HS SEDI
Homeland Security Systems Engineering and Development Institute

Multi-perspective Automated Analysis

Robert A. Martin
Sean Barnum

May 2011

Homeland Security

Agenda

8:00-8:45am	Software Security Knowledge about Applications Weaknesses
9:00-9:45am	Software Security Knowledge about Attack Patterns Against Applications
	Training in Software Security
10:15-11:00am	Software Security Practice
11:15-12:00am	Supporting Capabilities
	Assurance Cases
	Secure Development & Secure Operations

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

The HS SEDI FFRDC is managed and operated by The MITRE Corporation for DHS.

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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- 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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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 (SAFES)

The Need for Standards in Effective Integration

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **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

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

HS SEDI
Homeland Security Systems Engineering and Development Institute

Taming the Tower of Babel: Software Assurance Findings Expression Schema (SAFES)

Sean Barnum
MITRE

Homeland Security

Today's Challenge

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **There is no standard reporting format for SwA analysis**
  - Very difficult to combine results of multi-perspective analysis
  - Very difficult to combine results of multi-tool analysis
  - Very inefficient for tool vendors looking to integrate results with other tools (very costly and redundant)
  - Very difficult to trend across assessments from different tools or analysts
  - Very difficult to automate meta-analysis and the assessment process

SAFES Effort

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **Software Assurance Findings Expression Schema (SAFES)**
- **Sponsored by the NSA Center for Assured Software (CAS)**
- **Objectives:**
  - Enable and encourage consistency in software assurance tool, service and analysis practice findings
  - Establish more structured and effectively useful software assurance tool, service and analysis practice results
  - Enable integration of results from multiple software assurance tools, services or analysis practices
  - Enable automated processing of software assurance tool, service or analysis practice results

SAFES Approach

HS SEDI
Homeland Security Systems Engineering and Development Institute

Phase 1

- Community collaboration
- Build from state of the practice (considered ~20 tools & services)
- Enhance with state of the art
- Define a comprehensive schema covering all aspects of software assurance analysis reporting

Phase 2

- Enable & demonstrate practical use

Future

- Continually refine for coverage, consistency & efficiency
- Layer the schema into a framework for composable and focused use
- Strive for flexibility and extensibility
- Mature towards formalization

SAFES Initial Scope

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **In-scope perspectives for initial effort:**
  - Static source code analysis
  - Static binary code analysis
  - Web application penetration testing
  - Data security analysis
  - Fuzzing
  - Threat modeling
  - Architectural risk analysis
- **Some vendors actively collaborating others were passively incorporated**

HS SEDI
Homeland Security Systems Engineering and Development Institute

SAFES is a comprehensive and detailed schema

- **Info on findings**
  - Description
  - Categorization
  - Location
  - Prioritization
  - Correlations
- **Info on analysis approach**
  - Tool or service
  - Methodology
  - Detection mechanisms

- **Info on mitigation**
- **Info on meta-analysis**
- **Info on personnel**
- **Info on application**
  - Structure, content & configuration
  - Business/mission and security context
- **Info on assurance case**
- **Info on threat analysis**

Key Constructs

- Sub-Assessment scopes
- Traces
- Report views
- Assurance case
- Finding prioritization
- Tool-Service info
- Findings correlations

A Sampling of Potential Use Cases

HS SEDI
Homeland Security Systems Engineering and Development Institute

- Understand the Business Context of application
- Identify risks
- Map technical risks to business context
- Map the application attack surface
- Identify relevant threats
- Inventory and characterize assets
- Create threat model
- Define FISMA security categorization (FIPS-199)
- FISMA Security Planning (SP800-18)
- FISMA Risk Assessment (SP800-30)
- Conduct multi-tool/multi-perspective analysis
- Identify false positives
- Characterize risk
- Prioritize risk
- Correlate findings
- Stitch dynamic & static location results
- Integrate automated and manual analysis
- Reuse common mitigation advice
- Create assessment report
- Create different versions of report
- Define an assurance case for an application
- Create an assurance case compliance report
- Import CWE content into local context
- Identify common finding trends across portfolio by technology context
- Maintain analysis accountability
- Identify trends in tool and rule efficacy
- Mapping between various tool level definitions

SAFES Maturation Paths

HS SEDI
Homeland Security Systems Engineering and Development Institute

- **Usability: primarily focused on efforts surrounding the schema to make it more usable by the community such as native transforms, tooling, etc.**
- **Refinement: primarily focused on improving the quality and coverage of the schema itself with activities such as adding new perspectives, adding new schemas, fixing errors, etc.**
- **Formalization: primarily focused on gradually (as quickly as is prudent and accepted by the targeted user community) incorporating in formal standards-based approaches (vocabulary, structure, etc.) and working towards handoff of development to an appropriate community standards consortium body**

SAFES Phase 2

- Develop 5-10 transforms from native tool output to SAFES (currently for CAS internal use but hopefully will eventually be shared)
- Develop a demonstrative use case example for SAFES
- Develop lightweight initial prototype authoring/editing/reporting tools (very, very simple)
- Develop a real, permanent website as part of MSM
- Coordinate with standards organizations for planning towards future maturation and formalization

SAFES Next Steps Beyond Phase 2

HS SEDI
Homeland Security Systems Engineering and Development Institute

- Identify & support real-world prototype usage of SAFES
- Refine based on feedback
- Refine & extend authoring/editing/reporting tools with the goal of eventually transferring this work to other parties (vendors, open-source projects, consortia, etc.)
- Incorporate coverage for more tools, services & analysis practices
- Work with vendors (and OS projects) to develop more native transforms and encourage native output of SAFES
- Refine for efficiency
- Refine for flexibility (framework layering)
- Refine for formalization towards existing standards

HS SEDI
Homeland Security Systems Engineering and Development Institute

Questions?

Sean Barnum
MITRE
sbarnum@mitre.org