ASMONIA:
Integration of Software Integrity Protection (IP) into Mobile Networks

Manfred Schäfer,
Nokia Siemens Networks Research
‘Integrity Protection’ IP

Methods for
- Protection concepts
  - Network Element (NE)
    - Crypto-Integrity
    - Hardening
  - User Equipment (UE)
    - Crypto-Integrity
    - Hardening
    - Malware
- Anomaly detection
  - Integrity breaches (UE, NE)
  - Malware incidents (UE)

Dedicated mechanisms in
- Network elements
- User equipment
- Infrastructure (Vendors, Operators)
The Challenge: Covering relevant use cases for SW-IP

- With adequate protection paradigms (diversity of products in mind)
- Identifying common methods, entities, components, ...

- balancing crucial trade-offs ...
  - Re-usability vs. efficiency
  - Functional split between security layers
  - Reasonable mix of methods
  - Availability of a solution/HW vs. appropriateness for specific context
  - Cost of implementation vs. achievable security level
Identified SW-IP Use Cases for NE

- **SW Delivery / Distribution**
  Protect for receive-time validation
- **SW stored on repository**
  Long-time storage / versioned and revoked SW
- **SW installation**
  Update and installation, including 'download to install'
- **Boot time protection**
  - **Trusted boot** Remote attestation, trusted reporting (TCG based)
  - **Secure boot** Autonomous validation / authorized SW, as for HeNB / eNB
- **Runtime protection**
  - Load-time: each time before SW is used
  - Store-time: while SW remains installed
  - Execution-time (in memory): while SW is executed

**Meta-Requirements** (not directly security related)
- Applying same security infrastructure for all UC (harmonization)
- Covering wide range of products (NE), including virtualization
- Inducing only minimal efforts in standardized operator network
`Security Framework’
... stable for > 20 years

PKI / Signature based SW Signing Paradigms
- Perfectly allow fulfilling 3GPP SA3 requirements for eNB, HeNB ...
  and to assume a manufacturer’s responsibility
- Significantly, ease validation / enforcement (enabling autonomy)
- Allow building a common backbone for all SW-IP methods
- Enable efficient and flexible trust and SW management from remote
- Can be built on well established, stable, high level E2E security frameworks

Implementation & Infrastructure (Protection)

Verification & Enforcement

Reporting & Evaluation

Copyright © 2010-2013 ASMONIA Project. All rights reserved
For propagating SW-IP (failure) messages for NE through network ...

- Message structure defined, but ‘proprietary’ format can be used (e.g., as in ‘MIB’)
- Messages reflect cause, origin, object, time ..., other context specific information
- Central conversion via manufacturer-specific tools/scripts
- Existing protocols allow transport and even ‘sessions’ at NE – NE_EM level
- For message aggregation and routing existing OAM structure can be re-used
Verification and Enforcement

Boot and RT-Protection for NE

- Typically, operation time for NE can be very long (weeks .. months)
- Re-boot may only be done after SW/FW – updates
- Attestation only meaningful directly after boot ...
- Remote location: Recovery by service in field may become very expensive
- Autonomous self-defense including the runtime is a practicable way-out
HW / PROM based Root of Trust
(..for signature/PKI based protection scheme)

Authorized Flash Control Processor (AFCP)

- Memory controller protecting firmware and assets (R-CA cert, keys, CRLs, policies,...)
- ‘ROM “ for system, but ‘PROM’ for AFCP - needs owner(s) authorization
- De-couples access to mutable assets from system CPU / SW
- Autonomously executes validation / updates / reaction (e.g., key-lock/unlock)
- Provides autho. remote trust management and security functions (local API, HW)
- Implementable in different ways (PCB, FPGA, MCM, ASIC, ...)

'read only' for non secret data
Load-time protection (Linux)

- Event triggered, file based load-time protection through sys-call interception
- Verification & Enforcement implemented into Kernel (e.g., LSM hooks)
- Can block invalid resources and logs/sends appropriate message
- Applicable to any file-type via (remotely) signed database
- Harmonizing with SW creation processes
- Recommended to combine with
  - secure boot (e.g., based on AFCP) and verified installation
  - efficient hardening concept to prevent SW-IP mechanisms against CFI attacks

*HV/K protection {2} \ {1}*

Signed Database contains signatures, policies & references

**Applications accessing resources from FS**

- LSM hooks: Here we can monitor any file access! (Also enables host protection)
Load-time protection
(for KVM/QEMU based virtualization)

• Verification / enforcement via Kernel or Hypervisor (acting as NFS server)
• For reasonable security no modifications in guest required (but in HV kernel)
• For extended security modifications of VM + hardening are useful
• Trade-off: SW-IP well protected in HV, but loss of semantic information (acceptable if guest is ‘well known’, such as an NE)

- HV/K protection \{2\} \setminus \{1\}
- Guest protection \{2\} \cup \{1\}

NFS ...
• Setup
  - Root FS mounted via NFS
  - all other FS disabled by QEMU
• Guest booted and serviced only through NFS
• Uses RPC calls to talk with remote system (here: host)
• NFS (v4) calls defined in RFC 3530 (newest)

EXAMPLE 3
‘NE Integrity’ provides

- Dedicated approaches to improve implementation concepts for NE integrity & hardening, focusing on Linux solutions in mobile NW environment
  - for HW based roots of trust
  - for SW integrity protection, also covering long-term operation and virtualization
- Integration concepts comprising
  - Optimized, common cryptographic ‘backbone’
  - Integration into mobile NW security/OAM infrastructure
  - Generation and propagation of ‘anomaly messages’ to enable assessment ‘sanity state’ at network level
References


