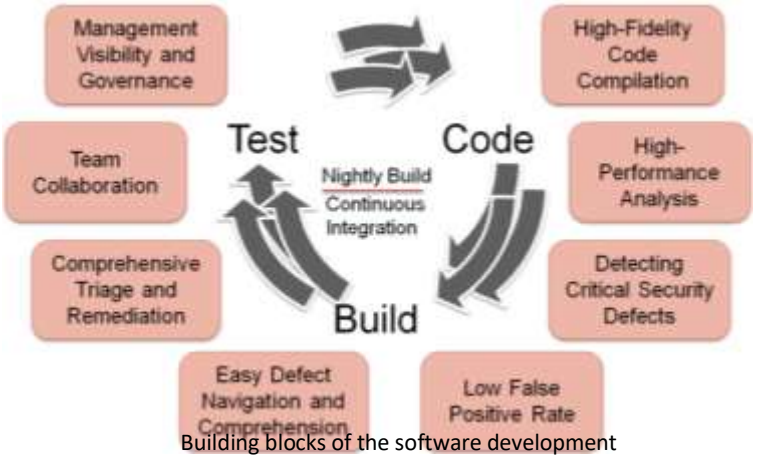
Introduction

Security & Quality Software GmbH offers services and tools for application protection with automatic, integrated testing of the security and quality of software during development with measurable results. OpenSource code and snippets are detected, legal information, versions and known exploits shown. When combining solutions to obfuscate the executables and protect against tampering in conjunction organizations can more securely build, analyze and release applications into production. The tools are integrated in your workflow, IDE's, bug tracking systems, dashboards and build environments. You save money, improve process efficiencies, shorten development cycles and reduce risk measurably.

Software In-house Development

Developers are supported to find critical security & quality flaws in the software early. Bugs are described, explained and the sources indicated. Security & Quality Metrics and Coding Guidelines are automatically applied. Results are:



Building blocks of the software development

- Improved Performance and Stability of the application and product
- Identification of security flaws and open doors for attackers
- Security & Quality of the software is measurable
- Information for a better planning of in-house and external software development
- Faster availability of the products/software with cost reductions
- Controlling and planning of delivery times (risk analysis)
- Increased customer satisfaction with more stable and secure products
- Better software maintainability and faster product changes
- Automatization of software development and testing leads to an increase of productivity

External developed Software

Possible third party software in your code

Using external developed software causes a residual risk. The same security & quality metrics and KPI's as for the in-house development described above have to be applied.

Situation

- Most of the IT departments lack the tools, specialists and money to test external developed code
- The traditional way to test and evaluate the code is expensive, In-complete and doesn't scale
- Supplier don't deliver or give access to their source code

Possible Solutions

1. You set up a Security & Quality Gate with all the metrics the software has to meet and give the supplier access to. This can be organized in a private cloud.
2. The supplier has tools in-house and tests the software according to your rules and metrics to generate a certificate as point of prove.
3. SQ Software tests the supplier Software according to your rules and delivers a certificate.

Code Obfuscation and Anti-Tampering against

- Reverse Engineering (Intellectual property theft, first step to modify the code and data, ...)
- Un-authorized modifications of the application integrity

Efficiency against reverse engineering

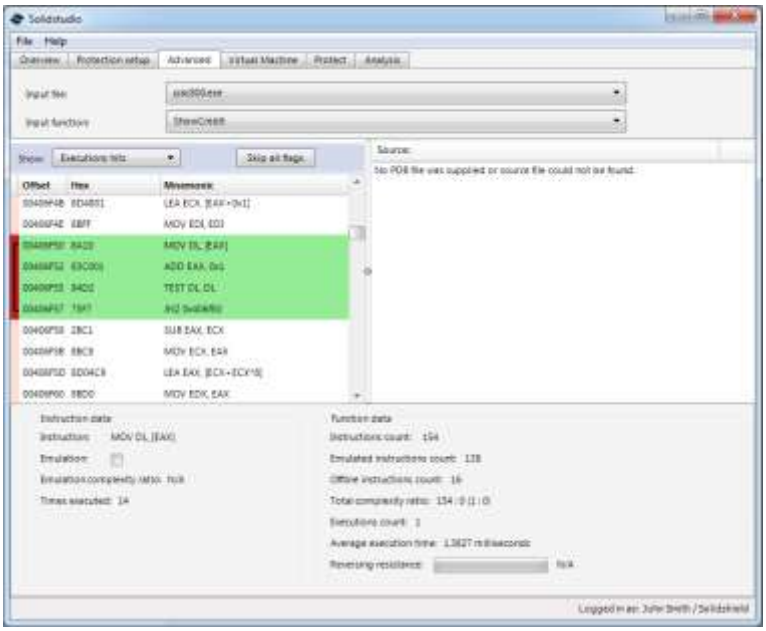
Code virtualization generates a complete de-structuring of the code by transforming the original executable code into data (specific syntax bytecode) and code (specific virtual machine). The original code is never present and visible in memory at any time.

Efficiency against tampering

At runtime, the line-by-line chained decoding processed by **SOLIDSHIELD** virtual machine depends on many code footprint cell reads. This process can't be easily stripped because there is no Boolean check to inverse. A full control for adjusting the density of these checks can be done interactively with **SOLIDSTUDIO.**

Performance Control

SOLIDSTUDIO has been designed to offer full visibility as well as adjustment on performance degradation. For that, Hotspots (meaning original code instructions which cause major performance degradation when virtualized) are automatically shown in the GUI. This enables you to remove them in one click. The impact on performance is immediately viewed (see screenshot right).



Red/black bar highlights the loop with instructions processed 14 times. They have been removed from the virtualization process (marked green).

JAVA Bytecode Virtualization

SOLIDSHIELD is the only available efficient JAVA code obfuscation tool in the market.

Remote code execution

CLOUDSHIFT offers the opportunity to execute functions remotely in a separate closed and secure processor (ARM based dongle). Extracting a function of your code and get it executed in the Cloud (or in a dongle) takes 5 seconds. Any standard machine can be used for remote execution (machine bound execution) as well. There is no change of the source code required.

Technical

- CPU support: x86 32/64bit, PowerPC, ARM (under development) and more
- Linux, Windows, VxWorks RTOS support: secures DLL's, Windows, Linux and VxWorks libraries, SO's
- Programming language independant: C, C++, ADA, Assembler, Fortran etc.
- Interpretative languages: **JAVA**, (C# and .Net under development)

Summary

Code Obfuscation WITH	Code Obfuscation WITHOUT
Controlled, (half-)automatic and user friendly virtualization and obfuscation of selected functions	Big performance degradation
Execution of functions remotely in a separate closed and secure processor (ARM based dongle)	Additional drivers
Hardware bound execution of (parts of) the software	Change of source code