How to Develop a Comprehensive Application Security Program

Kathy Doolittle
CISSP, CISM, QSA, MSIA
Director, Strategic Services
Agenda

- Organizational Challenges and Trends
- Elements of the Application Security Program
- Developing the Program
- Application Program Strategic Roadmap
Organizational Challenges

- Executive-Level Support
- Limited Awareness of Application Security Risk
- Uniqueness & Complexity of Applications
- Control of Source Code
  - (COTS and outsourced development)
- Confusion of Ownership
  - (Security Group, IT Audit, QA, or Developer Group)
- Establishing Application Security Standards
  - Variety of Application Technologies & Platforms Utilized
- Traditional Controls Ineffective
  - (FW, IDS, etc.)
- Developing Prevention & Protection Strategies
- Consumerization
- Pressure for Speed to Market
Current Trends

• Unprecedented growth in mobile computing and mobile applications

• Shift to Agile programming development, creates more challenges integrating security into the SDLC

• Increasing need for more secure software, better quality, and faster time-to-deployment as market differentiators

• Business justification of investing in security throughout the System Development Life Cycle (SDLC)

• High cost and irreparable brand damage resulting from high-profile hacking incidents and data breaches

• Proactive compliance with applicable laws and regulatory requirements
  
  • Twenty-five percent (25%) of the 150 companies surveyed by Forrester Consulting recently [2010] said that the most effective argument to obtain funding for software security was “to meet our compliance requirements”.

  • Compliance-centric strategy often encourages a behavior of “implementing the bare minimum” to get by and does not support a business-driven approach to secure SDLC
The desktop internet ramp was just a warm-up act for what we’re seeing happen on the mobile internet. The pace of mobile innovation is “unprecedented, I think, in world history.”

Mary Meeker, Morgan Stanley – April 2010
Mobile Enterprise Apps are Rapidly Evolving

*In House App Stores
Cost of Fixing Defects

According to a study conducted by IBM Systems Sciences Institute on the relative cost of fixing defects, the cost increases exponentially the later it is addressed in the SDLC.

<table>
<thead>
<tr>
<th></th>
<th>Design</th>
<th>Implementation</th>
<th>Testing</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements*</td>
<td>$139</td>
<td>$455</td>
<td>$977</td>
<td>$7,136</td>
</tr>
<tr>
<td></td>
<td>$1</td>
<td>Up to 6.5 times</td>
<td>Up to 15 times</td>
<td>Up to 100 times</td>
</tr>
</tbody>
</table>

*B. Boehm and V. Basili*
Sustained Maturity
Centralized People, Processes and Technology
Application security integrated seamlessly into quality lifecycle, becoming third pillar

Application security team has Enterprise influence
Security addressed throughout SDLC and applied retroactively to legacy applications

Security Fitness
Security baked into SDLC, discussed during design phase
Security checkpoints defined and enforced
Centralized, reusable resources for developers
Centralized testing and remediation tracking
Development mentors identified and trained

Proactive Security
Champion and stakeholders identified
Policies, standards & processes established
Tools evaluated and purchased
Automated and manual internal testing
Developer training and awareness

Reactive Security
Standards-based internal processes lead to a basic level of awareness
Some manual testing, looking into automation
Recognize need for application security, but don’t know where to start

Security Unaware
No documented Application Security practices
No internal testing, merely annual penetration test
No application security awareness or developer training

Increasing Maturity
Decreasing Overall Development Cost
# Application Security via SDLC

## Design
- **Threat Modeling**: Workshop session surrounding application risk, threat, attack, weakness, and vulnerabilities discussion
- **Architecture Review**: Analyze application tiered architecture based on design and implementation
- **Best Practices Documentation**: Design and customize BP specific to customer environment, language, policy, etc.

## Development
- **Secure SDLC**: Review existing security practices, procedures, and processes throughout the Software Development Life Cycle (SDLC) and recommend changes
- **Secure Coding Training**: The application security course brings awareness of common Web application vulnerabilities and the impact they have on businesses
- **Automated Assessment Tools**: Recommend and integrate 3rd-party assessment tools into the SDLC, provide developer/management training
- **Remediation Planning**: Take results of security assessments and develop a remediation plan. Work with business owners to prioritize and align remediation with business goals. Manage remediation process.

## QA
- **Runtime Analysis**: Standard black-box testing including the use of automated tools and manual analysis
- **Static Analysis**: Review source code for backdoors, logic bombs, and other vulnerabilities not identified by fault injection/black box testing. Static analysis must always be combined with runtime in order to verify results and increase overall efficiency of assessment. Use 3rd-party automated source scanner to obtain 100% code coverage. Augment with manual review.
- **Security Due Diligence**: Issue a client-facing or independent 3rd party letter to validate the application services provided by assessor for due diligence purposes. This could be just a letter describing what we tested all the way to a full-blown attestation process. Formal due diligence could be for a specific application or the development model as a whole.
Developing an Application Security Program

- Build upon existing program, framework or methodology - customized form or any derivation of the traditional SDLC process is acceptable

- Reference established or common industry methodology, framework, or standard
  - Software Assurance Maturity Model (SAMM) – built upon four core business functions with twelve security practices tied to them
  - Microsoft Security Development Lifecycle (SDL) – consists of seven phases with seventeen security practices
  - Other models and standards (BSIMM, CMM, ISO27035)

- Evaluate and adopt commonly known secure software development control techniques

- Combine with Corporate Initiatives to support and expand AppSec framework
Example: Referencing SAMM

- The Software Assurance Maturity Model (SAMM) is an open framework that could be used to implement a secure application security program.
- SAMM is built upon four core business functions with twelve security practices tied to them.
Elements of Application Security Program Initiative

1. Application & Information Inventory
2. Meeting and Maintaining Compliance Requirements
3. Developing Internal Application Security Standards
4. Establishing Initiative Sponsor & Owners
5. Internal IT Audit Function
6. Defining Methods of Application Security Due Diligence
7. Performing Due Diligence on Affiliates/Business Partner Applications
8. Outsourcing vs. In-House
9. Prioritization of Applications & Frequency of Testing
10. Training & Staffing Requirements
11. Application Solutions & Tools
13. Remediation Procedures
14. Metrics, Reporting & Documentation

*http://www.fishnetsecurity.com/6labs/resource-library/webinar/how-develop-application-security-program
Developing an Application Security Program - Governance

SAMM Deployment Security Practices

- Strategy & Metrics
- Policy & Compliance
- Education & Guidance

Program Elements for Consideration

- Meet/ Maintain Compliance
- Establish Sponsors/Owners
- Training & Staffing Reqs.
- Internal IT Audit Function
- Developing Standards
- Metrics, Reporting & Documentation

Sample Decisions/Considerations

- Overall Strategy – Secure at Source
- Identify Metrics
- Identify Compliance Drivers
- Role of IT Audit – Consult or Audit
- Classroom /CBT
- Develop Secure Coding Standards
Meeting and Maintaining Compliance Requirements

• What is my organization’s responsibility to meet application related standards or compliance requirements?

• Example: Payment Card Industry (PCI) Data Security Standard (DSS) version 2.0
  – 6.5 -- Develop applications based on secure coding guidelines. Prevent coding vulnerabilities via SDLC
  – 6.6 -- For public-facing web applications, address new threats and vulnerabilities on an ongoing basis and ensure these applications are protected against known attacks…” via manual or automated assessment or WAF
  – 11.3.2 -- states companies must conduct annual “application-layer penetration tests” at minimum for 6.5 vulnerabilities

• Others: HIPAA Security Rules, Sox, Data Protection Act, FISMA, GLBA, EU Safe Harbor, NCUA, COPPA, NERC
Develop Internal Application Security Standards

• Establish application specific security standards within the Software Development Lifecycle (SDLC)
• Develop mandatory application security requirement criteria for developers
• Provide security requirements surrounding development frameworks (i.e. .NET, Java)
• Ensure personnel involved understand these security requirements
• Code re-use repository (validation, encoding, encryption)
Training and Staffing Requirements

- Training internal security staff is critical and sometimes required for application security personnel (e.g. PCI 6.5a)

- High-level to specialized hands-on courses (focused on specific development language)
  
  - if hands on is not feasible initially because of size or disparity we have seen customized CBT to at least get the core concepts and examples out to developers

- Specialized requirements and skills for mobile apps

- Education is key
Developing an Application Security Program - Construction

SAMM Deployment

Security Practices

- Threat Assessment
- Security Assessment
- Security Architecture

Program Elements for Consideration

- Application and Information Inventory
- Define Methods for Security Due Diligence
- Perform Due Diligence on Affiliates/Business Partner Apps

Sample Decisions/Considerations

- Risk Ranking of Apps & Data
- Threat Models and Analysis
- Authentication, Access Control, Encryption, etc.
- Integration Requirements
- Software Framework / Design Principles
Application and Information Inventory

- Complete an inventory of all applications within the enterprise and the security controls / check points or each application
- Understand the business purposes of the application
- Identify the information/data that is flowing through these applications
- Start developing protection strategies and standards that will secure your organization’s most critical data
Defining Methods of Application Due Diligence

- Define corporate standards for testing applications
- Decide methods or approaches on appropriate level of due diligence for applications
- Methods of Due Diligence include the following:
  - Threat Modeling
  - Run-Time Analysis / Fault Injection Testing
  - Architecture/Design/Implementation Analysis
  - Static Analysis / Source Code Reviews
  - Database Vulnerability Testing
  - Host Configuration Reviews
  - Access Control Reviews
  - Software Development Lifecycle Reviews
  - Performance Load Testing
# Developing an Application Security Program - Verification

<table>
<thead>
<tr>
<th>SAMM Deployment Security Practices</th>
<th>Program Elements for Consideration</th>
<th>Sample Decisions/Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Design Analysis</td>
<td>- Outsourcing vs In-House</td>
<td>- Focus on Dev./ AppSec Resources</td>
</tr>
<tr>
<td>- Code Review</td>
<td>- Application Solutions &amp; Tools</td>
<td>- Periodic Testing/ Full App Assessment</td>
</tr>
<tr>
<td>- Remediation Procedures</td>
<td>- Automated Tools/Managed Services</td>
<td>- Run-Time Analysis/Code Reviews/Peer Review</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Investment for In-house Tools</td>
</tr>
</tbody>
</table>

**Governance**

**Construction**

**Verification**

**Deployment**

---

**INFORMATION ASSURANCE • SECURITY INTEGRATION • 24x7 SUPPORT • MANAGED SERVICES • TRAINING • STAFF AUGMENTATION • SECURITY TECHNOLOGY • INFRASTRUCTURE**
In-House vs. Outsourcing

- Unique skill set is required to test applications
- Necessary application development background
- Evaluate in-house staff
- Outsourcing application security – Gain knowledge / offset internal cost
- Application Managed Security Services
- Conduct cost/benefit analysis
- Keep separate testing/development in-house for PCI compliance
Remediation Procedures

- Develop remediation plans based upon results of due diligence testing (workflow capability in eGRC solutions)
- Review all associated risks, threats, vulnerabilities, and weaknesses discovered
- Organizations must assign the appropriate level of risk to the business and prioritize findings
- Business owners to decide if organization mitigates, transfers, or accepts risk
- Accountability is key
Developing an Application Security Program - Deployment

Program Elements for Consideration

- Prioritize Apps & Frequency of Testing
- Automated vs Manual Review
- Reporting & Documentation

Sample Decisions/Considerations

- Ongoing Testing Strategy
- Roles & Responsibilities for Ongoing Maintenances
- Additional Layers of Security & Controls Beyond App
- Change Management & Operational Controls
- Investment for In-house Tools
Prioritization of Application and Frequency of Testing

- Develop an evaluation process to determine which applications are most critical for testing
- Prioritize applications by setting a “risk” rating for each application
  - Not all applications are created equal!
- Assists in effective use of time, resources/staff, and budget
- Develop frequency standards on applications based on priority/criticality to business operations
Application Solutions and Tools

• Products and tools are NOT the “silver bullet”!
  – PCI suggests Security SDLC & Manual testing over dependence on WAF (info supplement)

• Solutions and tools by themselves cannot be relied upon

• Tools can bring tremendous value, but just do not forget people and processes

• Evaluate what, when, where, and how application security technologies can be utilized
Application Security Countermeasures

- Authentication
- Access Control
- Session Management
- Input Validation
- Error Handling
- Cryptography

- Cryptography
- Logging
- Monitoring & Alerting
- Change Management
- Incident Response
- Business Continuity

- Secure Data Storage & Transmission
OWASP 2010 Top 10

A1: Injection
A2: Cross-Site Scripting (XSS)
A3: Broken Authentication and Session Management
A4: Insecure Direct Object References
A5: Cross-Site Request Forgery (CSRF)
A6: Security Misconfiguration
A7: Insecure Cryptographic Storage
A8: Failure to Restrict URL Access
A9: Insufficient Transport Layer Protection
A10: Unvalidated Redirects and Forwards
OWASP Top 10: Mobile Risks

- Insecure Data Storage
- Weak Server Side Controls
- Insufficient Transport Layer Protection
- Client Side Injection
- Poor Authorization and Authentication
- Improper Session Handling
- Security Decisions Via Untrusted Inputs
- Side Channel Data Leakage
- Broken Cryptography
- Sensitive Information Disclosure

- Top 10 mobile controls and design principles
- GoatDroid (same concept as WebGoat)
Threat Modeling

- An exercise performed to provide valuable insight into urgent and/or significant risks related to the enterprise business application.
- The outcome of these exercises include the level of business impact and potential exposure to the client related to our findings (i.e. loss of revenue, loss of availability, confidentiality, integrity of data, and others).

Threat (Risk) Modeling:

**STRIDE** – Spoofing, Tampering Data, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege

**DREAD** – Damage Potential, Reproducibility, Exploitability, Affected Users, Discoverability

*Mobile OWASP (under development)
Application Firewalls

- Imperva
- F5 Networks
- Barracuda
- Citrix Netscaler
- Fortify

Benefits:
- Positive Security Model/Dynamic Profiling
- Adaptive Learning Engine
- Decrypts and inspects SSL Traffic
- High Availability (Active-Passive or Active-Active)
- F5 iControl® API provides integration between WhiteHat Security & F5 ASM

Challenges:
- Management and Maintenance
- Development of Custom Rules
- Protection from new adaptations of web application vulnerabilities
- Developers still writing insecure code
## What are businesses doing today?

<table>
<thead>
<tr>
<th></th>
<th>Client #1: Financial</th>
<th>Client #2: HealthCare</th>
<th>Client #3: Retail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meeting &amp; Maintaining Compliance</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Internal AppSec Standards</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Methods of AppSec Testing</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Establishing AppSec Owners/Sponsors</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Establishing IT Audit Involvement</td>
<td>n/a</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Outsource vs. In-House</td>
<td>Outsource</td>
<td>Outsource</td>
<td>In-House</td>
</tr>
<tr>
<td>Testing Affiliate Applications</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Frequency of Testing</td>
<td>2/year</td>
<td>1/year</td>
<td>1/year</td>
</tr>
<tr>
<td>Prioritizing Applications</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Training &amp; Staffing Requirements</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Application Solutions &amp; Tools</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Validation &amp; Remediation Procedures</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Application Security Roadmap

- Establish Application Security Committee (include Development Leads/PMs)
- Develop Strategic Plan for Addressing AppSec Risk
- Share Budgeting Dollars to Accomplish Goals
- Conduct an Application Inventory & Prioritize
- Integrate Security within the Software Development Lifecycle (SDLC)
- Raise Awareness & Educate: Executives, Developers, QA, and IT Audit
  - Develop High-Level Management Reports
- Collaboration between Security and Developer Groups (Remediation)
- Track Remediation Progress through Metrics
Application Security Key Resources

- Open Web Application Security Project (OWASP)
  - OWASP Top 10
  - OWASP Guide (Reviews the 300+ appsec issues)
  - OWASP Tools Project
  - http://www.owasp.org

- Web Application Security Consortium (WASC)
  - Web Security Threat Classification is a cooperative effort to clarify and organize the threats to the security of a web site.
  - http://www.webappsec.org/
Kathy Doolittle
Director, Strategic Services
D: 602.595.4146
M: 602.516.6438
kathy.doolittle@fishnetsecurity.com