TREDISEC Framework

Overview of the GUI

March 2018
Table of Content

- Framework Landing
- Commons
- Search
- Creation of artefacts
- Management of artefacts: tenancy access
- Modifications to Artefacts
- Removing artefacts
- Deprecated artefacts
- Administrator features
- Testing and Deployment
TREDISEC - Framework

Search

Use

Create

Manage

TREDISEC Framework GUI Tour, March 2018
TREDISEC - Framework

Support of different roles

User-friendly

Automatic deployment and testing

Repository of security and functional solutions
Home – Log In

Username: 
Password: 
Login

This website is the front-end of a software developed in the context of the TREDISEC project. This project has received funding from the European Union’s Horizon 2020 (H2020) research and innovation programme under Grant Agreement no 644412. This website and the content displayed in it do not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content.
Home – Logged In

My Security Primitive Patterns

Multitenancy ABAC primitive pattern

Author: beac
Tenant: ATOS
Documentation status: Passed

My Security Primitive Implementations

TRAVIS

Author: beac
Tenant: ATOS
Documentation status: Passed
Version: 1.0
Resources

TREDISEC Resources

TREDISEC Framework User Manual:

- HTML
- PDF

Creation Process:

- Primitive Pattern
- Primitive Implementation
- TREDISEC Recipe

Templates:

- Primitive Pattern
- Primitive Implementation
- TREDISEC Recipe
User Profile
# Show/Edit Profile

<table>
<thead>
<tr>
<th>beac</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fullname</strong></td>
</tr>
<tr>
<td><strong>Tenant</strong></td>
</tr>
<tr>
<td><strong>e-mail</strong></td>
</tr>
<tr>
<td><strong>Bio</strong></td>
</tr>
<tr>
<td><strong>Security Primitive Patterns</strong></td>
</tr>
<tr>
<td><strong>Security Primitive Implementations</strong></td>
</tr>
<tr>
<td><strong>TREDISEC Recipes</strong></td>
</tr>
</tbody>
</table>

- 
- ATOS
- beatriz.gallego-nicasio@atos.net
- Senior Research Engineer and Project Manager at Cybersecurity Lab in Atos Research and Innovation.
- 1
- 2
- 0

[Edit profile]
Password Change
Search - Basic

My Security Primitive Patterns

Multitenancy ABAC primitive pattern

Author: beat
Tenant: ATOS

Documentation status: Passed

My Security Primitive Implementations

TRAVIS

Author: beat
Tenant: ATOS

Documentation status: Passed
Version: 1.0
Pattern: Remote Attestation Component

TREDISEC Framework GUI Tour, March 2018
Search – Basic - Results

Security Primitive Patterns (1) found

Multitenancy ABAC primitive pattern

Author beac
Tenant ATOS

Documentation status: Passed

Security Primitive Implementations (0) found

TREDISEC Recipes (0) found
CREATION OF ARTEFACTS: PATTERNS, IMPLEMENTATION AND RECIPES
Creation of Security Primitive Patterns

TREDISEC Framework GUI Tour, March 2018
Wizard Step I
Creation of Security Primitive Implementations (Mode I)

**EPICA**
- **Author**: beas
- **Tenant**: ATOS
- **Documentation status**: Passed
- **Version**: 1.0
- **Pattern**: Multitenancy ABAC primitive pattern

**Container Isolation**
- **Author**: dimbro
- **Tenant**: GRNET
- **Documentation status**: Working
- **Version**: 0.1
- **Pattern**: Container Isolation

**TRAVIS**
- **Author**: beas
- **Tenant**: ATOS
- **Documentation status**: Passed
- **Version**: 1.0
- **Pattern**: Remote Attestation Component

TREDISEC Framework GUI Tour, March 2018
Wizard Step I

Security Primitive Implementation name *

Version *

Keywords
Comma separated list of keywords

Security Primitive Pattern *
Select the security pattern this package implements.
- Data Provisioning

Security Primitive Implementation package *
The Security Primitive Implementation package should be a ZIP archive.

[Choose File] No file chosen

[Cancel] [Submit]
Creation of Security Primitive Implementations (Mode II)

Remote Attestation Component

Overview

Remote Attestation is the activity of making a claim about properties of a target by supplying evidence to an appraiser over a network.

The Remote Attestation Component is composed of the appraiser part (Remote Attestation Client) that verifies the evidence, and the target part (Remote Attestation Server) which generates the evidence of whether or not the untrusted platform is running in the known state, and therefore, the result of the Third Party Solution is trustworthy.

Remove  Modify  Documentation  Manage Tenancy Access  Create Security Primitive Implementation
Wizard Step I

Security Primitive Implementation name *

Version *

Keywords
Comma separated list of keywords

Security Primitive Pattern *
Select the security pattern this package implements.

- Remote Attestation Component

Security Primitive Implementation package
The Security Primitive Implementation package should be a ZIP archive.

[Choose File] No file chosen

[Cancel] [Submit]
## Creation of TREDISEC Recipes (Mode I)

<table>
<thead>
<tr>
<th>Recipe</th>
<th>Author</th>
<th>Tenant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container Isolation</td>
<td>dimitro</td>
<td>GRNET</td>
</tr>
<tr>
<td>TRAVIS Recipe</td>
<td>carlos</td>
<td>ATOS</td>
</tr>
<tr>
<td>EPICA</td>
<td>carlos</td>
<td>ATOS</td>
</tr>
</tbody>
</table>
Creation of TREDISEC Recipes (Mode II)

Container Isolation

Overview

The Container Isolation module provides two functionalities. First, it implements a tool used to extract and encrypt a Docker container image layer in order to safely transfer it into a target Docker host. This way, the confidential data lying inside the image's top layer remain secure throughout the entire migration process. Second, it enables a container to store its data over encrypted storage mediums, in order to ensure that the confidential data cannot be retrieved by an adversary with access in the host's storage backend.

TREDISEC Framework GUI Tour, March 2018
Wizard Step I

TREDISEC Recipe name *

Version *

Testing Environment *
Select the Testing Environment for this recipe. Unsupported Testing environments are marked with (*)

TTE 02

Keywords
Comma separated list of keywords

Security Primitive Implementation *
Select the Security Primitive Implementations this recipe contains.

Container Isolation

TREDISEC Recipe package
The TREDISEC Recipe package should be a ZIP archive.

Choose File  No file chosen

Cancel  Submit
MANAGEMENT OF ARTEFACTS
TENANCY ACCESS
# All Artefacts Wizard – Step II

## Manage Tenancy Access

### Multitenancy ABAC primitive pattern

- **Author**: beac
- **Tenant**: ATOS

## Manage Access Level

There are currently three access levels:

- **Public**: The package is visible by all users.
- **Protected**: The package is visible as configured with the View and Download permissions below.
- **Private**: The package is visible only by users that with the same tenancy as the owner.

Select access level:

### Manage Tenancy Protected Access

- **Manage tenancy view access.**
  The users with the selected tenancy will be able to view the package but will not be able to download it.

  Select view access:

  - Nothing selected

- **Manage tenancy download access.**
  The users with the selected tenancy will be able to download and use the package.

  Select download access:

  - Nothing selected

Submit
MODIFICATIONS TO ARTEFACTS
Click on the Artefact Name
Artefact Details Page

Multitenancy ABAC primitive pattern

Overview

The aim of the primitive is to provide an enforcement component for distributed attribute-based access control (ABAC) policies that ensures that authorized users always get access to the selected cloud resource (either data or service) whilst the access is refused to malicious parties, in the context of a multitenant cloud infrastructure.

This primitive addresses requirements from the TREDISEC access control model supporting multitenancy, which is based on an ABAC model. This advanced access control mechanisms will allow tenants to define fine-grained policies on a per user basis, and which can be easily integrated in current cloud environments. Using this access control model the cloud service providers will be able to share resources (any kind of resource) among their users.

The main components of the access control model addressed by this primitive are:

- Policy Enforcement Point (PEP): for access request coming from outside the tenant, the access request is submitted to this component in order to generate the XCANL request to be sent to the PDP.
- Policy Decision Point (PDP): The PDP is the component which evaluates the access request against the policies in order to protect the resource trying to be accessed. Once the decision is made, a response will be sent back to the requesting PEP to take affect with respect to the requested access.
- Policies Repository: it stores the access control policies. Trust relations among tenants could be defined and supported extending the policies.
- Resource Manager (RM): the manager of the resource on which the Cloud User requests permission to perform an action. The Resource Manager may not have a direct relation with the cloud provider. The cloud provider assigns resources to a Resource Manager and the Resource Manager can assign permissions for the Cloud Users on the resources. For simplicity we will consider the Resource Manager as a component of the cloud system.
Wizard Step 1

Security Primitive Pattern name
Name of the security primitive pattern.
Multitenancy ABAC primitive pattern

Keywords
Comma separated list of keywords
abac, multitenancy

Documentation
Documentation should be a ZIP archive with a valid format.
Choose File No file chosen
## Manage Tenancy Access

### Multitenancy ABAC primitive pattern

<table>
<thead>
<tr>
<th>Author</th>
<th>beac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>ATOS</td>
</tr>
</tbody>
</table>

### Manage Access Level

There are currently three access levels:
- Public: The package is visible by all users.
- Protected: The package is visible as configured with the View and Download permissions below.
- Private: The package is visible only by users that have the same tenancy as the owner.

- **Protected**

### Manage Tenancy Protected Access

- **Manage tenancy view access.**
  - The users with the selected tenancy will be able to view the package but will not be able to download it.

  - **Nothing selected**

- **Manage tenancy download access.**
  - The users with the selected tenancy will be able to download and use the package.

  - **Nothing selected**

[Submit]
REMOVE ARTEFACTS
Select Artefact

Click on the Artefact Name

Data Provisioning
Author: andres
Tenant: SAP
Documentation status: Passed

Multiterancy ABAC primitive pattern
Author: beac
Tenant: ATOS
Documentation status: Passed

Data provisioning & Optimized Encryption
Author: benny
Tenant: SAP
Documentation status: Passed

Remote Attestation Component
Author: jose
Tenant: ATOS
Documentation status: Passed

Secure Deletion
Author: rikutait
Tenant: ETH
Documentation status: Passed
Multitenancy ABAC primitive pattern

Overview

The aim of the primitive is to provide an enforcement component for distributed attribute-based access control (ABAC) policies that ensures that authorized users always get access to the selected cloud resource (either data or service) whilst the access is refused to malicious parties in the context of a multi-tenant cloud infrastructure.

This primitive addresses requirements from the TREDISEC access control model supporting multitenancy, which is based on an ABAC model. This advanced access control mechanism will allow tenants to define fine-grained policies on a per user basis, and which can be easily integrated in current cloud environments. Using this access control model the cloud service providers will be able to share resources (any kind of resource) among their users.

The main components of the access control model addressed by this primitive are:

- Policy Enforcement Point (PEP): for access request coming from outside the tenant, the access request is submitted to this component in order to generate the XCANL request to be sent to the POP.
- Policy Decision Point (PDP). The PDP is the component which evaluates the access request against the policies in order to protect the resource trying to be accessed. Once the decision is made, a response will be sent back to the requesting PEP to take effect with respect to the requested access.
- Policies Repository: it stores the access control policies. Trust relations among tenants could be defined and supported extending the policies
- Resource Manager (RM): the manager of the resource on which the Cloud User requests permission to perform an action. The Resource Manager may not have a direct relation with the cloud provider. The cloud provider assigns resources to a Resource Manager and the Resource Manager can assign permissions for the Cloud Users on the resources. For simplicity we will consider the Resource Manager as a component of the cloud system.
Confirmation

Are you sure you want to delete the following Pattern?

Multitenancy ABAC primitive pattern

Author: deac
Tenant: ATOS

Keywords:
- abac
- multitenancy

[Cancel] [Confirm]
DEPRECIATED ARTEFACTS
Rationale

• User modifies/removes an artefact
• The removed artefact has dependent artefacts
  – Patterns with related implementations
  – Implementations with related Recipes
• The dependent artefacts are marked as “Deprecated”
  – Artefacts are still available for use
  – Owners are warned to proceed with updates
ADMINISTRATOR FEATURES
Administrative Roles

- Framework Administrator
- Tenant Administrator
Creation and Management of Users/Tenants

- Tenants
  - groups of users that belong to the same organization or entity
  - share permissions over artefacts configured as Protected
  - Tenant Administrators manage the users and permissions for Tenant group
Management of Testing Environments

TREDISEC Testing Environments:

<table>
<thead>
<tr>
<th>Environment name</th>
<th>Keywords</th>
<th>Tenancy</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOS-TTE provided by ATOS</td>
<td></td>
<td>ATOS</td>
<td></td>
</tr>
<tr>
<td><a href="http://37.48.68.242:9494">http://37.48.68.242:9494</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTE 03</td>
<td>Docker</td>
<td>GRNET</td>
<td></td>
</tr>
<tr>
<td><a href="http://snf-776700.vm.oceano.grnet.gr:8080">http://snf-776700.vm.oceano.grnet.gr:8080</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>testing-tte</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Not-linked Testing Environments:

<table>
<thead>
<tr>
<th>Environment name</th>
<th>Keywords</th>
<th>Tenancy</th>
<th>Actions</th>
</tr>
</thead>
</table>
Testing Environment Wizard

Platform name *
ATOS-TTE provided by ATOS

Host
If the platform is supported by the TREDISEC Framework please add the url/hostname of the server. If the platform is unsupported leave this field empty.
http://37.48.68.242:9494

Description
A short description of the platform

Keywords
Comma separated list of keywords

Tenant
ATOS
TESTING AND DEPLOYMENT
Testing / Deployment of Recipes

Cloud Platform supported

Automatic deployment is marked with a lightning bolt icon

EPICA Recipe

Documentation status: Passed
Platform: TTE 03 / Automatic Deploy
Version 1
Depreduced / Dependency updated

Author: carlos
Tenant: ATOS

Keywords:
- access control
- multitenancy

Overview

EPICA (Efficient and Privacy-respectful Interoperable Cloud-based Authorization) is a software component that implements the primitive pattern Multi-tenancy Attribute-based Access Control Primitive. The primitive pattern addresses an Attribute-Based Access Control (ABAC) model that satisfies a set of requirements that are crucial in order to maintain the cost reductions and high performance that cloud infrastructures provide.

EPICA builds upon version 3 of the XACML standard and advances the WSO2 Balanza Open Source implementation by extending it with new functionalities, improving existing ones and addressing some of the most important challenges that multi-tenancy poses to traditional authorization schemes (i.e. trust management, common vocabulary, tenant's policy isolation and shared resources), while remaining fully compatible with most popular storage efficiency implementation techniques.

Besides, EPICA supports high availability and performance deployments, implementing an efficient policy retrieval approach with scalable policy stores, which has been validated with the requirements of a real use case scenario that have a load balancing scheme in place. The architecture of EPICA has been designed taking into account interoperability and privacy concerns, so the information exchanged between the cloud provider and the user, required to perform authorization, remains minimal.

TREDISEC Framework GUI Tour, March 2018
Select Action: testing

Step 1

Select action:
- Deploy
- Testing

Step 2

Select action:
- Testing

Step 3: The testing process is running in background
Testing: Results

Notification: process finished

Results of the execution are rendered here
Select Action: deployment

Step 1

Select action

- Deploy
- Testing

Step 2

Select action

- Deploy

Step 3: The deployment process is running in background
Deployment: Results

TREDISEC Framework GUI Tour, March 2018
Making use of the deployed recipe

TREDISEC Testing Environments:

<table>
<thead>
<tr>
<th>Environment name</th>
<th>Keywords</th>
<th>Tenancy</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATOS-TTE provided by ATOS</td>
<td></td>
<td>ATOS</td>
<td></td>
</tr>
<tr>
<td><a href="http://37.48.68.242:9494">http://37.48.68.242:9494</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTE-03</td>
<td>GRNET</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The recipe is successfully deployed in this virtual environment

You can use this URL to connect to the environment via SSH (ask admin for credentials)

You can use this URL to remotely access API and services exposed by the primitive (if that is the case).

Just make sure the Testing Enviroment Admin provided you with the right permissions (ports opened, credentials, etc.)
Non-automatic: Testing/Deployment

Non-automatic recipes do not have the lightning bolt icon.

You can download the software package, deploy and run the tests on your own. Following instructions provided.
Thank you!

Contact:
Beatriz Gallego-Nicasio Crespo
TREDISEC project coordinator
Atos Research & Innovation
beatriz.gallego-nicasio@atos.net

Rodrigo Diaz Rodriguez
Head of Cybersecurity Lab
Atos Research & Innovation
rodrigo.diaz@atos.net

The work described in this presentation has been conducted within the project TREDISEC. This project has received funding from the European Union’s Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 644412. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of its content.