CERT'S PODCASTS: SECURITY FOR BUSINESS LEADERS: SHOW NOTES

Is There Value in Identifying Software Security “Never Events?”

Key Message: Now may be the time to examine our responsibilities when developing software with known, preventable errors – along with some possible consequences.

Executive Summary

Software and IT are not such young fields anymore. Many of the software errors that result in serious security breaches have been around for a long time. It should no longer be acceptable to deliver software that may cause harm. It may now be time to impose some level of discipline onto this field.

In this podcast, Bob Charette, founder of ITABHI Corporation and an internationally recognized expert in risk management, information systems and technology, and systems engineering, discusses the merits of developing a list of software security “never events.” As used in the healthcare community, this practice describes serious, preventable medical errors that should never occur.

PART 1: ERRORS IN MEDICAL CARE AS AN ANALOGY FOR ERRORS IN SOFTWARE

Medical Never Events and Their Consequences

The healthcare community uses the term “never event” to describe life threatening, serious, and entirely preventable medical errors.

It was first used in the mid-1990s by the National Quality Forum as part of their mission to implement a national strategy for healthcare quality measurement and reporting.

There is now a list of twenty-eight or so medical “never events.” These errors include:

- surgery on the wrong part of the body
- removing a kidney instead of a gall bladder
- leaving a sponge or scalpel in the patient after surgery
- receiving the wrong dose of medicine during hospital care

A medical “never event” is clearly identifiable, measurable, and preventable. Its consequences result in patient disability or death. It indicates a real problem in the safety and credibility of the healthcare facility.

The purpose of creating this list was to build a level of public accountability for errors that should never occur.

Both the U.S. government and many insurance companies will now no longer cover certain “never events.”

Software Security Never Events – Work in Progress

Medical never events are very extreme and rarely occur. For software, the consequences may not be as severe but the concept of eliminating vulnerabilities in software that should never occur does have merit.

Several reputable efforts are underway to create a consensus list of errors that should not occur in software, such as the CWE/SANS Top 25 Most Dangerous Programming Errors.

Such errors lead to security bugs that are used in committing cyber-crime, cyber-espionage, and cyber-warfare. To quote from the Top 25 Errors Initiative:

“The impact of these errors is far-reaching. Just two of them led to more than 1.5 million web site security breaches during 2008. And those breaches cascaded onto the computers of people who visited those web sites, turning their computers into zombies.”

Most errors of this type are not taught to, and thus are not well understood by, software developers and testers.

Given the increased frequency and severity of security incidents, now may be the time to start this conversation.

PART 2: IMPLICATIONS AND POTENTIAL CONSEQUENCES

Some Level of Software Discipline Is Overdue

Software and IT are not such young fields anymore; they have been developing for several decades. Moreover, many of the errors that are identified as contributing to insecure software have been around for a long time. These shouldn’t occur anymore.

It’s now time to start imposing some level of discipline onto this field. It should no longer be acceptable to do things that we know don’t work and may cause harm.

Some would argue that having mistakes is the price we pay for progress in our field. While there may be some merit to this argument, we need to examine our responsibilities when developing software with known, preventable risks.

The Implications of Enforcing Software Security Never Events

We can learn lessons from medical never events such as the U.S. government and insurance companies no longer covering specific events. While not optimal, this action has gotten the medical community’s attention.

Perhaps we should no longer pay for software containing, for example, any of the top 25 errors. Some companies might blanch at being held to such a standard where they are not paid or are liable for consequential damages resulting from known software security errors.

Start Small

We could start with a small subset of the agreed-to list, say three to five. What’s most important is starting the conversation.

The Need for Progress

As one example, the Payment Card Industry standard has added application security practices. The consequence of not adopting these is that non-compliant merchants cannot hold credit cardholder data.

Today people such as Grandma and Aunt Millie are required to use the web to fill out forms for information and services, check their bank statements, and access instructions for purchased products. This has great social impact.

If people are afraid and feel that the risk of using the web outweighs the benefit, it’s time to take action.

PART 3: GETTING THE BALL ROLLING

Testing for Feasibility

Bob is contacting everyone from university researchers to practitioners to determine the feasibility and practicality of creating an initial list of software security never events.

He plans to have a major magazine article describing the pros and cons this fall.

Traction in Related Disciplines

This same concept could have broader applicability to other areas of software assurance, including software safety and reliability.

Process improvement is another relevant discipline. The medical community uses military-style checklists to help mitigate the risks of medical never events.

For software, a process example may be ensuring an effective change control board and process before applying a patch.

At the End of the Day

We ought to be able to state “If you don’t follow these basic practices and there is an error that can be traced back to the absence of such practices, there should be consequences.”

If you wish to participate in Bob’s research, he can be contacted by sending email to podcast@cert.org.

Resources

National Quality Forum

Agency for Healthcare Research and Quality

“The term ‘Never Event’ was first introduced in 2001 by Ken Kizer, MD, former CEO of the National Quality Forum (NQF), in reference to particularly shocking medical errors (such as wrong-site surgery) that should never occur. Over time, the list has been expanded to signify adverse events that are unambiguous (clearly identifiable and measurable), serious (resulting in death or significant disability), and usually preventable. The NQF initially defined 27 such events in 2002 and revised and expanded the list in 2006. The list is grouped into six categorical events: surgical, product or device, patient protection, care management, environmental, and criminal.”

CWE/SANS Top 25 Most Dangerous Programming Errors

Open Web Application Security Project (OWASP) Top 10 2007

Software Assurance Forum for Excellence in Code (SAFECode)

Building Security In Maturity Model (BSIMM); CERT podcast on this subject

CERT’s Secure Coding Initiative; CERT podcast on this subject