

Software Assurance Forum March 2009

Multi-perspective Application Security Risk Analysis: A Toolbox Approach

Sean Barnum
Cigital Federal, Inc.
sbarnum@cigital.com



Today's Landscape

- Our entire lives today depend on technology (unless you're the Unabomber)
 - Finance
 - Power
 - Food
 - Communication
 - Travel/transport
 - Defense
 - Trade
 - Internet

Cybersecurity as a Growing Concern

- Technology is under constant attack
- Intelligent criminals no longer rob banks for thousands when they can hack banks for millions with a low chance of being caught
- Cyber attack now driven by money and ideology rather than ego and mischief
- It is now a question of **when**, not **if**, a piece of technology will be attacked
- It is impossible to be 100% secure
- Cybersecurity is a matter of **risk management**

Software Security as a Primary Element of Cybersecurity

- Software is the target of the vast majority of attacks
- 75% of attacks at Application Layer (Gartner)
- XSS and SQL Injection are #1 and #2 reported vulnerabilities (Mitre)
- 90% of sites are vulnerable to application attacks (Watchfire)
- 78% of easily exploitable vulnerabilities affected Web applications (Symantec)
- 80% of organizations will experience an application security incident by 2010 (Gartner)

Reality Recap

- Security issues are becoming increasingly critical to organizations
- More and more enterprises are becoming aware of the importance of software assurance as an element of their broader security focus
- This awareness typically comes from one of three sources:
 - The exploitation and breach of an individual fielded application
 - An external mandate from senior management or an external governing entity that the issue must be addressed
 - Internal epiphany or evolution of understanding

Typical Reactions to Software Assurance Awareness

- When an awareness is reached by an organization, one of several responses is usually taken:
 - Ignore the problem (aka head in the sand)
 - Undertake a paper exercise of policy and process that ultimately has no direct effect on the security of the software (aka lipstick on a pig)
 - Assess and remediate the individual exploited application (aka band-aid)
 - Seek to address the root problems by investigation and adoption of individual tactical application security practices such as penetration testing, static code analysis, security testing, etc (aka treating individual symptoms)
 - Address the issue comprehensively through strategic thought and action (aka treating the disease)

Key Role of Application Security Risk Analysis in the Cybersecurity Game

- Ultimate goal is to prevent security vulnerabilities from ever entering software
- Reality is they are already there and even new code from security-aware developers needs to be checked
- Application security risk analysis is the practice of:
 - checking software for weaknesses/vulnerabilities
 - characterizing the risk they pose
 - identifying and prioritizing mitigations

Varying Perspectives of Analysis

- static source code
- static binary code
- dynamic application scanning
- application penetration testing
- application data security
- fuzzing
- complexity
- composition & pedigree
- etc.

Varying Capabilities of Analysis Perspectives

- Different perspectives are effective at finding different types of weaknesses
- Some are good at finding the cause and some at finding the effect

	Static Code Analysis	Penetration Test	Data Security Analysis	Code Review	Architecture Risk Analysis
Cross-Site Scripting (XSS)	X	X		X	
SQL Injection	X	X		X	
Insufficient Authorization Controls		X	X	X	X
Broken Authentication and Session Management		X	X	X	X
Information Leakage		X	X		X
Improper Error Handling	X				
Insecure Use of Cryptography		X		X	X
Cross Site Request Forgery (CSRF)		X		X	
Denial of Service	X	X	X		X
Poor Coding Practices	X			X	

Automating Analysis Perspectives

- Automation should be leveraged wherever possible but should be combined with focused manual analysis
- Automated tools will find the low-hanging fruit much faster than manual analysis can
- Manual analysis will find less obvious and occasionally high-risk issues

Current State of the Practice

- Most organizations undertaking application security risk analysis only perform one or maybe two analysis perspectives and those are done as independent processes often by separate teams
 - If developer-centric organization, typically start with static analysis
 - If test-centric, typically start with application scanning and penetration testing
 - If information assurance or data-centric, typically start with data security scanning

The Gestalt of Multi-perspective Analysis

- Better situational awareness
- Reinforce confidence in findings of each perspective
- Combine the assurance of dynamic analysis with the detail of structure analysis to plan effective mitigation of high-criticality risk

The Challenges of Integrated Multi-perspective Analysis

- Varying perspectives have different drivers and priorities based on context
- Differing perspectives treat “location” of issue differently making correlation a challenge
- Each tool for each perspective has its own reporting schema
 - Need for a unified findings schema

The Need for Standards in Effective Integration

- Always make sure comparing apples to apples
- Weakness
 - Common Weakness Enumeration (CWE)
- Attack
 - Common Attack Pattern Enumeration and Classification (CAPEC)
- Vulnerability
 - Common Vulnerabilities and Exposures (CVE)
- Technical Context
 - Common Platform Enumeration (CPE)
- Mitigation
 - Common Control Enumeration (CCE)

A Recommended Baseline for Multi-perspective Analysis

- To effectively assess the security risk of an application, an assessment methodology should at a minimum include the following perspectives:
 - Static source code analysis
 - Application scanning & penetration testing
 - Application data security analysis

Static Source Code Analysis

- Analyze code without executing it
- Strengths
 - Fast compared to manual code review
 - Fast compared to testing
 - Complete, consistent coverage of source code (all paths)
 - Brings security knowledge with it
- Limitations
 - Only analyzes the source code you feed it
 - Doesn't find everything
 - Architecture errors
 - Bugs you're not looking for
 - System administration mistakes
 - User mistakes
 - False positives
- Multi-perspective integration value
 - Actual location of the weakness in code
 - Identify issues to target with penetration testing
 - Identify co-influencing weaknesses within relevant contexts

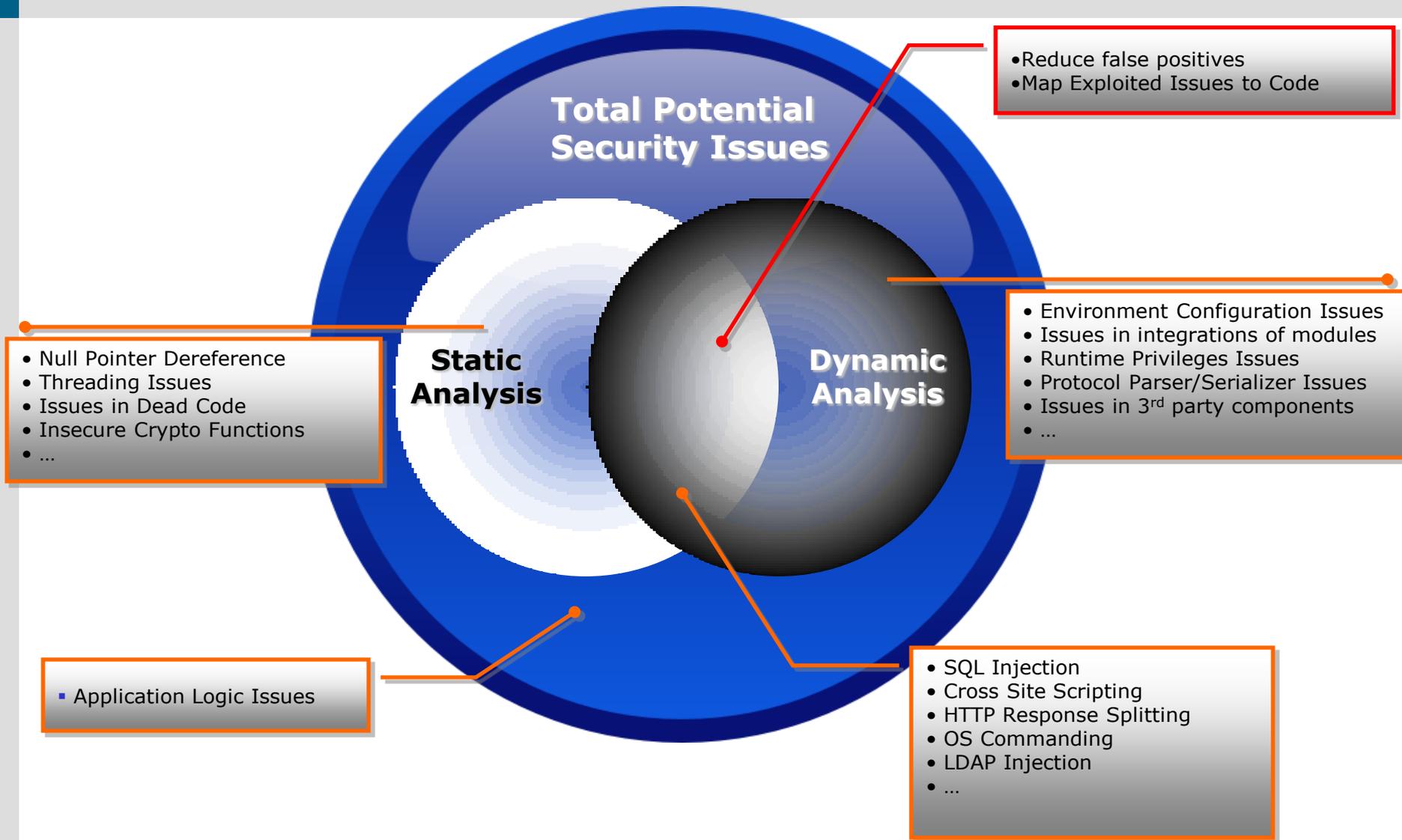
Application Scanning & Penetration Testing

- Security testing (black box) of applications through simulated attacks
- Strengths
 - Simulates the actual risk (attacker's action)
 - Tests full software stack
 - Low false positives
 - Mature technology
- Limitations
 - Only as good as what you scan (crawling limitations)
 - Analysis limited to the test cases executed
 - Must run tests often to stay protected
 - Can only be performed once code is 'runable'
 - Risky to run on production applications
 - Cannot identify the actual source of the problem, only the symptom
- Multi-perspective integration value
 - Confirming that weaknesses are vulnerable
 - Mapping penetration scans to locations in source code
 - Mapping data security findings to injection findings, privilege issues, etc.

Application Data Security Analysis

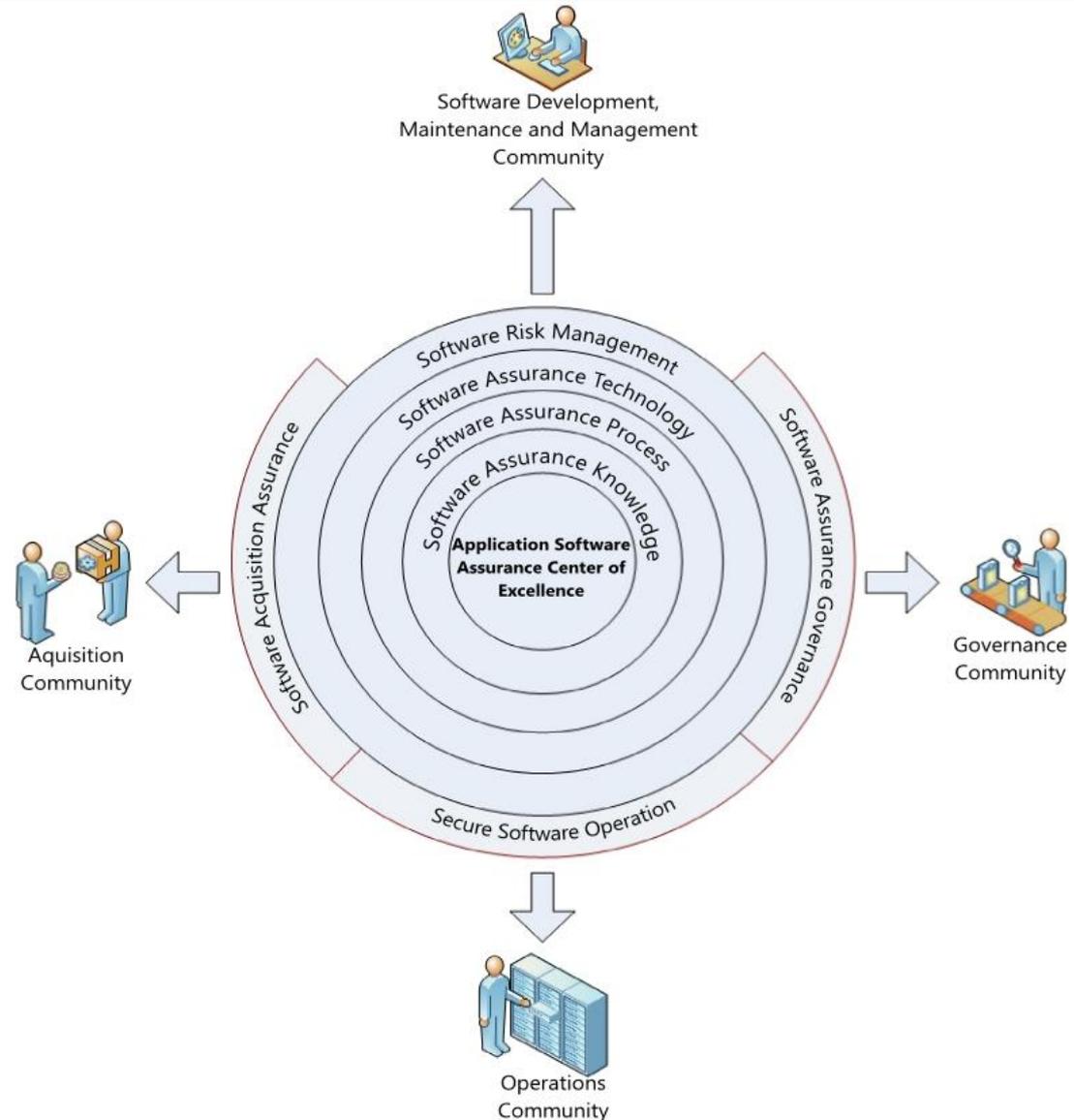
- Analyzing the security concerns of how an application accesses and manages its database
- Strengths
 - Analyzes a live, fully configured system rather than just source code
 - Good at catching really bonehead mistakes (they are more common than you think)
 - Helps mitigate both insider and external threats
- Limitations
 - Only as good as what you tell it to look for
 - Does not understand semantics of data (can use limited proxies)
- Multi-perspective integration value
 - Confirmation of likely weaknesses as vulnerabilities
 - Better contextual info about nature and severity of weaknesses
 - Improved understanding of likelihood of weaknesses being exploitable
 - Increases accuracy of forensic data
 - Improved data flow policies
 - Improved Access Control

Value of Aligning Multiple Perspectives



Practical Example: USAF ASACoE

- Application Software Assurance Center of Excellence (ASACoE)
- *The Focal Point for Air Force Software Assurance (SwA) capability with the goal of reducing software-induced risk from Air Force applications.*



Overview of Triage Assessment Process

- Establish buildable source code and executable test or operational environment
- **Run static source code analysis scan**
- **Run web application scan**
- **Run application data security scan**
- **Prioritize results analysis**
- **Eliminate obvious false positives**
- **Correlate results of different tools to confirm vulnerabilities or eliminate false positives**
- Conduct remaining analysis
- Characterize and classify findings
- Create integrated findings report
- Adorn integrated report with mitigation advice for findings

ASACoE Rationale for Multi-perspective Approach

- Air Force is looking to maximize its understanding of security risk in all areas of its applications (interfaces, business logic, data tier, etc.)
- ASACoE recognizes the difficulty and complexity of analyzing application security tool scan results
- ASACoE wants to provide as much context and guidance as possible to developers for mitigation and remediation

Summary and Conclusions

- Software Assurance analysis is increasingly becoming a high priority and is maturing in its capability
- Varying perspectives of analysis are available, each with their own unique value
- Blending multiple perspectives together yields better overall coverage and an integrated gestalt
- It is real and possible to begin pursuing this approach today