Application Aware Checking – Reliability and Security Engine

Nithin M. Nakka, Zbigniew T. Kalbarczyk, Ravi K. Iyer

RELIABILITY

Instruction Counter Heartbeat (ICH)
- Uses performance counters, present in most modern microprocessors
- When a process completes a fixed number of instructions, it updates a counter
- The counter is periodically checked by a hardware module embedded along with the processor
  - Issues: Contact Switch, Cache Misses, Networking, I/O events – take non-deterministic time
  - The counters can be disabled to ignore these events

Advantages:
- Application need not be instrumented
- Leverages performance counters, already present in current CPUs, and information about process state and I/O events
- Performance overhead is minimal since software-level OS interaction is eliminated
- Minimal hardware required – timers and mechanism to update counters

Process Hang Detection

Loop Hang Detection Module
- Detects application hang by tracking entry and exit points of a loop
- Timer is started when loop is entered
- Waits for Loop exit signal
- Timeout value estimated by profiling of application

Sequential Code Hang Detection Module
- Detects application hang by looking for a repetition of instructions while application is executing sequential code (within a basic block)
- Maintains a log of previously committed instructions
- Parameterized at the beginning of a block with the maximum length of repeated sequence to look for
- Repetition log indicates illegitimate loop, possibly leading to hang

Application-Level Dependency extraction

Checkpointing & Recovery: Rendezvous of Security and Reliability

Recovery
- Common concern for Security and Reliability
- When the security of a system is compromised it can be crashed or restored to a safe state
- When a system behaves undesirably due to faults need to bring it back to a consistent checkpoint state

Data Dependency Tracker
- Uses a copy-on-write mechanism to save checkpoints of memory pages written by threads
- On the crash of a thread due to a security attack or a failure two methods of recovery
  - Crash all the threads
  - Not very useful for composition intensive applications
  - Crash all the threads depending on the crashing thread
  - Dependencies threads are threads which execute state of the crashed thread
  - Rollback the crashed thread and its dependent threads to a consistent and safe point in execution

Reliability and Security Engine (RSE)

Future Directions
- Synthesize framework with DLX
  - Integrate reliability and security
- Fault-injection based assessment of techniques
- Dynamic Reconfiguration
- Insert assertions
- Compiler Support

SECURITY

Secure Return Address Stack
- Detects stack-based buffer overflow attacks when a malicious overwriting return address is being used for returning to the invoking function
- Prevents the attacker from successfully seizing control of the target application program
- Secure Return Address Stack (SRAS), checking done in parallel with the processor’s main pipeline – hides the checking latency
- Typical overhead of the proposed solution is in the order of 0.1%
- Compared to 24% for software-based approach shown in our previous study

Memory Layout Randomization
- Randomly relocates position independent regions including stack, heap
- Can mask a broader range of vulnerabilities due to low level programming errors, e.g., buffer overflow, format string, signed integer overflows
- MLR module implements
  - the executable header parsing, address computation algorithms and randomization of the stack and heap in hardware
  - Runtime tracking of stack and heap calculated from information embedded in the header of the executable file
  - Relocation of position-dependent regions (Global Offset Table) with OS Support
- Can be potentially used by different operating systems while the previous software implementation is always system specific

Data Dependency Tracker Module
- In multithreaded applications, on a single thread crash the traditional kill-all-thread approach leads to heavy overheads.
- Execution of healthy threads should be kept intact in such a case.
- Data Dependency Tracker (DDT) Module uses a page-based memory dependency checking mechanism to track dependences between threads.
- DDT tracks memory accesses and uses a copy-on-write mechanism for memory pages to checkpoint the memory state.
- All threads depend on faulty thread (either malicious or non-malicious) are killed
- Recovery uses two algorithms
  - Without execution rollback
  - With execution rollback

Data Dependency Tracker State Transition

Performance Evaluations for SRAS

www.iti.uiuc.edu
University of Illinois at Urbana-Champaign