Trends and New Opportunities in Software Protection

Mikhail Atallah

Purdue University (CS, CERIAS)

Arxan Tech. Inc.

Protection from what ?

- Piracy of the software itself
  - Unlicensed copies
- Piracy of data viewed using the software
  - Movies, e-books, etc
- Theft of secrets in the software
  - Crypto keys

Protection from … (cont’d)

- Theft of IP (e.g., algorithms)
  - Reverse engineering
  - Code-lifting
- Unauthorized modification
  - Remove or add functionalities
  - Restore pre-disabled functionalities
    - Turn demo version into full

Protection from who ?

- Adversary controls all processor(s)
- Adversary controls all but 1 processor
  - "who will protect me from that 1 chip in my PC that is under your control"
- Adversary control of data
  - Protect integrity of control flow

Standard techniques

- Encryption
  - Aucsmith, …
- Transformations
  - Collberg, Thomborson, …
  - Obfuscation (lexical, control, data)
  - Watermarking (static, dynamic)
  - Tamperproofing
- Revisit in context of multicore ?

Multi-core

- Lower protection footprint
  - Less performance penalty
- Better protection
  - Better obfuscation
- One core is tamper-resistant
  - More secure, but slower
  - How to use it effectively

Software splitting

- Zhang et al., Mana et al., Ceccato et al.
- Split software into …
  - Open components that run on unsecure processors
  - Hidden components that run on secure processors
- Hard for adversary to get hidden ones
- Requires communication

Software splitting (cont’d)

- Blocking – how long ?
- If secure processor is remote …
  - Latency (network)
  - Computation at remote end
- If secure processor is local …
  - Latency (bus)
  - Computation in secure processor (slower)

Software splitting (cont'd)

- Dvir et al.
  - Virtual leashing to mitigate latency problem
  - Split into active and lazy
  - Run active tasks on unsecure processor
  - Run lazy tasks on trusted processor

Replication

- Less likely for all copies to go wrong in same way
  - NASA (3-way)
- Johnson et al.
  - Within same processor

Attestation

- Integrity verification
- "Prove your integrity" challenges
- Trusted challenger
  - Issues challenges to responder
- Problems with binary attestation
  - Versions, patches

Attestation (cont’d)

- Property-based
  - Sadeghi, Stueble …
- Time-based
  - Kennell et al., Seshadri et al.
  - Shankar et al. (attacks)
  - Garay et al. (better challenges)

Attestation (cont’d)

- Anonymous
  - ZKP
- Scandariato et al.
  - Proofs-generating module
  - Run-time refresh of module

VMs

- No need to tamper: Run in VM
  - Trap unwanted functionalities
- Anti-VM
  - Similar to anti-debug
  - How to detect if running on a VM
- How to react
  - Cause crash ?

PUFs

- PUF = Physically Un-clonable Function
- Produces response R to input C
  - R is obtained from a physical device upon providing it with C as input
  - Devices with same blueprint from same production batch have different functions
- Impossible to mimic in software
  - Even when in physical possession of device
  - Attempted physical probing destroys it

Binding with PUFs

- Use PUF to bind software to a specific instance of a hardware
  - Bind PUF responses to encryption key
- Cannot run pirated software without access to PUF
- Can use multiple copies
  - "PUF server"

Binding (cont'd)

- Fake failure
  - Get additional copy
  - Herzberg et al.

Theory

- Goldreich, Ostrovsky
  - Prevent replication w. HW, encryption
  - Hide pattern of memory accesses
- Simulation on oblivious RAM
  - Input-independent memory accesses
- Polylogarithmic cost
  - Logarithmic lower bound

Theory (cont'd)

- Impossibility results
  - AV, obfuscation, …
- Not necessarily bad news
- "Good enough" protection
  - Protecting for 2 weeks often OK
  - Information is perishable
- Need to quantify

Metrics: What

- Strength of protection
  - Time & effort to defeat
  - Cost of applying protection
  - Effort, computation, $, …
- Footprint of protection
  - On performance (speed, space, …)
  - On user (convenience)
  - On QA process

Metrics: How

- The measurement problem
- Red-teaming ?
  - Team-dependent (experience, luck, …)
  - Non-repeatable
- Modeling & simulation ?
  - Difficult (dangerous?)
- Piggyback on other metrics work ?
  - E.g., software metrics

Metrics (cont’s)

- Let insurance companies do it?
  - Under-reporting
  - Mis-pricing
  - Too coarse