

Model Checking for Software Security Properties

**John D. Powell
and
David P. Gilliam**

***Caltech, Jet Propulsion Laboratory*
4800 Oak Grove Drive
Pasadena CA 91109-8099**

ABSTRACT

Concurrent software systems and software applications are frequently subject to software security vulnerabilities that may render an otherwise secure networked software environment unsafe. The addition of software systems/applications to a secure environment can affect the security and safety of the whole environment either by exploitable security vulnerabilities in the additional software or between two sets of individually secure software in a networked computing environment that, through their interaction, may cause an unwanted security exposure or vulnerability. Therefore, any system with such vulnerabilities can be compromised when the software on it contains insecurities or unexpectedly interacts in insecure ways. Further, any connected systems are also put at risk along with their resources, services, and data. If an intrusion goes undetected, due to the exposure or vulnerability, networked systems are subject to the domino effect, where access to one system will eventually yield access to others through the use of hacker tools. Given the conditions discussed above and the potentially catastrophic nature of intrusions in to systems now and in the future, it is crucial that such vulnerabilities and unwanted exposures be identified and mitigated. Vulnerabilities in software arise from a number of factors but often can generally be traced to poor software development practices, new modes of network attacks, mis-configurations, and insecure interaction between systems.

Formal software assessment methodologies, such as model checking, can provide greater assurance that software executing on critical systems and systems linked to them do not expose critical data and functional vulnerabilities resulting from inadequately specified software requirements and designs or exposures due to complex integration with other software.

This paper will present the formal modeling portion of the "Reducing Software Security Risk" research project. A new model checking approach called the Flexible Modeling Framework (FMF) is part of a software security assessment instrument to assist developers in the verification of security properties in software during the early phases of the development and maintenance lifecycles.

Modeling requirements and early lifecycle designs to discover software security vulnerabilities precipitated by the interaction of software components under development in a new system or proposed as additions to an existing system or environment provides early insight into potential software security problems. This early detection assists software development efforts to address and correct vulnerabilities at significantly less cost in terms of time and effort than allowing them to persist into later lifecycle phases. Vulnerabilities that do survive to later lifecycle phases are often addressed with cumbersome "patches" that can introduce new security problems or yet unknown exposures or vulnerabilities of their own. Information about the overall research effort regarding network security is available at: <http://security.jpl.nasa.gov/rssr>.

The research described in this paper is being carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.