Session 3 Rapporteur Notes

Robert Kaster, Bosch: Security for a Software-Defined Vehicle

Brock LaMeres, Montana State U.: Cyber Shield – An approach to defeat malware in an edge computer using hardware diversity

Lisa Spainhower, session chair
Carl Landwehr, rapporteur

Workshop Focus:
How to survive cyber attacks on safety-critical functions of intelligent vehicles

- Kaster: Providing high level view of the the problem of securing software defined vehicles
  - Three principles:
    - Defense in depth
    - Security by design: move from “V” to infinity (dev(sec?) ops)
    - Continuous risk management: “immune system” for connected SDV fleet

  Observation: good principles but the devil is in the details

  Bonus: personal research focus on attestation

    Self attestation: checking internally

    Remote attestation: checking from the outside

    Peer attestation: One ECU checking another?

  Observation: Great deal of work on attestation has been done in the cybersecurity world over the past 20+ years. Novelty here seems to be in the demands for real-time performance with minimal hardware support

Workshop Focus:

How to survive cyber attacks on safety-critical functions of intelligent vehicles

- LaMeres: providing a detailed technique for mitigating one type of threat
  - Focus on detecting/tolerating storage corruption on embedded controller through hardware instruction set diversification
  - Attack model: attacker inserts executable code in buffer and transfers control to it for execution (conventional overflow attack)
  - Defense is to alter the CPU instruction set so that attacker's inserted instructions won't have the intended effect
  - Implementation is to have one FPGA core running original instruction set and two others running permutations of that instruction set
  - Intended software needs to be compiled once for original instruction set and again for each of the permuted instruction sets
  - Three cores run in lockstep with comparator checking at each instruction execution whether two cores are executing the same opcode (should happen only if memory is corrupted)
- The idea generated considerable interest and discussion
- Observation: considerable similarity to work on software diversification originally initated by Stephanie Forrest at UNM more than 20 years ago and substantially advanced by Michael Franz and his group at UCI, only mapped to the hardware level. Looks to be effective for the attacks it aims to defeat. What if you only ran one (or two) of the permuted versions?
- Costs are argued to be low because extra FPGA cores are free

Summary

Both presentations addressed the workshop focus, from different perspectives

Both drew on the base of cybersecurity research of the past decades

It looks like we shall relearn the lessons from that research in this new domain, which presents some aspects and constraints that differ somewhat from past applications of the general principles