Application Portfolio Risk Ranking
Banishing FUD With Structure and Numbers

Dan Cornell

OWASP AppSec DC 2010
November 11th, 2010
Overview

• 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?
Desired Outcomes

- 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

- Denim Group CTO
- Developer by background
  - Java, .NET, etc
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
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
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 testing software 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 codebases can be millions of lines of code

• Industry is focused on web applications
  – Other software risks must be taken into consideration
    • Web services, mobile applications, 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
Goal for Our Model

• Transparent – Decisions and calculations should be explainable
• Adaptable – Not every organization has the same drivers, goals or resources
• Practical – Get something that works and iterate
Methodology

• Steal steal steal!
  – Andrew Jacquith’s *Application Insecurity Index (AII) from his book Security Metrics*
  – Nick Coblentz’s blog posts on the topic
  – Other example spreadsheets, etc

• Simplify simplify simplify!
  – Great is the enemy of the good enough
  – Any information collected should provide value
  – Work in progress

• Test with organizations

• Repeat
Step 1: Develop Initial Criteria

- Business Importance Risk
- Assessment Risk
Step 2 – Information Gathering

• Build a Portfolio of Applications
  – Public-facing web sites
  – Customer-facing web applications
  – Partner-facing web applications
  – Internal- or partner-facing web services
  – Customer Relationship Management (CRM) systems
  – Financial applications
  – “Green screen” mainframe applications
  – Software as a Service (SaaS) applications
Step 2 – Information Gathering (Continued)

- Collect Background Information
  - Development Details
  - Vendor (if any)
  - Audience
  - Hosting Details

- Assess the Data
  - Type (CCs, PII, ePHI, etc)
  - Compliance Requirements

- Determine the Scale
  - Lines of Code
  - Dynamic Pages
  - Concurrent Users
  - User Roles
Step 2 – Information Gathering (Continued)

• 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 3 – 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 3 – 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 3 – 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)*
Step 4: Determine Assessment Approach

- Currently using OWASP Application Security Verification Standard (ASVS)
- Determine what you consider to be a Critical, High, Medium, Low
- Determine what assessment approach/standard you want to use
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
Evaluation

• Pros
  – Provides for a structured approach
  – Calculations are observable
  – Standards can be set for specific organizations

• Cons
  – Can seem like a lot of data to collect
  – Technology Risk is hard to get at a proper level of granularity
  – Excel spreadsheet combines data and code
  – Needs work for dealing with “cloud” stuff
So where do you go from here?
Example Artifacts

• Application Tracking and Risk-Ranking Spreadsheet
  – What are the applications?
  – What are their characteristics?
  – How do they rank against one another?

• Risk-Ranking Planning Spreadsheet
  – Which applications are critical, high, medium or low?
  – How are you going to deal with each application?
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 (ASVS)
  – 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 with 3rd parties who can help you through the identification process
- Find a peer that is conducting the same risk ranking
- Familiarize yourself with OWASP OpenSAMM and OWASP ASVS
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

- Cloud Security Alliance
- “Securing your Applications,” Aberdeen Group, Brink, Derek, August 2010
Contact

Dan Cornell
dan@denimgroup.com
(210) 572-4400

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

Email me for a copy of the example Excel spreadsheet