the CxSAST installer on the condition that the Windows installation media is accessible).
- SQL 2012 Express is included with the CxSAST installer and is installed (if defined) in the event that no other version of SQL is already installed.

Installation

Once you have downloaded the CxSAST Installation package, run the **CxSetup.exe.** The **Checkmarx Welcome** window is displayed.

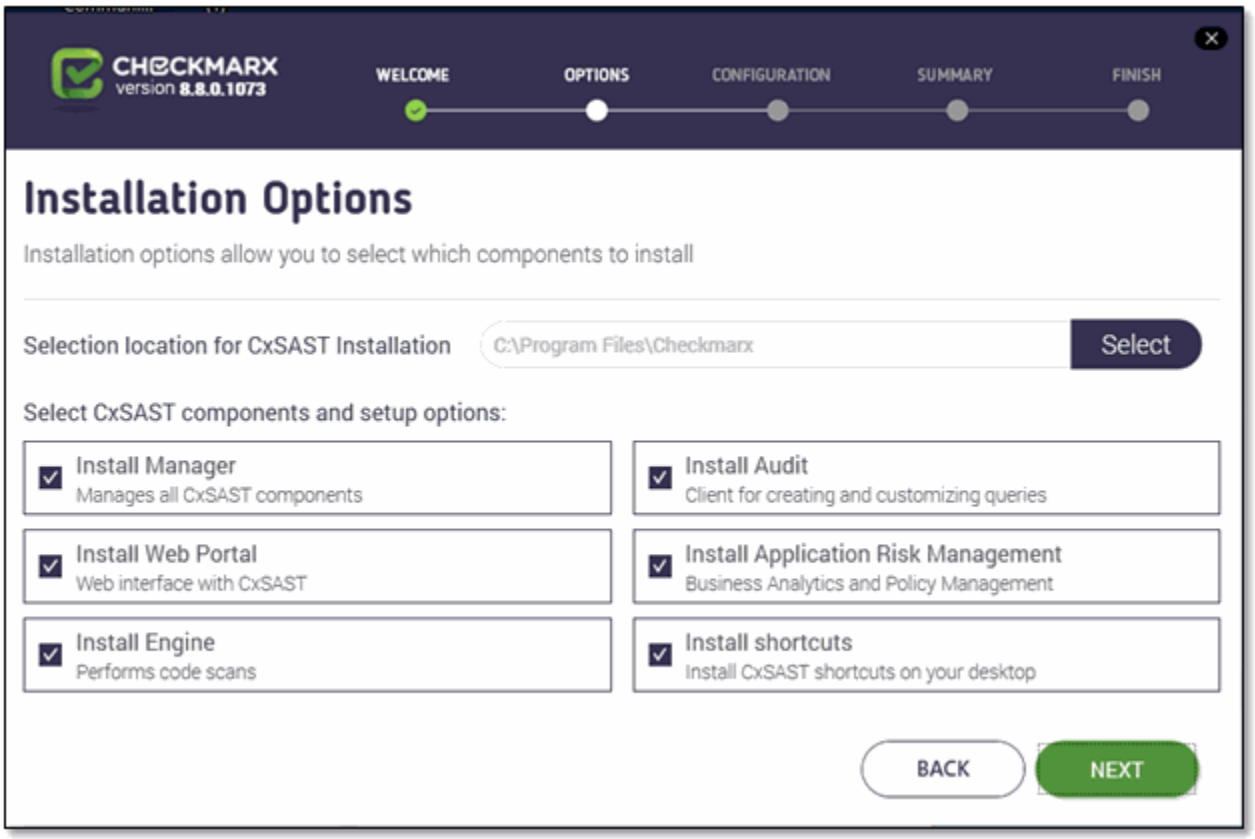


Click **ALL IN ONE** to continue, **ADVANCED** to define additional setup options, or **X** to exit. The **Checkmarx License Agreement** window is displayed.



Review and accept the license agreement by checking the '**I accept the terms in the License Agreement**' checkbox. Click **Next** to continue.

If you selected **ADVANCED**, the additional **Installation Options** window is displayed.



Click **Select** to define the CxSAST installation location.

Upgrade and Modify

For upgrades, previously installed location and product feature settings are loaded from the existing configuration and cannot be changed. You can however install or remove product features by using the modify feature.

Select the required product features for this installation from the available list. You can also select the option to install related shortcuts on your desktop.

Product Feature Selection

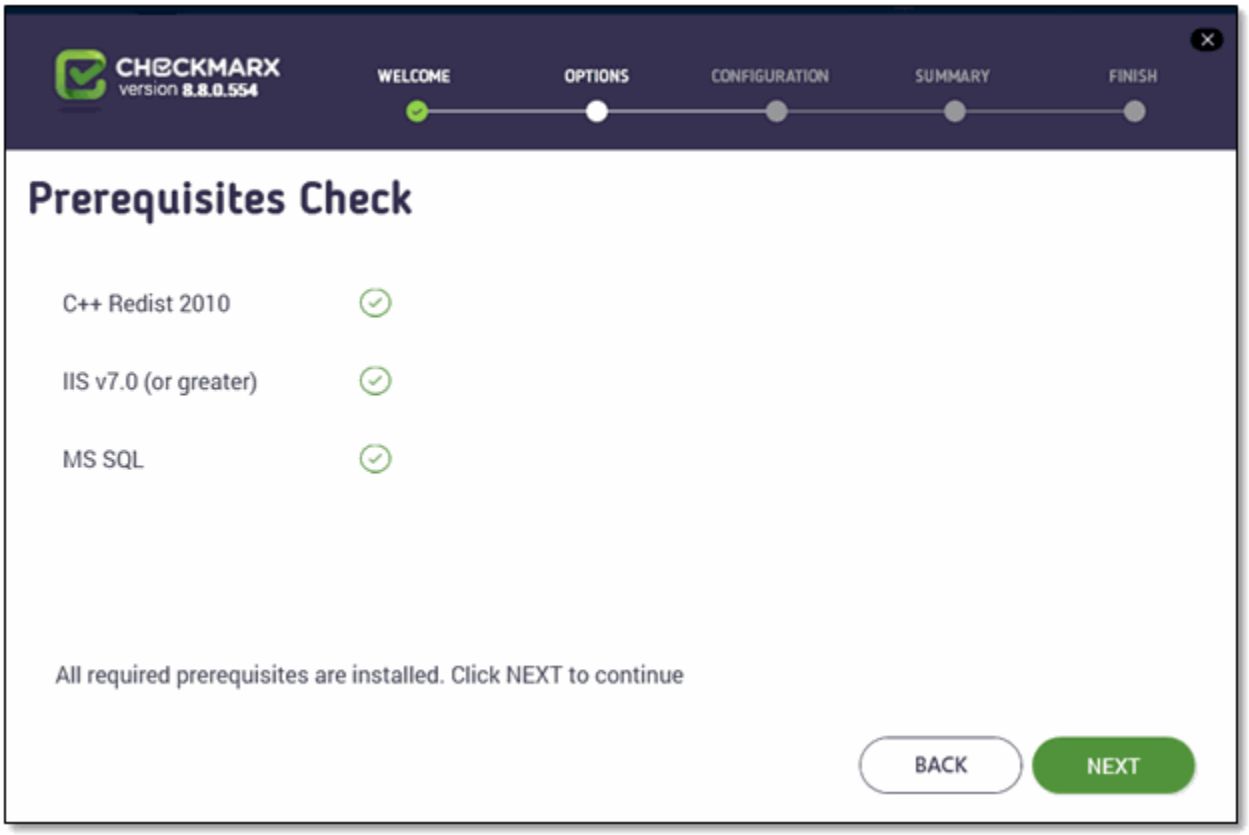
- POC/Evaluation - Select to install Audit, Engine, Manager, Application Risk Management and WebPortal
- Distributed Architecture
  - Select to install either Engine or Manager, Application Risk Management and/or WebPortal
- Centralized Architecture
  - Select to install Engine, Manager, Application Risk Management and WebPortal (select

  Audit, if you plan to create and customize queries on the host)
- CxEngine Server only - Select to install Engine (see Adding a CxEngine Server).

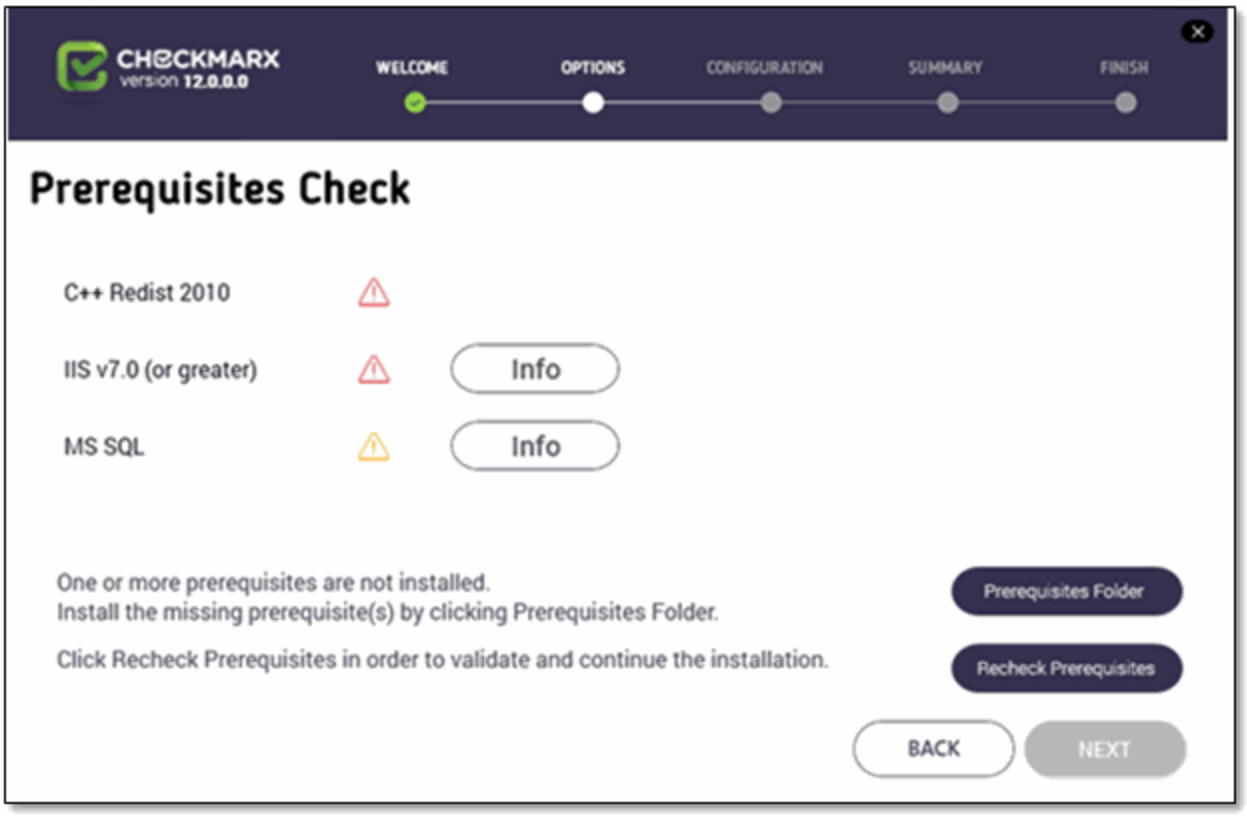
Install Application Risk Management

Checkmarx Application Risk Management (CxARM) – an application security risk management solution comprised of **CxARM Analytics** and **CxARM Policy Management** – for defining, tracking, evaluating and enforcing an organization's unified AppSec security policies, risks and status with a high level of visibility.

Click **Next.** The Prerequisites Check window is displayed, showing the status of all prerequisite components.

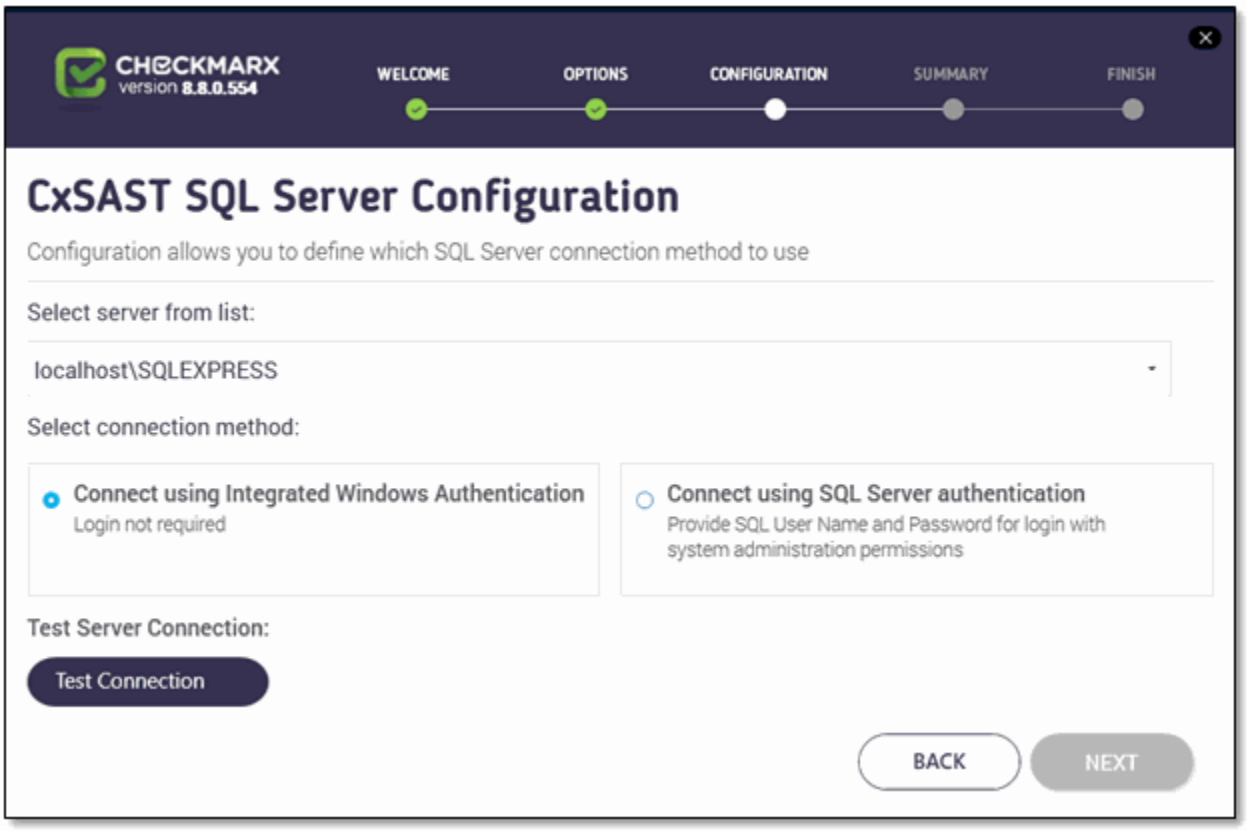


For any prerequisite not installed, click the respective **INFO** button for additional installation information, and then click **Prerequisites Folder** to install the missing component(s).



Click **Recheck Prerequisites** to confirm the installation status.

When all prerequisite components are installed, click **Next** to continue. The **CxSAST SQL Server Configuration** window is displayed.



For **CxSAST,** define a connection to the installed SQL Server or to any other SQL server on your network, by selecting one of the following:

- **Connect using integrated Windows authentication** (login not required)

- **Connect using SQL Server authentication** (provide SQL user name and password for login with SA permissions).

Click **Test Connection**. A "Connection OK" message is displayed upon confirmed connection to the SQL Server.

SQL Server Connection Failure

- If connection to the CxSAST SQL Server fails, a "Connection failure" message with the required action is displayed
- In order to continue with the installation, confirmed connection to the CxSAST SQL Server is required

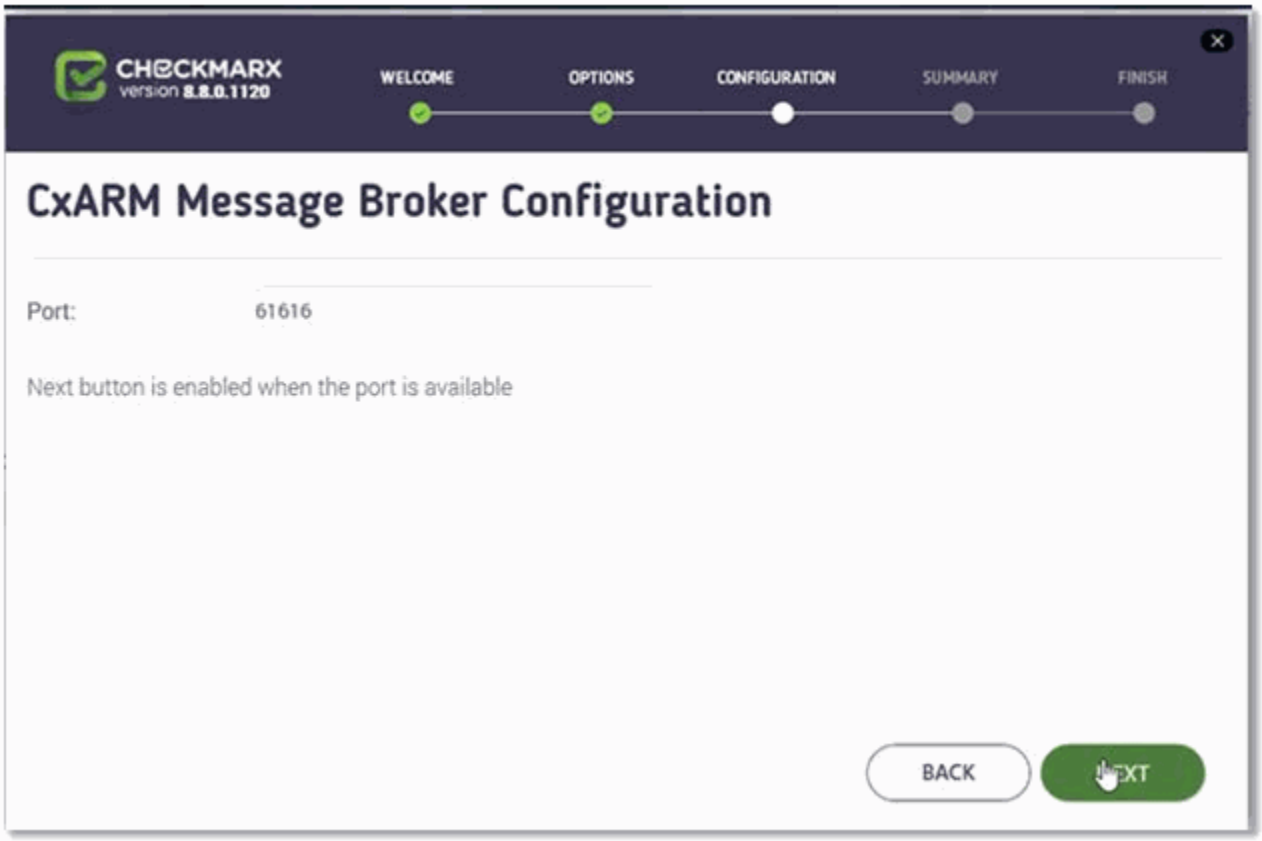
A notification displays if existing SQL Express files are detected.

Existing database detected

- To continue the installation using existing SQL Server databases (CxDB and CxActivity), click **OK**
- To perform a clean installation of SQL Server Express, click **CANCEL** and manually delete the existing CxDB and CxActivity databases

Click **OK** on the message, and then click **NEXT** to continue.

If installing CxARM, the **CxARM Message Broker Configuration** window is displayed.

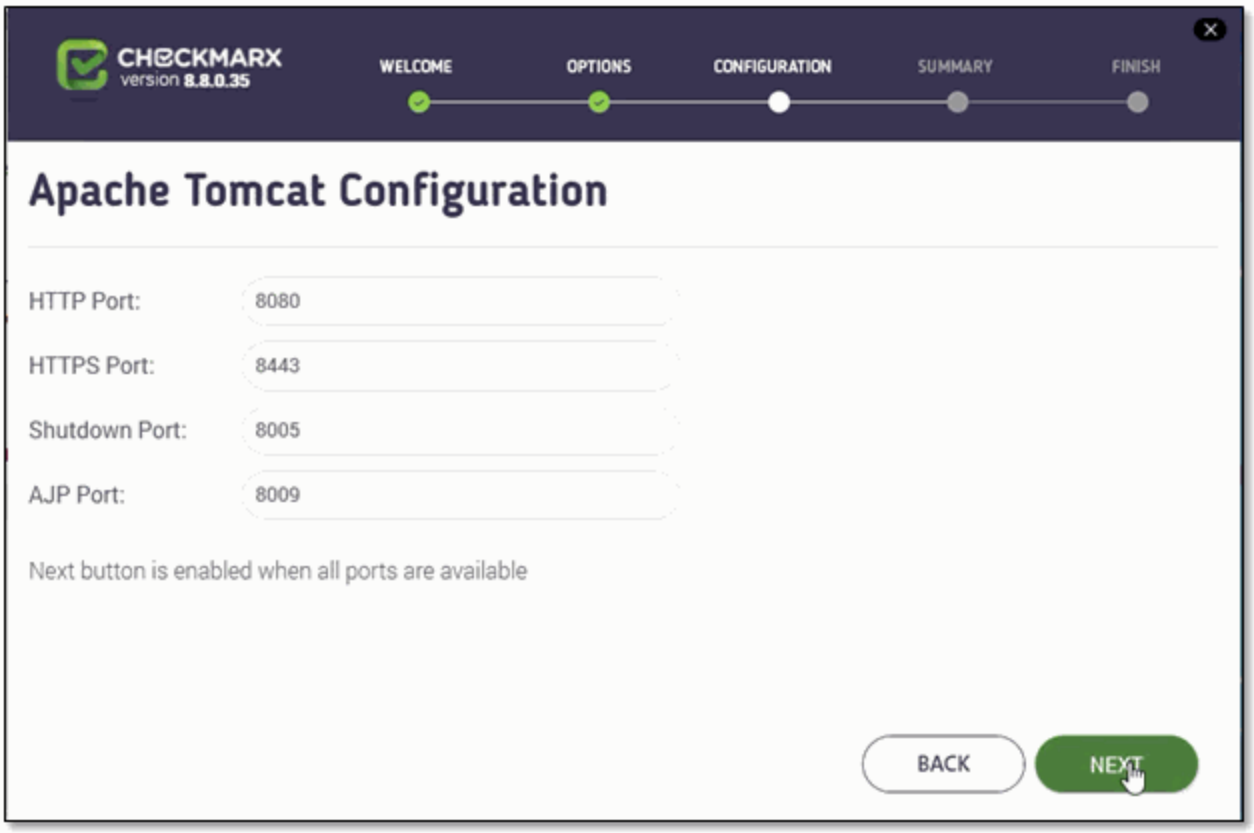


CxARM Message Broker Configuration

- Default port is 61616
- The NEXT button is enabled when the default port is available. If unavailable, define another available port.

Click **Next.**

If installing CxARM, the **Apache Tomcat Configuration** window is displayed.

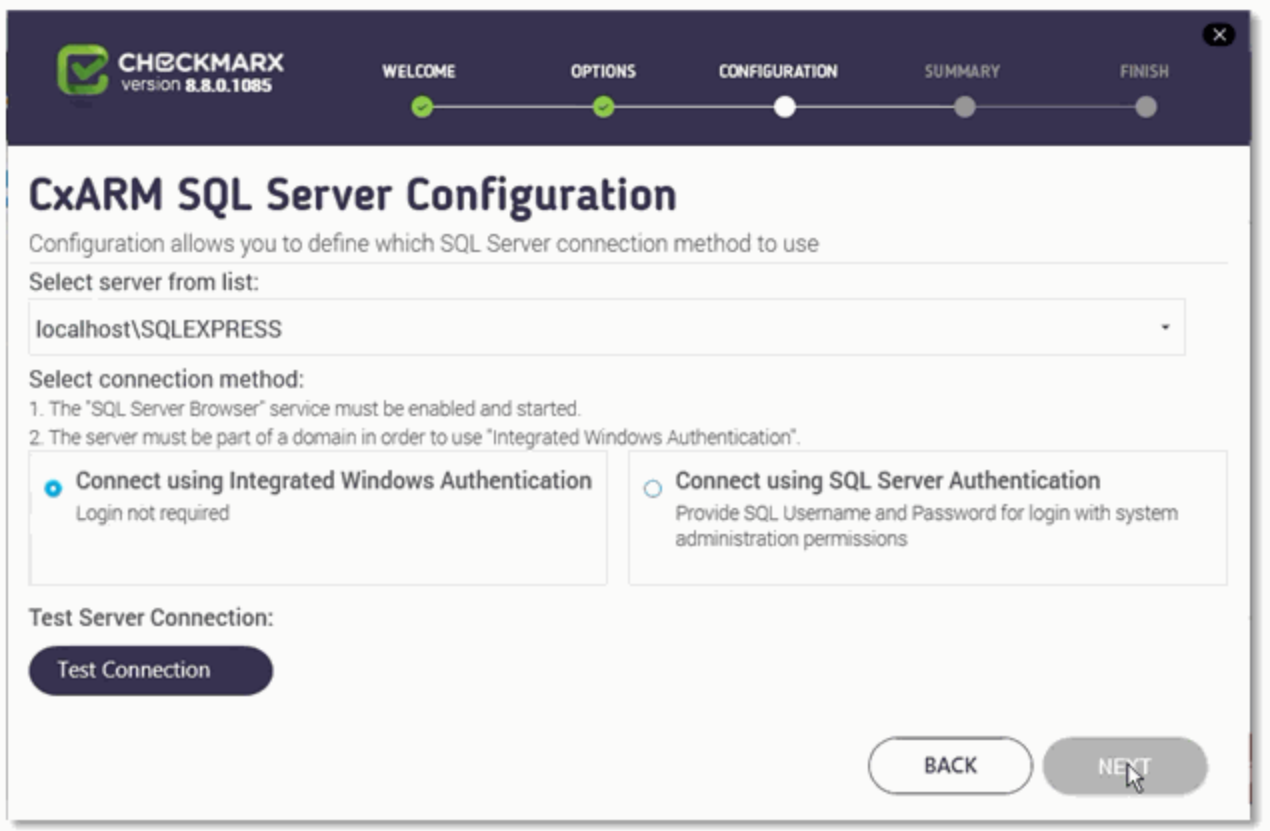


Apache Tomcat

- Default ports are displayed
- The NEXT button is enabled when the default ports are available. If unavailable, define another available port in the respective Port field.

Click **Next.**

If installing CxARM, the **CxARM SQL Server Configuration** window is displayed.



For **CxARM,** define the SQL Server connection by selecting one of the following:

- **Connect using Integrated Windows Authentication** (login not required)
- **Connect using SQL Server Authentication** (provide SQL user name and password for login with SA permissions)

Connection Requirements

For M&O Layer SQL Server connectivity, both **Dynamic** and **Static** port configurations are now supported. See Configuring Management & Orchestration SQL Server for Dynamic and Static Port Connectivity for additional information.

The following prerequisites and recommendations are required:

- For both connection methods the **SQL Server** and the **SQL Browser,** services must be enabled and started
- For the **Integrated Windows Authentication** method, the server must be part of a Windows domain

Click **Test Connection**. A "Connection successful" message is displayed upon confirmed connection to the SQL Server.

CxARM DB Connection Failure

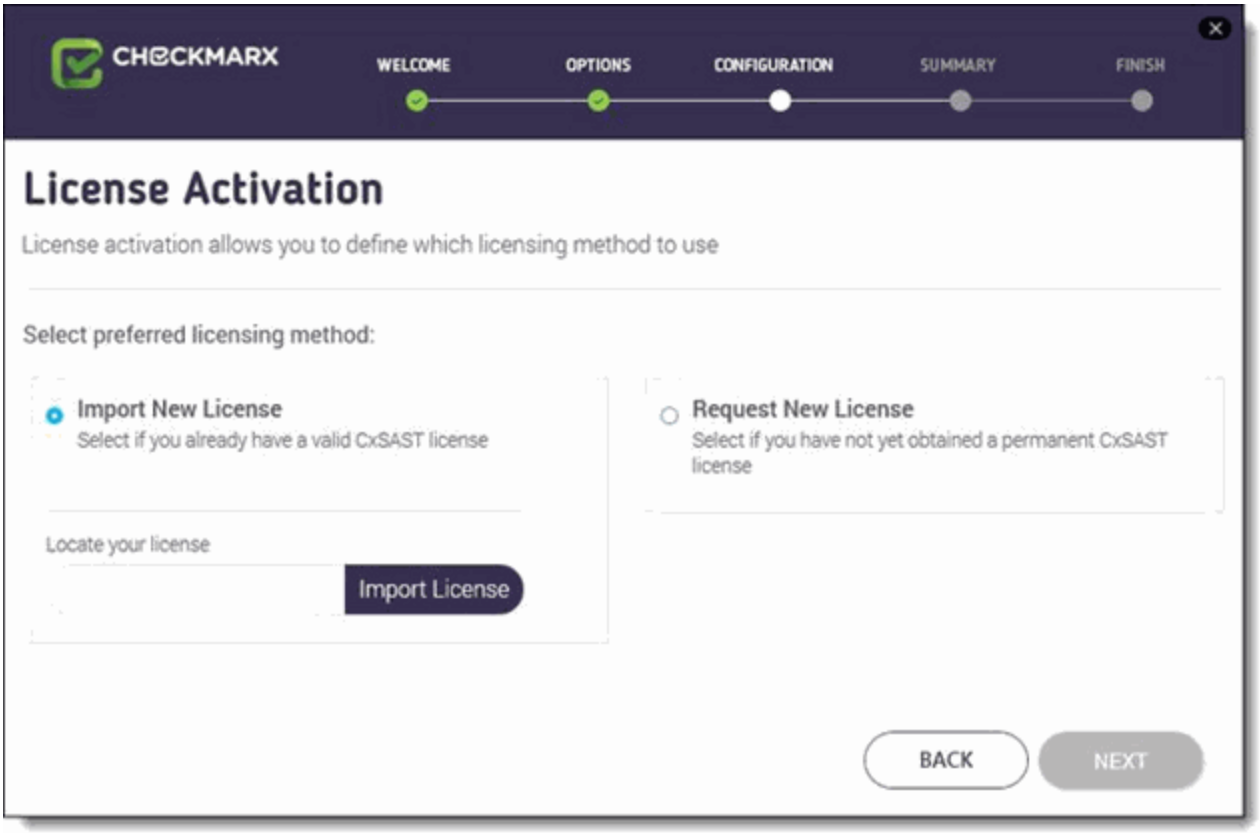
If connection to the CxARM database fails, in order to continue with the installation, a confirmed connection is required.

If the "SQL Connection Test Results" message indicates that connection to CxARM database has failed, verify the following:

- Host, port and login credentials are correct
- The CxARM machine is a member of a Windows domain (if not, either join the machine to a domain and perform a restart, or connect using SQL Server Authentication)
- The SQL Server Browser Windows service is running (if not, enable and start it)

Click **OK** on the message, and then click **NEXT.**

The **License Activation** window is displayed.

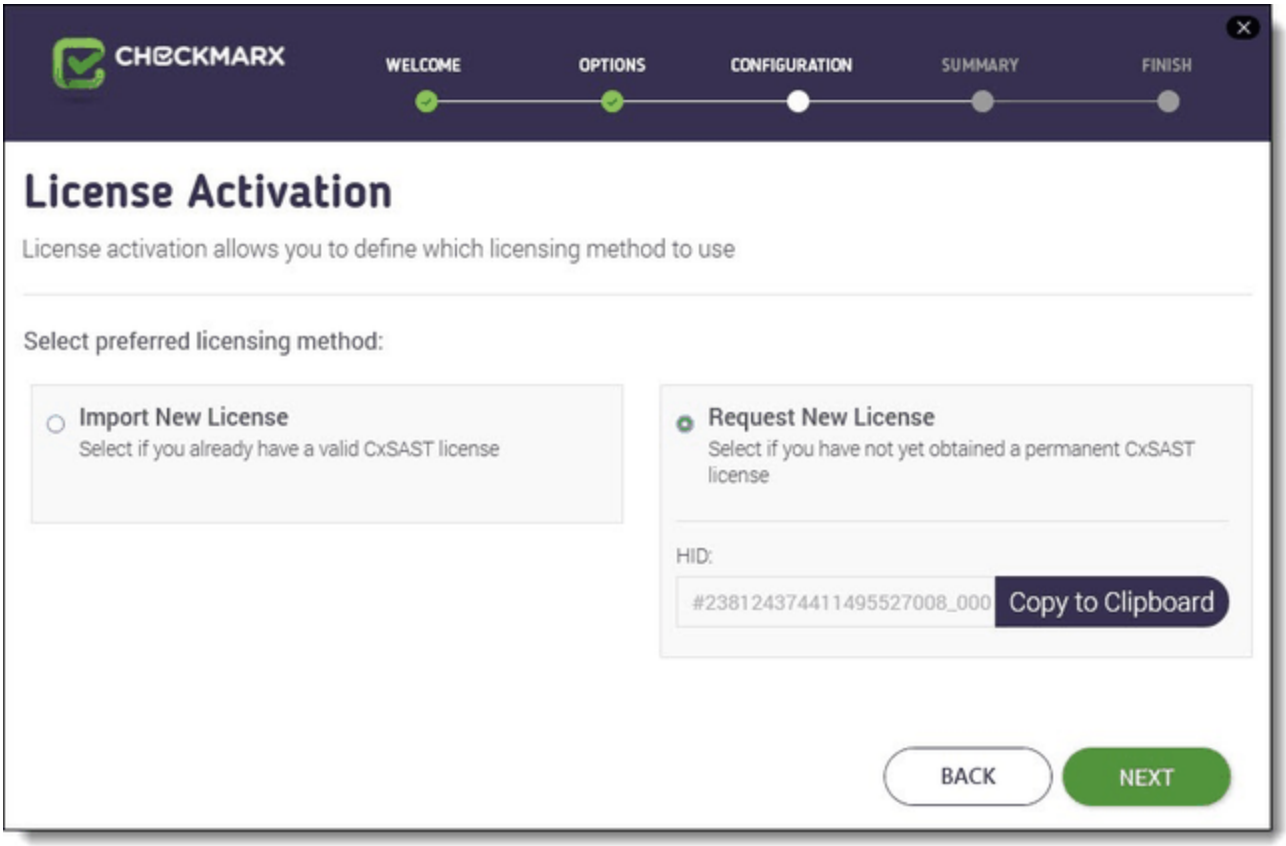


Upgrading an Existing License

For upgrades the license information (if it exists and is valid) is automatically loaded from the existing configuration and the License Activation window is not displayed.

Select the preferred licensing method by selecting one of the following:

- **Import new license:** If you already have a valid CxSAST license file, select the **Import New License** option and then click **Import License** Browse to the file location.
- **Request new license:** If you have not yet obtained a permanent CxSAST license. Select the **Request New License** option and then click **Copy to Clipboard**. Send the copied Hardware ID to your Checkmarx sales representative or contact Checkmarx support.



License Importer

Once you have obtained a new or updated Checkmarx license, you can use the license importer to import the license into CxSAST (see Updating the CxSAST License).

Click **NEXT** to continue.

HID Mismatch

If your license doesn't match your current hardware ID (HID) a warning message is displayed.
Please import a different license or request for a new one from your Checkmarx sales representative or contact Checkmarx support.

If the default port 80 is occupied, the **Validate Port** window is displayed.

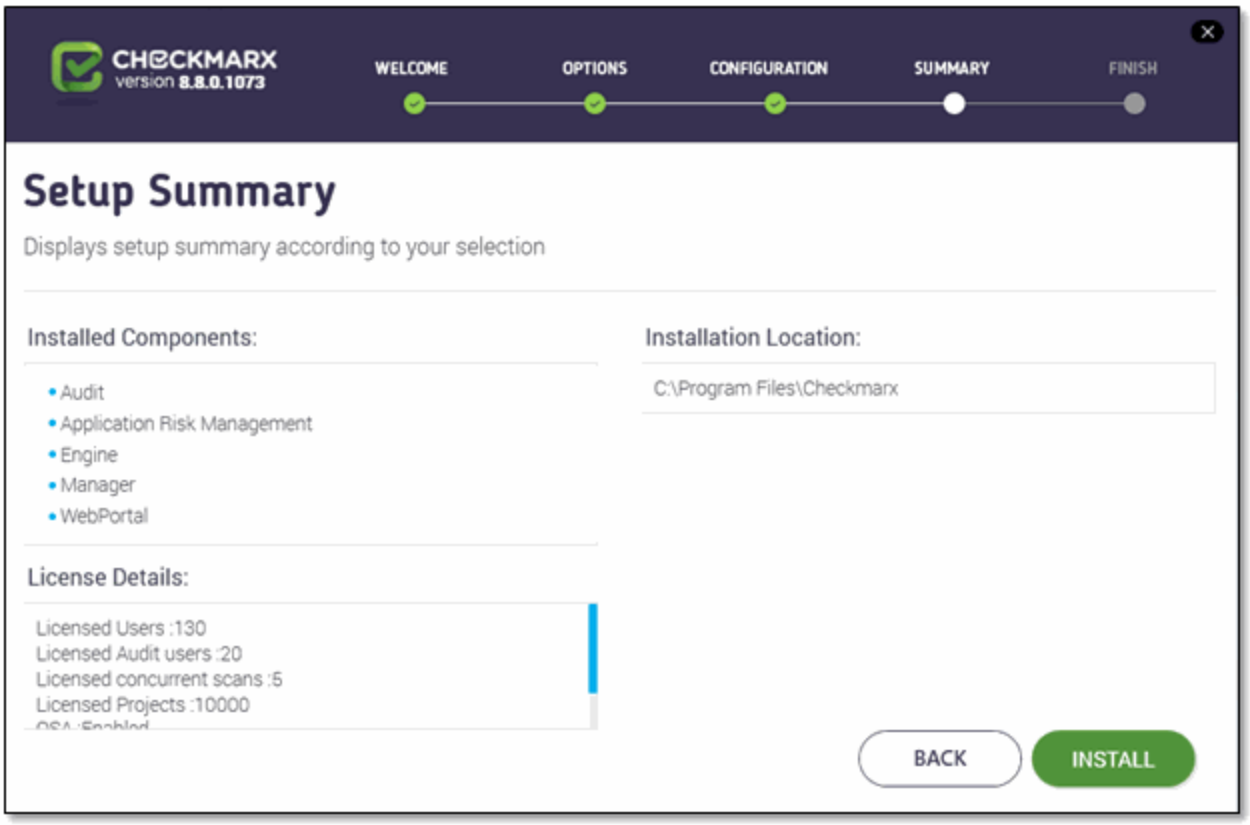
Default Port 80 Validation

Port 80 is allocated as the default port for Checkmarx applications. In clean installations the Validate Port window is displayed only if one of the following occurs:

- Port 80 is occupied by a non-default website or application
- Default website does not exist and port 80 is occupied by another application or website
- Default website does exist (occupies a different port) and port 80 is occupied by another application or website.

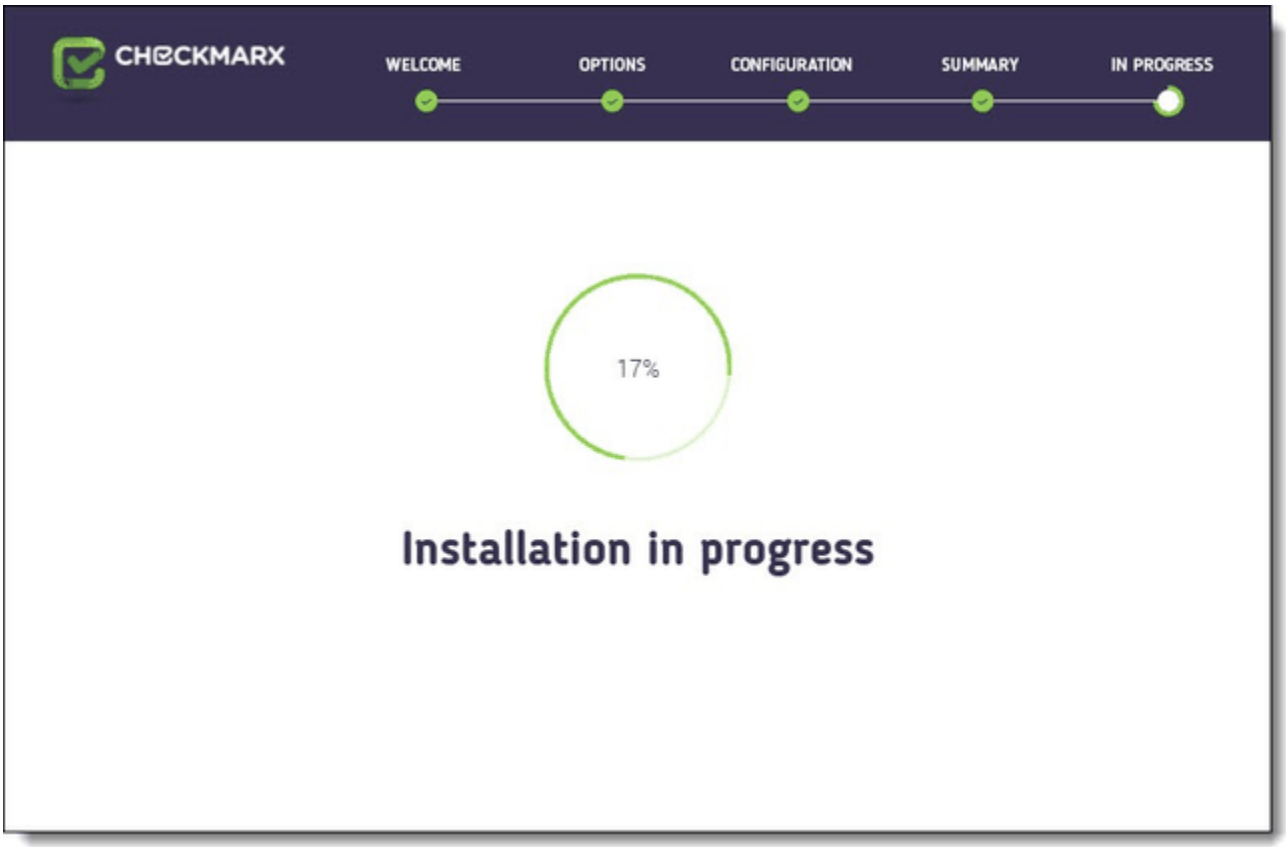
If required, select another port and click **Validate Port**.

Click **NEXT** to continue. The **Setup Summary** window is displayed.



Check the setup summary according to your selection.

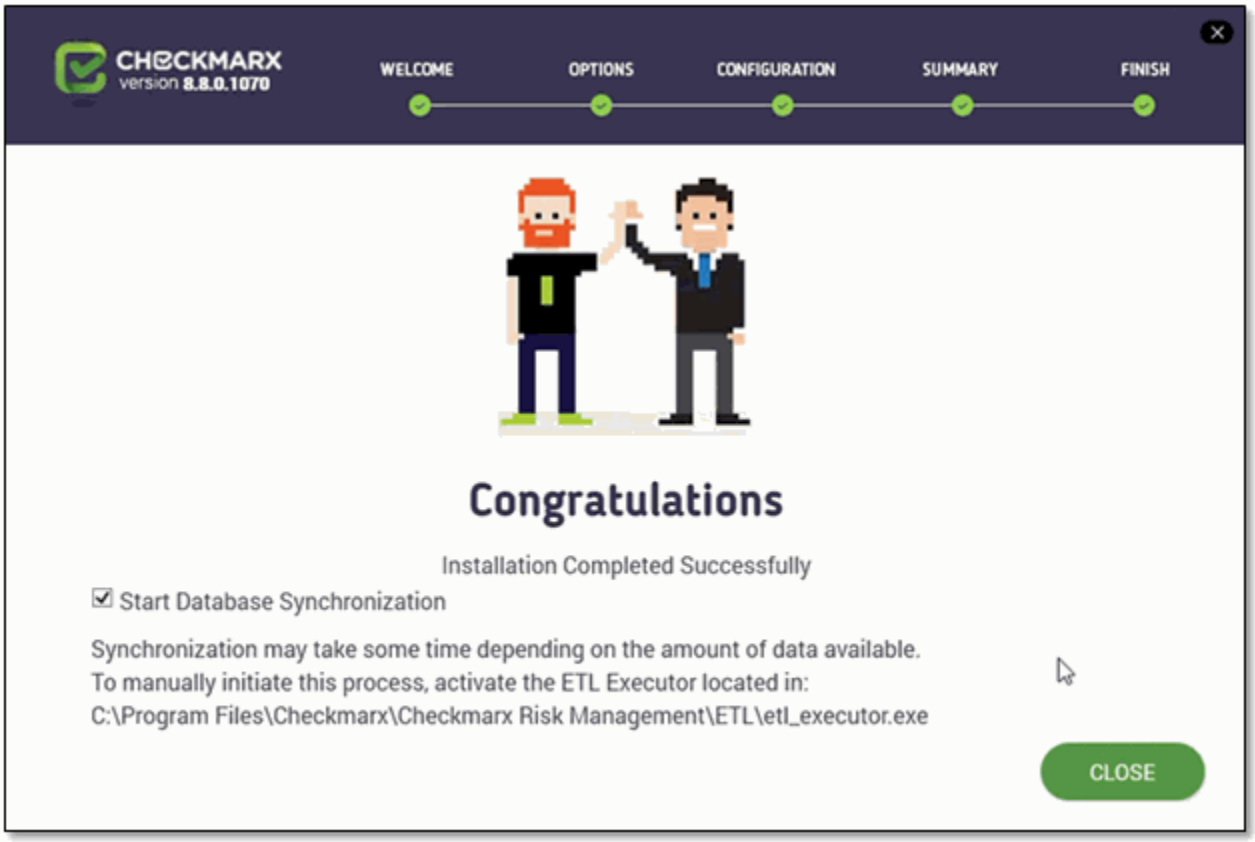
Click **INSTALL** to continue, **BACK** to return to the previous window, or **X** to exit. The **Installation in Progress** window is displayed.



Setup Failed

If the installation fails, the "**Setup failed**" message is displayed. For more information, see the installation logs. If you need further assistance, please contact Checkmarx support.

Once complete the **Installation Completed Successfully** window is displayed.



Start Database Synchronization

If you have installed Management and Orchestration, according to the Congratulations window, by default the **Start Database Synchronization** c heckbox is selected. This enables Management and Orchestration (CxARM) and initializes the automatic synchronization process that extracts data from the CxSAST database to the CxARM database. This process may take a while, depending on the amount of data being synchronized.

You can either perform the database synchronization now, or manually at a later time using the ETL Executor located in: **C:\Program Files\Checkmarx\Checkmarx Risk Management\ETL\etl_executor.exe**

NOTE: This folder may vary according to the selected Checkmarx installation folder.

For more information about Management and Orchestration prerequisites and recommendations, see Setting Up Management and Orchestration.

For more information about installing Management and Orchestration, see Installing Management and Orchestration.

Reinstalling CxSAST with an Already Existing CxARM DB

If attempting to install CxSAST with CxARM and connect to an existing CxARM DB, the subsequent ETL DB sync will fail, due to a limitation in CxARM. Therefore, in order to reinstall CxSAST with CxARM, either delete the existing CxARM DB before reinstalling, or reinstall with a new CxARM DB.

To continue now with the database synchronization:

Leave the checkbox selected, and then click **CLOSE.** If required, reboot the server (you will receive a prompt if rebooting is necessary). The database synchronization process starts automatically.

To perform the database synchronization at another time:

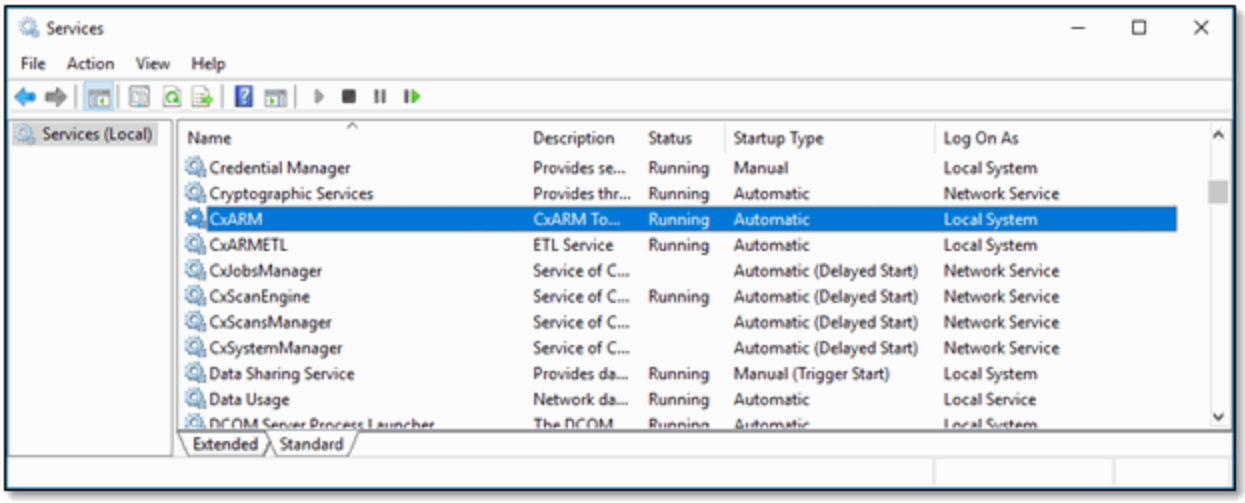
Alternatively, you can manually initiate the synchronization process at a later time by clearing the checkbox now, and clicking **Close.** At a later time use the ETL tool to perform the synchronization, located at: **C:\Program Files\Checkmarx\Checkmarx Risk Management\ETL\etl_executor.exe**

NOTE: This folder may vary according to the selected Checkmarx installation folder.

For more information on Application Risk Management, see Installing CxARM.

Installed Services Check

Go to **Start** > **Control Panel** > **System and Security** > **Administrative Tools** > **Services**



The database (DB) is required to be up and running in order for Checkmarx services to be able to run.

Make sure the following installed Checkmarx services are started:

On a centralized host:

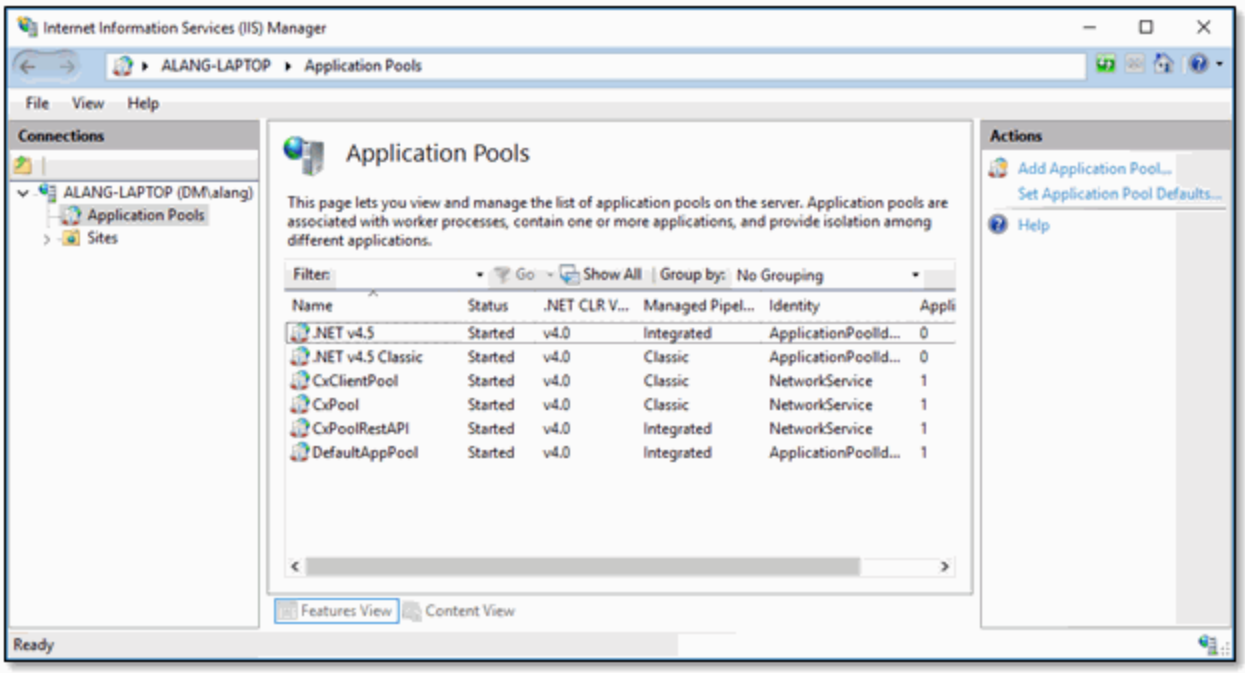
- **CxJobsManager**
- **CxScansManager**
- **CxSystemManager**
- **CxScanEngine**
- **Web Server:**
  - **IIS Admin Service**
  - **World Wide Web Publishing Service**
- **Application Risk Management:**
  - **CxARM**
  - **CxARMETL**

On a CxEngine host:

- **CxScanEngine**

Installed Application Pool Check

Go to **Start** > **Control Panel** > **All Control Panel Items** > **Administrative Tools** > **Internet Information Services (IIS) Manager**



Make sure the following installed application pools are started:

On a centralized host:

- **CxClientPool**
- **CxPool**
- **CxPoolRestAPI**

If the IIS Pools are not started automatically after installation, you should restart the machine.

Enable Long Path Support in CxSAST Application

.NET framework 4.6.2 and above supports the Long Path feature by default. The following actions should be taken in order for the Long Path feature to be enabled.

The following configuration should be added to the Web Service and REST API:

<httpRuntime targetFramework="4.6.2" />

The *web.config* file is usually located in the following path: *c:\Program Files\Checkmarx\Checkmarx Web Services\CxWebInterface\web.config*

For example:

```
<system.web>
    <httpRuntime targetFramework="4.6.2" />
    <compilation targetFramework="4.5.1" debug="true"/>
</system.web>
```

If the httpRuntime already exists, add the targetFramework attribute as follows:

<httpRuntime maxRequestLength="2097151" executionTimeout="36000" **targetFramework="4.6.2"** />

Keep in mind that this configuration should only added on a machine that has .NET 4.6.2 or above installed, otherwise there will be issues in the application.

Login to the Web Interface

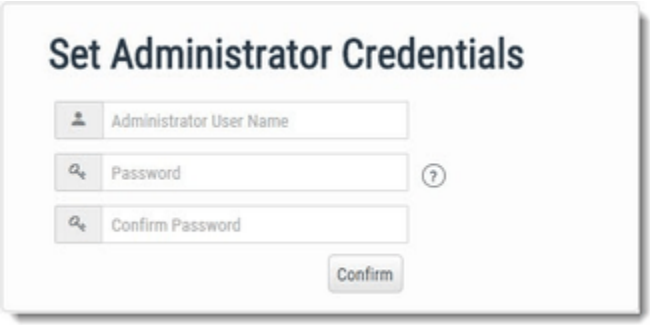
Access the CxSAST web interface in either of the following ways:

- Access CxSAST locally (from the server host) by using the **Checkmarx Portal** shortcut on the Desktop or navigate to the Checkmarx folder (**Start** > **All Programs** > **Checkmarx** > **Checkmarx Portal**).
- To access CxSAST from any other computer, make sure that organizational routing and firewall configuration allow the client computer to access the CxSAST server. Point your browser to: http://<server>/cxwebclient/login.aspx where <server> is the IP address or resolvable hostname of the CxSAST server.

Upon a fresh installation, a single Administrator Account needs to be created.

Once the Set Administrator Credentials window is displayed, add the following credentials:

- **Administrator User Name**
- **Password**
- **Confirm Password**



Password Complexity

The required password complexity is as follows: 9 to 400 characters, at least 1 uppercase letter, at least 1 lower case letter, at least 1 special character, at least 1 digit.

Click **Confirm** to complete.

You can subsequently change the Administrator password and add CxSAST users.

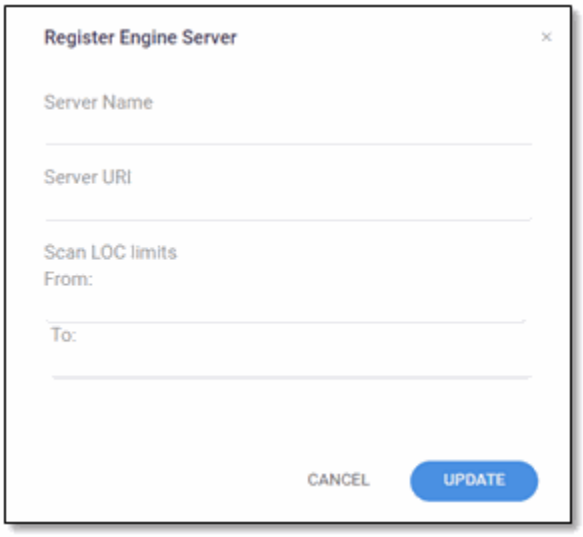
In a distributed architecture:

Go to **Management** > **Application Settings** > **Engine Management.** The Engine Management window is displayed.

Click **Register Engine Server**. The Register Engine Server window is displayed.

Give the Engine a **Server Name**, and provide the **Server URL**, so that CxManager will be able to communicate with CxEngine. The URL should be: **http://<Server_Name>/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc**
(where <Server_Name> is the CxEngine host's IP address or resolvable name).

Optionally define **Scan LOC Limits** (maximum lines of code allowed).



URL Check

It is recommended to check the defined URL by opening it in a browser on the CxManager Server to validate.

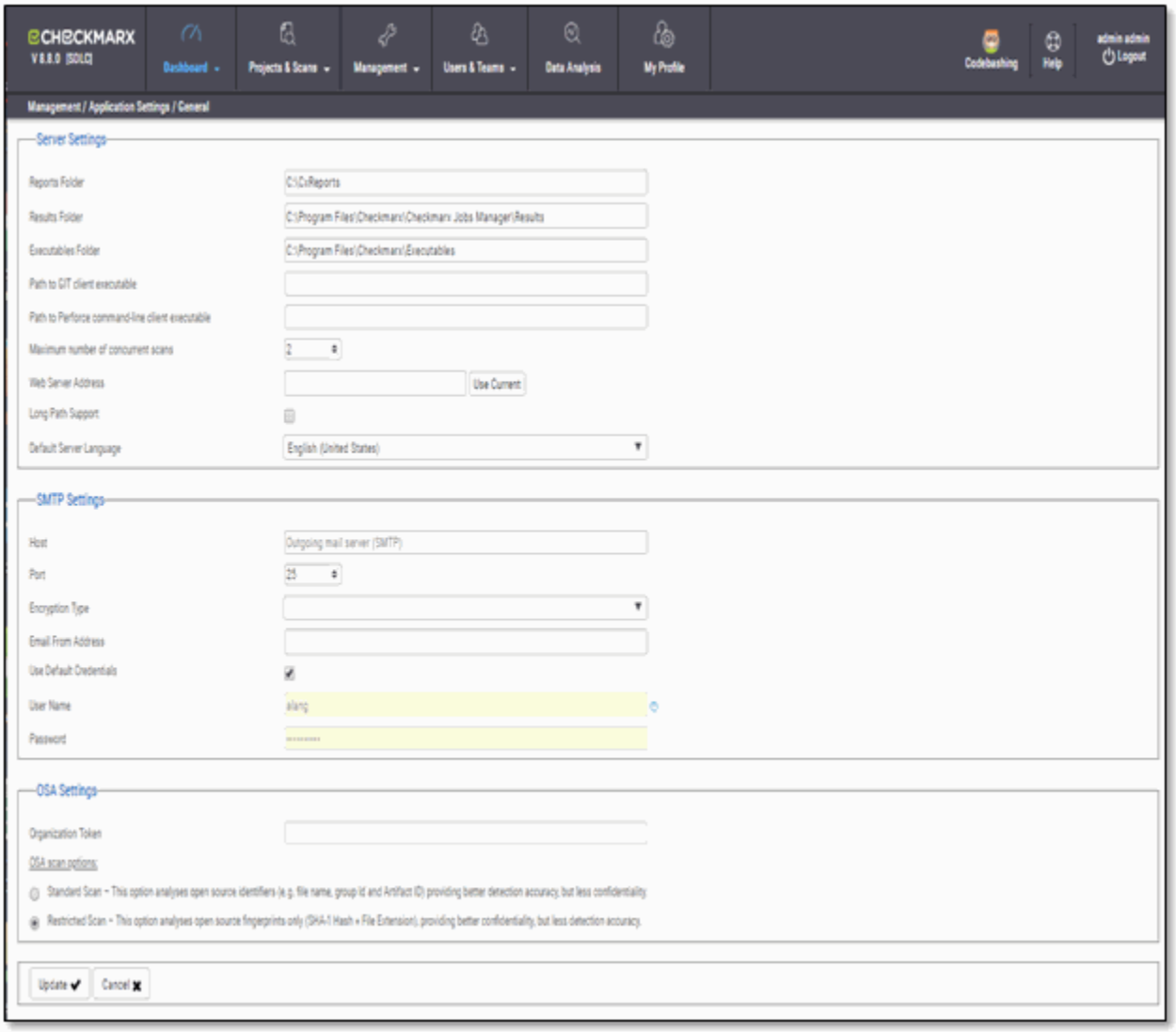
Click **Update**.

Multiple CxEngine Servers:

If you have multiple CxEngine Servers, repeat the above step for each one.

Go to **Management** > **Application Settings** > **General**.

After updating the information, at the bottom of the page, click **Update:**



Server Settings

If permitted by your CxSAST license, set the “Maximum number of concurrent scans“ to the desired number for all the CxEngine Servers.

Enable Long Path Support in Server Settings

In order for the long path feature to be fully supported in CxSAST, click **Edit** and check the **Long Path Support** checkbox.

Long Path Support

Click **Got It** on the message window to confirm your understanding that all application servers must support long paths, otherwise scans with long path files may fail.

Click **Update** to save the changes.

SMTP Settings

Provide **SMTP** settings. Other settings should usually be left as they are. Optionally, you can configure the "From" field of emails. If you don't configure it, it will be left empty.

Click **Update** to save changes.

OSA Settings

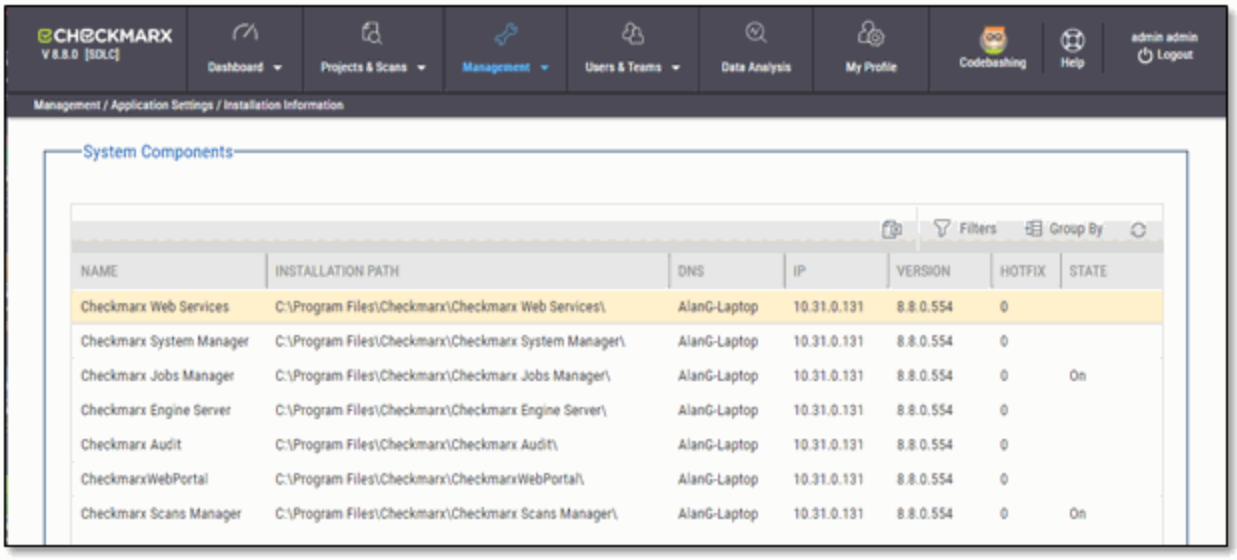
If licensed for CxOSA, select the OSA (Open Source Analysis) scan option and click **Update.**

Email Verification

Verify that the email address in the CxSAST profile settings (My Profile > Account Information) is of a valid format, i.e. John.Smith@example.com, and not John.Smith@example. This is required for AppSec Coach registration.

Installation Verification

Go to **Management** > **Application Settings** > **Installation Information.**



Validate that you have successfully installed the correct version and/or hot-fix and review all CxSAST system components ensuring that they are all of the same version.

Configuring Quality Gates

A Quality Gate is the best way to enforce a quality policy and it's there to answer one simple question; can I deliver my project to production or not? In order to answer this question, you must define a set of conditions based on measurement thresholds against which your projects are measured.

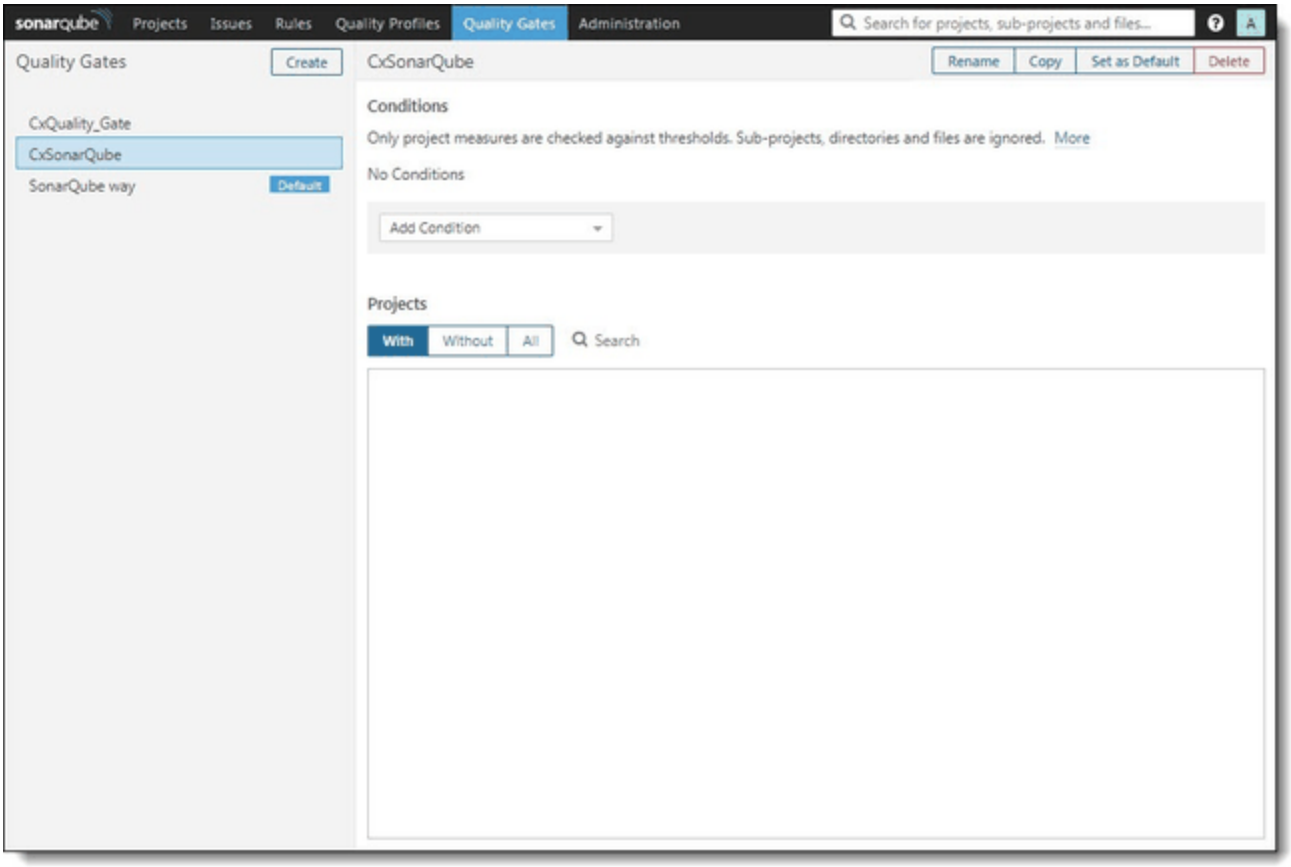
The quality gate "SonarQube way" is provided by SonarQube and activated by default. It is also possible to set a default quality gate, which can be applied to all projects.

Defining a Quality Gate

To create a new quality gate, refer to the SonarQube Documentation – Quality Gates.

To define an existing quality gate, click **Quality Gate** from the menu bar.

Once the **Quality Gate** page is displayed, select the desired quality gate, in this case "CxSonarQube", as seen below.



Click the **Add Condition** drop-down menu and select a condition from the list. The new quality gate condition row is displayed in the Conditions list.

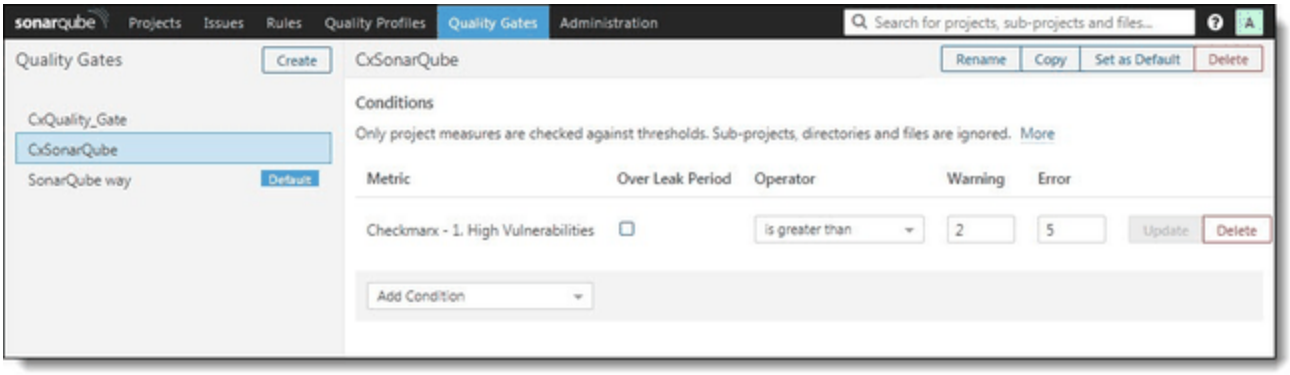
Define the new condition according to a combination of the following:

- **Metric**: Measurement, e.g. Checkmarx High Vulnerabilities
- **Period**: Value (to date) or Leak (differential value over the Leak period)
- **Operator**: Comparison, e.g. is greater than (>)
- **Warning**: Value (optional), e.g. 2
- **Error**: Value (optional), e.g. 5

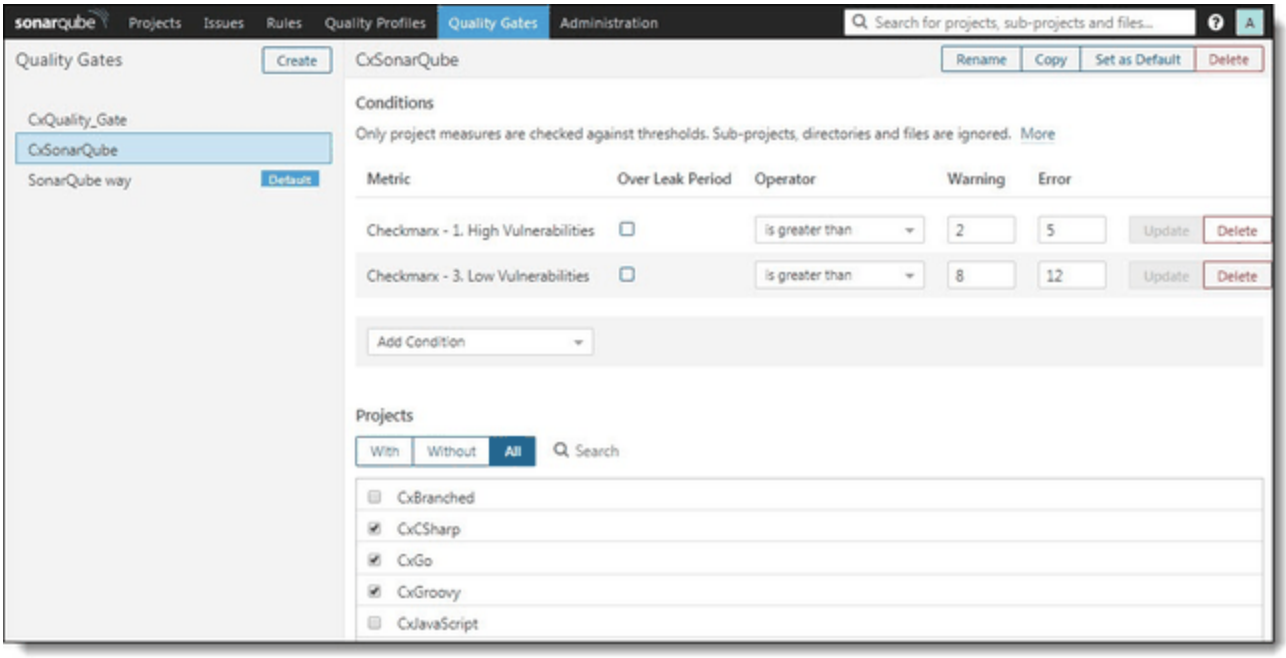
Example:

If there are **more than 2 High Vulnerabilities**, issue a **Warning**. If there are **more than 5 High Vulnerabilities**, it becomes an **Error**.

Once defined, click **Add** to save the condition.

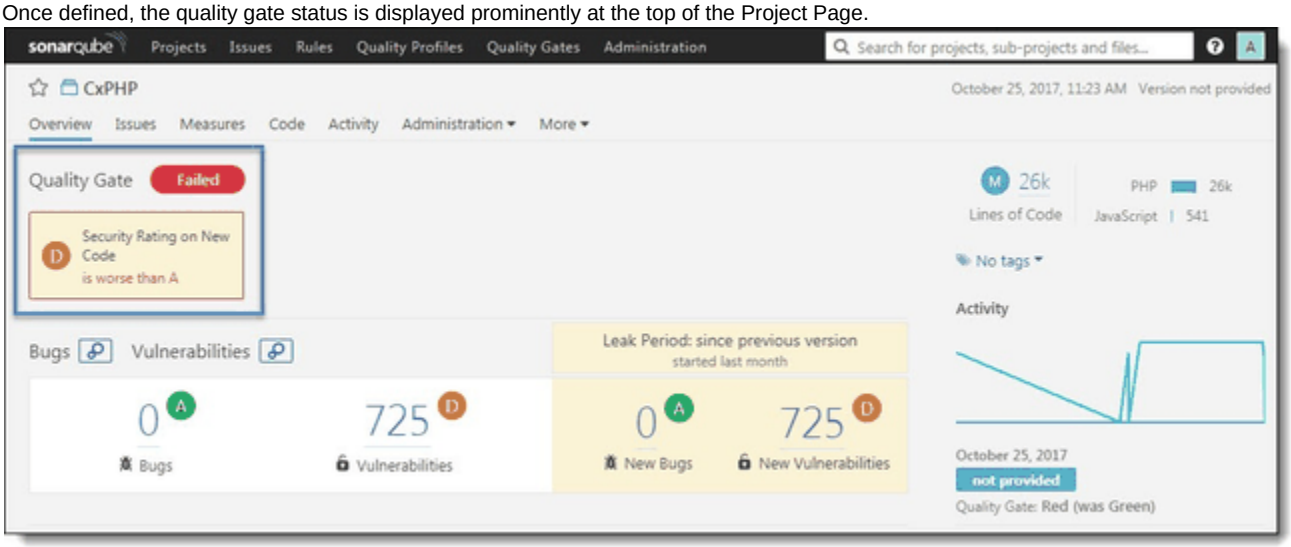


Once you have defined any remaining conditions, click **All** and select the Project(s) to which your defined Quality Gate condition apply.



You can also set your quality gate conditions as default for all projects by clicking the **Set as Default** option.

Once defined, the quality gate status is displayed prominently at the top of the Project Page.



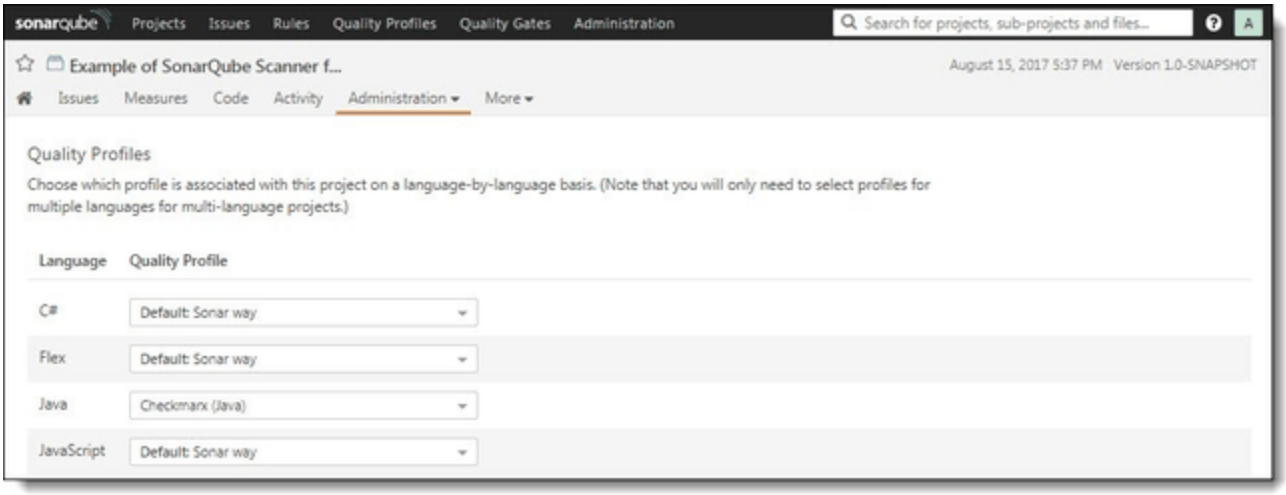
Configuring Quality Profiles (v8.5.0)

Quality Profiles are central to SonarQube, since this is where security related rules are defined and assigned to projects. For more information about Quality Profiles, refer to the SonarQube Documentation – Quality Profiles. Quality profiles for Checkmarx are automatically created by the Checkmarx SonarQube plugin, therefore security rules are already predefined. For more information about rules, please refer to the SonarQube Documentation - Rules.

Assigning a Checkmarx Quality Profile to a Project

Checkmarx Quality profiles and their predefined rules can be assigned to projects.

From within a specific project in the **Project** screen, click the **Administration** tab and select **Quality Profile**. The **Quality Profile** screen is displayed.



Click the relevant **Language** and select the desired **Quality Profile** from the list, in this case "**Checkmarx (Java)**".

You need to assign a quality profile to the project for each language that your project supports.

Pricing and Cost Advice

Checkmarx is expensive. It is priced per developer with a rough estimate of 12 Developers for $59k USD per year or 50 Developers for $99k USD per year. Checkmarx uses Whitesource for dependency scanning and charges an extra $12k USD per year for this open source scanning.

Be cautious of the one-year subscription date. Once it expires, your price will go up.We got a special offer for a 30% reduction for three years, after our first year. I think for a real source-code scanning tool, you have to add a lot of money for Open Source Analysis, and AppSec Coach (160 Euro per user per year).

Before implementing the product I would evaluate if it is really necessary to scan so many different languages and frameworks. If not, I think there must be a cheaper solution for scanning Java-only applications (which are 90% of our applications).

CxSAST Quick Start (v8.9.0 and up)

This Quick Start includes information on setting up first project scans and an overview of presets.

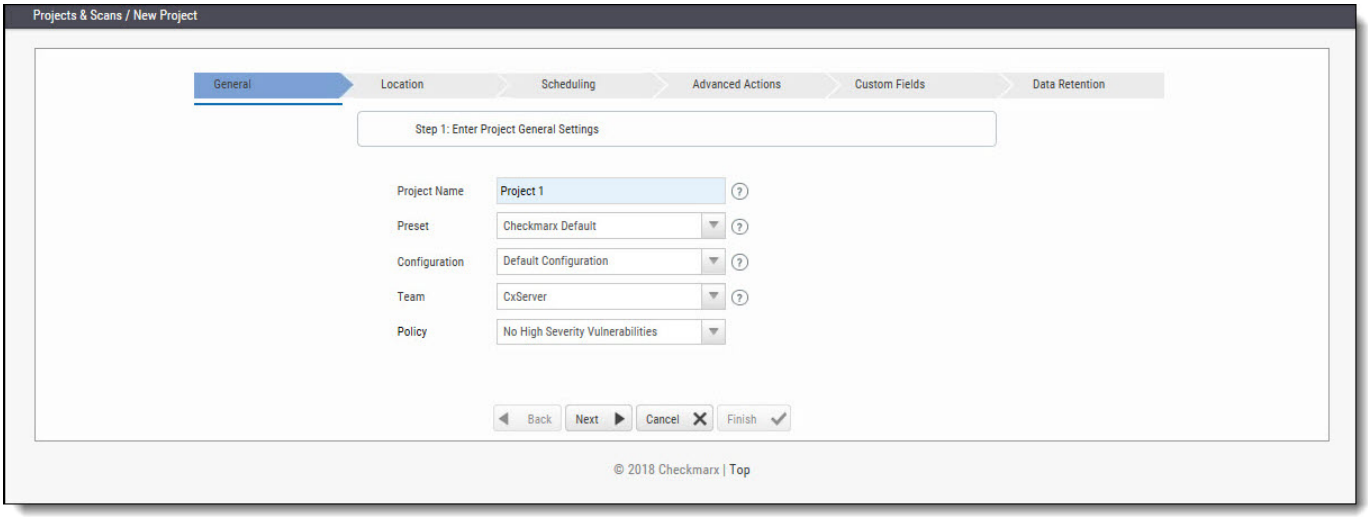
Setting Up

In the **Projects & Scans > Create New Project** window perform the following procedure:

Step 1: Enter Project General Settings

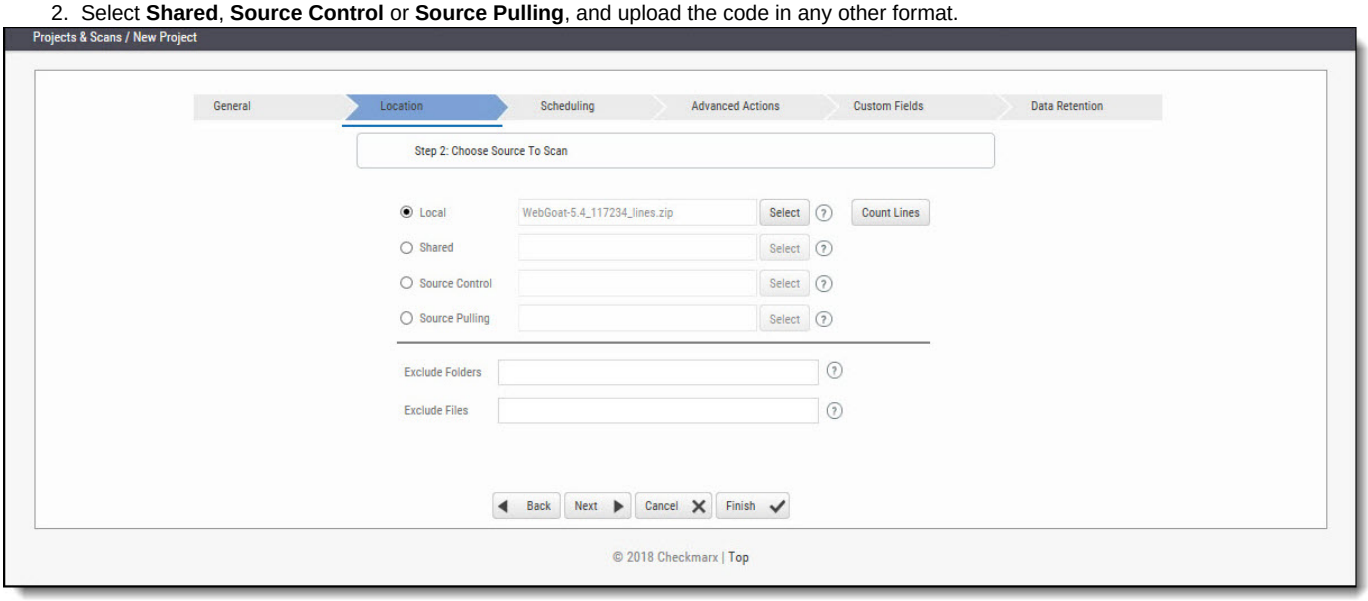
1. **Project Name**: Provide an appropriate Project Name for the project.
2. **Preset**: The Preset will determine the scan rules for the project. Select the appropriate scanning Preset from the drop-down list.
3. **Configuration**: Select the Configuration for the new project. For the trial version, it is advised to perform the default selection.
4. **Team**: Select the Team for the new project. For the trial version, it is advised to perform the default selection.
5. **Policy:** Select a policy for the project. For the trial version, it is advised to select from the default selection.

It is advised to leave the fields **Configuration** and **Team** unchanged in the trial.



Step 2: Select Source To Scan

1. Select **Local** to upload code as a ZIP file. The code must be zipped by MS zip. The test account is limited to 350,000 Lines of Code (LOC).
2. Select **Shared**, **Source Control** or **Source Pulling**, and upload the code in any other format.



Note that you can scan the "**OWASP Benchmark Project**" code; go to https://github.com/OWASP/benchmark , click the **Clone or download** butt on and select your preferred option.

1. Other sample code for scanning include:

Bookstore.Net; Bookstore.Java; Bookstore.php4; WebGoat5.0; WebGoat6.0; CPP Example; iGoat; Samples; Android.

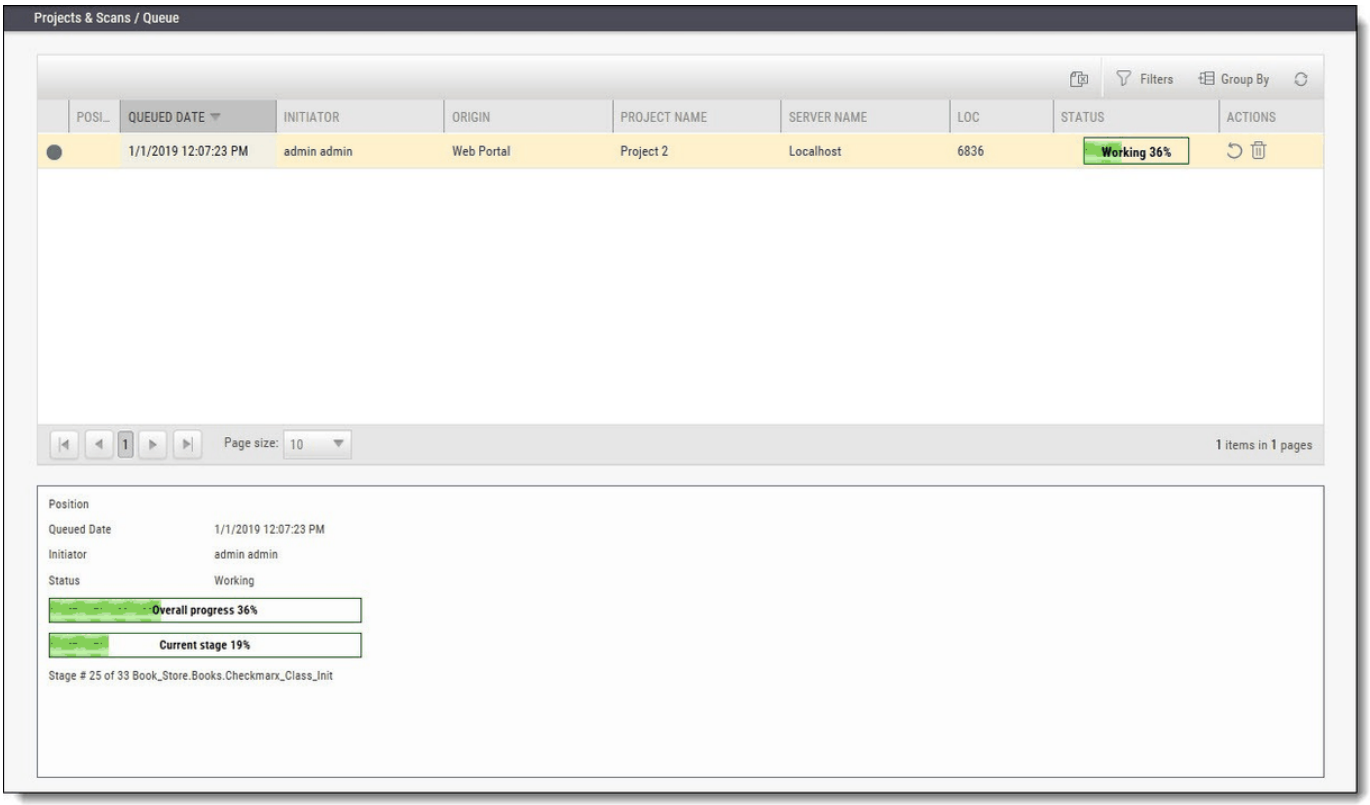
1. If using a Browser/ Eclipse/ Visual Studio/ IBM RAD, please start with the browser option.

1. When the Finish button becomes active, click

Finish to place the project into a queue.

Step 3: Scan Execution

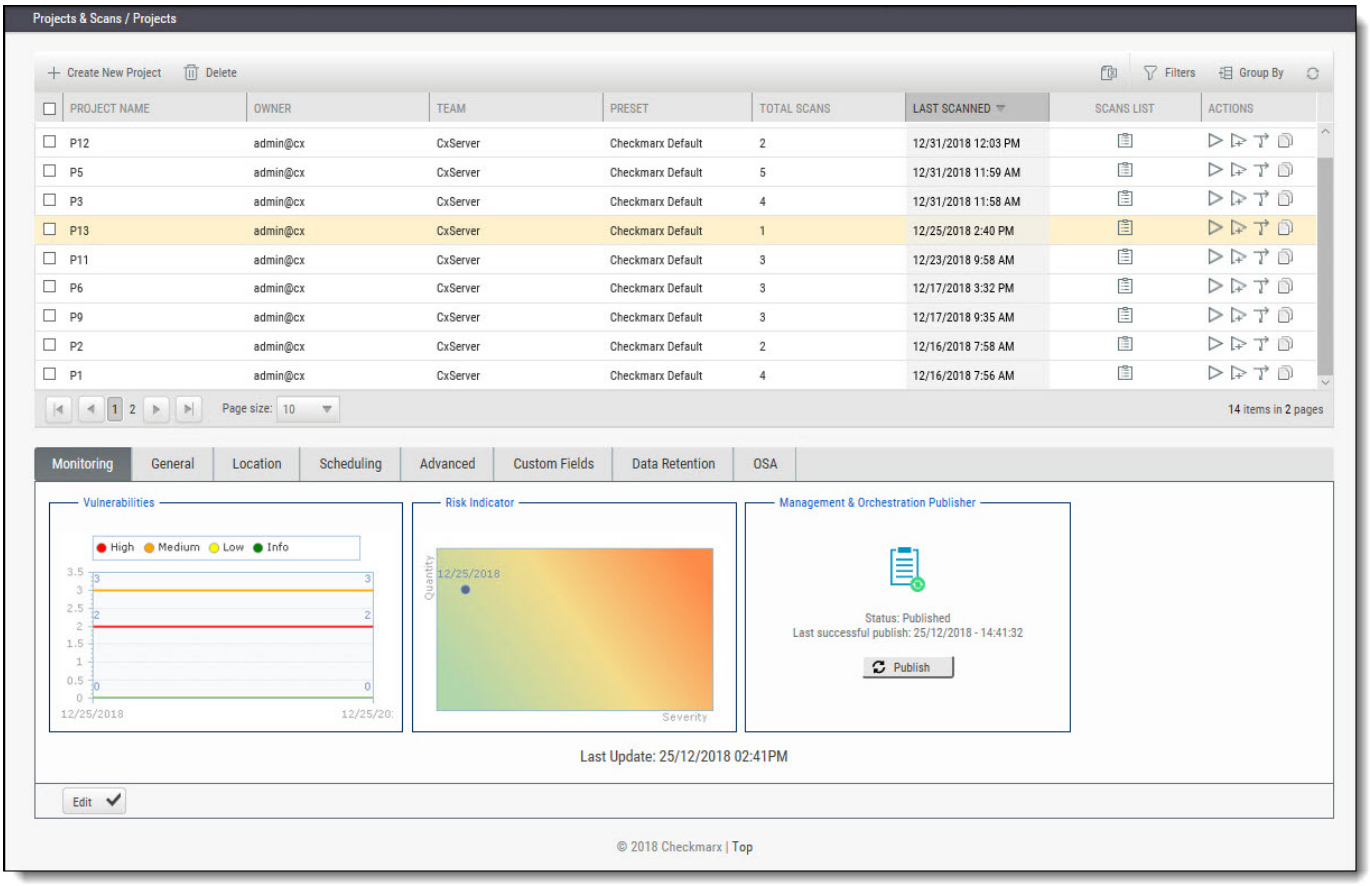
- In **Projects & Scans > Queue**, monitor the scan progress by clicking the project line in the queue table.



Reviewing Scan Results

Step 1 – Projects & Scans

- In **Projects & Scans > Projects**, click Scans List to view the high level summary of scan results and account activity.



For more information on Dashboards **click**.

Step 2 – Review Scan Results in the Source Code

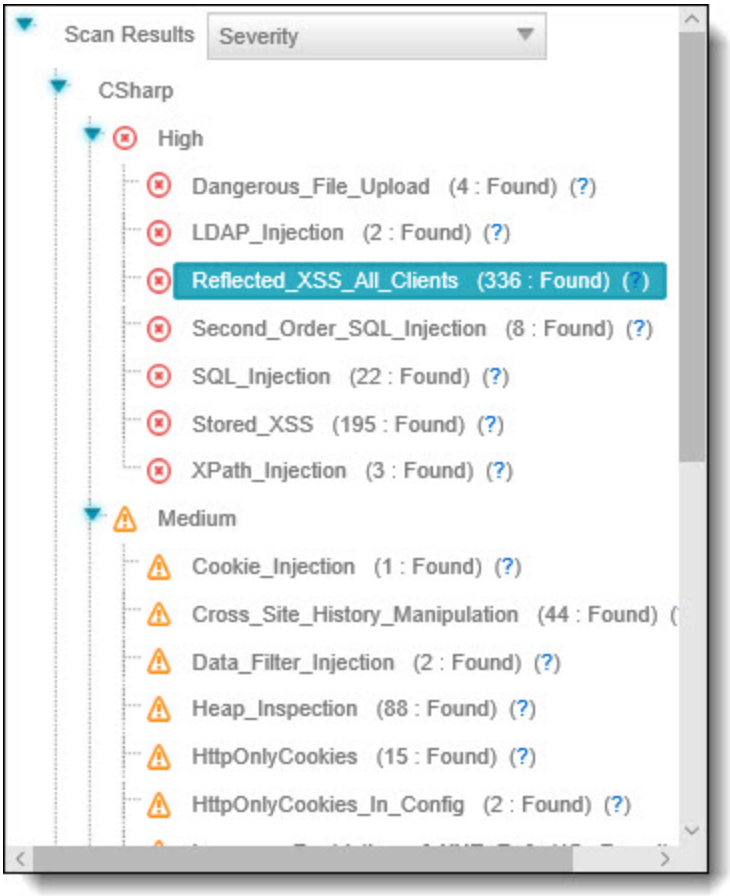
View detailed scan results within the Source Code. Vulnerabilities and navigated attack path are highlighted.

The View Results page is divided into four (4) sections:

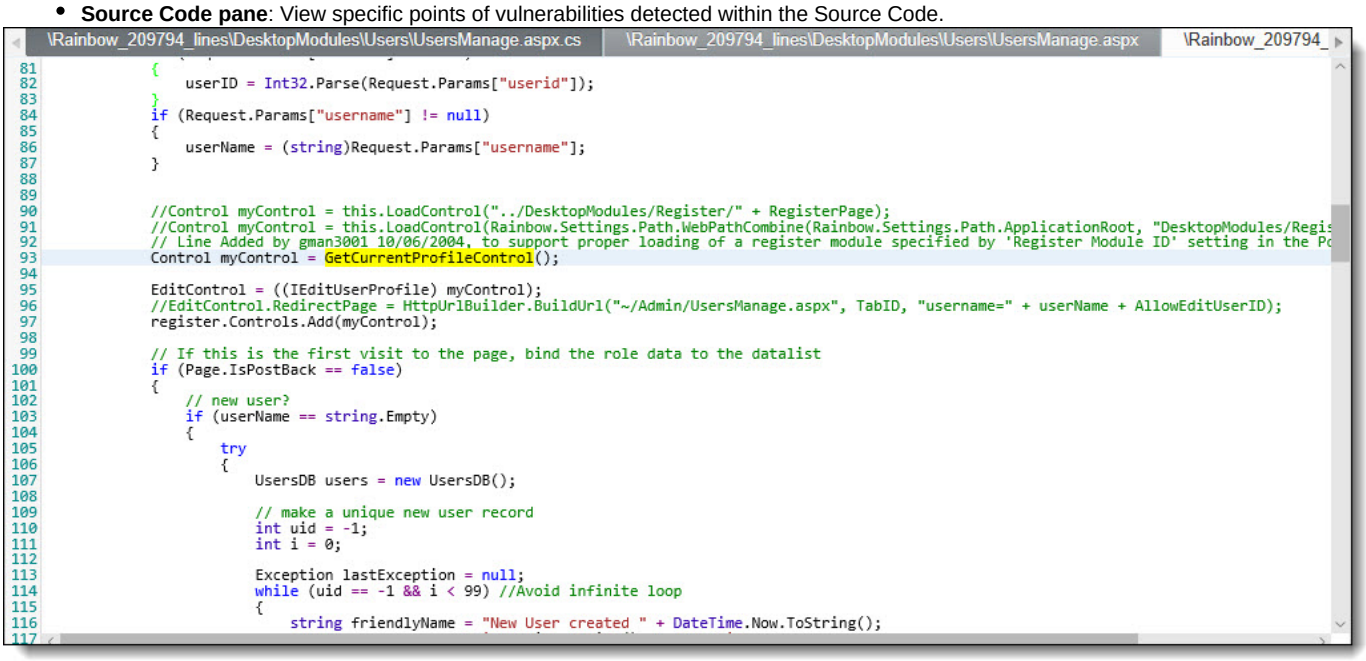
- Scan Results Summary by vulnerability,
- Results table or Graph,
- Attack Vector
- Source code

Scan Result Summary

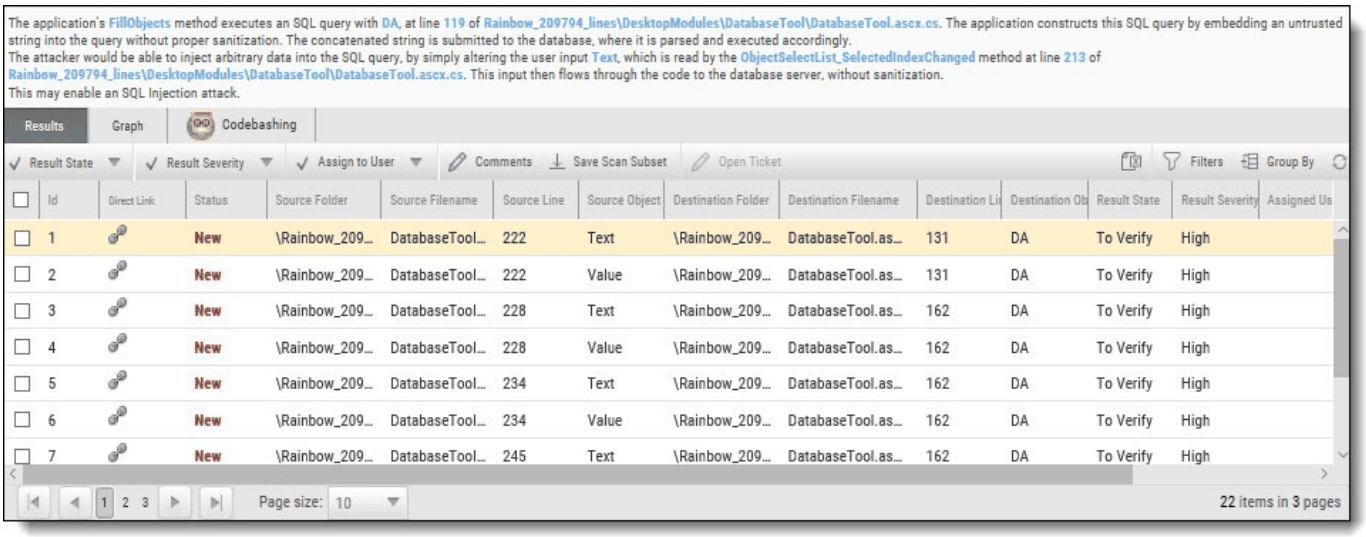
- **Scan Results Summary pane**: Summary of vulnerabilities detected, grouped by High, Medium and Low titles. The summary shows the number of instances of those vulnerability appearances in the code. The “tool tip” displays more information about the specific vulnerability and best practice technique for removal.



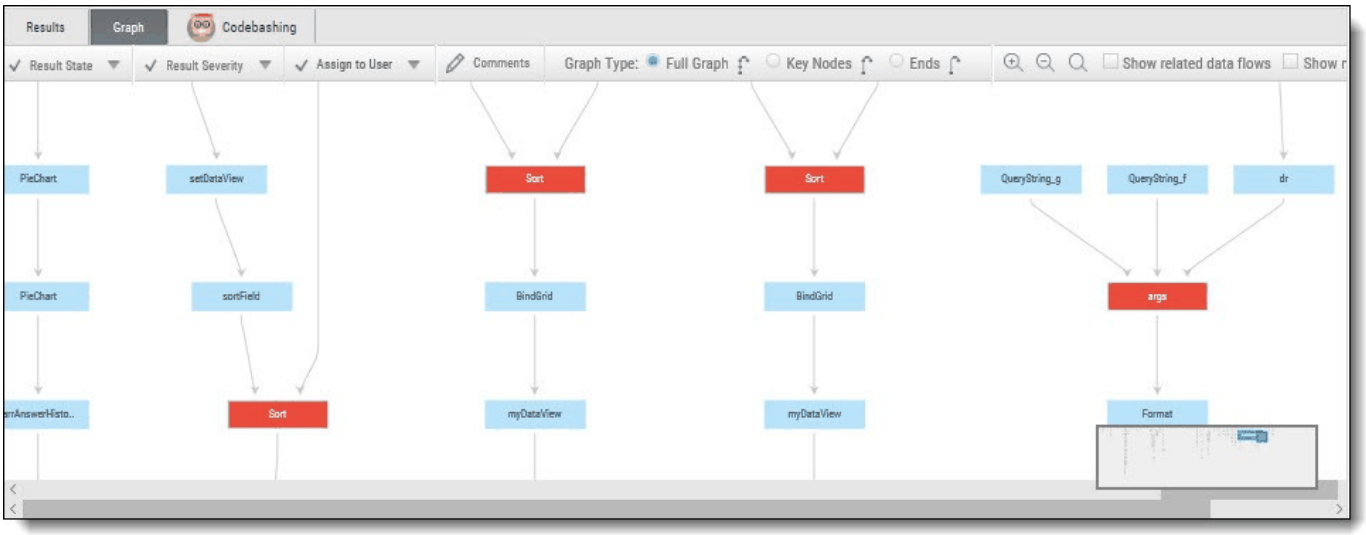
- **Source Code pane**: View specific points of vulnerabilities detected within the Source Code.



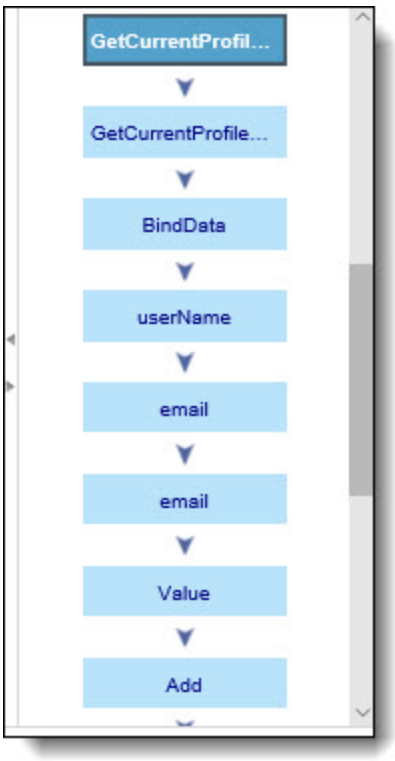
- **Results Table**: A listing of each vulnerability instance and detail. Manage results by using the Filter button to organizes data and saves results.



- **Graph**: Gain a macro chart perspective vulnerabilities found in code, see correlations and identify the optimal points for fix (red buttons).



- **Attack Vector**: Note the full path of code elements that constitute the vulnerability instance selected in the Results pane.

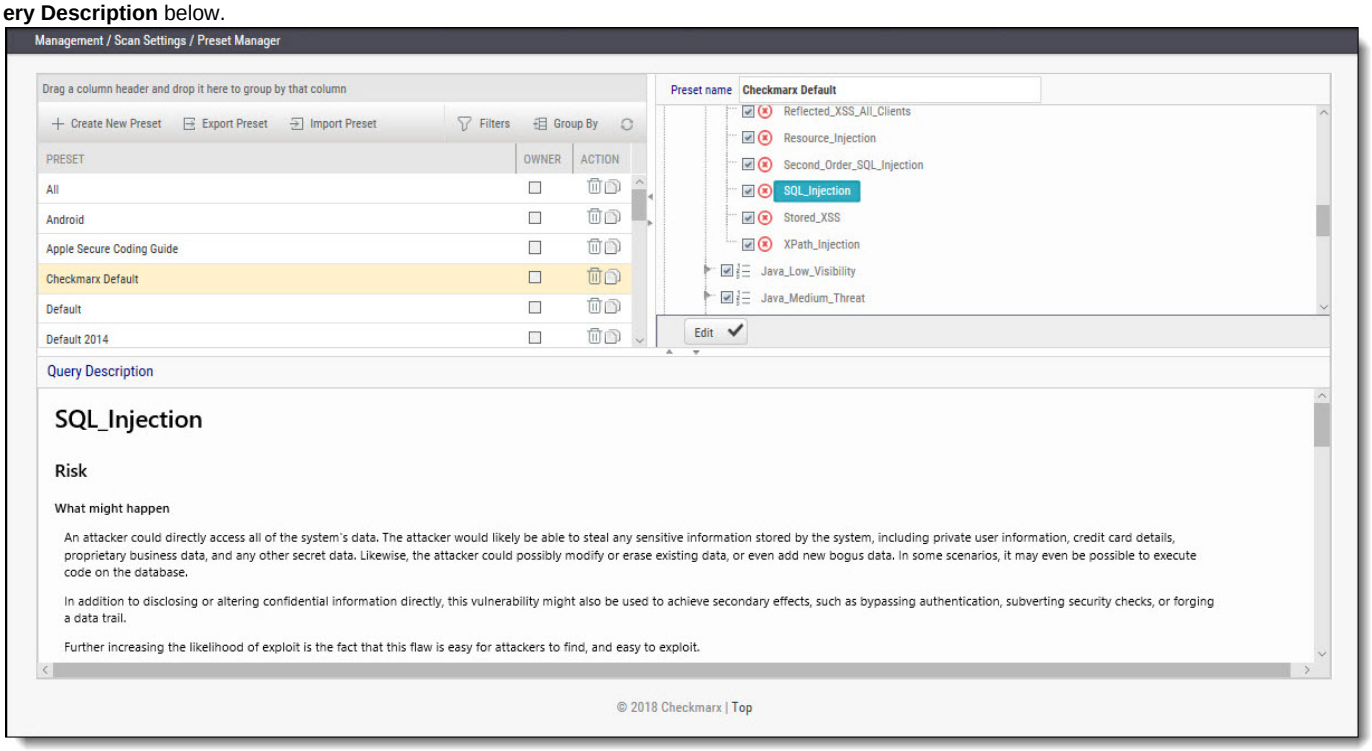


For more information on Working with Scan Results, click.

Preset Manager: Overview

A Preset Setting consists of a group of queries. The Preset Manager enables the viewing of query details in each Preset.

To access the Preset Manager go to **Management > Scan Settings > Preset Manager**.
Queries contained inside the preset are presented in the right pane and description of vulnerability discovered by each query are described in **Query Description** below.



Configuring a CxSAST Scan Action using Jenkins Pipeline (v8.9.0 and up)

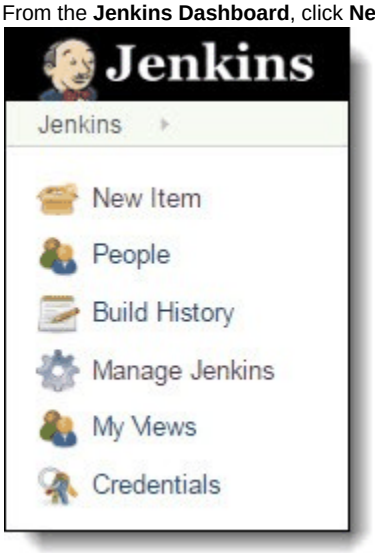
Before starting the configuration, please make sure you already have the Pipeline plugin installed on your Jenkins - https://jenkins.io/doc/pipeline/tour/hello-world/

Once the CxSAST Jenkins plugin is set up and configured (see Setting Up and Configuring the Jenkins Plugin) you can configure any Jenkins job/project to perform a CxSAST scan action using Jenkins Pipeline.

Jenkins allows to create multiple job types. The Checkmarx Jenkins plugin supports 'Freestyle project' and 'Pipeline' jobs. While other job types may work with the Checkmarx Jenkins plugin, they are not officially supported.

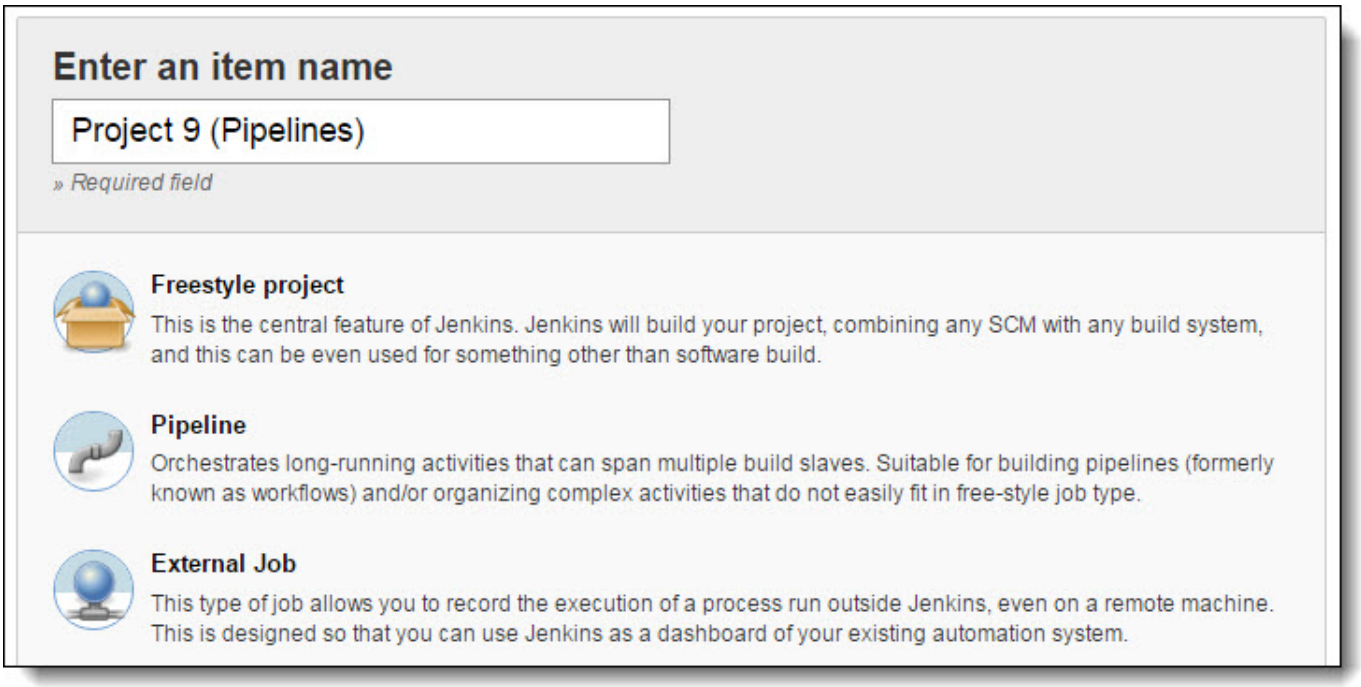
To configure a CxSAST scan action using Jenkins Pipeline:

From the **Jenkins Dashboard**, click **New Item**.



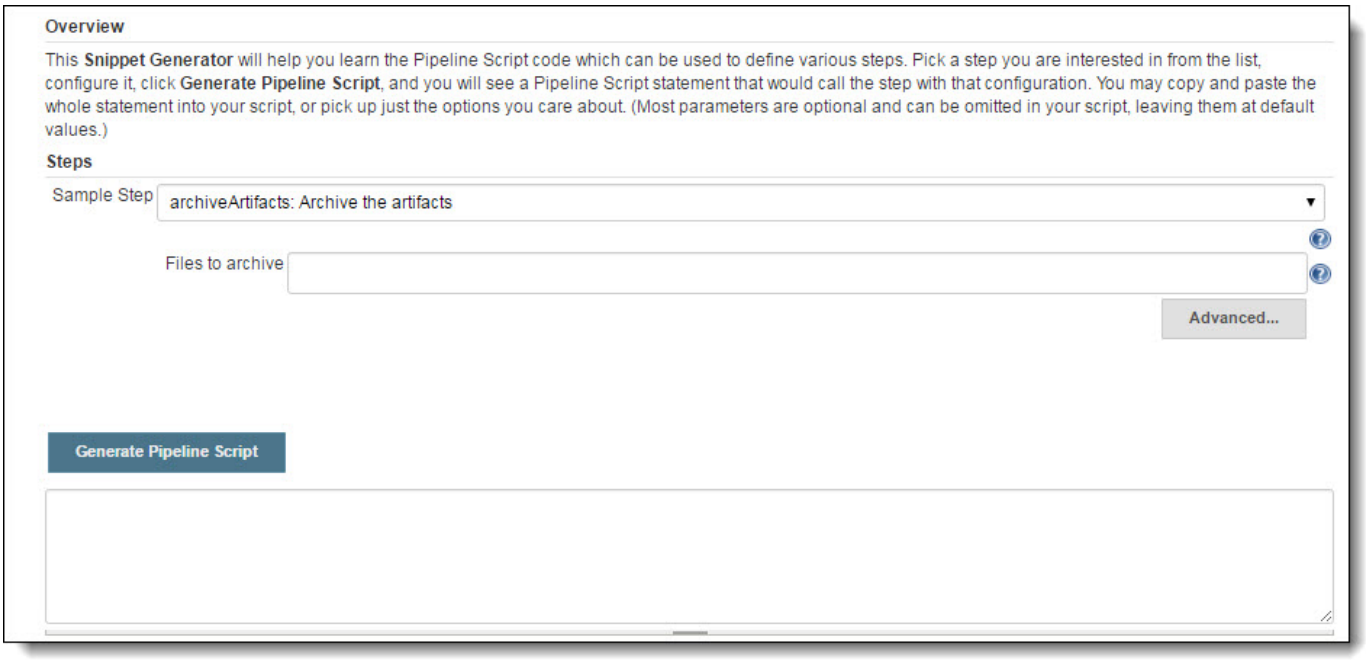
You can also select an existing pipeline Job/project from the **Dashboard** and click **Configure**.

Enter a name into the **Item Name** field.

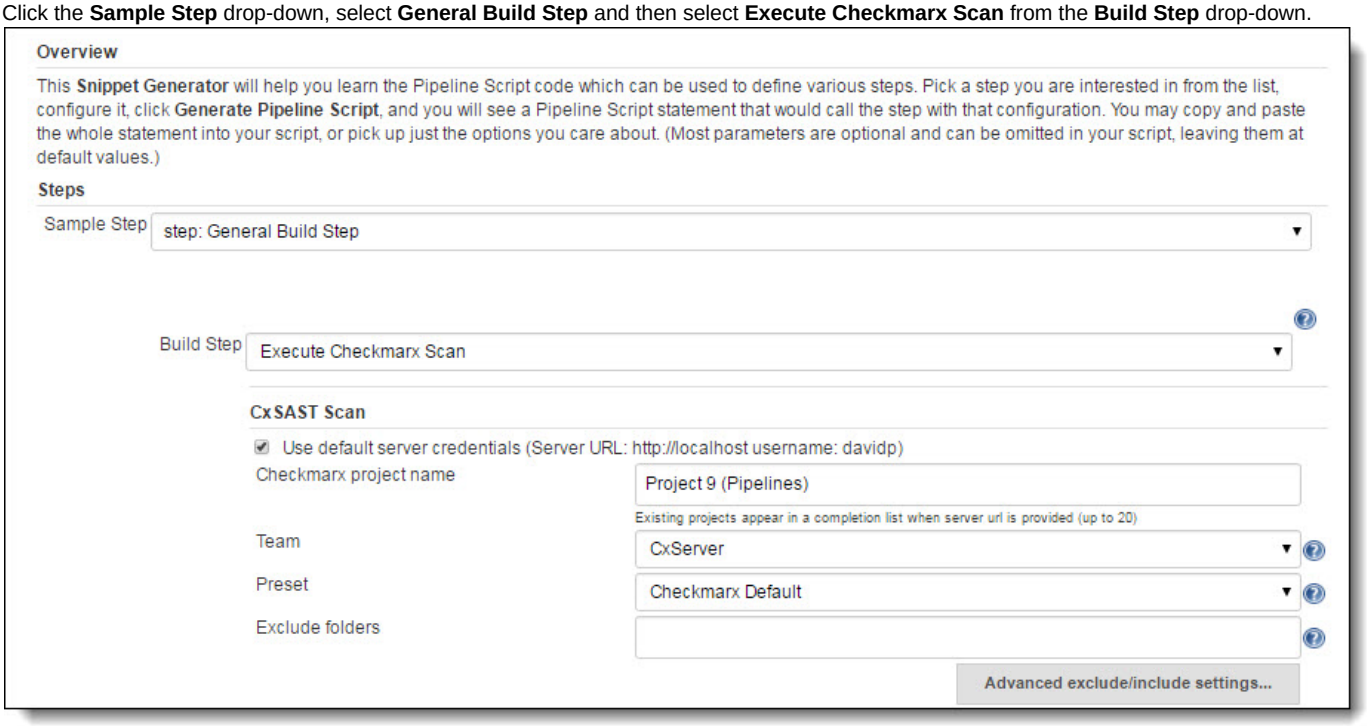


Select **Pipeline** and click **OK**.

Once the Job Configuration is displayed, scroll down to **Pipeline** and click **Pipeline Syntax**. The **Snippet Generator** is displayed.



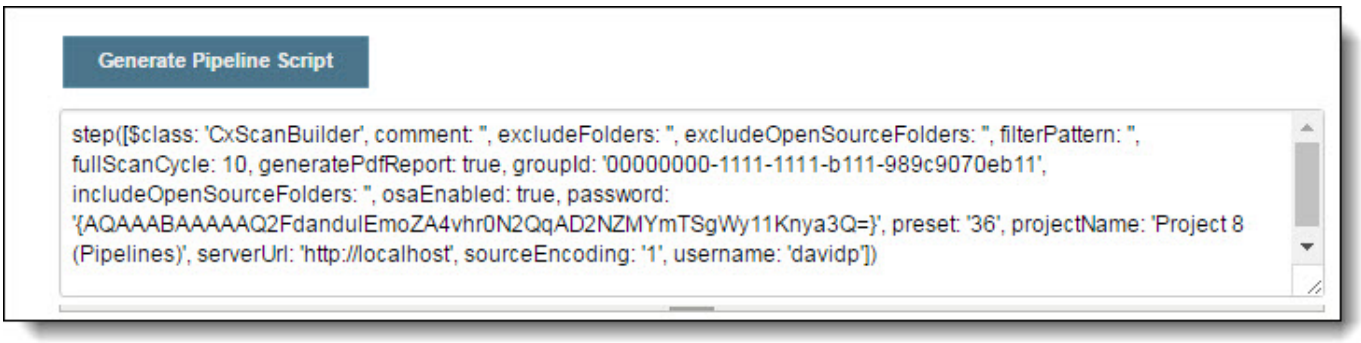
Click the **Sample Step** drop-down, select **General Build Step** and then select **Execute Checkmarx Scan** from the **Build Step** drop-down.



If the **Execute Checkmarx Scan** option is not available, check that you have installed the latest version of the CxSAST Jenkins plugin (8.42.0 and up).

Once the **CxSAST Scan Configuration** is displayed, define the relevant job/project scan parameters (see Configuring a CxSAST Scan Action an d Configuring a CxOSA Action.

Click the **Generate Pipeline Script** button. The generated pipeline script is displayed.



Generating a new pipeline script, by default, contains the 'groupId' parameter which represents the team path. If you would prefer to use the 'teamPath' (e.g. CxServer\\SP\\Company\\Users), you will need to add the 'teamPath' parameter to the script manually. You should also remove the 'groupId' parameter (recommended), but it is not mandatory as 'teamPath' will override 'groupId'.

For additional information about the pipeline script parameters, see Editing the Pipeline Script Parameters, below.

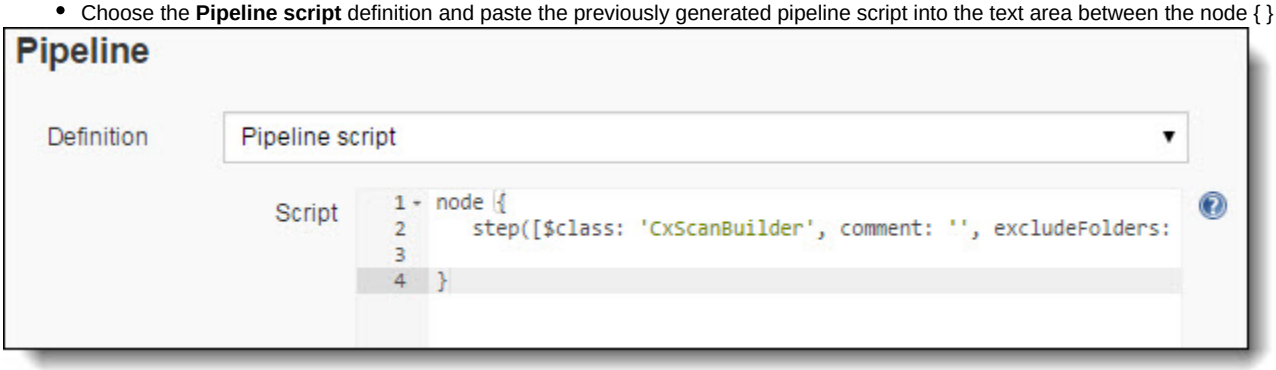
Copy the generated script to your clipboard and choose one of the following options:

Pipeline stored in the SCM

- Paste the generated script into an existing (or new) jenkinsfile, between the node { } markers, and commit to the SCM.
- Choose the **Pipeline script from SCM** definition in Jenkins pipeline configuration, select the relevant **SCM** (Git or Subversion) and define the relevant parameters and **Script Path**.

Internal Jenkins pipeline script

- Choose the **Pipeline script** definition and paste the previously generated pipeline script into the text area between the node { } markers.



Click **Save** to save the changes.

Click **Build Now** to perform a job/project scan using Jenkins Pipeline.

Editing the Pipeline Script Parameters

Once the pipeline script has been generated, you have the option to edit the script parameters accordingly.

Parameters	Description
avoidDuplicateProjectScans:	Enables the option that if there is a scan for this project in the queue in status working or queued do not send a new scan request. True=Enabled.
comment:	Includes optional remark for the scan action (e.g. scan originating from Jenkins).
credentialsId:	Defines your credentials Id as it is in the Jenkins credentials manager.
excludeFolders:	Comma separated list of folders to exclude from the CxSAST scan (e.g. folder 1, folder 2, folder 3).
excludeOpenSourceFolders:	Comma separated list of folders to exclude from the CxOSA scan (e.g. folder1 folder2 folder3)
exclusionsSetting:	Defines to use global include/exclude settings. This displays the values from the predefined global include/exclude settings (e.g. global).

failBuildOnNewResults:	Enables the option to fail the build according to the defined severity (or higher). This option works in addition to the regular thresholds. This means that if "x" total high vulnerabilities were found OR at least one NEW vulnerability is introduced, then fail the build). True = enabled. This option is only available if the ‘vulnerabilityThresholdEnabled’ parameter is enabled.
failBuildOnNewSeverity:	Defines the fail build severity (high, medium, low). This option is only available if the ‘failBuildOnNewResults’ parameter is enabled.
filterPattern:	Defines the include/exclude wildcard patterns (e.g. '"!**/_cvs/**/*, !**/.svn/**/*, !**/.hg/**/*, !**/.git/**/*, !**/.bzr/**/*, !**/bin/**/*,**/obj/**/*, !**/backup/**/*, !**/.idea/**/*, !**/*.DS_Store, !Checkmarx/Reports/*.*"'
fullScanCycle:	Defines the number of incremental scans to be performed, before performing a periodic full scan (e.g. 10).
generatePdfReport:	Enables the creation of a scan result report in PDF. True = enabled. NOTE: The report is available via a link in the scan results in Jenkins.
groupId:	Defines the relevant team Id (e.g. 00000000-1111-1111-b111-989c9070eb11). NOTE: The team Id can only be determined by using the Get All Teams CxREST API (GET /auth/teams) or by using the Pipeline syntax generator to create a new Checkmarx pipeline step.
highThreshold:	Defines the CxSAST high severity vulnerability threshold. If set, the threshold, is crossed if the number of high severity vulnerabilities exceeds it (e.g. 5). This option is only available if the ‘vulnerabilityThresholdEnabled’ threshold option is enabled.
includeOpenSourceFolders:	Defines a comma separated list of include or exclude wildcard patterns. Exclude patterns start with exclamation mark "!". Example: *.jar */folder/* */folder1/folder2/* */folder*/* */file.* */file*.jar */test/*file*.* May reference build parameters like ${PARAM}. Examples: "**/*.jar" matches all .jar jars in a directory tree. "*/test/a??.jar" matches all files/dirs which start with an 'a', then two more characters and then ".jar", in a directory called test. "**" matches everything in a directory tree. "**/test/**/XYZ*" matches all files/dirs which start with "XYZ" and where there is a parent directory called test (e.g. "abc/test/def/ghi/XYZ123"). An empty value includes all files for the CxOSA scan. This option is only available if ‘osaEnabled’ parameter is enabled.
incremental:	Enables incremental scan (scan only new and modified files relative to the project's previous scan). True = Enabled.
lowThreshold:	Defines the CxSAST low severity vulnerability threshold. If set, the threshold, is crossed if the number of low severity vulnerabilities exceeds it (e.g. 12). This option is only available if the ’vulnerabilityThresholdEnabled’ option is enabled.
mediumThreshold:	Defines the CxSAST medium severity vulnerability threshold. If set, the threshold, is crossed if the number of medium severity vulnerabilities exceeds it (e.g. 7). This option is only available if the ’vulnerabilityThresholdEnabled’ option is enabled.
osaArchiveIncludePatterns:	Defines a comma-separated list of archive wildcard patterns to include their extracted content for the scan (e.g. *.zip, *.jar, *.ear).
osaEnabled:	Enables option to initiate CxOSA scan for this project/job. True = Enabled.
osaHighThreshold:	Defines a threshold for the CxOSA high severity vulnerabilities. The build will be marked (failed or unstable) if the number of the high severity vulnerabilities is larger than the threshold (e.g. 1).
osaInstallBeforeScan:	Defines this option in order to be able to scan packages from various dependency managers (NPM, Nugget, Python and more) as part of the CxOSA scan. True = Enabled.
osaLowThreshold:	Defines a threshold for the CxOSA low severity vulnerabilities. The build will be marked (failed or unstable) if the number of the low severity vulnerabilities is larger than the threshold (e.g. 2).
osaMediumThreshold:	Defines a threshold for the CxOSA medium severity vulnerabilities. The build will be marked (failed or unstable) if the number of the medium severity vulnerabilities is larger than the threshold (e.g. 3).
password:	Deprecated and should not be used.
preset:	Defines a scan preset for the project. If the preset is not specified, the default preset for a new project will be used (e.g. 36).
projectName:	Define the relevant project name.
sastEnabled:	Enables the option to initiate CxSAST scan for this project/job. True = Enabled.

serverUrl:	Checkmarx Server URL or IP address with or without port (e.g. http://server-name or https://ip:port). This option is only available if the 'useOwnServerCredentials' parameter is disabled.
sourceEncoding:	Language encoding Id (Japanese, Korean, etc..).
useOwnServerCredentials:	Enables the use of the default server credentials or disable and provide server and credentials that override the defaults. True=Enabled.
username:	Deprecated and should not be used.
vulnerabilityThresholdEnabled:	Enables the vulnerability threshold option. This option is only available if the 'waitForResultsEnabled' parameter is set as enabled. True = Enabled.
vulnerabilityThresholdResult:	Defines the build status (failed of unstable) for when the result of scanned vulnerabilities exceed the threshold.
waitForResultsEnabled:	Enables the 'waitForResultsEnabled' (synchronous mode) option. Synchronous mode allows viewing scan results in Jenkins. Enable = True. If disabled (asynchronous mode) a link to the scan results in the Checkmarx web application is displayed in the Jenkins build results.

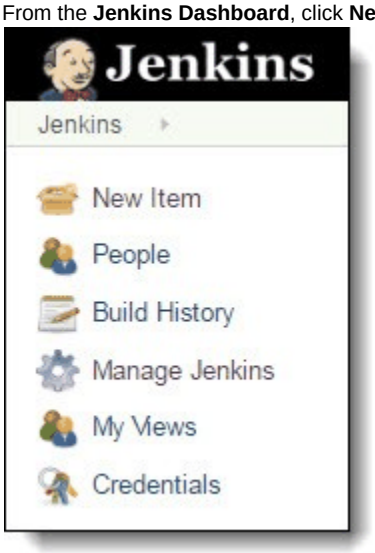
Configuring a CxSAST Scan Action using Jenkins Freestyle projects (v8.9.0 and up)

Once the CxSAST Jenkins plugin is set up and configured (see Setting Up and Configuring the Jenkins Plugin) you can configure any Jenkins job/project to perform a CxSAST scan action using Jenkins.

Jenkins allows to create multiple job types. The Checkmarx Jenkins plugin supports 'Freestyle project' and 'Pipeline' jobs. While other job types may work with the Checkmarx Jenkins plugin, they are not officially supported.

To configure a CxSAST scan action using Jenkins freestyle project:

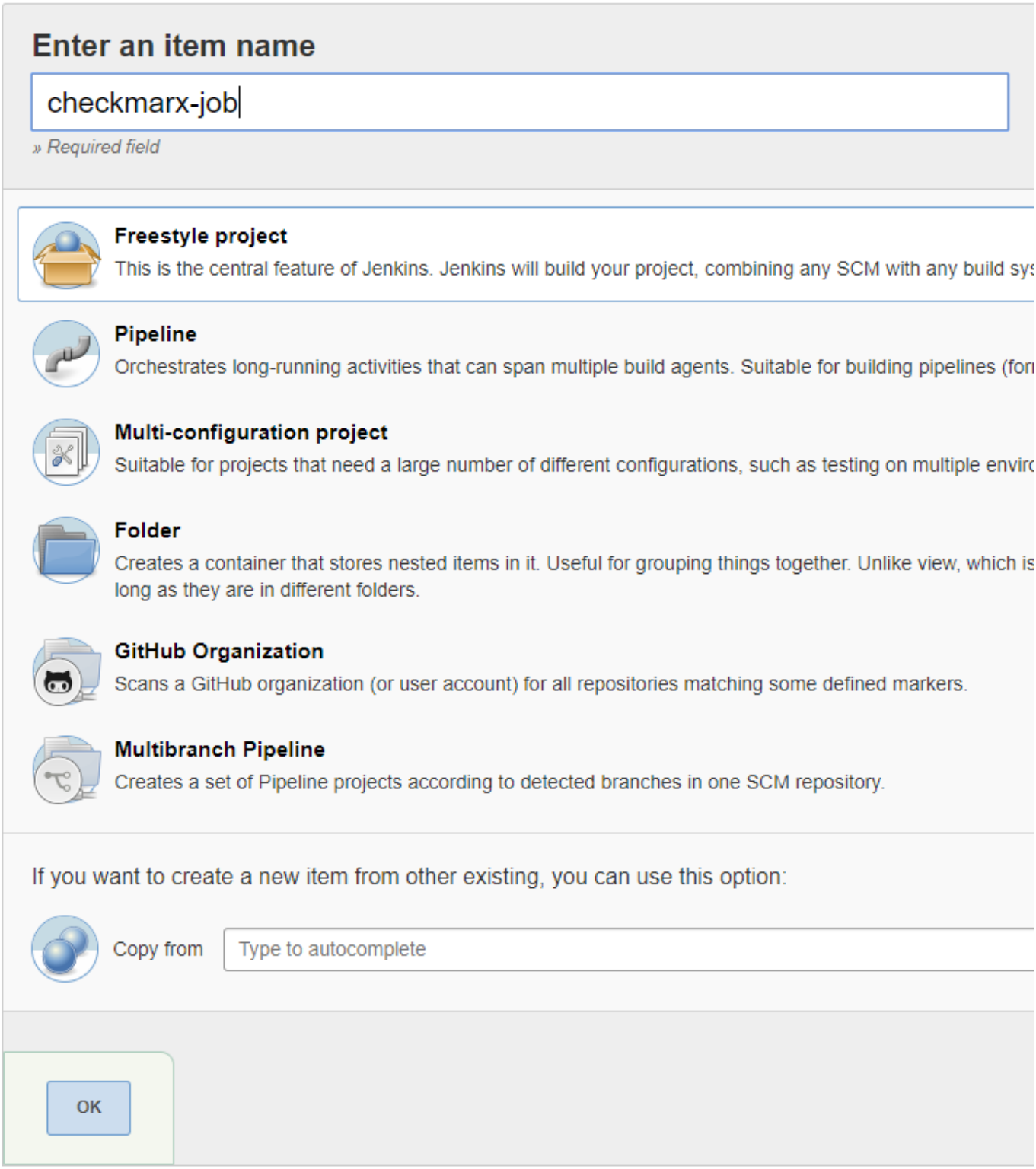
From the **Jenkins Dashboard**, click **New Item**.



You can also select an existing freestyle Job/project from the **Dashboard** and click **Configure**.

For a new freestyle project please enter a name into the **Item Name** field.

Select Freestyle Project and click OK.



Once the Job Configuration is displayed, provide details like description, Git server details.

Source Code Management

- None
- Git

Repositories

Repository URL: https://github.com/vpbobade/altimetrik_repo

Credentials: - none - | Add

Build

Add build step

- Execute Checkmarx Scan
- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

Build

Execute Checkmarx Scan

Checkmarx Server Settings

- [ ] Use default server credentials (not set)

Checkmarx server URL: http://localhost

Credentials: - none - | Add

Test Connection

Checkmarx project name: checkmarx-job

Existing projects appear in a completion list when server url is provided (up to 20)

Team: Provide Checkmarx server credentials to see teams list

CxSAST Scan

- [x] Enable CxSAST scan

Preset: Provide Checkmarx server credentials to see presets list

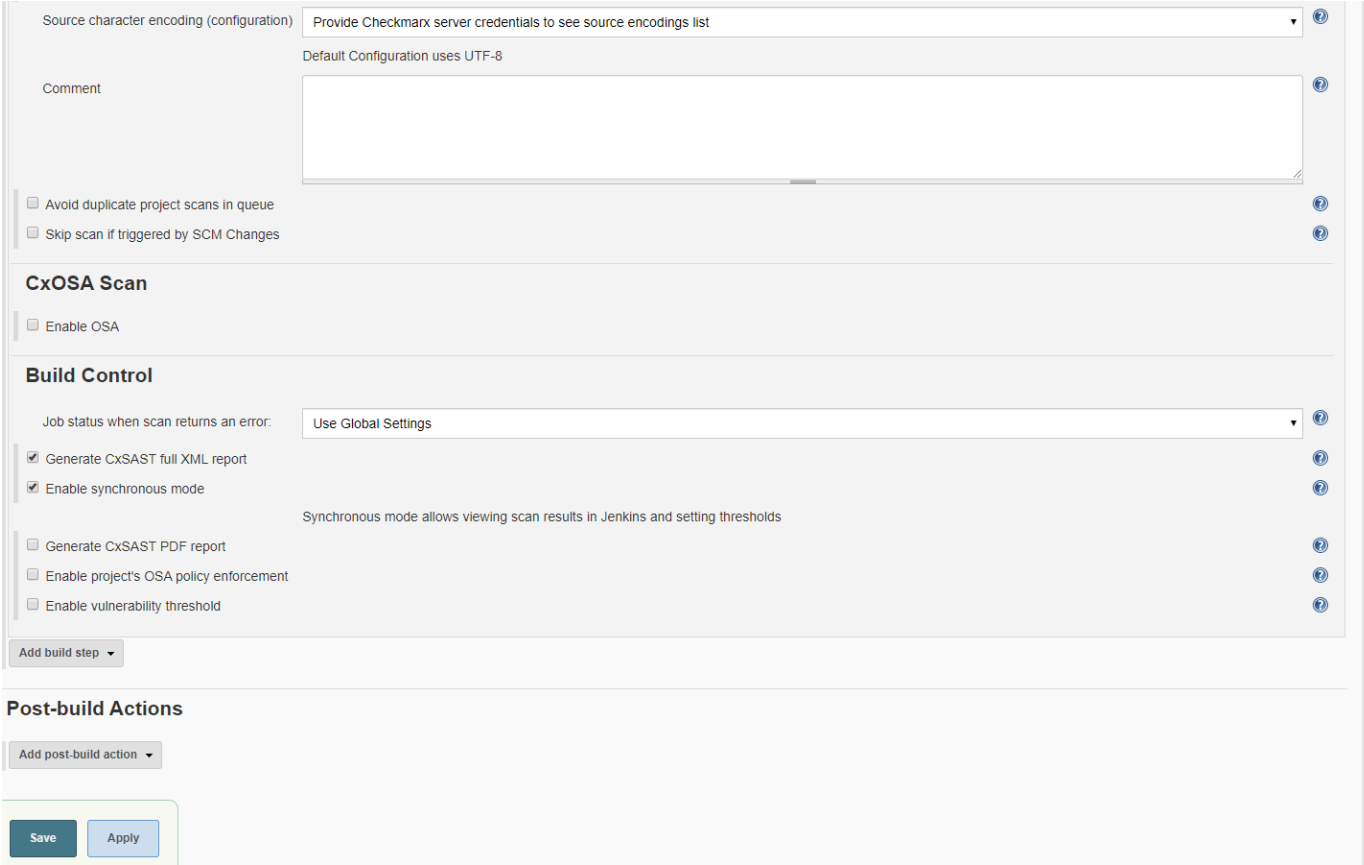
- (•) Use Global Include/Exclude Settings

Exclude Folders

Include/Exclude Wildcard Patterns

- ( ) Specific Include/Exclude Settings

Save | Apply



Click **Save** to save the changes.

Click **Build Now** to perform a job/project scan using Jenkins Pipeline.

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/checkmarx-job
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/vpbobade/altimetrik_repo
 > git init /var/lib/jenkins/workspace/checkmarx-job # timeout=10
Fetching upstream changes from https://github.com/vpbobade/altimetrik_repo
 > git --version # timeout=10
 > git fetch --tags --progress -- https://github.com/vpbobade/altimetrik_repo +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/vpbobade/altimetrik_repo # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/vpbobade/altimetrik_repo # timeout=10
Fetching upstream changes from https://github.com/vpbobade/altimetrik_repo
 > git fetch --tags --progress -- https://github.com/vpbobade/altimetrik_repo +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 3a454628c1e76416c89e09c030b963305c413a78 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 3a454628c1e76416c89e09c030b963305c413a78 # timeout=10
Commit message: "tp3"
First time build. Skipping changelog.
[Cx-Warning]: Invalid presetId: [null]. Using default preset.
[Cx-Warning]: Invalid source encoding (configuration) value: [Provide Checkmarx server credentials to see source encodings list]. Using default configuration.
[Cx-Info]: -------------------------------------Configurations:-----------------------------------
[Cx-Info]: plugin version: 8.90.4
[Cx-Info]: server url: http://localhost
[Cx-Info]: username:
[Cx-Info]: project name: checkmarx-job
[Cx-Info]: team path: null
[Cx-Info]: team id: null
[Cx-Info]: is synchronous mode: true
[Cx-Info]: deny project creation: false
[Cx-Info]: SAST scan enabled: true
[Cx-Info]: avoid duplicated projects scans: false
[Cx-Info]: enable Project Policy Enforcement: false
[Cx-Info]: preset id: 7
[Cx-Info]: SAST folder exclusions: null
[Cx-Info]: SAST filter pattern: null
[Cx-Info]: SAST timeout: -1
[Cx-Info]: SAST scan comment:
[Cx-Info]: is incremental scan: false
[Cx-Info]: is generate full XML report: true
[Cx-Info]: is generate pfd report: false
[Cx-Info]: source code encoding id: 1
[Cx-Info]: SAST thresholds enabled: false
[Cx-Info]: OSA scan enabled: false
[Cx-Info]: ------------------------------------------------------------------------------------------
[Cx-Info]: Initializing Cx client
[Cx-Info]: Logging into the Checkmarx service.
FATAL: org.apache.http.conn.HttpHostConnectException: Connect to localhost:80 [localhost/127.0.0.1] failed: Connection refused (Connection refused)
java.net.ConnectException: Connection refused (Connection refused)
        at java.net.PlainSocketImpl.socketConnect(Native Method)
```

The error above is because we do not have the Checkmarx service running there and for that we would need Checkmarx server up and running somewhere.

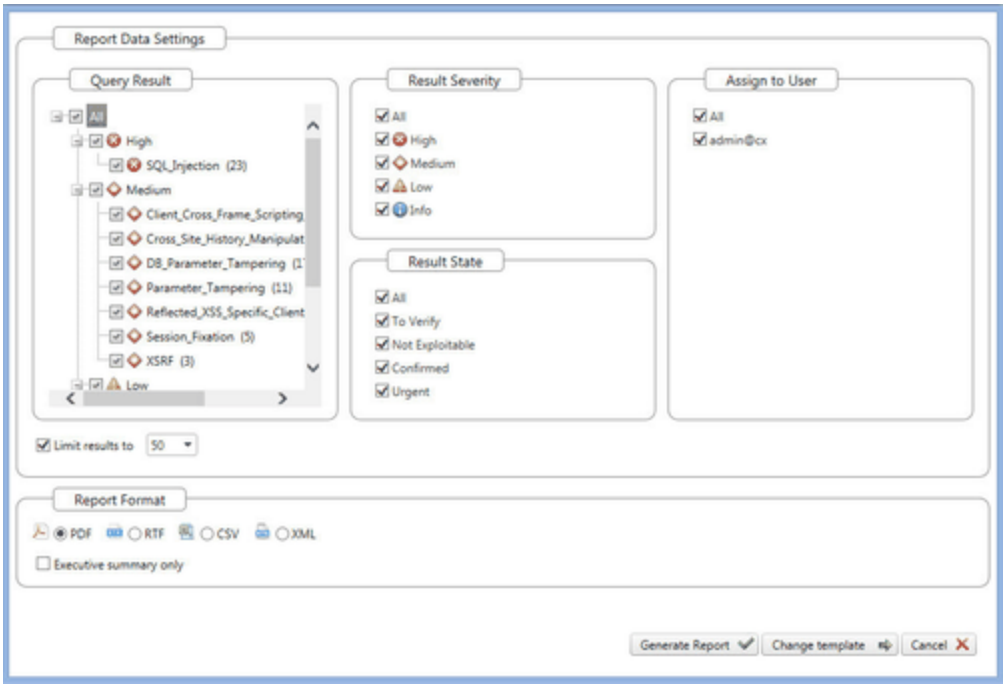
Generating a Scan Result Report

You can generate a report containing detailed scan results, in any of the following formats:

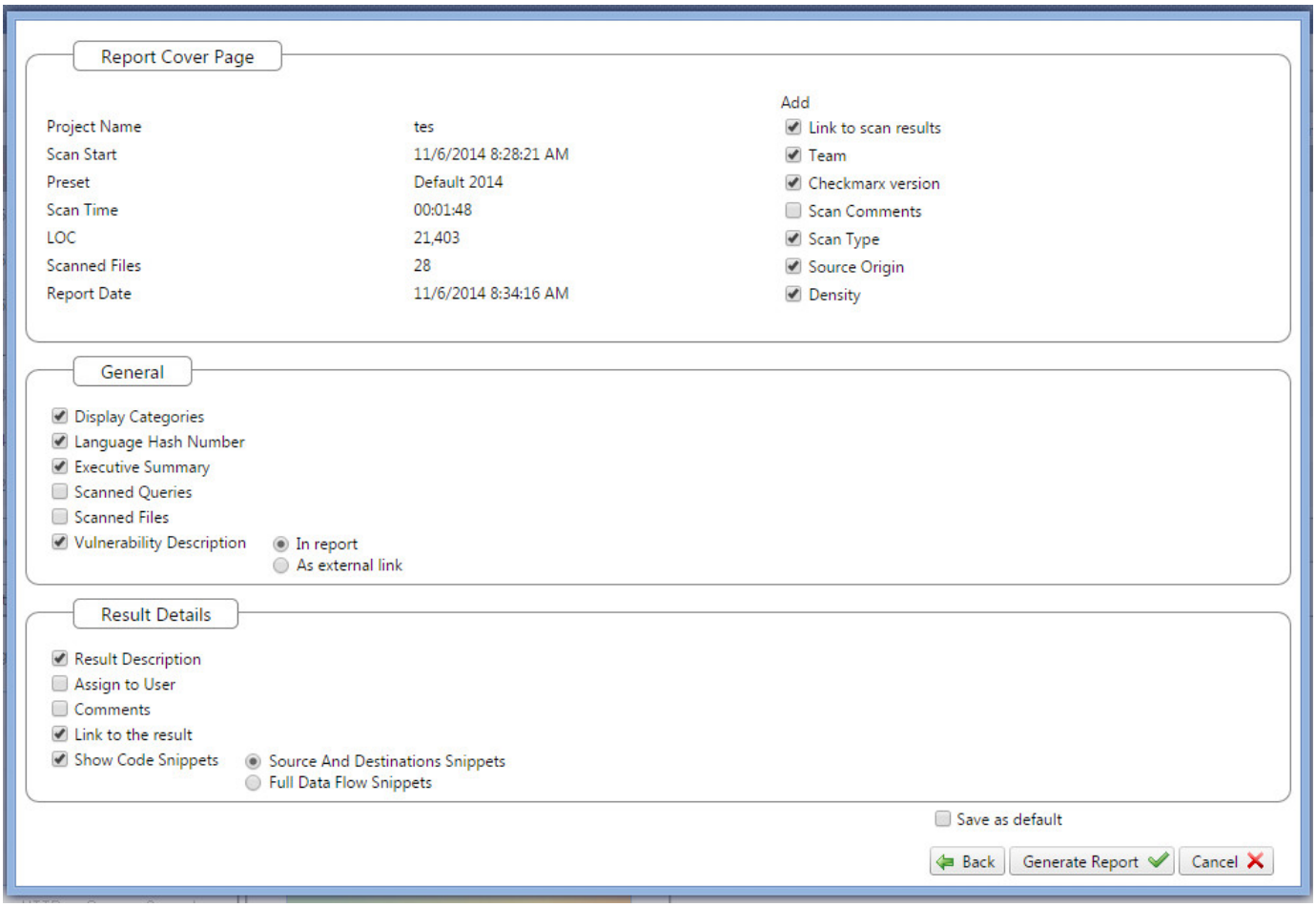
- PDF
- RTF
- CSV
- XML

To generate a report:

1. In a Scan Results table (for all projects or for an individual project), click .
2. Filter results in the generated report and report file format:



1. To change the report template:
    a. Select **Change template** and click **Next**.
    b. Select which details should be presented on the report cover page and in the report itself, and what details to show for each result:



1. Click **Generate Report**.

References:

Presentation:

https://www.slideshare.net/source-code-analysis/application-security-guide-for-beginners

https://www.slideshare.net/source-code-analysis/a-successfulsasttoolimplementation

Everything about Checkmarx:

CxSAST v8.9.0 S...de_new_logo.pdf

Video link for viewing results and understanding security issues via Checkmarx online scanner:

https://youtu.be/RS7NHIhEnIo

Checkmarx actual demo:

https://www.youtube.com/watch?v=-oROS-BH0Mc

Checkmarx Demo of CxSAST: Static Code Analysis Solution

https://www.youtube.com/watch?v=UTUmw2WglLM

Checkmarx & Jenkins integration:

https://www.youtube.com/watch?v=aXMUQj8Wsqg

Checkmarx Source Code Analysis for Eclipse

https://www.youtube.com/watch?v=7wcGp5jbrsE

https://www.youtube.com/watch?v=7xiBX48UbeI

Viewing results/reports and understanding security issues via Checkmarx online scanner

https://youtu.be/RS7NHIhEnIo