• Review of existing secure SDLC efforts
• Understanding the model
• Applying the model
• SAMM and the real world
By the end, you’ll be able to...

• Evaluate an organization’s existing software security practices
• Build a balanced software security assurance program in well-defined iterations
• Demonstrate concrete improvements to a security assurance program
• Define and measure security-related activities throughout an organization
Review of existing secure SDLC efforts
CLASP

- Comprehensive, Lightweight Application Security Process
  - Centered around 7 AppSec Best Practices
  - Cover the entire software lifecycle (not just development)
- Adaptable to any development process
  -Defines roles across the SDLC
  - 24 role-based process components
- Start small and dial-in to your needs
Microsoft SDL

• Built internally for MS software
• Extended and made public for others
• MS-only versions since public release
Touchpoints

- Gary McGraw’s and Cigital’s model
Lessons Learned

• Microsoft SDL
  • Heavyweight, good for large ISVs
• Touchpoints
  • High-level, not enough details to execute against
• CLASP
  • Large collection of activities, but no priority ordering
• ALL: Good for experts to use as a guide, but hard for non-security folks to use off the shelf
Drivers for a Maturity Model

- An organization’s behavior changes slowly over time
- Changes must be iterative while working toward long-term goals
- There is no single recipe that works for all organizations
- A solution must enable risk-based choices tailored to the organization
- Guidance related to security activities must be prescriptive
- A solution must provide enough details for non-security people
- Overall, must be simple, well-defined, and measurable
Therefore, a viable model must...

- Define building blocks for an assurance program
- Delineate all functions within an organization that could be improved over time
- Define how building blocks should be combined
- Make creating change in iterations a no-brainer
- Define details for each building block clearly
- Clarify the security-relevant parts in a widely applicable way (for any org doing software dev)
Understanding the model
SAMM Business Functions

• Start with the core activities tied to any organization performing software development

• Named generically, but should resonate with any developer or manager

Governance

Construction

Verification

Deployment
SAMM Security Practices

- From each of the Business Functions, 3 Security Practices are defined.
- The Security Practices cover all areas relevant to software security assurance.
- Each one is a ‘silico’ for improvement.
Under each Security Practice

- Three successive Objectives under each Practice define how it can be improved over time
- This establishes a notion of a Level at which an organization fulfills a given Practice
- The three Levels for a Practice generally correspond to:
  - (0: Implicit starting point with the Practice unfulfilled)
  - 1: Initial understanding and ad hoc provision of the Practice
  - 2: Increase efficiency and/or effectiveness of the Practice
  - 3: Comprehensive mastery of the Practice at scale
# Education & Guidance

<table>
<thead>
<tr>
<th>EG 1</th>
<th>EG 2</th>
<th>EG 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACTIVE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offer development staff access to resources around the topics of secure programming and deployment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Educate all personnel in the software life-cycle with role-specific guidance on secure development</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mandate comprehensive security training and certify personnel for baseline knowledge</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ACTIVITIES</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Conduct technical security awareness training</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Build and maintain technical guidelines</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Conduct role-specific application security training</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Utilize security coaches to enhance project teams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. Create formal application security support portal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B. Establish role-based examination/certification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Per level, SAMM defines...

- Objective
- Activities
- Results
- Success Metrics
- Costs
- Personnel
Approach to iterative improvement

• Since the twelve Practices are each a maturity area, the successive Objectives represent the “building blocks” for any assurance program.

• Simply put, improve an assurance program in phases by:
  1. Select security Practices to improve in next phase of assurance program
  2. Achieve the next Objective in each Practice by performing the corresponding Activities at the specified Success Metrics
Applying the model
## Conducting assessments

- SAMM includes assessment worksheets for each Security Practice

<table>
<thead>
<tr>
<th>Education &amp; Guidance</th>
<th>YES/NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>✦ Have most developers been given high-level security awareness training?</td>
<td></td>
</tr>
<tr>
<td>✦ Does each project team have access to secure development best practices and guidance?</td>
<td></td>
</tr>
<tr>
<td>✦ Are most roles in the development process given role-specific training and guidance?</td>
<td></td>
</tr>
<tr>
<td>✦ Are most stakeholders able to pull in security coaches for use on projects?</td>
<td></td>
</tr>
<tr>
<td>✦ Is security-related guidance centrally controlled and consistently distributed throughout the organization?</td>
<td></td>
</tr>
<tr>
<td>✦ Are most people tested to ensure a baseline skill-level?</td>
<td></td>
</tr>
</tbody>
</table>
Assessment process

- Supports both lightweight and detailed assessments
- Organizations may fall in between levels (+)
Creating Scorecards

- Gap analysis
- Capturing scores from detailed assessments versus expected performance levels
- Demonstrating improvement
- Capturing scores from before and after an iteration of assurance program build-out
- Ongoing measurement
- Capturing scores over consistent time frames for an assurance program that is already in place
Roadmap templates

- To make the “building blocks” usable, SAMM defines Roadmaps templates for typical kinds of organizations
  - Independent Software Vendors
  - Online Service Providers
  - Financial Services Organizations
  - Government Organizations
- Organization types chosen because
  - They represent common use-cases
  - Each organization has variations in typical software-induced risk
  - Optimal creation of an assurance
Building Assurance Programs

1. Start
2. Conduct initial assessment
3. Create empty roadmap
   - If yes, Select appropriate roadmap
   - If no, go to step 4
4. Adding another phase?
   - If no, go to step 5
   - If yes, go to step 6
5. Done
6. Select Practices to Mark selected improvements
   - Adjust roadmap to organization
   - Select appropriate roadmap template?
Case Studies

- A full walkthrough with prose explanations of decision-making as an organization improves
- Each Phase described in detail
  - Organizational constraints
  - Build/buy choices
- One case study exists today, several more in progress using industry partners
The SAMM 1.0 release
SAMM and the real world
Expert contributions

• Built based on collected experiences with 100’s of organizations
• Including security experts, developers, architects, development managers, IT managers

**Author & Project Lead**
Pravir Chandra

**Contributors/Reviewers**
Fabio Arciniegas  Brian Chess  Matteo Meucci
Matt Bartoldus  Dinis Cruz  Jeff Payne
Sebastien Deleersnyder  Justin Derry  Gunnar Peterson
Jonathan Carter  Bart De Win  Jeff Piper
John Steven  Chad Thunberg
Colin Watson  Jeff Williams
Industry support

- Several more case studies underway
The OpenSAMM Project

- [http://www.opensamm.org](http://www.opensamm.org)
- Beta released August 2008, 1.0 released March 2009
- Dedicated to improving the SAMM framework and related resources
- Always vendor-neutral, but lots of industry participation
- Open and community driven
- Targeting new releases every ~18-24 months
- Change management process
Future plans

- Mappings to existing standards and regulations (many underway currently)
  - PCI, COBIT, ISO-17799/27002, ISM3, etc.
- Additional roadmaps where need is identified
- Additional case studies
- Feedback for refinement of the model
- Translations into other languages
What to do next

• Download the OpenSAMM 1.0 PDF from http://www.opensamm.org (1 min)
• Read the Executive Summary (5 min)
• Do the Assessment Worksheets (20 min)
• Discuss your organization’s results with stakeholders (30 min)
Thanks for your time! Questions?

http://www.opensamm.org

Pravir Chandra
OpenSAMM Project Lead
chandra@owasp.org
Other modern approaches

- Microsoft SDL Optimization Model
- Fortify/Cigital Building Security In Maturity Model (BSIMM)
SDL Optimization Model

- Built by MS to make SDL adoption easier.

The four security maturity levels of the SDL Optimization Model:

- **Basic**
  - Security is reactive
  - Customer risk is undefined

- **Standardized**
  - Security is proactive
  - Customer risk is understood

- **Advanced**
  - Security is integrated
  - Customer risk is controlled

- **Dynamic**
  - Security is specialized
  - Customer risk is minimized

The five capability areas of the software development process:

- Training, Policy, and Organizational Capabilities
- Requirements and Design
- Implementation
- Verification
- Testing
BSIMM

- Framework derived from SAMM Beta
- Based on collected data from 9 large firms

<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>
BSIMM vs. OpenSAMM

- Top-down (BSIMM) vs. bottom-up (SAMM)
- Observations from unproven systems out of steady state
- Faith-based cargo cults being directed by experts
- Ask yourself...
  - Are you trying to measure yourself against the big dogs?
  - Are you just trying to get a program started?
  - Do observed activities show you the right stepping stones? Do you know when to say when (risk mgmt)?