Network & Security Technologies

SECURITY AWARENESS FOR DEVELOPERS

SECURITY AWARENESS FOR SOFTWARE DEVELOPERS

Knowledge is a Powerful Tool

Only careful design and coding can protect today's business applications. Most programmers, content managers and webmasters understand very little about secure development processes. Instead, they rely on network firewalls for security. Unfortunately, these firewalls cannot distinguish between legitimate application traffic and packets from a hacker intended to subvert the unprotected logic of the software. Just as importantly, the network mechanisms cannot classify sensitive data (e.g., account names, credit card numbers or passwords) passed from the application to unauthorized individuals. Thus, much software represents a "ticking time bomb" to the organization, vulnerable to a wide variety of attacks used to vandalize, disable or subvert their intended service.

Did you know that?

Over the past two years, there has been a sharp rise in security exploits against vulnerable application software. Many companies devote substantial resources to auditing their business applications. These same companies then spend money and time fixing the problems identified. Even worse, most companies expend much greater resources responding to attacks against vulnerable software. Often, these weaknesses cannot be identified during post-development audits, so companies spend twice.

A recent study compared the cost implementing security into applications at various stages of the development life cycle. Some of the interesting findings from that study include:

- Adding security during coding costs 6.5 times more than architecting it during the upfront software design process.
- Implementing security after deployment costs 15 times more than architecting it during the upfront software design process.
- Fixing security holes after deployment costs 100 times more than architecting it during the upfront software design process.
- On average, every 1,000 lines of code has at least 5 to 15 defects (United States Department of Defense and the Software Engineering Institute),

"An Ounce of Prevention" or "A Pound of Cure"

Fortunately, providing development staff with the knowledge and tools to avoid many of these pitfalls is easy and inexpensive. Protecting critical business applications is achievable and affordable with the "Security Awareness for Software Developers" one-day training curriculum from Network & Security Technologies. Through it, personnel will develop an understanding of fundamental rules for project managers, designers and programmers. These include:

- Understand attacks used against web-based and other modern software.
- Produce clear documentation of the security architecture used to safeguard operations and information.
- Conduct rigorous design and code reviews that demonstrate defense against common problems such as buffer overflows and race conditions.
- Test throughout the development cycle for security flaws.
- Maintain solid configuration management procedures that ensure the integrity of every byte and line of code and content.

MORE THREATS COULD BE PREVENTED WITH SIMPLE AND CONSISTENT SOFTWARE DEVELOPMENT

SECURITY AWARENESS FOR DEVELOPERS

Training, a Tried and True Solution

While programmers and designers represent one of the most highly educated segments of an organization's employees, few will take up the study of security techniques on their own. Books on the subject can reach some; however, the vast majority will require a classroom environment to enforce the learning process for those who need it the most. Network and Security Technologies (N&ST) has developed security training for staff involved in the software development process. This progressive curriculum can be tailored to any company's specific environment. It divides the material into two key audience segments:

I. Software Development Managers, Project Managers,
II. Software Architects and Designers, and Programmers.

Each segment is adapted to the technical depth appropriate to its attendees. These concepts apply equally to client-side applets, server-side applications, business logic engines, databases, core applications and interfaces to legacy systems. Examples include scripting languages (e.g., PERL, JavaScript, VBScript), object-oriented languages (e.g., VBasic, C++, Java, .NET), and low-level languages (e.g., C, Assembler, Fortran).

Student Materials and Curriculum

The standard N&ST course includes bound student workbooks covering all presented topics, with recommended additional reading. At your option, N&ST will administer a final exam to gauge students' absorption of the material.

SECURE SOFTWARE DEVELOPMENT FOR PROGRAMMERS COURSE OUTLINE

Module One

- I. Introduction
- II. Security Overview
  - i. Terminology
  - ii. Key Concepts
- III. Security Policy
  - i. Fundamental Definitions
  - ii. Classification
  - iii. Criticality
- IV. Secure Programming Process
  - i. Interpret the design
  - ii. Map execution flow
  - iii. Identify security features
  - iv. Define trusted modules
  - v. Lay out interfaces
  - vi. Provide for logging and exception recording
  - vii. Beware borrowed and open source
  - viii. Code, Test and Review

Module Two

- I. The Gotchas! - Some Popular Software Attacks
- II. The 10 Guiding Principles
- III. Code Review
- IV. Configuration Management
- V. Concluding Remarks