EVADE: Evidence-Assisted Detection and Elimination of Security Vulnerabilities

PIs: Emery Berger (Umass Amherst) and Tim Wood (GWU)

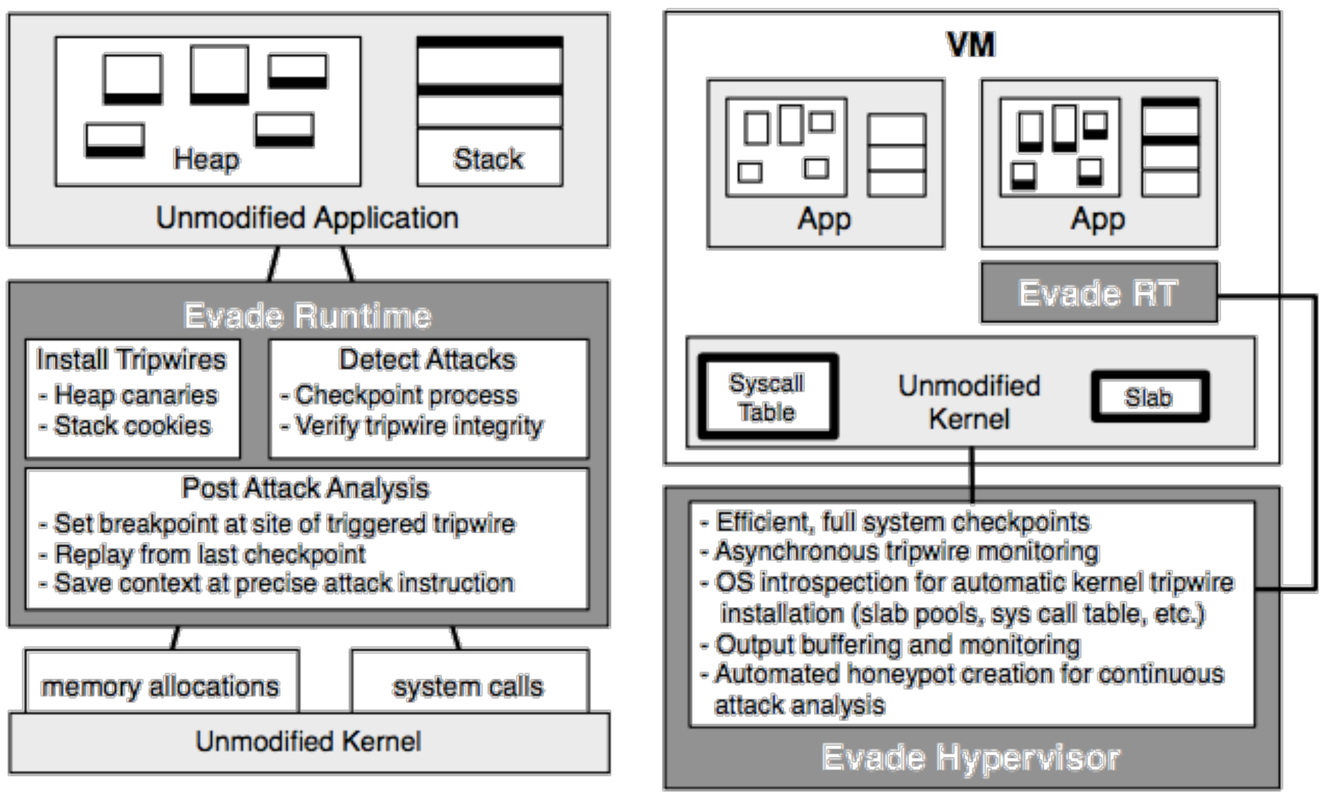
Many attacks leave behind evidence. Rather than try to prevent attacks, EVADE deploys lightweight tripwires that can be asynchronously scanned to find the trail left by an attack or exploit. By buffering external outputs until scans are complete, EVADE can guarantee that attacks will have no observable effect, while reducing overheads compared to memory safety techniques that require instrumentation.

EVADE Runtime

- Detects memory exploits or errors in unmodified programs

Evade VM

- Hypervisor-based detection for guest OS and applications



EVADE Approach

Tripwires

- Canaries, stack cookies, etc.
- Simplify detection of attacks

Output Buffering

- Hold network packets, disk writes, irrevocable system calls between scans

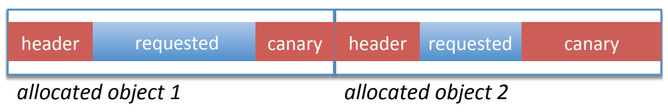
Post Attack Analysis

- Replay from last checkpoint to precisely identify root of attack
- Switch to honeypot mode and gather forensics while protecting data

Fast Precise Error Detection [ICSE16]

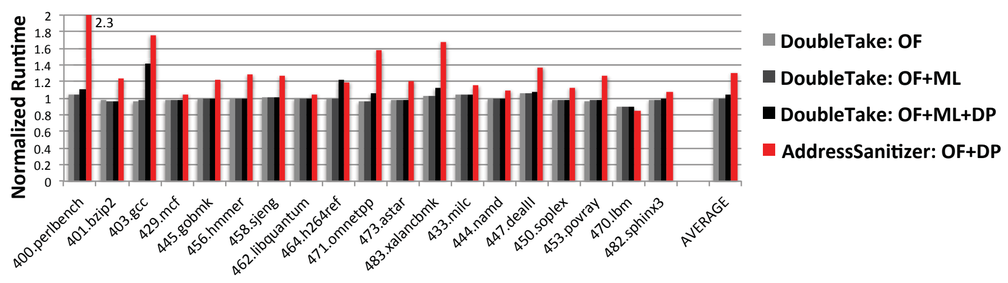
Runtime environment intercepts memory allocations and system calls

Adds canaries around allocated objects as tripwires

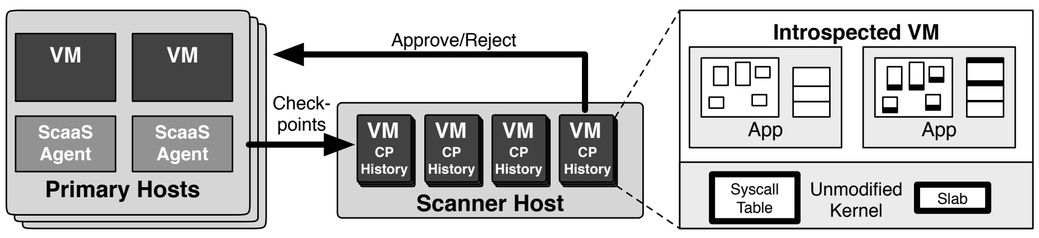


Detects buffer overflows, use-after-free, and other memory exceptions

Incurs less than 5% overhead



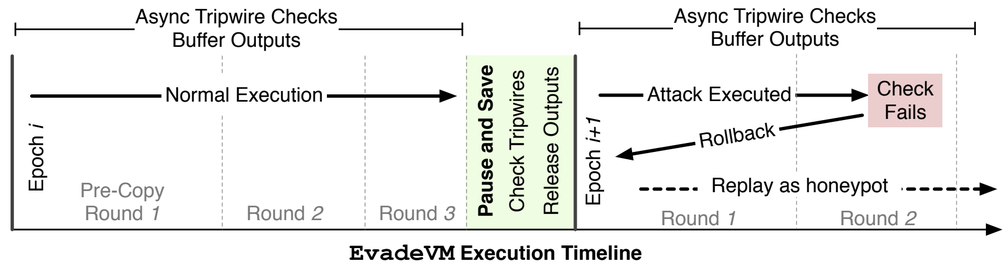
Security Scanning as a Service [HotCloud16]



Uses live VM checkpointing to save VM state

VM Introspection allows scanner to understand VM memory and detect attacks

Can rollback to past checkpoints for forensic analysis



Interested in meeting the PIs? Attach post-it note below!

NSF Secure and Trustworthy Cyberspace Inaugural Principal Investigator Meeting
Nov. 27 -29th 2012
National Harbor, MD

National Science Foundation
WHERE DISCOVERIES BEGIN

UMASS AMHERST

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC