Skeletons in the Closet: Securing Inherited Applications

John B. Dickson, CISSP

ISSA Washington, D.C
September 21, 2010
Key Questions for Today’s Session

• What applications represent the biggest risk?

• What attributes make them more or less risky?

• What are the most cost-effective courses of action given budget constraints in today’s business environment?
Key Goals for Today’s Session

• Understand risk-based options for managing the security of inherited applications
• Develop a framework for ranking risks with specific applications
• Understand some of the decision-making factors that come into play when risk-ranking applications
Personal Background

- 15-year information security consultant background
- Ex-Air Force security analyst at AFCERT
- Trident Data Systems, KPMG, SecureLogix, and Denim Group information security consultant
- Works with CIO’s and CSO’s to build successful software security initiatives
- Educates non-developer security professionals how to manage application risk
Denim Group Background

- *Professional services firm that builds & secures enterprise applications*

- *Secure development services:*
  - Secure .NET and Java application development
  - Post-assessment remediation
  - Secure web services

- *Application security services include:*
  - External application assessments
  - Code reviews
  - Software development lifecycle development (SDLC) consulting
  - Classroom and e-Learning instruction for developers
Background – the Current State of Affairs

- Creating meaningful enterprise-wide software security initiatives is hard
- The vast majority of info software security focuses on means to write more secure code or strategies for putting controls around the software development process
- Most organizations have hundreds or thousands of legacy applications that work!
  - They are viewed “part of the plumbing” by management
  - The code base can be millions of lines of code
Key Facts

• 66% have adopted a risk-based approach to remediation of application vulnerabilities
• 71% have an executive or team with primary ownership and accountability for application security
• 66% have defined communications channels between security, operations, and development teams

  – Source: “Securing Your Applications: Three Ways to Play,” Aberdeen Group, August 2010
Step 1 – Information Gathering

- Build a Portfolio of Applications
- Collect Background Information
  - Development Details
  - Vendor (if any)
  - Audience
  - Hosting Details
- Assess the Data
  - Type (CCs, PII, ePHI, etc)
  - Compliance Requirements
Step 1 – Information Gathering (Continued)

• Determine the Scale
  – Lines of Code
  – Dynamic Pages
  – Concurrent Users
  – User Roles

• Assess the Underlying Technology
  – Infrastructure (OS, hardware, etc)
  – Platform (.NET, Java, PHP, etc)
  – Versions

• Assess the Security State
  – Assessment Activity (type, date, etc)
  – Vulnerabilities (high, medium, low)
  – Protection (IDS/IPS, WAF)
Step 2 – Application Scoring

- Business Importance Risk
  - Business Function (customer interface, internal but public-facing, departmental use only)
  - Access Scope (external, internal)
  - Data Sensitivity (customer data, company confidential, public)
  - Availability Impact (serious, minor, minimal, or no reputation damage)
Step 2 – Application Scoring (Continued)

- **Technology Risk**
  - Authentication (*methods, enforcement*)
  - Data Classification (*formal approach or not*)
  - Input / Output Validation (*structured or not*)
  - Authorization Controls (*resource checks in place or not*)
  - Security Requirements (*explicitly documented or not*)
  - Sensitive Data Handling (*controls in place like encryption or not*)
  - User Identity Management (*procedures in place for account creation, access provisioning, and change control or not*)
  - Infrastructure Architecture (*network segmentation, patching*)
Step 2 – Application Scoring (Continued)

• Assessment Risk
  – Technical Assessment (assessment activity, vulnerabilities still present)
  – Regulatory Exposure (unknown, subject to regulation)
  – Third-Party Risks (outsourced development, SaaS hosting, etc)
Example Application Analysis

- Patient portal for hospital system
- Connects to back-end Electronic Medical Record system
- Microsoft.NET 3.5 framework
- Currently functionality being enhanced by internal development team
- Contains Electronic Patient (EPI) Data
- Audited once for PCI compliance in 2007
- Scanned by outside 3rd party for application security vulnerabilities in 2009
Application Comparisons
Conclusion

• Managing the security of inherited applications can present the most severe headaches for someone building a software security program.

• A risk-based approach is really the only economically feasible approach given the size/complexity of the problem.

• Understanding certain attributes of inherited applications is critical to applying a risk-based management approach.
So where do you go from here?
What you can do now!

• Collect or scrub your initial application inventory
• Develop relationships w/ 3rd parties who can help you through the identification
• Find a peer that is conducting the same risk ranking
• Exhaust Open Web Application Security Project (OWASP) resources!
• Familiarize yourself with OWASP OpenSAMM
Contact

John B. Dickson, CISSP
john@denimgroup.com
(210) 572-4400

www.denimgroup.com
blog.denimgroup.com
@johnbdickson