Moore, Jim

From: Ben Brosgol [brosgol@adacore.com]
Sent: Thursday, July 12, 2007 1:51 AM
To: Moore, Jim
Cc: Ben Brosgol
Subject: [owgv] Definition of "vulnerability"

Jim,

Since I'm having problems mailing to the list, I'll just send these comments to you. Use your judgment on whether you want to forward to the list.

It looks like the definition of "vulnerability" is still under discussion. The FAQ on the web site says:

<<[Tentative - subject to discussion.]

"A flaw in a product that makes it infeasible, even when using the product properly, to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, or assuming ungranted trust."
-- From Microsoft, "Definition of a Security Vulnerability">>

This is not really appropriate in our context. We need to deal with vulnerabilities in programming languages, and a programming language cannot be considered a "product". By performing some surgery on the above definition we can come up with:

"A flaw that makes it infeasible, even when using the language properly, to prevent an attacker from usurping privileges on the system on which a program written in the language is executed, regulating its operation, compromising data on it, or assuming ungranted trust."

But this is rather unwieldy, and "infeasible" is too strong a condition.

John Benito, in N0079, nicely distinguishes between Language Vulnerability and Application Vulnerability and offers this definition:

<<3.1 Language Vulnerability

A feature or combination of features of a programming language which can cause, or is strongly correlated with, a weakness, a hazard, or a bug.>>

This is better, but:
* I would argue that vulnerabilities in languages may arise not only from the presence of specific features, but also by their absence.
* It is simpler to characterize the undesired effect as an Application Vulnerability since that term is defined. No need to itemize as weakness, hazard, bug.
* The verb "cause" seems too strong; features don't cause bugs (dare I bring up the analogy "Guns don't commit crimes, people do")

Thus my proposed revision:

<<Language Vulnerability

A property (of a programming language) that can lead to, or that is strongly correlated with, application vulnerabilities in programs written in that language.>>

The term "property" can mean the absence of a specific feature. For

example, encapsulation (control of where names may be referenced from) is generally considered a good thing since it narrows the interface between modules and can help prevent data corruption. The absence of encapsulation from a programming language can thus be regarded as a vulnerability.

Note that a property together with its complement may both be considered language vulnerabilities. For example, automatic storage reclamation (garbage collection) is a vulnerability since it can interfere with time predictability and result in a safety hazard. On the other hand, the absence of automatic storage reclamation is also a vulnerability since programmers can mistakenly free storage prematurely, resulting in dangling references.