TCG TPM2 SOFTWARE STACK
& EMBEDDED LINUX

Philip Tricca
philip.b.tricca@intel.com
AGENDA
Background
• Security basics
• Terms
TPM basics
• What it is / what it does
• Why this matters / specific features
TPM Software Stack
• Architecture / Design
• Getting Started
• Getting Results
LEVEL SET

There is no magic, there are no silver bullets

• “security” takes the whole village

• Architecture to implementation to maintenance

• There is no such thing as “a secure system”, only secure enough

• Ideally the informed CUSTOMER defines “secure enough”
THE BASICS

Using the TPM does not a secure system make

• Disable services / exclude tools / minimize attack surface
• Use writable storage only when you must
• Regular updates, automatic updates! SIGNED UPDATES!
• Mandatory access control (SELinux!)
• Increase complexity in system, increase level of effort to secure it
  • Securing general purpose computers is a nightmare
  • Embedded systems -> security is more tractable
THREAT MODELING

A process by which we identify & document

• Assets
• Threats to them

• Prioritize: decide where your efforts are best spent
  • Identify trade-offs

• Accurately describe the properties of your system
  • What it protects against: risks mitigated
  • What it does not: risks accepted
  • And most importantly: why
IF YOUR TEAM DOESN’T MODEL THREATS ...

Please do?

• Much of the body of knowledge was developed in Microsoft
• MSDN has lots of free content
• OWASP Application Threat Modeling
  • https://www.owasp.org/index.php/Application_Threat_Modeling
• Adam Shostack’s book was my introduction (2014)
TERMS

Classic security concepts:

• Confidentiality
• Integrity
• Authentication
• Authorization (satisfy TPM2 policy)
• Non-repudiation

Use the TPM2 to build systems that implement these principles
**WHAT IS A TPM?**

Small Crypto Engine
- Cryptographic functions
- Hashing functions
- Key generation & protection
- RNG
- Integrity measurement / reporting

---

**Random # Generation**

**Key Generation**

**Power Mgmt**

**Execution Engine**

**Non-Volatile Memory**
- Hierarchy Seeds
- Monotonic Counters
- Storage

**Volatile Memory**
- PCR banks
- Transient Objects
- Sessions

**Authorization**

**Mgmt Operations**

**Symmetric Engine(s)**

**Hash Engine(s)**

**Asymmetric Engine(s)**

**I/O**
TPM2 IMPLEMENTATION: DOMAIN SEPARATION

Discrete IP Block (a chip)
- Shielded Location
- ...
- ...
- I/O
  
  BUS

Protected Capability
  
  ...

Integrated IP Block
- Shielded Location
- ...
- ...
- I/O

OS

Apps

Protected Capability

I/O

IP block

IP block
TPM PROTECTIONS

Documented in TPM Rev 2.0 Part-1: Architecture

• Frames protections offered by TPM2 in section 10:
  • Protected Capability
  • Shielded Location
  • Protected Object

• Protected capabilities must TPM severely memory constrained
  • offload storage to application / Resource Manager
  • encrypt protected objects when not in shielded location

• Nature of physical security protections dictated by customer
INTEGRITY: MEASURED BOOT
Platform Configuration Register (PCR) & the “Extend” operation

- PCR is a Shielded Location, Extend operation is Protected Capability
- PCR is volatile memory capable of holding hash value
- Typically 24 PCRs in a TPM, addressed with index: PCR[0] – PCR[23]
- PCR usage (hashes of components) defined in TCG platform specs

Software Measurement is synonymous with the hash produced

- Extend hash of object (executable, config etc) into PCR
- Extend: $\text{PCR}[0]_N = H(\text{PCR}[0]_{N-1} | X)$
- Requires hash function: computationally infeasible to forge, easy to verify
# TCG TPM2 Software Stack: Design Goals

<table>
<thead>
<tr>
<th>System API (SYS)</th>
<th>Enhanced SAPI (ESYS)</th>
<th>Feature API (FAPI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 1:1 mapping to TPM2 commands</td>
<td>• 1:1 mapping to TPM2 Commands</td>
<td>• File IO</td>
</tr>
<tr>
<td>• No</td>
<td>• Additional commands for utility functions</td>
<td>• Requires heap</td>
</tr>
<tr>
<td>– file IO</td>
<td>• Provides Cryptographic functions for sessions</td>
<td>• Must be able to do retries</td>
</tr>
<tr>
<td>– crypto</td>
<td>• No file IO</td>
<td>• Context based state</td>
</tr>
<tr>
<td>– heap</td>
<td>• Requires heap</td>
<td>• Must support the possibility of reduced application code size by offering static libraries</td>
</tr>
</tbody>
</table>

TPM Command Transmission Interface (TCTI)
• Abstract command / response mechanism
• Decouple APIs driving TPM from command transport / IPC

TPM Access Broker and Resource Manager (TABRM)
• Power management
• Potentially no file IO – depends on power mgmt.

• Abstract Limitations of TPM Storage
• No crypto

• No crypto
• No heap, file I/O
TPM2 SOFTWARE STACK

System API & TCTI specification

- **TPM2 Command Transmission Interface (TCTI)**
  - Abstraction to hide details of IPC mechanism
  - libtcti-device & libtcti-socket
  - Adds flexibility missing from 1.2 TSS

- **System API (SAPI)**
  - Serialize C structures to TPM command buffers
  - One-to-one mapping to TPM commands (all 100+)
  - Minimal external dependencies: libc
  - Suitable for highly embedded applications / UEFI
Intel implementing TCG TSS as Open Source

- Project hosted under '01.org' on Github
  - https://github.com/01org/tpm2.0-tss
  - https://github.com/01org/tpm2.0-tools
- 3-clause BSD == maximum flexibility
- Development on GitHub “in the open”
  - I don’t always have the answer, someone else may though
  - Packages working their way into distros
- Lots of churn in the next few months
EMBEDDED BUILDS

My personal OSS work

- meta-measured [https://github.com/flihp/meta-measured](https://github.com/flihp/meta-measured)
  - TPM1.2 & 2.0 packages
  - Reference ‘live’ images & initrds
  - Grub2 patches extend measured launch (soon obsoleted by upstream!)
  - + BSP for Minnowboard Max to add TPM2 support as MACHINE_FEATURE
- Working on ARM reference platform + Infineon SPI TPM
  - Still some work in TSS code to support big-endian systems (facepalm)
TPM requires RNG for key creation, nonce generation.

- an entropy source and collector
- mixing function (typically, an approved hash function)
- Differentiation between TPMs w/ certification (NIST SP800-90 A)
- TPM RNG integrated with Linux kernel RNG
  - If you need an entropy source DO NOT use TPM RNG alone
  - Load the ‘tpm_rng’ kernel driver & setup rng-tools
  - Use /dev/(u)?random
  - [https://scotte.org/2015/07/TPM-for-better-random-entropy](https://scotte.org/2015/07/TPM-for-better-random-entropy)
USE CASE: CRYPTO OPERATIONS

TPM2 for basic crypto: sign / encrypt / hash

- HMAC required for authorization
- Asymmetric algorithm, RSA 2k for compatibility, usually ECC
- See Davide Guerri’s blog for a great howto: https://dguerriblog.wordpress.com/2016/03/03/tpm2-0-and-openssl-on-linux-2/
  - tpm2_getpubek: create TPM2 primary key & export pub & name
  - tpm2_getpubak: create TPM2 signing key & export pub & name
  - tpm2_hash: hash some file / data & generate ticket
  - tpm2_sign: use key (from getpubak) to sign hash
USE CASE: SEALED STORAGE AKA LOCAL ATTESTATION

TPM2 policy authorization as access control on TPM protected object

• Microsoft Bitlocker uses this mechanism for disk crypto keys
• OpenXT virtualization system uses similar mechanism
• Assumes measured boot records TCB in PCRs: software identity
  • Create TPM object holding auth data for disk crypto
  • Bind object to PCR policy: select PCRs based on TCB & requirements
  • On successful boot w/ PCRs in expected state, load object
  • Can be used to hold secrets for LUKS volumes
SHOUT-OUTS!

Many thanks for contributions to materials:

• Monty Wiseman @ General Electric
• Andreas Fuchs @ Fraunhofer SIT
• Lee Willson @ Security Innovation

& Everyone who’s contributed code / answered questions on GitHub!

• Bill Roberts @ Intel OTC
• Imran Desai @ Intel IOTG
THANKS!
Threat Modeling: Designing for Security – Adam Shostack

Trusted Platforms UEFI, PI and TCG-based firmware
• https://people.eecs.berkeley.edu/~kubitron/cs194-24/handouts/SF09_EFIS001_UEFI_PI_TCG_White_Paper.pdf

Open Security Training Trusted Computing Module:
• http://opensecuritytraining.info/IntroToTrustedComputing
RESOURCES(2)

Davide Guerri TPM2.0 talk @ FOSDEM
• https://fosdem.org/2017/schedule/event/tpm2/

TPM RNG linux howto:
• https://scotte.org/2015/07/TPM-for-better-random-entropy