Implementing N-Variant Systems

Benjamin Cox

Work with David Evans, Adrian Filipi, Jonathan Rowanhill, Wei Hu, Joel Winstead, Jack Davidson, John Knight, Anh Nguyen-Tuong and Jason Hiser

University of Virginia
N-Variant System Overview

- **Input Replicator**
- **Server Variants**
  - Derived from the same source code
  - Artificially diversified
- **Monitor**
  - Observes and compares behavior

![Diagram]

Input (Possibly Malicious) → Input Replicator → Variant 1, Variant 2 → Monitor → Output
Defensive Properties

- Thwarts any attack that cannot simultaneously compromise all variants

- Example Variations
  - Disjoint Address Space
    - Any attack that references an absolute address
  - Different Instruction Sets
    - Code Injection Attacks
Kernel Implementation

- Modify task structures
- Create new system calls to bring N-Variant System into execution
- Intercept system calls of variants
  - Ensure variants call the same system call with the same parameters
  - Perform the system call once
  - Return result to all variants
Current Status

- Successfully thwarted an attack on a vulnerable web server.
- Open Questions
  - What kinds of variations work well?
  - Performance improvements
  - Non-System Call Channels
    - Memory Mapped I/O