CYBER SECURITY DIVISION
2013 PRINCIPAL INVESTIGATORS’

Code Pulse: Dynamic Augmented Static Analysis

Secure Decisions
Hassan Radwan

Sept. 16th, 2013
Secure Decisions

We help you **make sense of data**
- Analyze security *decision-making* processes
- Build *visual analytics* to enhance security decisions and training

Our expertise starts where automated security sensors and scanners leave off

We **transition** our R&D into **operational use**, in government and industry

Grounded in commercial software and product development
- Division of Applied Visions, developer of commercial software
- 40 people, most with clearances, and secure facilities
Secure Decisions SwA Tools

**Code Dx**
DHS funded Phase II SBIR
Currently TRL 7

**Code Pulse**
DHS CSD funded BAA
20 month project, PoP through April ‘14

Converged static source analysis

Dynamic augmented static source analysis
Typical Static Analysis Workflow

1. Static analysis tools process source code
   Many potential vulnerabilities found

2. Security Analyst/Auditor processes and prioritizes the list
   Short list of top issues

3. Development team processes the list fixing some, ignoring others
Need: Relevant Vulnerabilities

- Full Application Source
- Code Executed Source
- Attack Surface
- Relevant Vulnerabilities
- Short List
Need: Remediation Context

Programming context recall decay

Recall Ability

Time

Developers fixing a bug/vulnerability when the source context is still fresh will be quicker and less error-prone

As time progresses their speed and ability to ensure system integrity with new changes diminishes
Terminology

**Static Analysis**
- At the source or binary levels
- Scans to detect potential vulnerabilities
- No runtime context

**Dynamic Analysis**
- In SwA used to describe the process to detect potential vulnerabilities at runtime (A.K.A. Penetration Testing, Black Box Testing)
- Source code context not available

**Dynamic Tracing**
- Monitor runtime execution
- Used by profiling tools
- Identify which methods are called and when to observe: call graph, call durations, and call frequencies
Code Pulse Scope

Static source analysis

+ = Code Pulse

Dynamic tracing

for Java
Approach

Static analysis tools

Dynamic augmented static source analysis

Vulnerability analysis in execution context

Visual analytics

DYNAMIC TRACES

Static & dynamic fusion

Dynamic augmented static source analysis
Use Cases

3 primary high-level use cases

- Dynamic trace collection
- Prioritization
- Remediation
Focus the prioritization and remediation of static software analysis.
Competition

• Dynamic application security testing
  – Black box testing
  – Little insight into the internals of the target software

• Hybrid application security testing
  – Combination of static and dynamic testing
  – Better security coverage
  – **But** static analysis results still need to be processed using traditional techniques without runtime context

• No existing solution for automated runtime correlation with static analysis results
Task Overview

Investigate tools
UX design

Develop Prototype
Tracing data
Prioritization use case
Remediation use case

Test, demonstrate, and evaluate
Alpha
Beta

Prepare for technology transition
Current Status

- Investigate tools
- UX design
- Develop Prototype
- Tracing data
- Prioritization use case
- Remediation use case
- Test, demonstrate, and evaluate
- Alpha
- Beta
- Prepare for technology transition

Status:
- Not yet started
- In progress
- Completed
Next Steps

Investigate tools  UX design  Develop Prototype

Tracing data  Prioritization use case  Remediation use case

Current Focus
Wrap-up prioritization use case
Alpha – gather initial user feedback

In the next 8 Months
Remediation use case
Beta – improve usability and performance
Tech transition

Prepare for technology transition

Code Dx Integration  SWAMP Integration
Contact Information

Hassan Radwan
hassan.radwan@securedecisions.com
518-207-3106