A Policy-Based Vulnerability Analysis Framework

Sophie J. Engle
sjengle@ucdavis.edu
Framework Goals

Build a repeatable and practical framework for vulnerability analysis
Framework Goals

Build a **repeatable** and practical framework for vulnerability analysis

– Theoretical foundation
Framework Goals

Build a repeatable and practical framework for vulnerability analysis
– Theoretical foundation
– Practical levels of abstraction
Terminology Overview

- **Ideal Policy Oracle**
- **Feasible Policy Oracle**
- **Configured Policy Oracle**
- **Instantiated Policy Oracle**

#### Vulnerability Hierarchy
- **Inherent Vulnerability**
- **Configuration Vulnerability**
- **Implementation Vulnerability**

#### Policy Hierarchy
- **Preconditions**
- **Policy Violations**

#### Vulnerability Model
- **Characteristics**
- **Symptoms**

#### Vulnerability Classification
- **IVAB / IVEC**

#### Vulnerability Analysis
Talk Outline

• Section 1: Security Policy
• Section 2: Vulnerability Hierarchy
• Section 3: Vulnerability Model
• Section 4: Vulnerability Classification
• Section 5: Vulnerability Analysis
Security Policy

Section 1
Terminology

• Policy Event
  – \( E = ( \text{subject, object, action, boolean condition} ) \)

• Global Policy Event Space
  – Universe of policy events \( E = S \times O \times A \times B \)

• Policy Oracle
  – Oracle function \( \mathcal{P}( E ) = \{ \text{yes, no, unknown} \} \)
Policy Hierarchy

- Ideal Policy Oracle
  - Which policy events should be authorized (ideally)

\[ P_{id}(Xander, \text{control room, enter, true}) = \text{yes} \]
Policy Hierarchy

• Ideal Policy Oracle
  – Which policy events should be authorized (ideally)

• Feasible Policy Oracle
  – Which policy events are authorized (realistically)

\[ P_{fe}( \text{bid:14, room:21, enter, true } ) = \text{yes} \]
Policy Hierarchy

• Ideal Policy Oracle
  – Which policy events *should be authorized* (ideally)

• Feasible Policy Oracle
  – Which policy events *are authorized* (realistically)

• Configured Policy Oracle
  – Which policy events *are allowed* (by configuration)

\[ P_{co}(\text{bid:14, room:21, enter, true}) = \text{no} \]
Policy Hierarchy

• Ideal Policy Oracle
  – Which policy events should be authorized (ideally)

• Feasible Policy Oracle
  – Which policy events are authorized (realistically)

• Configured Policy Oracle
  – Which policy events are allowed (by configuration)

• Instantiated Policy Oracle
  – Which policy events are possible (by implementation)

\[ P_{in}(\text{bid:14, room:21, enter, true}) = yes \]
Policy Hierarchy

- Policy violations occur between oracles
  - $P_{id}(\text{Xander, control room, enter, true}) = yes$
  - $P_{fe}(\text{bid:14, room:21, enter, true}) = yes$
  - $P_{co}(\text{bid:14, room:21, enter, true}) = no$
  - $P_{in}(\text{bid:14, room:21, enter, true}) = yes$
Terminology Overview

- Ideal Policy Oracle
- Feasible Policy Oracle
- Configured Policy Oracle
- Instantiated Policy Oracle

Policy Hierarchy

- Inherent Vulnerability
- Configuration Vulnerability
- Implementation Vulnerability

Vulnerability Hierarchy

- Preconditions
- Policy Violations

Vulnerability Model

- Characteristics
- Symptoms

Vulnerability Classification

- IVAB / IVEC

Vulnerability Analysis
Vulnerability Hierarchy

- A *vulnerability* is the set of conditions that enable an unequivocal policy violation.

```
Ideal Policy Oracle

Feasible Policy Oracle

Configured Policy Oracle

Instantiated Policy Oracle

Inherent Vulnerability

Configuration Vulnerability

Implementation Vulnerability
```
Inherent Vulnerabilities

- Result of intentional compromises
- Indicates where functionality, configuration, manageability, or usability may be improved
Configuration Vulnerabilities

- Indicates that the policy as configured is incorrect
- Caused by difficult to configure or maintain security mechanisms, or poorly articulated policies
Implementation Vulnerabilities

- Captures the traditional notion of a vulnerability
- Indicates that the mechanism’s implementation does not properly enforce the policy
Vulnerability Model

Section 3
Terminology Overview

- Ideal Policy Oracle
- Feasible Policy Oracle
- Configured Policy Oracle
- Instantiated Policy Oracle
- Inherent Vulnerability
- Configuration Vulnerability
- Implementation Vulnerability
- Preconditions
- Policy Violations
- Characteristics
- Symptoms
- IVAB / IVEC

Policy Hierarchy
Vulnerability Hierarchy
Vulnerability Model
Vulnerability Classification
Vulnerability Analysis
Terminology

• Security Policy
  – Traditionally defined as a *partition of states*
  – Instead define as a *language of configurations*

  Example: State $q_i$ is authorized if $w$ is on the tape.

• Policy as a partition:
  – Must design TM and split $q_i$ into two states

• Policy as a configuration:
  – $\{ uq_i.v : u \circ v \equiv w \}$
Terminology

• Policy Violation
  – A configuration that is either valid but unauthorized, or authorized but invalid

• Precondition
  – A language of configurations describing trace prior to the policy violation

• Implementation Vulnerability
  – A policy violation and its associated preconditions
Vulnerability Classification

Section 4
Terminology Overview

- Ideal Policy Oracle
- Feasible Policy Oracle
- Configured Policy Oracle
- Instantiated Policy Oracle

- Configuration Vulnerability
- Implementation Vulnerability
- Inherent Vulnerability

- Preconditions
- Policy Violations
- Characteristics
- Symptoms

- IVAB / IVEC

Policy Hierarchy
Vulnerability Hierarchy
Vulnerability Model
Vulnerability Classification
Vulnerability Analysis
Perfect Knowledge Assumption

- Why is our formal model impractical?
  - Do not have the formal specification
  - Do not have access to computation trace
  - Do not have an explicit set of systems
Perfect Knowledge Assumption

• Why is our formal model impractical?
  – Do not have the formal specification
  – Do not have access to computation trace
  – Do not have an explicit set of systems

• End result:
  – Defining a precondition is impractical
  – Defining a policy violation is impractical
  – *Defining an implementation vulnerability is impractical*
Vulnerability Abstraction

• Characteristic
  – A set of similar known preconditions
  – Example: \( X_{null} = \{ t : t \text{ contains the null character } \backslash 0 \} \)

• Symptom
  – A set of similar known policy violations
  – Example: \( Y_{incr} = \{ u : \text{VALID}(M) \setminus L(P) \} \)
    i.e. \( u \) is a valid configuration, but not authorized by policy
Vulnerability Abstraction

• Implementation Vulnerability: $V = (U, T)$
  – $T$ is the set of policy violations
  – $U$ is the set of associated preconditions
Vulnerability Abstraction

• Implementation Vulnerability: $V = (U, T)$
  - $T$ is the set of policy violations
  - $U$ is the set of associated preconditions

• Vulnerability Abstraction (IVAB): $Z = (X, Y)$
  - $X$ is the basic characteristic set for $U$
  - $Y$ is the basic symptom set for $T$
Vulnerability Abstraction

- **Implementation Vulnerability:** \( V = (\ U,\ T ) \)
  - \( T \) is the set of policy violations
  - \( U \) is the set of associated preconditions

- **Vulnerability Abstraction (IVAB):** \( Z = (\ X,\ Y ) \)
  - \( X \) is the basic characteristic set for \( U \)
  - \( Y \) is the basic symptom set for \( T \)

- **Equivalence Class (IVEC):** \( Z = (\ X,\ Y ) \)
  - The set of equivalent IVABs
Vulnerability Classification

- Master Classification Tree
  - Characteristic Classification Tree
  - Symptom Classification Tree

- Vulnerability Classification Tree
Buffer Overflow
Characteristic Tree
Direct Executable Buffer Overflow
Direct Executable
Buffer Overflow
Direct Executable Buffer Overflow
Vulnerability Analysis

Section 5
Terminology Overview

Policy Oracle

Ideal Policy Oracle

Feasible Policy Oracle

Configured Policy Oracle

Instantiated Policy Oracle

Vulnerability

Inherent Vulnerability

Configuration Vulnerability

Implementation Vulnerability

Preconditions

Policy Violations

Characteristics

Symptoms

IVAB / IVEC

Vulnerability Analysis
Analysis Goals

• Shift focus from *if* a system is secure to *when* a system is secure

• Locate and mitigate implementation vulnerability (equivalence classes) via characteristic-based analysis
Analysis Overview

• Phase 1: Preparation
  – Define global policy event space
  – Approximate configured oracle

• Phase 2: Analysis
  – Approximate instantiated oracle
  – Identify confirmed IVECs and characteristics

• Phase 3: Mitigation
  – Identify target characteristics
  – Disable target characteristics
Analysis Overview

phase 1: preparation

start

- determine analysis scope
- approximate configured oracle

refine scope

phase 2: analysis

- IVECs (Confirmed)

phase 3: mitigation

- vulnerability mitigation
- mitigate target characteristics
- identify target characteristics

end

if unmitigated IVECs < threshold, continue loop
Phase 2 Analysis

• Characteristic Analysis
  – Develops set of suspected characteristics

• Environment Analysis
  – Determines if suspected characteristics exist

• Vulnerability Analysis
  – Develops set of suspected IVECs

• Instantiated Oracle Analysis
  – Determines if suspected IVECs exist
Phase 2 Overview

characteristic analysis

environment analysis

instantiated oracle analysis

vulnerability analysis

confirmed IVECs

suspected IVECs

scope IVECs

suspected characteristics

confirmed characteristics

external expertise

step 2

step 3

step 1

step 4
Phase 3 Mitigation

- Identify target characteristics
  - Frequent, i.e. associated with most IVECS
  - Dangerous, i.e. associated with worst symptoms

- Disable target characteristics
  - Some may be impossible or infeasible to fully disable

- Mitigate vulnerabilities
  - Compare confirmed IVECs with disabled characteristics
  - Update set of confirmed IVECs
Phase 3 Overview

Phase 1: preparation
- Start
- Determine analysis scope
- Approximate configured oracle
- Refine scope

Phase 2: analysis

Phase 3: mitigation
- Vulnerability mitigation
- Mitigate target characteristics
- Identify target characteristics

If unmitigated IVECs < threshold, continue loop
Conclusion
Terminology Recap

- Ideal Policy Oracle
- Feasible Policy Oracle
- Configured Policy Oracle
- Instantiated Policy Oracle
- Inherent Vulnerability
- Configuration Vulnerability
- Implementation Vulnerability
- Preconditions
- Policy Violations
- Characteristics
- Symptoms
- IVAB / IVEC

Policy Hierarchy
Vulnerability Hierarchy
Vulnerability Model
Vulnerability Classification
Vulnerability Analysis
Framework Recap

phase 1: preparation
- start
  - determine analysis scope
  - approximate configured oracle
  - refine scope

phase 2: analysis
- IVECs (Confirmed)

phase 3: mitigation
- vulnerability mitigation
- mitigate target characteristics
- identify target characteristics

if unmitigated IVECs < threshold, continue loop
Contributions

- Policy-Based Vulnerability Hierarchy
  - Can incorporate both security procedures and security mechanisms
  - Captures high-level and low-level vulnerabilities

- Formal Implementation Vulnerability Model
  - Policy as a language of configurations, instead of a partition of states
  - Theoretical foundation for classification scheme
Contributions

• Characteristic-Based Vulnerability Classification
  – Makes “perfect knowledge assumption” explicit
  – Provides reversible layers of abstraction

• Policy-Based Vulnerability Analysis Framework
  – Capable of repeatable vulnerability analysis results
  – Practical for stable, small-scale environments
Future Work

• Theoretical Results
  – Decidability of different security problems

• Vulnerability Database
  – Characteristic-based classification
  – Classification versus clustering

• Extended Case Study
  – Hypothetical electronic voting environment
Extended Case Study

• Four Analysis Teams
  – Environment: *Develops hypothetical environment*
  – Alpha: *Performs analysis using framework*
  – Beta: *Performs analysis using framework*
  – Control: *Performs ad-hoc analysis*

• Compare Results
  – Number of vulnerabilities found
  – Consistency of results across teams
Questions?
General Information

• Dissertation:

• Committee:
  – Professor Matt Bishop (Chair)
  – Professor S. Felix Wu
  – Professor Karl Levitt
  – Professor Sean Peisert
Selected References

• **Vulnerability Analysis: An Extended Abstract**

• **We Have Met the Enemy and He is Us**

• **A Taxonomy of Buffer Overflow Preconditions**

• **The Unifying Policy Hierarchy Model**

• **Protocol Vulnerability Analysis**
Contact Information

Sophie Engle
sjengle@ucdavis.edu
Insider Threat Case Study

Supplemental Slides
Insider Threat Case Study

• Demonstrates vulnerability analysis using the Policy-Based Vulnerability Hierarchy

• Insider threat exists whenever:
  – Someone has more privileges at a lower policy level than at a higher policy level
  – The “insiderness” captures number of extra privileges

• Focus on identifying potential for misuse of privileges, not potential for abuse of any particular user
Insider Threat Case Study

• Two Primary Phases:
  – Inherent vulnerability analysis, such that $\mathcal{P}_{fe}(E) = yes$ and $\mathcal{P}_{id}(E) = no$
  – Absolute vulnerability analysis, such that $\mathcal{P}_{in}(E) = yes$ and $\mathcal{P}_{id}(E) = no$

• See dissertation for details
Electronic Voting Case Study

Supplemental Slides
Electronic Voting Case Study

• Demonstrates the Policy-Based Vulnerability Analysis Framework

• Target Environment:
  – Electronic voting setup for a single precinct
  – Ideal due to precise set of systems and procedures

• See dissertation for details
Buffer Overflow Characteristics

Supplemental Slides
Buffer Overflow Characteristic Tree
Direct Executable
Buffer Overflow IVEC
Indirect Executable Buffer Overflow IVEC
Direct Data
Buffer Overflow IVEC
Indirect Data
Buffer Overflow IVEC