

Fine-Grained Runtime Data Section Attestation for Embedded Software

Chongkyung Kil and Peng Ning

The Growing Vulnerability: Ubiquitous Embedded Systems!

- Networked embedded systems are everywhere: routers, PDAs, cars, cell phones
- They used to provide simple, limited functions, but now we require them to do more sophisticated work, e.g. Intelligent Home Control
- Very diverse SW development environment. Providing more services are highest priority in the current market, not security!

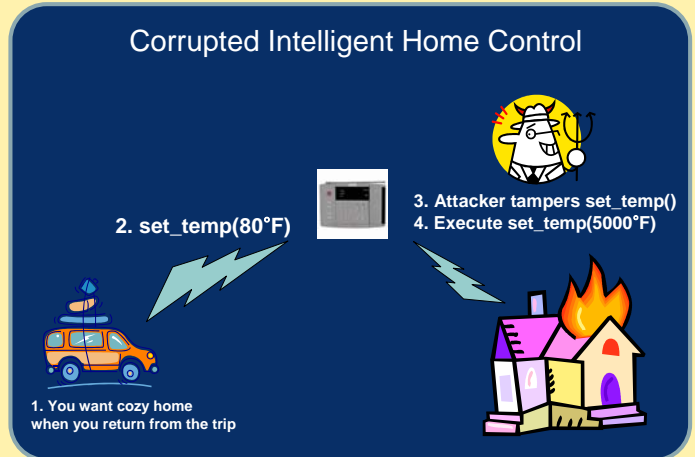
How do you believe your embedded systems are correctly operating?



Software Attestation Can Help!

Attest by Merriam-Webster

- 1a to affirm to be true or genuine
- b to authenticate officially



Comparison to state of the art

Current approaches

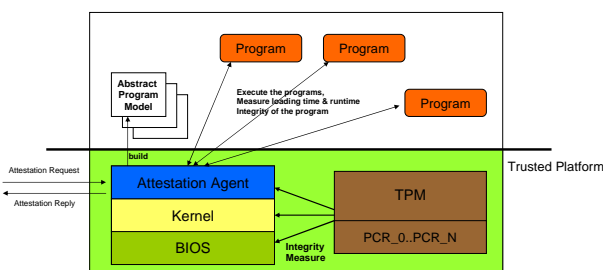
- HW-based attestation (Hash)
- SW-based attestation (Memory Traversal)
- Hybrid attestation

Our new approach

- **Focus on the program's runtime data section**
- **Constraints-based attestation**

- ❑ Our goal: verify the integrity of the data section during the runtime
- ❑ Our method
 - Derive all possible constraints of valid (not compromised) runtime data section so that we can determine whether current data section's state satisfies such constraints
 - Build the abstract program model that keeps the all constraints information of the target program
 - Abstract program model comprises several sub-models (e.g. abstract stack tree) that keep specific constraints of each segment in the data section
- ❑ Currently 12 constraints are derived (there will be more later), e.g. boundary constraint, return address constraint, type constraint, frame pointer constraint

Attestation Framework



Open Discussion

- Any more constraints?
(heap, .got, .ctors, .dtors, .plt, ...)
- Hybrid or SW-based attestation?
- Implementation Issues
- Still far to go!

Help Wanted