Design and Implementation of a TCG-based Integrity Measurement Architecture

Reiner Sailer, Trent Jaeger, Leendert van Doorn, and Xiaolan Zhang
Secure Systems Department

August 2004 | Usenix Security Symposium 2004
Overview

- Problem – Runtime integrity guarantees

- Solution – Hierarchical software-stack measurements
  - Load guarantees
  - Property attestation

- Current Implementation

- Future Work
Problem – What is the Integrity of a System?

- Insecure networked world
- SSL and IPSEC provide secure channels
  Answers: With whom am I interacting securely?

Open Problem:
How can you trust this system?
Trusted Computing Group Architecture

Execution Flow

Measurement Flow

TCG-based Integrity Measurement Architecture

Defined by Grub (IBM Tokyo Research Lab)

Defined by TCG (Platform specific)

Platform Configuration Registers 0-23

0-7

4-7

>= 8
Integrity Measurement Architecture – Solution

**Attested System**

- **Data**
- **Program**
- **Config data**
- **Kernel**
- **Kernel module**
- **Boot-Process**

**Measurement**

- SHA1(Boot Process)
- SHA1(Kernel)
- SHA1(Kernel Modules)
- SHA1(Program)
- SHA1(Libraries)
- SHA1(Configurations)
- SHA1(Structured data)

**Signed TPM Aggregate**

**System-Representation**

**System Properties**

- ext. Information (CERT, ...)

**Analysis**

**Known Fingerprints**
TPM-Based Integrity Measurement Architecture

- Achievement of our Integrity Measurement Architecture (IMA)
  
  Extend TPM-based attestation into the system runtime
  
  → Attest the Software Stack

- IMA-Guarantees
  
  - Non-intrusive (not changing system behavior)
  
  - Load-guarantees for code loaded into the system run-time
  
  - Detects systems cheating with the measurement list

- Goals
  
  - Negligible overhead on attested system
  
  - Usability
Example: Web Server

- **Executables** (Program & Libraries)
  - `apachectl`, `httpd`, `java`, ...
  - `mod_ssl.so`, `mod_auth.so`, `mod_cgi.so`...
  - `libc-2.3.2.so`, `libjvm.so`, `libjava.so`...

- **Configuration Files**
  - `httpd.conf`, `html-pages`,
  - `httpd-startup`, `catalina.sh`, `startup.sh`

- **Unstructured Input**
  - HTTP-Requests
  - Management Data

---

Example: Web Server

- **Executables** (Program & Libraries)
  - `apachectl`, `httpd`, `java`, ..
  - `mod_ssl.so`, `mod_auth.so`, `mod_cgi.so`, ..
  - `libc-2.3.2.so`, `libjvm.so`, `libjava.so`, ...

- **Configuration Files**
  - `httpd.conf`, `html-pages`,
  - `httpd-startup`, `catalina.sh`, `startup.sh`

- **Unstructured Input**
  - HTTP-Requests
  - Management Data

---

Example: Web Server

- **Executables** (Program & Libraries)
  - `apachectl`, `httpd`, `java`, ..
  - `mod_ssl.so`, `mod_auth.so`, `mod_cgi.so`, ..
  - `libc-2.3.2.so`, `libjvm.so`, `libjava.so`, ...

- **Configuration Files**
  - `httpd.conf`, `html-pages`,
  - `httpd-startup`, `catalina.sh`, `startup.sh`

- **Unstructured Input**
  - HTTP-Requests
  - Management Data

---

Example: Web Server

- **Executables** (Program & Libraries)
  - `apachectl`, `httpd`, `java`, ..
  - `mod_ssl.so`, `mod_auth.so`, `mod_cgi.so`, ..
  - `libc-2.3.2.so`, `libjvm.so`, `libjava.so`, ...

- **Configuration Files**
  - `httpd.conf`, `html-pages`,
  - `httpd-startup`, `catalina.sh`, `startup.sh`

- **Unstructured Input**
  - HTTP-Requests
  - Management Data
IMA Implementation – File Measurements

**Measurement** = SHA1(File Contents) at load time

- **Kernel** measures: kernel modules, programs, and shared libraries
- **Applications** measure their own critical input
  
  Examples:
  
  Bash Shell measures: scripts before execution
  Future: Java, Perl, Apache, Jakarta Tomcat …

**Advantage**

**Unique** Software-Fingerprints (e.g. sendmail-8.12.8-9.90)

→ Secure hash represents well known security properties
IMA Implementation – Measurement List Maintenance

Measurement list aggregation:

- **Compute** 160bit-SHA1 over the contents of the data (measurement)
- **Adjust** Protected hw Platform Configuration Register (PCR) to maintain measurement list integrity value
- **Add** measurement to ordered measurement list
  - Executable content is recorded before it impacts the system
  - That is, before it can corrupt the system

\[
SHA1( PCR_k \parallel \text{new measurement})
\]
IMA Implementation – Measurements by the Kernel

Linux Security Module

Traditional execution path

**Measurement List (Kernel-held)**

**Integrity Value**

**SHA1**

**Execute**

(*file*)

**/bin/bash**

**Memory Map**

**Schedule**
IMA Implementation – Measurements by Applications

```c
fd = open("http.conf")
measure(fd)
read(fd)
close(fd)
```

**SHA1**

**/etc/http.conf**

**Integrity Value**

**Measurement List (Kernel-held)**

**ToM-ToU Detection:**
reader/writer lock

**Kernel**
Example: Rootkit compromise analysis

<table>
<thead>
<tr>
<th>Measurement List</th>
<th>Fingerprint DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 : D6DC07881A7EFD58EB8E9184CCA723AF4212D3DB</td>
<td>boot_aggregate</td>
</tr>
<tr>
<td>001 : 84ABD2960414CA4A448E0D2C9364B4E1725BDA4F</td>
<td>init</td>
</tr>
<tr>
<td>002 : 194D956F288B36FB46E46A124E59D466DE7C73B6</td>
<td>ld-2.3.2.so</td>
</tr>
<tr>
<td>003 : 7DF33561E2A467A87CDD4BB8F68880517D3CAECB</td>
<td>libc-2.3.2.so</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>110 : F969BD9D27C2CC16BC668374A9FBA9D35B3E1AA2</td>
<td>syslogd</td>
</tr>
</tbody>
</table>
| ... | ...

(a) THE GOOD CASE

... 110 : F969BD9D27C2CC16BC668374A9FBA9D35B3E1AA2 | syslogd |
... 525 : 4CA3918834E48694187F5A4DAB4EECD540AA8EA2 | syslogd-LRK5 |
... (b) LRK5-COMPROMISED SYSLOGD
Results

Attested System:

- Implementation: ~ 5000 LOC (LSM kernel module)
- About 400-600 measurements for Fedora C2, Apache, Jakarta Tomcat, etc.
- Measurement Overhead

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Application</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean Hit</td>
<td>~ 0.1 µs</td>
<td>~ 5 µs</td>
</tr>
<tr>
<td>New (TPM) Measurement</td>
<td>~ 5 ms + SHA1 (~80MB/s)</td>
<td>~ 5ms + SHA1 (~80MB/s)</td>
</tr>
</tbody>
</table>

Attestation service:

- Known Fingerprint DB ~ 20 000 Fingerprints (RedHat 9.0, Fedora, ES3)
- Attestation: 1-2 second “latency” (unoptimized demonstration)
Ongoing & Future Work

Open-Source Integrity Measurement Architecture

→ LSM kernel module

Isolation of unknown or distrusted measurements

→ Measure Information flow between executables

Predict future system states

→ Measure SELinux policy and enforcement
Thank You!

Further Information

http://www.research.ibm.com/secure_systems_department/projects/tcglinux
BACKUP
Trusted Platform Module (TPM)

- Trusted Computing Group – Industry Consortium
  → open interface specification of TPM (current version 1.2)
- Independent Manufacturers for TPM chips, glued onto system
- TPM offers protected hw registers (and more)
- TPM includes Public Endorsement Key
  Endorsement key signature provides identity and integrity
  Endorsement key certificate includes platform information
- TCG specifies use of TPM to realize:
  o Hardware Platform Certification
  o Boot Configuration attestation up to boot-loader
IMA Implementation (V) – Measurement list

Real measurement list (cat /proc/tcg/measurements):

#000: D6DC07881A7EFD58EB8E9184CCA723AF4212D3DB
#001: 84ABD2960414CA4A448E0D2C9364B4E1725BDA4F
#002: 9ECF02F90A2EE2080D4946005DE47968C8A1BE3D
#003: 336536B0E22FF762BB539D7FCB7CD283D4622342
#004: A4DC5EDF06698646CD76916F16E95C37E55DC12B
...  
#027: 2AC8FD6000DDEAA7BD10D7D4E3CE56868100980D
#028: C0F7BCEF34A2AA7DAECC2B1648C02FB7CFEC9A3D
...  
#070: 01B33D515C3B23F1AB0BAEF845F0A3CA079E9A1E
#071: CEBA19AE012DBC2E1A3E428070D2C90A695F1AA4
#072: 2998794AD01E64155EA6831B05584298ED4
#073: 68464F40452A4B4B63707C3925CA7EC71A7E3B72D
#074: 4CAFF329BF20F736E9E14B4123E0E5F88D8D418D
...  
#080: 147D5593003A8A0DDFA3B27430734F804F388168
#081: F94054C5C1B38136C72FB1A56EE6F100AB090115
...  
#244: D312491B8247E897D708407543B5369D12E0DA7C
#245: BB2CEA2BA56CCC3D20BDECA554D2976DA0D5AAB3
...
Measurement Implementation (II) – Example

execve(bash) → file_mmap(/bin/bash) → M=SHA1(/bin/bash)

M new?

No → execve(bash)

Yes → Add M to List → Extend TPM

mmap/run(bash)
TPM Register Example

Manufacturer: 0x41544d4c
TCG version: 1.1
Firmware version: 0.6
PCRs: 16
DIRs: 2
Slots: 10

DIR-00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
DIR-01: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR-00: 8F 99 36 66 55 E5 09 69 AF FB 3B 08 C9 F6 9B 38 7E 62 D3 75
PCR-01: 99 7D 7E DD 91 D3 8D BE 19 1F 78 F2 1A 4C 7E 9C 65 C6 BA 61
PCR-02: EB B3 BA AE E7 57 4B B6 37 AA AB 67 0F 9A C1 BC EB 6F 80 F3
PCR-03: 04 FD EC DD 50 1D AF 0F 62 4C 1F 99 60 12 CF 30 44 FF 46 10
PCR-04: BC 68 F2 66 F8 B7 5E 55 8C D2 74 70 B7 0B 53 20 0B 48 0F AB
PCR-05: DF 82 43 3F B8 7C 24 09 31 B0 8A 39 B9 63 4A 48 58 A7 FB 97
PCR-06: 04 FD EC DD 50 1D AF 0F 62 4C 1F 99 60 12 CF 30 44 FF 46 10
PCR-07: 31 20 AA 10 F2 D4 23 E9 4D 2C 59 3A 00 1B 02 44 42 B1 DE 65
PCR-08: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR-09: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

PCRs:

PCR-10: 51 69 57 67 92 71 56 BB 18 D2 5C 9B 87 CB 01 C8 45 FD 7D 65

Type: Atmel TPM
Bus controller: Intel ICH4 LPC
Recap

- Applications
- Kernel
- Boot configuration
- Platform (HW, BIOS)

Future (Ongoing) Work

- Integrity Measurement Architecture
- Open-Source IMA
  - LSM kernel module
- Information flow analysis
  - Attestation space partitioning
  - Integration of information flow into the LSM kernel module
- Attest Policy Enforcement
  - Predictable properties
  - SELinux-Integration (subject, object label)
Integrity Measurement Architecture – Validation

Detecting Cheating Systems

1. send 160bit-nonce (unpredictable)
2.a receive measurement list
2.b receive \{nonce, PCR\}_{TPM\_Signature}
3. validate TPM\_Signature and nonce
4. validate list (virtual extension)

Tampering with the signed PCR is recognized in 3
Tampering with the list is recognized in 4
IMA – Software Stack Measurements

Measurement

Attested System

- Data
- Config data
- Boot Process
- Kernel
- Kernel modules
- Programs

SHA1(boot)
SHA1(kernel)
SHA1(kernel modules)
SHA1(programs)
SHA1(libraries)
SHA1(configurations)
SHA1(structured data)
...

TPM Aggregate PCR