DENIM GROUP

build | integrate | secure

Skeletons in the Closet: Securing Inherited Applications

Baltimore ISSA

April 27, 2011

John B. Dickson, CISSP #4649

Overview for Today's Session

- The Problem
- Information Gathering
- Application Scoring
- Risk Rank & Tradeoff Analysis
- Discussion
- Conclusion, Next Steps, and Q&A

Some Key Questions for Today's Session

- Where do you start?
- What applications represent the biggest risk?
- What attributes make applications more or less risky?
- What are the most cost-effective way to manage the risk of inherited applications?
- What approaches might work for your organization?

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
- Apply one tactic from what you learn today next week at your organization

Personal Background

- 15-year information security consultant background
- Ex-Air Force SIGINT officer & security analyst at AFCERT
- Trident Data Systems, KPMG, SecureLogix, and Denim Group information security consultant
- Works with CIOs and CSOs to build successful software security initiatives
- Educates non-developer security professionals how to manage application risk

What you Don't know CAN Hurt You

- Passion: Get security professionals to ask a better set of questions
- Today's presentation focuses on helping you increase your IQ in the arena of software portfolio risk

Denim Group Background

- *Professional services firm that builds & secures enterprise applications*
- *Secure development services:*
  - Secure .NET and Java application development
  - Post-assessment remediation
- *Application security services include:*
  - External application assessments
    - Web, mobile, and cloud
  - Software development lifecycle development (SDLC) consulting
- *Classroom and e-Learning instruction for developers*
- *CRADA with the 688th Network Warfare Wing*

Denim Group Background

"CSRF Explained"

http://threadstrong.com/courses-csrf-explained.html

Background – the Current State of Affairs

- Creating meaningful enterprise-wide software security initiatives is hard
- The vast majority of info regarding software security focuses on writing more secure code or SDLC process improvement
- Most organizations have hundreds or thousands of legacy applications that work!
  - *They represent money already spent – ROI?*
  - *They are viewed "part of the plumbing" by management*
  - *The code base can be millions of lines of code*
- Focus on web applications
  - *Other software risks must be taken into consideration*
    - Web services, Saas, certain desktop applications

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

- 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

Application #1

- Publicly accessible staff scheduling application
- 1 million lines of code
- Written in classic ASP by an outsourced company this is no longer under contract
- <$1m year sales goes through the application in a $5b company
- No mitigating security technologies in place
- Still processes orders efficiently; supported by application maintenance group

Application Comparisons

Application #2

- External company website with limited functionality
- Site build it in Microsoft SharePoint 2007 technologies
- Custom web parts provide some interactive
- Site actively managed by corporate marketing team
- Not in scope for past outside security audit given marketing responsibility

Application Comparisons

Application #3

- J2EE-based corporate E-commerce site
- 500K lines of code
- Lots of revenue ($$$) for the company
- Regular security audits and scans by 3rd parties
- Web services to various internal applications

Application Comparisons

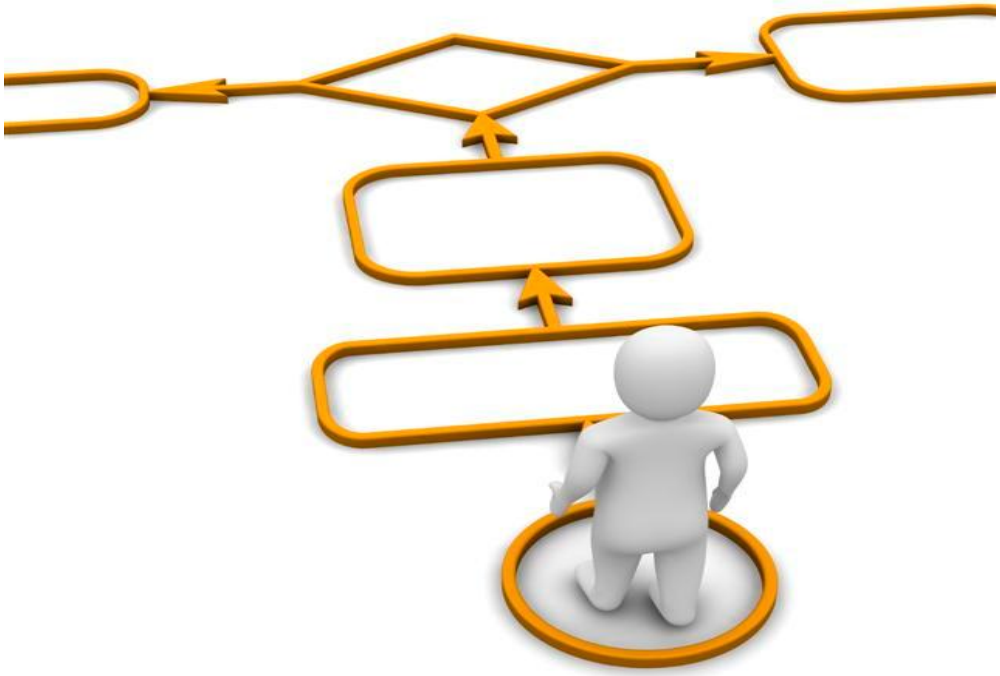
Application #4

- 3rd-party software-as-a-service (SaaS) CRM platform
- Sensitive client data included in database
- Functionality allows sales people to export data
- Managed by VP of Sales
- Software application “in the cloud”

Results Comparison

- Let's analyze our results
- Apply quantitative decision-making analysis concepts
  - *Want to understand what level of effort addresses the highest amount of risk*
- Tradeoff analysis

So where do you go from here?



Potential Follow-up Options

- End of Life
- Remediate
- Potential Testing Approaches
  - *Tailoring to Documented Risk*
  - *Work identified list from top to bottom*
- Application Security Verification Standard
  - *Levels of application-level security verification that increase in breadth and depth as one moves up the levels*
  - *Verification requirements that prescribe a unique white-list approach for security controls*
  - *Reporting requirements that ensure reports are sufficiently detailed to make verification repeatable, and to determine if the verification was accurate and complete.*

What you can do now!

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

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

Resources

- “Web Application Security Portfolios, *ISSA Journal*, May 2009, Coblentz, Nick.
- Open Web Application Security Project Open Software Assurance Maturity Model, www.owasp.org
- Open Web Application Security Project Application Security Verification Standard, www.owasp.org
- “How-to-Guide for Software Security Vulnerability Remediation,” Dan Cornell, Denim Group, October 2010
- Cloud Security Alliance
- “Securing your Applications,” Aberdeen Group, Brink, Derek, August 2010

Contact

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

www.denimgroup.com
blog.denimgroup.com
Twitter: @johnbdickson