Overview

✅ Wide support of relevant programming languages, ease of integration and excellent scan engine technology are the reasons Checkmarx was chosen over the competition, HP Fortify.

✅ User friendly and easy to setup and automate. This helped achieve developer adoption.

✅ Checkmarx helps developers remediate security vulnerabilities early in the development lifecycles.

✅ Over 100 Java and .NET applications scanned, with over 30 Million lines of code.

✅ The integration with JIRA helps automate the vulnerability remediation process by immediately assigning remediations to the relevant developers.

✅ Checkmarx complies with industry regulations and standards.

✅ Top quality scanning results

> We really like that we get advice on how to remediate issues. Appsec coach is very useful for developers, they immediately know how to fix issues.

Requirements When Choosing a SAST Solution

Apart from the obvious need to comply with the industry's software security regulations and requirements, the bank realized that 70% of attacks aim at the application level. This made it clear that a SAST solution was needed with zero tolerance to code vulnerabilities. The number one criteria when evaluating SAST vendors was the quality of the scan engine technology and a broad support of programming languages. In 2014 the bank evaluated both HP Fortify and Checkmarx, the latter was chosen following a successful POC.

Checkmarx proved to be easy to set up, it integrated smoothly into the developers' IDE (eclipse) and supported the required development languages. Checkmarx is now used for automated, regular periodical scans of the bank's source code.

Challenges and Overcoming Them

One of the challenges in implementing a SAST solution is developer adoption. Getting developers onboard with secure code development, which requires certain adaptation, is crucial for a successful SAST implementation. The process of integrating SAST requires buy-in from security experts and developers. The bank chose to integrate Checkmarx one application at a time, along with the security teams, to ensure they were all involved in the process.

The bank was successful in securing developer secure-code awareness and its developers are fully aware of the importance of secure coding. Checkmarx's friendly user interface along with its training proved to be helpful in ensuring that developers not only welcome secure coding, they follow it and know how to remediate vulnerabilities as they code.

We can rely on the scans and send out automated scan reports to the relevant teams.

Using Checkmarx

The bank runs automated CxSAST security scans on more than 100 applications, which were all developed in-house and contain over 30 million lines of code. Checkmarx's Eclipse plug-in ensures a smoother work flow for the developers as they use it within their IDE. Checkmarx's integration with JIRA proves to be helpful as it makes it easy to assign remediations to certain developers.

The case study interview was conducted with an Application Security Engineer at a leading Swiss private bank.