SwapGuard: A Software-Only Solution for Attesting Hot-Swappable Devices in Power Grids

Xinshu Dong, Sumeet Jauhar, Binbin Chen

Advanced Digital Sciences Center
Singapore

SmartGridComm’16, Nov 7, Sydney
Hot-swappable Devices in Power Grids

Benefits:
• Maintainability
• High availability

One down side:
• A potential attack vector
Risk Associated with Hot-swapping

• Swapping-in a malware-infected device is a local attack assisted by innocent maintenance staff
  – How malware may come in: Field devices can be sourced from and handled by multiple parties

• A malware penetrates in this way bypasses all perimeter-based security defence mechanisms
Existing Ways to Control the Risk

• Intrusion detection / anti-virus solutions often require substantial computing resources and frequent updates of their databases
  – A host-based IDS itself may be compromised
  – A network-based IDS cannot deal with legitimately looking but malicious messages
Software Attestation

• Field devices run and only run known software images
  – This makes software attestation easier

• Challenges:
  – Lack of hardware-based Root of Trust, e.g., Trusted Platform Modules (TPMs)
Prover needs to return correct checksum results within an expected time.
Benefits of a Software-only Attestation Solution

• Low deployment overhead
  – No need to upgrade the hardware

• High assurance
  – Only expected legitimate code resides on the device memory
Roadmap

• Background

• **Problem Setup**

• Design of SwapGuard

• Evaluation results
System Model
Threat model

- The verifier is trusted

- The grid network (between verifier and proofer) is closed, and does not contain external devices

- The staff who installs the device is not malicious

- The swapped-in device may contain malware, but does not have altered or added hardware components
Challenges

• Heterogeneous boards
• No encryption in existing CAN bus protocol
• The need for low overhead and high availability
• Proxy attacks among interconnected devices
Roadmap

• Background

• Problem Setup

• Design of SwapGuard

• Evaluation results
SwapGuard Design

Defend pre-attestation attacks

Defend in-attestation attacks

Success of attestation ensures a malware-free new device

Attesting all devices to bootstrap the security
Defending pre-attestation attacks

• SwapGuard uses lightweight encryption (stream cipher HC128) to protect CAN bus communication
  – This prevents an unattested device from sending malicious commands/wrong data to other devices
Defending pre-attestation attacks (2)

• Even with encryption, an unattested board can still attack
  – DoS attack
  – Sending analog outputs

• We leverage an out-of-band channel and the assumption that the installer is not malicious to deal with such attacks
Defending in-attestation attacks

• For a new device to acquire the symmetric key, it needs to explicitly issue an admission request (in plaintext) to the verifier for attestation.

• For now let us assume all existing boards are malware-free.
Defending in-attestation attacks (2)

• Impersonation attack is possible:
  – a new device $d$ may pretend to be another existing device $d'$, and cheat $d'$ into performing the attestation
  – Once the attestation succeeds, the key is broadcast in the CAN network, so $d$ learns about the key
Defending in-attestation attacks (3)

• To deal with impersonation attack
  – If the new device is an I/O board: the bridging device will do a local broadcast to verify $d'$ indeed initiates the request
  – In case the new device is the bridging device: we require the admission request to the verifier to be encrypted with the current key, unless the request claims to be from the bridging device itself
Putting things together

New Device Swapped In
  └── Keep Silent?
      └── Y
          └── Absent Board Detected
              └── Detect Malicious Signals to IEDs
                  └── Prevented by Out-of-Band Channel
                      └── Failure Handling
  └── N
      └── Attack Right Away?
          └── N
              └── Request Attestation with Admission Request
                  └── Impersonate Another?
                      └── N
                          └── Attestation Succeeds?
                              └── N
                                  └── Malware-Free Obtain Session Key & Communicates with Other Devices
                                      └── Y
                                          └── Detected by Status-Inquiry / Lack of Session Key
                                              └── Prevented by Out-of-Band Channel
                                                  └── Failure Handling
                                                    └── Y
                                                        └── Detected by Bridging Device / Verifier's Polling
                                                            └── Prevented by Out-of-Band Channel
                                                                └── Failure Handling
                                                                  └── Y
                                                                      └── Detect Malicious Signals to IEDs
                                                                          └── Prevented by Out-of-Band Channel
                                                                              └── Failure Handling
To bootstrap the whole system

• We have discussed how to handle a single new board, while assuming all existing boards are malware-free

• How to ensure all existing boards are malware-free?
  – SwapGuard put all boards into attestation at the same time
Simultaneous All-board Attestation

- Unified checksum response time (*UTime*)
  - Recall that there can be heterogeneous types of devices, which are connected with heterogeneous networks (e.g., Ethernet, CAN bus)

- *UTime* characterizes the overall time needed for a device to respond with a correct checksum, including both computation and communication time
Simultaneous All-board Attestation (2)

- We make the UTime of all devices roughly equal
  - So all boards will be kept busy, and the faster ones cannot help the slower ones

- We control UTime by adjusting the number of checksum computation iterations of different devices
  - The iteration number needs to be greater than certain threshold
Roadmap

• Background

• Problem Setup

• Design of SwapGuard

• Evaluation results
Evaluation setup

• The main board has a WIZnet W5100 Ethernet chip that implements a fully hardwired TCP/IP stack. It supports 60 I/O modules connected via CAN bus.

• The I/O boards sample analog and digital data from the connected sensors and meters, send digital signals to IEDs, and communicate with the main board via the CAN bus.

<table>
<thead>
<tr>
<th>Board</th>
<th>Processor</th>
<th>CPU Clock</th>
<th>RAM</th>
<th>ROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>NXP LPC2292, ARM7TDI-S</td>
<td>60MHz</td>
<td>16KB</td>
<td>256KB</td>
</tr>
<tr>
<td>I/O</td>
<td>NXP LPC1756, Cortex-M3</td>
<td>100MHz</td>
<td>16/32KB</td>
<td>256KB</td>
</tr>
</tbody>
</table>
Time for attesting individual devices

Cumulative Distribution Function for One-board Attestation Time (in seconds)

Compared to the manual process of hot swapping a board, which usually takes more than 10 seconds, such an extra overhead of 1 ~ 2 seconds is likely acceptable.
Overhead of encrypting CAN messages

Cumulative Distribution Function for Round Trip Time (in milliseconds)

Without encryption

With encryption (HC128 stream cipher)
Time needed for all-board attestation

The total duration for attesting the 5 boards is only slightly longer than that for attesting the main board (2.14 vs 2.03 seconds)
Conclusion

• We identify key challenges in attesting hot-swappable devices

• We design SwapGuard, a software-only solution that provides a secure and efficient way to attest swapped-in devices

• We implement SwapGuard on hot-swappable boards of a remote terminal unit model that is used in power grid systems

• Our evaluation shows low overhead of SwapGuard
Thank you!

Contact: binbin.chen@adsc.com.sg