MINESTRONE: Combining Static and Dynamic Analysis for Software Protection

Angelos D. Keromytis, Junfeng Yang, Sal Stolfo (Columbia)

Angelos Stavrou, Anup Ghosh (GMU)

Dawson Engler (Stanford)

Marc Dacier, Matthew Elder, Darrell Kienzle (Symantec)

MINESTRONE

- Address the problem of security of software of unknown provenance
  - open source or COTS software is brought within an organization
    - might contain intentional vulnerabilities (backdoor or active information leakage)
    - certainly contains unintentional vulnerabilities
  - how do we establish some measure of trust and/or assurance?

http://nsl.cs.columbia.edu/projects/minestrone

Project Focus

- Continuous feedback between dynamic confinement and static analysis techniques to improve vulnerability detection and reduce performance impact of security
- Multi-thrust approach, focusing on legacy applications written in unsafe languages for which source code may be available
  - however, not always desirable/feasible to operate on source code
- Looking at current and future vulnerability classes
  - e.g., problems introduced by increased use of multicore CPUs and parallelism

Runtime Confinement

- Exploring different approaches for introducing an adaptive inline reference monitor
  - binary instrumentation
  - source-code rewriting
  - binary injection
  - lightweight virtualization containers
- Experimental evaluation along performance and effectiveness axis
- Related capabilities: self-healing, leakage detection, multi-core and GPU exploitation

Concurrent Analysis

- Continuous symbolic execution
  - Use dynamic instrumentation to prune/direct state-space exploration
- Static analysis often generates a lot of “noise”
  - Conservatively follow leads by applying dynamic instrumentation (through IRM)
    - Do not rollout unnecessary instrumentation
  - Over time, remove instrumentation deemed “unnecessary”
  - Optimize instrumentation for common case
- Expose information gleaned from source code (e.g., types, information flow) to dynamic confinement component

Software Diversification

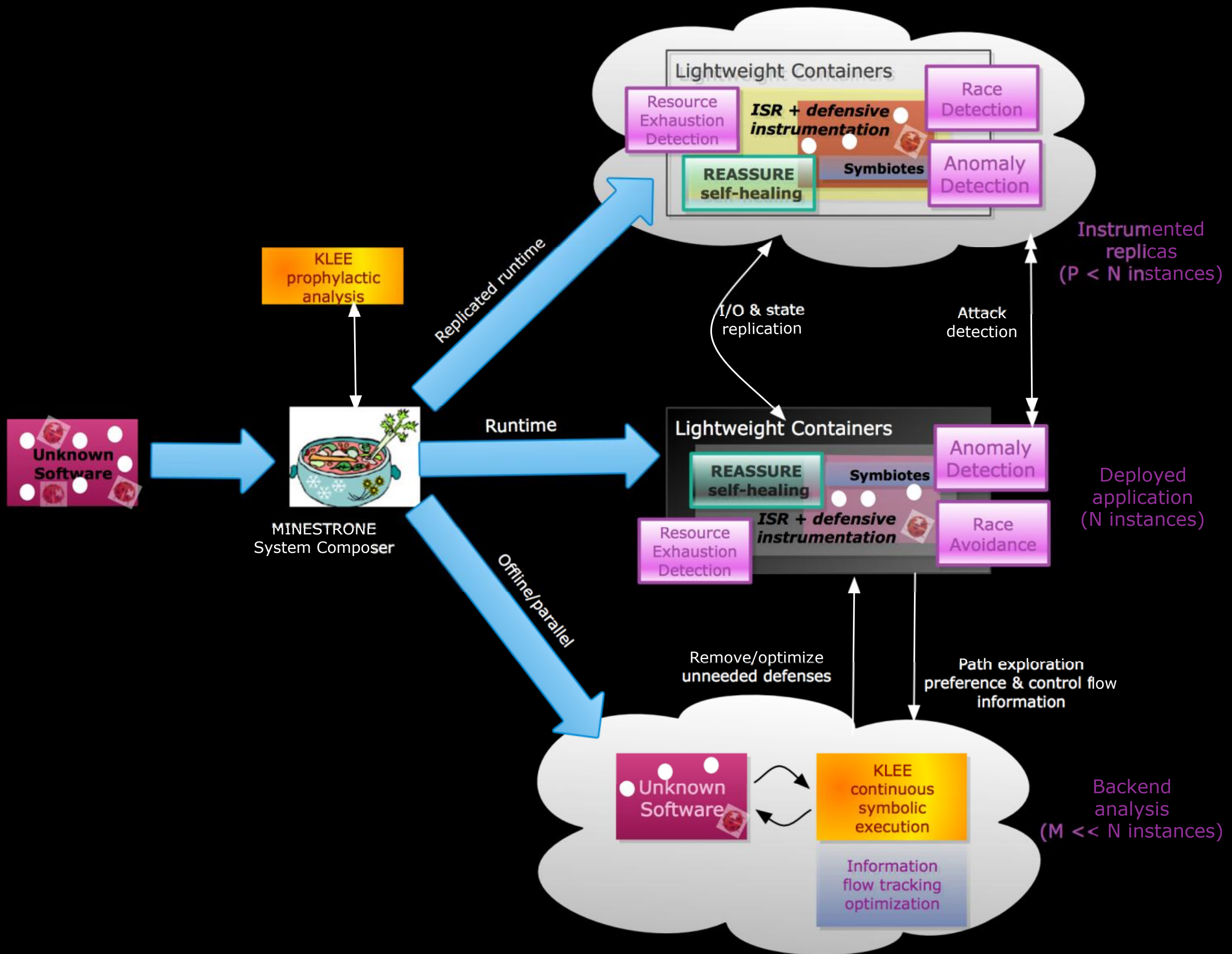
- Mitigation mechanism for certain classes of vulnerabilities
  - Code injection (SQL, binary, ROP, …)
  - Sensor for a posteriori detection of attacks/vulnerabilities
- Key starting technology: Instruction Set Randomization (ISR)
  - On-the-fly creation of diversified runtimes (x86, SQL)

Backup Material

Limitations in state of the art

- Dynamic confinement techniques impose performance and functionality limitations
- Static analysis techniques do not scale much beyond 10,000 LoC
  - improvements basically track Moore's law

MINESTRONE Architecture



Evaluation

- Test against a variety of attacks
  - synthetic
  - hand-crafted
  - real exploits
- Eval scope is unprecedented
  - contribution by itself

Outcomes to date

- Publications (7) and prototypes (3)

- "Practical, low-effort verification of real code using under-constrained execution"
  David A. Ramos and Dawson Engler. To appear in the 23rd International Conference on Computer Aided Verification (CAV). July 2011, Snowbird, UT.
- "Retrofitting Security in COTS Software with Binary Rewriting"
  Padraig O'Sullivan, Kapil Anand, Aparna Kothan, Matthew Smithson, Rajeev Barua, and Angelos D. Keromytis. To appear in the the 26th IFIP International Information Security Conference (SEC). June 2011, Lucerne, Switzerland.
- "Global ISR: Toward a Comprehensive Defense Against Unauthorized Code Execution"
  Georgios Portokalidis and Angelos D. Keromytis. In Proceedings of the ARO Workshop on Moving Target Defense. October 2010, Fairfax, VA.
- "Stable Deterministic Multithreading through Schedule Memoization"
  Heming Cui, Chia-che Tsai and Junfeng Yang. In Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 207 – 222. October 2010, Vancouver, Canada.
- "Bypassing Races in Live Applications with Execution Filters"
  Jingyue Wu, Heming Cui and Junfeng Yang. In Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 135 - 150. October 2010, Vancouver, Canada.
- "Fast and Practical Instruction-Set Randomization for Commodity Systems"
  Georgios Portokalidis and Angelos D. Keromytis. In Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), pp. 41 - 48. December 2010, Austin, TX.