



Future Test: Making Security More than Just an Afterthought

A WHITE PAPER PROVIDED TO ASPE BY SECURITY INNOVATION

Future Test: Making Security More than Just an Afterthought

Written for *Software Test & Performance*

Edward A. Adams

CEO
Security Innovation

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



PREPARED BY
SECURITY INNOVATION[®]
THE APPLICATION SECURITY COMPANY

Security Innovation, Inc.
187 Ballardvale Street, Suite A170
Wilmington, MA 01887
+1.978.694.1008
www.securityinnovation.com



I used to shoot people for a living. That experience gave me valuable insight into developing quality software. Before you call America's Most Wanted to turn me in, let me explain. Years ago I was a mechanical design engineer working on non-lethal weapons systems. One weapon I helped design was a perimeter-weighted net that could be fired as a ballistic projectile at a subject from a shotgun mount. The purpose was to restrain but not harm the target. As part of our field testing, I would take fellow (unsuspecting) engineers out into a field and "shoot" gun mounted canisters containing these nets at them, and then we would see if they could escape. It was the most fun "beta" testing a product I ever had!

OK, so now you know I was not a hit man, but you still may wonder just what that experience has to do with software development. Well, before we went out to shoot nets at our coworkers, we had already done a tremendous amount of work in the design phase of the project. We performed extensive testing before we constructed a prototype to test on live people. In the mechanical design world, there is an established process for assessing the quality of your design before you build it:

- ▶ You model the application -- in this case the net and canister propulsion system -- typically using a computer aided design (CAD) system.
- ▶ Once you have a design, you *test the design* using computer aided engineering (CAE) tools. With these kinds of tools you can put a load on a beam, put some flow through gas pipes, or stress test a net. But most importantly, you make sure there are no safety or security flaws before you build it.
- ▶ Model test results are analyzed and any needed design changes are made. Then the improved design is fed back into the test workflow, and you assess the new design. You repeat this process on the model until it passes the requirements -- both functional and safety/security requirements -- placed on it.
- ▶ Wash, rinse, repeat.

When we built our weapon, we already knew that it was going to work and that it would be safe (a key requirement since this was a non-lethal weapon). At that point we were just doing fine tweaks; there were no costly design changes or architectural changes late in the game. Only **after** we tested our model and verified that the design is architecturally sound and safe did we start building the prototype -- or the Beta to follow the analogy into the software world. Just like your last software project, right? :-) This is where software development and test is going ... or where it *should* be going in the future.

Where did all the Developers Go?

Enrollment in Computer Science as a major has declined in the US to only 1.1% of all majors last year, down from 3.7 the year before. This is largely because of all our outsourcing to other countries, a fine trend; however, it does necessitate the retention of qualified testing and assessment capabilities for companies who outsource development. And unlike functionality or performance, security is a critical component to software testing -- if you ship with a product that's difficult to use or has poor performance, your users may get annoyed; however, if you ship without mind to security you are putting your entire business at risk: you lose profit, customer loyalty, market share, and your executives or company might get sued by someone like the FTC (see <http://www.ftc.gov/opa/2005/06/bjswholesale.htm> or http://www.infoworld.com/article/06/05/10/78177_HNftcsettlescase_1.html)

Even though courses in software security are more common and many testers now understand what it means to test for security, this does not mean that anybody can do it -- security testing takes imagination, technical expertise, and a security mindset, which means thinking in terms of abuse cases rather than use cases (e.g., how can an actor attack this interface?) and negative requirements rather than positive requirements (e.g., the application shall be hardened against SQL injection attacks). You

can teach anybody how to run test cases in a test plan, but finding security bugs takes extra training and a mindshift.

Tools, Vaults and Locks

Commonly, I hear testers asking for a security tool that “does it for me” finding all security vulnerabilities in my software. Though there are a couple of decent security testing tools on the market today, most try to do too much and fail at it most of the time. Future test will see security tools take the form of utilities that are built for specific tasks, e.g., finding Buffer Overflows for any interface. These tools will be nearly useless, however, in 5 years since most new applications will be written (or rewritten) in java or .net languages. Testing web applications will also evolve to dive faults from both the server and client, modify results as necessary through a proxy and recording the outcome. Fault simulation, e.g., low memory resources on the server, will also be a critical future test category for security. Having such a model allows a tool to automatically test many components of the system at the same time rather than isolated components.

Security will *have* to be integrated into organizations’ software development lifecycle (SDLC) via guidance and education. Top-down management pressures will force development and test teams to learn new security skills and they will seek this in the form of training and knowledge retention – I envision a security vault of knowledge, accessible to all development team members to draw upon at will. Ideally, this vault is integrated with the development and test environments so instant access and integration with existing infrastructure is achieved – open the vault and unlock your mind.

The Future for SDLC and Test Teams

The biggest change to the SDLC: Threat Modeling. This is fast becoming ubiquitous in Software Companies, but in 5 years Threat Modeling will become a standard practice early in the SDLC with the aid of business analysts or product managers to get a full picture of the problems. Threat models give you a view to the biggest threats fast; but more valuably, they can be re-used as new vulnerabilities become known. Pump a new threat into the model and instantly determine if there is a risk for your specific context and application. And validations/checks against the threat model can be peppered throughout the SDLC, becoming a process much like requirements are today. In software that really works threat models will be tightly coupled to the SDLC, in less successful projects they will be left by the wayside early.

Testers are stretched pretty thin as is, and with new security requirements, test teams are going to have to develop specialists. Security testers need to be dedicated to security throughout the SDLC to be productive. There will still be plenty of “legacy code” (i.e., anything 5+ years old) that needs to be validated and tested; fortunately, a lot of that can be automated, unlike security. I have already seen the role of Security Guru at several companies. It’s usually one person that used to be a tester that has been dedicated to testing security – thinking like an attacker or hacker and perpetually trying to compromise the application. This is a good start, but we’ve got a long way to go to reach Nirvana.

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.