This paper is part of the following report:


To order the complete compilation report, use: ADA485570

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP023711 thru ADP023727
Summary and Outlook – Research Challenges
Open Research Questions

- Adversary models
  - Define/Formalize adversary models
  - Need to incorporate characteristics of new technologies and applications
  - Need to consider the threats to the defense mechanisms
- Define performance/security metrics
- Develop process/methodology for developing adversary models

Languages and Software Engineering (1)

- Embedded constraints affect security
  - Quantify capabilities, limit scope, target 8-bit & larger
  - beyond sensor nodes --> deeply embedded systems
  - Issues: critical, non-recoverable, special netw., no sys admin
- Design for security (not retrofit) → formalize
- Need for metrics & models (some differ for embedded)
  - e.g. higher reliability probability for safety crit. affects security
- Need for diversity (e.g., via dynamic adaptation)
  - Self-modifying apps (e.g., via dyn. Transformation of pgm)
  - Self-protecting/self-checking apps?
  - Dynamic updates (in 24/7 operation)
  - Classification in terms of threat models
Languages and Software Engineering (2)

- Need to limit overhead, retain predictability, low cost
- Incorporate real-time requirements
  - Hard timing constraints limit security options
  - May undermine existing network protocols, add overhead...
  - Interface b/w RT and non-RT components problematic
  - Embedded clients + server need protection of both
- Need hardware assistance

Software Security

- Can light-weight, effective, semantics-based, compiler-level reasoning systems to characterize program behavior, malware, etc be built?
- Can effective hybrid reasoning systems be built by combining static analysis and runtime monitoring systems?
- Can evaluation contexts be characterized to simplify and reduce efforts in reasoning about software artifacts?
- How can binaries be reasoned about without too many false alarms?
- Can binaries be instrumented to discover security threats and to degrade gracefully in the event of a security fault?
- How to characterize threats, vulnerabilities,...?
- How to build provably correct core components of OS against specific threat definitions?
- Code integrity to leverage to achieve system security.
- Secure and scalable software update on deployed systems.
- Tool support for code obfuscation.
Hardware Security (1)

Problems
- Interactions/problems across all levels
- Types of attacks:
  - HW-layer attacks
  - Upper-layer attacks with HW solutions (to reduce cost)
- What adversary/threat models for HW?
  - New ones like those on FPGA
- What channels possibly under attacks in HW?
  - bus, power/current, timing, keystrokes, ...
- Types of attacks from another perspective:
  - Malicious observation/privacy attacks (e.g., digital rights management)
  - Malicious tempering/integrity attacks

Hardware Security (2)

Approaches
- What defending HW features like obfuscation/randomization, encryption, authentication, ...?
  - solutions at HW layer $\leftrightarrow$ HW-layer attacks
  - solutions at HW layer $\leftrightarrow$ upper-layer attacks
- Software solutions vs. hardware solutions
- What types of protection at the upper layer (e.g., soft guards) may (not) be sufficient against certain types of HW-layer attacks?
  - solutions at upper layers $\leftrightarrow$ HW-level attacks
- Classification of solutions in terms of threat models
- How to develop holistic/hybrid solutions across layers?
- How effective are solutions in addressing/alleviating the problems (e.g., metrics)?
- How to address cost constraints (e.g., in time, space, power, ...)?
Security of Embedded Networks (1)

- How to provide efficient, secure and reliable distributed services in embedded networks?
  - Challenges: faults, dynamic population, mobility, resource constraints, node compromises, real-time requirement
- How to detect and recover from attacks?
  - Self-healing
- Strong security v.s. probabilistic and adaptive security
- How to provide secure and reliable architecture and interaction between embedded networks?
  - System of systems (or hybrid embedded networks?)
  - Decentralized v.s. centralized views
- How to achieve survivability and intrusion resilience?
- How to protect collected data?
  - Resource constraints

Security of Embedded Networks (2)

- How to provide secure initialization?
  - Quickly and securely “pair” groups of sensors (scalability, usability)
- How to make tradeoff between performance, security, and fault tolerance?
  - E.g., degrade/relax security services?
  - Metrics? New vulnerabilities? Degrees of security?
- How to reason about design principles?
- How to accommodate different vulnerability stages and emerging properties, and prevent unwanted side effects when systems evolve?
- Whither RFID/Sensor hybrid?
- How to protect network topology (even from the insiders)?
- How to keep node behavior (movements) private?
- What are the best way to provide diversity in embedded networks?
  - Analysis techniques, metrics, management issues, ...
- How to detect attacks/anomalies in embedded networks?
  - Sensor networks, MANET, mesh networks, ...
- Database of threat models?