CodeHawk: Sound Static Analysis for Proving the Absence of Memory Related Software Vulnerabilities

Submitted by drsmith17 on Thu, 04/23/2015 - 6:36pm. Contributor: Douglas Smith

Abstract:
Most software vulnerabilities are due to coding errors. Testing is commonly the main means for detecting vulnerabilities, but testing alone only explores a small fraction of the possible behaviors of software. Sound static analysis is a technology that can examine source code and reason about all of its behaviors in order to detect coding errors that lead to vulnerabilities.

Kestrel Technology (KT) is developing a breakthrough static analysis tool, called CodeHawk, with an initial focus on memory related errors in C source code (covering 50 memory CWEs). CodeHawk is an advanced implementation of Abstract Interpretation, which works by automatically generating invariants for all code locations in a program and then using the invariants to prove safety and security conditions. CodeHawk also accepts human guidance in the form of strengthened invariants -- the effect is to increase the percentage of memory accesses that are proved safe.

Currently CodeHawk exhibits near-perfect precision on standard benchmark codes produced by NIST and vulnerability experts. That is, when code contains a memory related error, CodeHawk will almost certainly report the error and will report few "false positives" (reports of code locations that are not errors). CodeHawk automatically provides mathematical proofs as evidence of safety and security claims for constructing an assurance case.

CodeHawk has been applied to a wide range of open-source C applications as part of a recent DHS-sponsored research project, and when run without application-specific preparation, CodeHawk can automatically prove 80%+ of memory accesses safe. For example, CodeHawk can currently prove 82% of memory accesses safe in the OpenSSL code that contains the HeartBleed vulnerability. KT researchers are working toward increasing the percentage of proved safe accesses to 100%. KT now believes it is practical to prove complete memory safety in a critical infrastructure program like OpenSSL by using CodeHawk with human guidance. Analysis time for proving automatically 82% of OpenSSL memory accesses safe was 9 hours using a single core PC but could be reduced to minutes using multiple cores.

The pressures of cyber-crime and cyber-espionage will lead to increased demands from customers for credible evidence of the absence of exploitable vulnerabilities and assurance that Heartbleed type attacks cannot be successful. CodeHawk is a front-runner in a new generation of automated tools that support cybersecurity analysis and the construction of assurance cases with mathematical proofs as evidence.

Biography:
Dr. Douglas R. Smith is Principal Scientist at Kestrel Institute, and President of Kestrel Technology LLC. He is a Fellow of the American Association of Artificial Intelligence (AAAI) and an ASE Fellow (Automated Software Engineering). He taught an advanced graduate course on knowledge-based software development at Stanford University during 1986-2000. Dr. Smith was Chairman of IFIP Working Group 2.1 on Algorithmic Languages and Calculi from 1994-2000.

His main research interest has been mechanizing the development of software from formal specifications. He is mainly responsible for the development of KIDS (Kestrel Interactive Development System), which has been used to synthesize many algorithms, including a variety of complex high-performance scheduling applications for the US Air Force. Theoretical developments arising from experience with KIDS were a key motivation for building the Specware, Designware, and Planware systems. His current research focuses on automating system design, including the automatic
enforcement of cross-cutting safety and security policies, and system code generation from formalized design patterns and software architectures.

Dr. Smith has over 30 years experience in the field of automated program synthesis and has published over 75 papers (see the CiteSeer surveys of most cited papers in Software Engineering and most cited authors in Computer Science). He has one patent. He received the Ph.D. in Computer Science from Duke University in 1979.

Other available formats:

**CodeHawk: Sound Static Analysis for Proving the Absence of Memory Related Software Vulnerabilities**

Switch to normal viewerSwitch to experimental viewer

Tests Validation and Verification Foundations assurance CodeHawk proofs tools National HCSS Conference 2015 Research Organization Abstract Presentation HCSST5 Proof Engineering (HCSST5)