Decoupling
Dynamic Program Analysis from Execution in Virtual Environments

J. Chow, T. Garfinkel and P. M. Chen
Presented by Sandra Rueda

September 19, 2008
Motivation

- Do not debug in production environments
Area

- Analyzing behavior of running programs (dynamic analysis) on production environments
  - Performance decreases
    - There is overhead because of complex analysis techniques
  - Perturbation (Heisenberg effect)

Decoupled Analysis
Aftersight

- **Virtual machine based architecture** that decouples execution from analysis by logging non-deterministic inputs and replaying them on a separate platform
Related Work

• Dynamic analysis
  ‣ Array bound violation analysis
  ‣ Invalid pointer access analysis
  ‣ Un-initialized variable detection

• Previous projects .. (some of them):
  ‣ Watchdog processors
  ‣ Shadow processing
  ‣ Taint-based analysis
Watchdog Processors

- Coprocessor used to perform error detection
- Runs concurrently with the main processor
- Setup phase, checking phase
Shadow Processing

- Second process monitoring a first one’s instructions
- It uses source program analysis to generate “the shadow version”
- Main and shadow programs run concurrently
Tracing of Tainted Inputs

- Detect situations on which non-trusted inputs affect security sensitive operations
- Identify untrusted inputs and security sensitive operations. Add annotations in the source code.
- Use information flow tracking techniques (programs are modified to insert additional code for tracking)
Issues

• Additional Hardware
• Additional code
• Overhead in production environment
Aftersight Architecture

VM

VMM (Aftersight)

Instrumented Analysis

VMM/simulator (Aftersight)

log

record

synchronization

replay
Usage Models

- Synchronous safety
- Best-effort safety
- Offline Analysis
Implementation

- “To support replay, a VM must record and replay all inputs to the CPU that are not included in the state of the guest memory, register, or disk”.

- Correctness of replay relies on domino effect

- Analysis runs during replayed execution

- Builds on VMM and CPU simulator technologies
Synchronization

Aftersight on the analysis VM side supports:

- Speculative execution
- Deferred outputs (until the analysis reaches the same point of execution)

Improving performance on the second environment

- Interrupt delivery is immediate
- Device I/O can be accelerated
- Exception checking results are registered
Aftersight Advantages

- Dynamic configuration of analysis tools
- Reduces the bottleneck caused by analysis tools
- Enables parallel execution of analysis tools
Performance with synchronous safety. Fetch directory of linked web pages from a local web server. The check running in the analysis VM mimics an access virus scanner by computing for 2 ms on each disk request.
Performance with best-effort safety. Do heap overflow bounds checks on dynamically allocated kernel objects.
Decoupling Analysis. (Interactive desktop workload)
Fig. 1 Primary and Analysis VM
Fig. 2 Computing rates
Fig. 3 Lag in progress
Examples

- ESX server
  - references to data located in popped off stack frames
- Linux kernel
  - use of un-initialized variables
What can we do with this?

• We can run almost any analysis on the second environment
  ‣ In some cases non-deterministic inputs are enough
  ‣ In some cases it is important to reproduce the complete software stack

• What about
  ‣ Functions that depend on platform specifics
  ‣ Performance analysis
  ‣ Secret information in the log
Take Away

• Aftersight enables analysis on production workloads
• Replay on the second environment enables execution of almost any checking
• But … the second environment must be set according to analysis requirements
• And .. Consider the risks of logging non-deterministic inputs