10th UCAAT User Conference on Advanced Automated Testing

Applying TDL-TO in Automated Static and Dynamic Application Security Testing

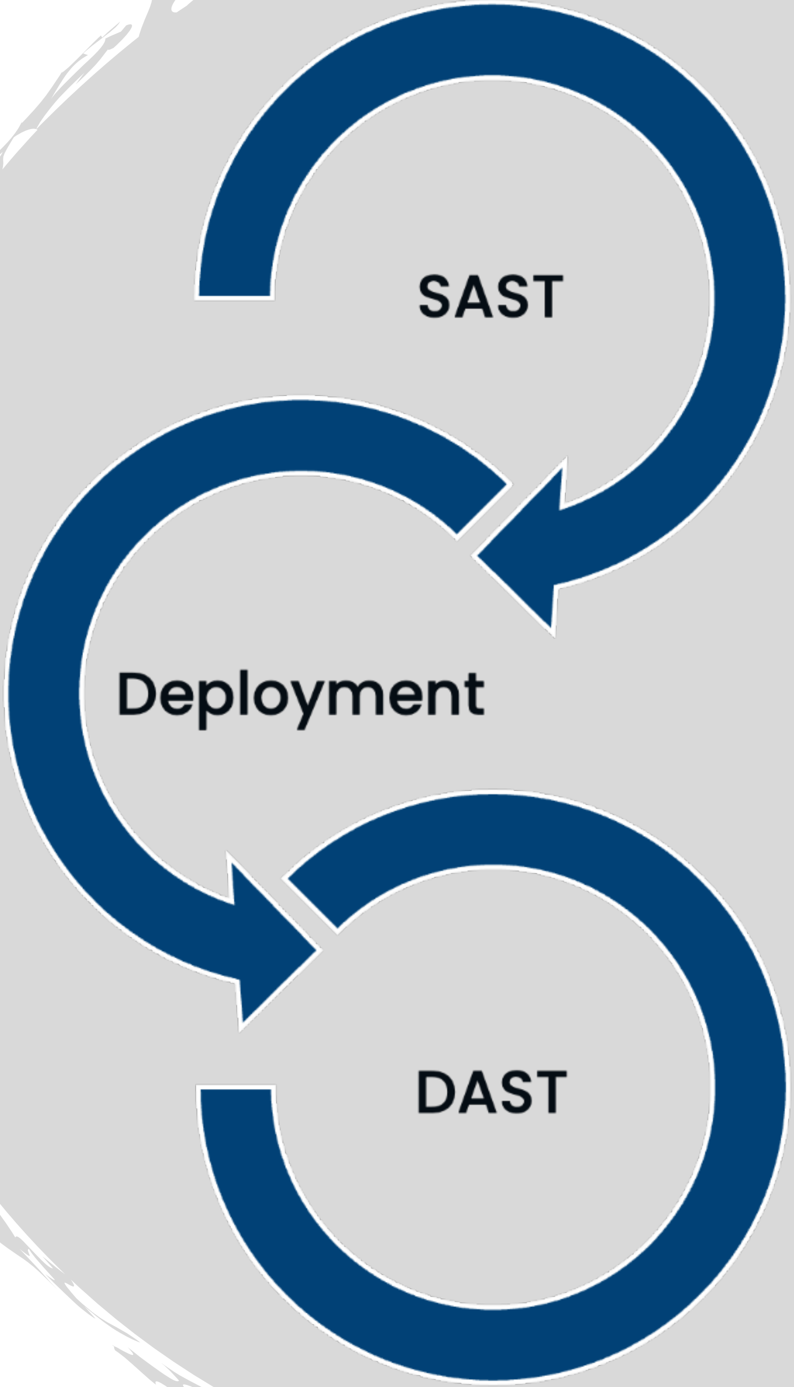
Dr Marija Jankovic

Centre for Research and Technology Hellas (CERTH)

ETSI TC MTS Vice Chair

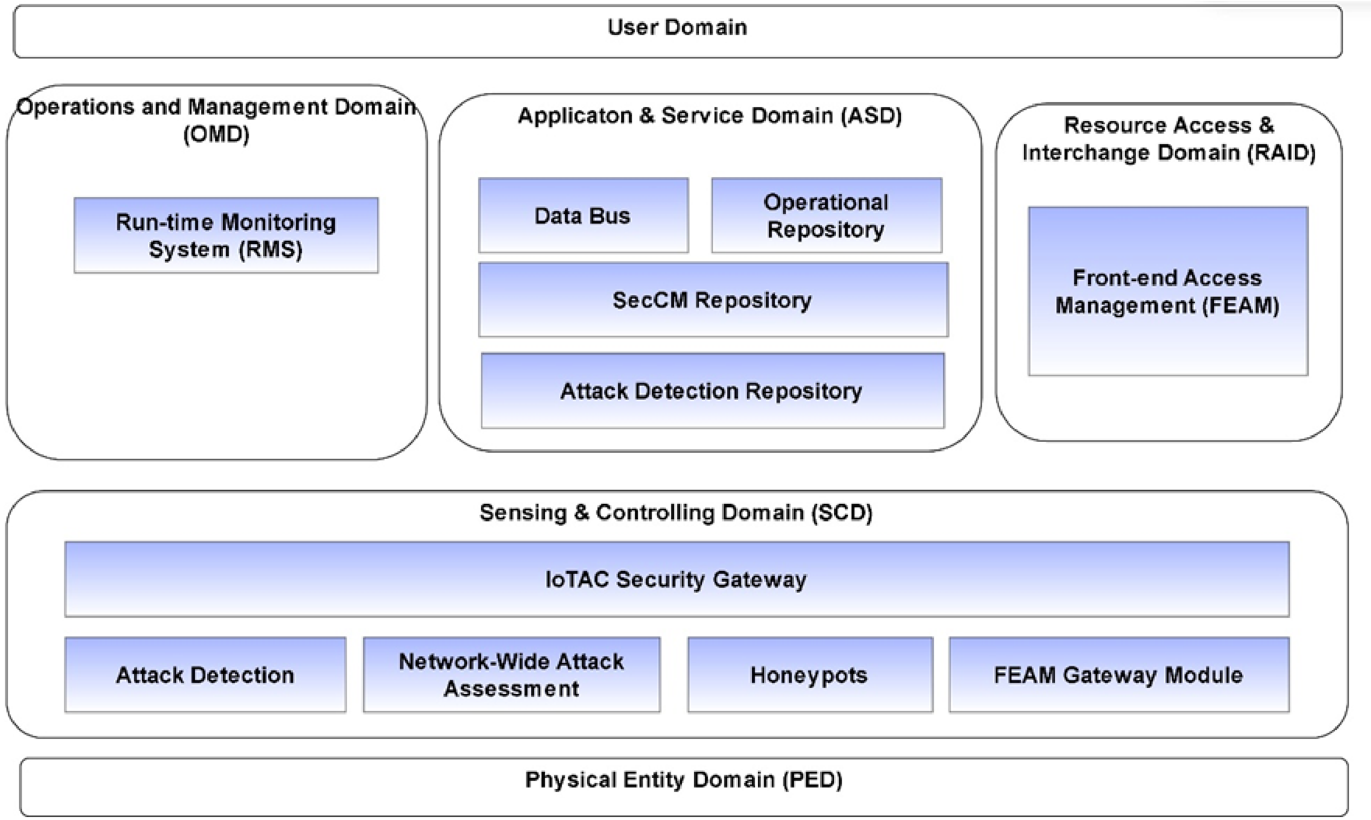
CERTH CENTRE FOR RESEARCH & TECHNOLOGY HELLAS

15/11/2023



Summary

- Goal – to investigate the use of TDL-TO for test specification in DevSecOps pipelines
- IoTAC Reference Architecture – extended ISO/IEC 30141 IoT RA


User Domain
Operations and Management Domain (OMD)
Run-time Monitoring System (RMS)
Applicaton & Service Domain (ASD)
Data Bus
Operational Repository
SecCM Repository
Attack Detection Repository
Resource Access & Interchange Domain (RAID)
Front-end Access Management (FEAM)
Sensing & Controlling Domain (SCD)
IoTAC Security Gateway
Attack Detection
Network-Wide Attack Assessment
Honeypots
FEAM Gateway Module
Physical Entity Domain (PED)

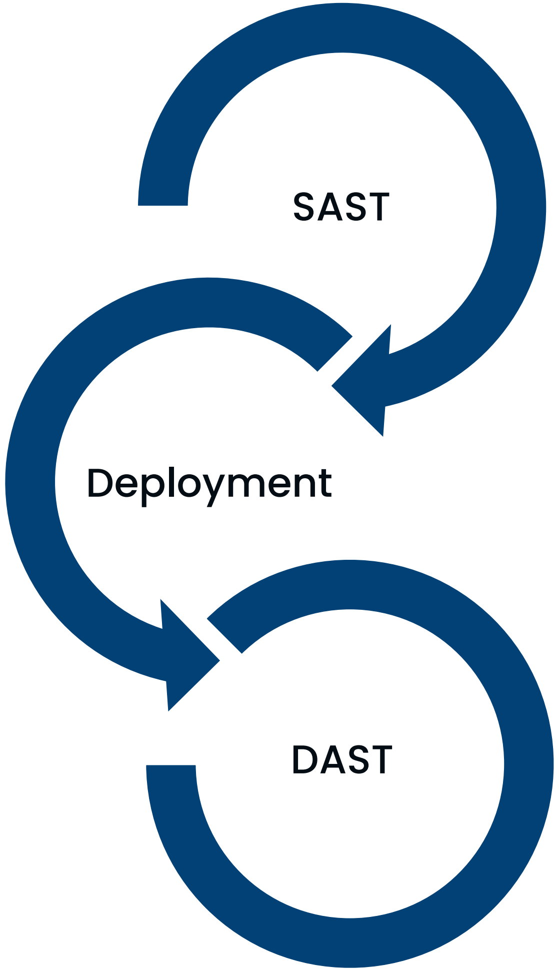
IoTAC Runtime Modules

- Front-end Access Management
- Security Gateway
- Run-time Monitoring System
- Attack Detection
- Honeypots

From DevOps to DevSecOps

The approach for testing the IoTAC runtime components

- Functional security testing
  - Testing of the functional requirements of IoTAC run-time modules
  - Focus on requirements traceability
  - TDL-TO has proven to be effective for the specification of functional TP
  - Inter-component (48 TDL-TO TP) &Intra-component (5 TDL-TO TP)
- Static Application Security Testing (SAST)
  - A white-box testing method that analyses the source code of an application for security vulnerabilities without executing it
  - Successful translation of SAST program-specific rules to TDL-TO TP
  - 8 SAST TPs
- Dynamic Application Security Testing (DAST)
  - A black-box testing method on the running application to identify vulnerabilities.
  - Proved to be challenging due to the structured nature of the TDL-TO



Results

- ETSI TS 103 942 V1.1.1 (2023-11): IoT Security Functional Modules
- Specification of the IoTAC functional modules
- Methodology for defining functional and SAST TDL-TO test purposes
- Detailed list of test purposes
- Recomendations for integrating TDL-TO into DevSecOps pipelines

Suggestion 1: Creating a CLI or Microservice

- To be incorporated into an automation server (e.g., Jenkins) TDL-TO must be offered as a Command Line Interface / Tool.
- Alternatively, as a microservice (running locally or in the Cloud) exposing an API that can be invoked using the “curl” command or a script.

Suggestion 2: Defining Quality Gates via TDL-TO

- To standardize TDL-TO syntax for the definition of customizable Quality Gates

Thank you!

jankovicm@iti.gr