Abstract

Formalizing security requirements has received a significant attention since the 70s. However, a general method for specifying security requirements is still missing. Especially, little work has been presented on specifying and verifying that a given application is a secure resource consumer. The purpose of this work is to set up a methodology for (1) specifying security requirements of service providers and (2) proving that some application securely uses some resources. The developed theory will be evaluated and applied in two different areas: secure mobile code development and secure COTS-based software development.

1 Motivation

"Security like beauty is the eye of the beholder". Ranging from household computers to those powering nuclear power plants, security is of critical importance. But, how can an application be proved secure? Two challenges are presented for requirements engineering for security: eliciting what are the requirements for a particular system, and figuring out how to specify them in a way that is both perspicuous (to the problem owner) and useful (to the developer). In this sense, the problem of building dependable systems can be viewed as that of requirements engineering.

2 Related Works

Formalizing security requirements is by no means a new topic in software engineering. The term security has received diverse meanings, guided by diverse models and properties e.g. integrity, noninterference, nonductibility, trusted users, causal security, etc. Security specification followed the same route. However, despite the huge amount of work in this area, the problem remains complex. It is argued that multilevel security and the related notion of information flow are passé today. Another field that actually receives a particular attention is that of mobile code security. Among works that have been presented, the only we are aware of, that advocates constructing proofs for mobile code is Proof-Carrying Code (PCC); but PCC provides no methodology for this.

3 Proposed Solution

In this work, we are interested in specifying and proving that an application is a secure resource consumer (SRC). This has not been done elsewhere. Given an application $A$ running in an environment $E$, we are interested in proving that $A$ securely uses resources made available by $E$. The implication of our work is the provision of a method for building secure COTS-based software. The second domain of application of the methodology is for constructing secure mobile code. In this case, the application is straightforward since every mobile code is intrinsically a resource consumer. Diverse paradigms will be introduced in the context of our work: retrospection, action flow diagram, offensive (resp. defensive) security specification, and retrospective weakest precondition calculus.

4 Conclusion and Expectation

We outlined a novel approach in security specification and verification. Instead of proving that a security-sensitive system satisfies some security properties, we envision proving that an application securely consumes the resources made available by the runtime environment. Though we will present a rigorous approach, we also expect a relaxation of the method into security testing. The derived testing process will likely replace actual techniques that have methodological and fundamental shortcomings.