Nathan Dykstra
SE 4930
Article Summary 4
January 7, 2012

Article Choice

I chose to read the article *Engineering Secure Systems* by Cynthia E. Irvine and J.R. Rao. It is the guest editor's introduction from the January/February 2011 volume of the IEEE Security and Privacy magazine.

What Was Discussed

This article is basically an introduction to the articles following it and touches on several aspects of secure software design. It begins by stating many of the ways in which our modern society relies on secure software and platforms. The authors point out that most end users depend on their systems to do what 'they are supposed to do". But the question what we mean by "supposed to do". The authors don't directly answer this question, but seem to leave it for the reader to ponder.

They go on to describe how a "typical developer might think that a system is acceptable if it provides the customer's requested functionality; a wise developer might also ensure that the system isn't a danger to the user's health or safety" but neither of these approaches address security at all. Both of these methodologies don't account for entryways unintentionally left open in the development of a system. Most of what the authors describe is unintended functionality or unanticipated operation of a system through these means.

They also describe how the changing environments affect security. Most software is designed with a specific set of platforms in mind and may be considered very secure on this set. However, as the computer industry continues to grow, the range and variations of platforms increases and these systems were not designed with this in mind. As such, the assurances of security are weakened.

The article mentions some of the history of computer security as well. It started as a way to address the previously mention problems of unspecified functionality. A great deal of this work was done by the military, as computers were typically only operated by large institutions and the government. The authors state the good security is designed in from the beginning of development, when intelligent choices have the largest impact on the system.

Much of the remainder of the article talks about the three articles it is introducing. These range in topics from smart cards and public-key infrastructure, to high assurance distributed systems. It doesn't go into much depth on these subjects though, as they are described later in

more detail.

What I Learned

I learned a bit about how the security aspect of software development started with the military, which is not surprising. They focused on ensuring systems did not contain and were not able to perform unspecified functionality. Another historical point I found interesting was that updates were delivered via couriers whereas nowadays they are delivered by a network connection. This makes sense when one thinks about how old computer operated and the lack of connectivity.

The authors also mention something called Evaluation Assurance Level or EAL. This is a rating system used internationally to provide confidence levels that a system can resist cyber attacks. EAL7 is the highest rating and entails a program is capable of resisting the most sophisticated of these attacks.

The article also makes mention of RAS practices. These are reliability, availability, and serviceability. These are non-functional requirements that software systems must satisfy in addition to security. Many properties are being coalesced under the auspice of resiliency, that is failing gracefully.

How It Will Help

The RAS principles are good to keep in mind whilst developing software, and we've talked about this 'resiliency' concept in at least one class. It makes sense for a program to fail gracefully, whether through diminishing functionality or through some other means. Also, the authors talk a bit about user acceptability and how good security should be invisible. This seems to make it difficult for users to discern that a system is secure if it is invisible, but making these features to intrusive turns users away. There is a fine balance to be had in this conflict.

Questions

Have you heard of EAL? What kind of testing do systems have to go through for this certification and who issues the criteria and certificates?