Enemy in the Clouds

Confidential Computing Group@MMS

Dr. Ivan Gudymenko, IT Security Architect
"Joining your team has given me the curiosity to learn more about new technologies. The work environment is a source of motivation to work harder towards the resolution of business problems. What also amazed me is the sense of sharing, collaboration, and teamwork between the different members to work on new approaches for an ongoing project."

“I join this team because I believe the idea of securing our confidential data on the cloud while it’s being processed is an utmost security concern which we need to address as early as possible and as an emerging technology I want to be a pioneer in this field.”

“I wanted to learn about new methods that even ensure the security against strong attacker models”
Confidential Computing Motivation

- Leverage the benefits of Cloud Computing
- Outsourcing the operations of infrastructure
  - retain the control over app, keys, etc
- Ensure privacy and security compliance (e.g. Gematik requirements)
- Separate infrastructure and application OPS
Die medizinische Versorgung ist generell gewährleistet, da als Ersatzverfahren (bei Ausfall von Diensten bzw. Störungen) auf das bisherige Papierformular (Muster 16) zurückgegriffen wird.

„Unzureichendes Verständnis bei Verschlüsselung“


Mit Ende-zu-Ende-Verschlüsselung könnte zukünftig z. B. kein in Köln ausgestelltes E-Rezept in Madrid eingelöst werden – was zwischen anderen europäischen Ländern bereits möglich ist und auch für deutsche Versicherte durch die Anbindung an den Europäischen Raum für Gesundheitsdaten künftig möglich werden soll.
What should be protected and why

- Application data
  - especially data-in-use!
- Code
- Secrets (tokens, passwords, master keys, etc)
Securing the 3 States of Data

Data at rest

Data in transit

Data in use

This is a challenge!
Trust assumptions

Conventional model:
- trusting the underlying software
  - Operating System
  - VM Layer
  - BIOS/Firmware

Confidential Computing:
- trusting only hardware (TEE)
  - TEE Hardware
    - (Intel SGX, AMD SEV, Arm TrustZone hardware)

Minimize attack surface
Trust anchor notion in confidential computing

Taken from whitepaper: Enemy in the clouds: protecting your cloud, assets from powerful adversaries
Examples of Trusted Execution Environments?

- Smart cards
- SIM Cards
- TPM (trusted platform modules)
- ...

T-Systems: Let’s power
gerher performance
The Notion of Attestation

- ensure execution in a secure container hosted by the trusted hardware
- is NOT code signing but rather measuring
- Software measuring (measurement hash)
  - compare the expected hash with the measured one

Figure taken from https://de.freepik.com
Example: Intel Attestation

Secure Enclave and the metaphor of a security domain

C represents a security domain of B in the environment under the control of A
Secure Enclave and the metaphor of a security domain

C represents a security domain of B in the environment under the control of A

Application in the cloud
Secure Enclave and the metaphor of a security domain
Confidential Computing und Secure Enclaves

• Secure Enclave: in essence a subclass of Trusted Execution Environment (TEE)

• Secure Enclaves to protect the data in use (Confidentiality, Integrity)
  • Relaxes the trust model against the cloud provider
  • Allows for additional security against administrators

• Migration of legacy applications is possible
  • A number of tools/frameworks/libraries are available
  • Migration of complex applications by partitioning out the security critical components into the enclave
**Real-world use case: Signal Secure Value Recovery**

**Secure Value Recovery Service (Beta)**

Building the SGX enclave (optional)

Building reproducibly with Docker

Prerequisites:

- GNU Make
- Docker (able to run Debian image)

```bash
$ make -C <repository_root>/enclave
```

The default docker-install target will create a reproducible build environment using `enclave/install/
```

The Dockerfile will download a stable Debian snapshot Debian Docker image. The Debian project builds their Docker images reproducibly, based on the a snapshot of the Debian repo on the date of the build from the Debian Snapshot Project. Make will then be run inside the newly built Docker Debian image at `infinite Building with Debian` section below.

NB: the installed enclave will be signed with the SGX debug flag enabled by an automatically generated signing key. Due to Intel SGX licensing requirements, a debug enclave can currently only be run with SGX debugging enabled, allowing inspection of its encrypted memory, and invalidating its security properties. To use an enclave in production, provide the Intel-whitelisted signing key at `enclave/build/enclave_hermeten.key` before building. Alternately, the generated enclave at `enclave/build/enclave_hermeten.enc` may be signed and stored at `enclave/build/enclave_hermeten.sig`.

Building with Debian

Prerequisites:

- GNU Make
- maked
- t signals
- g++

```
https://github.com/signalapp/SecureValueRecovery/blob/master/README.md
```

See also: [https://signal.org/blog/secure-value-recovery/](https://signal.org/blog/secure-value-recovery/)
Case Study: Cloud Encryption Proxy

Confidential Computing Environment

• Protecting against cloud admins
• Data-in-use protection

User view (plaintext)
Name: Max Müller
Account: 1223123
Blutgruppe: AB
Geburtsdatum: 15.06.1980

Data in the cloud (encrypted/tokenized)
Name: Tp5. &qi fcj4&kL
Account: 378338590
Blutgruppe: AB
Geburtsdatum: 08.02.1980
Intel SGX

- SGX = „Software Guard Extensions“
- Extended security-related instruction codes of certain Intel CPU
- Confidential memory areas rendering EPC (Enclave Page Cache); „SGX RAM“
The notion of an enclave

Untrusted Part of App

Create Enclave

Call Trusted Func.

(etc.)

Trusted Part of App

Call Gate

Execute

Return

m8U3bcV#zP49Q

Privileged System Code:

OS, VMM, BIOS, SMM, ...

Taken from [Dror Caspi Intel software guard extensions (SGX)]
Secure Enclaves to protect the data in use (Confidentiality, Integrity)
- Execute the sensible data inside of secure enclave
- Relaxes the trust model against the cloud provider
- Allows for additional security against administrators
- Especially interesting in Cloud environment

Migration of legacy applications is possible
- A number of tools/frameworks/libraries are available
- Migration of complex applications by partitioning the security critical components into the enclave
Creation of an SGX Application
Creation of an SGX Application

- EPC access only by the respective enclave
- No debugging
- Uninterrupted encryption
- Hardware based *root keys*
- Verification of the enclave and of the execution environment through remote attestation
Recall:

Application partitioning and TCB implications

Privileged System Code:
OS, VMM, BIOS, SMM, ...

Taken from [Dror Caspi Intel software guard extensions (SGX)]
## Application partitioning and TCB implications

<table>
<thead>
<tr>
<th>Confidential App design</th>
<th>Effort</th>
<th>TCB size</th>
<th>Approach/Lib</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning</td>
<td>High</td>
<td>Small</td>
<td>Intel SDK („native“)</td>
</tr>
<tr>
<td>Without Partitioning</td>
<td>Low/Middle</td>
<td>Large</td>
<td>Scone, Ego, Gramine</td>
</tr>
</tbody>
</table>
Developing and Deploying Confidential Applications

Taken from whitepaper: Enemy in the clouds: protecting your cloud, assets from powerful adversaries
App Migration

Migrating the applications into the enclave infrastructure.
Frameworks and tools to support app migration

Figures taken from: scontain.com, grapheneproject.io, occlum.io, anjuna.io
A practical example
Confidential Patient Records in the Cloud

- Confidential Patient Records as a simple key-value database service in the public cloud
- Deployed on Azure Kubernetes Service (AKS) cluster with confidential computing nodes using Azure CLI

Demo: Deploy an AKS with CC nodes

Quickstart: Deploy an Azure Kubernetes Service (AKS) cluster with confidential computing nodes using Azure CLI (preview)

This quickstart is intended for developers or cluster operators who want to quickly create an AKS cluster and deploy an application to monitor applications using the managed Kubernetes service in Azure.

Overview

In this quickstart, you'll learn how to deploy an Azure Kubernetes Service (AKS) cluster with confidential computing nodes using the Azure CLI and run an hello world application in an enclave. AKS is a managed Kubernetes service that lets you quickly deploy and manage clusters. Read more about AKS here.

Deployment pre-requisites

1. Have an active Azure Subscription. If you don’t have an Azure subscription, create a free account before you begin.
2. Have the Azure CLI version 2.8.64 or later installed and configured on your deployment machine (Run `az --version` to find the version. If you need to install or upgrade, see Install Azure CLI).
3. Use the preview version: minimum version 0.4.62.
4. Have a minimum of six DCx-v2 cores available in your subscription for use. By default, the VM cores quota for the confidential computing per Azure subscription is 8 cores. If you plan to provision a cluster that requires more than 8 cores, follow these instructions to raise a quota increase ticket.

Confidential computing node features (DCx-v2)

1. Linux Worker Nodes supporting Linux Containers Only
2. Ubuntu Generation 2, 10.04 Virtual Machines
Demo: Application deployment in the enclave

demo@nms:/sgx/scone/flask_example$
demo@nms:/sgx/scone/flask_example$
Typical use cases for Confidential Computing

- Digital Rights Management
- Cloud-based operation of e-prescriptions (eRezept)
- Handover of Ambulance Service → Hospital
- Outsourcing Organ Donation Data
- Key and Access Control Management (e.g. Vault)
- Privacy-preserving Data Analytics
- Federated Learning
- Multi-Party Computation
- Email Encryption Proxy

https://www.gematik.de/media/erezept/_processed_/7/e/csm_gematik_App_Mockup_Startseite_01_cj_e983d21f8e.png
Availability of Confidential Computing

Premier Members:
- Accenture
- ARM
- Facebook
- Google
- Huawei
- Intel
- Microsoft
- Red Hat

Open Telekom Cloud

Azure
Confidential Computing: Beyond Intel SGX

- Intel SGX
- Intel TDX
- AMD SEV SNMP
- Arm TrustZone
- SGX and SEV are already available on e.g. Azure
Confidential Computing: an important innovation topic

https://www.youtube.com/watch?v=CCPI7C1hh0
https://www.youtube.com/watch?v=pv6e1izDcj0

Thank you

For your attention!