Building Security In Maturity Model
Fifteen Things That Everybody Does

Gary McGraw, Ph.D.
Chief Technology Officer, Cigital

February 2010
Cigital

- Founded in 1992 to provide software security and software quality professional services
- Recognized experts in software security and software quality
  - Widely published in books, white papers, and articles
  - Industry thought leaders
Breaking new ground

- Building Security In Maturity Model
- Real data from (30) real initiatives
- McGraw, Chess, & Migues
58 software security initiatives

- 31 Financial
- 9 ISV
- 9 Tech
- 2 Defense
- 5 Retail
- 1 Oil
- 1 Behemoth

- visa europe
- thomson/reuters
- BP
- SAP
- nokia
- ebay
- mckesson

- ABN/amro
- ING
- telecom italia
- swift
- standard life
- cigna
- AON

- microsoft
- dtcc
- emc
- fidelity
- adobe
- wells fargo
- goldman sachs
- google
- qualcomm
- morgan stanley
- usaf
- dell
- pershing
- the hartford
- barclays capital
- bank of tokyo
- ups
- bank of montreal
- sterling commerce
- coke
- mastercard
- apple

- cisco
- bank of america
- walmart
- finra
- vanguard
- college board
- oracle
- state street
- omgeo
- motorola
- general electric
- lockheed martin
- intuit
- vmware
- amex
- bank of ny mellon
- harris bank
- paypal
- symantec
- AOL
- CA
- time warner
BSIMM original nine

And two unnamed financial services firms
BSIMM Europe (nine EU firms)

And four unnamed firms
On cargo cults and divining rods

- InformIT article on BSIMM website http://bsi-mm.com
A Software Security Framework

The Software Security Framework (SSF)

<table>
<thead>
<tr>
<th>Governance</th>
<th>Intelligence</th>
<th>SSDL Touchpoints</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy and Metrics</td>
<td>Attack Models</td>
<td>Architecture Analysis</td>
<td>Penetration Testing</td>
</tr>
<tr>
<td>Compliance and Policy</td>
<td>Security Features and Design</td>
<td>Code Review</td>
<td>Software Environment</td>
</tr>
<tr>
<td>Training</td>
<td>Standards and Requirements</td>
<td>Security Testing</td>
<td>Configuration Management and Vulnerability Management</td>
</tr>
</tbody>
</table>

- Four domains
- Twelve practices
- See informIT article on BSIMM website
  http://bsi-mm.com
## Training practice skeleton

<table>
<thead>
<tr>
<th>Objective</th>
<th>Activity</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1.1: Promote culture of security throughout the organization</td>
<td>Provide awareness training</td>
<td>1</td>
</tr>
<tr>
<td>T1.2: Ensure new hires enhance culture</td>
<td>Include security resources in onboarding</td>
<td></td>
</tr>
<tr>
<td>T1.3: Act as informal resource to leverage teachable moments</td>
<td>Establish SSG office hours</td>
<td></td>
</tr>
<tr>
<td>T1.4: Create social network tied into dev</td>
<td>Identify satellite during training</td>
<td></td>
</tr>
<tr>
<td>T2.1: Build capabilities beyond awareness</td>
<td>Offer role-specific advanced curriculum (tools, technology stacks, bug parade)</td>
<td>2</td>
</tr>
<tr>
<td>T2.2: See yourself in the problem</td>
<td>Create/use material specific to company history</td>
<td></td>
</tr>
<tr>
<td>T2.3: Keep staff up-to-date and address turnover</td>
<td>Require annual refresher</td>
<td></td>
</tr>
<tr>
<td>T2.4: Reduce impact on training targets and delivery staff</td>
<td>Offer on-demand individual training</td>
<td></td>
</tr>
<tr>
<td>T2.5: Educate/strengthen social network</td>
<td>Hold satellite training/events</td>
<td></td>
</tr>
<tr>
<td>T3.1: Align security culture with career path</td>
<td>Reward progression through curriculum (certification or HR)</td>
<td>3</td>
</tr>
<tr>
<td>T3.2: Spread security culture to providers</td>
<td>Provide training for vendors or outsource workers</td>
<td></td>
</tr>
<tr>
<td>T3.3: Market security culture as differentiator</td>
<td>Host external software security events</td>
<td></td>
</tr>
</tbody>
</table>
15 Things Everybody Does

- identify gates
- unify regulations
- know PII obligations
- publish policy
- awareness training
- data classification
- identify features
- security standards
- review security features
- static analysis tool
- QA boundary testing

- external pen testers
- good network security
- incident response
- close ops bugs loop

BSIMM TOP TEN AVG (10)  FIRM
Top 10 things
- green = good?
- red = bad?

“Blue shift” practices to emphasize
- activities you should maybe think about in brown
How good are the levels we chose?

Maturity Level 1

Maturity Level 2

Maturity Level 3
Identify gate locations, gather necessary artifacts. The software security process will eventually involve release gates at one or more points in the software development lifecycle (SDLC) or SDLCs. The first two steps toward establishing these release gates is to identify gate locations that are compatible with existing development practices and to begin gathering the input necessary for making a go/no go decision. Importantly at this stage, the gates are not enforced. For example, the SSG can collect security testing results for each project prior to release, but stop short of passing judgment on what constitutes sufficient testing or acceptable test results.
Know all regulatory pressures and unify approach. If the business is subject to regulatory or compliance drivers such as FFIEC, GLBA, OCC, PCI DSS, SOX, SAS 70, HIPAA or others, the SSG acts as a focal point for understanding the constraints such drivers impose on software. The SSG creates a unified approach that removes redundancy from overlapping compliance requirements. A formal approach will map applicable portions of regulations to control statements explaining how the organization will comply.
Identify PII obligations. The way software handles personally identifiable information (PII) could well be explicitly regulated, but even if it is not, privacy is a hot topic. The SSG takes a lead role in identifying PII obligations stemming from regulation, customer demand, and consumer expectations. It uses this information to promote best practices related to privacy. For example, if the organization processes credit card transactions, the SSG will identify the constraints that PCI DSS places on the handling of cardholder data.
Create policy. The SSG guides the rest of the organization by creating or contributing to policy that satisfies regulatory requirements and customer-driven security requirements. The policy provides a unified approach for satisfying the (potentially lengthy) list of external security drivers. As a result, project teams can avoid learning the details involved in complying with all applicable regulations. Likewise, project teams don’t need to re-learn customer security requirements on their own. The SSG policy documents are sometimes focused around major compliance topics such as the handling of personally identifiable information or the use of cryptography.
Provide awareness training. The SSG provides awareness training in order to promote a culture of security throughout the organization. Training might be delivered by members of the SSG, by an outside firm, by the internal training organization, or through a computer-based training system. Course content is not necessarily tailored for a specific audience. For example, all programmers, quality assurance engineers, and project managers could attend the same Introduction to Software Security course.
Create data classification scheme and inventory. The organization agrees upon a data classification scheme and uses the scheme to inventory its software according to the kinds of data the software handles. This allows applications to be prioritized by their data classification. Many classification schemes are possible—one approach is to focus on PII. Depending upon the scheme and the software involved, it could be easiest to first classify data repositories, then derive classifications for applications according to the repositories they use.
Build/publish security features (authentication, role management, key management, audit/log, crypto, protocols). Some problems are best solved only once. Rather than have each project team implement all of their own security features, the SSG provides proactive guidance by building and publishing security features for other groups to use. Project teams benefit from implementations that come pre-approved by the SSG, and the SSG benefits by not having to repeatedly track down the kinds of subtle errors that creep into features such as authentication, role management, audit/logging, key management, and cryptography.
Create security standards. Software security requires much more than security features, but security features are part of the job as well. The SSG meets the organization’s demand for security features by creating standards that explain the accepted way to adhere to policy and carry out specific security-centric operations. A standard might describe how to perform authentication using J2EE or how to determine the authenticity of a software update. (See [SFD1.1] Build and publish security features for one case where the SSG provides a reference implementation of a security standard.)
Perform security feature review. To get started with architecture analysis, center the analysis process on a review of security features. Reviewers first identify the security features in an application (authentication, access control, use of cryptography, etc.) then study the design looking for problems that would cause these features to fail at their purpose or otherwise prove insufficient. At higher levels of maturity this activity is eclipsed by a more thorough approach to architecture analysis not centered on features.
Use automated tools along with manual review. Incorporate static analysis into the code review process in order to make code review more efficient and more consistent. The automation does not replace human judgment, but it does bring definition to the review process and security expertise to reviewers who are not security experts.
Ensure QA supports edge/boundary value condition testing. The QA team goes beyond functional testing to perform basic adversarial tests. They probe simple edge cases and boundary conditions. No attacker skills required.
Use external penetration testers to find problems. Many organizations are not willing to address software security until there is unmistakable evidence that the organization is not somehow magically immune to the problem. If security has not been a priority, external penetration testers demonstrate that the organization’s code needs help. Penetration testers could be brought in to break a high-profile application in order to make the point.
Ensure host and network security basics are in place. The organization provides a solid foundation for software by ensuring that host and network security basics are in place. It is common for operations security teams to be responsible for duties such as patching operating systems and maintaining firewalls.
Create or interface with incident response. The SSG is prepared to respond to an incident. The group either creates its own incident response capability or interfaces with the organization’s existing incident response team. A regular meeting between the SSG and the incident response team can keep information flowing in both directions.
Identify software defects found in operations monitoring and feed them back to development. Defects identified through operations monitoring are fed back to development and used to change developer behavior. The contents of production logs can be revealing (or can reveal the need for improved logging).
Using BSIMM

- BSIMM released March 2009 under creative commons
  - http://bsi-mm.com
  - v1.5 includes Europe (November 2009)
  - Italian and German translations
  - steal the data if you want
- BSIMM is a yardstick
  - Use it to see where you stand
  - Use it to figure out what your peers do
- BSIMM is growing (30+)
  - BSIMM Europe
  - BSIMM Begin
Where to Learn More
informIT & Justice League

- www.informIT.com
- No-nonsense monthly security column by Gary McGraw
- www.cigital.com/justiceleague
- In-depth thought leadership blog from the Cigital Principals
  - Scott Matsumoto
  - Gary McGraw
  - Sammy Migues
  - Craig Miller
  - John Steven
IEEE Security & Privacy Magazine + 2 Podcasts

- Building Security In
- Software Security Best Practices column edited by John Steven
- www.computer.org/security/bsisub/

www.cigital.com/silverbullet
www.cigital.com/realitycheck
Software Security: the book

- How to DO software security
  - Best practices
  - Tools
  - Knowledge
- Cornerstone of the Addison-Wesley Software Security Series
- www.swsec.com
For more on BSIMM

- http://bsi-mm.com
- See the Addison-Wesley Software Security series
- Send e-mail: gem@cigital.com

“So now, when we face a choice between adding features and resolving security issues, we need to choose security.”

-Bill Gates