SEED4C General Presentation

Context

• Enterprises are moving their data & applications in the cloud (even for a time-bound project)
  - Data (e.g., HR, business), apps and policies (regulation, enterprise, end-user)
• Key issue: End-to-end protection in the cloud of the sensitive data and apps
Enterprises are moving their data & applications in the cloud (even for a time-bound project)

- Data (e.g., HR, business), apps and policies (regulation, enterprise, end-user)

Key issue: End-to-end protection in the cloud of the sensitive data and apps

SEED4C General Presentation

Cloud SLA

- Quality of Service
  - Availability, latency, etc.

- Security
  - Data storage
    - Data location
    - Data access control per app/per user
    - Data retention and deletion
    - Data usage tracing
    - Data breach notification
    - etc.

  - Data processing (in Virtual Machines)
    - VM location and co-location constraints
    - VM isolation
    - VM security level
    - etc.

  - Network configuration
    - Secure VM connection
    - etc.

How to enforce these SLA security policies?
How to monitor/certify the enforcement of these policies?
SEED4C approach
From an isolated security to a coordinated security

- **Secure Element Extended (SEE)**
  - Securely store critical data and securely execute critical apps
  - Support multi-tenant data & apps

- **Network of Secure Element Extended (NoSEE)**
  - Secure exchange of SEE contexts
  - Eg. allow critical data to only be transferred in secure & compliant VMs

Isolated Security  →  Coordinated Security
SEED4C General Presentation

Various types of use-cases at different cloud levels (IaaS, PaaS, SaaS)
**Modeling**

Virtualized Application Security

- **1 model = 3 views**

**Sam4C Modeling Tool**

1. Virtualized Application

2. Application Security

3. Resources Mapping

Example of properties: Integrity, Confidentiality, Isolation, etc.
Modeling
Virtualized Application

Screenshot of Airport Management Application (called “Musik”) in Sam4C Modeling Tool
Modeling
Application Security

Integrity, Confidentiality, Isolation properties

```
#context ctxMusikServiceMusik = {Airport_MAD.seed4c_musik}:(Service="App_Musik"): (AppDomain="App_Musik"): (Role="MusikAdmin");
#property Integrity(ctxMusikLogMusik, ctxMusikServiceMusik);
```

Authentication property

```
#context ctxDevice = (Client = "seed4c_device");
#context ctxDeviceMAD = ctxDevice:(Domain="Airport_MAD");
#context ctxDeviceEAS = ctxDevice:(Domain="Airport_EAS");
#context ctxMusikServiceSSH = {Airport_MAD.seed4c_musik}:(Service="SSH");
#property Authentication(ctxDevice, ctxMusikServiceSSH, [ctxDeviceMAD, ctxDeviceEAS]);
```

Network property

```
#context ctxAnyIP = (IP=".*");
#context ctxHttpPort = (Port="80");
#context ctxMADHttpPort = {Airport_MAD.seed4c_musik} :ctxHttpPort;
#property Access (ctxMADHttpPort, ctxAnyIP);
```

Screenshot of Airport Management Application (called “Musik”) in Sam4C Modeling Tool
#context ctxFileMusik = (Data="File"): (AppDomain="App_Musik") ;

```
0 */opt/musik(/.*)?" ctxFileMusik
0 */opt/musik/properties(/.*)?" ctxConfigMusik
0 */opt/musik/musik.lic" ctxConfigMusik
0 */opt/musik/rsc(/.*)?" ctxConfigMusik
0 */opt/musik/log(/.*)?" ctxLogMusik
0 */opt/musik/webapps/musik.war" ctxBinaryMusik

0 */opt/apache-.*/conf/Catalina/localhost(/.*)?" ctxConfigMusik

0 */opt/apache-.*(/.*)?" ctxFileMusik
0 */opt/apache-.*/bin(/.*)?" ctxBinaryMusik
0 */opt/apache-.*/lib(/.*)?" ctxBinaryMusik
0 */opt/apache-.*/webapps/*.war" ctxBinaryMusik
0 */opt/apache-.*/conf(/.*)?" ctxConfigMusik
0 */opt/apache-.*/log(/.*)?" ctxLogMusik

0 */opt/devmconn(/.*)?" ctxFileMusik
0 */opt/devmconn/bin(/.*)?" ctxBinaryMusik
0 */opt/devmconn/lib(/.*)?" ctxBinaryMusik
0 */opt/devmconn/log(/.*)?" ctxLogMusik
0 */opt/devmconn/conf(/.*)?" ctxConfigMusik
0 */opt/devmconn/properties(/.*)?" ctxConfigMusik
```
Deployment
Placement with Constraints

- Resources constraints
  - CPU,
  - RAM,
  - Disk,
  - Location.

- Security constraints
  - Integrity
  - Isolation
  - etc.

Sam4C Deployment Tool
**Placement-based security**
- **Idea**: Do not share physical machines with unwanted neighborhood.
- **Properties**: Isolation, Integrity, Confidentiality between VMs.
- **Innovation**: Core, Cache, RAM granularity.

**Security properties matching**
- **Idea**: A VM / PM provides security capabilities via a SEE.
- **Properties**: All (except properties between VMs).
- **Innovation**: Independent from security mechanism implementation.
Security Policy Enforcement

SEE: Secure Element Extended

- Receives properties from Sam4C
  - Expressed using capabilities
    - Capabilities abstract the mechanisms’ functions (generate_key, open_port...)
    - Confidentiality: generate_key (SE, JCE...) + encrypt_flow (SSH, OpenVPN...)
  - Select mechanisms to enforce the security properties
  - Automatically configure the mechanisms
Cooperative Security: the SEE model

- Secure Element (SE)
  - Multi-tenants hardware component (isolated security domains)
  - Cryptographic functions
  - SEE: Extends the SE model to other mechanisms
  - SEE: Uses the SE’s security services
  - Two admin domains: NoSEE admin /Tenant admin

\[\text{SE}\]

\begin{align*}
\text{Tenant 1 security domain} & \\
\text{Data} & \\
- \text{Keys} & \\
- \text{Functions} & \\
- \text{...} & \\
\text{Tenant 2 security domain} & \\
\text{Data} & \\
- \text{Keys} & \\
- \text{Functions} & \\
- \text{...} & \\
\text{Shared security domain} & \\
\text{Data} & \\
- \text{Location} & \\
- \text{Time/date} & \\
- \text{...} & \\
- \text{-Functions} & \\
- \text{Encrypt/decrypt} & \\
- \text{...} & \\
\end{align*}

SE = SmartCard, MicroSD, etc.
Security Policy Assurance

- Assurance checks are generated during the enforcement step:
  - Check the status of the mechanisms
  - Check the enforcement of the security properties

- Assurance Engine:
  - Collect assurance data from assurance mechanisms configured by the SEE
  - Send data to the assurance dashboard
Conclusion

**SEED:** *A minimal trusted computing base spread within the cloud*

A network of seeds will provide a **Trusted Cloud Computing Base**

Ensure the end-to-end security of “cloudified” apps

⇒ [http://www.celticplus-seed4c.org/](http://www.celticplus-seed4c.org/)