Making Programs Forget: Enforcing Lifetime for Sensitive Data

Jayanthkumar Kannan (Google Inc), **Gautam Altekar** (UC Berkeley), Petros Maniatis (Intel Labs), Byung-Gon Chun (Intel Labs)
The Problem: Lingering Data

- How long is your data around? (Chow et al. '04)
  - Where in memory?
  - Maybe on disk?
Hard to Provide Sensitive Data Lifetime

Existing approaches fall short

• Shutdown the application?
• Reboot?
• Rely on application support?
• Memory scrubbing? (Chow et al '05: Data shredding)
• Change user behavior? (Borders et al '09: Capsules)
• Time-based data access control? (Perlman '05)
Goal: Guaranteed Data Lifetime

- Guarantee: Data indicated as sensitive is not retrievable from system beyond specified time limit

- Requirements
  - No application support
  - Non-disruptive: shouldn’t crash, interrupt your normal workflow

- Contribution: Promising start, much further to go
Observation: State Equivalence

• For any program state computed from sensitive data, there usually exists an equivalent state not derived from the sensitive data

• Example:
  o You get a sensitive email, read it, and then send and read some other emails
  o Equivalent State: Send and read other emails
Approach: State Reincarnation

• Replace current sensitive state with equivalent non-sensitive state

• Challenge: How do we derive equivalent non-sensitive state?
Deriving an Equivalent State

- Key idea: deterministic replay with perturbed input

1. Original execution (record all inputs)
   - sys_read(buf)
   - S
   - Sensitive input (user-designated)
   - Sensitive state

2. Replay execution (replace sensitive inputs)
   - sys_read(buf)
   - S'
   - Substitute w/ Non-sensitive input
   - Non-sensitive state
Challenges

• Picking the sensitive-input replacements
• Completeness: Eliminating all sensitive data
• Overhead: Run-time cost
Picking sensitive-input replacements

- Given sensitive input I, and subsequent input I1, I2, we compute I' which leads to same execution path
  - Using tainting and constraint solving (Altekar '09)
- Replay with I'
- Hard-cases: Spell-checker, Hashing
Completeness

• Sensitive data can linger in various areas (OS buffers); how can we remove all of it?

• Technique: Implement perturbed replay in VM

• Need to trust VM not to retain data
Overhead

• We implemented recording at user-level

• Slowdown: ~1.2X on bash
Conclusion

• Contributions:
  o Guaranteed Lifetime Property
  o State Reincarnation

• Future work:
  o Picking right inputs for replay
  o Measuring overhead for consistent substitution