EVADE: EVidence-Assisted Detection and Elimination of Vulnerabilities

Challenge:

- Many attacks leverage memory errors (e.g., buffer overflows) to exploit a system
- Current solutions incur excessive overhead to prevent attacks

Solution:

- Use "trip wires" to collect evidence
- Asynchronously scan for attacks to reduce overhead
- Rollback and carefully analyze to find root causes of attacks

Unmodified Application Evade RT **Detect Attacks** Unmodified Slab Heap canaries Checkpoint process Table Kernel Stack cookies Verify tripwire integrity Post Attack Analysis Efficient, full system checkpoints Set breakpoint at site of triggered tripwire Asynchronous tripwire monitoring Replay from last checkpoint OS introspection for automatic kernel tripwire Save context at precise attack instruction installation (slab pools, sys call table, etc.) Output buffering and monitoring Automated honeypot creation for continuous memory allocations system calls Unmodified Kernel Evade Hypervisor **EVADE** runtime & hypervisor

Scientific Impact:

- Provides lowoverhead attack detection and analysis
- Cross-layer detection of attacks in applications or OS

Broader Impact:

- Prevents common attacks on unmodified applications
- Practical, low overhead approaches, realistic for deployment
- Several undergraduate research students involved with the project

Awards 1525888 and 1525992 Emery Berger (Umass Amherst) and

Tim Wood (George Washington University)