CxSAST v9.2.0

Configuration Guide

This document is non-binding and for information purposes only

Contents

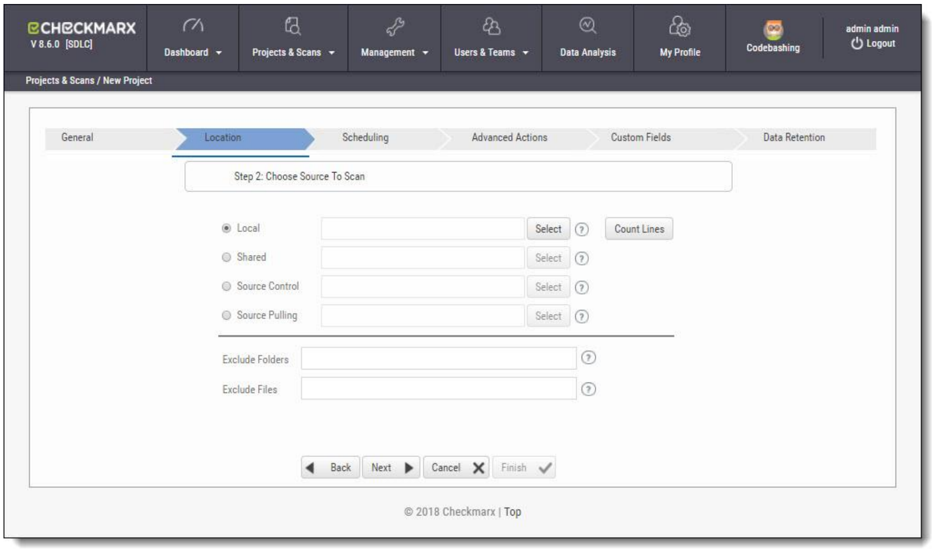
CxSAST Configuration Guide

The Configuration Guide includes advanced configuration procedures and explanations or uncommon scenarios.

Refer to CxSAST Troubleshooting & FAQ for additional information and frequently asked questions.

Configuring the Connection to a Source Control System

When creating a project and the source code **Location** is set to **Source Control**, you can define to which source control system to connect by selecting a source control type (TFS, SVN, GIT or Perforce).

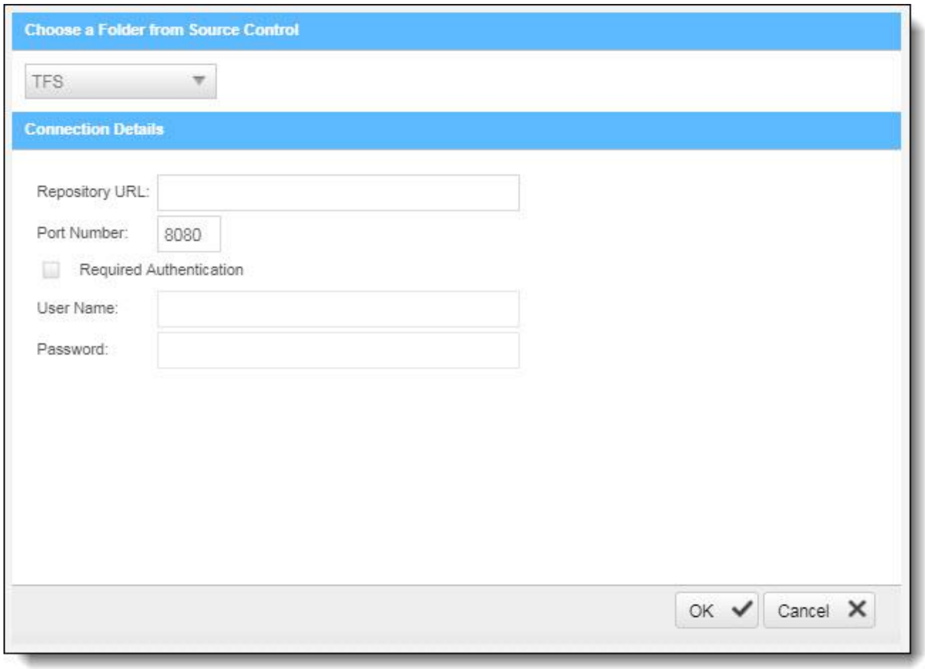


With Source Control option checked, click **Select.** The Source Control window is displayed (see below for connection options).

Files inside a zip file that are located inside a repository will not be sent for scanning. Unzip the contents of the zip file to the repository before scanning.

Defining Source Control for TFS

1. Select **TFS** from the drop-down. The TFS Connection Details panel is displayed.



TFS

Port Number: 8080

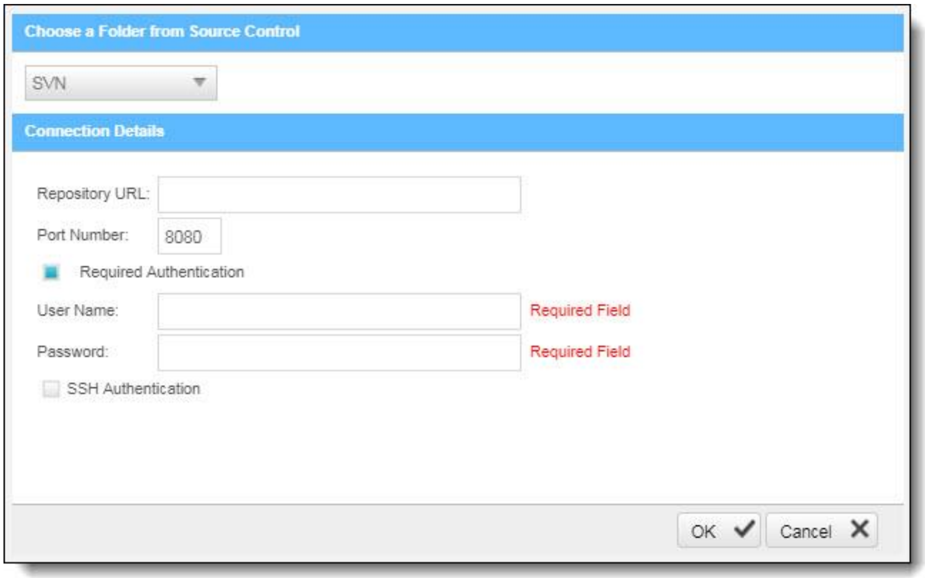
The TFS Connection Details panel includes the following parameters:

- **Repository URL** - the repository URL address (Supports HTTP and HTTPS, i.e. <protocol>://<site name>:<port>/tfs/<Collection> (must point to the repository named <Collection>)).
- **Port Number** - the port number
- **Required Authentication** - select to enforce authentication
- **User Name** - the user name (required with enforced authentication)
- **Password** - the password (required with enforced authentication)

2. Click **OK**.

Defining Source Control for SVN

1. Select **SVN** from the drop-down. The SVN Connection Details panel is displayed.



Choose a Folder from Source Control

SVN

Connection Details

Repository URL:

Port Number: 8080

■ Required Authentication

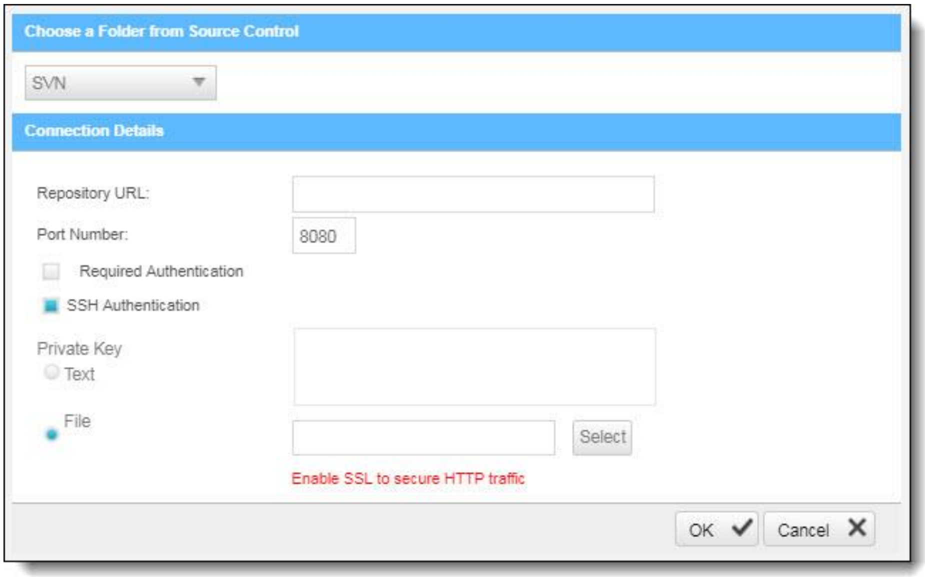
User Name: Required Field

Password: Required Field

□ SSH Authentication

The SVN Connection Details panel includes the following parameters:

- **Repository URL** - the repository URL address (Supports HTTP, HTTPS and SSH private/public key infrastructure,
  i.e. <protocol>://<server_ip>/<repository_name>)
- **Port Number** - the port number
- **Required Authentication** - select to enforce authentication
- **User Name** - the user name (required with enforced authentication)
- **Password** - the password (required with enforced authentication)
- **SHH Authentication** - select to use secure authentication with SSH



Choose a Folder from Source Control

SVN

Connection Details

Repository URL:

Port Number: 8080

□ Required Authentication

■ SSH Authentication

Private Key

○ Text

● File

Enable SSL to secure HTTP traffic

Selecting SHH Authentication displays the following additional parameters:

- **Private Key Text** - add private key text
- **Private Key File** - select and upload a private key file

> - Checkmarx does not support SSH keys with a passphrase.
> - For best results, use ssh-keygen, per these instructions, and not PuTTYgen

2. Click **OK**.

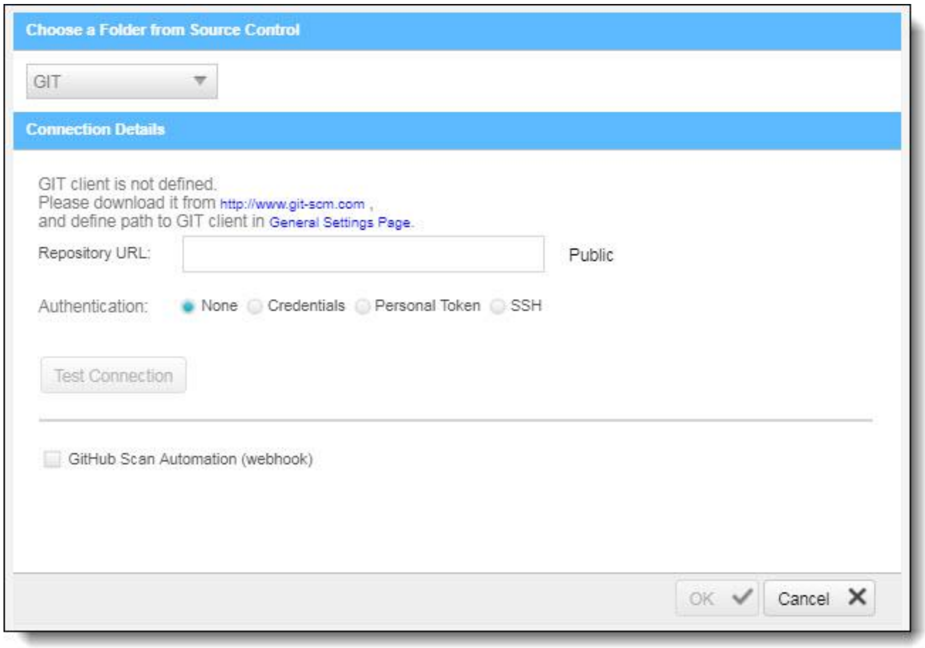
Defining Source Control for GIT

Requirements for Using the GIT Repository

1. Download GIT Installation Package and perform the installation on CxSAST Manager Server (use installation defaults)
2. Define Path+ exe file in CxSAST **Management** > **Application Settings** > **General** > **Path to GIT Client Executable** (i.e. C:\Program Files\Git\bin\git.exe).

Defining the Source Control

1. Select **GIT** from the drop-down. The GIT Connection Details panel is displayed.
2. Click Test Connection. Once the 'Connection Successful' message is displayed, you can continue.



Choose a Folder from Source Control

GIT ▾

Connection Details

GIT client is not defined.
Please download it from http://www.git-scm.com ,
and define path to GIT client in General Settings Page.

Repository URL: Public

Authentication: ● None ○ Credentials ○ Personal Token ○ SSH

□ GitHub Scan Automation (webhook)

The GIT Connection Details panel includes the following parameters:

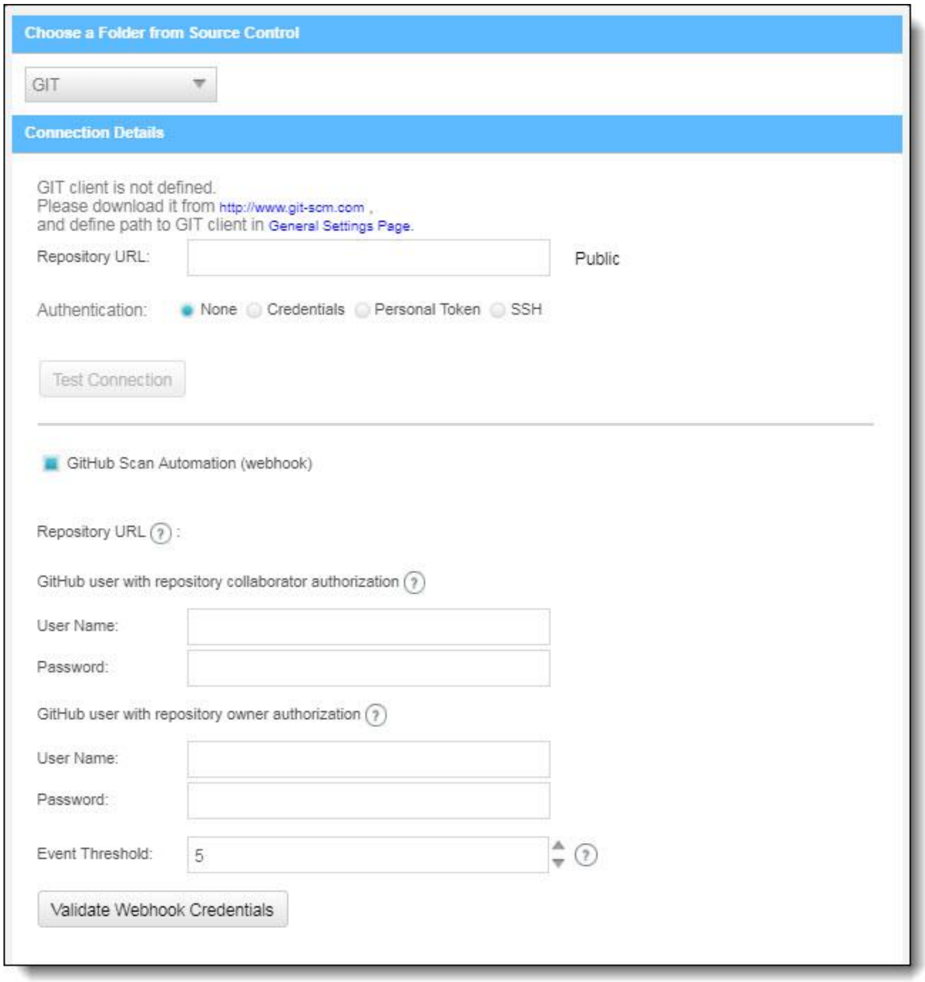
- **Repository URL** – The repository URL address (Supports HTTP, HTTPS, i.e. <protocol>://<user>:<password>@<server_ip>/<repository_name>.git or SSH private/public key infrastructure, i.e. git@<git_site>:<user_name>/<repository_name>.git).

> - If your repository URL contains the character "@", replace it with "%40" (html encoding) before inserting the URL.
> - For a hint to find your GIT Repository URL, refer to GitHub - Tips on Finding Git / GitHub Repository URLs

- **Authentication** – Select an authentication method.

> - For more information about the various authentication methods, refer to **Configuring a Project with Git Integration.**

- **GitHub Scan Automation** – Check to include GitHub Integration.



Choose a Folder from Source Control

GIT ▾

Connection Details

GIT client is not defined.
Please download it from http://www.git-scm.com ,
and define path to GIT client in General Settings Page.

Repository URL: Public

Authentication: ● None ○ Credentials ○ Personal Token ○ SSH

Test Connection

☑ GitHub Scan Automation (webhook)

Repository URL ? :

GitHub user with repository collaborator authorization ?

User Name:

Password:

GitHub user with repository owner authorization ?

User Name:

Password:

Event Threshold: 5 ?

Validate Webhook Credentials

3. Enter the GitHub repository owner and collaborator credentials into the relevant User Name and Password fields.

- The GitHub user with repository owner authorization is used for creating and using a GitHub WebHook (see **GitHub Webhooks**).
- The GitHub user with repository collaborator authorization is used to create commit comments.

4. Configure the Event threshold. A scan in Checkmarx CxSAST will be initiated only after this number of events has occurred, since the last triggered scan.
5. By default, the event threshold value is set to 5, because triggering a scan after fewer events may overload the system. If the user specifies a lower number, a warning message is displayed.
6. Click **Validate Webhook Credentials** to confirm authentication to the GitHub webhooks works correctly. A 'Server Connection Verified Successfully' message is displayed.
7. Click **OK** to complete the procedure.

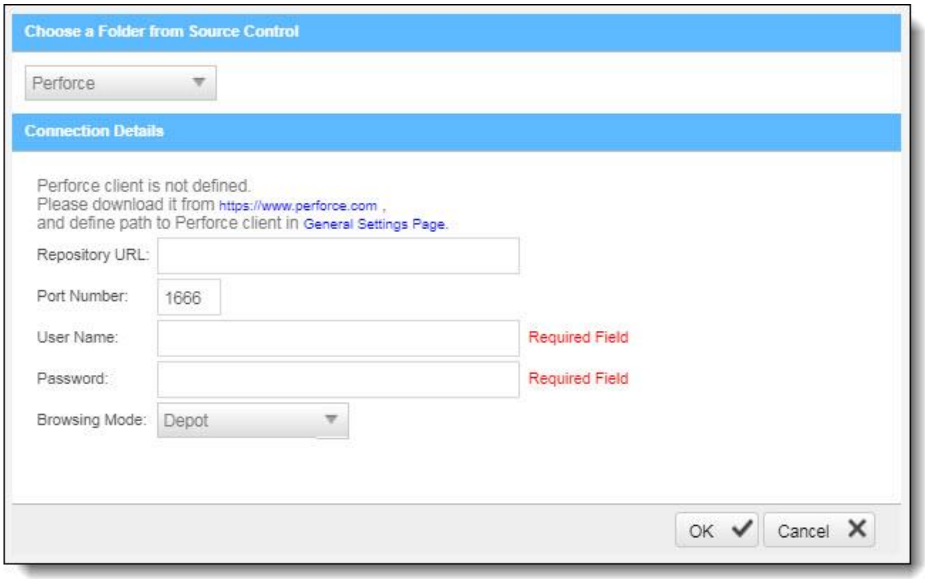
For more information about the various options for GitHub integration, please refer to Github Integration

Defining Source Control for Perforce

Currently CxSAST is unable to scan code from any system that contains symbolic links.

- Select **Perforce** from the drop-down list.

The Perforce Connection Details panel is displayed.



Choose a Folder from Source Control

Perforce

Connection Details

Perforce client is not defined.
Please download it from https://www.perforce.com ,
and define path to Perforce client in General Settings Page.

Repository URL:

Port Number: 1666

User Name: Required Field

Password: Required Field

Browsing Mode: Depot

The Perforce Connection Details panel includes the following parameters:

- **Repository URL** - the repository URL address (i.e. SSL:<server_ip> or <server_ip>)
- **Port Number** - the port number
- **User Name** - the user name
- **Password** - the unique password
- **Browsing Mode** - select **Depot** (for shared file repositories) or **Workspace** (for grouped file repositories).

8. Click **OK**.
9. To set the Perforce client executable path, refer to the Path to P4 command line client executable parameter in the Server Settings.

You can now continue to configure the project.

- **For All connections** – The connection between CxManager Server and the 3rd party repo server is established with the credentials that have been configured for the CxPool IIS Application Pool.

Configuring CxConsole to Use a Proxy

If your network requires a proxy to connect to the CxSAST server, you'll need to configure CxConsole as follows:

1. Open the following file for editing:
   ...\CxConsole\runCxConsole.cmd
2. Find the following line:
   java -jar CxConsolePlugin-CLI-9.00.2.jar %*
3. Change the above line to:
   java -Xmx1024m -Dhttp.proxyHost=<proxy server> **-Dhttp.proxyPort=**<port> **-DsocksProxyHost=**<proxy server> **-jar CxConsolePlugin-CLI-9.00.2.jar %***
   where <proxy server> (appears twice) is the IP address or resolvable name of your network proxy server, and <port> is the port on which the proxy server is listening.
   For example:
   java -Xmx1024m -Dhttp.proxyHost=10.31.0.128 -Dhttp.proxyPort=808 -DsocksProxyHost=10.31.0.128 -jar CxConsolePlugin-CLI-9.00.2.jar %*

- We recommend that, rather than edit our script every time an update is required, setting the following in the global environment, or before calling the script:

- **Linux** - export JAVA_TOOL_OPTIONS="-Dhttp.proxyHost=http-proxy -Dhttp.proxyPort=3128 -Dhttps.proxyHost=http-proxy -Dhttps.proxyPort=3128"
- **Windows** - set JAVA_TOOL_OPTIONS="-Dhttp.proxyHost=http-proxy -Dhttp.proxyPort=3128 -Dhttps.proxyHost=http-proxy -Dhttps.proxyPort=3128"

For additional information on connecting via a proxy from a Java command, see here.

Configuring SSL for the Checkmarx Software Exposure Platform

Overview (SSL)

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and intact. To be able to create an SSL connection the web server requires an SSL Certificate.

Checkmarx Software Exposure Platform (SSL)

To secure communications between all Checkmarx Software Exposure Platform components, we recommend that you install signed certificates and enable SSL on all machines/servers to enforce SSL security (HTTPS). These instructions guide you through the procedure to configure the Secure Sockets Layer (SSL) Protocol for the Checkmarx Software Exposure Platform in Distributed or High Availability environments. They also include links to topics that are directly related to this procedure.

Configuring SSL

SSL can be configured via the Checkmarx Software Exposure Platform components for each machine/server accordingly. To configure the SSL, follow the instructions below:

- All machines/servers in the Checkmarx Software Exposure Platform must be part of the same domain and configured in the Domain Name System (DNS), when using machine names

1. Enable SSL support for Access Control by configuring the appsettings.json file (<dir>:\Program Files\Checkmarx\Checkmarx Access Control\appsettings.json):
2. Update *ExternalListenUrls* to reflect IIS configured bindings:

   http(s)://*(port)
   Example: "ExternalListenUrls": "https://*:443"

   If more than one binding is configured:

http(s)://*(port);http(s)://*(port)
Example: "ExternalListenUrls": "https://*:443;https://*:123"

3. Enable SSL Support on the CxManager according to these instructions.
4. Configure all Checkmarx Software Exposure Platform components for HTTPS in the DB table **[CxDB].dbo.[cxComponentConfiguration]** as follows:

   Replace **IdentityAuthority, CxARMPolicyURL**, **CxARMURL**, **CxSASTManagerUri** and **WebServer** keys to include HTTPS.

   SQL Query to view current values:

```
SELECT * FROM [CxDB].[dbo].[CxComponentConfiguration] WHERE [Key] = 'IdentityAuthority'
SELECT * FROM [CxDB].[dbo].[CxComponentConfiguration] WHERE [Key] = 'CxARMPolicyURL'
SELECT * FROM [CxDB].[dbo].[CxComponentConfiguration] WHERE [Key] = 'CxARMURL'
SELECT * FROM [CxDB].[dbo].[CxComponentConfiguration] WHERE [Key] = 'CxSASTManagerUri'
SELECT * FROM [CxDB].[dbo].[CxComponentConfiguration] WHERE [Key] = 'WebServer'
```

   SQL Queries to set URL’s to SSL:

```
UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://{server FQDN}/CxRestAPI/auth'
WHERE [Key] = 'IdentityAuthority'

UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://{server FQDN}:8443'
WHERE [Key] = 'CxARMPolicyURL'

UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://{server FQDN}:8443'
WHERE [Key] = 'CxARMURL'

UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] ='https://{server FQDN}'
WHERE [Key] = 'CxSASTManagerUri'

UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://{server FQDN}'
WHERE [Key] = 'WebServer'
```

   Example:

```
UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://cxsast.checkmarx.net/CxRestAPI/auth'
WHERE [Key] = 'IdentityAuthority'
```

```
UPDATE [CxDB].[dbo].[CxComponentConfiguration]
SET [Value] = 'https://cxsast.checkmarx.net:8443'
WHERE [Key] = 'CxARMURL'
```

5. Enable SSL Support on the CxEngine(s) according to these instructions.
6. Restart the ActiveMQ and all CxManager services, for any changes in the database.
7. Enable SSL support on the load balancer according to instructions provided by your vendor. If you are using **Nginx** as the load balancer, you can use the following configuration:

```
worker_processes  1;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;

        upstream gol-ha2-lb.cxquality.com {
                ip_hash;
                server gol-ha2-mn1.cxquality.com:443;
                server gol-ha2-mn2.cxquality.com:443;
        }

    server {
                listen       80;
                listen       443 ssl;
                server_name gol-ha2-lb.cxquality.com;

                ssl_certificate Cert/newcert.cer;
                ssl_certificate_key Cert/newkey2.key;

                ssl_protocols TLSv1.2;
                ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512:ECDHE-RSA-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384;
                ssl_prefer_server_ciphers on;
                add_header Strict-Transport-Security "max-age=31536000; includeSubDomains; preload";

                ssl_session_timeout 5m;
                ssl_session_cache shared:SSL:50m;
                ssl_session_tickets off;
                server_tokens off;

location / {
         root   html;
         index  index.html index.htm;
                        proxy_pass https://gol-ha2-lb.cxquality.com/;
      }
   }
}
```

8. Enable TLS support (on all machines/servers) according to these instructions.
9. Enable FIPS compliance (on all machines/servers) according to your supported operating system (see example).
10. Enable SSL support and FIPS compliance for Management & Orchestration (M&O) according to these instructions.

Enabling SSL Support on the CxManager

Overview (SSL)

SSL (Secure Sockets Layer) is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. To be able to create an SSL connection the web server requires an SSL Certificate.

CxManager (SSL)

To secure communications between all Checkmarx Software Exposure Platform components, we recommend that you install a signed certificate and enable SSL on the CxManager to enforce SSL security (HTTPS). This instruction defines the procedure for enabling SSL support on the CxManager.

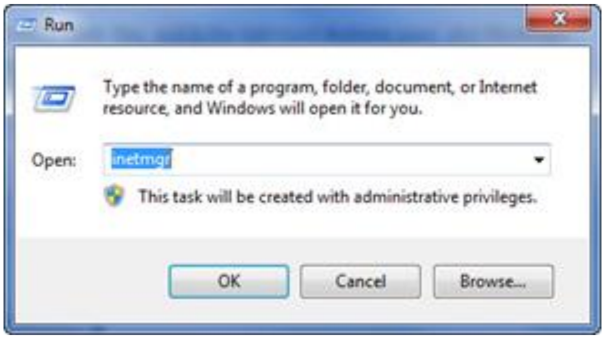
Enabling SSL Support

Support for SSL can be enabled via the IIS Management console on the CxManager server. The enablement steps can be performed manually from the CxManager server:

1. Prepare a **CA certificate** for the Checkmarx Software Exposure Platform Server (in a distributed deployment - for CxManager), signed by a third-party certificate authority such as **VeriSign** and install it on the Server or CxManager.

> - Although it is not considered as safe as CA certification, SSL can also be enabled using Self Signed Certificates, see Create a Self-Signed Server Certificate in IIS.

2. From the **Start** menu, select **All Programs**. Click **Accessories**, and then click **Run**. The Run window is displayed.



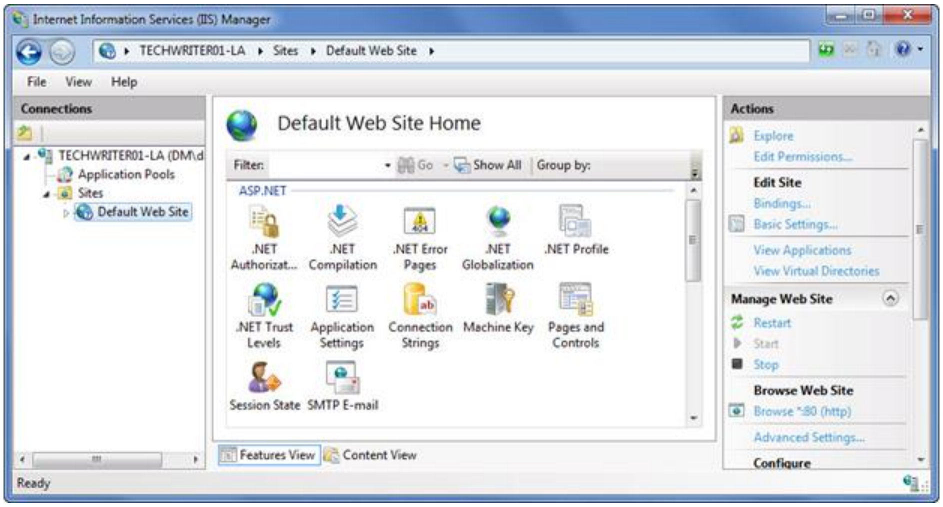
Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

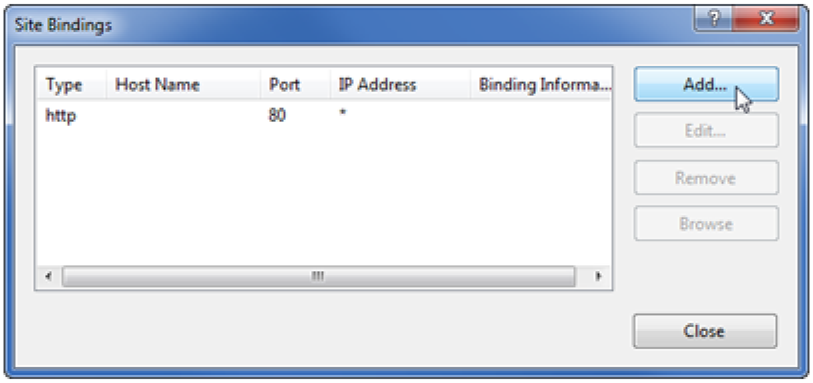
Open: inetmgr

This task will be created with administrative privileges.

3. In the **Open** box, type **inetmgr** and then click **OK**. The IIS Manager window is displayed.

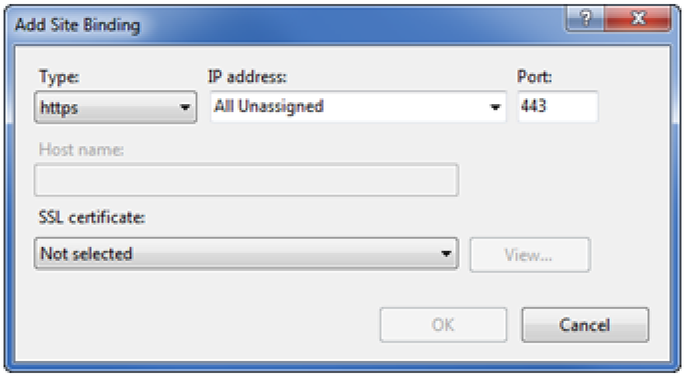


4. Select **Default Web Site** from the **Connections**
5. Select **Bindings** from the **Actions** pane. The Site Bindings window is displayed.



Type	Host Name	Port	IP Address	Binding Informa...
http		80	*	

6. Click **Add**. The Add Site Bindings window is displayed.



Type: https

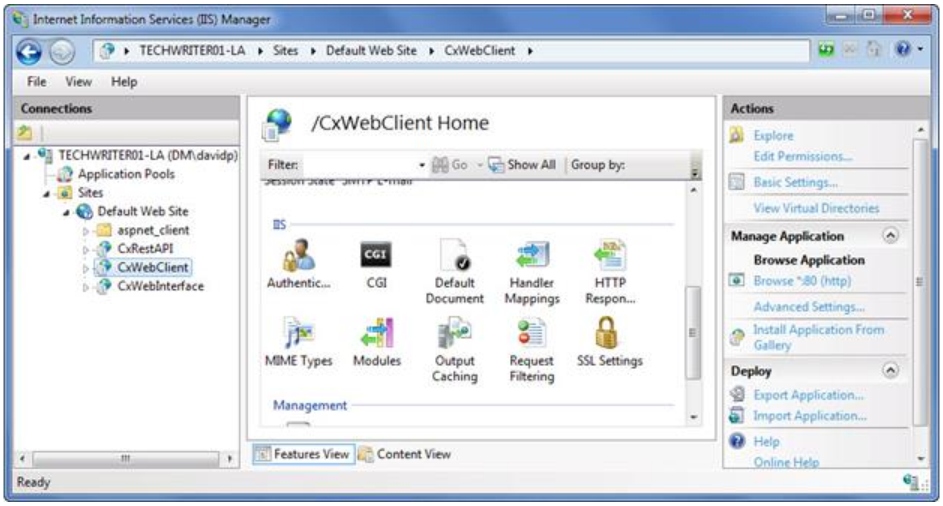
IP address: All Unassigned

Port: 443

Host name:

SSL certificate: Not selected

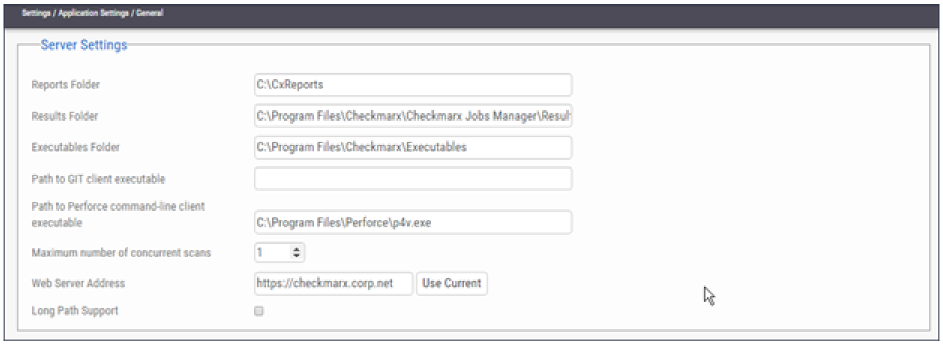
7. Click **Type** and select **https**.
8. Enter the host name of your station.
9. Select the **SSL Certificate** and select the your pre-installed certificate from the list.
10. Click **OK** and then **Close**.
11. If you want the Checkmarx Software Exposure Platform users to be able to use **only HTTPS/SSL**, return to the IIS Manager window and, for each relevant web service (CxWebClient, CxWebInterface), perform the following:
12. Double-click **Default Web Site** from the **Connections** pane.



13. Select **CxWebClient** and double-click on **SSL Settings**.
14. Select **Require SSL** and click **Apply** from the Actions pane.

- Perform the same SSL settings actions for CxRestAPI as well as CxWebInterface.

15. Go to **C:\Program Files\Checkmarx\CheckmarxWebPortal\Web**, open the **web.config** file for editing and using the Search tool, search for "**CxWSResolver.CxWSResolver**".
16. Change the value "**http://**" to "**https://**" and replace the value "**localhost**" (if available) with your pre-installed certificate's <name/subject>.
17. Right-click on the **Server** (highest level in the hierarchical tree) and select **Stop** from the drop-down. Once stopped right-click on the **Server** again and select **Start**.
18. In the **Checkmarx Software Exposure Platform Web Interface**, go to **Management > Application Settings > General**. The **General Settings** window is displayed.



Reports Folder	C:\CxReports
Results Folder	C:\Program Files\Checkmarx\Checkmarx Jobs Manager\Resul
Executables Folder	C:\Program Files\Checkmarx\Executables
Path to GIT client executable	
Path to Perforce command-line client executable	C:\Program Files\Perforce\p4v.exe
Maximum number of concurrent scans	1
Web Server Address	https://checkmarx.corp.net
Long Path Support	□

19. Click **Edit.**
20. Enter your **Server URL** (e.g. https://checkmarx.corp.net) into the **Web Server Address**
21. Click **Update** to save the changes.

Defining HTTPS Settings

After installing CxSAST, define the IIS bindings at the **ExternalListenUrls** key in the **appsettings json** file. If, for example, port 80 (HTTP) and port 443 (HTTPS) have to be bound, the syntax looks as follows: **"ExternalListenUrls": http://*:80;https://*:443 .** The **appsettings json** file resides in the **Checkmarx Access Control** folder.

Configuring SSL between the Manager & Engine

CxSAST supports a secure communication between Cx Manager and Cx Engine based on SSL certificates.

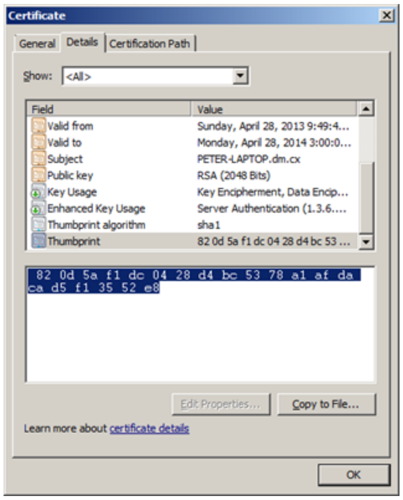
As the Cx Engine is working on WCF service that is not managed through the IIS console – the following steps describes the steps for configuring the secure connection on both Cx Manager and Cx Engine servers.

As the secure connection will be implemented between 2 servers only – it can be configured with Self Signed Certificates or real CA`s certificates – for generating Self Signed Certificates on the Cx Engine server, refer to Appendix 1.

On the Engine Server

Install the certificate to the Engine Server through the Certificates MMC – [Personal Container]. If self-signed certificate used verify that publisher (CxEngine server) is added to Trusted Publishers container].

1. Open the Certificate properties – Details tab.



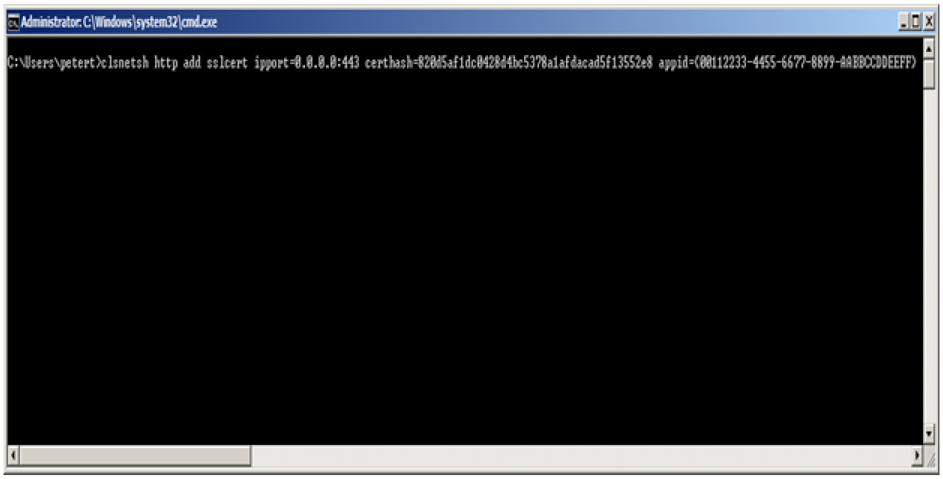
Show: <All>

Field	Value
Valid from	Sunday, April 28, 2013 9:49:4...
Valid to	Monday, April 28, 2014 3:00:0...
Subject	PETER-LAPTOP.dm.cx
Public key	RSA (2048 Bits)
Key Usage	Key Encipherment, Data Encip...
Enhanced Key Usage	Server Authentication (1.3.6....
Thumbprint algorithm	sha1
Thumbprint	82 0d 5a f1 dc 04 28 d4 bc 53 ...

```
 82 0d 5a f1 dc 04 28 d4 bc 53 78 a1 af da
ca d5 f1 35 52 e8
```

2. Copy certificate thumbprint (**Copy to notepad and delete spaces**)
3. Register certificate to port (**443** or other)
4. Run cmd
5. Run the next command with parameters:
   - **certhash** is thumbprint (step 3) with no spaces
   - **appid** is GUID (you may use the one in this example as well)

   netsh http add sslcert ipport=0.0.0.0:**443 certhash**=**820d5af1dc0428d4bc5378a1afdacad5f13552e8 appid**={00112233-4455-6677-8899-AABBCCDDEEFF}



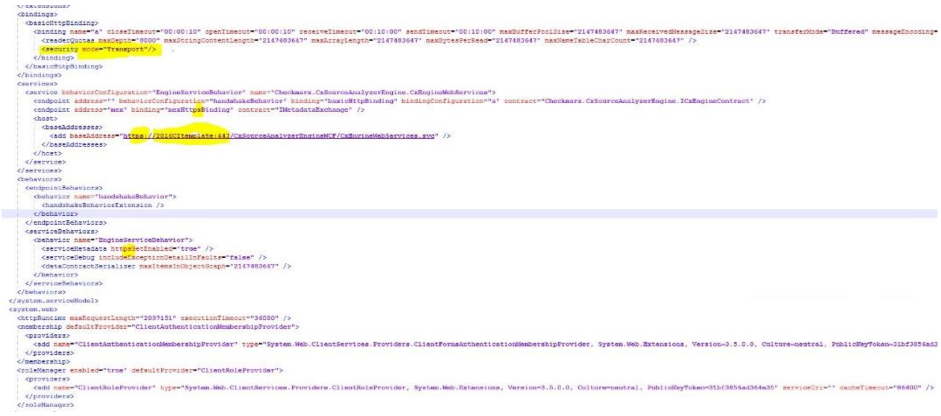
```
C:\Users\petert>clsnetsh http add sslcert ipport=0.0.0.0:443 certhash=820d5af1dc0428d4bc5378a1afdacad5f13552e8 appid={00112233-4455-6677-8899-AABBCCDDEEFF}
```

6. Back up and edit the Engine Configuration file as illustrated below:

   <Checkmarx Installation Folder>\Checkmarx Engine Server\CxSourceAnalyzerEngine.WinService.exe.config

   - Change binding security mode to “Transport” (should be added manually)
   - Should be written – <security mode="Transport"/> (Case Sensitive!)
   - Change service binding to “mexHttpsBinding”
   - Change baseAddress to https prefix and port
   - Change serviceMetadata to “httpsGetEnabled”.

Example:



```
</extensions>
<bindings>
  <basicHttpBinding>
    <binding name="a" closeTimeout="00:00:10" openTimeout="00:00:10" receiveTimeout="00:10:00" sendTimeout="00:10:00" maxBufferPoolSize="2147483647" maxReceivedMessageSize="2147483647" transferMode="Buffered" messageEncoding=
      <readerQuotas maxDepth="8000" maxStringContentLength="2147483647" maxArrayLength="2147483647" maxBytesPerRead="2147483647" maxNameTableCharCount="2147483647" />
      <security mode="Transport"/>
    </binding>
  </basicHttpBinding>
</bindings>
<services>
  <service behaviorConfiguration="EngineServiceBehavior" name="Checkmarx.CxSourceAnalyzerEngine.CxEngineWebServices">
    <endpoint address="" behaviorConfiguration="handshakeBehavior" binding="basicHttpBinding" bindingConfiguration="a" contract="Checkmarx.CxSourceAnalyzerEngine.ICxEngineContract" />
    <endpoint address="mex" binding="mexHttpsBinding" contract="IMetadataExchange" />
    <host>
      <baseAddresses>
        <add baseAddress="https://2016CItemplate:443/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc" />
      </baseAddresses>
    </host>
  </service>
</services>
<behaviors>
  <endpointBehaviors>
    <behavior name="handshakeBehavior">
      <handshakeBehaviorExtension />
    </behavior>
  </endpointBehaviors>
  <serviceBehaviors>
    <behavior name="EngineServiceBehavior">
      <serviceMetadata httpsGetEnabled="true" />
      <serviceDebug includeExceptionDetailInFaults="false" />
      <dataContractSerializer maxItemsInObjectGraph="2147483647" />
    </behavior>
  </serviceBehaviors>
</behaviors>
</system.serviceModel>
<system.web>
  <httpRuntime maxRequestLength="2097151" executionTimeout="36000" />
  <membership defaultProvider="ClientAuthenticationMembershipProvider">
    <providers>
      <add name="ClientAuthenticationMembershipProvider" type="System.Web.ClientServices.Providers.ClientFormsAuthenticationMembershipProvider, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad3
    </providers>
  </membership>
  <roleManager enabled="true" defaultProvider="ClientRoleProvider">
    <providers>
      <add name="ClientRoleProvider" type="System.Web.ClientServices.Providers.ClientRoleProvider, System.Web.Extensions, Version=3.5.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35" serviceUri="" cacheTimeout="86400" />
    </providers>
  </roleManager>
```

7. Restart Engine service.

If the Engine Service fails to start after completing the steps and the following errors are reported in the log file:

```
INFO - Trying to open WCF
ERROR - WCF was stopped to listen - trying to reopen
ERROR - WCF error: System.EventArgs
ERROR - WCF was stopped to listen - trying to reopen
ERROR - WCF error: System.EventArgs
```

Then perform the following steps:

```
Open cmd as an admin, and run the following command:
netsh http add urlacl
url=https://+:443/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc user="NT
AUTHORITY\NETWORK SERVICE"
```

Change the user running CxScanEngine service to NETWORK SERVICE and restart this service (If required)

The above step reserves the specified URL for non-administrator users and accounts.

On Management Server

1. Back up and edit the Job Manager Configuration file:

   In CxJobsManagerWinService.exe.config file (<CHECKMARX_INSTALL_DIR>\Checkmarx Jobs Manager\bin\CxJobsManagerWinService.exe.config), change the "Security mode" to **Transport**.

2. Back up and edit the Scan Manager Configuration file:

   In CxScansManagerWinService.exe.config file (<CHECKMARX_INSTALL_DIR>\Checkmarx Scan

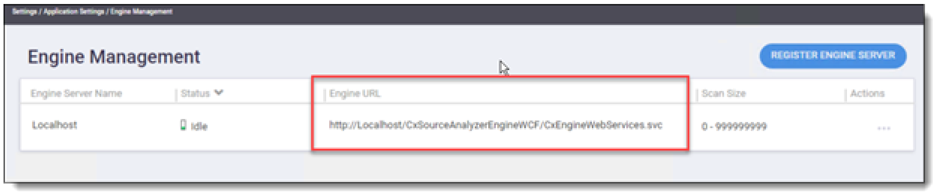
Manager\bin\CxScansManagerWinService.exe.config), change the "Security mode" to **Transport**.

3. Back up and edit the System Manager Configuration file:

   In CxSystemManagerService.exe.config file (<CHECKMARX_INSTALL_DIR>\Checkmarx System Manager\bin\CxSystemManagerService.exe.config), change the "Security mode" to **Transport**

```
  <security mode="Transport">
          <transport clientCredentialType="None" proxyCredentialType="None" 
realm="" />
          <message clientCredentialType="UserName" algorithmSuite="Default" />
   </security>
```

4. Start the web client.
5. Go to Management->Application Settings→Engine Managment.
6. Change the engine URL to HTTPS and set the server/subject name as shown in the certificate



Engine Management

Engine Server Name	Status	Engine URL	Scan Size	Actions
Localhost	Idle	http://Localhost/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc	0 - 999999999	...

7. Browse from Cx manager server to engine address with https using IE – If Certificate error is shown – install the certificate in the “Trusted Root Certification Authority” of the Cx Manager server.
8. Restart Manager service.

Appendix 1 – Self Sign Certificate

For information on creating a self-sign certificate, refer to - Create a Self-Signed Server Certificate in IIS

Enabling TLS 1.2 Support on the CxManager

Overview (TLS)

TLS (Transport Layer Security) and its now-deprecated predecessor, SSL (Secure Sockets Layer) are cryptographic protocols designed to provide communications security over a computer network. Websites can use TLS to secure all communications between their servers and web browsers. The TLS protocol aims primarily to provide privacy and data integrity between two or more communicating computer applications.

CxManager (TLS)

After configuring the CxManager for SSL support, the CxPortal may stop working after updating the CxWSResolver url in the CxPortal web.config to https:, this could mean that the environment requires TLS 1.2 support. This instruction defines the procedure for enabling TLS 1.2 support on the CxManager.

Enabling TLS 1.2 Support

Support for TLS 1.2 can be enabled via the Windows registry on the CxManager server. The enablement steps can be performed automatically or manually from the CxManager server.

Automated Enablement

1. Download the attached registry file (TLS1.2.reg) to the CxManager desktop.

   TLS1.2.reg - Registry Entries

   ```
   Windows Registry Editor Version 5.00
   [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319]
   "SchUseStrongCrypto"=dword:00000001
   [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.30319]
   "SchUseStrongCrypto"=dword:00000001
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2]
   [HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client]
   "DisabledByDefault"=dword:00000000
   "Enabled"=dword:00000001
   ```

2. Right click and select **Merge**.
3. Reboot the server.

Manual Enablement

1. Start the Windows registry editor: Start | Run or Search for "**regedit**".
2. Enable TLS 1.2 client keys:

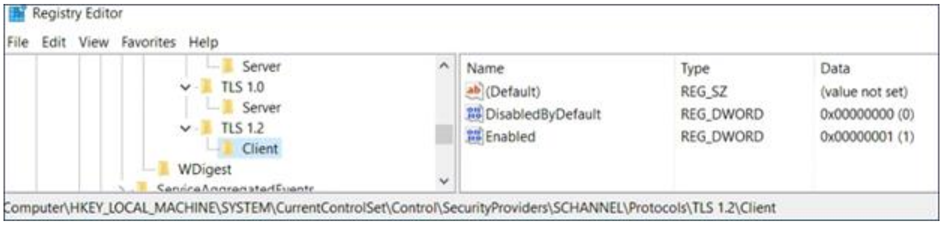
   a) Browse to the following registry key: **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols**

   b) Right click on the **Protocols** folder, select **New | Key**. Rename this key/folder **TLS 1.2**.

   c) Right click on the key/folder you just added, select **New | Key**. Rename this key/folder **Client**

d) Right click on the Client key/folder, select **New | DWORD (32-bit) Value**. Rename the DWORD to **DisabledByDefault**. Ensure that the value is set to **0**.

e) Right click on the Client key/folder, select **New | DWORD (32-bit) Value**. Rename the DWORD to **Enabled**. Set the value to **1**.



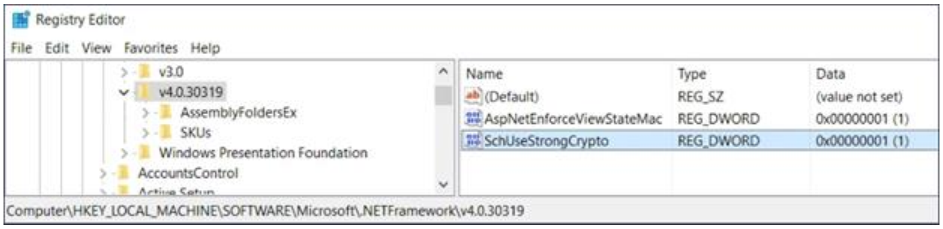
Name	Type	Data
(Default)	REG_SZ	(value not set)
DisabledByDefault	REG_DWORD	0x00000000 (0)
Enabled	REG_DWORD	0x00000001 (1)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client

3. Configure .NET to use TLS 1.2

a) In regedit, browse to the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NetFramework\v4.0.30319

b) Right click in the right pane and create a new **DWORD (32-bit) Value**. Rename the DWORD to **SchUseStrongCrypto**. Set the value to **1**.



Name	Type	Data
(Default)	REG_SZ	(value not set)
AspNetEnforceViewStateMac	REG_DWORD	0x00000001 (1)
SchUseStrongCrypto	REG_DWORD	0x00000001 (1)

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319

c) Browse to the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NetFramework\v4.0.30319

d) Repeat step 3.b.

4. Reboot the server.

Configuring Management & Orchestration for SSL and FIPS Compliancy

This instruction defines the procedure for configuring Management & Orchestration for SSL and FIPS compliancy for CxSAST v9.0.0 and higher.

Management and Orchestration should be defined according to the CxSAST secure communication protocol definition. This means that If CxSAST is defined for HTTPS then Management and Orchestration should also be defined in the same way.

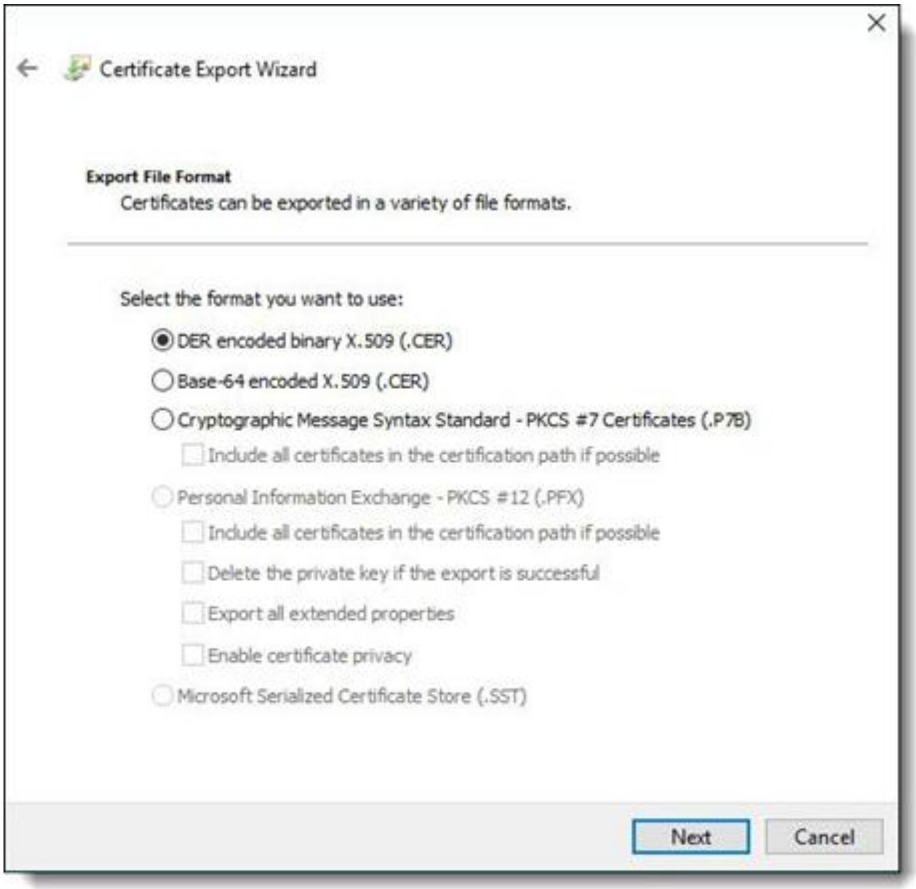
The following instructions for running the Management and Orchestration over HTTPS offers a general procedure for configuring HTTPS in Apache Tomcat. If you require more specific instructions, please refer to the Apache Tomcat documentation.

Configuring HTTPS in Apache Tomcat

➢ **To configure HTTPS in Apache Tomcat:**

1. In order to use the same certificate export the certificate from the IIS as a .pfx file (use the same password that you will use in the Tomcat server.xml file). See here for instructions.
2. Go to M&O Apache Tomcat configuration folder, for example: \Checkmarx\Checkmarx Risk Management\Tomcat\conf
3. In the server.xml configuration file, add the following connector:

   ```
   <Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
   disableUploadTimeout="true" enableLookups="false" maxThreads="25"
   port="8443" keystoreFile=<Absolute path to the pfx file> keystorePass=<The
   password used when exported the certificate> keystoreType="PKCS12"
   protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
   secure="true" sslProtocol="TLS" />
   ```
4. Export the certificate from the browser as a .CER file (select the DER encoded binary X.509 format).



Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

● DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

○ Personal Information Exchange - PKCS #12 (.PFX)

□ Include all certificates in the certification path if possible

□ Delete the private key if the export is successful

□ Export all extended properties

□ Enable certificate privacy

○ Microsoft Serialized Certificate Store (.SST)

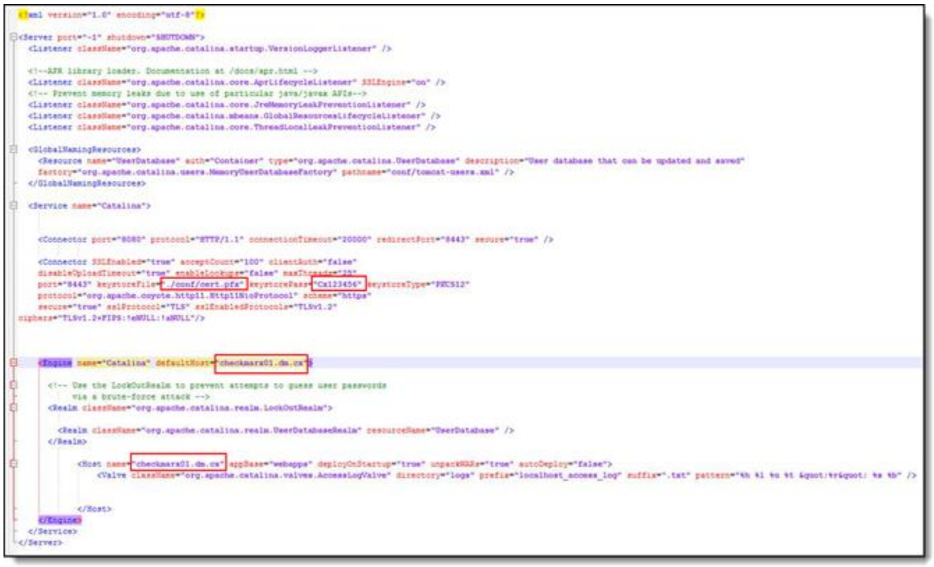
5. Save the .CER file in the following folder: C:\Program Files\Checkmarx\Checkmarx Risk Management\jre\lib\security, and then import the certificate into the cacerts file in order to create a chain of trust by entering the following cmd line:

   <keytool location>keytool" -import -alias <alias name> -file <cer file location>MnOManagerH03.cer" -keystore "cacerts

For example:

C:\Program Files\Java\jdk1.8.0_201\bin\keytool" -import -alias MnOManagerH03 -file "C:\Program Files\Checkmarx\Checkmarx Risk Management\jre\lib\security\MnOManagerH03.cer" -keystore "cacerts

6. The default password for the cacerts file is changeit
7. Edit the following configuration file in the Checkmarx\Checkmarx Risk Management\tomcat\conf\server.xml:
    - keystoreFile – path to the .pfx file
    - keystorePass – password used when exporting the certificate
    - <host name> – change it in both places to the M&O Server host name



```
<?xml version="1.0" encoding="utf-8"?>

<Server port="-1" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />

  <!--APR library loader. Documentation at /docs/apr.html -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs-->
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container" type="org.apache.catalina.UserDatabase" description="User database that can be updated and saved"
    factory="org.apache.catalina.users.MemoryUserDatabaseFactory" pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <Service name="Catalina">


    <Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" secure="true" />

    <Connector SSLEnabled="true" acceptCount="100" clientAuth="false"
    disableUploadTimeout="true" enableLookups="false" maxThreads="25"
    port="8443" keystoreFile="./conf/cert.pfx" keystorePass="Cx123456" keystoreType="PKCS12"
    protocol="org.apache.coyote.http11.Http11NioProtocol" scheme="https"
    secure="true" sslProtocol="TLS" sslEnabledProtocols="TLSv1.2"
ciphers="TLSv1.2+FIPS:!eNULL:!aNULL"/>


    <Engine name="Catalina" defaultHost="checkmarx01.dm.cx">

      <!-- Use the LockOutRealm to prevent attempts to guess user passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">

        <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase" />
      </Realm>

          <Host name="checkmarx01.dm.cx" appBase="webapps" deployOnStartup="true" unpackWARs="true" autoDeploy="false">
            <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />

          </Host>
    </Engine>
  </Service>
</Server>
```

Configuring FIPS Compliancy

In order to make the Apache Tomcat FIPS compliant, the SSL connector in Tomcat needs to be updated with FIPS-compliant ciphers limitation.

To demonstrate, the following example of SSL connector without ciphers should be changed to the example below:

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" connectionTimeout="20000" maxThreads="200" scheme="https" secure="true" SSLEnabled="true" keystoreFile="C:\Program Files\OpenSSL-FIPS\out\iis-cert.pfx" keystorePass="Cx123456" clientAuth="false" sslProtocol="TLS">
<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
</Connector>

Example of how it should be changed (changes in red):

<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol" connectionTimeout="20000" maxThreads="200" scheme="https" secure="true" SSLEnabled="true" keystoreFile="C:\Program Files\OpenSSL-

```
FIPS\out\iis-cert.pfx" keystoreType="PKCS12" keystorePass="Cx123456" clientAuth="false" sslProtocol="TLS"
sslEnabledProtocols="TLSv1.2" ciphers="TLSv1.2+FIPS:!eNULL:!aNULL">
<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
</Connector>
```

Enabling TLS Protocol Connection to the ActiveMQ

These instructions define the procedure for enabling the TLS protocol connection to the ActiveMQ.

TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are cryptographic protocols designed to provide communication security over networks. Websites can use TLS to secure all communications between their servers and web browsers. TLS aims primarily to provide privacy and data integrity between two or more communicating applications.

ActiveMQ supports secure communication channels. The most common way to establish a secure communication channel is to associate a certificate with the target (broker). This section provides instructions on how to enable the TLS protocol connection to the ActiveMQ. The instructions include links to topics that are directly related to this procedure.

Upgrading the ActiveMQ from CxSAST 9.0.0 to CxSAST 9.2.0

To upgrade the ActiveMQ for use with CxSAST v9.2.0, you have to follow the steps listed below. These instructions assume that you already upgraded CxSAST to v9.2.0.

1. Make sure that you have your custom cxuser password and the AMQ TLS available.
2. Copy the cx.keystore.password field from the credentials-enc_backup file to the credentials-enc.properties file
3. Copy <sslContext> and the <transportConnectors> nodes from activemq_backup.xml to activemq.xml
4. Restart ActiveMQ Service.
5. Restart Cx Services for the upgrade to take effect.

Handling ActiveMQ Broker Certificates

1. Navigate to the Checkmarx ActiveMQ\conf folder.
2. Enter **cmd** in the Search field and execute the relevant command according to the respective scenarios listed below:
    - Enter **cmd** in the Search field and execute the relevant command according to the respective scenarios listed below:

```
keytool -importkeystore -srckeystore cert.pfx -srcstoretype pkcs12 -destkeystore server.ks -deststoretype JKS
```

- Create the broker certificate (self-signed):

```
keytool -genkey -alias amq-server -keyalg RSA -keysize 2048 -validity 999 -keystore server.ks
```

- Export the broker certificate:

```
keytool -export -alias amq-server -keystore server.ks -file broker_cert
```

- Import the self-signed certificate to a client as explained below.

➢ **To install the Windows Certificate Store:**

- Copy the certificate to the client station and then double-click, or install using the Certificate Management tool (**certmgr.msc**).

➢ **Configuring to load from file:**

- Copy **broker_cert** to the client station and then use the path below in the ULR encoded form for the connection string. For example, if a certificate was placed in **c:\certificates\broker_cert**, the connection string would look similar to the following:

```
ssl://activemq:61617?transport.BrokerCertFilename=c%3A%5Ccertificates%5Cbroker_cert
```

The self-signed certificates must be installed on each client machine.

Enabling TLS Protocol Connection

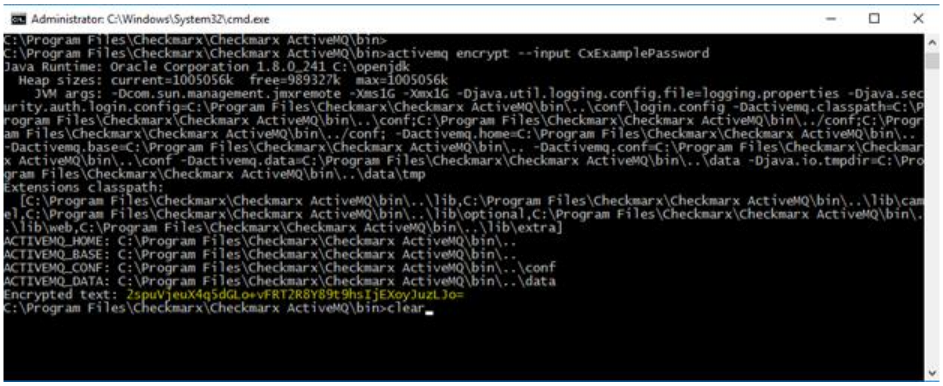
The TLS protocol connection to the ActiveMQ can be enabled by first configuring the ActiveMQ broker via a configuration file (activemq.xml) and then updating the ActiveMQ client configuration via the database (CxDB).

Configuring the ActiveMQ Broker

Once the CxSAST environment is set up and fully configured, perform the following:

1. Navigate to the **Checkmarx ActiveMQ\conf** folder and open **activemq.xml**.
2. Edit the <sslContext> tag accordingly:

```
<sslContext>
    <sslContext keyStore="file:${activemq.conf}/server.ks"
keyStorePassword="${cx.keystore.password}" />
</sslContext>
```



```
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>activemq encrypt --input CxExamplePassword
Java Runtime: Oracle Corporation 1.8.0_241 C:\openjdk
  Heap sizes: current=1005056k  free=989327k  max=1005056k
    JVM args: -Dcom.sun.management.jmxremote -Xms1G -Xmx1G -Djava.util.logging.config.file=logging.properties -Djava.sec
urity.auth.login.config=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf\login.config -Dactivemq.classpath=C:\P
rogram Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf;C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\../conf;C:\Progr
am Files\Checkmarx\Checkmarx ActiveMQ\bin\../conf; -Dactivemq.home=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
-Dactivemq.base=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\.. -Dactivemq.conf=C:\Program Files\Checkmarx\Checkmar
x ActiveMQ\bin\..\conf -Dactivemq.data=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data -Djava.io.tmpdir=C:\Pro
gram Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data\tmp
Extensions classpath:
  [C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\cam
el,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\optional,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\.
.\lib\web,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\extra]
ACTIVEMQ_HOME: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
ACTIVEMQ_BASE: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
ACTIVEMQ_CONF: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf
ACTIVEMQ_DATA: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data
Encrypted text: 2spuVjeuX4q5dGLo+vFRT2R8Y89t9hsIjEXoyJuzLJo=
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>clear_
```

3. Navigate to **C:\Program Files\Checkmarx\Checkmarx ActiveMQ\conf** and edit **credentials-enc.properties**.
4. Add a new variable in the same format as the existing value to the text file with the encrypted text between the brackets, for example **keystore.password=ENC(2spuVjeuX4q5dGLo+vFRT2R8Y89t9hsljEXoyJuzLJo=)**
5. Navigate to **C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin** and open a command line window (cmd instance) for that folder.
6. Use the ActiveMQ encryption utility to encrypt your password by entering the following command in the command line: **bat encrypt –input CxExamplePassword**
7. Copy the encrypted text (yellow fonts in the screen image below).

- Additional information about certificate manipulation in JAVA is available in the Oracle documentation: https://docs.oracle.com/javase/8/docs/technotes/tools/unix/keytool.html. Some examples can be found in the Create Broker Certificate section.
- Keystore passwords must be encrypted. For detailed instructions, go to https://activemq.apache.org/encrypted-passwords.

8. Edit the <transportConnectors> tag accordingly:

   <transportConnectors>

   <transportConnector name="ssl" uri="ssl://0.0.0.0:61617?transport.enableProtocols=TLSv1.2,TLSv1.3&amp;transport.enableCipherSuites=TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,ECDHE-ECDSA-AES128-GCM-SHA256,ECDHE-RSA-AES128-GCM-SHA256,ECDHE-ECDSA-AES256-GCM-SHA384,ECDHE-RSA-AES256-GCM-SHA384,ECDHE-ECDSA-CHACHA20-POLY1305,ECDHE-RSA-CHACHA20-POLY1305,DHE-RSA-AES128-GCM-SHA256,DHE-RSA-AES256-GCM-SHA384"/>

   </transportConnectors>

- It is required to use TLS v1.2 (and higher) and relevant Cipher suites:
  - TLS 1.2:
    ECDHE-ECDSA-AES128-GCM-SHA256, ECDHE-RSA-AES128-GCM-SHA256, ECDHE-ECDSA-AES256-GCM-SHA384, ECDHE-RSA-AES256-GCM-SHA384,

ECDHE-ECDSA-CHACHA20-POLY1305, ECDHE-RSA-CHACHA20-POLY1305,DHE-RSA-AES128-GCM-SHA256, DHE-RSA-AES256-GCM-SHA384

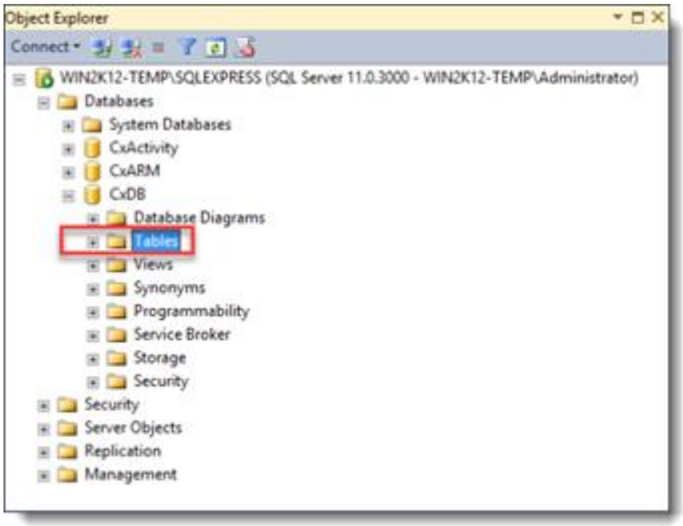
- TLS 1.3:
  TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384,
  TLS_CHACHA20_POLY1305_SHA256

9. Define additional connection parameters as follows:
   parameter=value&amp;parameter2=list_valueA, list_valueB
10. Repeat this procedure for every ActiveMQ broker, if High Availability cluster setup is used.
11. Place a certificate provided by the Certification Authority of your organization, or a public Certification Authority in the ActiveMQ certificate store.

Once you have completed the ActiveMQ broker configuration, you can configure the ActiveMQ clients.

Configuring ActiveMQ Clients

1. Open **MS SQL Server Management Studio**.
2. Connect to the SQL server.
3. Navigate to **Databases > CxDB > Tables**.



4. Expand the Tables repository to view its content and navigate to **dbo.CxComponentConfiguration** .
5. Right-click **dbo.CxComponentConfiguration** and then select **Edit Rows**.
6. In the 'ActiveMessageQueueURL' key field, enter the relevant URL in the Value field according to one of the following scenarios:
   - For certificates provided by a trusted authority:

   ssl://activemq:61617

- For self-signed certificate imported to the certificate store under Current User/Enterprise Trust:

ssl://activemq:61617?transport.KeyStoreName=Enterprise%20Trust

- For self-signed certificate placed on a file system at **c:/certificates/broker_cert**

ssl://activemq:61617?transport.BrokerCertFilename=c%3A%5Ccertificates%5Cbroker_cert

7. In the Tables repository, right-click **Config.CxEngineConfigurationKeysMeta** and select **Edit Rows.**
8. In the 'ACTIVE_MESSAGE_QUEUE_URL' key field, enter the relevant URL in the Value field according to one of the following scenarios:

- For certificate provided by trusted authority:

ssl://activemq:61617

- For self-signed certificate imported to certificate store under Current User/Enterprise Trust:

ssl://activemq:61617?transport.KeyStoreName=Enterprise%20Trust

- For self-signed certificate placed on a file system at c:/certificates/broker_cert:

ssl://activemq:61617?transport.BrokerCertFilename=c%3A%5Ccertificates%5Cbroker_cert

➢ **To configure a secure channel:**

1. In the connection string, first update the URL (ssl://) and then configure the socket parameters using the relevant 'transport' prefix according to one of the following scenarios:

- Central key store related parameters are as follows:

transport.KeyStoreLocation=LocalMachine

This indicates the Key store location. Known locations are; 'LocalMachine' or 'CurrentUser'. By default 'CurrentUser' is used.

transport.KeyStoreName=Enterprise%20Trust

This indicates the Key store name. By default 'Personal' is used [*]

transport.ServerName=<name>

This indicates the name of the Server's Certificate. By default, the Host name of the remote server is used [**]

- File key store related parameter (used with self-signed certificates) is as follows:

transport.BrokerCertFilename=broker_cert

This indicates the location of the broker Certificate to use when the broker uses a self-signed certificate [*] [***]

- [*] Parameter value should be ULR encoded
- [**] Disabling server name verification is not recommended
- [***] High Availability setup broker certificate must be deployed to all clients in a similar environment.

- Step 4 and 5 above can be performed automatically by using the following DB TSQL script:

```
DECLARE @AmqString varchar(1000)
SET @AmqString =
'ssl://activemq:61617?transport.KeyStoreName=Enterprise%20Trust'
Update [CxDB].[dbo].CxComponentConfiguration
set [Value] = @AmqString
where [Key] = 'ActiveMessageQueueURL'
Update [CxDB].[Config].[CxEngineConfigurationKeysMeta]
set [DefaultValue] = @AmqString
where [KeyName] = 'ACTIVE_MESSAGE_QUEUE_URL'
```

Changing the Server Name, IP or Port for Checkmarx Components

Overview

A number of additional components have been introduced to the latest versions of CxSAST (v9.0) that includes Access Control, Management & Orchestration, and ActiveMQ. The definition values for these components are saved in the database (CxDB) as {Protocol}://{FQDN}:{Port}. Fully Qualified Domain Name (FQDN) is the complete domain name for the machine and consists of the hostname and domain name (e.g. http://mqserver.company.com:5555). If you want to rename the server, change the IP or change the port, you will need to change the key value definitions in the relevant database tables.

Changing the Server Name, IP or Port

Changing the server name, IP or port for Checkmarx Components can be performed via the relevant database table and then through the Web Portal. These steps can be performed manually for each server:

1. Insert the new server definitions into the Domain Name System (DNS).
2. Open MS SQL Server Management Studio.
3. Go to: **Databases** > **CxDB** > **Tables** > **dbo.CxComponentConfiguration** and update the following database key values:
   - **ActiveMessageQueueURL**- {Protocol}://{**Server Name**, **IP**}:{**Port**}
   - **CxARMPolicyURL** - {Protocol}://{**Server Name**, **IP**}:{**Port**}
   - **CxARMURL** - {Protocol}://{**Server Name**, **IP**}:{**Port**}
   - **IdentityAuthority** - {Protocol}://{**Server Name**, **IP**}:{**Port**}/CxRestAPI/auth
   - **EX_SOURCE_PATH** - {dir}:\{CxSrc folder}
   - **SOURCE_PATH** - {dir}:\{CxSRC folder}
   - **WebServer** - {Protocol}://{**Server Name**, **IP**}:{**Port**}
4. Go to: **Databases** > **CxDB** > **Tables** > **CxARM** > **dbo.DBSources** and also update the **DB_HOST** database key value.
5. For each server, update the server name, the IP address or port in the relevant configuration file:
   - For CxManagers/Web Portals:
     - Go to **C:\Program Files\Checkmarx\CheckmarxWebPortal\Web**, open the **web.config** file for editing and using the Search tool, search for '**CxWSResolver.CxWSResolver**' and update accordingly.
     - Go to **C:\Program Files\Checkmarx\Configuration**, open the **DBConnectionData.config** file for editing and using the Search tool, search for '**Data Source**' and update accordingly.
   - For Management and Orchestration (if installed):
     - Go to **C:\Program Files\Checkmarx\Checkmarx Risk Management\Config**, open the **db.properties** file for editing and using the Search tool, search for '**DB_HOST**' and update accordingly.
   - For CxEngines:
     - Go to **C:\Program Files\Checkmarx\Checkmarx Engine Server**, open the **CxSourceAnalyzerEngine.WinService.exe.config** file for editing and using the Search tool, search for '**baseAddress**' and update accordingly.

➢ **To update the Java version or Java path, apply the correct Java settings as follows:**

1. Right-click **This PC** and select **Properties** to display the System information and settings.
2. Go to **Advanced System Settings** to open the System Properties dialog box.

3. If not already open, open the Advanced tab.
4. On the Advanced tab, click **Environment Variables...** to open the Environment Variables dialog box.
5. Navigate to **CX_JAVA_HOME** to point to where the JRE folder is located, for example **C:\Program Files\Java\jdk1.8.0_241\jre**.
6. Update the Load Balancer configuration file (e.g. Nginx.conf) accordingly.
7. Update Access Control by configuring the **appsettings.json** file (**<dir>:\Program Files\Checkmarx\Checkmarx Access Control\appsettings.json**):
8. Update ExternalListenUrls to reflect IIS configured bindings:

   http(s)://*(port)
   e.g. "ExternalListenUrls": "https://*:433"

   If more than one binding is configured:

   http(s)://*(port);http(s)://*(port)
   e.g. "ExternalListenUrls": "https://*:433;https://*:123"

   Restart all Cx Windows Services and the IIS.

   Log in to the Web Portal and manually update the CxEngine Server name, IP or port according to these instructions.
9. If required, update the IP Address in the TCP/IP Properties in the SQL Server Configuration Manager (see example).

Configuring CxSAST to Use a Non-Default Port

By default, CxSAST uses port 80 for communications. You can change the port, for example to port 8080.

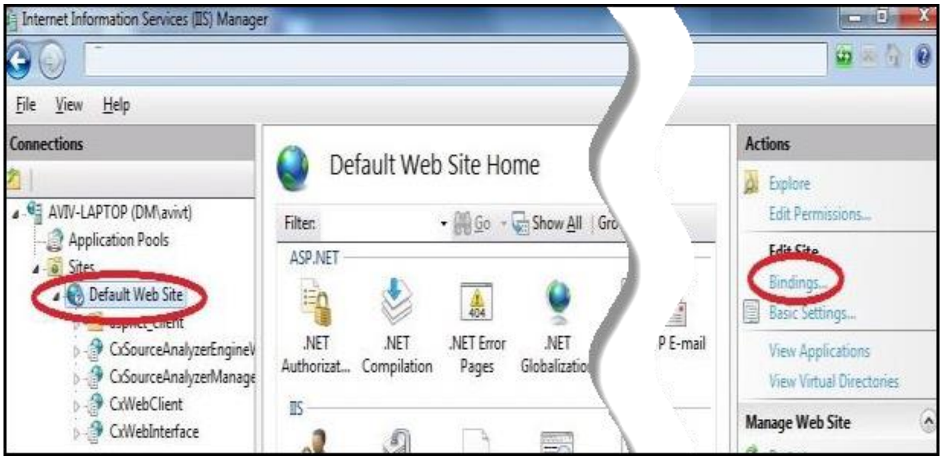
To change the CxSAST communication port, you need to first change the web server listening port, and then perform additional configuration on CxSAST, as explained in the sections below.

Changing the Web Server Listening Port

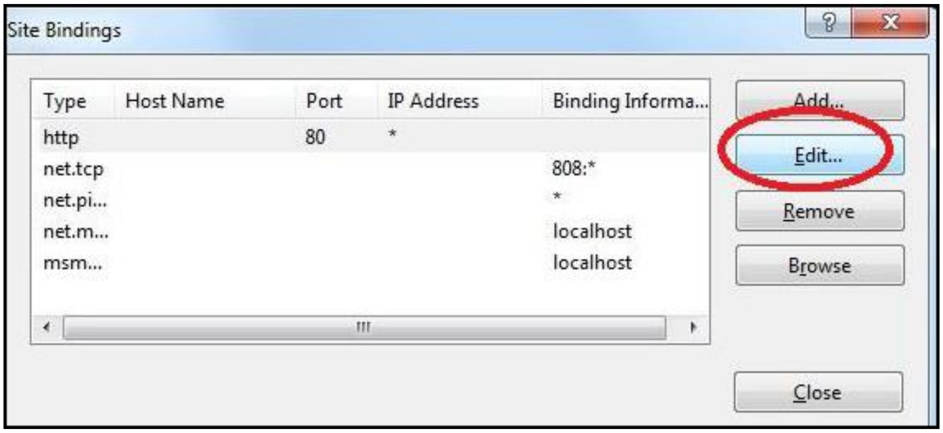
Change the web server listening port according to one of the following procedures, depending on which web server CxSAST is using:

➢ **To change the port on IIS:**

1. On the CxSAST Server or CxManager host, open the IIS Manager.
2. In the left-hand navigation pane, select **Sites** > **Default Web Site**.
3. On the right, under **Actions** > **Edit Sites**, click **Bindings**:

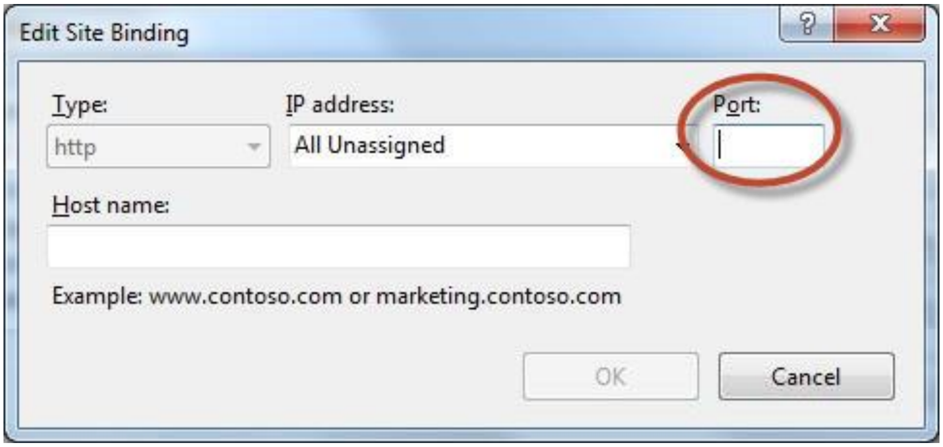


4. Select http and click Edit:


Site Bindings

Type	Host Name	Port	IP Address	Binding Informa...
http		80	*	
net.tcp				808:*
net.pi...				*
net.m...				localhost
msm...				localhost

5. Under **Port**, enter the new port number:


Edit Site Binding

Type: http

IP address: All Unassigned

Port:

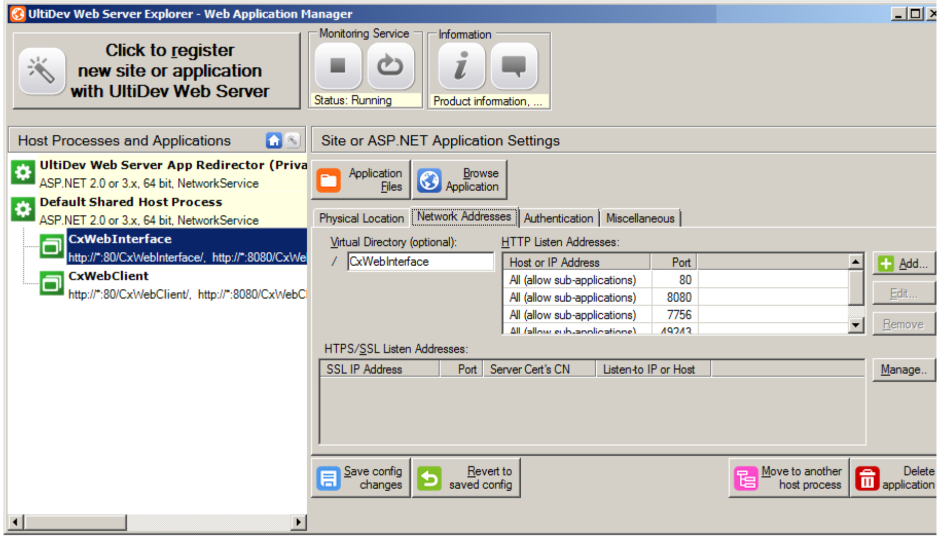
Host name:

Example: www.contoso.com or marketing.contoso.com

6. Click **OK**.

➢ **To change the port on UltiDev:**

1. On the CxSAST Server or CxManager host, open the **UltiDev Web App Explorer** (for example, from the Windows/Start menu).
2. On the left, select **CxWebInterface**, and in the **Network Addresses** tab, click **Add** and add the new port; select **80** and click **Remove**:



UltiDev Web Server App Redirector (Priva
ASP.NET 2.0 or 3.x, 64 bit, NetworkService
Default Shared Host Process
ASP.NET 2.0 or 3.x, 64 bit, NetworkService
CxWebInterface
http://*:80/CxWebInterface/, http://*:8080/CxWe
CxWebClient
http://*:80/CxWebClient/, http://*:8080/CxWebC

Virtual Directory (optional): / CxWebInterface

HTTP Listen Addresses:

Host or IP Address	Port
All (allow sub-applications)	80
All (allow sub-applications)	8080
All (allow sub-applications)	7756
All (allow sub-applications)	49243

HTPS/SSL Listen Addresses:

SSL IP Address	Port	Server Cert's CN	Listen-to IP or Host

3. Select **CxWebClient**, and in the **Network Addresses** tab, click **Add** and add the new port; select **80** and click **Remove.**
4. Click **Save config changes.**

Additional CxSAST Configuration

Now that that the web server listening port is configured, perform the following CxSAST configuration:

1. Open the following file for editing:
   C:\Program Files\Checkmarx\CheckmarxWebPortal\Web\web.config
2. Find the key **CxWSResolver.CxWSResolver**, and change the line to:
   <add key="CxWSResolver.CxWSResolver"
   value="http://localhost:<port>**/CxWebInterface/CxWSResolver.asmx" />**
   where <port> is the new port number. For example:
   <add key="CxWebServices.CxWSResolver"
   value="http://localhost:8080/CxWebInterface/CxWSResolver.asmx" />
3. Open the following file for editing:
   C:\Program Files\Checkmarx\Checkmarx Engine
   Server\CxSourceAnalyzerEngine.WinService.exe.config
4. Open the following file for editing: **"<Checkmarx Installation Folder>\Checkmarx Engine Server\CxSourceAnalyzerEngine.WinService.exe.config**".

5. Find the phrase "**http://**" and change the line to: **<add baseAddress="http://localhost:<port>/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc"/>** where <port> is the new port number. For example: **<add baseAddress="http://localhost:8080/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc"/>"**
6. Run the following command in elevated CMD with correct parameters:
   netsh http add urlacl url=http://+:80/CxSourceAnalyzerEngineWCF/CxEngineWebServices.svc user="NT AUTHORITY\NETWORK SERVICE"

- The number 80 in - ".....=http://+:80/Cx...."
  The domain\user in - "....vc user="NT AUTHORITY\NETWORK SERVICE" if the user running the Cx Engine service is not the default "Network Service"

7. In order for CxARM, plugins and the CLI tool to work with the new port, you need to run the following DB query::
   UPDATE [CxDB].[dbo].[CxComponentConfiguration]
   SET Value = '**http(s)://your_cx_portal_hostname:port**'
   WHERE [Key] = 'IdentityAuthority'
8. In the Windows Service Manager, restart the following services:
   CxScanEngine
   CxScanManager
9. In a distributed deployment:

   a) Log into the CxSAST web interface, using the new port number in the browser address bar. For example:
   http://localhost:8080/CxWebClient/login.aspx
   b) Go to **Management** > **Server Settings** > **Installation Information**, and under **Engine Servers**, edit the server address to include the new port number.

10. If you use CxAudit:

    a) On the CxSAST server, open the following file for editing:
    C:\Program Files\Checkmarx\Checkmarx Audit\CxAudit.exe.config
    b) Find the **SERVER_ADDRESS** key, and edit its **value** to include the new port number. For example:
    <add key="SERVER_ADDRESS" value="http://localhost:8080" />
    c) If CxAudit was already run, repeat the previous step on the following file:
    C:\Users\USERNAME\AppData\Local\Checkmarx\CxAudit\CxAudit.exe.config

Configuring CxSAST for use with a non-default user (Network Service) CxServices & IIS Application Pools

The CxSAST is based on IIS Application Pools and CxServices. For each of these the local Network Service is defined as default by the CxSAST installer.

By default, all product services are installed and configured to run with Windows Network Service account. Thiese instructions describes how to configure CxSAST for use with a non-default user (Network Service) that includes CxServices & IIS Application Pools.

Outline

It is important to differentiate between the components as not all of them are used for the same purpose.

IIS Application Pools

- **CxClientPool** - Application Pool of the Web Portal - the user that is defined here will not influence any external tools.
- **CxPool** - Application Pool of the CxManager - the user that is defined here is used for connection to a third party server, e.g. TFS, GIT, SVN, etc..
- **CxPoolRestAPI** - Application Pool of the RestAPI - the user that is defined here will not influence any external tools.
- **CxAccessControl -** Application Pool of the Access Control portal

The user assigned to the IIS App Pools must have access to the CxDB and CxActivity databases in the SQL Server, if **'Integrated Security=True'** in the
- DBConnectionData.config file.

Services

Services for the CxManager - the user that is defined here (and to the CxPool) is used for the connection to the SQL server (if 'Integrated Security=True' in
- DBConnectionData.config file):

- **CxSystemManager**
- **CxJobsManager**
- **CxScansManager**

- CxSastResults
- CxScanEngine
- Web server:
  - World Wide Web Publishing Service
  - IIS Admin Service
  - Access Control

- Management and Orchestration:
  - CxARM
  - CxARMETL
  - CxRemediationIntelligence

- Shared services:
  - ActiveMQ – Message Broker (Apache message queue broker) for communicating between Checkmarx products

Service for the CxEngine. The user that is defined here is required to be in the Administrators group of the server or (recommended!) - run the netsh command for this user

- CxScanEngine

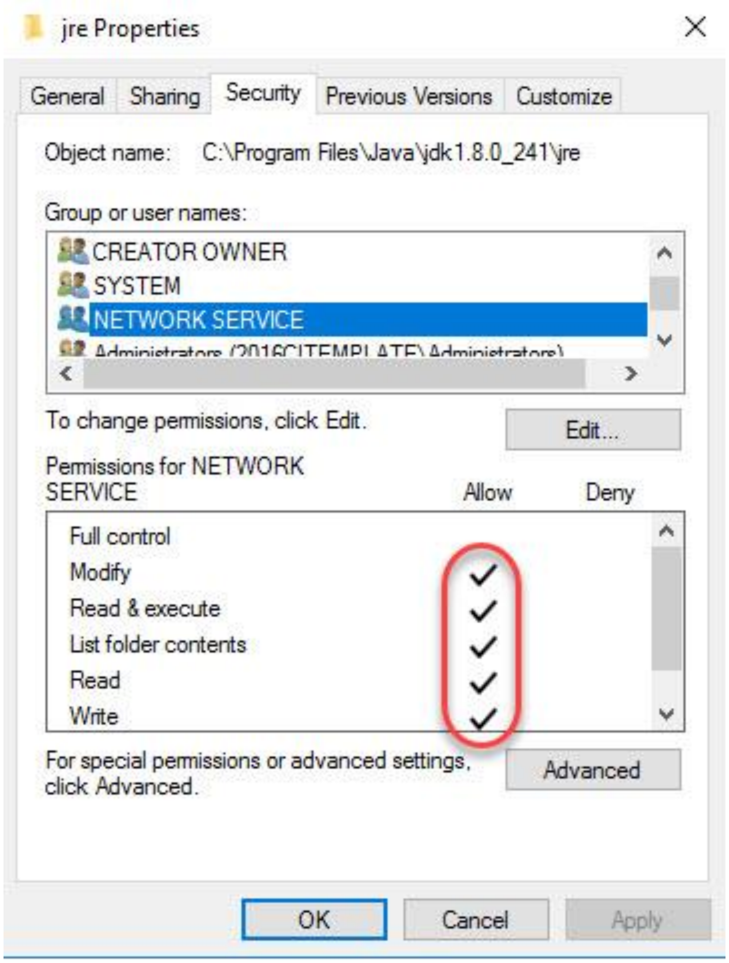
> - For resolving issues, it is recommended to keep all the CxServices defined with same user.

Java Folders

In order to allow Cx services to read and write in the Java folder when users use their own non-default service account, the relevant must grant Read/Write/Modify permissions to the Java folders (**<root directory>:\Program Files\Java\jdk1.8.0_241\jre**).

➢ **To grant permissions on the Java folders:**

1. Ensure that the relevant user is logged on to the station with admin rights.
2. Navigate to the **jdk1.8.0_241** folder, which is usually located at **C:\Program Files\Java\jdk1.8.0_241**
3. Right-click **jre** and select **Properties** from the menu to open the jre Properties dialog.



jre Properties

Object name: C:\Program Files\Java\jdk1.8.0_241\jre

Group or user names:

- CREATOR OWNER
- SYSTEM
- NETWORK SERVICE
- Administrators (2016CITEMPLATE\Administrators)

To change permissions, click Edit.

Permissions for NETWORK SERVICE	Allow	Deny
Full control		
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

For special permissions or advanced settings, click Advanced.

4. Navigate to the desired non-default service (not the default **Network Services**) and add permissions to read, write and modify. The permission profile must look as illustrated in the jre Properties screen image above.
5. Apply the new settings and close the folders.

Storage Folders

Ensure that the user who accesses the Cx storage folders (**CxSrc**, **CxReports**, **ExtSrc**) has the appropriate read/write permissions.

Configuration

CxServices

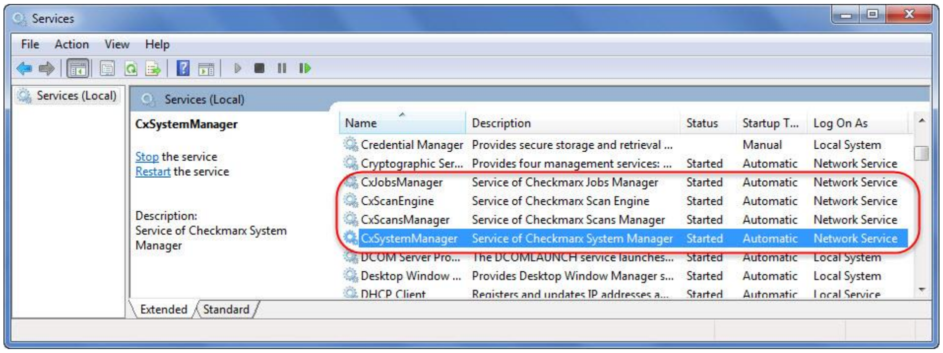
1. Ensure that the user running the CxServices has the appropriate authorization, i.e. has domain access, administration rights, etc.

2. In the Service Manager (**services.msc**), check the **Log On As** user accounts for each of the following:

    - **CxJobsManager**
    - **CxScanEngine**
    - **CxScansManager**
    - **CxSystemManager**
    - **CxARM**
    - **CxARMETL**
    - **CxRemediationIntelligence**
    - **ActiveMQ**

- If any of the CxServices are anything other than the default **Network Service**, make sure you know the user account's full credentials.

3. Open Windows Services:



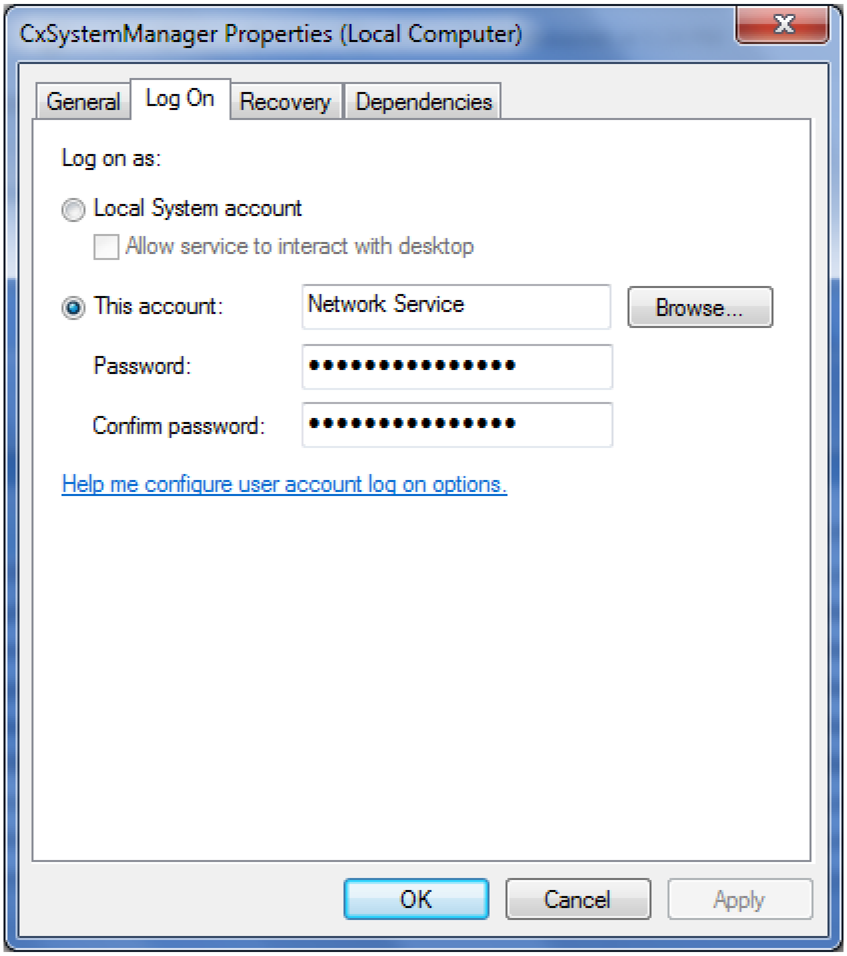
CxSystemManager

Stop the service
Restart the service

Description:
Service of Checkmarx System Manager

Name	Description	Status	Startup T...	Log On As
Credential Manager	Provides secure storage and retrieval ...		Manual	Local System
Cryptographic Ser...	Provides four management services: ...	Started	Automatic	Network Service
CxJobsManager	Service of Checkmarx Jobs Manager	Started	Automatic	Network Service
CxScanEngine	Service of Checkmarx Scan Engine	Started	Automatic	Network Service
CxScansManager	Service of Checkmarx Scans Manager	Started	Automatic	Network Service
CxSystemManager	Service of Checkmarx System Manager	Started	Automatic	Network Service
DCOM Server Pro...	The DCOMLAUNCH service launches...	Started	Automatic	Local System
Desktop Window ...	Provides Desktop Window Manager s...	Started	Automatic	Local System
DHCP Client	Registers and updates IP addresses a...	Started	Automatic	Local Service

4. Right click on a **CxService** and select **Properties**.


CxSystemManager Properties (Local Computer)
General
Log On
Recovery
Dependencies
Browse...
Help me configure user account log on options.
OK
Cancel
Apply

Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: Network Service

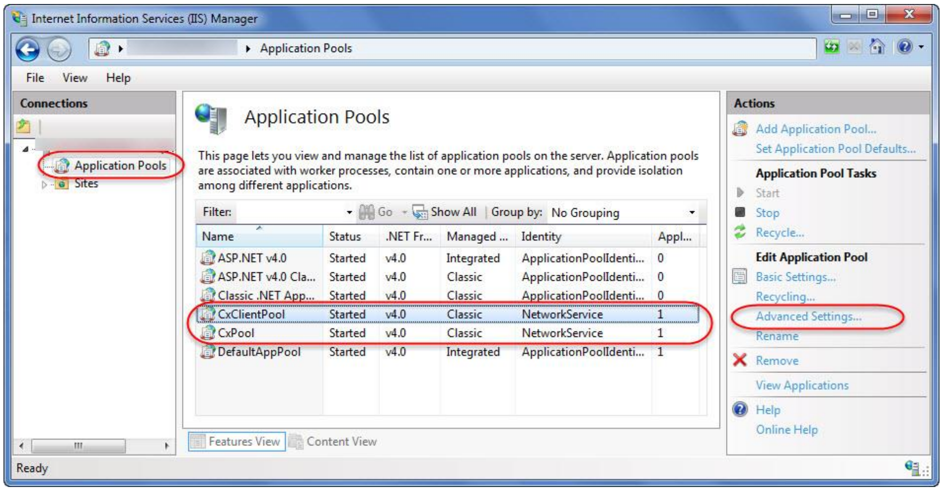
Password: ••••••••••••••••

Confirm password: ••••••••••••••••

5. Select the **Log On** tab, enter the appropriate user credentials and click **OK**.

IIS (Application Pools)

1. In the IIS Manager, navigate to **Application Pools**, and check the user **Identity** of each of the following:
   - **CxClientPool**
   - **CxPool**
   - **CxPoolRestAPI**
   - **CxAccessControl**
2. If any of the Cx Application Pools are anything other than the default **Network Service**, make sure that you know the user account's full credentials.
3. Open IIS Manager Console:

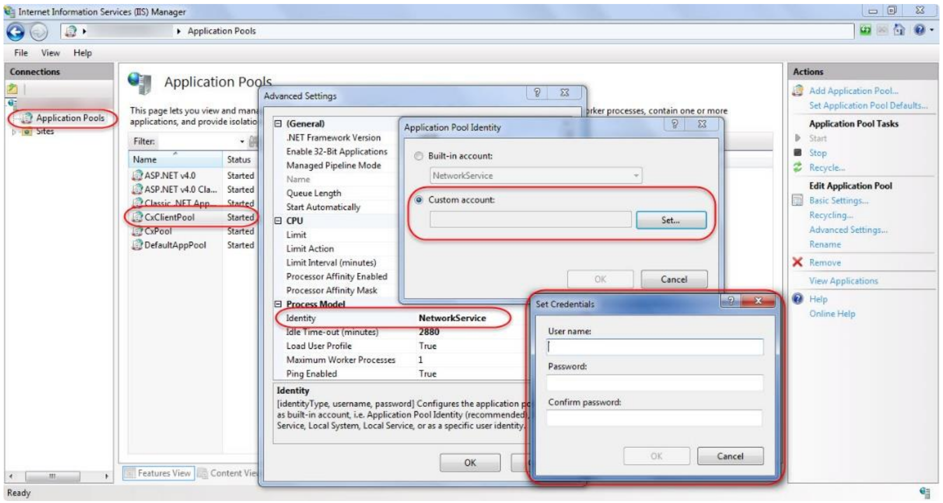


Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different applications.

Name	Status	.NET Fr...	Managed ...	Identity	Appl...
ASP.NET v4.0	Started	v4.0	Integrated	ApplicationPoolIdenti...	0
ASP.NET v4.0 Cla...	Started	v4.0	Classic	ApplicationPoolIdenti...	0
Classic .NET App...	Started	v4.0	Classic	ApplicationPoolIdenti...	0
CxClientPool	Started	v4.0	Classic	NetworkService	1
CxPool	Started	v4.0	Classic	NetworkService	1
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolIdenti...	1

4. Click **Application Pools** and then select any of the **Cx Application Pools**.
5. Click **Advanced Settings** on the Action menu.



This page lets you view and mana... applications, and provide isolatio... ...rker processes, contain one or more

⊟ (General)
.NET Framework Version
Enable 32-Bit Applications
Managed Pipeline Mode
Name
Queue Length
Start Automatically
⊟ CPU
Limit
Limit Action
Limit Interval (minutes)
Processor Affinity Enabled
Processor Affinity Mask
⊟ Process Model
Identity NetworkService
Idle Time-out (minutes) 2880
Load User Profile True
Maximum Worker Processes 1
Ping Enabled True

Identity
[identityType, username, password] Configures the application p... as built-in account, i.e. Application Pool Identity (recommended)... Service, Local System, Local Service, or as a specific user identity.

6. In Advanced Settings window, scroll to **Identity** (under **Process Model**) and double click the user that is defined.
7. In the Application Pool Identity window, select the **Custom Account** radio button and click **Set.**
8. Enter the appropriate user credentials and click **OK**.

Cx Storage Folders

1. Ensure that the user accessing the Cx storage folders (CxSrc, CxReports, ExtSrc) has the appropriate read/write permissions.
2. To modify the read/write permissions for Cx storage folders:
3. Navigate to the Cx storage folder (C:\CxSrc, C:\CxReports, C:\ExtSrc)
4. Right-click on the folder, click **Properties**, and then click the **Security** tab.

5. Click **Edit** and select the user or group that you want to change the permissions for.
6. Check the boxes for the permissions you want to add for that user or group.
7. For single manager with local folders define read/write permissions.
8. Click **Apply** to save the changes.
9. Repeat the same procedures for the remaining Cx storage folders.

Configuring Single Sign-On (SSO)

You can configure CxSAST to automatically use the Windows credentials of the user that is logged on to Windows, so that registered domain users do not need to independently log into CxSAST.

- Single sign-on authentication is available only to Active Directory users.

➢ **To configure single sign-on (SSO):**

- These instructions apply to IIS 8, 8.5 and 10

1. Make sure that the CxSAST server is in the organizational domain.
2. On the CxSAST server, activate IIS Windows Authentication. In a distributed deployment, you have to activate IIS Windows Authentication on the CxManager. To activate IIS Windows Authentication, do the following:

    a) Open **Turn Windows features on or off** (you can find it from the Windows search bar) or in **Windows Server Manager > Manage > Add Roles and Features**.

    b) Under **Internet Information Services** > **World Wide Web Services** or **Web Server (IIS) > Web Server**
    select **Security > Windows Authentication** and install.

3. Open the IIS Manager, and apply Windows Authentication to the CxSAST web services. Do one of the following for **CxWebClient**, **CxWebInterface** and **CxRestAPI**:

    a) In the left-hand **Connections** pane, navigate to and select the **'Default Web Site'** web service and in the IIS section, double-click **Authentication**.

    - By default the following web applications are installed under **'Default Web Site'** :
      CxRestAPI,
      CxWebClient,
      CxWebInterface

These applications inherit the changes outlined below.
If the web applications are on a custom IIS Site, make the change on that site.

b) Right-click **Windows Authentication** and select **Enable**.

- If the Windows authentication is Kerberos:

i. Right-click **Windows Authentication** and select **Providers**.
ii. Under **Available Providers**, add **Negotiate**, if not already listed.
iii. Move **Negotiate** above **NTLM**.
iv. Click **OK**.

- In the IIS, set **useAppPoolCredentials** to **True**:

v. Select the '**Default Web Site**"
vi. Under **Management**, double-click **[Configuration Editor]** and set **'useAppPoolCredentials'** to **True** under this section: **system.webServer/security/authentication/windowsAuthentication**

c) If your Cx Application Pools (**CxAccessControl**, **CxClientPool**, **CxPool**, and **CxPoolRestAPI**) Login Identity is configured for a ***'Custom Domain Service Account'*** (login service account), modify as follows:

ii. Select the '**Default Web Site\CxRestAPI\auth**' application
iii. Under **Management**, double-click **[Configuration Editor]** and set **'useAppPoolCredentials'** to **False** under this section: **system.webServer/security/authentication/windowsAuthentication**

4. On the CxSAST server, open the following file for editing:
<Installation path>\Checkmarx\CheckmarxWebPortal\Web\web.config, for example **C:\Program Files\Checkmarx\CheckmarxWebPortal\Web\web.config**
Under **<appSettings>**, navigate to the **UseSSOLogin** key, and change its value to **true** as noted below:
<add key="UseSSOLogin" value="true"/>

CxSAST is now authorized CxSAST Active Directory users by their being logged into Windows.

Configuring the Checkmarx Web Portal on a Dedicated Host

The Checkmarx Software Exposure Platform supports Distributed Architecture, where some or all the Checkmarx Software Exposure Platform components can be installed on a dedicated station (host). This instruction defines the procedure for configuring the Cx Web Portal on a dedicated host. For further information and instructions on configuring the Web Portal on a dedicated host, refer to this page.

Configuring the Proxy from a Checkmarx Server

- **To configure the proxy from the Checkmarx Server:**

In the web configuration file that is located in C:\Program Files\Checkmarx\Checkmarx Web Services\CxWebInterface\web.config, add following syntax to the <system.webserver> section.

```
<system.net>
 <defaultProxy enabled=""true"" useDefaultCredentials=""true"">
 <module type=""Checkmarx.CxInfraSructures.WebProxy.CxWebProxy,
CxInfraSructures""/>
 <proxy bypassonlocal=""True"" proxyaddress=""http://10.31.1.111:8085"" />
<bypasslist>
 <clear/>
 <add address=""10.31.8.*""/>
 <add address=""10.31.98.*""/>
 </bypasslist>
 </defaultProxy>
</system.net>
```

- In the XML code, the """" are optional. These are addresses that CxServer can connect to bypass the proxy server.

Proxy Server Settings

In the Proxy Server Settings window, fill in the following properties:

- **Name** - Name of proxy server
- **Protocol** - Select HTTP Proxy
- **Client Port** - Select the client port to access
- **Use HTTP Authentication** - Select Use HTTP Authentication option
- **Authentication Challenges** - Select Basic
- **Read Timeout** - Configure to 30 seconds (default)
- **Connect Timeout** - Configure to 30 seconds (default)
- **Report all Connects** - Select the check-box
- **Log File** - Location of the log file

Proxy Service Console

In the Proxy Status window, fill in the following properties and then click **OK**:

- **Current Status** - Service/ console: Running/Non-Running
- **Service mode** - Start/Stop/Restart
- **Console mode** - Start/Stop

Configuring CxOSA with a Proxy Server

In order to run Open Source Analysis (CxOSA) and generate and download analysis result reports (PDF) when using a proxy server, perform the following configuration steps:

➢ **To run CxOSA:**

1. Navigate to C:\Program Files\Checkmarx\Checkmarx Scans Manager\bin\CxScansManagerWinService.exe.config.
2. Insert the <system.net> tag just before the closing </configuration> tag and add the <defaultProxy> tag and contents as follows:

```
<system.net>
<defaultProxy useDefaultCredentials="true">
```

```
<proxy usesystemdefault="False" proxyaddress="http://<proxy ip
address>:<proxy port>" bypassonlocal="False" />
<bypasslist>
<add address="<ip address range>" />
</bypasslist>
</defaultProxy>
</system.net>
```

- The <bypasslist> parameter is required in situations where the traffic for remote CxEngines should not go through the defined Proxy. Symptoms are that after adding this proxy configuration, the CxOSA scans works however the CxSAST scans do not. The add address value in this example is a Regex to capture a range of IP Addresses starting 10.65 . Further information on this parameter may be found here.

Example:

```
<proxy usesystemdefault="False" proxyaddress="http://127.0.0.1:8888"
bypassonlocal="True" /> <bypasslist> <add address="10\.65\.\d{1,3}\.\d{1,3}"
/>
```

➢ **To generate and download analysis result reports (PDF):**

1. Navigate to C:\Program Files\Checkmarx\Checkmarx Web RestAPI\CxRestAPI\Web.config,
2. Under the <system.net> tag, add the same <defaultProxy> tag and its contents as in running CxOSA, above.
3. Restart all Cx services (CxJobsManager, CxScansManager, CxSystemManager and CxScanEngine services.
4. Restart the IIS web server.

Configuring Management & Orchestration SQL Server for Dynamic and Static Port Connectivity

These instructions define the procedure for configuring the Management & Orchestration SQL server for both static and dynamic port connectivity for CxSAST V8.9 RC6 and above.

Static Port Configuration

Prerequisites

The following perquisites apply:

- For Windows authentication, the machine must be in a domain
- SQL Server service is up and running
- TCP/IP is enabled

Static Port Connection

The Management & Orchestration SQL server installation defines the port connection according to one of the following port connection arrangements:

- Instanced SQL: <host>\<instance>,<port>

  Examples:
  localhost\sqlexpress,1435
  10.0.0.10\sqlexpress,1435
  .\sqlexpress,1435
  10.0.0.10,1435

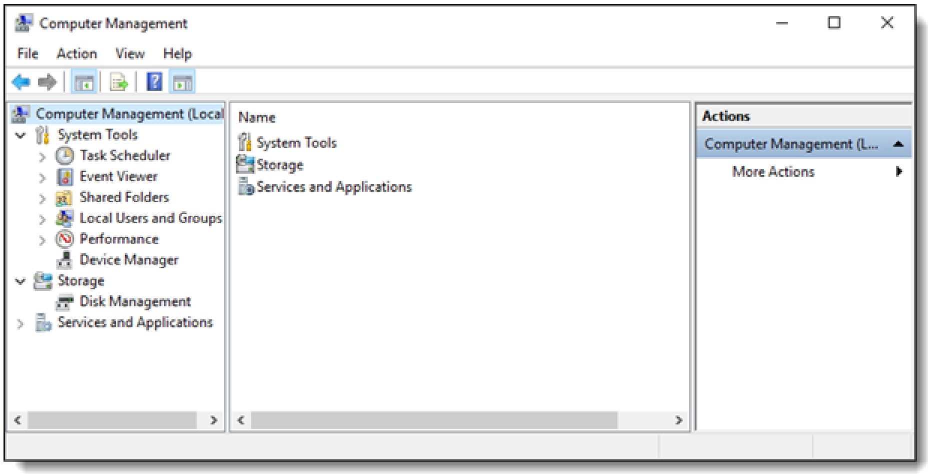
- Non Instanced SQL: <host>,<port>

  Examples:
  localhost,1435
  10.0.0.10,1435
  .,1435
  10.0.0.10,1435

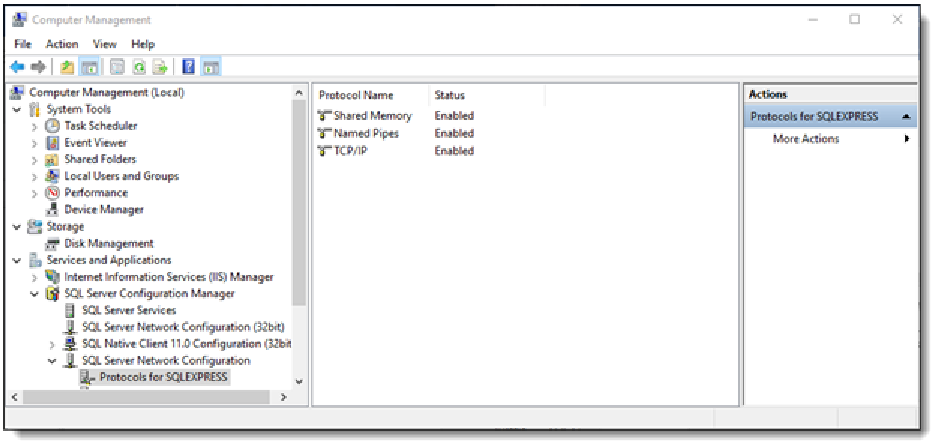
Configure Static Port

➢ **To configure static port connectivity for the Management & Orchestration SQL Server:**

1. Go to the Computer Management (Start > Control Panel > System and Security > Administrative Tools > Computer Management).

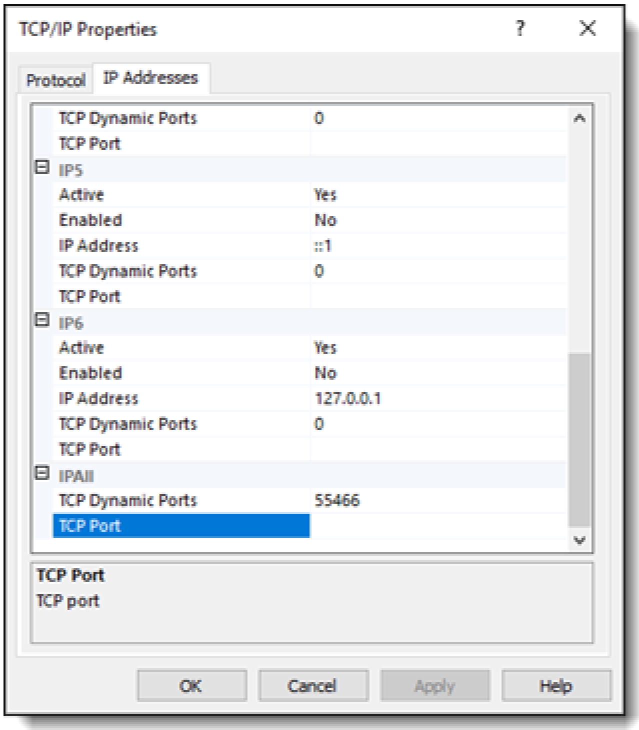


2. Select Services and Applications > SQL Server Configuration Manager > SQL Server Network Configuration > Protocols for <SQLSERVER>.



Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled

3. Right click on TCP/IP, select Properties and click the IP Address tab.



TCP Dynamic Ports	0
TCP Port	
⊟ IP5	
Active	Yes
Enabled	No
IP Address	::1
TCP Dynamic Ports	0
TCP Port	
⊟ IP6	
Active	Yes
Enabled	No
IP Address	127.0.0.1
TCP Dynamic Ports	0
TCP Port	
⊟ IPAll	
TCP Dynamic Ports	55466
TCP Port	

TCP Port
TCP port

4. Select TCP Port under IPALL and define the port as defined in the Management & Orchestration SQL server installation (see Port Connection Arrangement).
5. Click Apply and then OK to complete.
6. Restart the SQL Browser service and then the SQL Server service.
7. If there are two instances of the SQL Server service, restart the corresponding instance of the SQL Browser service.

Dynamic Port Configuration

Prerequisites

The following perquisites must be in place:

- For Windows authentication the machine must be in a domain
- SQL Server and SQL Browser services are up and running.

    - If the SQL Brower service is already up, restart the SQL Server service again. If there are two instances of the SQL Server service, verify that the corresponding instance of SQL Browser service is up and running

- TCP/IP is enabled

Dynamic Port Connection

The Management & Orchestration SQL server installation defines the port connection according to one of the following port connection arrangements:

- Instanced SQL: <host>\<instance>

  Examples:
  localhost\sqlexpress
  10.0.0.10\sqlexpress
  .\sqlexpress
  hostname\ sqlexpress

- Non-instanced SQL: <host>,<port>

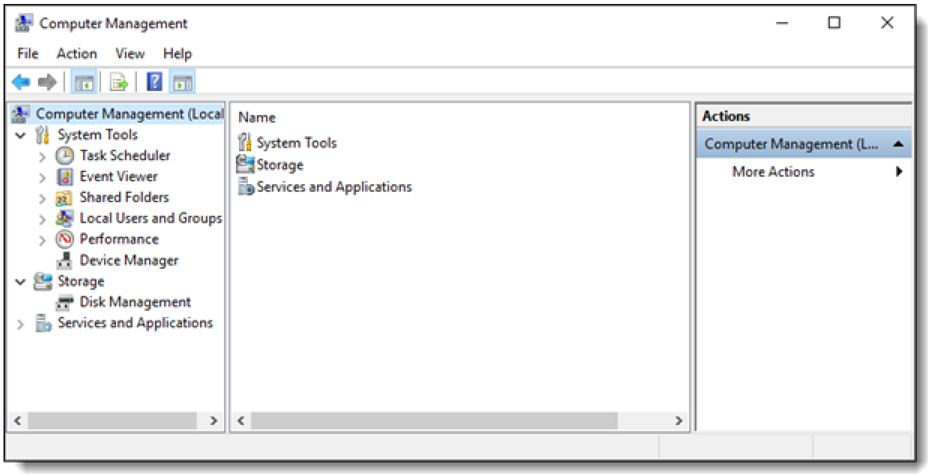
  Examples:
  localhost
  10.0.0.10
  .
  hostname

- For connection arrangements with unnamed instance and without port, <.>, <ip> or <host_name>, the default static port must be set (see Static Port Connection).

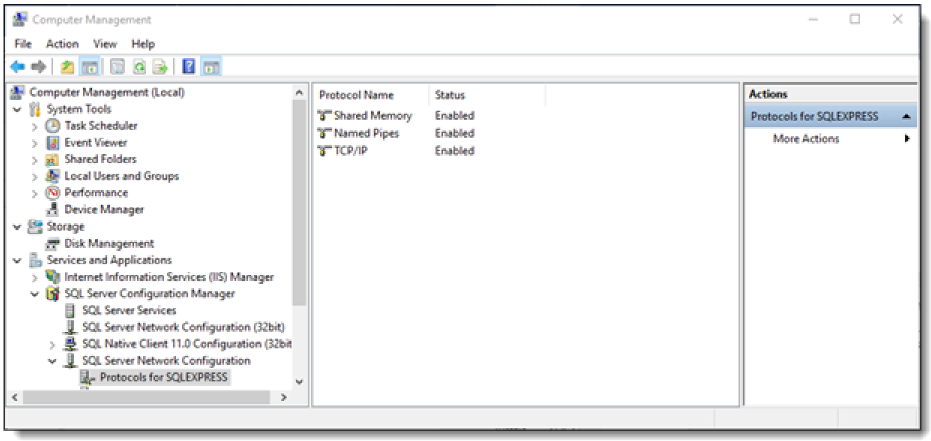
Configure Dynamic Port

To configure dynamic port connectivity for the Management & Orchestration SQL Server:

1. Go to the Computer Management (Start > Control Panel > System and Security > Administrative Tools > Computer Management).

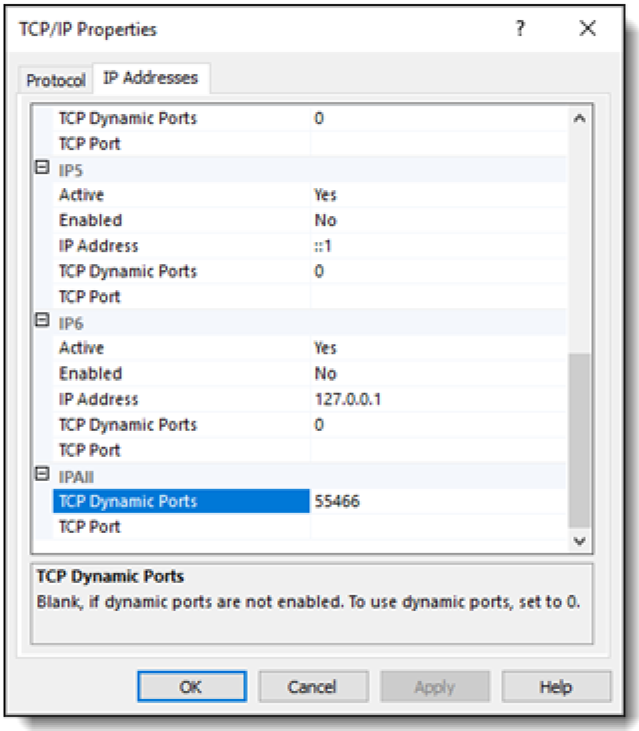


2. Select Services and Applications > SQL Server Configuration Manager > SQL Server Network Configuration > Protocols for <SQLSERVER>.



Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled

3. Right click on TCP/IP, select Properties and click the IP Address tab.



TCP Dynamic Ports	0
TCP Port	
⊟ IP5	
Active	Yes
Enabled	No
IP Address	::1
TCP Dynamic Ports	0
TCP Port	
⊟ IP6	
Active	Yes
Enabled	No
IP Address	127.0.0.1
TCP Dynamic Ports	0
TCP Port	
⊟ IPAll	
TCP Dynamic Ports	55466
TCP Port	

TCP Dynamic Ports

Blank, if dynamic ports are not enabled. To use dynamic ports, set to 0.

4. Select TCP Dynamic Ports under IPALL and define the port as defined in the Management & Orchestration SQL server installation (see Port Connection Arrangement).
5. Click Apply and then OK to complete.
6. Restart the SQL Browser service and then the SQL Server service.

- If there are two instances of the SQL Server service, restart that the corresponding instance of SQL Browser service

Port Connection Test

The following command can be run to test the port connection:

```
jre\bin\java -cp "etl.jar;Libraries\mssql-jdbc-6.4.0.jre8.jar" -
Djava.library.path="Libraries" com.checkmarx.sync.utl.mainflow.TestConnection
10.31.2.36 null null CxARM SQLEXPRESS
```

- The command should be run from the folder where the etl.jar is located
- You can provide “user” and “password” instead of “null” “null”.
- 10.31.2.36 is the IP of the machine where the database is installed
- SQLEXPRESS is the instance name (“null” can be supplied in unnamed instance).

M&O Connectivity Mitigation

- **Use Case**: Machine not in domain + SQL windows authentication
  Mitigation: Use user + password authentication
- **Use Case**: Dynamic port + SQL Browser service is turned off + SQL Server with named instance
  Mitigation: Configure static port, or turn SQL Browser on
- **Use Case**: TCP/IP disabled
  Limitation: No solution, TCP/IP is a must

Configuring CxManager with Windows Authentication to the DB with a separate Client Portal

These instructions define the procedure for configuring CxManager with Windows authentication to the database with a separate client portal (for (v8.0.0 and up)

Prerequisites

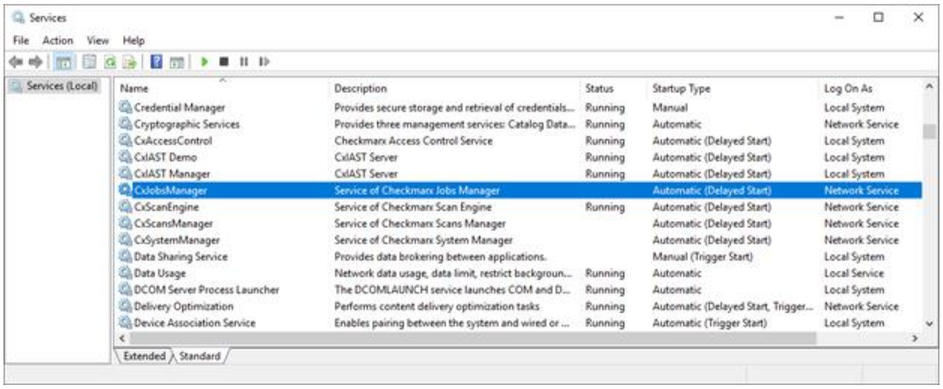
The following prerequisites are required:

- Distributed environment with each component on a separate machine (4 machines in total - Client Portal, Manager, Database and Engine).
- All machines installed inside a domain with administrator access available

- CxManager machine installed with Windows authentication with its own domain user
- Remaining machines can be configured for regular users

Manager Configuration

1. Once the Manager machine is installed, go to the Services (Start > Control Panel > System and Security > Administrative Tools > Services).

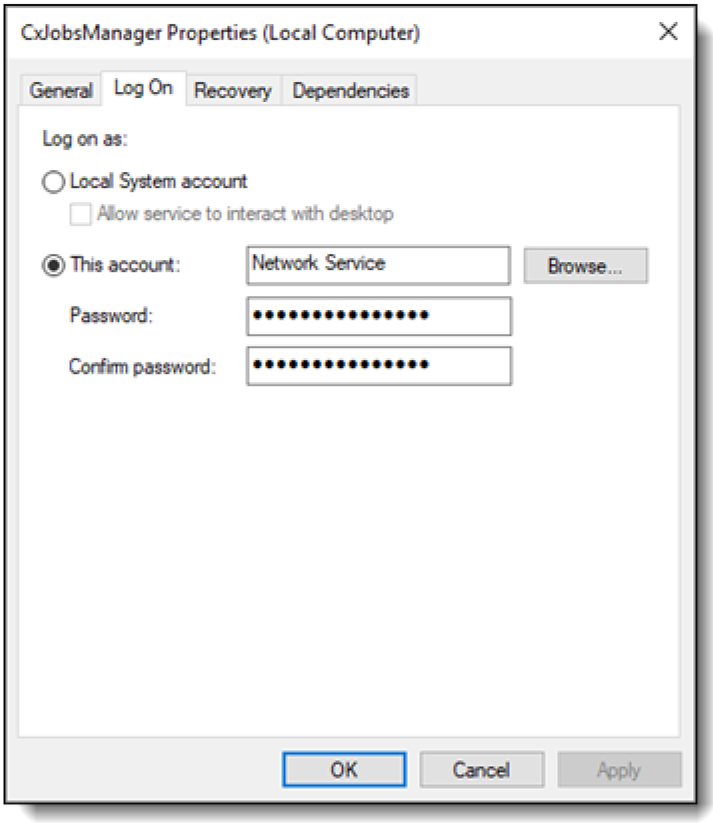


Name	Description	Status	Startup Type	Log On As
Credential Manager	Provides secure storage and retrieval of credentials...	Running	Manual	Local System
Cryptographic Services	Provides three management services: Catalog Data...	Running	Automatic	Network Service
CxAccessControl	Checkmarx Access Control Service	Running	Automatic (Delayed Start)	Local System
CxIAST Demo	CxIAST Server	Running	Automatic (Delayed Start)	Local System
CxIAST Manager	CxIAST Server	Running	Automatic (Delayed Start)	Local System
CxJobsManager	Service of Checkmarx Jobs Manager		Automatic (Delayed Start)	Network Service
CxScanEngine	Service of Checkmarx Scan Engine	Running	Automatic (Delayed Start)	Network Service
CxScansManager	Service of Checkmarx Scans Manager		Automatic (Delayed Start)	Network Service
CxSystemManager	Service of Checkmarx System Manager		Automatic (Delayed Start)	Network Service
Data Sharing Service	Provides data brokering between applications.		Manual (Trigger Start)	Local System
Data Usage	Network data usage, data limit, restrict backgroun...	Running	Automatic	Local Service
DCOM Server Process Launcher	The DCOMLAUNCH service launches COM and D...	Running	Automatic	Local System
Delivery Optimization	Performs content delivery optimization tasks	Running	Automatic (Delayed Start, Trigger...	Network Service
Device Association Service	Enables pairing between the system and wired or ...	Running	Automatic (Trigger Start)	Local System

Depending on the CxSAST version installed, the following Cx services are available:

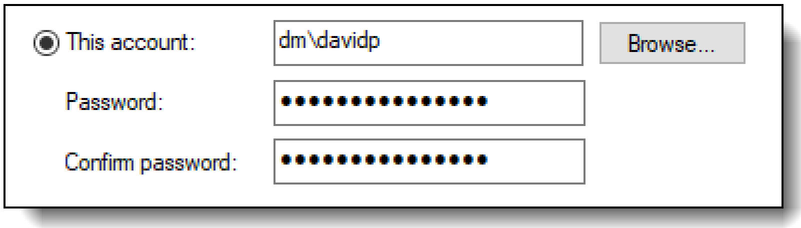
- CxJobsManager (v8.x)
- CxScansManager (v8.x)
- CxSystemManager (v8.x)
- CxSastResults (v9.0 and up)
- CxScanEngine (v8.x)
- CxARM (v8.8 and up)
- CxARMETL (v8.8 and up)
- CxRemediationIntelligence (v9.0 and up)
- CxAccessControl (v9.0 and up)

2. Right click on the first Cx Service, select Properties and open the Log On tab.



This account: Network Service
Password: •••••••••••••••
Confirm password: •••••••••••••••

3. Enter the defined domain user (dm\<username>) into the 'This account' field.

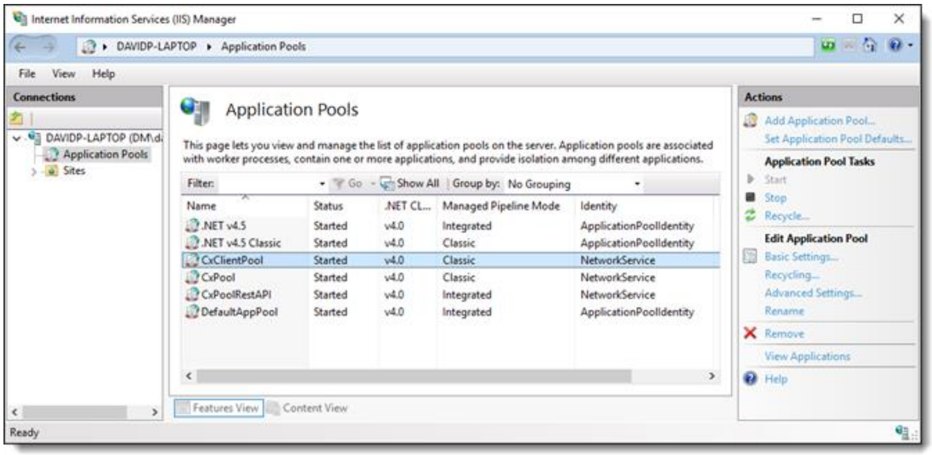


● This account: dm\davidp
Password: •••••••••••••••
Confirm password: •••••••••••••••

4. Enter the domain user password into the 'Password' and 'Confirm Password' fields.
5. Click Apply to save the changes.
6. Perform the same procedure for each of the available Cx Services.

- The available Cx Services will depend on the CxSAST version installed (see Cx Service/CxSAST Version information above).

7. Once complete, manually restart all Cx Services.
8. Once the Cx Services have been started, go to the IIS Manager (Start > Control Panel > System and Security > Administrative Tools > Internet Information Services (IIS) Manager) and enter to the Application Pools.



Application Pools

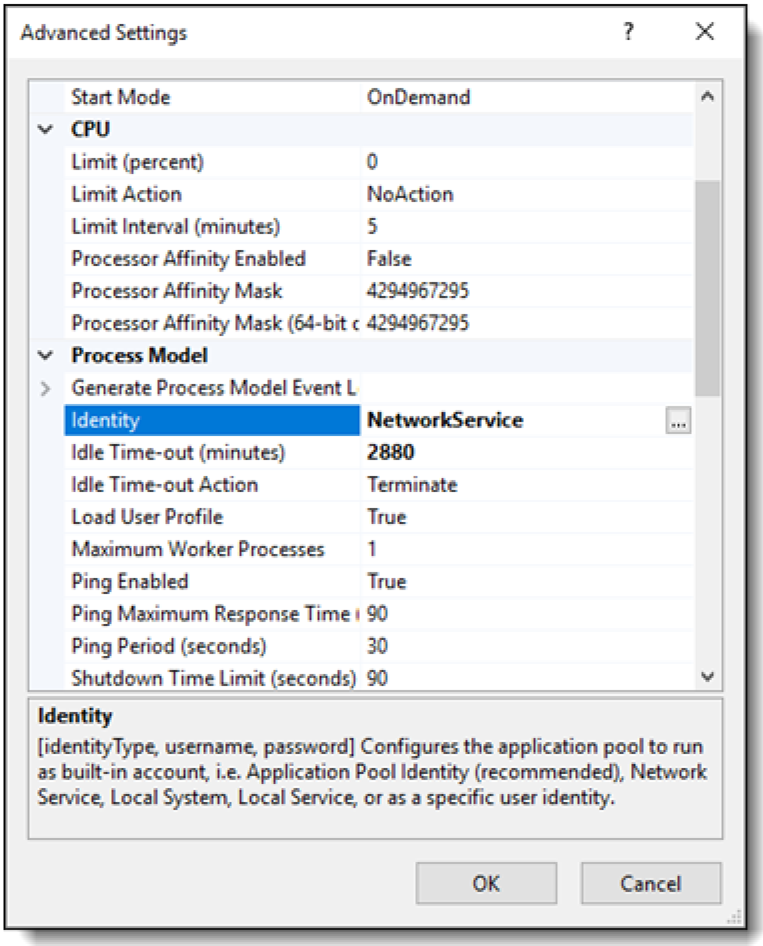
This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different applications.

Name	Status	.NET CL...	Managed Pipeline Mode	Identity
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolIdentity
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolIdentity
CxClientPool	Started	v4.0	Classic	NetworkService
CxPool	Started	v4.0	Classic	NetworkService
CxPoolRestAPI	Started	v4.0	Integrated	NetworkService
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolIdentity

The following Cx Application Pools are available:

- CxClientPool
- CxPool
- CxRestPool

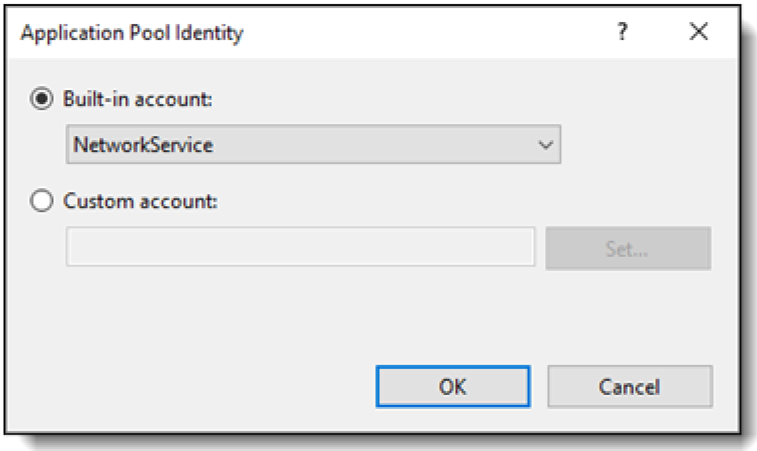
9. Right click on the first Cx Application Pool and select Advanced Settings.



Start Mode	OnDemand
CPU	
Limit (percent)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit c	4294967295
Process Model	
Generate Process Model Event L	
Identity	**NetworkService**
Idle Time-out (minutes)	**2880**
Idle Time-out Action	Terminate
Load User Profile	True
Maximum Worker Processes	1
Ping Enabled	True
Ping Maximum Response Time (	90
Ping Period (seconds)	30
Shutdown Time Limit (seconds)	90

Identity
[identityType, username, password] Configures the application pool to run as built-in account, i.e. Application Pool Identity (recommended), Network Service, Local System, Local Service, or as a specific user identity.

10. Open the Application Pool Identity account window.

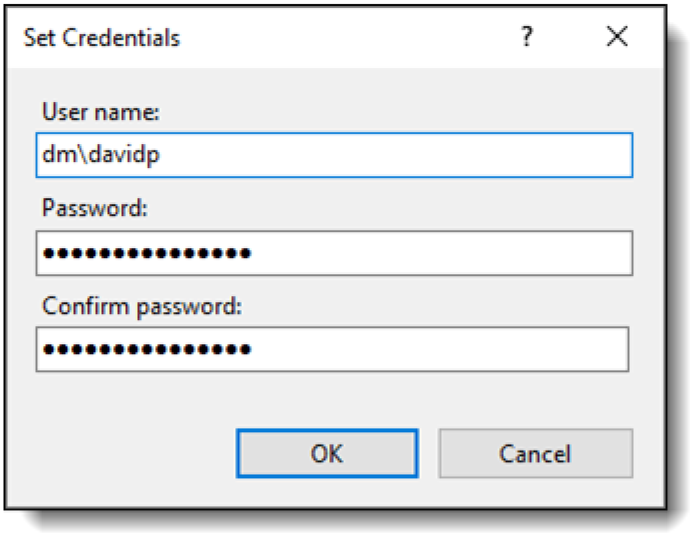


● Built-in account:

NetworkService

○ Custom account:

11. Select Custom account, click the Set button and enter the defined domain user (dm\<username>) into the 'User name' field.



Set Credentials

User name: dm\davidp

Password: ●●●●●●●●●●●●●●●

Confirm password: ●●●●●●●●●●●●●●●

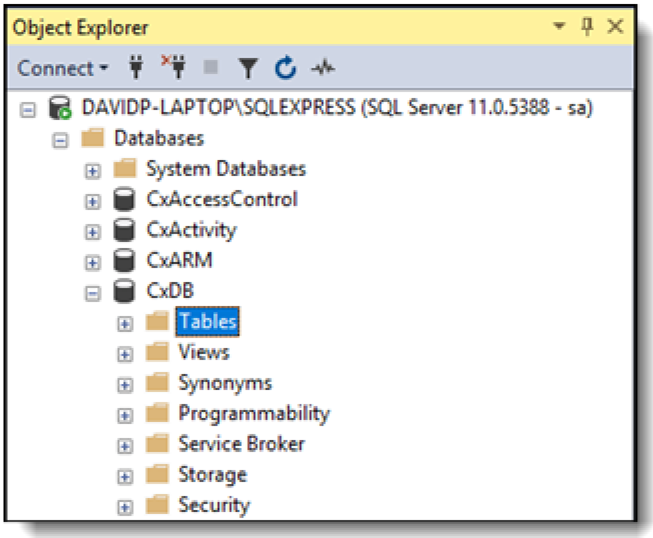
12. Enter the domain user password into the ‘Password’ and ‘Confirm Password’ fields.
13. Click OK to save the changes.
14. Using the CMD, restart the Internet Information Services (IIS) Manager.

Once you have completed the Manager configuration, you can now configure the database machine.

Database Configuration

Once you have completed the Manager configuration, you can now configure the DB machine.

1. With the DB machine installed, open MS SQL Server Management Studio.
2. Connect to the SQL server.
3. Enter to Databases > CxDB > Tables.

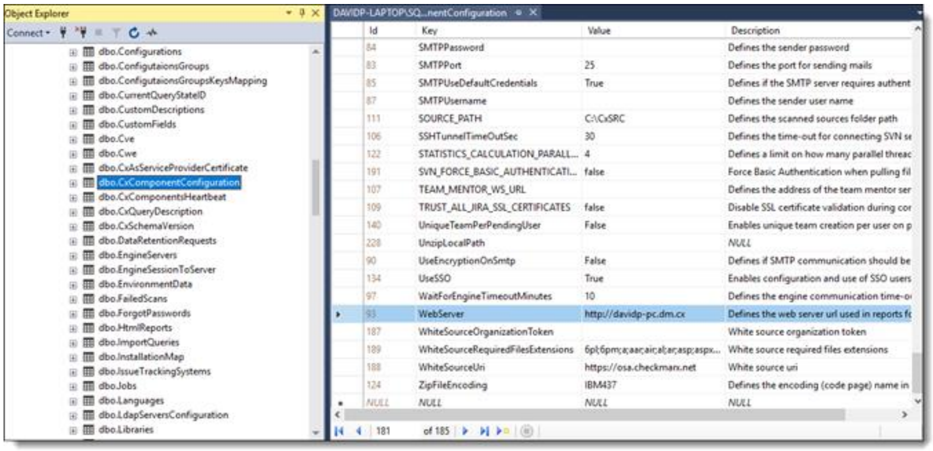


Object Explorer

Connect

DAVIDP-LAPTOP\SQLEXPRESS (SQL Server 11.0.5388 - sa)
- Databases
  - System Databases
  - CxAccessControl
  - CxActivity
  - CxARM
  - CxDB
    - Tables
    - Views
    - Synonyms
    - Programmability
    - Service Broker
    - Storage
    - Security

4. Right-click on the CxComponentConfiguration table and select Edit Rows.



Object Explorer

Connect

- dbo.Configurations
- dbo.ConfigutaionsGroups
- dbo.ConfigutaionsGroupsKeysMapping
- dbo.CurrentQueryStateID
- dbo.CustomDescriptions
- dbo.CustomFields
- dbo.Cve
- dbo.Cwe
- dbo.CxAsServiceProviderCertificate
- dbo.CxComponentConfiguration
- dbo.CxComponentsHeartbeat
- dbo.CxQueryDescription
- dbo.CxSchemaVersion
- dbo.DataRetentionRequests
- dbo.EngineServers
- dbo.EngineSessionToServer
- dbo.EnvironmentData
- dbo.FailedScans
- dbo.ForgotPasswords
- dbo.HtmlReports
- dbo.ImportQueries
- dbo.InstallationMap
- dbo.IssueTrackingSystems
- dbo.Jobs
- dbo.Languages
- dbo.LdapServersConfiguration
- dbo.Libraries

DAVIDP-LAPTOP\SQ...nentConfiguration

Id	Key	Value	Description
84	SMTPPassword		Defines the sender password
83	SMTPPort	25	Defines the port for sending mails
85	SMTPUseDefaultCredentials	True	Defines if the SMTP server requires authent
87	SMTPUsername		Defines the sender user name
111	SOURCE_PATH	C:\CxSRC	Defines the scanned sources folder path
106	SSHTunnelTimeOutSec	30	Defines the time-out for connecting SVN se
122	STATISTICS_CALCULATION_PARALL...	4	Defines a limit on how many parallel threac
191	SVN_FORCE_BASIC_AUTHENTICATI...	false	Force Basic Authentication when pulling fil
107	TEAM_MENTOR_WS_URL		Defines the address of the team mentor ser
109	TRUST_ALL_JIRA_SSL_CERTIFICATES	false	Disable SSL certificate validation during cor
140	UniqueTeamPerPendingUser	False	Enables unique team creation per user on p
228	UnzipLocalPath		NULL
90	UseEncryptionOnSmtp	False	Defines if SMTP communication should be
134	UseSSO	True	Enables configuration and use of SSO users
97	WaitForEngineTimeoutMinutes	10	Defines the engine communication time-o
93	WebServer	http://davidp-pc.dm.cx	Defines the web server url used in reports fc
187	WhiteSourceOrganizationToken		White source organization token
189	WhiteSourceRequiredFilesExtensions	6pl;6pm;a;aar;air;al;ar;asp;aspx...	White source required files extensions
188	WhiteSourceUri	https://osa.checkmarx.net	White source uri
124	ZipFileEncoding	IBM437	Defines the encoding (code page) name in
NULL	NULL	NULL	NULL

181 of 185

5. In the WebServer key field, enter the full domain name of the client machine as follows:

```
http://{full_client_portal_manchine_name}.dm.cx
```

6. Save the DB changes.

Configuring a Non-default Location for Management and Orchestration and CxSAST Component Data

The purpose of these instructions is to configure a non-default path location for Management and Orchestration and CxSAST component data (i.e. kahaDB). This can be performed in order to withstand hotfixes and upgrades and can be achieved by changing the component data file path in the associated component '.xml' file with a new data file path.

Changing the Management and Orchestration and CxSAST Component Data File Location

1. Locate the 'activemq.xml' file under *C:\Program Files\Checkmarx\Checkmarx ActiveMQ\conf)*. For this example, the kahaDB 'activeMQ.xml' file is used.
2. Open the 'activeMQ.xml' file and define the <kahaDB> key with the desired path.

- In case of ActiveMQ cluster environment, this new path should be defined for all ActiveMQ instances.

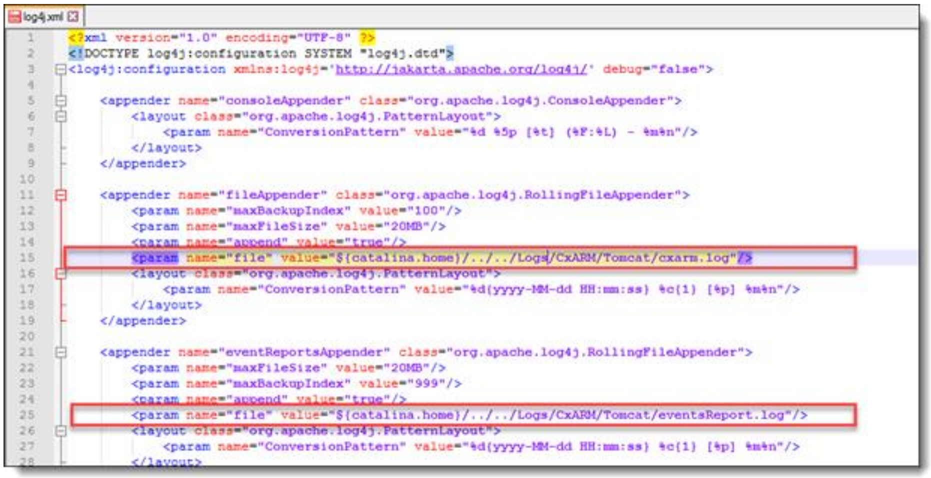
3. Restart the ActiveMQ.

Configuring a Non-default Location for Management and Orchestration Component Logs

The purpose of these instructions is to configure a non-default location for Management and Orchestration component logs (i.e. Policy Manager). This can be performed in order to withstand hotfixes and upgrades and can be achieved by changing the component log file path in the associated component '.xml' file with a new log file path.

Changing the Management and Orchestration Component Log File Location

1. Locate the 'log4j.xml' file under *C:\Program Files\Checkmarx\Checkmarx Risk Management\Tomcat\webapps\cxarm#policymanager\WEB-INF\classes*. For this example, the Policy Manager 'log4j.xml' file is used.
2. Open the 'log4j.xml' file and define the <file> keys with the desired path.



```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/' debug="false">

    <appender name="consoleAppender" class="org.apache.log4j.ConsoleAppender">
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d %5p [%t] (%F:%L) - %m%n"/>
        </layout>
    </appender>

    <appender name="fileAppender" class="org.apache.log4j.RollingFileAppender">
        <param name="maxBackupIndex" value="100"/>
        <param name="maxFileSize" value="20MB"/>
        <param name="append" value="true"/>
        <param name="file" value="${catalina.home}/../../Logs/CxARM/Tomcat/cxarm.log"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %c{1} [%p] %m%n"/>
        </layout>
    </appender>

    <appender name="eventReportsAppender" class="org.apache.log4j.RollingFileAppender">
        <param name="maxFileSize" value="20MB"/>
        <param name="maxBackupIndex" value="999"/>
        <param name="append" value="true"/>
        <param name="file" value="${catalina.home}/../../Logs/CxARM/Tomcat/eventsReport.log"/>
        <layout class="org.apache.log4j.PatternLayout">
            <param name="ConversionPattern" value="%d{yyyy-MM-dd HH:mm:ss} %c{1} [%p] %m%n"/>
        </layout>
```

3. Perform the same procedure for all the Management and Orchestration components:

    - **Policy Manager**: *C:\Program Files\Checkmarx\Checkmarx Risk Management\Tomcat\webapps\cxarm#policymanager\WEB-INF\classes*
    - **Dashboard**: *C:\Program Files\Checkmarx Risk Management\Tomcat\webapps\cxarm#dashboardapi\WEB-INF\classes*
    - **Analytics**: *C:\Program Files\Checkmarx Risk Management\Tomcat\webapps\cxarm#cxanalytics\WEB-INF\classes*
    - **ETL**: *C:\Program Files\Checkmarx\Checkmarx Risk Management\ETL*
    - ***Dashboard swagger***: *C:\Program Files\Checkmarx\Checkmarx Risk Management\Tomcat\webapps\cxarm#dashboardapi#swagger\WEB-INF\classes*
    - **Policy Manager swagger**: *C:\Program Files\Checkmarx Risk Management\Tomcat\webapps\cxarm#policymanager#swagger\WEB-INF\classes*

4. Go to the Services (*Start > Control Panel > System and Security > Administrative Tools > Services*) and restart CxARM service.

Configuring User Credentials for CxDB Connectivity

The purpose of these instructions is to configure the user credentials used for CxDB connectivity.

➢ **To change the user credentials used for CxDB connectivity:**

1. Locate the properties configuration file, found in the Config folder in the Checkmarx Risk Management folder (for example: *C:\Program Files\Checkmarx\Checkmarx Risk Management\Config*).
2. Locate the 2 keys relevant to the CxDB connection: DB_USER and DB_PASSWORD.

3. Enter both new user name and new password, and then save the file.
4. Restart the CxARM service, which will automatically update the values and encrypt the password data in the CxARM DB.

- After updating the user name and password, these values will be deleted from the db.properties file.

Performing Protocol, Machine Name and Port Changes for Cx Components

Background

A number of additional components have been introduced (Access Control, CxSAST, Management & Orchestration, and ActiveMQ) to the latest versions of CxSAST. The endpoints for these components are saved in the database (CxDB) as {Protocol}://{FQDN}:{Port}. Fully Qualified Domain Name (FQDN) is the complete domain name for the machine and consists of the hostname and domain name (e.g. http://mqserver.company.com:5555).

- For clean installations (v9.0.0 and up), component endpoints are, by default, saved as HTTP. Upgrades (to v9.0.0) always keep previous component endpoint values.

Use Cases

This instruction defines the procedure for changing component endpoint configurations in the database, in cases where the current configurations need to be changed, The following use cases will determine if and how these component endpoints should to be changed.

- **Use-Case 1:** If the machine is only reachable by the IP and not by the FQDN, for instance, if a DNS Server is not used, you will need to change the table key value definitions in the database.
- **Use-Case 2:** If you change the CxSAST, Management & Orchestration or ActiveMQ ports after installation, you will need to change the table key value definitions in the database.

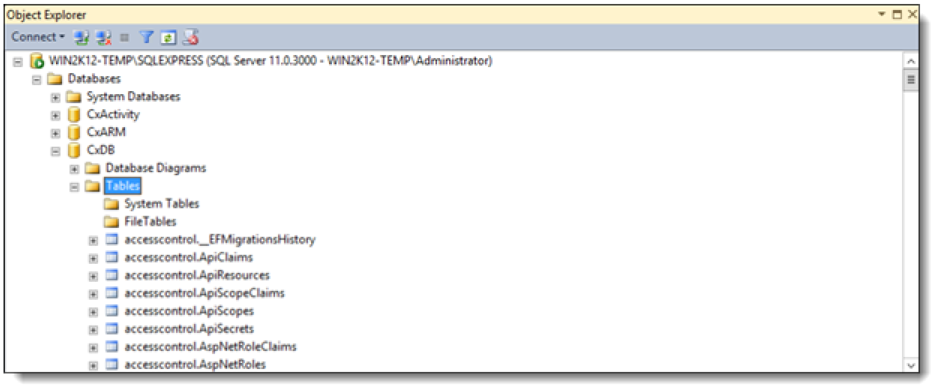
- **Use-Case 3:** If you manually configured the environment as Secure Sockets Layer (SSL), you may need to change the table key value definitions in the database.
- **Use-Case 4:** If you rename the machine, you will need to change the table key value definitions in the database.
- **Use-Case 5:** If you configured the system to connect users via a corporate proxy server, you will need to change the table key value definitions in the database.
- **Use-Case 6:** If you add a load balancer, for instance, in HA deployments where the load balancer endpoint is used instead of the machine name, you will need to change the table key value definitions in the database.

For further instructions and examples, refer to Accessing the Database Table and Changing Table Key Value Definitions.

Accessing the Database Table

➢ **To access the database:**

1. Once the CxSAST (v9.0.0 and up) environment is installed and fully configured, do the following:
2. Open **MS SQL Server Management Studio.**
3. Connect to the **SQL server**.
4. Go to: **Databases > CxDB > Tables.**


Object Explorer
Connect ▾
WIN2K12-TEMP\SQLEXPRESS (SQL Server 11.0.3000 - WIN2K12-TEMP\Administrator)
Databases
System Databases
CxActivity
CxARM
CxDB
Database Diagrams
Tables
System Tables
FileTables
accesscontrol.__EFMigrationsHistory
accesscontrol.ApiClaims
accesscontrol.ApiResources
accesscontrol.ApiScopeClaims
accesscontrol.ApiScopes
accesscontrol.ApiSecrets
accesscontrol.AspNetRoleClaims
accesscontrol.AspNetRoles

Changing Table Key Value Definitions

1. In the **Tables** folder (**Databases > CxDB > Tables**), right-click on the required **Table**, according to the table/key value definitions table below, and select **Edit Top 200 Rows.**
2. For each **Key**, change the **Value** field according to the relevant use case (refer to Use Cases).

Table	Key	Value
dbo.CxComponentConfiguration	**IdentityAuthority** (i.e. Access Control URL)	{Protocol}://{Machine}:{Port}/CxRestAPI/auth Default HTTP port = 80 When upgrading to v9.0.0, the **IdentityAuthority** value is preserved unless it is empty or set to **localhost**. In these cases, the full URL of your local station is added.
	CxSASTManagerUri (i.e. SAST Manager URI)	{Protocol}://{Machine}:{Port} Default HTTP port = 80
	CxARMURL (i.e.CxAnalytics URL)	{Protocol}://{Machine}:{Port} Default HTTP port = 8080
	CxARMPolicyURL (i.e.Policy Manager URL)	{Protocol}://{Machine}:{Port} Default HTTP port = 8080
	ActiveMessageQueueURL (i.e.ActiveMQ URL)	{Protocol}://{Machine}:{Port} Default TCP port = 61616
config.CxEngineConfigurationKeysMeta	**ACTIVE_MESSAGE_QUEUE_URL** (i.e.ActiveMQ URL)	{Protocol}://{Machine}:{Port} Default TCP port = 61616
accesscontrol.ConfigurationItems	**SERVER_PUBLIC_ORIGIN**	Change according to the following conditions: If the machine is configured for IP, then: **SERVER_PUBLIC_ORIGIN** = {Protocol}://{Machine_IP}:{port} and **IdentityAuthority** = {protocol}://{Machine_IP}:{port}/CxRestAPI/auth If the machine is configured for FQDN and SERVER_PUBLIC_ORIGIN is not empty, then: **SERVER_PUBLIC_ORIGIN** = {protocol}://{Machine_FQDN}:{port} and **IdentityAuthority** = {Protocol}//{Machine_FQDN}:{port}/CxRestAPI/auth If the machine is configured for FQDN and SERVER_PUBLIC_ORIGIN is empty, then leave empty. In all instances, the protocol should be the same for both **SERVER_PUBLIC_ORIGIN** and IdentityAuthority keys.

3. Save your changes.
4. On the CxManager machine, perform IIS reset (run ‘iisreset’ from elevated CMD or Reset action for the server name in IIS Console).
5. Restart all Cx Windows Services.

Manual Changes to SAML Identity Provider Attributes for Upgrading from CxSAST V8.8/8.9 to v9.x

This instruction defines the procedure for the manual changes that need to be performed within the SAML Identity Provider, for User and Team Attributes, when upgrading to CxSAST v9.x from v8.8\8.9.

After upgrading SAST 9.0, make sure to modify the link for the Sign-On URL for the SAML server from **http{s}://{server}:{port}/CxRestAPI/auth/samlAcs** to **http{s}://{server}:{port}/CxRestAPI/auth/identity/samlAcs**. Otherwise the access link to the SAML server is broken as the login page of the SAML server cannot be reached.

Please note that the URL is **case-sensitive**.

Prerequisites

The following prerequisites must be in place:

- Checkmarx CxSAST/CxOSA v9.0.0 (with Access Control migration performed)
- Active SAML 2.0 identity provider account (e.g. OKTA)

User Attribute Changes

The following manual changes must be performed within the Identity Provider when upgrading from 8.8\8.9 to v9.x:

- Change 'Organization_Tree' to 'Team' and define multi-value attribute
- Update 'Role' to multi-value attribute
- Remove 'Is_Auditor' and replace with a role
- Remove 'Role_Attribute' and replace with a set of roles

Refer to Creating and Mapping User Attributes in OKTA for more information.

Team Attribute Changes

Each user can be assigned to multiple teams. A ‘String Array’ type should be defined for Team attribute. Each team assignment requires an additional sub-attribute. For v9.0.0, the backslash ‘\’ is replaced with a slash ‘/’.

For example: Team=/CxServer/Team1
/CxServer/Team2
/CxServer/Team3)

Refer to Creating and Mapping User Attributes in OKTA for more information.

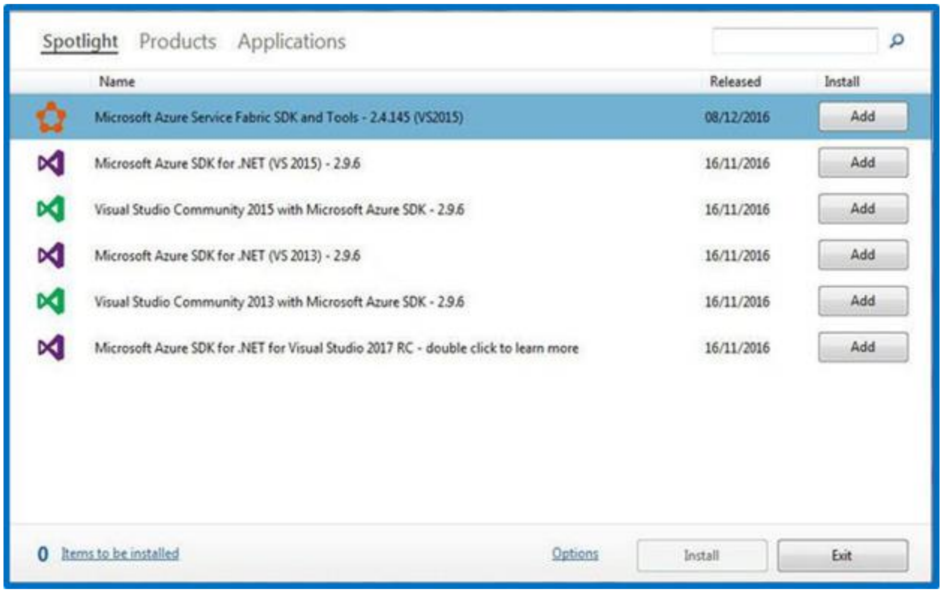
CxSAST Server Web Portal Installed on Dedicated Hosts (v8.8.0 and up)

CxSAST supports Distributed Architecture, where any or all of the CxSAST server components are installed on dedicated hosts.

The following procedure must be implemented in all installations or upgrades to any version that includes the new IIS application (8.1.0 and up).

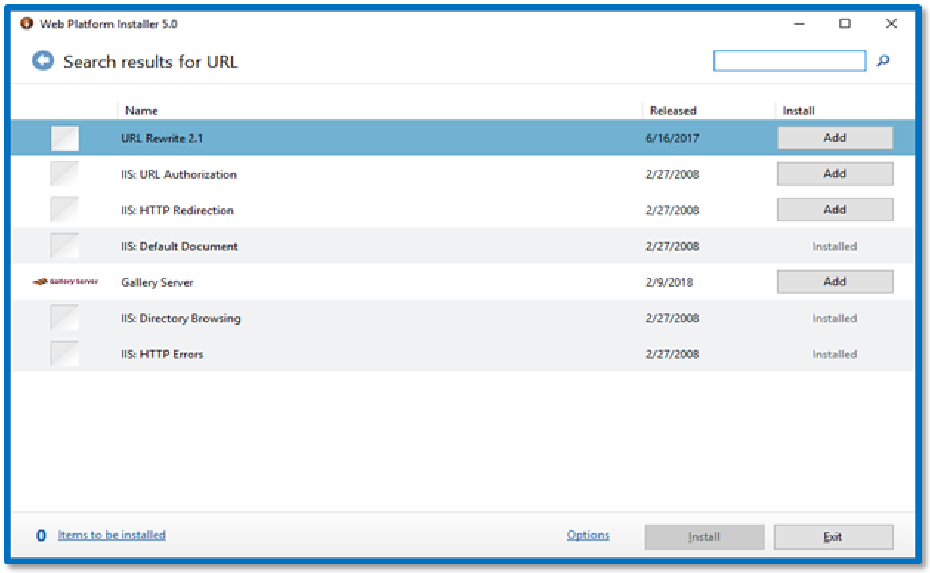
Once the IIS application components of the CxSAST setup have been installed, perform the following procedures:

1. Go to the Microsoft Web Platform Installer and click **Install this extension to download the installation file.**
2. Run the **Microsoft Web Platform Installer 5.0** on the **Portal Server**. The **Microsoft Web Platform Installer** is displayed.



Name	Released	Install
Microsoft Azure Service Fabric SDK and Tools - 2.4.145 (VS2015)	08/12/2016	Add
Microsoft Azure SDK for .NET (VS 2015) - 2.9.6	16/11/2016	Add
Visual Studio Community 2015 with Microsoft Azure SDK - 2.9.6	16/11/2016	Add
Microsoft Azure SDK for .NET (VS 2013) - 2.9.6	16/11/2016	Add
Visual Studio Community 2013 with Microsoft Azure SDK - 2.9.6	16/11/2016	Add
Microsoft Azure SDK for .NET for Visual Studio 2017 RC - double click to learn more	16/11/2016	Add

3. Search for the **Add URL Rewrite 2.1** module and click **Add.**

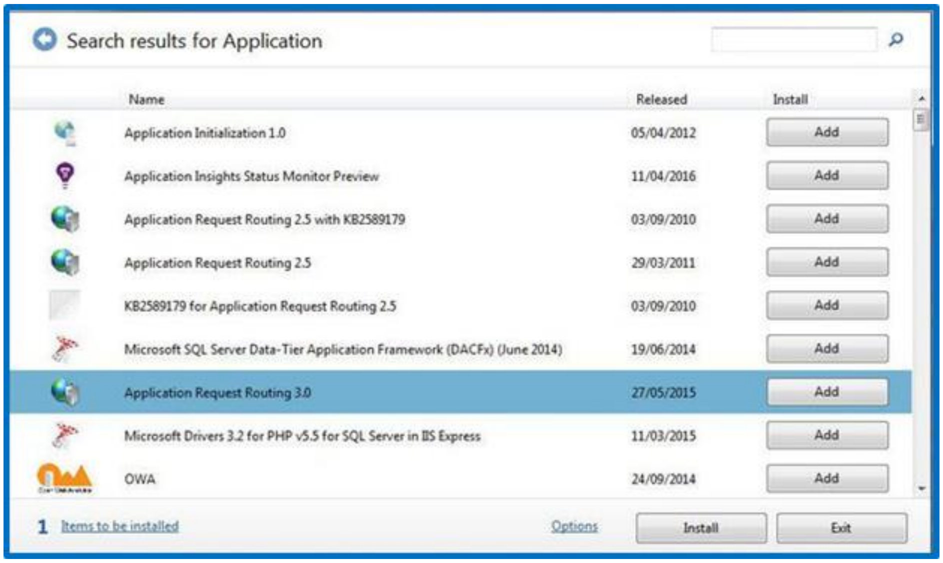


Web Platform Installer 5.0

Search results for URL

Name	Released	Install
URL Rewrite 2.1	6/16/2017	Add
IIS: URL Authorization	2/27/2008	Add
IIS: HTTP Redirection	2/27/2008	Add
IIS: Default Document	2/27/2008	Installed
Gallery Server	2/9/2018	Add
IIS: Directory Browsing	2/27/2008	Installed
IIS: HTTP Errors	2/27/2008	Installed

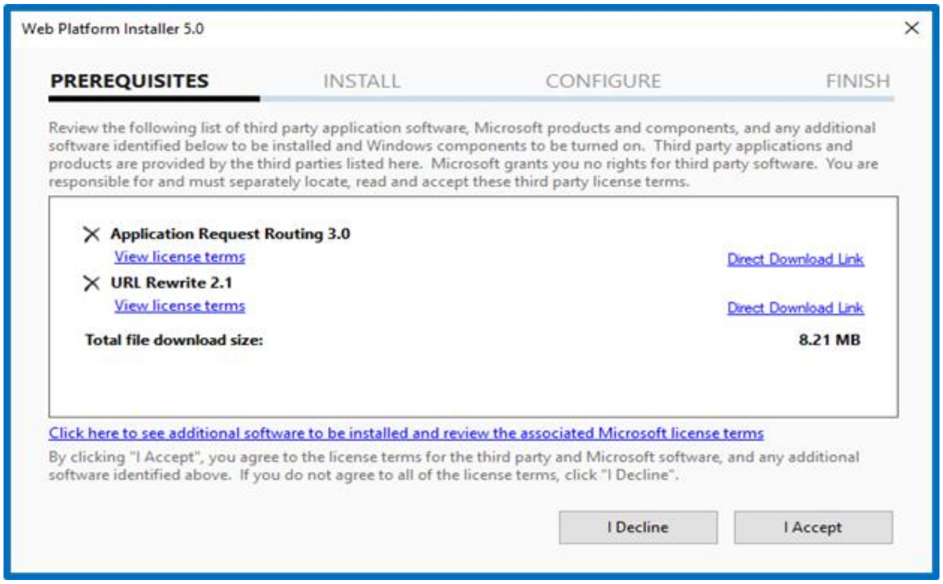
4. Search for the **Application Request Routing 3.0** module and click **Add**.



Search results for Application

Name	Released	Install
Application Initialization 1.0	05/04/2012	Add
Application Insights Status Monitor Preview	11/04/2016	Add
Application Request Routing 2.5 with KB2589179	03/09/2010	Add
Application Request Routing 2.5	29/03/2011	Add
KB2589179 for Application Request Routing 2.5	03/09/2010	Add
Microsoft SQL Server Data-Tier Application Framework (DACFx) (June 2014)	19/06/2014	Add
Application Request Routing 3.0	27/05/2015	Add
Microsoft Drivers 3.2 for PHP v5.5 for SQL Server in IIS Express	11/03/2015	Add
OWA	24/09/2014	Add

5. Click **Install**. The prerequisites for Web Platform Installer 5.0 are displayed.



Web Platform Installer 5.0

Review the following list of third party application software, Microsoft products and components, and any additional software identified below to be installed and Windows components to be turned on. Third party applications and products are provided by the third parties listed here. Microsoft grants you no rights for third party software. You are responsible for and must separately locate, read and accept these third party license terms.

✕ Application Request Routing 3.0
View license terms — Direct Download Link

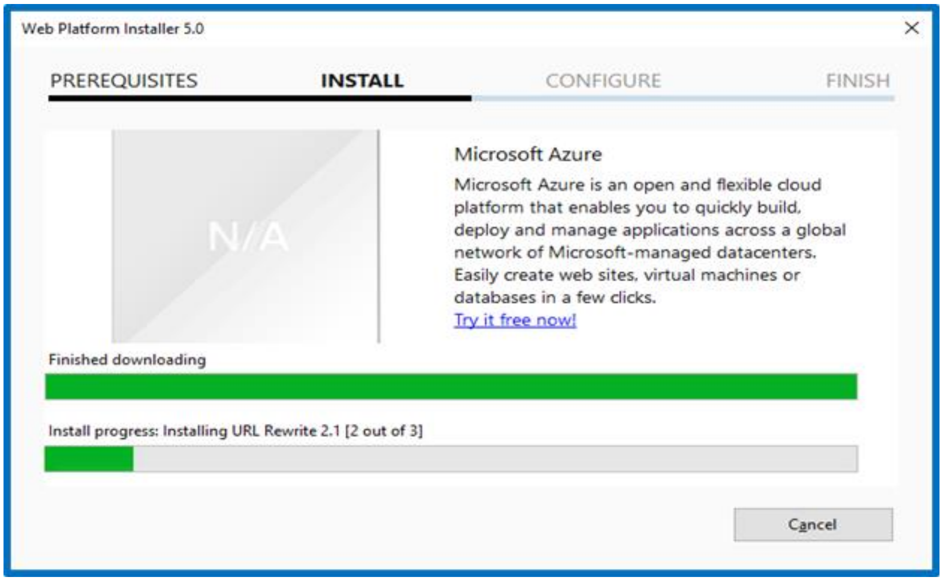
✕ URL Rewrite 2.1
View license terms — Direct Download Link

Total file download size: 8.21 MB

Click here to see additional software to be installed and review the associated Microsoft license terms

By clicking "I Accept", you agree to the license terms for the third party and Microsoft software, and any additional software identified above. If you do not agree to all of the license terms, click "I Decline".

6. Click **I Accept**. The installation progress is displayed. You are notified upon successful installation.



Web Platform Installer 5.0

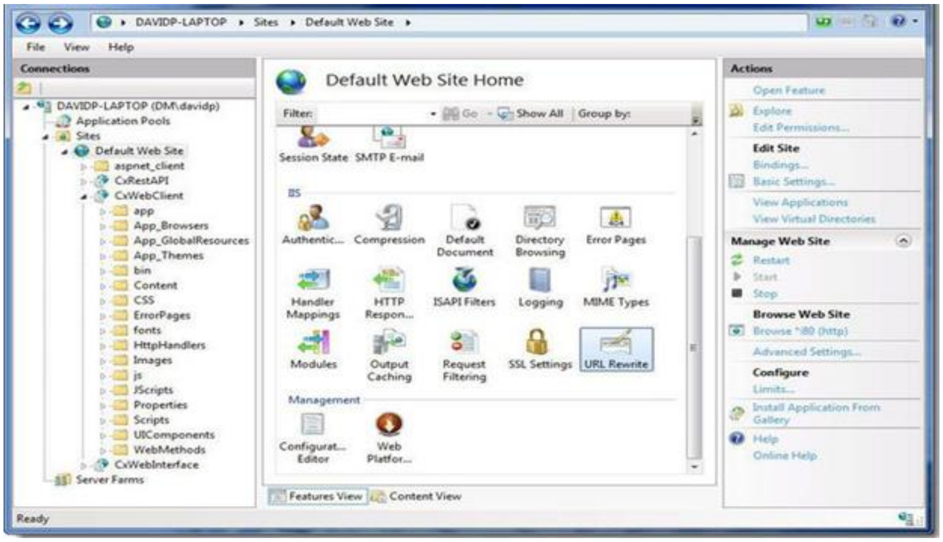
Microsoft Azure

Microsoft Azure is an open and flexible cloud platform that enables you to quickly build, deploy and manage applications across a global network of Microsoft-managed datacenters. Easily create web sites, virtual machines or databases in a few clicks.
Try it free now!

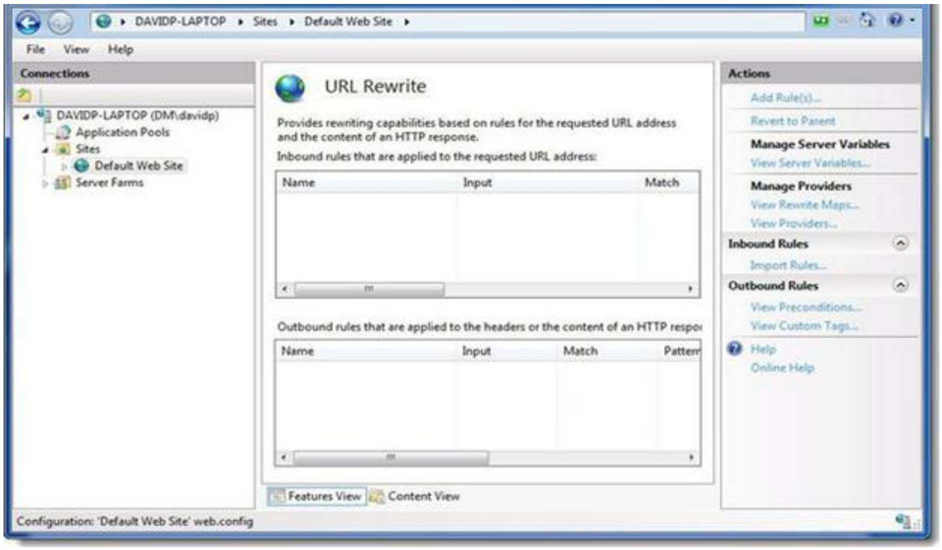
Finished downloading

Install progress: Installing URL Rewrite 2.1 [2 out of 3]

7. Click **Finish**.
8. Open the **Internet Information Services (IIS) Manager** on the Portal Server (**IIS Manager > Sites > Default Web Site > IIS > URL Rewrite**).



9. Select **Open Feature**. The **URL Rewrite Rule** is displayed.

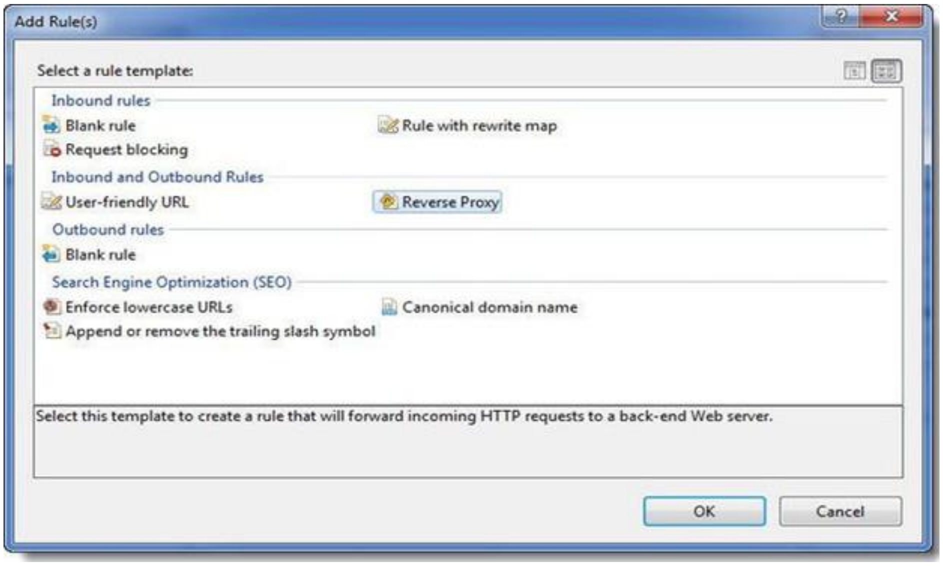


Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

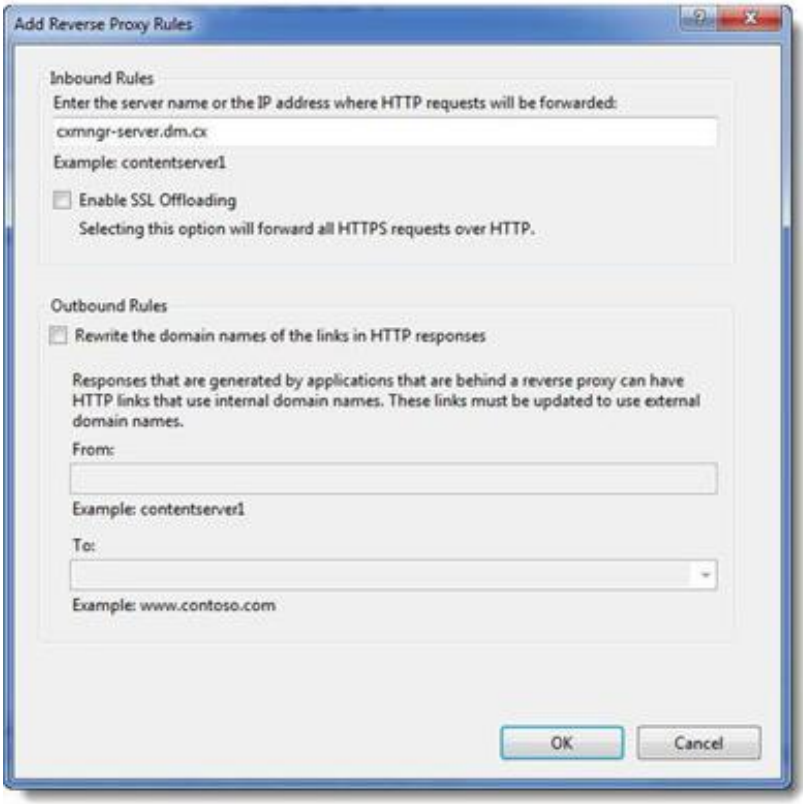
Outbound rules that are applied to the headers or the content of an HTTP respo

10. Select **Add Rule(s)**. The **Rule Templates List** is displayed.



Select this template to create a rule that will forward incoming HTTP requests to a back-end Web server.

11. Select **Reverse Proxy.** The **Add Reverse Proxy Rules** dialog is displayed.



Enter the server name or the IP address where HTTP requests will be forwarded:

cxmngr-server.dm.cx

Example: contentserver1

□ Enable SSL Offloading

Selecting this option will forward all HTTPS requests over HTTP.

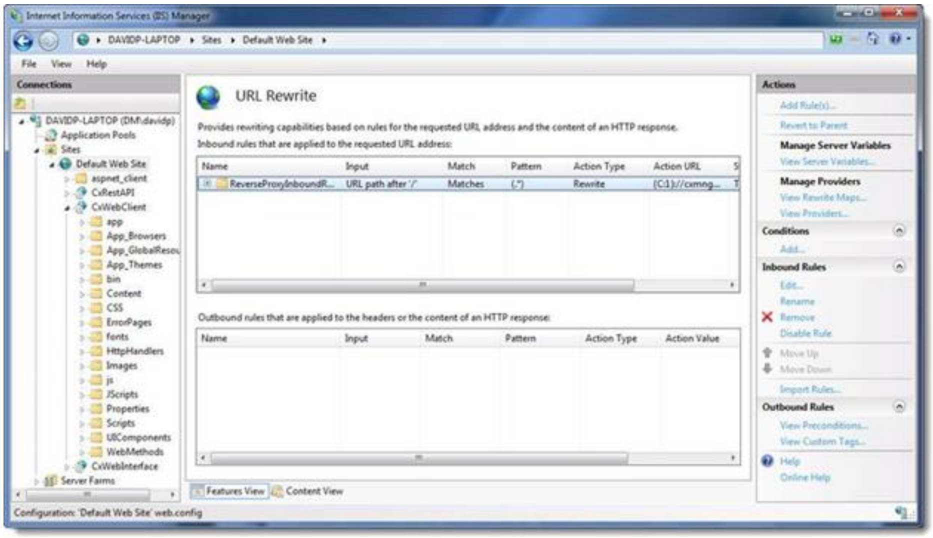
□ Rewrite the domain names of the links in HTTP responses

Responses that are generated by applications that are behind a reverse proxy can have HTTP links that use internal domain names. These links must be updated to use external domain names.

Example: contentserver1

Example: www.contoso.com

12. Enter the **CX Manager Server** name into the **Inbound Rules** field (e.g. cxmngr-server.dm.cx).
13. Disable the **SSL Offloading** option.
14. Click **OK** to save the changes.



URL Rewrite

Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

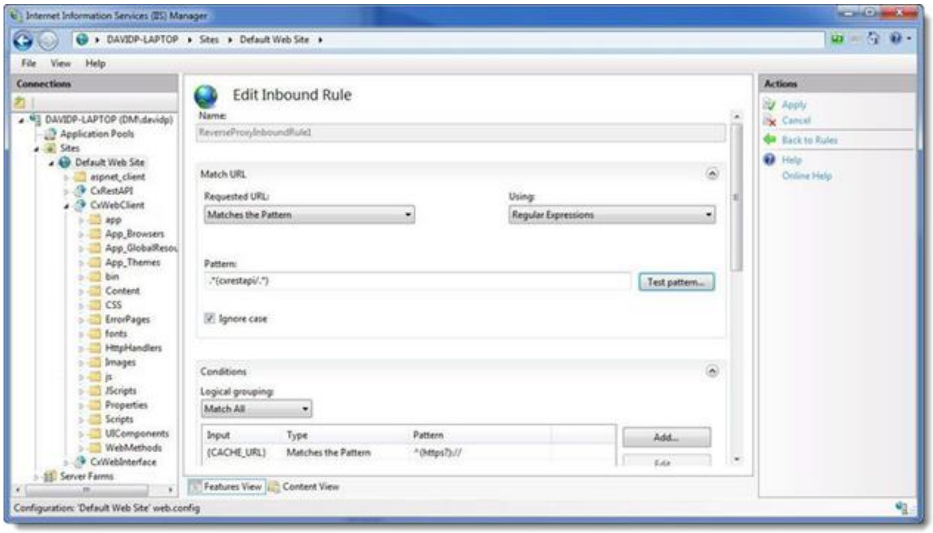
Name	Input	Match	Pattern	Action Type	Action URL	S
ReverseProxyInboundR...	URL path after '/'	Matches	(.*)	Rewrite	{C:1}://cxmng...	T

Outbound rules that are applied to the headers or the content of an HTTP response:

Name	Input	Match	Pattern	Action Type	Action Value

Configuration: 'Default Web Site' web.config

15. Select the newly created **Rule** and under Inbound Rules (right pane), click **Edit**. The **Edit Inbound Rule** window is displayed.



Edit Inbound Rule

Name: ReverseProxyInboundRule1

Match URL

Requested URL: Matches the Pattern

Using: Regular Expressions

Pattern: .*(cxrestapi/.*)

☑ Ignore case

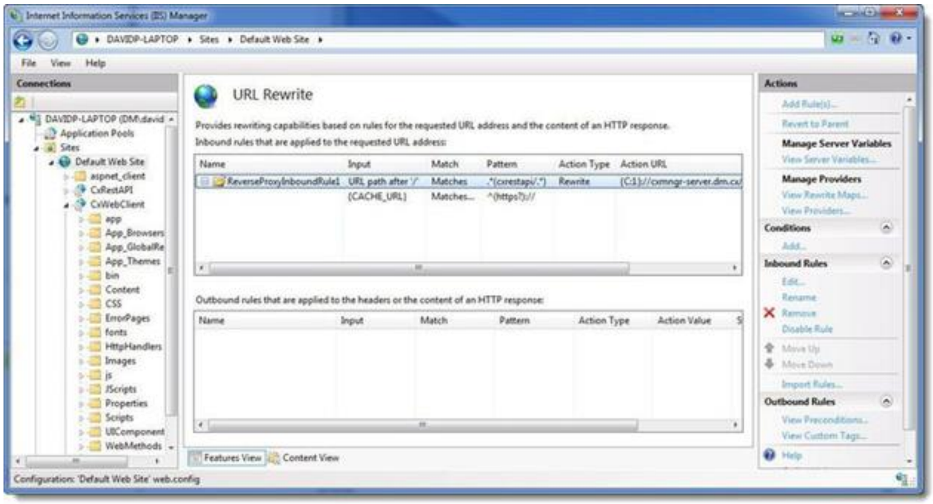
Conditions

Logical grouping: Match All

Input	Type	Pattern
{CACHE_URL}	Matches the Pattern	^(https?)://

Configuration: 'Default Web Site' web.config

16. Change the **Pattern** to .*(cxrestapi/.*) and click **Apply**.
17. Verify the changes in the URL Rewrite rule.



Provides rewriting capabilities based on rules for the requested URL address and the content of an HTTP response.

Inbound rules that are applied to the requested URL address:

Name	Input	Match	Pattern	Action Type	Action URL
ReverseProxyInboundRule1	URL path after '/'	Matches	.*(cxrestapi/.*)	Rewrite	{C:1}://cxmngr-server.dm.cx/
	{CACHE_URL}	Matches...	^(https?)://		

Outbound rules that are applied to the headers or the content of an HTTP response:

Name	Input	Match	Pattern	Action Type	Action Value	S

Configuration: 'Default Web Site' web.config

On the Web portal machine (directory: C:\Program Files\Checkmarx\CheckmarxWebPortal\Web), open the ‘web.config’ file in a text editor and update the value of “**CxWSResolver.CxWSResolver**”with the Manager server IP/domain name.

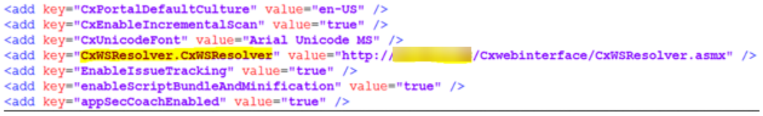
Example:

from....

<add key="CxWSResolver.CxWSResolver" value="http://localhost:80/Cxwebinterface/CxWSResolver.asmx" />

to....

<add key="CxWSResolver.CxWSResolver" value="http://manager-domain-name.com/Cxwebinterface/CxWSResolver.asmx" />



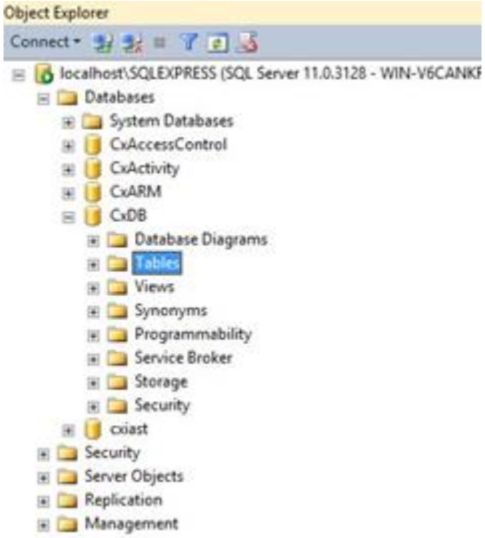
```
<add key="CxPortalDefaultCulture" value="en-US" />
<add key="CxEnableIncrementalScan" value="true" />
<add key="CxUnicodeFont" value="Arial Unicode MS" />
<add key="CxWSResolver.CxWSResolver" value="http://[redacted]/Cxwebinterface/CxWSResolver.asmx" />
<add key="EnableIssueTracking" value="true" />
<add key="enableScriptBundleAndMinification" value="true" />
<add key="appSecCoachEnabled" value="true" />
```

18. Test the **CxSAST** application.

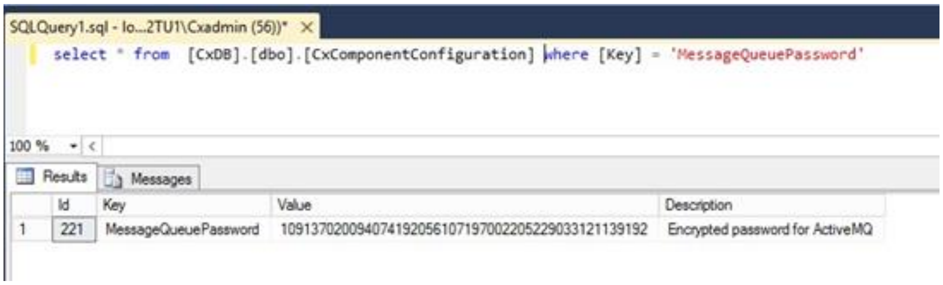
Configuring the ActiveMQ Password

The ActiveMQ **cxuser** account password is stored in the Checkmarx SAST 9.0 Database. To change the default password, the encryption secret must be changed as well. To change the password and the encryption secret, follow the instructions below:

1. Open **MSSQL Server Management Studio.**
2. Connect to the SQL server.
3. Navigate to **Databases > CxDB > Tables.**



4. Under **Tables,** navigate to [CxDB].[dbo].[CxComponentConfiguration].
5. Enter the desired AMQ password (for example ***MyPassword***) in plain text as follows: update [CxDB].[dbo].[CxComponentConfiguration] set **MyPassword** [Value] = '' where [Key] = 'MessageQueuePassword'.



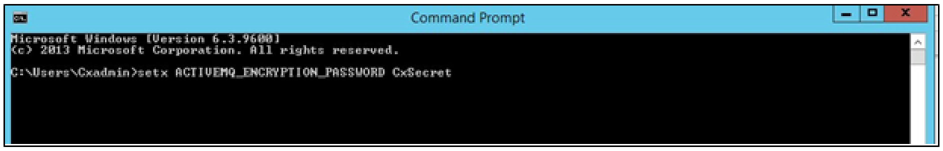
```
select * from [CxDB].[dbo].[CxComponentConfiguration] where [Key] = 'MessageQueuePassword'
```

	Id	Key	Value	Description
1	221	MessageQueuePassword	109137020094074192056107197002205229033121139192	Encrypted password for ActiveMQ

6. Restart the IIS Service and browse to the Checkmarx portal to force the clear text password to be encrypted.
7. Enter **cmd** in the Windows search field to open the command prompt.

8. Change the value of environment variable ACTIVEMQ_ENCRYPTION_PASSWORD to the desired secret in plain text (for example ***CxSecret***):

   setx ACTIVEMQ_ENCRYPTION_PASSWORD **CxSecret**



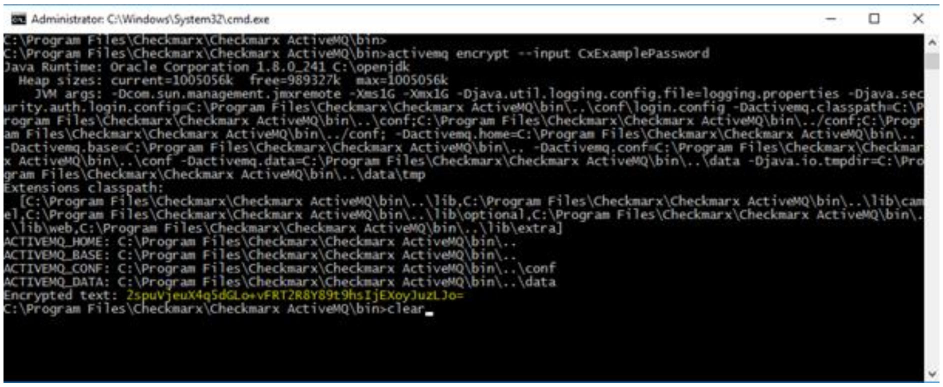
```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Cxadmin>setx ACTIVEMQ_ENCRYPTION_PASSWORD CxSecret
```

- After setting a new environment variable, it is best to restart your machine in order to make sure that ActiveMQ process will load the new value.

9. Run the standard ActiveMQ utility to encrypt the secret by using the previously selected password:

   > cd <Checkmarx Install Folder>\Checkmarx ActiveMQ\bin > activemq encrypt --password **CxSecret** --input **MyPassword**

10. Copy the encrypted password from the output (yellow fonts in the example below).



```
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>activemq encrypt --input CxExamplePassword
Java Runtime: Oracle Corporation 1.8.0_241 C:\openjdk
  Heap sizes: current=1005056k  free=989327k  max=1005056k
    JVM args: -Dcom.sun.management.jmxremote -Xms1G -Xmx1G -Djava.util.logging.config.file=logging.properties -Djava.sec
urity.auth.login.config=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf\login.config -Dactivemq.classpath=C:\P
rogram Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf;C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\../conf;C:\Progr
am Files\Checkmarx\Checkmarx ActiveMQ\bin\../conf; -Dactivemq.home=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
-Dactivemq.base=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\.. -Dactivemq.conf=C:\Program Files\Checkmarx\Checkmar
x ActiveMQ\bin\..\conf -Dactivemq.data=C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data -Djava.io.tmpdir=C:\Pro
gram Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data\tmp
Extensions classpath:
  [C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\cam
el,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\optional,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\.
.\lib\web,C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\lib\extra]
ACTIVEMQ_HOME: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
ACTIVEMQ_BASE: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..
ACTIVEMQ_CONF: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\conf
ACTIVEMQ_DATA: C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin\..\data
Encrypted text: 2spuVjeuX4q5dGLo+vFRT2R8Y89t9hsIjEXoyJuzLJo=
C:\Program Files\Checkmarx\Checkmarx ActiveMQ\bin>clear
```

11. Navigate to **C:\Program Files\Checkmarx\Checkmarx ActiveMQ\conf\**

12. Open **credentials-enc.properties** and edit the password variable by changing the value between the brackets () to the new encrypted text.

13. Restart **Checkmarx ActiveMQ service** .

14. Navigate to table: [Config].[CxEngineConfigurationKeysMeta]

15. Update the AMQ password for the Engine configuration as follows:

    update [Config].[CxEngineConfigurationKeysMeta] set [DefaultValue] = (SELECT TOP 1 [Value] FROM [CxDB].[dbo].[CxComponentConfiguration] where [Key] = 'MessageQueuePassword') where [KeyName]='MESSAGE_QUEUE_PASSWORD'

16. Restart the Checkmarx Services.

CxSAST Engine Configuration (v9.2.0 and up)

The engine configuration information on this page is solely for CxSAST administrators and it is mostly for information purposes.

- It is recommended to consult with Checkmarx support before changing any values.

Parameter Name	Value Type	Default Value
ABS_INT_RESOLVE_MEMBER_ACCESSES_LANGUAGES	string	["JavaScript"]
ACTIVE_MESSAGE_QUEUE_URL	string	
CALCULATE_CONFIDENCE_LEVEL	bool	true
CASE_SENSITIVE_FILENAMES	bool	false
CLIENTS_CONFIDENCE_LEVEL_COLLECT	string	
CONFIDENCE_LEVEL_COLLECT_DATA_FILE_PATH	string	C:\\Temp\\ConfidenceLevel\\
CXAUDIT_TREE_VIEW_FLAT	bool	false
EXCLUDE_PATH	string	jquery;angular.js;angular-animate.js;angular-aria.js;angular-cookies.js;angular-messages.js;angular-mocks.js;angular-resource.js;angular-route.js;angular-sanitize.js;angular-touch.js;angular-scenario.js;angular-loader.js;angular.min.js;angular-resource.min.js;angular-cookies.min.js;angular-loader.min.js;angular-aria.min.js;angular-messages.min.js;angular-mocks.min.js;angular-route.min.js;angular-sanitize.min.js;angular-touch.min.js;angular-scenario.min.js;jsoneditor.js;jsoneditor.min.js
MAX_ALLOWED_RESULTS_FILE_SIZE_IN_MB	int	100
MAX_QUERY_TIME	int	60
MESSAGE_QUEUE_DELAY_BETWEEN_RETRIES	int	1000
MESSAGE_QUEUE_NUMBER_OF_OPEN_RETRIES	int	10
MESSAGE_QUEUE_NUMBER_OF_SEND_RETRIES	int	90
MESSAGE_QUEUE_OPEN_TIMEOUT	int	10
MESSAGE_QUEUE_TTL_DAYS	int	1
NUMBER_OF_RESULTS_FOR_CONFIDENCE_LEVEL_DATA_COLLECTION	int	150
TIME_LIMIT_WAITING_FOR_CONFIDENCE_LEVEL_DATA_COLLECTION	int	180000
USE_ROSLYN_PARSER	bool	true

Parameter Name	Value Type	Default Value
WRITE_CONFIDENCE_LEVEL_TO_LOG	bool	false
ENABLE_SAVE_CPP_PREPROCESSED_FILES	bool	true
ENCODING	string	utf-8
LANGUAGE_THRESHOLD	double	2.0
MULTI_LANGUAGE_MODE	int	1
SCAN_BINARIES	bool	false
SUPPORTED_LANGUAGES	string	1,32;128,256;4,2048
TYPES_TO_DECOMPILE	string	*
PRINT_DEBUG	bool	false
PRINT_LOG	bool	true
ENABLE_CPP_IBM_DECODE	bool	false
BEAUTIFIER_MIN_NUMBER_OF_WORDS_ IN_MINIMIZED_LINE	int	500
BEAUTIFIER_NUMBER_OF_ROWS_TO_CHECK	int	3
BEAUTIFIER_TIMEOUT_IN_SEC	int	180
MAXFILESIZEKB	int	1000
PARAMETER_VALUE_CORES_NUMBER	string	SingleSocket,0;MultiSocket,0
PROCESS_AFFINITY_MANAGER_SETTINGS	string	SingleSocket,NoLimitation; MultiSocket,NoLimitation
MAX_PATH_LENGTH	int	57
MAX_QUERY_TIME_PER_100K	int	15
ENABLE_FICTITIOUS_DEFINITION	bool	false
ABS_INT_LAMBDAS_IMPLICIT_INVOCATION (new in CxSAST 9.2)	bool	false
ABS_INT_CALL_STACK_DEPTH	int	3

Parameter Name	Parameter Description
ABS_INT_RESOLVE_MEMBER_ACCESSES_LANGUAGES	Activate the Abstract Interpretation based resolver to resolve member accesses for specific languages (if ABS_INT_RESOLVE_MEMBER_ACCESSES is set to false).
ACTIVE_MESSAGE_QUEUE_URL	Message queue URL.
CALCULATE_CONFIDENCE_LEVEL	Calculates the Confidence Level for each results and prints it as well as the additional data needed for ML to the results xml.
CASE_SENSITIVE_FILENAMES	For case-sensitive OS (Linux) the value should be true, for non-case-sensitive OS (Windows) it should be false. The value refers to the OS on which the sources compile, not the current OS.

Parameter Name	Parameter Description
CLIENTS_CONFIDENCE_LEVEL_COLLECT	Used when collecting data of results for confidence level future machine learning model training. The values are 'CxAudit' and/or 'EngineAgent' separated with ';' (e.g. CxAudit;EngineAgent).
CONFIDENCE_LEVEL_COLLECT_DATA_FILE_PATH	Used when collecting data of results for confidence level future machine learning model training. The location of the resutls data files.
CXAUDIT_TREE_VIEW_FLAT	Defines the project Treeview structure as flat or regular.
EXCLUDE_PATH	Semicolon separated list of file names to exclude from the scan (e.g. file1;file2;file3). Include only file names, not paths.
MAX_ALLOWED_RESULTS_FILE_SIZE_IN_MB	Max query result file size in MB.
MAX_QUERY_TIME	Defines part of a formula to calculate the maximum execution time allowed for a single query. After the set time, the query execution is terminated, the result is empty and the log indicates that its execution failed.
MESSAGE_QUEUE_DELAY_BETWEEN_RETRIES	The time delay in milliseconds between retries, a connection or sending a message to the message queue.
MESSAGE_QUEUE_NUMBER_OF_OPEN_RETRIES	The number of retries to perform, when opening a message queue connection.
MESSAGE_QUEUE_NUMBER_OF_SEND_RETRIES	The number of retries to perform, when sending a message to the message queue.
MESSAGE_QUEUE_OPEN_TIMEOUT	The time to wait (in seconds) while trying to open a connection to the queue.
MESSAGE_QUEUE_TTL_DAYS	The time unclaimed messages will wait in the MQ before being deleted.
NUMBER_OF_RESULTS_FOR_CONFIDENCE_LEVEL_DATA_COLLECTION	Used when collecting data of results for confidence level future machine learning model training. Defines the maximal number of results that are collected per query.
TIME_LIMIT_WAITING_FOR_CONFIDENCE_LEVEL_DATA_COLLECTION	Limited time in milliseconds to wait for the confidence level data collection tasks.
USE_ROSLYN_PARSER	Enable the use of Roslyn parser to scan C# files.
WRITE_CONFIDENCE_LEVEL_TO_LOG	Write confidence level calculation tracing to a file in order to help understand why a certain confidence level was given to a certain result.
ENABLE_SAVE_CPP_PREPROCESSED_FILES	Enable/disable the ability of CPP Preprocessor to save the preprocessed files.
ENCODING	Character encoding of source files.
LANGUAGE_THRESHOLD	Sub-setting of MULTI_LANGUAGE_MODE. The minimal percentage of complete number of files required to scan a language. Should be set to 0.0 (and MULTI_LANGUAGE_MODE=2) to match the Portal_s Multi-language mode. See MULTI_LANGUAGE_MODE parameter for more details.

Parameter Name	Parameter Description
MULTI_LANGUAGE_MODE	Defines which languages the application should scan. 1 = One Primary Language, 2 = All Languages, 3 = Matching Sets, 4 = Selected Languages.
SCAN_BINARIES	Whether or not to scan binary files (only available for .jar files – Java – and for .dll files – C#). *Note*: Requires Java to be installed on the machine.
SUPPORTED_LANGUAGES	Sub-setting of MULTI_LANGUAGE_MODE. If MULTI_LANGUAGE_MODE = 1 or 2 ignore/meaningless. If MULTI_LANGUAGE_MODE = 4 then languages are separated by commas. See MULTI_LANGUAGE_MODE parameter for more details.
TYPES_TO_DECOMPILE	When SCAN_BINARIES is set to true, this flag should be used to specify which packages/namespaces should be decompiled and then included in the scan. Format x.y.* can be used to specify that all the types under package/namespace x.y should be decompiled and scanned. The list of packages/namespaces should be separated by a semicolon (;).
PRINT_DEBUG	Defines whether writing additional details to application logger with debug orientation is enabled or not. True = Enabled, False = Disabled.
PRINT_LOG	Defines whether the output of Function log.Write is printed to the log or not. True = Print, False = Dont Print.
ENABLE_CPP_IBM_DECODE	Enable the C++ Preprocessor to search, file by file, for IBM pragma filetag directive, in order to find the correct encode.
BEAUTIFIER_MIN_NUMBER_OF_WORDS_IN_MINIMIZED_LINE	BEAUTIFIER: If length of line bigger then this value - this is min.js file.
BEAUTIFIER_NUMBER_OF_ROWS_TO_CHECK	BEAUTIFIER: number of last rows to check. If they are longer than BEAUTIFIER_MIN_NUMBER_OF_WORDS_IN_MINIMIZED_LINE - this is min.js file.
BEAUTIFIER_TIMEOUT_IN_SEC	After this value of seconds the beautification of single file will aborted and the original file will returned. Put 0 to disable the watchdog.
MAXFILESIZEKB	Files exceeding the set size (in KB) will not be scanned.
PARAMETER_VALUE_CORES_NUMBER	Parameter value for method SetToAllCores in EngineInfrastructure.ProcessAffinityManager class - setting cores number for current process. Different parameter for single and multi-socket (e.g. SingleSocket,0;MultiSocket,0).
PROCESS_AFFINITY_MANAGER_SETTINGS	Settings for methods of the EngineInfrastructure.ProcessAffinityManager class. Possible values one of OldVersion,NewVersion, NewVersionOneSocketOnly,NoLimitation. Different parameter for single and multi-socket. (e.g. SingleSocket,OldVersion;MultiSocket,NoLimitation).
MAX_PATH_LENGTH	Defines the maximum amount of flow elements allowed in an influence flow calculation. Paths with length exceeding this number are ignored.
MAX_QUERY_TIME_PER_100K	Sub setting of MAX_QUERY_TIME. Defines part of formula to calculate the maximum execution time allowed for a single query. See MAX_QUERY_TIME parameter for more details.
ENABLE_FICTITIOUS_DEFINITION	Enables the use of the Fictitious Definitions inside the Java Resolver.

Parameter Name	Parameter Description

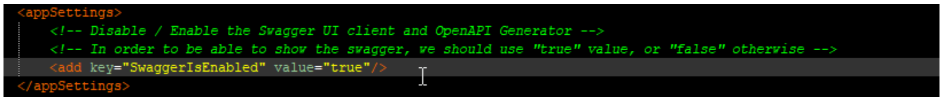
`ABS_INT_LAMBDAS_IMPLICIT_INVOCATION`
`(new in CxSAST 9.2)`
	Currently, lambda expressions are only processed by AbsInt, if they are invoked somewhere. However, in some cases, we want to process the lambda expressions even when their invocations are not explicit (eg: partial scans).
	In order for this to be possible, a flag was added to the Engine configuration: ABS_INT_LAMBDAS_IMPLICIT_INVOCATION (acceptable value is a boolean; default value is false).
	This functionality is still in the test phase and must therefore be used with caution.
	Activating this flag may create unexpected flows or unexpected Abstract values because the lambdas are invoked with the context/environment of their declaration rather than their invocation.
	This flag is linked with ABS_INT_CALL_STACK_DEPTH flag. Since its default value is 3 (levels in stack depth), it might be necessary to increase it in order to find the relevant flows. Be aware that performance will be affected.

`ABS_INT_CALL_STACK_DEPTH`	

Disabling the Swagger UI Client

To disable the Swagger UI, do the following:

1. On the host with CxManager installed, navigate to the Cx installation folder.
2. Under ..\ **Checkmarx\Checkmarx Web RestAPI\CxRestAPI\** edit the web.config file.
3. Search for the **SwaggerIsEnabled** key and change the value to **false**.



```
<appSettings>
    <!-- Disable / Enable the Swagger UI client and OpenAPI Generator -->
    <!-- In order to be able to show the swagger, we should use "true" value, or "false" otherwise -->
    <add key="SwaggerIsEnabled" value="true"/>
</appSettings>
```

4. Restart the CxSystemManager service and IIS.

Configuring CxSAST for High Availability

Configuring Checkmarx Software Exposure Platform for High Availability

Overview

High availability refers to a system that is durable and operates continuously without failure and is always available to the system users and the clients. Such a high availability system is realized by installing CxSAST in a High Availability architecture, where two or more CxManager servers are installed and run in active-active mode and can access the same database to ensure that the system continues operating, if one CxManager fails. The highly available components are the following and are laid out as illustrated:

- ActiveMQ (active-passive)
- CxEngine (active-active)
- CxManager + Access Control (active-active)

ActiveMQ is configured on two hosts to immediately become available in case of failure of the active host. In addition, this configuration allows for load balancing and not just redundancy. In order to configure CxSAST in high availability mode, you have to use an external load balancer (for example Nginx, AWS etc.).

Configuring High Availability

High Availability is configured via the Checkmarx Software Exposure Platform components for each machine/server accordingly. The configuration steps can be performed manually using the following steps:

1. Install all Checkmarx Software Exposure Platform components for the High availability environment independently (not in parallel) according to these instructions.

> - Installing Checkmarx Software Exposure Platform components in parallel could cause database locking issues.
> - If required, rename the servers for all Checkmarx Software Exposure Platform components according to these instructions.

2. Manually add the CxEngine Server(s) according to these instructions, and then remove the default (localhost) CxEngine from the Web Portal.
3. Configure all Checkmarx Software Exposure Platform components for **SOURCE_PATH** and **EX_SOURCE_PATH** - DB table **dbo.cxComponentConfiguration**: Replace the local path (**C:\<folder>\...**) with the relevant network path, for example **\\<hostname>\<folder>**.

> - Server names must be 12 characters or less and must be a part of the domain.

4. Configuring ActiveMQ for High Availability according to these instructions.
5. Configuring Access Control for High Availability according to these instructions.
6. Configuring the Checkmarx Web Portal on a dedicated host according to these instructions.

Restoring the SessionState Value after Upgrading

Users who use the SessionState value in the Web.Config file will have to restore this line in the Web.Config file once the upgrade to Version 9.0.0 is complete. The original file is automatically backed up during the upgrade process. To restore the SessionState value, you have to restore the entire line in the new Web.Config file from the backed-up file as follows:

1. Navigate to **..\Program Files\checkmarx \checkmarxwebportal** and open **web.config_backup** for editing.
2. Search for **<sessionState configSource="sessionState.config" />** and copy the entire line.
3. Open web.config and navigate to the location where this line is located in the backup file.
4. Paste the line in this location, save and then close **web.config**.
5. Do not simply replace the **web.config** with the backup file **web.config_backup** as additions may be lost and the CxSAST application stops operating.

Configuring Access Control for High Availability Environments

This instruction defines the procedure for configuring Access Control in High Availability environments for v9.0.0 and up.

Configuring Access Control for High Availability (HA) in CxSAST v9.0.0 and up ensures optimal operational performance – even at times of high loads, and provides failover support in case of CxManager failure to ensure application availability.

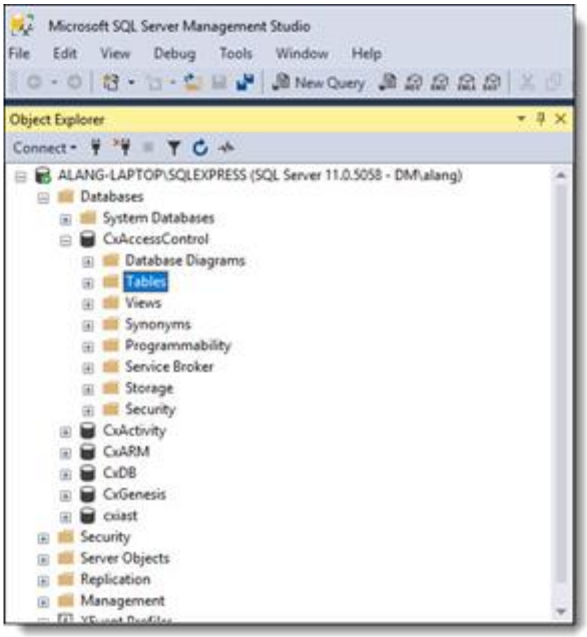
This High Availability architecture supports two or more servers (CxManager) installed behind the organization's external network load balancer that allows for accessing the same DB, to ensure full system operability in the event a machine failure.

- High Availability can be configured on a local server, or via a cloud environment.

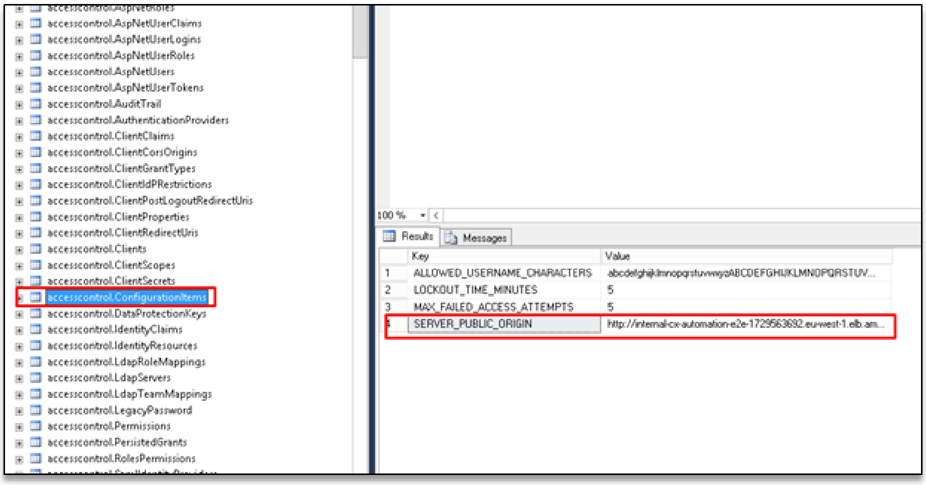
Configuring the CxAccessControl Database

Once the CxSAST v9.0.0 (and up) environment is installed and fully configured, do the following:

1. Open MS SQL Server Management Studio.
2. Connect to the SQL server.
3. Go to: Databases > CxAccessControl > Tables.



4. Right-click on the accesscontrol.ConfigurationItems table and open Edit Top 200 Rows.


accesscontrol.AspNetRoles
accesscontrol.AspNetUserClaims
accesscontrol.AspNetUserLogins
accesscontrol.AspNetUserRoles
accesscontrol.AspNetUsers
accesscontrol.AspNetUserTokens
accesscontrol.AuditTrail
accesscontrol.AuthenticationProviders
accesscontrol.ClientClaims
accesscontrol.ClientCorsOrigins
accesscontrol.ClientGrantTypes
accesscontrol.ClientIdPRestrictions
accesscontrol.ClientPostLogoutRedirectUris
accesscontrol.ClientProperties
accesscontrol.ClientRedirectUris
accesscontrol.Clients
accesscontrol.ClientScopes
accesscontrol.ClientSecrets
accesscontrol.ConfigurationItems
accesscontrol.DataProtectionKeys
accesscontrol.IdentityClaims
accesscontrol.IdentityResources
accesscontrol.LdapRoleMappings
accesscontrol.LdapServers
accesscontrol.LdapTeamMappings
accesscontrol.LegacyPassword
accesscontrol.Permissions
accesscontrol.PersistedGrants
accesscontrol.RolesPermissions
accesscontrol.SamlIdentityProviders
100 %
Results
Messages

	Key	Value
1	ALLOWED_USERNAME_CHARACTERS	abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUV...
2	LOCKOUT_TIME_MINUTES	5
3	MAX_FAILED_ACCESS_ATTEMPTS	5
4	SERVER_PUBLIC_ORIGIN	http://internal-cx-automation-e2e-1729563692.eu-west-1.elb.am...

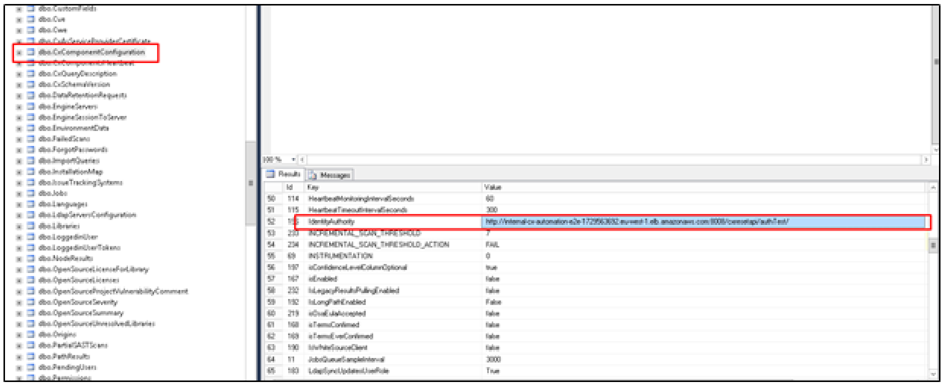
5. In the field of the SERVER_PUBLIC_ORIGIN key, and enter the load balancer URL in the Value field as follows:

   http:// {Access Control URL in Load Balancer: Port}

6. Go to **Databases > CxDB > Tables**, right-click the dbo.CxComponentConfiguration table and then open **Edit Top 200 Rows.**

7. Select the IdentityAuthority key and enter the same load balancer URL (the base URL of the identity server) in the Value field as follows:

   http:// {Access Control URL in Load Balancer: Port} /cxrestapi/auth



dbo.CustomFields
dbo.Cve
dbo.Cwe
dbo.CxAcServiceProviderCertificate
dbo.CxComponentConfiguration
dbo.CxComponentsHeartbeat
dbo.CxQueryDescription
dbo.CxSchemaVersion
dbo.DataRetentionRequests
dbo.EngineServers
dbo.EngineSessionToServer
dbo.EnvironmentData
dbo.FailedScans
dbo.ForgotPasswords
dbo.ImportQueries
dbo.InstallationMap
dbo.IssueTrackingSystems
dbo.Jobs
dbo.Languages
dbo.LdapServersConfiguration
dbo.Libraries
dbo.LoggedinUser
dbo.LoggedinUserTokens
dbo.NodeResults
dbo.OpenSourceLicenseForLibrary
dbo.OpenSourceLicenses
dbo.OpenSourceProjectVulnerabilityComment
dbo.OpenSourceSeverity
dbo.OpenSourceSummary
dbo.OpenSourceUnresolvedLibraries
dbo.Origins
dbo.PartialSASTScans
dbo.PathResults
dbo.PendingUsers
dbo.Permissions

	Id	Key	Value
50	114	HeartbeatMonitoringIntervalSeconds	60
51	115	HeartbeatTimeoutIntervalSeconds	300
52	[illegible]	IdentityAuthority	http://internal-cx-automation-e2e-1729563692.eu-west-1.elb.amazonaws.com:8008/cxrestapi/auth/Test/
53	233	INCREMENTAL_SCAN_THRESHOLD	7
54	234	INCREMENTAL_SCAN_THRESHOLD_ACTION	FAIL
55	69	INSTRUMENTATION	0
56	197	isConfidenceLevelColumnOptional	true
57	167	isEnabled	false
58	232	IsLegacyResultsPullingEnabled	false
59	192	IsLongPathEnabled	False
60	219	isOsaEulaAccepted	false
61	168	isTermsConfirmed	false
62	169	isTermsEverConfirmed	false
63	190	IsWhiteSourceClient	false
64	11	JobsQueueSampleInterval	3000
65	183	LdapSyncUpdatesUserRole	True

Configuring the Host Header in the Load Balancer / Proxy

When working with a load balancer or proxy it is important that requests that reach the Access Control service contain the original 'Host header'. In certain instances, there may be errors, and this is usually because the 'Host' header contains the address of the backend server instead of its original value. By default, the 'Host header' changes once a new request is created. To fix this, you can manually configure the 'Host header' in the load balancer / proxy configuration. For this example, we will use NGINX as the installed load balancer.

➢ **To configure the 'Host header' in the load balancer:**

1. Open the NGINX configuration file (**<nginx installation path>/conf/nginx.conf**) using a text editor.
2. Navigate to the 'http.server.location' segment as illustrated in the example below.

```
worker_processes  auto;
        events{
        worker_connections 4096;
        }
        http {
           include     mime.types;
           index    index.html index.htm index.php;
            default_type application/octet-stream;
            sendfile      on;
            tcp_nopush   on;
            server_names_hash_bucket_size 128;
            upstream vm-s-880 {
            server 10.35.0.30;
            server 10.35.0.31;
        }
        server {
        client_max_body_size 1000M;
        listen 8070;
        location / {
        proxy_pass http://vm-s-880;
        proxy_set_header Host vm-s-880;
        proxy_buffer_size         128k;
        proxy_buffers             4 256k;
        proxy_busy_buffers_size    256k;
        }
       }
    }
```

3. Set the ‘Host header under the ‘http.server.location’ segment as: **proxy_set_header** Host {**load_balancer_address**}, where the **load_balancer_address** is defined as the station on which NGINX is installed.

> - The ‘Upstream’, **proxy_pass** and **proxy_set_header Host** definitions must match.
> - When uploading a large source zip archive to the SAST Portal, make sure that the **Client_max_body_size** value is set to at least the size of this zip archive. The example above reflects this value set to 1000 MB.

4. Save the NGINX configuration file and exit the editor.
5. Restart NGINX.

Configuring ActiveMQ for High Availability Environments

This instruction defines the procedure for configuring ActiveMQ in High Availability (Cluster) environments for v9.0.0 and up.

The ActiveMQ implementation is intended for sending messages between two applications, or two components inside one application. The ActiveMQ support distributed messaging across a network of brokers. This allows a client to connect to any broker in the network - and fail over to another broker if there is a failure - providing from the client’s perspective a high availability cluster of brokers.

Make sure that port 61616 is open in all relevant firewalls between the ActiveMQ server and the following components:

- CxManager servers (for Access Control, Scan Manager and Results Services). This includes high availability configurations with multiple CxManagers.
- CxEngine servers
- M&O server

Configuring ActiveMQ Brokers

Once the CxSAST environment is set up and fully configured, do the following:

1. Navigate to the Checkmarx ActiveMQ\conf folder and open the 'activemq.xml' file.
2. Edit the <persistenceAdapter> tag accordingly:

```
<persistenceAdapter>
<kahaDB directory="/sharedFileSystem/sharedBrokerData"/>
</persistenceAdapter>
```

   Make sure that the DB directory navigates to your shared directory.

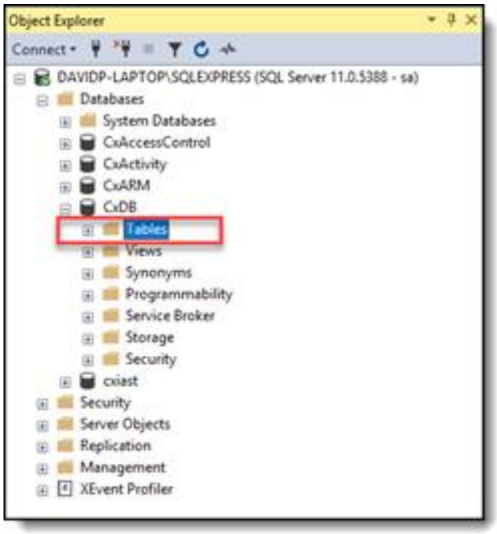
3. Perform the same procedure for all ActiveMQ brokers in the high availability cluster.

Once you have completed the ActiveMQ broker configuration, you can now configure the ActiveMQ clients.

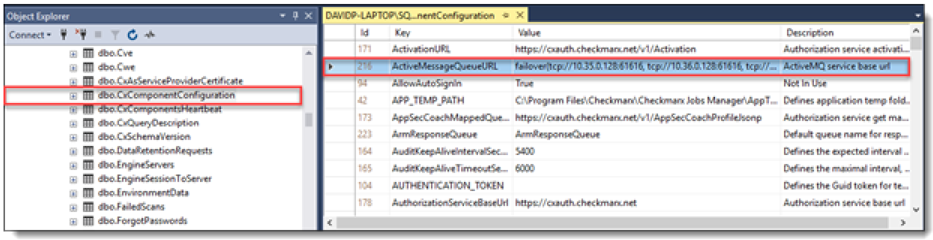
Configuring ActiveMQ Clients

➢ **To configure ActiveMQ clients:**

1. Open MS SQL Server Management Studio.
2. Connect to the SQL server.
3. Go to: Databases > CxDB > Tables.



4. Right-click on the CxComponentConfiguration table and select Edit Rows.


Object Explorer
Connect
dbo.Cve
dbo.Cwe
dbo.CxAsServiceProviderCertificate
dbo.CxComponentConfiguration
dbo.CxComponentsHeartbeat
dbo.CxQueryDescription
dbo.CxSchemaVersion
dbo.DataRetentionRequests
dbo.EngineServers
dbo.EngineSessionToServer
dbo.EnvironmentData
dbo.FailedScans
dbo.ForgotPasswords
DAVIDP-LAPTOP\SQ...nentConfiguration

Id	Key	Value	Description
171	ActivationURL	https://cxauth.checkmarx.net/v1/Activation	Authorization service activati...
216	ActiveMessageQueueURL	failover(tcp://10.35.0.128:61616, tcp://10.36.0.128:61616, tcp://...	ActiveMQ service base url
94	AllowAutoSignIn	True	Not In Use
42	APP_TEMP_PATH	C:\Program Files\Checkmarx\Checkmarx Jobs Manager\AppT...	Defines application temp fold...
173	AppSecCoachMappedQue...	https://cxauth.checkmarx.net/v1/AppSecCoachProfileJsonp	Authorization service get ma...
223	ArmResponseQueue	ArmResponseQueue	Default queue name for resp...
164	AuditKeepAliveIntervalSec...	5400	Defines the expected interval ...
165	AuditKeepAliveTimeoutSe...	6000	Defines the maximal interval, ...
104	AUTHENTICATION_TOKEN		Defines the Guid token for te...
178	AuthorizationServiceBaseUrl	https://cxauth.checkmarx.net	Authorization service base url

5. In the ActiveMessageQueueURL key field, enter each broker URL in the Value field as follows:

```
failover:(tcp://broker1:61616,tcp://broker2:61616,tcp://broker3:61616)
```

6. Perform the same procedure for all ActiveMQ clients in your environment.

Final steps:

1. Open the Windows Services and stop the ActiveMQ service (if the recovery option is enabled, disable it).
2. Once the file lock, inside the shared ActiveMQ folder, disappears, delete the data folder.
3. Restart \Checkmarx\Checkmarx ActiveMQ\bin\win64\activemq.bat (not services.msc).

The files have now been recreated at the new shared location, where the user configured it in the database.