"**Continuous integration** (**CI**) is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day." - Wikipedia

THE CHALLENGES OF CONTINUOUS INTEGRATION AND APPLICATION SECURITY TESTING

Continuous Integration allows companies to generate up to hundreds of builds a day and cope with multiple code changes in a short timeframe.

Continuous integration has become more popular following the increase of software development groups adopting the agile methodology.

Agile development provides opportunities to assess the direction throughout the development lifecycle. Using regular cadences of work, known as Sprints or iterations, at the end of which teams must present a potentially shippable product increment achieves the ability to assess the product throughout development. Rather than looking at a single planned result, agile allows teams to assess their results every short period of time, usually around two weeks, and if required steer the project in a new direction.

In the world of SaaS, agile and continuous Integration/Delivery has and is being adopted quite widely. Development teams are now able to deliver and modify their releases multiple times a day using the Cloud's flexibility and control while at the same time implementing the right techniques to develop quicker and better using agile and continuous integration/delivery.

Traditional application security methods are not able to support these quick cycles:

1. Most security testing techniques are dependent on project completion and usually take a long time to carry out which halt the development process and can cause delays to scheduled release dates.
2. Developers are accountable for their code and as such, also accountable for security vulnerabilities found in the code. Traditional security testing solutions are employed at the end of the development process, sometime months after development of the code is completed. In such instances, developers have to go back to review old code, re-familiarize themselves with it, which takes a long time to do. Traditional Application Security testing solutions tend to ignore or are incapable of handling the process of solving vulnerabilities once found. This is critical in continuous integration to prevent long "road blocks" and enhance developer education on the most effective mitigation locations and techniques.
3. Penetration tests will, in most cases, run as black box tests and require a separate team of testers or a third party vendor who will perform the tests at the end of the development cycle. Pen tests are expensive and slow. On top of that, they have to be performed on every new release to changes have not affected security. This is virtually impossible and financially overwhelming . when talking about CICD (Continuous Integration Continuous Deployment)

How does Checkmarx address Continuous Integration?

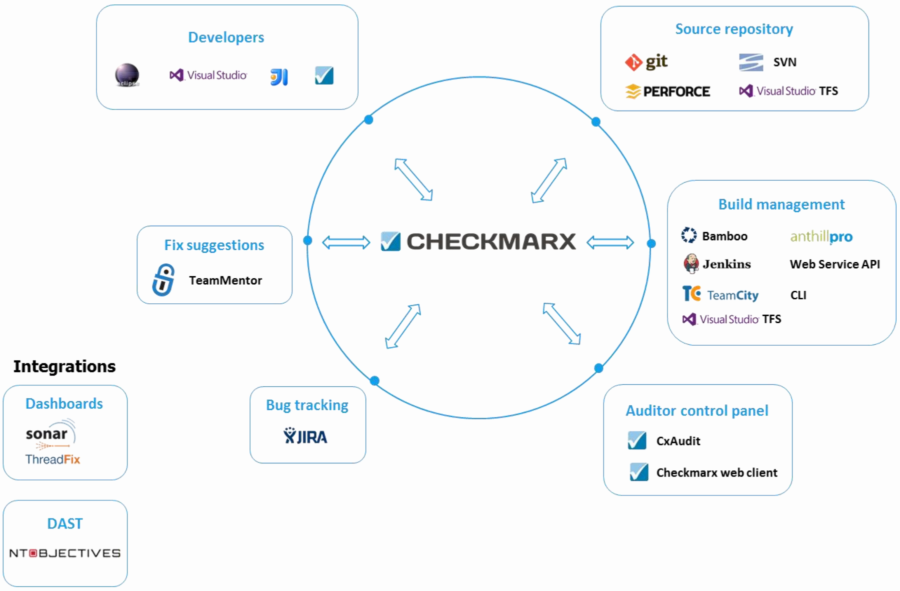
Checkmarx CxSAST is a highly accurate and flexible Application Security analysis product that allows organizations to automatically scan un-compiled/un-built code and identify hundreds of security vulnerabilities in the most prevalent coding languages. CxSAST is effectively integrated into the SDLC process to streamline the detection and remediation process.

Checkmarx's main mission is to make Application security an integral part of the SDLC (Software Development Life Cycle). Continuous integration is a significant part of agile software development lifecycle and increases developer's ability to detect and resolve security related bugs in short cycles, allowing a noteworthy improvement of "bug free" builds and releases.

Checkmarx not only detects the vulnerabilities on raw source code/non-buildable code, allowing either complete or partial code scanning at any given time, but goes a great length forward providing the developer with:
Graphical representation of the vulnerable data flows Specific optimized locations to fix the code (i.e. best fix locations), Recommendations how to fix the code in the most efficient way.

Similar to QA processes, Application security should and can be integrated into the development and testing environments. Therefore all this is done during the SDLC process and with minimal to no impact on the build release.

Enforce your security policy in the SDLC



Checkmarx's CxSAST fully supports Continuous Integration and Continuous Deployment for management of Security vulnerabilities.

1. **Out of the box integration** with the most popular "Continuous Integration servers" makes this task as simple and straight forward as it gets: Jenkins, Bamboo, TeamCity, Visual Studio TFS, Anthillpro.
2. **Web Services API** - CxSAST API provides the ability to create client scripts orchestrating CxSAST projects and easily integrating with Build Servers and Source repositories of any kind.
3. **CLI -** CxSAST scans can be invoked from the CxConsole Command Line Interface (CLI) command. This is especially useful for embedding calls in software management tools, allowing Continuous Integration in the Software Development Lifecycle

4 Simple Steps to ensure secure coding as part of your Build Management process

INTEGRATE Checkmarx is flexible enough to integrate seamlessly within your SDLC, we support the most common build servers or alternatively you can implement a simple API to provide high quality SAST (Static Application Security Testing) in a smooth and simple to operate environment.

ANALYZE & TRACK Checkmarx's CxSAST produces a high-level vulnerability report which is linked to a color coded HTML report that identifies the specific areas of code in which the vulnerabilities exist allowing easy and speedy fix. Bug tracking is made easy with CxSAST's integration. Automatically communicating with different bug tracking tools such as Jira to reflect the results (open bugs automatically) and define specific areas in the code or specific coders that are causing vulnerabilities.

SET & ENFORCE Enforce your security policy and ensure that flawed code doesn't move into production. Just mark the build as Unstable and send an alert or break the build if the security vulnerability instances in the build exceed the specified threshold set by your organization's security policy.

AUTOMATE Continuous integration servers are set to trigger a Checkmarx scan allowing complete automation of the process leading to:

- Detection of Security flaws as they appear, reducing time to fix thereby reducing development costs.
- Reducing developer efforts and frustration by remediating issues immediately rather than going back to long forgotten code
- Eliminating release delays
- Cutting down ongoing Pen-test costs and using those for verification tests only