



## Secure Software Begins in the Development Process

A WHITE PAPER PROVIDED TO ASPE BY SECURITY INNOVATION

# Secure Software Begins in the Development Process

written for CIO Update

Edward Adams

CEO  
Security Innovation

 **ASPE**  
SDLC TRAINING **Presents...**



PREPARED BY  
**SECURITY INNOVATION**<sup>®</sup>  
THE APPLICATION SECURITY COMPANY

Security Innovation, Inc.  
187 Ballardvale Street, Suite A170  
Wilmington, MA 01887  
+1.978.694.1008  
[www.securityinnovation.com](http://www.securityinnovation.com)



Deployed software is continuously under misuse or attack. Whether it's an internal application or a public-facing website, software vulnerabilities cause headaches for both the teams that build applications and the teams that manage them. Though there are many risk mitigation options for the software management lifecycle (firewalls, intrusion detection/prevention systems, etc.) these don't address the root cause of the problem— a software development lifecycle that is not integrating security at each phase.

Perimeter defenses are simply unable to stop most software attacks, because abusers are focusing on the application layer, shying away from attacks against the system and networking layer. Only a concerted effort by the software development team to produce more secure applications will protect you from exploitation. Secure applications are a software development challenge that will never be solved until security is addressed *as part of* the software development process.

## A New Business Twist

A "successful application" used to be defined as an application that solved the problems it was designed to solve, and did so with some reasonable performance. From a business standpoint, however, this definition must be rethought. Not because performance alone is a poor objective; but rather, because applications can be leveraged to exploit and destroy your business model – the very model you're building it to assist! – and this must be considered *during* development. Though applications can be "retrofitted" for security post-deployment, this is a very expensive and ineffective approach. According to research from Gartner, the cost of addressing security vulnerability during the development cycle is **less than two percent** the cost of removing such a defect from a deployed production application<sup>1</sup>. And this only counts internal costs. When you add the customer dissatisfaction, loss of reputation, and cost to deploy and support patches, the difference is even greater.

## A New SDLC

All software development methodologies, e.g. Iterative, Agile Waterfall, *et al.*, will benefit from a focus on security practices. In each case, incorporating security-based activities in each phase of development will improve quality and resistance to attack in the final product. Here are the main activities you should incorporate in each phase with some examples to illustrate the points.

- **Define** – Defining requirements explicitly, including how applications should and should not interact with their environment, ensures that projects start with the right foundation. When constructing security requirements consider what a system must not do – think "abuse case" rather than "use case." For example, describe how a malicious user might attack the system or misappropriate the data the application touches. This will force design decisions like when and where to use encryption to protect your data. When defining security requirements, you also need to understand risk – the *business* risk of a successful exploit against the application and how that exploit may affect users. What business processes would be compromised, damaged, or controlled? The costs of liability, patch development, and damage to brand and market share need to be understood and considered in this phase.
- **Design** - The most expensive security defects are those introduced during design. Design considerations include both architectural issues as well as individual component design. At the system level, the objective is to use components and configurations that reduce the application's attack surface. Threat modeling is a very useful activity during this phase as it allows you to identify potential threat quickly and create persistent models with which you can test various configurations and known risks quickly with different architectures. The best tactic here: *education*. Train your architects and

developers on proper use of libraries and patterns, showing them how to implement each component securely.

- **Develop** – Clear requirements and well-tested designs will make for a less troubling and vague development process. But even with those great starts, developers are prone to mistakes like all humans, especially when they aren't properly trained. Educate your team on proper error handling routines by seeking secure coding training. Such courses can show developers how to avoid dangerous code constructs, and how to properly validate input, use encryption and create secure data transport mechanisms. Also, capture knowledge during development. One of the largest costs of secure coding is the cost associated with making the same mistake over and over again. Conducting security code reviews is a very valuable activity – and one of the best places to find that knowledge that you should capture in an internal knowledge base or defect management system.
- **Test** – Testers need to think differently and learn new activities to effectively find security defects. Testers need to give more attention to the application's environment, network connections, configuration, customization options, and non-functional operation (pushing the application to do things it was not designed to do). Instead of thinking "does the application do X, as it's supposed to?" testers need to think "what *if* the application did Y?" and then test for that. It is in the unintended behavior of applications where security defects hide.
- **Deploy and Maintain** – Secure maintenance practices emphasize the importance of understanding the existing security infrastructure and what risk mitigations are already in place. Maintain documentation and utilize monitoring tools to track any changes and audit abnormal behavior patterns for risks they impose on the overall security of the system. These practices become critically important when a security bug is discovered and must be fixed and patched in an already deployed application. While fixing the defect is paramount, the use of a well-crafted incident response plan will help ensure a smooth process that fixes the problem with minimum risk of introducing additional security defects in the application. This will also minimize user down time and make for happier clients. Often a high quality security response program is just as valuable as inherent security quality in your application.

Addressing security in each phase of the SDLC is the most effective way to create highly secure applications. And sourcing training to fill holes in security awareness for managers, architects, developers, testers and administrators will enable the more secure SDLC you need. It will also have the positive side effect of building skills in your team and boosting morale throughout your ranks. Combine this with lower total cost of ownership and enhanced understanding of business risks your applications pose, and you've got a winning solution!

### About the Author

Ed Adams is the President and CEO of Security Innovation, the independent authority on application security risk assessment, risk mitigation and education. He is a seasoned software executive with successful business experience in various-sized organizations that serve the IT security and quality assurance industries.

As CEO, Mr. Adams applies his technical and business skills, as well as his pervasive industry experience in the Application Quality space, to direct world-renowned application security experts and deliver world-class professional services to many of the most recognizable companies in the world including Microsoft, IBM, Visa, Fedex, ING, Symantec, SAP and HP.

Mr. Adams is the founder and business owner of the Application Security Industry Consortium, Inc. (AppSIC), an association of industry technologists and leaders establishing and defining cross-industry application security guidance and metrics. He is on the board of the National Association of Information Security Groups (NAISG).

No stranger to the podium, Mr. Adams has presented to thousands at numerous seminars, software industry conferences, and private companies. He has contributed written and oral commentary for business and

technology media outlets such as New England Cable News, *CSO Magazine*, *SC Magazine*, *CIO Update*, *Investors Business Daily*, *Optimize* and *CFO Magazine*.

Mr. Adams is in the process of writing a book titled "Information Security Management: Survival Guide", which will be published by Wiley & Sons and is due out in August of 2007. He also has maintains a blog with *CSO Magazine* and is a columnist for *CIO Update*.

### **About Security Innovation**

Security Innovation is the authority on application security and a leading provider of risk analysis, risk mitigation and education services to mid-size and Fortune 500 companies. Global technology vendors and enterprise IT organizations rely on our services to identify security risks in their software systems and development processes and facilitate the changes needed to mitigate them. The company is headquartered in Wilmington, MA and has offices in Seattle, WA and Amsterdam, the Netherlands.