

Submitted by
Michael Preisach,
BSc
1155264

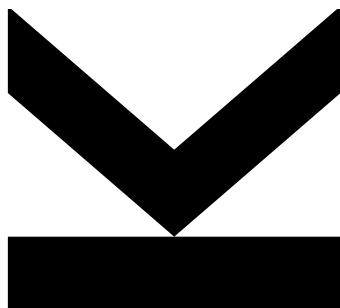
Submitted at
Institute for
Networks and
Security

Supervisor and First
Examiner
Univ.-Prof. DI Dr.
René Mayrhofer

Second Examiner
DI **Tobias Höller**

April 30, 2020

Project Digidow: Biometric Sensor



Master Thesis

to obtain the academic degree of

Master of Science

in the Master's Program

Computer Science

Abstract

What is it all about? Why is that interesting? What is new in this thesis? Where is the solution directing to?

Zusammenfassung

Das am Institut für Netzwerke und Sicherheit entwickelte Projekt *Digital Shadow* benötigt in vielen Bereichen ein prüfbares Vertrauen um eine Erkennung von Nutzern anhand ihrer biometrischen Daten zu erkennen und Berechtigungen zuzuteilen. Das Vertrauen soll dem Nutzer die Möglichkeit geben, die Korrektheit des Systems schnell und einfach zu prüfen, bevor er/sie dieses System biometrische Daten zur Verfügung stellt. Diese Masterarbeit beschäftigt sich nun mit den existierenden Werkzeugen, die ein solches Vertrauen schaffen können. Das implementierte System kombiniert diese Werkzeuge, um damit sensible Daten von Nutzern aufzunehmen und im Netzwerk von Digital Shadow zu identifizieren. Es soll dabei sicher gestellt sein, dass eine fälschliche Verwendung der sensiblen Nutzerdaten ausgeschlossen wird. Anhand dieses Systems werden die Eigenschaften einer vertrauenswürdigen Umgebung für Software diskutiert und notwendige Rahmenbedingungen erläutert.

Contents

1	Introduction	1
1.1	introduction in project digidow	1
1.2	Biometric Sensor use case in DigiDow	2
1.3	Definitions and Requirements	2
1.3.1	Requirements	3
1.4	Description of structure	3
2	Related Work	4
3	Concept	5
3.1	Definition of the Biometric Sensor	5
3.1.1	Definitions	5
3.1.2	What has the BS to do?	5
3.2	Attack Vectors and Threat Model	6
3.2.1	The Threat Model	6
3.3	Trust and Security	6
3.4	Systems of Trust	6
3.4.1	Secure Boot, TXT,	7
3.4.2	TPM1.2	7
3.4.3	TPM2.0	7
3.5	Integrity Measurements	7
3.6	Verify Trust (DA and DAA)	7
4	Implementation	8
4.1	Trusted Boot	8
4.2	Integrity Measurement Architecture	8
4.2.1	Handling external hardware	8
4.3	Interaction with TPM2	8
4.4	Direct Anonymous Attestation	9
5	Conclusion and Outlook	10
5.1	Testing	10
5.2	Limitations	10
5.3	Outlook	10
	Installation instructions	13
1	Installing IMA on Arch	13

1 Introduction

We all live in a world full of digital systems. They appear as PCs, notebooks, cellular phones or embedded devices. Especially the footprint of embedded computers became so small that they can be used in almost all electrical devices. This product category forms the so called *smart* devices.

With all these new devices a lot of societal problems could be solved in the past few decades. Many of them automate services to the public like managing the bank account, public transportation or health services. There is an endless list of services that can be done by a computer.

The downside of all these digital services is that using these services generate a lot of data. Besides the intended exchange of information, many of the services try to extract metadata as well. Which IP is connected? What kind of device is that? Is the software up to date? Was this device here in the past? Which other sites did the user browse? This is an endless list of questions which can be answered with a set of metadata. And all this data is collected when users browse the Internet. At the end the user may not be charged financially but one pays with this metadata. The customer becomes the product.

However when a project is financed by the public, it should be possible to show users that there is a difference in the usage. It should be possible to prove that an application or a computer system is honest to the user. People should be convinced of this honesty and build trust in using this kind of software.

1.1 introduction in project digidow

The Project *Digital Shadow* is under ongoing development at the Institute of Networks and Security and creates a scalable system for authentication. Key feature is privacy by design and a provable system to create trust to the end user.

At this early stage the interfaces and interaction points are not fully defined.

This is a brief description of the process of authentication:

1.2 Biometric Sensor use case in DigiDow

derive the use case of the Biometric sensor out of the above model.

1.3 Definitions and Requirements

- privacy
- integrity
- trust
- security
- Usage Model of Biometric Sensor

This thesis will describe a system, which is part of the Digital Shadow network. Therefore it has to meet the common principles in information security, namely:

- *Availability*:
- *Integrity*: ISO 27000 (Data Integrity)
- *Confidentiality*: ISO 27000

Upon AIC it should be possible for users to prove honesty of the system. This is what *trust* defines in information security

1.3.1 Requirements

- given a set of software, this system should provide information that exactly this version of software is running on the system. (Integrity)
- The system must furthermore show that it is a member of valid biometric sensors (Attestation)
- All the given information should be anonymized. It should not be possible to gain any other information about the system (Anonymity)
- It should be ensured that no sensitive data is stored at the biometric sensor

Scope of this thesis is on implementing the system from hardware to application layer. It is not supposed to think about the network communication.

1.4 Description of structure

1. What exists out there?
2. What is the theoretical solution
3. What about the implementations used - what is the limitation of the used tools?
4. How far are we? what has to be considered next?

2 Related Work

- What exists in the field?
- Keylime
- Xaptum ECDA
- FIDO 2 ECDA
- Strongswan Attestation
- Linux IMA
- Secure Boot
- Intel TXT
- Trusted Execution Environment (TEE)
- nanovm (nanovms.com)

3 Concept

The theoretical tool that should be formed to one whole system implementation in this thesis.

3.1 Definition of the Biometric Sensor

What part fulfills the BS and what needs to be done. Record Sensor data, Network Discovery, send sensor data via trusted channel to PIA

3.1.1 Definitions

- Sensitive Data
- Privacy
- Metadata
- Attribute

3.1.2 What has the BS to do?

1. Listen for a Trigger to start the Authentication Process
2. Collect Sensor Data (Picture, Fingerprint) and calculate a biometric representation
3. Start Network Discovery and find the PIA of this person
4. Create a trusted and secure channel and transmit the attributes for verification
5. Restore the state of the system as it was before this transaction

3.2 Attack Vectors and Threat Model

3.2.1 The Threat Model

- Definition of sensitive data / privacy / metadata
- This version of BS is not owned by the user, there is no personal data in the System
- Rogue Personal Identity Agent (PIA)
- Metadata Extraction
- Attribute extraction
- Sensor Data Modification/manipulation
- Wiretap between Sensor and System (USB or network)
- Physical Manipulation of the BS-System
- Network - Retransmission of sensor data of a rogue BS
- Network - Blocking Data transmission of a rogue BS
- Rogue BS Sensor Data aggregation
- Rogue BS Sensor data modification before transmission

3.3 Trust and Security

Differentiation between trust and security — and the problem that not everyone is using that right.

3.4 Systems of Trust

All trust systems are built on the standards of Trusted Computing Group.

3.4.1 Secure Boot, TXT, ...

Trusted Boot is not the same as Secure Boot. Explain the difference

3.4.2 TPM1.2

Initial Version of the Cryptocoprocessor, successfully spread into many systems, but hardly any integration in Trust/security Software

3.4.3 TPM2.0

Current Version (published 2014) with some improvements.

- Hierarchies
- Endorsement Key
- Attestation Identity Key
- Key management

3.5 Integrity Measurements

Extend the Chain of Trust beyond the boot process. The Kernel can measure many different types of Resources. What is a useful set of measurements

3.6 Verify Trust (DA and DAA)

Use the TPM to proof trustworthiness to other instances like the PIA

4 Implementation

4.1 Trusted Boot

- Trusted Boot with GRUB 2.04: TPM support available; PCR mapping
- Secure Boot with Unified Kernel; another PCR mapping
- Benefits and Drawbacks of both variants

Limitations due to bad implementation on BIOS-Level, no Certificate Verification Infrastructure available for TPMs? Needs to be proven for correctness.

4.2 Integrity Measurement Architecture

Available on Ubuntu, RedHat and optionally Gentoo. The Kernel has the correct compile options set.

4.2.1 Handling external hardware

How can camera and fingerprint sensor be trusted? What is the limitation of this solution?

4.3 Interaction with TPM2

tpm2-tools 4.x are usable to interact with the TPM from the command line. Available on all major releases after summer 2019. Fallback is using the TPM2 ESAPI or SAPI, which is available on almost all Linux distributions.

4.4 Direct Anonymous Attestation

DAA Project from Xaptum: Working DAA handshake and possible TPM integration. Requires an Attestation Key which is secured with a password policy.

5 Conclusion and Outlook

5.1 Testing

These are the test results

5.2 Limitations

Still hard to set up a system like that. Documentation is available, but hardly any implementations for DAA and IMA.

5.3 Outlook

Hardening of the system beyond IMA useful. Minimization also useful, because the logging gets shorter.

Table 5.1 is an example of a table, in which the numbers are aligned at the comma, every second line is colored and the commands `\toprule`, `\midrule` and `\bottomrule` are used [1].

Table 5.1: Example

Länge l in m	Breite b in m	Höhe h in m
12.454	1.24	335.3
543.22	32.123	33.21
353.0	33.0	33.0
23.3	333.2	32.4

List of Figures

List of Tables

5.1	Example	10
-----	-------------------	----

Bibliography

- [1] Will ARTHUR, David CHALLENGER, and Kenneth GOLDMAN. *A Practical Guide to TPM 2.0*. Jan. 2015. DOI: 10.1007/978-1-4302-6584-9.

Installation instructions

1 Installing IMA on Arch

https://wiki.archlinux.org/index.php/Kernel/Arch_Build_System in combination with https://wiki.gentoo.org/wiki/Integrity_Measurement_Architecture:

```
1 sudo pacman -S asp base-devel
2 cd ~
3 mkdir build && cd build
4 asp update linux
5 asp export linux #Linux repo exported to this directory
```

Change *pkgbase* in `linux/PKGBUILD` to custom name, e.g. `linux-ima`. Check `linux/config` for the following settings:

```
1 CONFIG_INTEGRITY=y
2 CONFIG_IMA=y
3 CONFIG_IMA_MEASURE_PCR_IDX=10
4 CONFIG_IMA_LSM_RULES=y
5 CONFIG_INTEGRITY_SIGNATURE=y
6 CONFIG_IMA_APPRAISE=y
7 IMA_APPRAISE_BOOTPARAM=y
```

For optimizing file access, add to every `fstab`-entry *iversion*. It prevents creating a hash of the file at every access. Instead the hash will only be created when writing the file.

`updpkgsums` generates new checksums for the modified files.

`makepkg -s` then makes the new kernel

SSLlMMxxii Hallowelt