Center for Embedded Systems

An NSF Industry/University Cooperative Research Center

Verifiable Firmware Update Mechanisms for Embedded Systems

Ning Weng, SIUC

Spyros Tragoudas, Joseph Lenox, Veeresh Dandur

SIU Southern Illinois University Carbondale

National Science Foundation

ASU Ira A. Fulton Schools of Engineering Arizona State University

Project Overview and Description

- **Project Description**
  - Verify/Validate software running on remote untrusted ES
  - Empower ES to determine whether software updates are authorized

Project Overview and Description

- **Current mechanisms**
  - Public key infrastructure
  - Secure hardware extension
- **Challenges**
  - Embedded systems constraints: resource, connectivity
  - Operational issues: multiple independent root authorities, certification revocation,

Practical integrity solution for constrained ES w/existing mechanisms

Approach

- **Authority-based**
  - Software signature
    - Local verification
    - Whitelist
  - Public Key Infrastructure
    - ex: X.509
  - Challenge
    - Network connectivity

Approach

- **Host-assisted**
  - Rule/Anomaly-based
  - Rules are preloaded
  - Challenge
    - Securing host monitoring
  - Existing Hardware extension solution
    - Secure hardware extension (SHE) on Freescale MPC564xs

Project Tasks/ Deliverables

	Description	Date	Status
1	Review adversary attack methods and verification mechanism specification	10/13	
2	Authority-based solution	1/14	
3	Host-based solution, acquire candidate systems for platform evaluation	7/14	
4	Prepare final report	8/14	

Technical Detail

- **Authority-based**
  - Store whitelist on remote authority
  - ES has the ability to verify the signature of incoming changes and compare
  - ES has access to an asymmetric key infrastructure & write-protected storage

Technical Detail

- **Protocol of Authority-based**
  - ES sends challenge to ST.
  - ST forwards challenge to remote authority, along with a hash of firmware.
  - Remote authority compares provided hash with a whitelist and prepares a response of the nonce & approved hash encrypt with private key & sends to ST.
  - ST forwards response to ES.
  - ES decrypts response and verifies.

Technical Detail

- **System/infrastructure requirements of Authority-based**
  - Sufficient RNG on ES
  - Write-protected memory on ES
  - CPU capability to generate hash on ES

Technical Detail

- **Variation of authority-based solution**
  - ES share symmetric key with remote authority
  - ES has public key of remote authority
  - ST share symmetric key with remote authority

Technical Detail

- **Protocol: Variation of authority-based**
  - ST has to prove its identity to remote authority
  - Remote authority encrypts “Firmware” with private key and “Signs” it. Passed to ST.
  - ES gets the firmware from ST, decrypts the contents, verifies signature.
  - If signature is verified, the firmware is updated.

Technical Detail

- **Host-monitoring**
  - Rule/Anomaly based approach
    - Rules are preloaded before deployment and are securely stored.
    - The chance of rules change is very small.
    - This approach will have special hardware requirement to ES.

Technical Detail

- **Current Support Mechanisms**
  - Identity - X.509 Certificates
  - Key Storage and Crypto - Secure Hardware extension (SHE)

Technical Detail

- **X.509 Certificates**
  - Tool for PKI, binds public keys for a host/device to an established name
  - *Root* authority
  - Security industry standard
    - Several terms ill-defined
  - Designed for computer networking applications
  - Allows for revocation/expiration

Technical Detail

- **X.509 Limitations**
  - Does not match ES requirements well
    - Multiple Independent Root Authorities
    - Certificate revocation / expiration
      - Low connectivity in ES applications
        - » Difficult to ascertain current status
      - Possible time resets due to malfunction
    - Certificate Attributes
      - Limited to particular applications and hierarchy

Technical Detail

- **Multiple Independent Root Authorities**
  - Certificate authorities (CA) are independent and accessible to users
  - In real time, ES don't have access to CA, so there is no update of credentials
  - Multiple independent root authorities was proposed as solution.

Technical Detail

- **Secure Hardware Extension**
  - Specification for hardware cryptography tools and support structure
    - Enable Secure zone with hardware
    - Protect cryptographic keys from software attacks
    - Support authentic software environment
  - Cryptographic Service Engine

References

1. Ang Cui, Michael Costello, Salvatore J. Stolfo; "When Firmware Modifications Attack: A Case Study of Embedded Exploitation;" NDSS 2013; 2013
2. Basile, C.; Carlo, S.D.; Scionti, A., "FPGA-Based Remote-Code Integrity Verification of Programs in Distributed Embedded Systems," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.42, no.2, pp.187--200, March 2012
3. SHE: Secure Hardware Extension, https://www.escrypt.com/fileadmin/escrypt/pdf/WEB_Secure_Hardware_Extension_Wiewesiek.pdf. . [Accessed 12 April 2013].
4. Mao, S.; Wolf, T., "Hardware Support for Secure Processing in Embedded System," IEEE Trans. Computers 59(6): 847-854 (2010)
5. Freescale, " MPC564xS: Qorivva MCU for instruments cluster," 2013. [Online]. Available: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MPC564xS. [Accessed 2 April 2013].
6. S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady. Security in embedded systems: Design challenges. ACM Trans. Embed. Comput. Syst., 3(3):461–491, 2004.
7. OSSEC: Open source host-based intrsion detection system, http://www.ossec.net/ [Accessed 9 April 2013].
8. ZyTrax, Inc. (2013, Februrary) Survival guides - ssl/tls and x.509 (ssl) certificates. www. ZyTrax, Inc. [Online]. Available: http://www.zytrax.com/tech/survival/ssl.html
9. S. L. Geoff Emerson, Jurgen Frank. (2011, June) Using the Cryptographic Service Engine (CSE): An introduction to the CSE module. www. Freescale Semiconductor. [Online]. Available: http://cache.freescale.com/files/32bit/doc/app note/AN4234.pdf