Trusted Platform Module (TPM)
What’s in a TPM?
Volatile Memory

- PCR Banks
- In-use keys
- Sessions
- Etc.
Platform Configuration Register (PCR)

- Contain hashes of programs
- Modification: TPM2_Extend()
  \[ PCR_{new} = H(PCR_{old} || data_{new}) \]
- Attestation: TPM2_Quote()
Non-Volatile Memory

- Platform Seed
- Endorsement Seed
- Storage Seed
- Monotonic Counters
- Etc.
Seeds used for attestation, etc

- Endorsement seed used to derive endorsement keys (EKs)
- Manufacturer attests validity of EKs
- EKs used to attest other TPM-derived values including
  - other keys: TPM2_Certify()
  - audit logs: TPM2_GetSessionAuditDigest()
Binding and Sealing
Bitlocker

Enter the password to unlock this drive

Press the Insert key to see the password as you type.
ARM TrustZone
Android Full Disk Encryption
Android Full Disk Encryption

Android FDE's KDF
SGX

Skylake
Enclaves

- Isolated computing environments
- Access hardware-derived keys
- Provide
- Provide local and remote attestation
Isolation
Sealing
SGX in use
Vulnerabilities

Trustworthiness of Trusted Code

Side Channels

Algorithm 1 Fixed-window exponentiation

**Input:** $a, e, N \in \mathbb{N}$

**Output:** $x \leftarrow a^e \mod N$

1. Precompute $g[i] \leftarrow a^i$ for $1 \leq i \leq 2^k$
2. Let $e = (e_j, e_{j-1}, \ldots, e_1, e_0)$ be the base $2^k$ representation of the exponent $e$ with $e_j \neq 0$
3. Initialize $x \leftarrow e_j$
4. for $i \leftarrow j - 1$ down to 0 do
5. \hspace{1em} $x \leftarrow x^{2^k} \mod N$
6. \hspace{1em} if $e_i \neq 0$ then
7. \hspace{2em} $x \leftarrow g[e_i] \cdot x \mod N$
8. \hspace{1em} end if
9. end for
## Comparison of Hardware Solutions

<table>
<thead>
<tr>
<th>Adversary</th>
<th>Attack</th>
<th>TPM</th>
<th>TrustZone</th>
<th>SGX</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>direct probing</td>
<td>n/a (OS measured)</td>
<td>access checks on TLB misses</td>
<td>Access checks on TLB misses</td>
</tr>
<tr>
<td>OS</td>
<td>page faults</td>
<td>n/a (OS measured)</td>
<td>secure page tables</td>
<td>X</td>
</tr>
<tr>
<td>OS</td>
<td>cache timing</td>
<td>n/a (OS measured)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Another container</td>
<td>direct probing</td>
<td>n/a</td>
<td>n/a (secure world trusted)</td>
<td>access checks on TLB misses</td>
</tr>
<tr>
<td>Another container</td>
<td>cache timing</td>
<td>n/a</td>
<td>n/a (secure world trusted)</td>
<td>X</td>
</tr>
<tr>
<td>Peripheral</td>
<td>DMA</td>
<td>X</td>
<td>bus bounces accesses</td>
<td>IOMMU bounces DMA</td>
</tr>
<tr>
<td>Physical attacker</td>
<td>Physical DRAM</td>
<td>X</td>
<td>n/a (on-chip SRAM only)</td>
<td>memory encryption</td>
</tr>
</tbody>
</table>
So where are we?
The Bigger Picture

Attacks are perpetrated by threats that inflict harm by exploiting vulnerabilities which are controlled by countermeasures.
Threats

A principal that has potential to cause harm to assets

- **Adversary** or **attacker**: a human threat, motivated and capable
- Sometimes humans aren't malicious: accidents happen
- Sometimes non-humans cause harm: floods, earthquakes, power outage, hardware failure
Threats
Vulnerabilities

- Session Management: 64%
- Cross-Site Scripting (XSS): 62%
- Web Server Vulnerabilities: 41%
- Authentication and Authorization: 61%
- Cross-Site Request Forgery (CSRF): 23%
- SQL Injection: 29%

(DIRTY COW)
Attacks

Procedure A

... call B ...

B Param 3
B Param 2
B Param 1
Ret Addr Ptr A
Stack Ptr A
B Local Var 1
B Local Var 2
C Param 2
C Param 1
Ret Addr Ptr B
Stack Ptr B

Procedure B

...

call C ...

Procedure C

...

Return-Oriented Programming

is a lot like a ransom note, but instead of cutting out letters from magazines: you are cutting out instructions from text segments.

Bot
Bot
Bot
Bot
Bot
C&C

Internet

Victim
Functional Goals -> Harms -> Security

Confidentiality
Integrity
Availability

CIA
Countermeasures
And now...

WE'VE BEEN TRYING FOR DECADES TO GIVE PEOPLE GOOD SECURITY ADVICE. BUT IN RETROSPECT, LOTS OF THE TIPS ACTUALLY MADE THINGS WORSE.

MAYBE WE SHOULD TRY TO GIVE BAD ADVICE? I GUESS IT'S WORTH A SHOT.

SECURITY TIPS

(PRINT OUT THIS LIST AND KEEP IT IN YOUR BANK SAFE DEPOSIT BOX)

- DON'T CLICK LINKS TO WEBSITES
- USE PRIME NUMBERS IN YOUR PASSWORD
- CHANGE YOUR PASSWORD MANAGER MONTHLY
- HOLD YOUR BREATH WHILE CROSSING THE BORDER
- INSTALL A SECURE FONT
- USE A 2-FACTOR SMOKE DETECTOR
- CHANGE YOUR MAIDEN NAME REGULARLY
- PUT STRANGE USB DRIVES IN A BAG OF RICE OVERNIGHT
- USE SPECIAL CHARACTERS LIKE & AND %
- ONLY READ CONTENT PUBLISHED THROUGH TOR.COM
- USE A BURNER'S PHONE
- GET AN SSL CERTIFICATE AND STORE IT IN A SAFE PLACE
- IF A BORDER GUARD ASKS TO EXAMINE YOUR LAPTOP, YOU HAVE A LEGAL RIGHT TO CHALLENGE THEM TO A CHESS GAME FOR YOUR SOUL.