Reducing Software Security Risk through an Integrated Approach

David P. Gilliam
Caltech, Jet Propulsion Laboratory
david.p.Gilliam@jpl.nasa.gov

John D. Powell
Caltech, Jet Propulsion Laboratory
John.Powell@jpl.nasa.gov

John C. Kelly
Caltech, Jet Propulsion Laboratory
john.c.kellyw@jpl.nasa.gov

Matt Bishop
University of California at Davis
bishop@cs.ucdavis.edu

Abstract

This abstract presents joint work by the California Institute of Technology's Jet Propulsion Laboratory and the University of California at Davis (UC Davis) sponsored by the National Aeronautics and Space Administration Independent Verification and Validation Center to develop a security assessment instrument for the software development and maintenance life cycle.

Vulnerabilities in operating systems and software applications render an otherwise secure environment insecure. Any operating system or application added to a secure environment that has an exploitable security vulnerability affects the security of the whole environment. An otherwise secure system can be compromised easily if the system or application software on it, or on a linked system, has vulnerabilities. Therefore, it is critical that software on networked computer systems be free from security vulnerabilities.

Security vulnerabilities in software arise from a number of programming factors; but these vulnerabilities can generally be traced to poor software development practices, new modes of attacks, mis-configurations, and unsecured links between systems.

Currently, there is a lack of Security Assessment Tools (SAT) for use in the software development and maintenance life cycle to mitigate these vulnerabilities. The National Aeronautics and Space Administration (NASA) Independent Verification and Validation (IV&V) Center, has funded the Jet Propulsion Lab in conjunction with the University of California at Davis (UC Davis) to develop a software security assessment for use in the software development and maintenance life cycle.

The goal of the effort is the use of a formal analytical approach for integrating security into existing and emerging techniques for developing high quality software and computer systems. The approach is to develop a security assessment instrument consisting of a collection of tools, procedures and instruments to support the generation of secure software. Specifically, the instrument offers a formal approach for engineering network security into software systems and application throughout the software development and maintenance life cycles.

The security assessment instrument has three primary foci: a Vulnerability Matrix (VM), a collection of Security Assessment Tools (SAT) which includes the development of a Property-Based Testing (PBT) tool, and a Model-Based Verification (MBV) instrument.

The VM is a database maintained by UC Davis as part of the Database of Vulnerabilities, Exploits, and Signatures (DOVES) project. It contains a list of vulnerabilities, the associated platform/application, and the exploit signature fields.

The VM provides a searchable knowledge base from which properties may be extrapolated for use with PBT and MBV. This knowledge base can also accommodate the discovery of new attacks not yet seen on the internet, but which may be discovered through MBV techniques.

The SAT is a collection of tools and programs that can be used to check the security of both compiled and pre-compiled software code. Each of the SAT's includes a description of the tool and it use, its pros and cons, related tools, and where the particular tool can be obtained.

As part of the SAT, UC Davis is developing from a prototype a PBT tool. This PBT will slice software code looking for specific vulnerability properties. Property based testing is a tool that verifies properties against the code level implementation of a system. These properties are extracted from the VM. PBT is equipped with its own libraries that contain readily testable properties. Finally,

used with the MBV, the PBT can provide verification of a model's fidelity to the system in the MBV.

The MBV component of the research is an operational proof of concept to perform verification of software designs for compliance to security properties. The concepts is based on an innovative model checking approach that will facilitate the development and verification of software security models as composable components

Model based verification uses precise abstractions. It offers the ability to verify security properties early in the life cycle – before an implementation exists. MBV can be used to identify and augment the VM with security anomalies that have not been discovered as a result of a known network security attack. Anomalies that are found in early lifecycle phases through out the examination of abstractions (models) can be preserved and later passed on to the PBT for verification at the code level.

The four parts of this research form a coherent technique for examining new and existing systems and software code for security flaws. When used in conjunction with each other, each part leverages cumulative benefits to classify and focus upon security properties that will be modeled and tested. The VM and model-based checking provide the properties that the software must meet; the property-based tester checks that the programs do indeed meet these properties. The VM forms the beginning of a library of TASPEC properties. Property-based testing requires properties expressed in TASPEC to test against. The additional benefits of leveraging the four parts of the instrument in combination include:

- *Reduced rework to identify security properties*
- *Increased confidence in the system through verification at multiple phases of the development and maintenance lifecycle*
- *One tool is capable of verifying the input and output of another tool in the instrument*
- *The potential to discover additional attacks that have not yet been seen in operational environments and test for their viability and severity*

Information gained through the use of these techniques will form the basis of training in the writing of more secure programs flows. Placing the benefits of this work in the context of a particular language and environment is an important part of improving the quality of software and systems.

Keywords

Security Toolset, Vulnerability Matrix, Property-Based Testing, Model Checking, Security Verification

References

[1] D. Gilliam, J. Kelly, M. Bishop, "Reducing Software Security Risk Through an Integrated Approach," Proc. of the Ninth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (June, 2000), Gaithersburg, MD, pp.141-146.

[2] Published and maintained by Mitre. The CVE listing can be found at: http://cve.mitre.org/

[3] G. Fink, M. Bishop, "Property Based Testing: A New Approach to Testing for Assurance," *ACM SIGSOFT Software Engineering Notes* **22**(4) (July 1997).

[4] M. Bishop, "Vulnerabilities Analysis," *Proceedings of the Recent Advances in Intrusion Detection* (Sep. 1999).

[5] J. Dodson, "Specification and Classification of Generic Security Flaws for the Tester's Assistant Library," M.S. Thesis, Department of Computer Science, University of California at Davis, Davis CA (June 1996).

[6] J. R. Callahan, S. M. Easterbrook and T. L. Montgomery, "Generating Test Oracles via Model Checking," NASA/WVU Software Research Lab, Fairmont, WV, Technical Report # NASA-IVV-98-015, 1998.

[7] P. E. Ammann, P. E. Black and W. Majurski. "Using Model Checking to Generate Test Specifications," 2nd International Conference on Formal Engineering Methods (1998) pp. 46-54.

[8]G. Lowe. Breaking and Fixing the Needham-Schroeder Public Key Protocol Using CSP and FDR. In TACAS96, 1996.

[9] W. Wen and F Mizoguchi. Model checking Security Protocols: A Case Study Using SPIN, IMC Technical Report, November, 1998.