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

An Experience-Based Maturity Model for Software Security

Key Message: Observed practice, represented as a maturity model, can serve as a basis for developing more secure software.

Executive Summary

As a community, we have been working to identify and use practices for developing more secure software for about ten years. Consequently, we have sufficient experience to create a yardstick that you can use to measure your software security efforts, determine where you stand with respect to your peers, and decide what steps are needed to move forward.

In this podcast, Gary McGraw (CTO, Cigital), Brian Chess (Chief Scientist, Co-founder, Fortify Software), and Sammy Migues (Principal Consultant, Director of Knowledge Management, Cigital) discuss their efforts to develop a maturity model for building security into software throughout the development life cycle. The model is based on analyzing real word experiences from nine organizations.

PART 1: MOTIVATION AND TARGET AUDIENCE; DRAWING FROM OBSERVED PRACTICES

The Time Is Now

A number of organizations have been using software security practices for the past decade. We now have sufficient experience to draw from by studying actual programs.

There are several existing methods for developing more secure software including Cigital’s Touchpoints, Microsoft’s SDL (security development lifecycle), and OWASP’s CLASP (Open Web Application Security Project; Comprehensive Lightweight Application Security Process).

That said, the model authors determined that now was the right time to study successful programs. They selected nine from the thirty three they were familiar with.

Their objective was to create a yardstick to:

- measure your own software security initiative
- determine where you stand in comparison to your peers
- help plan improvements to your software security program

The Main Constraint

The authors agreed that a practice would be included in the model if and only if it was based on real data and actual observations.

Target Audience

The Building Security In Maturity Model (BSIMM) is intended for:

- executives, including CIOs and CISOs, who are responsible for enterprise software security initiatives
- software security groups (SSGs) that can use BSIMM as an organizing structure for current practices and to introduce new practices

Observed Practices from Nine Organizations

The model was built by interviewing executives at nine companies who are in charge of their software security initiatives.

These included four financial services companies (Wells Fargo, Depository Trust, Clearing Corporation); three software companies (Adobe, Microsoft, QUALCOMM); and two technology companies (EMC, Google). Two of the nine companies have chosen to remain anonymous.

The intent is to continue adding organizations and practices to BSIMM using the same interviewing and observation methods.

PART 2: STRUCTURE AND SCOPE; WHERE TO START

The Model's Structure

The model is built upon a Software Security Framework that was developed prior to starting the interview process. The framework comprises:

- 4 major domains (governance, intelligence, SSDL touchpoints, deployment)
- 12 practices across the 4 domains
- 110 activities across the 12 practices

Model Practices and Activities

As examples, practices include training, architecture analysis, and strategy and metrics.

[From the BSIMM web site, the other practices are: compliance and policy, attack models, security features and design, standards and requirements, code review, security testing, penetration testing, software environment, and configuration management and vulnerability management.]

Each activity description includes an objective, actions to take, and 2-3 examples based on actual practice.

Activities include such familiar tasks as secure coding, the use of static analysis tools, assurance cases, and attack patterns.

The authors outlined some surprises in an InformIT article titled "Software Security Top Ten Surprises." One of the most unexpected findings was the use of fuzz testing and tools in a very sophisticated manner, taking advantage of class structure and APIs (application programming interface).

Getting Started

It is not appropriate to apply BSIMM at the individual software project level. The intent is to build a software security initiative at the enterprise level that is then applied to each software project.

The first condition for success is to ensure the level of executive buy-in is sufficient to allocate resources for the SSG. You're not ready if you cannot dedicate staff to making sure that your software is secure.

Estimating Time and Resources

Interviewed organizations have been practicing software security for 4.5 years on average.

One of the most interesting outcomes is that observed size of SSGs tends to average one percent of the size of their software development group.

Starting from Strength

Financial services organizations tend to be stronger in policy and compliance practices. Software vendors are better with testing practices. In other words, you start with effective, current practices and build from your strengths.

BSIMM describes ten practices that all nine organizations are currently using. These include training and making sure your SSG is good at executing an activity before it is rolled out.

[From the BSIMM model, the other eight practices are: create evangelism role/internal marketing; create policy; create/use material specific to company history; build/publish security features; use automated tools with manual review; integrate black box security tools into the QA process; use external pen testers to find problems; ensure host/network security basics in place.]

PART 3: SETTING EXPECTATIONS; MAKING THE BUSINESS CASE

Return on Investment

The first, most obvious benefit is having a yardstick based on actual experience against which to compare your current practices. In addition, the model presents a structure that is easy to use and consume.

The harder question is “when will I see improvement in a specific activity?” This really depends on the organization, its current state, and its ability to prioritize activities.

Looking for a few quick wins and demonstrating success early on will help build momentum. The idea is to find some things you’re really good at and start there (such as vulnerability testing, network security, and training).

Making the Business Case

Every executive wants to know if they’re spending enough or too much. BSIMM provides a clear yardstick and set of benchmarks to compare your current practices against those of your peers.

BSIMM presents goals and objectives that easily map to business goals and objectives, such as informed risk management decisions, cost reduction, and improved code quality. Once the business goal is established, it’s clear which BSIMM activities best support it.

That said, there is little to no benchmark data on the costs and benefits of a specific practice or activity. The problem in comparing across organizations is that everyone uses metrics (such as cost savings and costs to remediate) but each metric is only applicable in a particular business culture.

More anthropology is needed.

In Closing

With respect to software security, the time for alchemy is over. Given that we can perform data-driven activities based on actual field experience, the time for applying science and engineering to software security is now.

Resources

BSIMM is licensed under the Creative Commons Attribution-Share Alike 3.0 License.

Contact the authors at the BSIMM web site with follow up questions and requests to participate.

The BSIMM web site

BSIMM InformIT articles:

- “Software [In]security: Software Security Top 10 Surprises,” 15 December 2008.
- “Software [In]security: Nine Things Everybody Does: Software Security Activities from the BSIMM,” 9 February 2009.
- “Software [In]security: The Building Security In Maturity Model (BSIMM),” 16 March 2009.

BSIMM reviews, articles, and blogs

The Microsoft SDL Optimization Model (complimentary to BSIMM)