Decilog

Constituents – Financial Services, Independent Software Vendors, Academic Institutions, U.S. Government Agencies
Software Assurance

CASE STUDY

Secure Software for Mission-Critical Systems

Note: A Decilog affiliate participated in the following project.

Developing and Implementing Secure Software

CHALLENGE

Attacks on the application layer of computer systems have become more common and effective. In order to achieve an acceptable level of assurance that applications can withstand attacks, a common approach to designing, developing, testing and implementing secure software is needed.

APPROACH

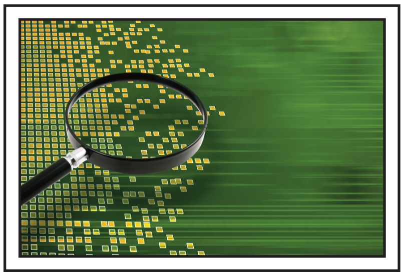
The first task was to review the many methods used for engineering secure software and then selecting the most effective approaches. It is generally held that it is much cheaper and much more effective to "build security in" than to bolt it on when the software is already being used in a production mode. The Security Development Lifecycle (SDL) from Microsoft was evaluated as well as best practices as described in the Building Security In Maturity Model (BSIMM) supported by the U.S. Department of Homeland Security.

It was determined that one must not only consider building in security during the development lifecycle but also ensure that secure methods are used during the deployment, production, and decommissioning phases.

It became apparent during the project that many institutions and vendors do not follow preferred procedures in the development, acquisition and/or deployment of software and that the result is high exposure to attacks and the insider threat.

BENEFITS

It proved very helpful to participants in the project to be given guidance as to how to approach the software assurance issue. Additionally, a number of areas, such as security architecture, context, infrastructure, resiliency, and relevant data collection instrumentation were shown to be key success factors required for achieving the goal of implementing more secure software.



RESULTS

The results of the initial scoping project included a list of preferred ways in which to develop, test, and implement software in order to ensure strong security traits. It was determined that the standard approaches available today, if followed carefully, lead to a higher degree of security, but that there are gaps in the approaches that need to be filled. In addition, it was recognized that there is a need for shared software testing and certification facilities in order to make the requisite testing of commonly-used software more cost-effective.