**CODE SCAN USING CHECKMARX**

**Document Prepared by:** Nagabhushan R Anvekar

**Reviewed by:** Swapna Katta

Introduction : Checkmarx is a static code analysis tool used to analyze the code vulnerabilities. Other example for Source code analysis tools are Fortify, Coverity etc., Checkmarx has standard approach towards analyzing Javascript codes and hence it is preferred over other code analysis tools for XSJS and also may be extended for UI5

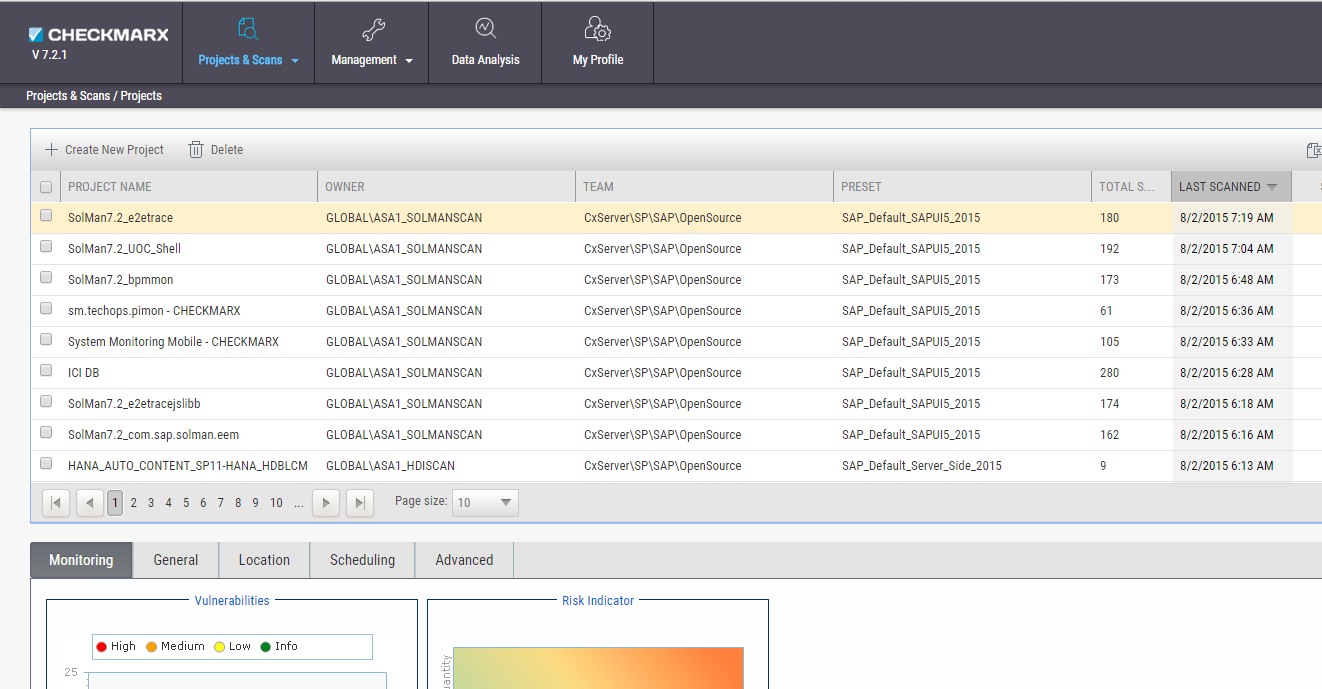
This document provides a brief description how we can perform codescan using Checkmarx for an XSJS project and UI5 projects along with an overview of reported vulnerabilities.

**Table of Content:**

1. Before We Start
2. Exporting an XSJS project from WEB IDE
3. Scanning XSJS Project with Checkmarx
4. Analyzing Results
5. Analyzing UI5 project with Checkmarx
6. Overview of XSJS vulnerabilities
7. Overview of UI5 vulnerabilities
8. Integrating Checkmarx with Jenkins for Continuous Integration
9. **Before We Start:**

For a Checkmarx code scan, the source code of the project needs to imported into the Checkmarx server and then subjected to scan based on the suitable preset. We will discuss about the importing of the project and about the presets in the following sections. Since the analysis of Checkmarx happens at the centralized server there is no need for installation of any local application (except in some condition) but we need to be provided with the access to the centralized server. For requesting access to the server at SAP use the [link](https://jam4.sapjam.com/wiki/show/pGMUe7npYjRqxkEDxBpRX1). And also please join the JAM page at SAP for Checkmax for more details at link: [link](https://jam4.sapjam.com/questions/xa7UIcgIuJGd3lW0YrnvKO)

Below is the screenshot of Checkmarx server at SAP once we get the access



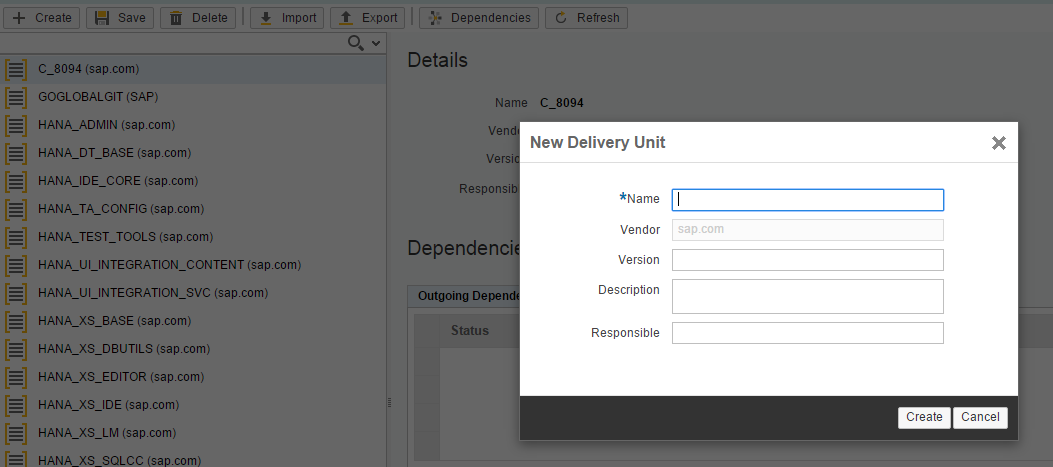
The entries shown in the list in the above screenshot are projects created in the Checkmarx server which are available for immediate and scheduled scans

Before creating and importing a project for code scan, we first have to export our XSJS project from the WEB IDE in the following section detailed steps for the export of project is mentioned.

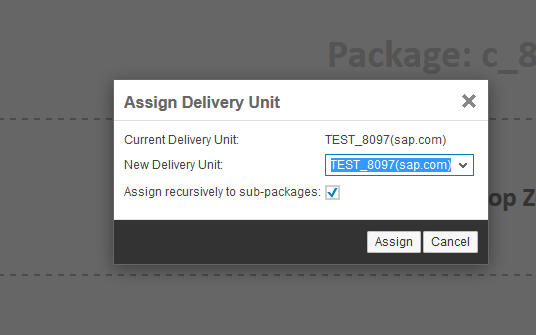
1. **Exporting an XSJS project from WEB IDE**

An XSJS project may be exported in two ways for code scan in Checkmarx. One way is to export the delivery unit of the project and the other way is the export the project package from WEB IDE

Steps to Download the DU Delivery Unit:  
1. Use the Delivery Unit Manager link, <http://lddbjwt:8000/sap/hana/xs/lm/> (For JWT system) click on the Delivery Unit tile, now click on “+create” button and fill relevant details in the popup.



2. Once Delivery Unit is created you can use HANA web IDE. By right clicking on the project you can assign your package with the Delivery Unit as shown below  
Right Click on package -> Assign Delivery Unit



3. You can now download the latest delivery unit from the Delivery Unit manager by clicking the export button as shown. Once you download the delivery unit you have to convert the .tgz file of delivery unit to .zip and also the inner xsjs and xsjslib files have to be suitably converted, you may use the script available at [link](https://jam4.sapjam.com/questions/ftN37iAsH8IgqBL0LHETKP) to do the conversion.

The other way is to simply export the package from WEB IDE, by right clicking on the package you want to export you get options to export your package

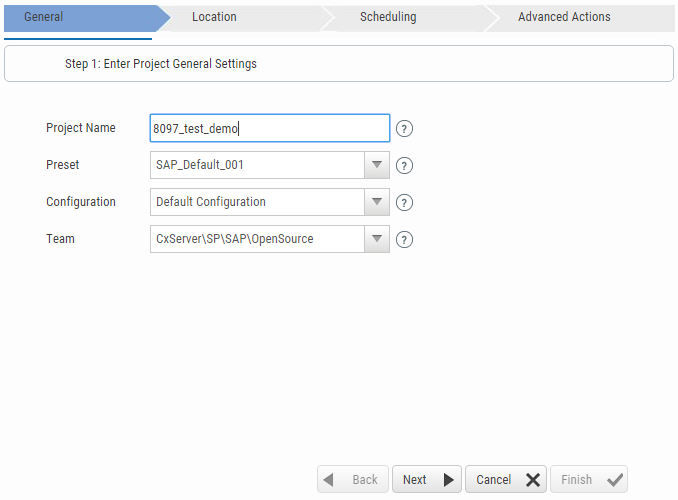
OR

(https://<server-name:8041>/sap/hana/xs/dt/base/xfer/export/<package-name>.zip <- you may use the link to directly download the source zip)

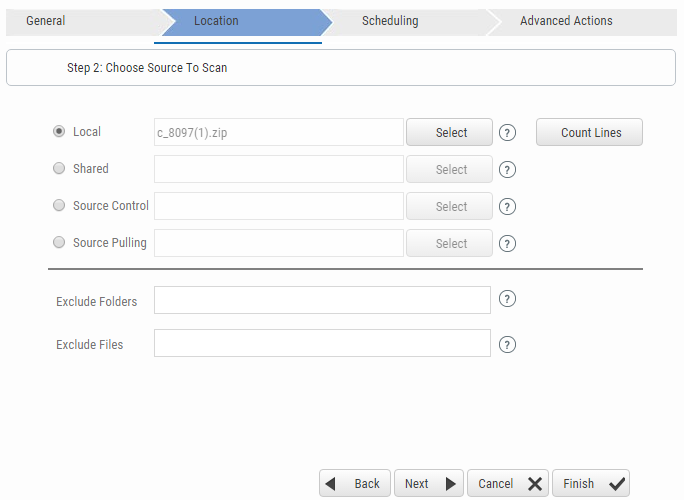
1. **Scanning XSJS Project with Checkmarx**

Once the XSJS project is exported we can perform the scan on the Checkmarx server as mentioned in the below points

1. Click on the “+ Create New Project” in the home page
2. This will lead to the screen as shown :

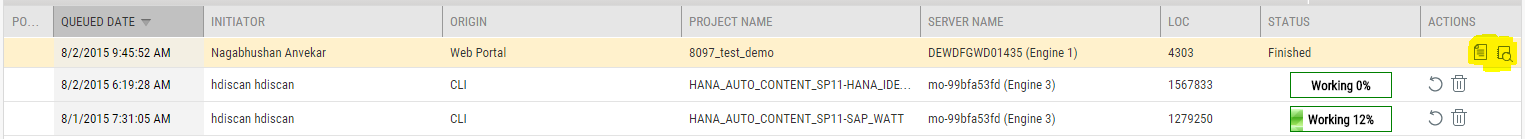


1. Fill the project name with a desired name, Choose a preset option based on the kind of project you are scanning, Presets are set of query or tests that run on your code in order to check vulnerabilities, Refer the [link](https://jam4.sapjam.com/wiki/show/vpoUYz30nZx4pH57d1rJZc) for more details on presets. You may use “SAP\_Default\_Server\_Side\_2015” as this seems to be best suited for XSJS [SAP\_Default\_001 used for current example]. The configuration and Team names have been set to default and click next to proceed
2. In the locations tab we get following options :

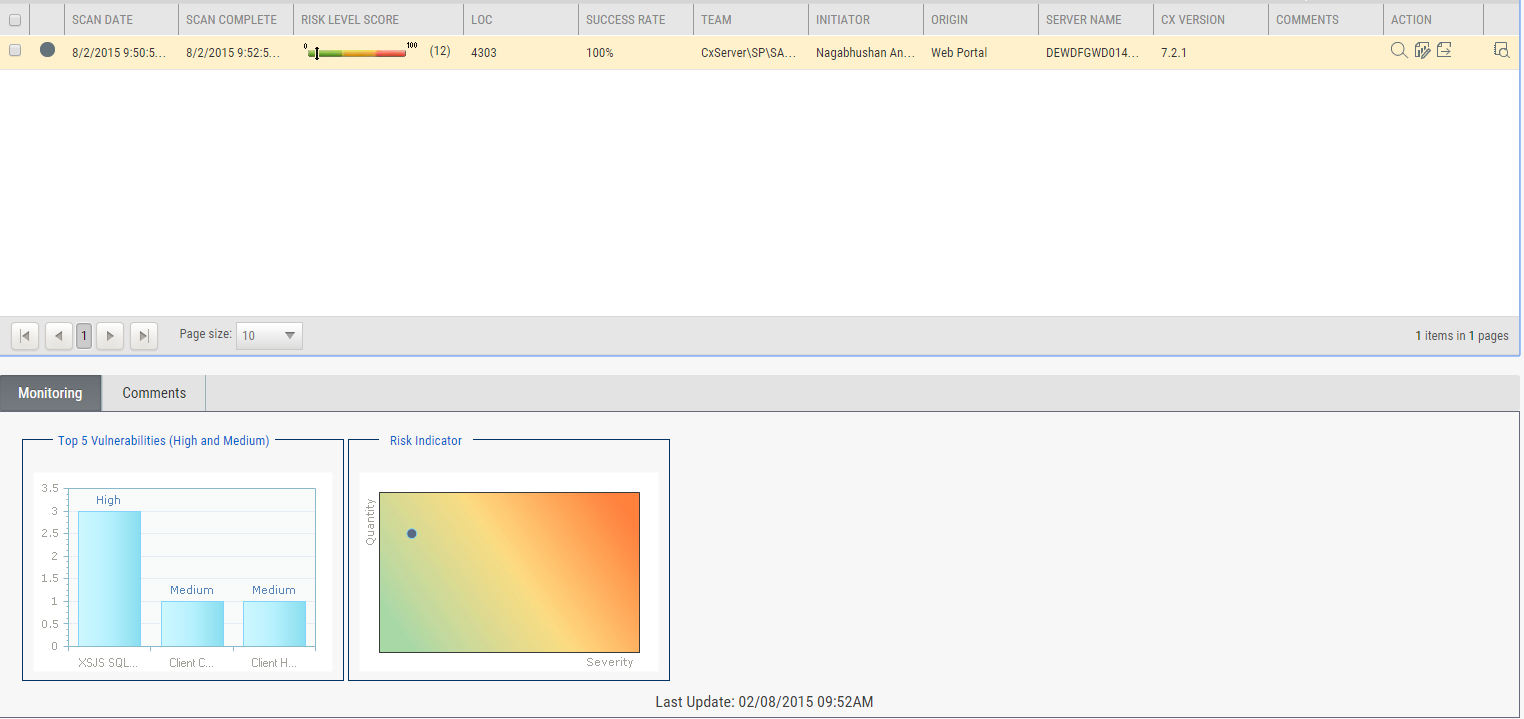


Since XSJS project developed in WEB IDE cannot be mapped with GIT or SVN we have to select the local archive which we downloaded earlier

1. In the scheduling tab, we get only “Now” option highlighted as we did not give source control location so that the CXServer can pull code in scheduled time slots. Proceed to next options
2. In the advanced actions tab, you get options to provide E-mail addresses of recipients who are supposed to receive mails of the scanned reports. Provide the mail addresses and proceed by finishing
3. In the next screen you get a list of all the projects queued for scanning in the server, regular updates are provided on the scanning statuses and Scan would be finished depending on the size of the projects



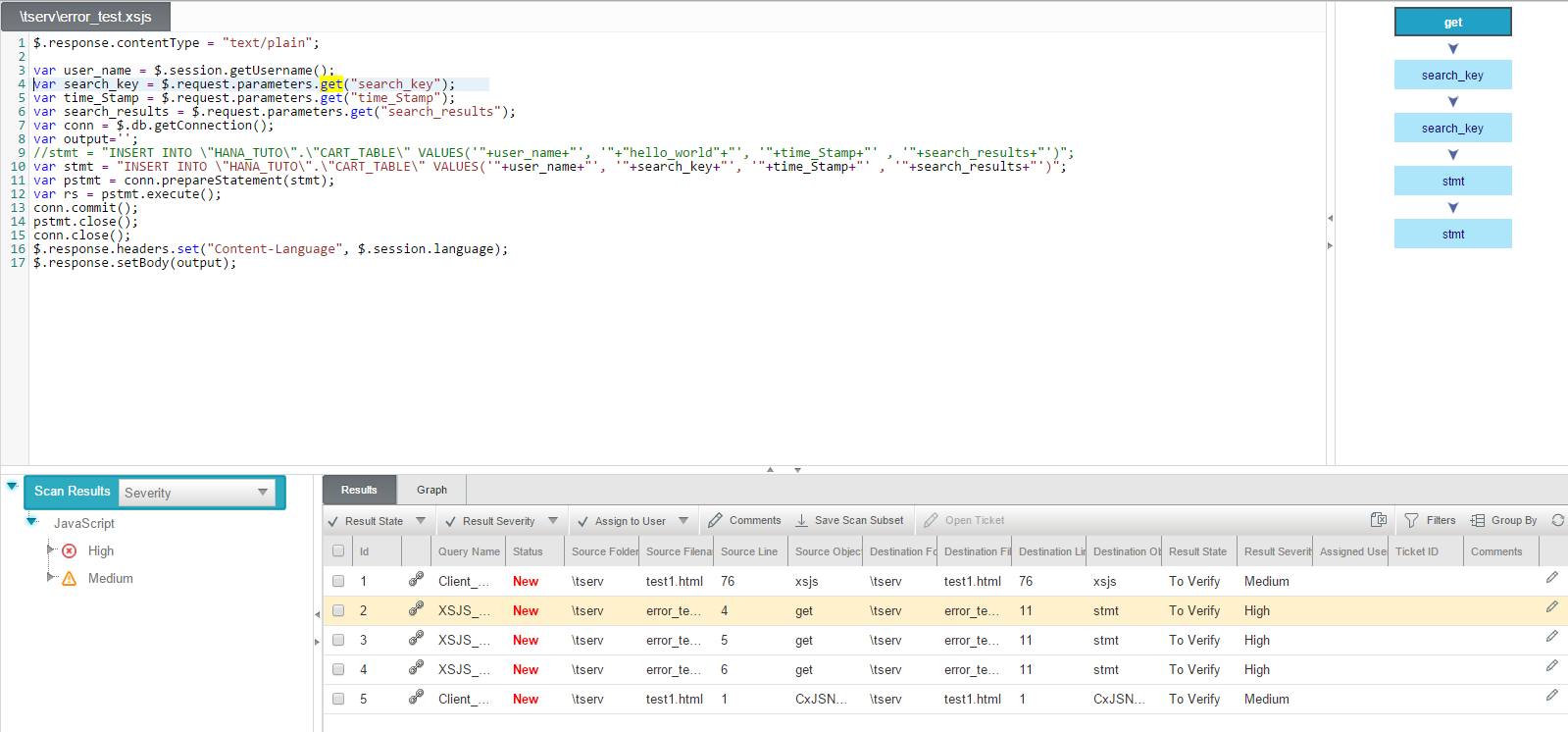
1. **Analyzing Results**
2. On clicking check results(High lighted in yellow in previous screenshot) a new screen appears giving overall scan results as shown :



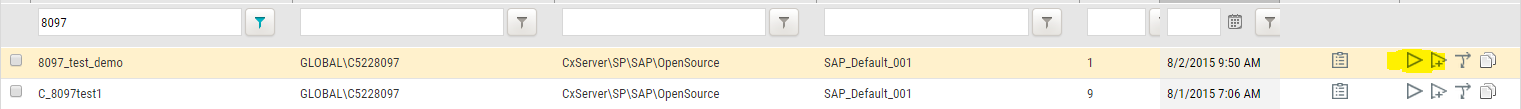
1. Click on “Open Viewer” to get detailed view of risks :



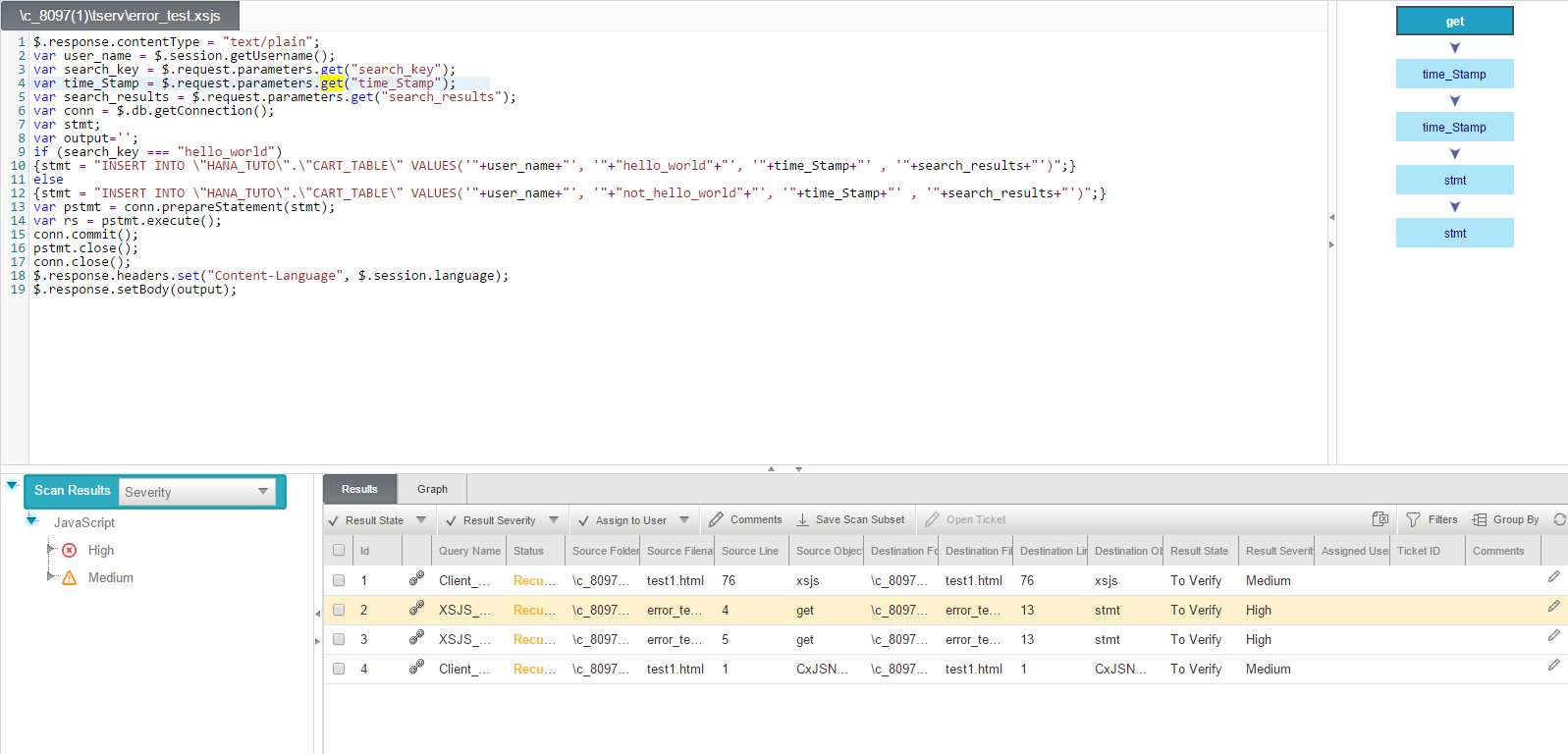
As highlighted above



1. Here the risk is reported as the variable read from user input is used directly in SQL without sanitizing (Possible SQL Injection). In the next scan we will change this by updating new value in the query and analyze. You can do a rescan on the same project by clicking full scan as shown :



1. With the code change the high risks are reduced by one.



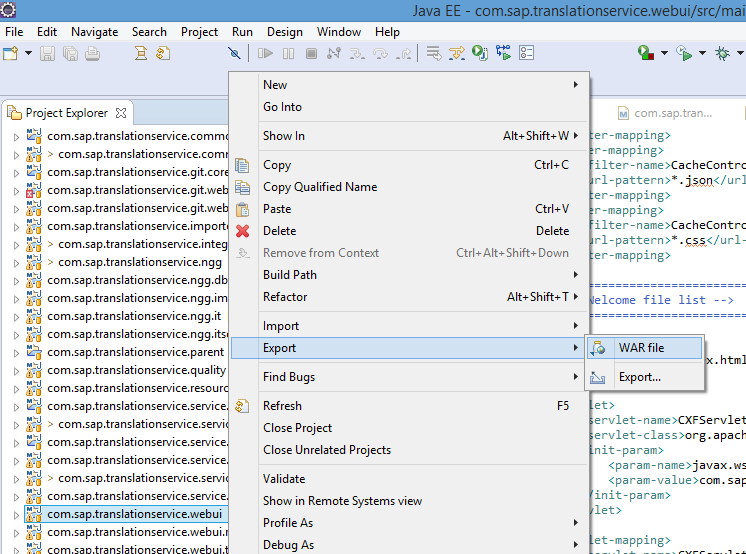
Here the search key is not directly used in the query instead enclosed with IF condition and hence the risk avoided

1. **Analyzing UI5 project with Checkmarx**

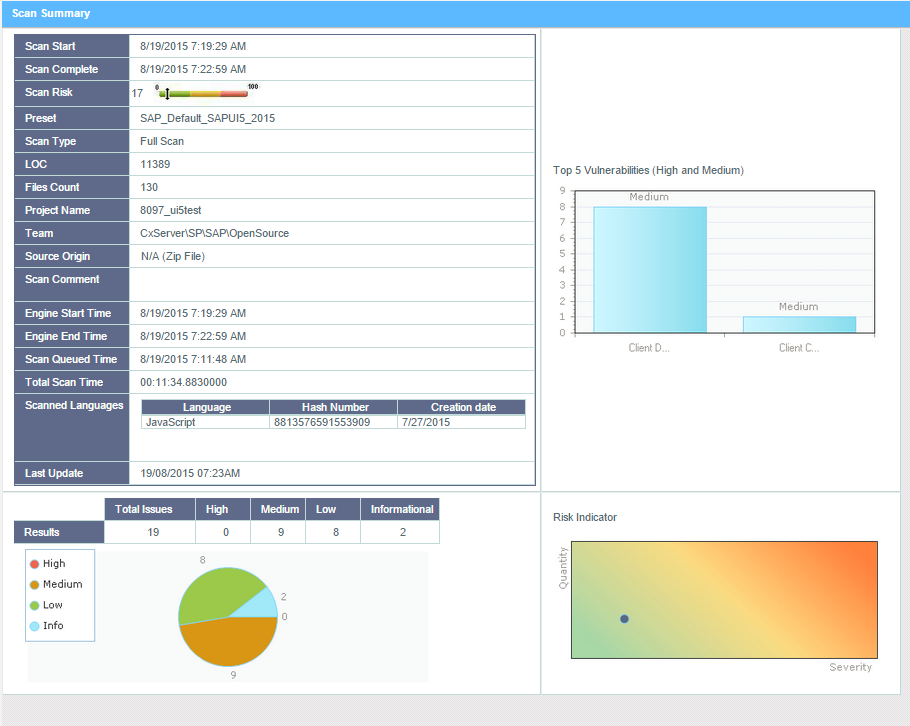
Analyzing an UI5 project with Checkmarx is very much similar to analyzing other code like that of XSJS etc. The source code is directly zipped and uploaded for scan

The changes may be in exporting the project and choosing the right preset. Steps to be followed are as shown:

1. Export the UI5 project from eclipse as .war, Unzip the .war file and Zip it again to get the file in .zip format that can be uploaded to Checkmarx server



1. The steps to create the project remains the same as in XSJS project but the choice of preset changes here in this example for UI5 we have to choose preset, “SAP\_Default\_SAPUI5\_2015” for scanning. The analysis of the result remains the same as discussed above. In UI5 checkmarx analyzes the javascript, HTML files in the project. An example of result summary of a scanned UI5 project is shown



A total of 19 issues are reported the detailed vulnerabilities can be seen by clicking open viewer

1. **Overview of XSJS vulnerabilities**

SQL injections : SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application.

Ex: In a web application if we have a username validation system where the user enters his name and a query in the background is run to see if this username exist by executing a query:

SELECT \* FROM USERS WHERE USERNAME=ABC

If the user is allowed to enter any value without validation he/she may enter:

<ABC or 1=1>

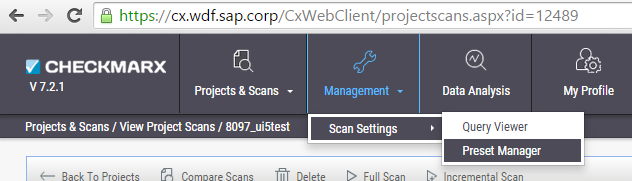
And the input query to the database would be:

SELECT \* FROM USERS WHERE USERNAME = ABC or 1=1

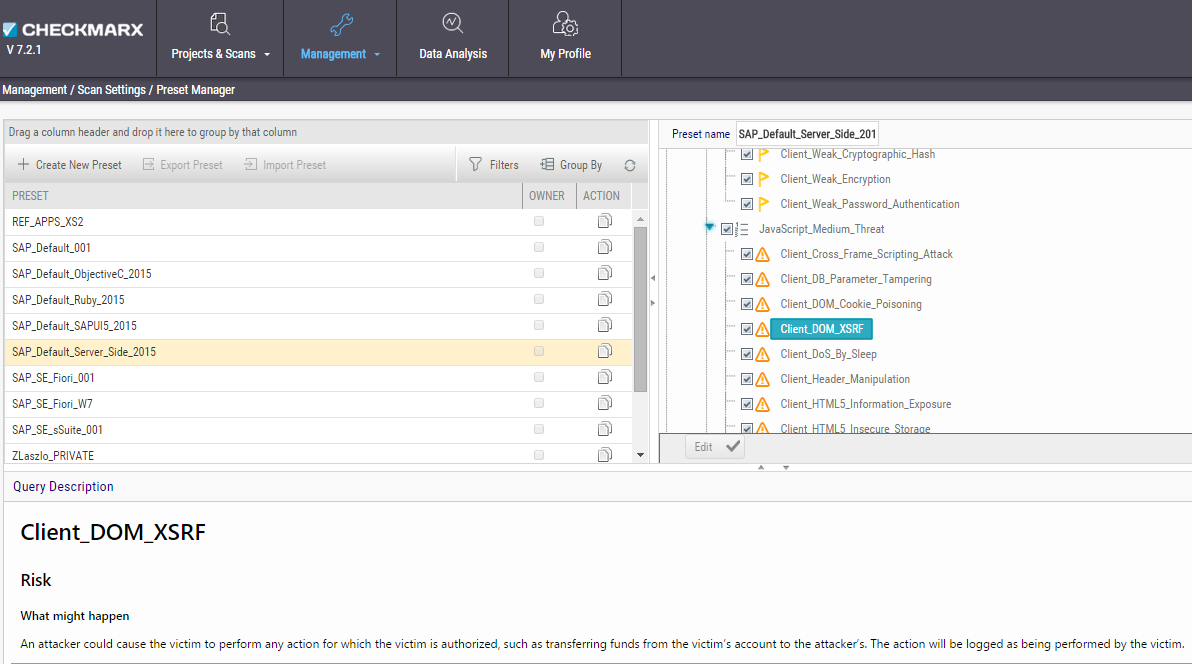
Which will be true for any input thus the sensitive data of the database will be exposed to the user. Apart from the above example there can be various other possible vulnerabilities in the code

Vulnerabilities that are scanned and detected by Checkmarx can be checked in the Checkmarx servers’, Preset Manager to got to Preset Manager follow

Management -> Scan Settings -> Preset Manager



In preset Manager All the presets are listed and scans that run for each preset are listed and the scan description can also be checked in the same window as shown in screenshot below:



1. **Overview of UI5 vulnerabilities**

As mentioned above the vulnerabilities reported by Checkmarx for UI5 can also be checked similarly for preset: “SAP\_Default\_SAPUI5\_2015”. Some examples of checks for UI5 presets are

Client\_Hardcoded\_URL, Client\_Manual\_XHR\_Handling, Client\_DOM\_Code\_Injection etc

1. **Integrating Checkmarx with Jenkins for Continuous Integration**

Checkmarx provides a plugin for Jenkins that allows to issue scans from within Jenkins as well as to access scan results/reports from Jenkins. The instructions for the installation can be found on the Knowledge Center: <https://checkmarx.atlassian.net/wiki/display/KC/Jenkins+Plugin+Overview>.

The plugin works on Linux or Windows systems.

**Default Server URL**: https://cx.wdf.sap.corp  
**Default Username**: GLOBAL\d(i)-user

For the Jenkins plugin user, we strongly recommend you to use a service user of the GLOBAL domain instead of using your personal account. Such service user can be requested using the self-service offered by SAP IT: <https://service-accounts.wdf.sap.corp/>. When you get the email from SAP IT with the user id, please communicate it to cx@sap.com so that an account on the Checkmarx server can be created. Don't forget to specify to which team the user should belong so that the user can access the right projects.

The project that you would like to scan doesn't need to exist on the server when you configure Jenkins. Checkmarx will create a local project on the server automatically. If it however already exists, Checkmarx will not use the source code defined in the project, it will scan the code sent to it from Jenkins. There is therefore no need to define your project on the server as a GIT project. GIT projects are scanned based on the Checkmarx scheduler, it has nothing to do with Jenkins.

You can find more information here on the Jenkins plugin:<https://jam4.sapjam.com/groups/7LepwhgKOBeEeW6x76ke1N/documents/MyzHIpRb47OS0BK5VwUAvH/slide_viewer>.

**Jenkins Best Practices**

Checkmarx recommends that you perform one full scan for every 5 incremental scans.

In the Jenkins plugin, this can be conveniently done in the following manner:  
i. When defining the Checkmarx scan step of a Jenkins build process, Check the "Incremental" check-box.  
ii. Another Check-box should be visible now -->"Schedule Periodic full scans", check this box as well.  
iii. Define 5 "Number of incremental scans between full scans".

Above document covers a brief overview of covered topics. Please check the official site of Checkmarx at: <https://www.checkmarx.com/>

Also please feel free to contact at : [Nagabhushan.anvekar@sap.com](mailto:Nagabhushan.anvekar@sap.com) for updates and queries