A Ridiculously Short Intro into Device Attestation

Dimitar Tomov, DesignFirst, Estonia
Ian Oliver, Nokia Bell Labs, Finland
Overview

• What is Attestation? (Ian, 10 mins)
• TPM and Measurements
  • How, What, Why
• Reporting and Quoting
  • Structure and Contained Data
• What to Attest
  • Identity, Firmware, Configuration, etc
• Rules
  • Attest, Verify, Decide
• Example (Dimitar, 10 mins)
  • Time Attestation for Network Monitoring
What is attestation?

- The process of providing evidence that something is true

Our system

- HSM / TPM
- Minimal Code Base
- OS
- Config
- User apps
- User data

Attestation Server

- Attested Data
- Control Layer
TPM and Measurements

How the **** do things boot and what gets measured?!
Reporting and Quoting
What to Attest?

- Quote Type/Magic
- Identity (signature and signer)
- Configuration
- Clock
- Firmware
- Nonce & Arbitrary Data
- History
# Attestation Rules

<table>
<thead>
<tr>
<th>TPM 2.0 Quote Rules</th>
<th>Attestation Timeliness Rules</th>
<th>History/Assurance Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it a quote?</td>
<td>Did the device respond to the quote request in a timely manner?</td>
<td>Has the device changed in any way since the last quote?</td>
</tr>
<tr>
<td>Signed and matches the qualified signer?</td>
<td>Did the device process the quote request in a timely manner?</td>
<td>...and for what properties?</td>
</tr>
<tr>
<td>Nonce + additional data correct?</td>
<td>Was the response consistent with network latencies (where applicable)?</td>
<td>Does the device verify against the selected rules for its LoA?</td>
</tr>
<tr>
<td>Does the attested value match the known good value?</td>
<td>-</td>
<td>What set of PCRs is required for a minimum LoA?</td>
</tr>
<tr>
<td>Is the device running the correct firmware?</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Has the device been rebooted?</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Is the clock increasing correctly?</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Was the device shutdown correctly?</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Who decides if it is trusted?

- CRTM/SRTM Measures from BIOS/UEFI/Firmware/ACM etc.
- TPM 2.0 Root of Trust for Reporting
- Attestation and Verification Services
- End user
Example – Time attestation

• How to attest time?
  • TPM time evidences
• How to attest identity?
  • TPM key based identity
• Choosing a TPM stack(library)
• Time attestation as a timestamp
• Attestation server
• Use cases
  • Data center
  • IoT fleet

Periodically attest
Tamper-proofed HW time
Verify
Decide
Motivation

• Establish identity & temporal trust across devices & systems
  • Attesting the TPM time and clock is a special case
    • Root of Trust for Reporting is the TPM
    • The data being attested is physically internal to the TPM (hint: Vs. Data is fed to the PCRs)
  • Periodic attestation

<table>
<thead>
<tr>
<th>Equipment or device type</th>
<th>Importance</th>
<th>Sampling period (of attestation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical infrastructure</td>
<td>High</td>
<td>1 minute</td>
</tr>
<tr>
<td>User stations, Nodes in the field</td>
<td>Medium</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Everything else</td>
<td>Low</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

- Internal attestation server (on-premise)
- External attestation server (3rd party)
GetTime vs Quote(PCRs)

- Data is fed to the PCR – How to guarantee what is being measured?
GetTime vs Quote (PCRs)

- TPM2_Quote gives an evidence
- But who creates the evidence?
- We need secure environment
- We need secure application

Vs

- TPM2_GetTime gives an evidence
- The evidence is created completely internally to the TPM
- No need of secure environment

Note: In both cases an action is needed when a fresh evidence is not received. It would be a form of denial-of-service (DoS) attack.
Choosing a TPM stack(library)

- Mature stacks vs New stacks

<table>
<thead>
<tr>
<th>TPM stack</th>
<th>Interface(s)</th>
<th>Attestation example</th>
<th>Embedded Systems use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infineon/Intel TSS</td>
<td>TCG spec. ESAPI, (soon) FAPI</td>
<td>No. Separate project, &quot;CHARRA&quot; by Fraunhofer</td>
<td>Yes for Linux-based systems</td>
</tr>
<tr>
<td>IBM TSS</td>
<td>Own rich API (ESAPI like)</td>
<td>Yes. &quot;IBM open-source attestation server(ACS)&quot;</td>
<td>Yes for Linux-based systems</td>
</tr>
<tr>
<td><strong>New</strong> Google Go-TPM</td>
<td>1:1 TPM commands + mild layer on top</td>
<td>Yes. &quot;Go-Attestation&quot;</td>
<td>Needs Golang for non-Linux embedded system</td>
</tr>
<tr>
<td><strong>New</strong> WolfSSL WolfTPM</td>
<td>Own rich API (wrappers) 1:1 TPM commands</td>
<td>Yes. Signed timestamp and local attestation</td>
<td>Baremetal and Linux-based</td>
</tr>
</tbody>
</table>
How to attest time?

• Trust the only IC in your system with physical tamper protection
• Use a standard TPM2.0 command TPM2_GetTime

• Get signed evidence of
  • Built-in hardware time
    • Current uptime of the TPM since the last power-on
  • Built-in hardware clock
    • Total time the TPM has ever been on
  • Reset counter
    • How many times the system has been rebooted since a TPM clear (i.e. provisioning)
Time attestation as a timestamp

- TPM uptime since last power-up
  - 44 hours 57 minutes

- Total time the TPM has been on
  - 57 days 8 hours 52 minutes

- Reset count
  - 17 power cycles
What is in the TPM signed time evidence?

• Standard TPM-generated attestation block with
  • TCG defined data structure called TPMS_TIME_ATTEST_INFO
  • TPMT_SIGNATURE holding the signature over the data

TPMS_TIME_ATTEST_INFO

  firmware version

TPMS_TIME_INFO

  time

TPMS_CLOCK_INFO

  Clock
  ResetCount
  RestartCount
  Safe
How to attest identity?

• Use a standard TPM2.0 command TPM2_Create

• Create asymmetric key pair known as “Attestation Key” (AK)
  • Private part can be used only by the TPM that created the AK
  • Public part naturally used to verify the evidence signature and decrypt
  • Possible to have a certificate authority and have rolling AK
  • Possible to have anonymous attestation for privacy reasons

NB: The AK is a key generated from the TPM that cannot be migrated between TPMs. Internally, the TPM can use AK only for signing specific TPM-generated structures. No other keys have this property. Therefore, the EK and AK are effectively a unique identity for that TPM.
Attestation Server

Decision making

- Slicing between trusted and untrusted information
  - TPM attested data: TPM time, TPM clock, TPM reset counter
  - On-premise network monitoring data
  - Third party data from cloud monitoring
Use cases

Data center
Limited trust in the HW and software vendor
Allows to verify the maintenance periods
Allows to verify the network and monitoring data
End users and customers can have digital trust in their rented or cohosted servers.

IoT Fleet
Improves security for Edge devices with high risk of physical tampering
Helps protect maintenance and battery indicator
Generating rich attestation data on IoT devices is not possible or it is expensive
Set of trust qualities needed
TLDR

• A system can have **trusted qualities** instead of being 100% trusted
• Cross-referencing different types of attestation data can provide evidence for trusted qualities.
• The decision of whether a device is trusted is not responsibility of the attestor and verifier – these just gather and check the evidence.
• TPM time attestation can be trusted without trust in the system.
• Multiple attestation servers (including external attestation servers) are useful for cross-checking attested data.
Contact us for more information

Dimitar Tomov, DesignFirst
• https://tpm.dev
• Twitter @tomov_eu

Ian Oliver, Nokia Bell Labs
• https://www.bell-labs.com/usr/ian.oliver
• Twitter @i_j_oliver