# Enhancing Server Availability and Security Through Failure-Oblivious Computing

Martin Rinard, Cristian Cadar, Daniel Dumitran, Daniel M. Roy,Tudor Leu, and William S. Beebee, Jr.

**Student presentation**

# Problem

- Memory Errors and Memory Corruption
  - Buffer Overflow
  - Out of Bounds Array Accesses
  - Invalid Pointer Accesses
- Importance
  - Exploits
  - Program Termination / Service Availability Lost
  - System Robustness

# Problem

- Memory Errors can cause the computation to:
  - Terminate with addressing exception
  - Become stuck in an infinite loop
  - Change flow of control
  - Corrupt data structures that must be consistent
  - Produce unacceptable results

# Approach

- Failure-Oblivious Computing
  - Mechanism to protect against memory errors and corruption
    - Ignore invalid writes
    - Manufacture values for invalid reads
    - Program does not know it has made an error – Oblivious
    - Program continues execution
  - Implemented at the compiler level
    - Inserts dynamic boundary checks
    - Inserts continuation code

# Evaluation

- Assumptions
  - Tests limited to buffer overrun attacks
  - Servers tested have short error propagation distances
- Weaknesses
  - Unanticipated Execution Paths
    - Manufactured results can lead the program down an unexpected path leading to incorrect results
  - Bystander Effect
    - Create dependency on the mechanism and overall production quality is decreased

# Evaluation

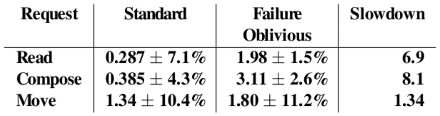
- Strengths
  - Availability
    - Program remains available after failure occurs
  - Security
    - Program is invulnerable to common memory related attacks
  - Minimal Adoption Cost
    - Implemented by the compiler – No code modification necessary
  - Reduced Administration Overhead
    - Patches for the sole purpose of fixing memory related security holes can be safely ignored

# Evaluation

- Testing
  - Evaluated impact on several widely used open-source servers with known memory errors
    - Pine, Apache, Sendmail, MC, Mutt
  - Three versions of each program
    - Standard Compilation
    - CRED Compilation
    - Failure-Oblivious Compilation
  - Criteria
    - Security and Resilience
    - Performance
    - Stability

# Evaluation

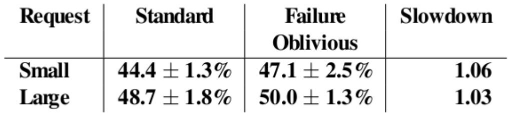
- Pine
  - Error
    - Escaping “From” field into heap-allocated buffer
  - Security and Resilience
    - Standard version results in a Segmentation Fault, CRED version catches the error and terminates program
    - Both leave pine unusable as the error occurs during initialization
    - Failure-Oblivious causes field to be truncated
      - Different execution path correctly parses field allowing successful execution
  - Stability
    - 25 messages a day interleaved with malicious input
    - Input of 100,000 messages
  - Performance



| Request | Standard | Failure Oblivious | Slowdown |
|---|---|---|---|
| Read | 0.287 ± 7.1% | 1.98 ± 1.5% | 6.9 |
| Compose | 0.385 ± 4.3% | 3.11 ± 2.6% | 8.1 |
| Move | 1.34 ± 10.4% | 1.80 ± 11.2% | 1.34 |

# Evaluation

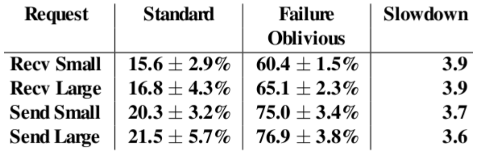
- Apache
  - Error
    - URL re-write match pattern offsets saved into static buffer
  - Security and Resilience
    - Standard version results in Segmentation Violation, CRED catches error and terminates
    - Apache starts a new child process to continue serving requests
    - Failure-Oblivious ignores the invalid writes, preventing the attack and process termination
  - Stability
    - 400 requests a day in addition to tens of thousands of requests from local box, interleaved with malicious input
  - Performance



| Request | Standard | Failure Oblivious | Slowdown |
|---|---|---|---|
| Small | 44.4 ± 1.3% | 47.1 ± 2.5% | 1.06 |
| Large | 48.7 ± 1.8% | 50.0 ± 1.3% | 1.03 |

# Evaluation

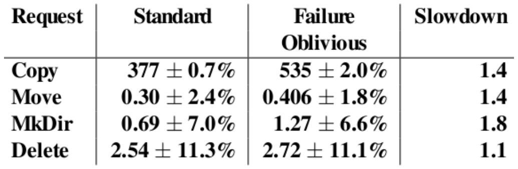
- Sendmail
  - Error
    - Translation of address into static buffer
  - Security and Resilience
    - Standard version results in Segmentation Violation, CRED catches error and terminates
    - CRED version completely disabled by another memory error during initialization
    - Failure-Oblivious version ignores error, continues execution
  - Stability
    - Used to send hundreds of thousands of messages, interleaved with malicious input
  - Performance



| Request | Standard | Failure Oblivious | Slowdown |
|---|---|---|---|
| Recv Small | 15.6 ± 2.9% | 60.4 ± 1.5% | 3.9 |
| Recv Large | 16.8 ± 4.3% | 65.1 ± 2.3% | 3.9 |
| Send Small | 20.3 ± 3.2% | 75.0 ± 3.4% | 3.7 |
| Send Large | 21.5 ± 5.7% | 76.9 ± 3.8% | 3.6 |

# Evaluation

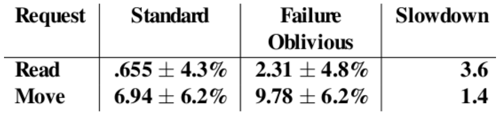
- Midnight Commander
  - Error
    - Accessing uninitialized buffer when parsing links in tgz files
  - Security and Resilience
    - Standard version results in Segmentation Violation, CRED catches the error and terminates
    - Failure-Oblivious allows program to continue and display results
  - Stability
    - Daily use with interleaved accesses of problematic files
  - Performance



| Request | Standard | Failure Oblivious | Slowdown |
|---|---|---|---|
| Copy | 377 ± 0.7% | 535 ± 2.0% | 1.4 |
| Move | 0.30 ± 2.4% | 0.406 ± 1.8% | 1.4 |
| MkDir | 0.69 ± 7.0% | 1.27 ± 6.6% | 1.8 |
| Delete | 2.54 ± 11.3% | 2.72 ± 11.1% | 1.1 |

# Evaluation

- Mutt
  - Error
    - Converting from UTF-8 to UTF-7 into heap-allocated buffer
  - Security and Resilience
    - Standard version results in Segmentation Fault, CRED version catches the error and terminates
    - Failure-Oblivious version effectively truncates the name
  - Stability
    - Daily use interleaved with malicious input
    - Processed 100,000 emails successfully
  - Performance



| Request | Standard | Failure Oblivious | Slowdown |
|---|---|---|---|
| Read | .655 ± 4.3% | 2.31 ± 4.8% | 3.6 |
| Move | 6.94 ± 6.2% | 9.78 ± 6.2% | 1.4 |

# Related Work

- CRED
  - Safe-C compiler
    - Terminates the program with an error message at first memory error
    - Similar to safe languages such as ML and Java which throw exceptions
- Acceptability-Oriented Computing
  - Acceptability Properties
    - Must hold for program execution to remain acceptable
  - Acceptability Enforcement
    - Built by programmer to ensure Acceptability Properties hold

# Related Work

- Variants and Extensions
  - Boundless Memory Blocks
    - Insert code to save invalid writes into table to retrieve later
  - Redirected invalid access back at appropriate offset
- Transactional Function Termination
  - Dynamically detect Buffer Overflows
    - Terminate Execution of function immediately.
- Static Analysis
  - Program Annotations
  - Heuristics

# Related Work

- Buffer-Overrun Detection Tools
  - StackGuard
  - StackShield
- Rebooting
- Manual Error Detection and Recovery
  - Failure Recovery Blocks and Exception Handlers
    - Programmer anticipates error, provides recovery strategy
  - Data Structure Repair
    - Programmer provides data structure consistency specification

# Result

- Failure-Oblivious Computation
  - Enhances availability, resilience, and security
    - Error does not corrupt address space and data structures of the computation
    - Continued execution through error
    - In many cases, converts unexpected or malicious input into a predetermined error case
  - Possible solution to one of the main goals of computer science
    - Create robust, resilient software that handles unexpected errors