Software Engineering Research Center (SERC)

Ball State University, Wayne Zage, Director, 765.285.8664, wmzage@bsu.edu
Ball State University, Kirsten Smith, 765.285.1889, kirstensmith@bsu.edu
Purdue University, Aditya Mathur, 765.494.7823, apm@cs.purdue.edu
West Virginia University, Don McLaughlin, 304.293.0405x2514, don.mclaughlin@mail.wvu.edu

Center website: http://www.serc.net

Provably Secure Software Systems Development through Security Typed Languages

This work has extended security-typed languages from the realm of theoretical programming language tools to an apparatus for building secure software systems. Researchers at the Software Engineering Research Center have developed software engineering practices and tools that allow mapping of high-level security goals into the applications that must realize them in a secure-typed language. These tools have been used to develop provably secure large-scale security-labeled email systems, as well as other secure systems. Past software security techniques have attempted to isolate applications (as seen in sandboxing techniques), introspect possible vulnerabilities (as in static and dynamic analysis tools), or apply formal analysis to achieve high assurance applications. However, such approaches are often ad hoc, incomplete or difficult to use.

This work has addressed these past limitations by identifying, formalizing, and automating the complex processes of identifying security relevant data and its potential for leakage or corruption. In so doing, usable infrastructures are available for defining and developing provably secure software. This work advances the state of the art in software systems development for homeland security by providing guaranteed compliance with security goals. Further efforts have identified new algorithms for important problems such as security level inference and credentials discovery. Such discoveries are being used not only in security typed languages, but are also helping the operating systems community to define policies and services tailored to the security requirements of applications. This will significantly enhance the capabilities of commercial software developers to articulate and realize security goals. The researchers are working with Motorola to evaluate how the tools can be used to make applications of embedded devices such as cellular phones more secure. For more information, contact Dr. Patrick McDaniel at The Pennsylvania State University, 814.863.3599, mcdaniel@cse.psu.edu.

Design Metrics Technology

Improvements in the software development process depend on our ability to collect and analyze data drawn from the various phases of the development life cycle. The Software Engineering Research Center (SERC) Design Metrics Team has developed a metrics-guided methodology for software reliability that begins with architectural design. This technology provides a framework that is unbiased, efficient, and cost-effective to determine design improvements, code-modification, and testing and management strategies. Applying this methodology to software designs highlights the stress points and determines overall design quality. Stress points are defined as critical components in software where errors in coding and programming logic are likely to occur. Identifying such components in advance and applying mitigating approaches results in effective resource allocation. In the coding phase, the technology can identify those stressful components and provide change impact analysis. In testing, the metrics can assist in determining where testing effort should be focused and the types of test strategies to employ. In the thirteen years of metrics validation, on a wide variety of projects ranging from missile defense, satellite, accounting, and telecommunications systems to interactive games, the design metrics have identified at least 75 percent of the error-prone components 100 percent of the time with very few false positives. Applying the design metrics technology can assist developers in engineering quality into the software product. For more information, contact Dr. Wayne M. Zage, 765.285.8664, wmzage@bsu.edu.

Spotlighting the Code

Professor Norman Wilde has developed a technology called the Reconnaissance Tool to find specific sections of computer code that carry out specific functions. During the Y2K transition, the tools quickly identified sections of computer code spotlighting with date information that were susceptible to date problems. The software is widely available. For more information, contact N. Wilde, University of West Florida, 850.474.2548, nwilde@uwf.edu.

Smart Manuals

Large organizations such as the U.S. government and industrial operations often have many thousands of pages of manuals. These present considerable challenges not only in terms of locating information, navigating large document systems, eliminating overlaps/redundancies, and upgrading information, but also in providing guidance for users in carrying out work based on the manual information. Professor Hany Ammar of West Virginia University has worked with a government contractor called ManTech Inc. to develop a prototype design of a smart interactive electronic technical manual that provides many advantages over previous approaches. The prototype not only enhances the interactions of users with these large manuals, but can also receive input from users to help in diagnosing problems and can guide them through the repair process. The method uses Bayesian nets to determine the most likely problems and best courses of action. The method is more flexible, more interactive, accepts more user inputs, and provides more guidance than previous technologies. For more information, contact Hany Ammar, West Virginia University, 304.293.0405 ext. 2514.

Safety in SmartHomes

With the increasing availability of low cost wireless devices such as cellular phones, medical devices, and home networking devices, it is now possible to remotely and programmatically control various devices and events at home, and outside. However, while offering increased flexibility and convenience to humans, these devices have also raised the possibility of serious malfunction due to proximity. For example, aircraft navigation systems and medical implants are affected adversely by cellular devices and other devices that emit large amounts of radiation such as an NMR. Despite stringent emission standards, proximity of two or more devices, often of different kinds, can raise serious threats to human life. The SmartHome project has investigated, among others, issues of safety. A key contribution of this work is universal software/hardware architecture for devices that radiate. Such an architecture, obtained with the help of Digital Device Manuals, allows controllers embedded in various life threatening environments such as hospitals and aircraft, to ensure safe operation of mobile and other radiating devices. For more information, contact Aditya Mathur, 317.494.7823, apm@cs.purdue.edu.

Provably Secure Software Systems Development through Security Typed Languages

This work has extended security-typed languages from the realm of theoretical programming language tools to an apparatus for building secure software systems. Software engineering practices and tools have been developed that allow the mapping of high level security goals into the applications that must realize them in a secure-typed language. These tools have been used to develop provably secure large scale systems, e.g., a security-labeled email system for homeland security and other challenges.

Past software security techniques have attempted to isolate applications (as seen in sandboxing techniques), introspect possible vulnerabilities (as in static and dynamic analysis tools), or apply formal analysis to achieve high assurance applications. However, such approaches are often ad hoc or incomplete. More recently, security typed language techniques simply disallow applications to be written, but have been historically intractable to use. This work has addressed these past limitations by identifying, formalizing, and automating the complex processes of identifying security relevant data and its potential for leakage or corruption as needed to build security-typed applications. In so doing, a usable infrastructure for defining and developing provably secure software is provided. For more information, contact Patrick McDaniel, Pennsylvania State University, 814.863.3599.