6. Trustworthy Computing
ENEE 757 | CMSC 818V

Prof. Tudor Dumitraș
Assistant Professor, ECE
University of Maryland, College Park

http://ter.ps/757
https://www.facebook.com/SDSAtUMD

Today’s Lecture

• Where we’ve been
  – Crypto basics
  – OS protection mechanisms
  – Network security

• Where we’re going today
  – Trusted computing

• Where we’re going next
  – Guest lecture: Ksenia Dmitrieva, Cigital – Password storage
  – Next week: Password security
A Note on Critique Submissions

• Pay attention to assignment numbers
  – 10 + number of critique .bib files so far
    • <PATH>/submit 2015 fall <COLLEGE> <COURSE> 0101 11 os_protection_mechanisms.bib
    • <PATH>/submit 2015 fall <COLLEGE> <COURSE> 0101 12 network_security_basics.bib
    • <PATH>/submit 2015 fall <COLLEGE> <COURSE> 0101 13 trustworthy_computing.bib
    • <PATH>/submit 2015 fall <COLLEGE> <COURSE> 0101 14 passwords.bib

• Use course-specific submit (instead of the general GRACE submit)
  – /afs/glue.umd.edu/class/fall2015/enee/757/0101/bin/submit
  – Parses your submission and leaves a copy in your home dir

Who Can You Trust?

• Where is the request “from”?
  – The user? The workstation? The application? The network channel? All of the above?
  – Which of these actors do you trust?
Trusted Computing Base (TCB)

• TCB = the collection of hardware and software that security depends on
  – Thus, TCB should be as small as possible
    • Must be carefully tested, analyzed and protected
    – “Trust” is something that is bad and must be avoided when possible

• Trusted components may operate alongside untrusted components
  – Example: you cannot trust every host connected to the Internet

• A trustworthy system or component is one that will not fail
  – “Fail secure” or “fail safe”: abort operation when uncertain where the request or inputs come from
  – “Trusted computing” really means “trustworthy computing”

Trusted Computing Group (TCG)

• Historically, PC manufacturers have chosen flexibility over security
  – User can modify the PC in any way she likes
  – PC does not have hardware protection for boot procedure, does not validate O/S before loading it, does not validate other programs

• Today this is changing with efforts like the Trusted Computing Group (TCG)
  – www.trustedcomputinggroup.org
  – Alliance formed in Jan 1999 by Compaq, HP, IBM, Intel & Microsoft
  – More than 150 companies by 2002
  – Developing a standard for a “trusted platform” (TP)
  – Scope of specs is at hardware, O/S and BIOS levels
    • Main spec released in Aug 2000 (v1.0) and Feb 2001 (v1.1)
    • PC-specific spec released in Sep 2001
**Trusted Platforms**

- Some goals of TP
  - Enable local and remote users to obtain reliable information about the software running on the platform
  - Provide a basis for secure key storage
  - Enable conditional release of secret information to the TP based on the software running

- TP enabled by a “trusted processing module” (TPM)
  - A hardware processing component that is isolated from software attacks and at least partially resistant to hardware tampering

- Each TPM is equipped with a different private key $K_{TPM}^{-1}$
  - And a certificate signed by the manufacturer (more on this later)

---

**TCG “Roots of Trust”**

The standard specifies two logical “roots of trust”

- **Root of trust for measurement (RTM):** A platform-dependent component that starts *measurement* of software running
  - Measurements are hashes of the platform state (e.g. applications started & configuration files used)
  - In practice, this means that the RTM must run first (or everything that is run before it is trusted)
    - e.g., BIOS boot block, called the “core root of trust for measurement” (CRTM)

- **Root of trust for reporting (RTR):** A platform-independent component that stores measurements as they happen, in such a way that measurements cannot be *undone*
  - RTR is implemented by the TPM
TPM Platform Configuration Registers

- TPM (version 1.1) contains sixteen 20-byte **platform configuration registers** (PCRs)
  - 20 bytes in order to store a SHA-1 hash value
- Each PCR records the last in a **chain of hashes** of the software that has been loaded and executed

![Diagram of TPM Platform Configuration Registers]

- PCR is updated before newly loaded software gets control
  - This is called **extending the PCR**
- PCR cannot be erased except by reboot (or **SKINIT** / **GETSEC[ENTER]** instruction in v1.2 TPMs)
- In this way, PCR contains record of software running

---

TCG Authenticated Boot

![Diagram of TCG Authenticated Boot]

- BIOS boot block (CRTM)
- BIOS
- ROMS
- OS loader
- OS
- OS components

![Diagram of TCG Authenticated Boot]
**TCG Secure Boot**

- Non-volatile “data integrity registers” (DIRs) are loaded with expected PCR values
  - DIRs are contained within TPM and require owner authorization to write

- If a PCR value, when computed, doesn’t match corresponding DIR value, then boot is canceled

---

**TCG Integrity Challenge and Response**

- Remote machine can query TPM for contents of PCRs

- TPM responds with signed PCR values
  - Signed with Attestation Identification Key (AIK) $K_{AIK}^{-1}$
  - The AIK pair is generated by the TPM
    - The AIK is endorsed by a certification authority

- TP additionally responds with records (hints) of what is “summarized” in the PCR values
  - Records could contain software itself, but more likely contains name, supplier, version, and URL for software
  - Enables remote machine to reconstruct and check PCR values
  - Records not trusted and so are stored outside TPM

- This is called **remote attestation**
TPM Sealed Storage

- Data encrypted with RSA key stored in the TPM
  - Slow public-key crypto

- **TPM seal**: bind data to current software state
  - Content of PCRs is encrypted along with the data

- **TPM unseal**: decrypt the sealed data
  - TPM will decrypt the data only if the PCRs contain the same values
  - Ensures that encrypted data can be trusted

- Applications:
  - Full Disk Encryption (FDE)
  - Digital Rights Management (DRM)

Full Disk Encryption: BitLocker Architecture

- **Static root of trust**: measurements of early boot components
**Disk Layout and Key Storage**

Operating System Volume
Contains
- Encrypted OS
- Encrypted page file
- Encrypted temp files
- Encrypted data
- Encrypted hibernation file

Where's the Encryption Key?
1. SRK (Storage Root Key) contained in TPM
2. SRK encrypts the VMK (Volume Master Key)
3. VMK encrypts FVEK (Full Volume Encryption Key) – used for the actual data encryption
4. FVEK and VMK are stored encrypted on the Operating System Volume

**Attacks Against Software FDE**

[Halderman et al., 2008]
- Cold boot attack on secrets in memory
- DRAM does not immediately lose its value upon power-off
- This can be used to extract BitLocker keys if done skillfully
After 5 seconds

Source: Lest We Remember: Cold Boot Attacks on Encryption Keys. J. Alex Halderman et al. 2008 USENIX Security Symposium

After 30 seconds
After 60 seconds

After 5 minutes
Late Launch (TPM v. 1.2)

- **Static root of trust**: must reboot to restart measurement
- **Dynamic root of trust** (late launch): properties similar to reboot
  - Without a reboot!
  - Removes many things from TCB
    - BIOS, boot loader, DMA-enabled devices, ...
    - Long-running OS and Apps if done right

- Modern CPUs provide special instructions that interact with TPM chip
  - AMD SVM: **SKINIT** instruction
  - Intel TXT: **GETSEC**[**SENTER**] instruction

- Initial intent: virtualization support
  - Virtual machine monitor (VMM) can be measured (MVMM)
  - Integrity of loaded code can be attested
  - Untrusted legacy OS can coexist with trusted software

Motivating Example: OPM Data Breach

- Ongoing remote attack against Office of Personnel Management (OPM) detected in April 2015
  - Data of 4.2 million Federal government employees stolen
  - Including security clearance investigations
  - Hackers gained valid credentials to the OPM systems => full disk encryption would not have prevented the data breach

- What if...
  - A compromised OS cannot learn the sensitive data
  - Only essential code can access the data
    - Decrypt stored data
    - Update it
    - Encrypt it back
    - Report result of operation
  - Can remotely verify this is so
Flicker Discussion

Is Flicker Enough?

- Many components manipulate user data
  - Want to reason about security properties based on all these components
- Web applications evolve
  - Cannot expect clients to know the most recent hash
- Logical attestation (Nexus)
  - Uses authorization logic (more on this later)
Review: Security Principles
[J. Saltzer & M. Schroeder, SOSP 1973]

- **Economy of mechanism**: Keep the protection mechanism as simple and small as possible
- **Fail-safe defaults**: Base access decisions on permission rather than exclusion
- **Complete mediation**: Check every access to every object
- **Open design**: Do not keep the design secret
- **Separation of privilege**: Require two keys to unlock, not one
- **Least privilege**: Grant every program/user the least set of privileges necessary to complete the job
- **Least common mechanism**: Minimize the amount of mechanism common to more than one user and depended on by all users
  - This principle talks about confinement
- **Psychological acceptability**: Design interfaces for ease of use
- **Work factor**: Increase the cost of circumventing the protection mechanism, compared to the attacker’s resources
- **Compromise recording**: Reliably detect that a compromise has occurred

Review of Lecture

- What did we learn?
  - Trusted Platform Module (TPM)
  - Remote attestation
  - Sealed storage

- Sources
  - Mike Reiter, Jonathan McCune

- What’s next?
  - Guest lecture: Ksenia Dmitrieva, Cigital

- Deadline reminders
  - Pilot project reports due in one week
Backup slides
Does program P compute F?
Is F what the programmer intended?

Bootstrapping Trust
What F will this machine compute?

Bootstrapping Trust is *Hard!*

Challenges:
- Hardware assurance
- Ephemeral software
- User Interaction

Safe?

Yes!
Bootstrapping Trust is *Hard*!  
**Challenges:**  
- Hardware assurance  
- Ephemeral software  
- User Interaction

Establish Trust in Software

- Software is *ephemeral*  
- We care about the software *currently* in control  
- Many properties matter:  
  - Proper control flow

*Which property matters most?*
A Simple Thought Experiment
• Imagine a perfect algorithm for analyzing control flow
  – Guarantees a program always follows intended control flow
• Does this suffice to bootstrap trust?  No!

We want code identity

What is Code Identity?
• An attempt to capture the behavior of a program
• Current state of the art is the collection of:
  – Program binary
  – Program libraries
  – Program configuration files
  – Initial inputs
• Often condensed into a hash of the above
Code Identity as Trust Foundation

• From code identity, one may be able to infer:
  – Proper control flow
  – Type safety
  – Correct information flow

• Reverse is not true!

What Can Code Identity Do For You?

• Research applications
  – Secure the boot process
  – Count-limit objects
  – Improve security of network protocols
  – Thwart insider attacks
  – Protect passwords
  – Create a Trusted Third Party

• Commercial applications
  – Secure disk encryption (e.g., Bitlocker)
  – Improve network access control
  – Secure boot on mobile phones
  – Validate cloud computing platforms
Establishing Code Identity

Establishing Code Identity

$X_{\text{Other}} \rightarrow X_{\text{Alice}} \rightarrow Y_{\text{Alice}} \rightarrow Y_{\text{Other}} \rightarrow F$

$X_{\text{Other}} \rightarrow X_{\text{Alice}} \rightarrow f_1 \rightarrow f_2 \rightarrow \ldots \rightarrow f_N \rightarrow Y_{\text{Other}} \rightarrow Y_{\text{Alice}}$
Establishing Code Identity

[Gasser et al. '89], [Arbaugh et al. '97], [Sailer et al. '04], [Marchesini et al. '04], ...

Secure Booting Based on Code Identity

[Gasser et al. '89], [Arbaugh et al. '97], ...
Secure Booting Based on Code Identity

[Gasser et al. ’89], [Arbaugh et al. ’97], ...

Root of Trust

\[ \text{Sign}_{K_{\text{priv}}} (\ldots) \]

Local Access Control (“Sealing”)

Root of Trust

Data?

No!

Data

Secure Booting Based on Code Identity

[Gasser et al. ’89], [Arbaugh et al. ’97], ...

Root of Trust

\[ \text{Sign}_{K_{\text{priv}}} (\ldots) \]

Local Access Control (“Sealing”)

Root of Trust

Data?

No!

Data
Authenticated Boot: Recording Code Identity


Attestation: Conveying Records to an External Entity

Interpreting Code Identity

Traditional
[Gasser et al. ’89], [Sailer et al. ’04]

Policy Enforcement
[Marchesini et al. ’04], [Jaeger et al. ’06]

Virtualization
[England et al. ’03], [Garfinkel et al. ’03]
Interpreting Code Identity

**Traditional**
[Gasser et al. '89], [Sailer et al. '04]

**Policy Enforcement**
[Marchesini et al. '04], [Jaeger et al. '06]

**Virtualization**
[England et al. '03], [Garfinkel et al. '03]

**Late Launch**
[Kauer et al. '07], [Grawrock '08]

---

**Targeted Late Launch**
[McCune et al. '07]
Interpreting Code Identity

Load-Time vs. Run-Time Properties

- Code identity provides load-time guarantees
- What about run time?
- Approach #1: Static transformation

[Erlingsson et al. '06]
Load-Time vs. Run-Time Properties

- Code identity provides load-time guarantees
- What about run time?
- Approach #1: Static transformation

**Open Question:**
How can we get complete run-time properties?

Roots of Trust

**Open Question:**
What functionality do we need in hardware?

- Code
- Integrity Checked
- Run Time
- Load Time

---

Cheaper

<table>
<thead>
<tr>
<th>Reference</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weingart '87</td>
<td>[1]</td>
</tr>
<tr>
<td>White et al. '91</td>
<td>[2]</td>
</tr>
<tr>
<td>Yee '94</td>
<td>[3]</td>
</tr>
<tr>
<td>Smith et al. '99</td>
<td>[4]</td>
</tr>
<tr>
<td>ARM TrustZone '04</td>
<td>[5]</td>
</tr>
<tr>
<td>TCG '04</td>
<td>[6]</td>
</tr>
<tr>
<td>Zhuang et al. '04</td>
<td>[7]</td>
</tr>
<tr>
<td>Chun et al. '07</td>
<td>[8]</td>
</tr>
<tr>
<td>Levin et al. '09</td>
<td>[9]</td>
</tr>
<tr>
<td>Spinellis et al. '00</td>
<td>[10]</td>
</tr>
<tr>
<td>Seshadri et al. '05</td>
<td>[11]</td>
</tr>
<tr>
<td>ARM TrustZone '04</td>
<td>[12]</td>
</tr>
<tr>
<td>TCG '04</td>
<td>[13]</td>
</tr>
<tr>
<td>Zhuang et al. '04</td>
<td>[14]</td>
</tr>
<tr>
<td>Chun et al. '07</td>
<td>[15]</td>
</tr>
<tr>
<td>Levin et al. '09</td>
<td>[16]</td>
</tr>
<tr>
<td>Spinellis et al. '00</td>
<td>[17]</td>
</tr>
<tr>
<td>Seshadri et al. '05</td>
<td>[18]</td>
</tr>
</tbody>
</table>
Human Factors

Open Questions:

State Continuity

- Sealed state may be bound to software identity
- What happens when that state changes?
  - Old state will still decrypt and verify
  - (It was created by the intended software)
- State must include version information
  - Secure counter
  - Hash chain

Crash Resilience
- Data may become permanently unusable

More on this on Wednesday
2nd-Generation Trusted Computing

- First generation of Trusted Computing was all about measurement and cryptographic key storage
  - (And Remote Attestation; Adrian will get into this tomorrow)
- What do we learn from measuring millions of lines of code?
  - Axiom still holds: at least one more flaw!
  - Can we somehow measure less code?
  - Can we somehow reduce the TCB?

2nd-Generation Trusted Computing

- AMD: Secure Virtual Machine (SVM)
- Intel: Trusted eXecution Technology (TXT)
  - Formerly LaGrande Technology (LT)
- Common to both:
  - Late launch of a measured block of code
  - Hardware support for virtualization
- Differences:
  - AMD provides measured environment only
  - Intel adds authenticated code capabilities
    - The system’s chipset contains a public key to verify signed code
    - That code executes “on die” directly out of cache
Hardware Support for Trusted Computing

- Trusted Computing Group (TCG) proposed Trusted Platform Module (TPM) chip
  - Already included in many platforms (200M+)
  - Cost per chip less than $1 today

- 2nd Generation: “Dynamic root of Trust”
  - CPU and chipset support in addition to TPM
  - Modern microprocessors provide special instructions that interact with TPM chip
  - AMD SVM: SKINIT instruction
  - Intel TXT: GETSEC[SENTER] instruction

TCG Trusted Platform Module (TPM)
Trusted Computing in the Real World

- Commercial products using these ideas
- Two main areas
  - Full Disk Encryption (FDE)
  - Rollback protection

- We will get into TC research starting tomorrow

Microsoft Bitlocker*

“Some of the largest and medium-sized U.S. airports report close to 637,000 laptops lost each year, according to the Ponemon Institute survey released Monday.”
—PC World June 2008

“More than 100 USB memory sticks, some containing secret information, have been lost or stolen from the Ministry of Defense since 2004, it has emerged.”
—BBC News July 2008

* Slides from Troy Funk, Microsoft PM, WinHEC 2008
Challenge: Reducing the Trusted Computing Base

- Today’s OSes have too much power
- Total access to application data

TCB: Set of all HW & SW that must be trusted for an application to do its job

Trusted computing base for App 1 includes majority of: OS, drivers, and privileged applications!!!
### BitLocker™ in Windows Vista and Windows Server 2008

<table>
<thead>
<tr>
<th>Drive Type</th>
<th>Unlock Methods</th>
<th>Recovery Methods</th>
<th>Management</th>
<th>Other requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System Drives</td>
<td>TPM</td>
<td>Recovery password</td>
<td>Group policy controlled options presented to users</td>
<td>Use of the BitLocker Drive Preparation Tool to create a system partition where boot files are located</td>
</tr>
<tr>
<td></td>
<td>TPM+PIN</td>
<td>Recovery key</td>
<td></td>
<td>System partition size: 1.5GB</td>
</tr>
<tr>
<td></td>
<td>TPM+Startup key</td>
<td>Active Directory backup of recovery password</td>
<td></td>
<td>System partition assigned a drive letter</td>
</tr>
<tr>
<td></td>
<td>TPM+PIN+Startup Key*</td>
<td></td>
<td></td>
<td>NTFS file system</td>
</tr>
<tr>
<td>Fixed Data Drives</td>
<td>Automatic unlocking</td>
<td>Same as OS drive</td>
<td>No policies</td>
<td>Operating System drive must be encrypted</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NTFS file system</td>
</tr>
</tbody>
</table>

* USB storage device

- Attestation: can be checked by external entity
- Secure counter
- Hash chain
- Late launch
A Note on Paper Reviews

• Make your contribution statements more summarizing than descriptive
  – Remembering "introduced a methodology" isn't as useful to you later on as "the key idea is XYZ"
  – Most of you wrote precise statements for weaknesses – do the same for contributions!