Fraunhofer FOKUS

FRAUNHOFER INSTITUTE FOR OPEN COMMUNICATION SYSTEMS FOKUS

STANCE

Contakt
Dr.-Ing. Jens Gerlach
Head of Verification
System Quality Center – SQC
Phone +49 30 3463-7458
jens.gerlach@fokus.fraunhofer.de

Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin
Germany

www.fokus.fraunhofer.de/go/STANCE_en

When it comes to control software for mission-critical systems – for instance in power grids or public infrastructure – security requirements are high. Systems such as these have to function reliably and securely under all circumstances, such that no risks arise. In German, the term "Sicherheit" has two meanings here, which translate as "safety" and "security" in English. Operational safety deals with the issue as to whether unacceptable risks can arise for humans or the environment as a result of the system operating. When it comes to attack security, the system should be protected from external attacks – hackers, for instance. An example would be a hacker modifying the software of a train control system, thereby bringing rail traffic to a standstill. The STANCE (Source Code Analysis Toolbox for Software Security Assurance) project primarily deals with the problem of software attack security.

Trust in attack security

In our environment, the security of information and communication technologies must be guaranteed, such that individuals can place their trust in them. Indeed, the claim is that technology shouldn't just make things possible, but also protect its users! For this reason, the STANCE project was tasked with researching and developing technologies that guarantee the security of software against attacks. Ten European partners from science and industry want to provide a toolbox for source code analyses, which can be used to verify that software systems are immune to certain categories of attacks. With these technologies, software developers can prove that programs and their functionalities always correspond to their behavior, which was specified beforehand. If this is possible, users develop a strong and well-justified degree of trust in the systems developed.

IN THE STANCE PROJECT, ANALYSIS TOOLS ARE BEING DEVELOPED TO VERIFY THE ATTACK SECURITY OF COMPLEX SOFTWARE SYSTEMS

Static analysis of weak points in programs

In the project, a series of program analysis tools capable of verifying the attack security of complex software systems are to be defined, implemented and validated. Program analysis refers to various – to a certain extent – formal methods that can be used to uncover unplanned software system behavior in a semi-automatic fashion. Until now, formal methods have only been rarely used with regard to program analysis in the area of attack security. The reason for this, among other things, is that the popular programming languages – including Java and C++ – are extremely complex, which makes the use of formal methods more difficult. Tools and methods that can be used to extensively analyze the security properties of critical program components are thus being developed within the project. The program analysis tool Frama-C from CEA-LIST and the verification tool VeriFast from KU Leuven, which until now have mainly been used for C, are to be expanded within this context such that Java and C++ software can be examined, too. Existing solutions (formal methods, modern static and dynamic program analysis tools, existing expertise in security evaluation and relevant, industry-specific knowledge) are both being exploited and significantly expanded within the STANCE project.

Static and dynamic analysis

The project wants to build a bridge between so-called static and dynamic program analysis tools. Dynamic analysis refers to the analysis of programs that takes place whilst the program is running – in other words, it's being tested. During this process, the compiled software code with specified test values is run. In this way, it should be possible to uncover unexpected program behavior. On the other hand, there are static analysis methods which inspect software without running it. In so doing, the source code or, also, the compiled code is examined, such that weak points that could provide a gateway for attacks can be identified at an early stage. The project focuses on static analysis, but dynamic methods are also taken into account. First of all, typical software weak points are formalized (i.e. at a higher abstraction level described with mathematical strictness), such that these can be found by the analysis tool during the verification process as part of a tool-supported procedure. In order that the completed software is as well prepared as possible against potential attackers and no longer features any attack areas or weak points as far as is possible, both dynamic and static analyses are combined in the STANCE project. This combination is meaningful, as the use of static analysis methods ensures greater efficiency and thus lower costs for software quality assurance.

making cities smart

bad | 1402 (Photos: Matthias Heyde; iStockphoto.com)