TCIPG

Intrusion Detection for Smart Grid Components by Leveraging of Real-Time Properties

Man-Ki Yoon, Fardin Abdi, Sibin Mohan, Rakesh Bobba, and Lui Sha

RETHINKING REAL-TIME SYSTEM SECURITY

- The power grid has many real-time embedded systems deployed in the field (e.g., relays, IEDs).
- Running traditional IDS in real-time embedded systems is not practical.
  - Mainly because of real-time and resource constraints.
- Modern real-time systems are smarter but less secure than traditional systems.
  - **Traditional RTS**: Physically isolated, limited capability, use of specialized protocols.
  - **Modern:** More networked, open, standard (COTS) platforms, sensitive/private information.
- Need IDS that is aware of real-time environments and can leverage their properties.

FEATURES OF REAL-TIME EMBEDDED SYSTEMS

Limited Resources
- Computational power, energy, cost

Timing Requirement
- Safety, reliability, quality of service

System Upgrade
- Verifiability

Resource Overhead	• Fine-grained monitoring • Instructions, memory access, function/system calls
High Upgrade Cost	• Formal verification for every new update
Increasing Complexity	• Capabilities and access control
Adaptability	• Domain-specific, unforeseen vulnerabilities

RESEARCH PLAN

- **Goal:** Detection of anomalous/malicious behavior.
  - Detection, not prevention!
- **Idea:** Based on predictable behavioral patterns.
  - Timing, control flow, memory usage.
  - I/O activity, power consumption, etc.
- **Approach:**
  - Profiling by machine learning or compile-time analysis.
  - Inspection by core-to-core monitoring in multicore systems.
  - Detection by deviation from legitimate behavior.
- Embedded systems are **predictable by design**.
  - Finite set of operational modes.
  - Periodic jobs.
- **Deviation** from expected behavior: **Abnormality**.

Threat Analysis
- Identification of the aspects of system behavior that are affected by attack type
- Studying attacks that mimic system behavior

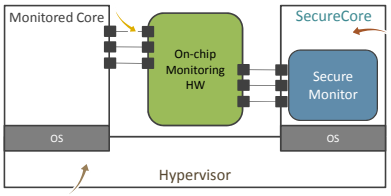
Development of Profiling Method
- How to effectively capture all behavioral variations excluding interference
- Platform-independent profiling

Implementation
- Minimal processor architecture modification
- High-resolution monitoring with minimal overhead
- Tamper-resistant secure core

RESEARCH RESULTS

On-chip Monitoring HW Unit

- Observes the state of monitored cores, I/O activities, physical states, etc.
- Invisible to all but SecureCore, nonintrusive.

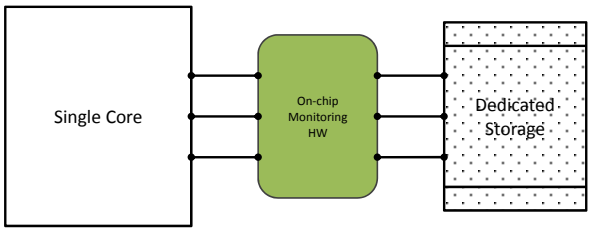


Secure Monitor

Software process that performs monitoring and detection using observed behavior

Hypervisor-based SecureCore Protection

- Resource virtualization: memory space separation, I/O device consolidation.
- Additional HW-based protection (e.g., ARM TrustZone).



- Dedicated storage contains profiles of running programs.
- Dedicated memory is isolated and intrusion-tolerant.
- Monitoring hardware is hooked to internal registers of processor.
- On-chip monitoring hardware does not interfere with the core's execution.
- Monitoring HW compares the run-time signature with the profile.

Types of attacks that can be detected:

- Return address modification attacks.
- Code injection attacks.
- Some code replacement attacks.

BROADER IMPACT

- **Higher security guarantees**
  - Far fewer bugs in hardware design.
  - Hardware-based techniques are not vulnerable.
- **Low overhead**
  - Monitor coarse-grained "aggregated behavior."
  - Periodic or event-driven watch point.
  - Not in the critical path: offloading of monitoring.
- **High adaptability**
  - Can detect attacks on unforeseen vulnerabilities.
  - No assumptions on specific threat models.

FUTURE EFFORTS

- Apply techniques to real-world systems such as power grid controllers, mobile devices, etc.
- Develop the technique for use on more general-purpose systems.
- Building a test-bed (in conjunction with industry partners).

FUNDING SUPPORT PROVIDED BY DOE-OE AND DHS S&T