TREDISEC Framework architecture

March 2018
Introduction and Resources

• Goals of TREDISEC:
  • Design and develop solutions that fulfill both security and functional requirements of cloud systems
  • Develop a framework that supports the creation, management and use of such solutions

• References (http://www.tredisec.eu/deliverables):
  • D2.3. TREDISEC architecture and initial framework design
  • D2.4. Final Architecture and Design of the TREDISEC Framework
  • D6.1 TREDISEC framework implementation
Framework Overview

TREDISEC Framework architecture, March 2018
Framework Overview

- Multi-purpose system that supports the life-cycle of the security primitives
- Facilitates creation/management/testing of security primitive implementations and TREDISEC Recipes
- Supports different cloud platforms for development and testing of primitives and recipes
High-Level Architecture

TREDISEC Framework architecture, March 2018
Architecture Components

• **UI**
  - Orchestrator of the other modules
  - Graphical and API
  - Different functionalities according to the role

• **Security Primitive Component**
  - Manages the security primitives in the framework
  - Provides testing and deployment information
  - Provides a repository for security primitives
Architecture Components

• **TREDISEC Recipe Component**
  - Management of TREDISEC Recipes
  - Testing and deployment information
  - Repository for TREDISEC Recipes

• **Testing Component**
  - Manages testing information of security primitive implementation and TREDISEC Recipes
  - Different testing strategies
  - Stores, manages and allows access to testing results
Architecture Components

• **Deployment Component**
  • Provides support for the deployment, build and configuration of security primitive implementations and TREDISEC Recipes

- Deployment package
  - Written information (diagrams or screenshots if any)
  - Scripts (e.g. executables, batch files, etc.)
  - Application-specific scripts (e.g. Docker, etc.)
TREDISEC Framework - Roles

- **TREDISEC Security Admin**
  - Manages the TREDISEC Framework (administrative point of view)
  - Checks correct functionality and availability of the framework
  - Verifies security primitive implementations and TREDISEC Recipes

- **TREDISEC End-user**
  - Searches and uses TREDISEC Recipes
  - Could represent a single user, organisation, etc.
TREDISEC Framework - Roles

- **Security Expert Engineer**
  - Creates security primitive patterns
  - Focuses in architectural solutions (security and functional)

- **Security Technology Provider**
  - Creation and management of security primitive implementations and TREDISEC Recipes
  - Uses testing component
  - Creates testing procedures
  - Creates deployment procedures
TREDISEC Framework – Security Primitives

Security Primitive Pattern
- Name and identifier
- Security and functional capabilities (domain specific)
- Description and goal
- Description of the solution (high level)
- Requirements, threats and vulnerabilities

Security Primitive Implementation
- Name and identifier
- Information derived from the security primitive pattern it implements
- Description and goal (implementation specific)
- Implementation specific info:
  - Description of the solution
  - Known conflicts
  - Known incompatibilities
  - Testing information
- Requirements, threats, attacks, etc. (implementation specific)

TREDISEC Recipe
- Name and identifier
- Information derived from the security primitive implementations
- Security and functional capabilities it provides
- Applicability of the TREDISEC Recipe
- Deployment information
- Testing and validation information

TREDISEC Framework architecture, March 2018
TREDISEC Framework –
Security Primitives

• Security Primitive Pattern
  • Abstract (high level definition) of a security primitive
  • Describes security and functional requirements
  • Applicability and high-level interfaces

• Security Primitive Implementation
  • Instantiation of a security primitive pattern
  • Implementation specific information of a security primitive pattern
  • Testing information
  • Deployment information
TREDISEC Framework – Security Primitives

• TREDISEC Recipe
  • Used in target cloud system
  • Integrates one or more security primitive implementations
  • Testing information
  • Deployment information
TREDISEC Framework - Process
TREDISEC Framework - Process

• Development mode
  • Security Experts design Security Primitive Patterns and document these in the framework
  • Security Technology providers implement Security Primitive Patterns choosing specific technologies to develop Security Primitive Implementations
  • Security Technology providers uses the cloud-based testing environments to support their own Continuous Integration and Improvement software lifecycle
TREDISEC Framework - Process

• **Maintenance Mode**
  • Security Experts design *Security Primitive Patterns*, document these in the framework and share with the community
  • Security Technology providers document their *Security Primitive Implementations*, optionally including specific information for testing and deployment, and share these with the community
  • The Community provides *feedback* that Security Experts and Technology Providers can use to improve their primitives
TREDISEC Framework - Process

• Provisioning Mode

  • TREDISEC Recipes permit building and configuring one or more primitive implementations for specific Cloud-based environments, in automatic, semi-automatic or manual mode. Making them available to end-users for easy deployment and integration in their own technologies.
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.