ANASTACIA

Advanced Networked Agents for Security and Trust Assessment in CPS/IoT Architectures

D6.2

Initial use cases implementation and tests Report

This deliverable presents the results of ANASTACIA Task 6.2. The aim of the task is to define the deployment and operational aspects of the use cases in the frame of ANASTACIA, as well as to support the implementation and testing of the considered use cases.

Distribution level	PU
Contractual date	31.05.2018 [M17]
Delivery date	04.06.2018 [M18]
WP / Task	WP6
WP Leader	UBITECH
Authors	Yacine Khettab, Ivan Farris (AALTO), Diego Rivera (MONT), Rafael Marín Pérez (ODINS), Jorge Bernal, Alejandro Molina (UMU), Dallal Belabed (THALES), Alie El-Din Mady, Piotr Sobonski, Deepak Mehta (UTRC),
EC Project Officer	Carmen Ifrim carmen.ifrim@ec.europa.eu
Project Coordinator	Softeco Sismat SpA Stefano Bianchi Via De Marini 1, 16149 Genova – Italy +39 0106026368 stefano.bianchi@softeco.it
Project website	www.anastacia-h2020.eu

ANASTACIA has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement N° 731558 and from the Swiss State Secretariat for Education, Research and Innovation

Table of contents

PUBLIC SUMMARY 5
1 Introduction 6
1.1 Aims of the document 6
1.2 Applicable and reference documents 6
1.3 Revision History 6
1.4 Acronyms and Definitions 7
2 KPI implementation in ANASTACIA test cases 8
2.1 Methodology 8
2.2 Evaluation framework 9
2.3 Key Performance Indicators 9
2.3.1 External KPIs processing 9
3 Test Use-Cases 11
3.1 Test Case Naming convention 11
3.2 Test Case Identification process 11
3.3 Test-Cases TC_BMS.2: Insider Attack on the Fire Suppression System 12
3.3.1 Individual Component Test-Cases 12
3.3.2 Components Interaction Test-Cases 13
3.3.3 Integration Test-Case 21
3.4 Test-Case TC_BMS.3: Remote Attack on the Building Energy Microgrid 22
3.4.1 Individual Component Test-Cases 24
3.4.2 Components Interaction Test-Cases 26
3.4.3 Integration Test-Cases 29
3.5 Test-Case TC_BMS.4: Cascade Attack on a Megatall Building 34
3.5.1 Individual Component Test-Cases 35
3.5.2 Components Interaction Test-Cases 39
3.5.3 Integration Test-Case 40
3.6 Test-Case TC_MEC.3: DoS/DDoS Attacks using Smart Cameras and IoT Devices 43
3.6.1 Individual Component Test-Cases 43
3.6.2 Components Interaction Test-Cases 44
3.6.3 Integration Test-Case 50
3.7 End-User Questionnaire 52
3.7.1 Questionnaire for TC_BMS.2.9 52
3.7.2 Questionnaire for TC_BMS.3.5 53
3.7.3 Questionnaire for TC_BMS.3.6 53

3.7.4 Questionnaire for TC_BMS.4.4 53
3.7.5 Questionnaire for TC_MEC.3.7 54
4 Summary and future work 55
5 References 56
6 Appendix 57
6.1 Example of test event message for external KPI report 57

Index of figures

Figure 1. ANASTACIA testing methodologies scope. 8
Figure 2. ANASTACIA test case KPI measurement architecture. 10
Figure 3. ANASTACIA test case identification methodology steps. 12
Figure 4 Sequence diagram for TC_BMS.2.2 – network Authentication and device registration 13
Figure 5 Sequence diagram for TC_BMS.2.3 – unauthorized actuation detection 15
Figure 6 Sequence diagram for TC_BMS.2.4 – monitoring management 16
Figure 7 Sequence diagram for TC_BMS.2.5 – seal management 17
Figure 8 Sequence diagram for TC_BMS.2.6 – reaction management 18
Figure 9 Sequence diagram for TC_BMS.2.7 – orchestration management 19
Figure 10 Sequence diagram for TC_BMS.2.8 – enforcement management 20
Figure 11 Sequence diagram for TC_BMS.2.9 – full integration of ANASTACIA framework for use case BMS.2 21
Figure 12 BMS.3 Test Case General Workflow 23
Figure 13 Sequence diagram for TC_BMS.3.1 – MMT-Probe basic network analysis 24
Figure 14 Sequence diagram for TC_BMS3.2 – Monitoring Module detection 25
Figure 15 Sequence diagram for TC_BMS.3.3 – Reaction to SQLi attack on SCADA network 27
Figure 16 Sequence diagram for TC_BMS.3.4 – Full BMS.3 scenario test with Legit SQL traffic 28
Figure 17 Sequence diagram for TC_BMS.3.5 – Full BMS.3 scenario test 30
Figure 18 Test case TC_BMS.3.6 – Reaction Enforcement Process 32
Figure 19. SW architecture for test case TC_BMS_4 34
Figure 20. Temperature script change algorithm that will emulate adversary in final test case TC_BMS_4.4. 35
Figure 21. Sequence diagram for TC_BMS_4.1 – connection with OdinS IoT broker. 36
Figure 22. Sequence diagram for TC_BMS_4.2 – attack detection based on IoT monitoring data 37
Figure 23. Sequence diagram of TC_BMS_4.3 – reaction to attack on sensor. 39
Figure 24. Sequence diagram of TC_BMS_4.4 – Full BMS.4 scenario test. 41
Figure 25. Sequence diagram of TC_BMS_4.4 – Details of orchestration processing section. 42
Figure 26 Sequence diagram for TC_MEC3.1 – Device actuation 43

Figure 27 Sequence diagram for TC_MEC3.2 – Monitoring processing 45
Figure 28 Sequence diagram for TC_MEC3.3 – Reaction processing 46
Figure 29 Sequence diagram for TC_MEC3.4 – Seal processing 47
Figure 30 Sequence diagram for TC_MEC3.5 – Orchestration processing 48
Figure 31 Sequence diagram for TC_MEC3.6 – Enforcement 49
Figure 32 Sequence diagram for TC_MEC3.7 – Full MEC.3 test case 50

Index of tables

Table 1. Test case TC_BMS.2.1 – Validating a capability token for authorization of device actuation 12
Table 2. Test case TC_BMS.2.2 – Network Authentication and Device Registration 14
Table 3. Test case TC_BMS.2.3 – Unauthorized Actuation Detection 15
Table 4. Test case TC_BMS.2.4 – Monitoring Management 16
Table 5. Test case TC_BMS.2.5 – Seal Management 17
Table 6. Test case TC_BMS.2.6 – Reaction Management 18
Table 7. Test case TC_BMS.2.7 – Orchestration Management 19
Table 8. Test case TC_BMS.2.8 – Enforcement Management 20
Table 9. Test case TC_BMS.2.9 – Full integration of ANASTACIA framework for use case BMS.2 22
Table 10. Test case TC_BMS.3.1 – MMT-Probe basic network analysis 24
Table 11. Test case TC_BMS.3.2 – Monitoring Module detection 25
Table 12. Test case TC_BMS.3.3 – Reaction to SQLi attack on SCADA network 27
Table 13. Test case TC_BMS.3.4 – Monitoring and Reaction BMS.3 scenario test with Legit SQL traffic 28
Table 14. Test case TC_BMS.3.5 – Monitoring and Reaction BMS.3 scenario test 30
Table 15 Test Case TC_BMS.3.6 – Attack Detection, Reaction and Security Orchestration in BMS.3 scenario test 33
Table 16. Test case TC_BMS.4.1 – Connection with OdinS IoT-broker 36
Table 17. Test case TC_BMS.4.2 – Attack detection based on IoT monitoring data 38
Table 18. Test case TC_BMS.4.3 – Reaction to attack on sensor 40
Table 19. Test case TC_BMS.4.4 – Full BMS.4 scenario test 42
Table 20. Test case TC_MEC.3.1 – Device actuation 43
Table 21. Test case TC_MEC.3.2 – Monitoring processing 45
Table 22. Test case TC_MEC.3.3 – Reaction processing 46
Table 23. Test case TC_MEC.3.4 – Seal processing 47
Table 24. Test case TC_MEC.3.5 – Orchestration processing 48
Table 25. Test case TC_MEC.3.6 – Enforcement 49
Table 26. Test case TC_MEC.3.7– Full MEC.3 test case 50
Table 27. Generic test case questionnaire 52
Table 28. Questionnaire for TC_BMS.2.9 53

Table 29. Questionnaire for TC_BMS.3.5 53
Table 30. Questionnaire for TC_BMS.3.6 53
Table 31. Questionnaire for TC_BMS.4.4 53
Table 32. Questionnaire for TC_MEC.3.7 54

PUBLIC SUMMARY

Following, the deployment of the integrated ANASTACIA platform (outcome of in T6.1) for serving as the pilot environment, the ANASTACIA use cases needs to be tested. The scope of this deliverable is to define the deployment and operational aspects of the use cases in ANASTACIA project that were defined in D1.2 as part of work defined in T1.2. Out of them consortium decided to further test four use cases described in detail below in the document. They are identified to validate and test all the components and features introduced in ANASTACIA framework. This deliverable illustrates the implementation and testing of the considered use cases. Each use case scenario has been divided into test cases that will enable testing of ANASTACIA use cases in multiple steps. The document also defines the exact requirements per test case, the current status, and the expected achievement through the use of the ANASTACIA framework. Furthermore, an evaluation plan including the methodology, the definition of expected results, questionnaires, and key performance indicators will be formulated per test-case. The plan will be used for the framework evaluation and validation which will be conducted in the context of T6.3.

1 INTRODUCTION

1.1 AIMS OF THE DOCUMENT

The main goal of this document is to provide a clear guideline of the implementation of use cases and identify several test cases for each use case, where test cases are used to examine critical components and features in the ANASTACIA framework. In addition, each test case will be associated with expected outcomes that will be used in T6.3 for validation and evaluation. The aggregation of all the test case outcomes will perform the main objective of the use case. Each use case is structured around three main steps (i.e. attack, detection and mitigation), as explained in D6.1. Therefore, each test-case should be formed to examine one or more of these steps.

1.2 APPLICABLE AND REFERENCE DOCUMENTS

This document refers to following documents:

- D1.2 User-centered Requirement Initial Analysis.
- D6.1 Initial Technical integration and validation Report.

1.3 REVISION HISTORY

Version	Date	Author	Description
V0.1	10/01/2018	UTRC	Table of contents
V0.2	01/02/2018	ALL	Documents structure feedback included
V0.7	02/05/2018	ALL	First full document draft completed
V0.8	03/05/2018	OdinS, UTRC	Content contribution to use cases TC_BMS.2 and TC_BMS.4
V0.9	07/05/2018	MMT	Content contribution to use case TC_BMS.3
V1.0	09/05/2018	AALTO, THALES	Content contribution to use case TC_MEC.3
V1.1	11/05/2018	ALL	Initial full draft of deliverable
V1.3	17/05/2018	MMT, UTRC	Additional changes to use TC_BMS.3 and chapter 4
V1.4	18/05/2018	UTRC	Editing content for MEC.3, BMS.4 and expanding section 3.7
V1.7	24/05/2018	THALES	MEC.3 case content change
V1.8	04/06/2018	OdinS	BMS.2 case diagram updates

1.4 ACRONYMS AND DEFINITIONS

Acronym	Meaning
KPIs	Key Performance Indicators
TC	Test-Case
TE	Test event
UC	Use-Case
BMS	Building Management System
SDN	Software Defined Network
IoT	Internet of Things
SEP	Security Enforcement Plane
REST	Representational State Transfer
API	Application Programming Interface
Kafka	Message broker used in ANASTACIA framework to enable distributed communication between components

2 KPI IMPLEMENTATION IN ANASTACIA TEST CASES

2.1 METHODOLOGY

Figure 1 illustrates all available approaches in functional and non-functional testing used software. From practical perspective ANASTACIA test cases will be evaluated only on preselected methodologies that were marked as ANASTCIA use case test scope. From functional testing methodologies perspective we skipped acceptance testing as ANASTACIA is low TRL system and doesn't have business stakeholders that could help provide full business oriented insight. We will use part of this methodology to construct user questionnaire to get additional feedback from stakeholders. The questions are defined in section 3.7.

From non-functional testing ANASTACIA will not perform security testing of their components as they are on different stages of development and some project partners might not have resources to conduct full pen testing of their modules. Usability testing will look at each component from user perspective however in this scenario most of the ANSTACIA components are running on the background and only UI part of ANASTACIA can be evaluated. We believe that more testing in this space can be completed at the end of the project when all functional, integration and system testing will be already completed. From compatibility perspective at this stage ANASTACIA encourages usage of Dockerized components and Docker *[1]* containers technology will provide maximum coverage in this space.

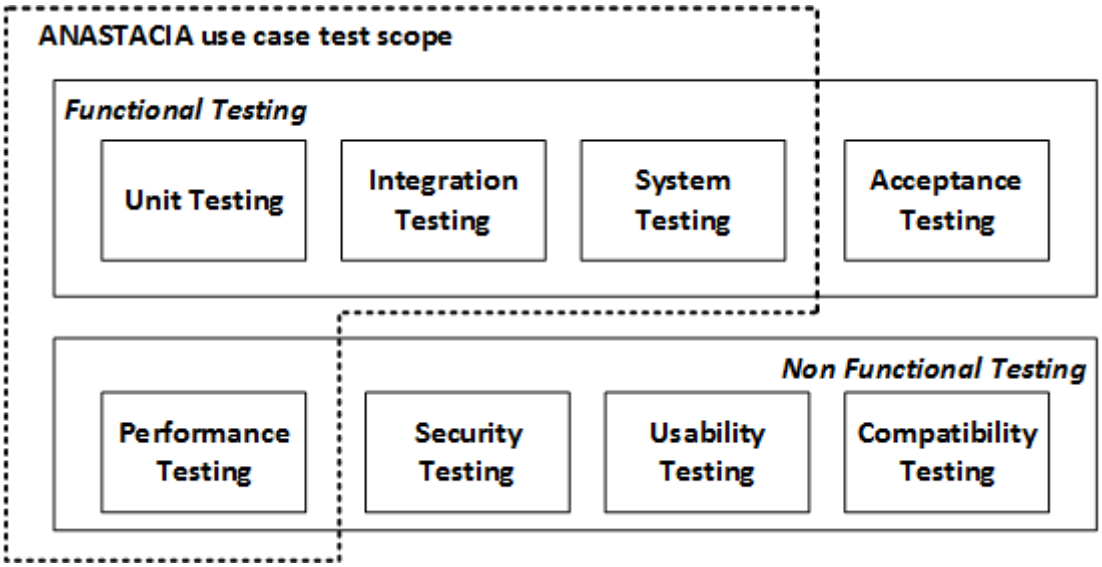

Figure 1. ANASTACIA testing methodologies scope.

From ANASTACIA use case test scope following testing methodologies were selected:

- Unit Testing – will be used to validate functionality at individual level. Each project partner will be required to provide as much cover for required functionalities in their components at their level. This methodology will utilize best practices used in every modern programming language. Initial test cases enclosed in this document belong to this category.
- Integration Testing – main scope of this document will be around helping defining test cases for integration between built components and help them integrate via APIs, interfaces and commons means of communication. Sub-sections on below describes KPIs sharing mechanism to help test cases integration.
- System Testing – at this level ANASTACI part of the system or whole system will be tested against specific four scenarios defined below. Each of the test cases contains one system test level case detailed with sequence diagram.
- Performance Testing – across all test cases a selection of KPIs will be calculated to check against test cases definition of acceptable level of KPIs established for each instance separately.

2.2 EVALUATION FRAMEWORK

ANASTACIA evaluation framework as discussed in D 6.1 is based on ISO/IEC 25010:2011 standard. The standard provides guideline to help formulate system objectives under tests. D6.1 predefined and formulated KPIs for each ANASTACIA framework component. This document focus will be on describing on how KPIs can be captured and processed from implementation perspective.

2.3 KEY PERFORMANCE INDICATORS

The KPI measurements will be determined depending on test case scenario. There are two types of measures that test cases framework will provide:

- **Internal** – measured locally by each component, increase where external inference cannot determine required KPI, or KPI externally depends on characteristics measured in other ANASTACIA component.
- **External** – computed on ANASTACIA framework level using events sent by the components i.e. processing time, reaction time by each of them separately to distributed bus such as Kafka. More information about this KPI processing can be found in next section of this document.

Specific component KPIs will be formulated in the chapter 3 when all details about test cases will be illustrated. Examples of internal and external components scripts are included in attachment to this document in section 0.

2.3.1 External KPIs processing

Figure 2 illustrates External KPI measurement architecture. On the diagram the test case script is the initiator that triggers test case execution. When test start-up criteria for given test case have been meet, an attack generation script is being executed. Next ANASTACIA components in test case C1-CN will monitor, react to the thread and will generate test events about their actions to distributed bus queue.

Chapter 0 provides message format example for test event that will be sent via distributed bus (Kafka broker). Test event message provides flexibility in what can be sent and is divided on 4 parts:

1. Test event source identification – component name, test case name and timestamp,
2. Triggers – List of triggers that switched component into action,
3. Actions – List of actions that component took in given test case,
4. KPIs – list internally measured KPIs delivered by the component.

In order to capture external KPIs in distributed system a distributed bus is required to capture all events coming from all ANASTACIA components under test. In ANASTACIA case generic test case KPI script is provided to help capture test events being sent by the components. Script acquires all events and out of them computes execution times and provides detailed time series on what happened during test case execution. The output of the script can be further used in D 6.3 work for test case evaluation and validation.

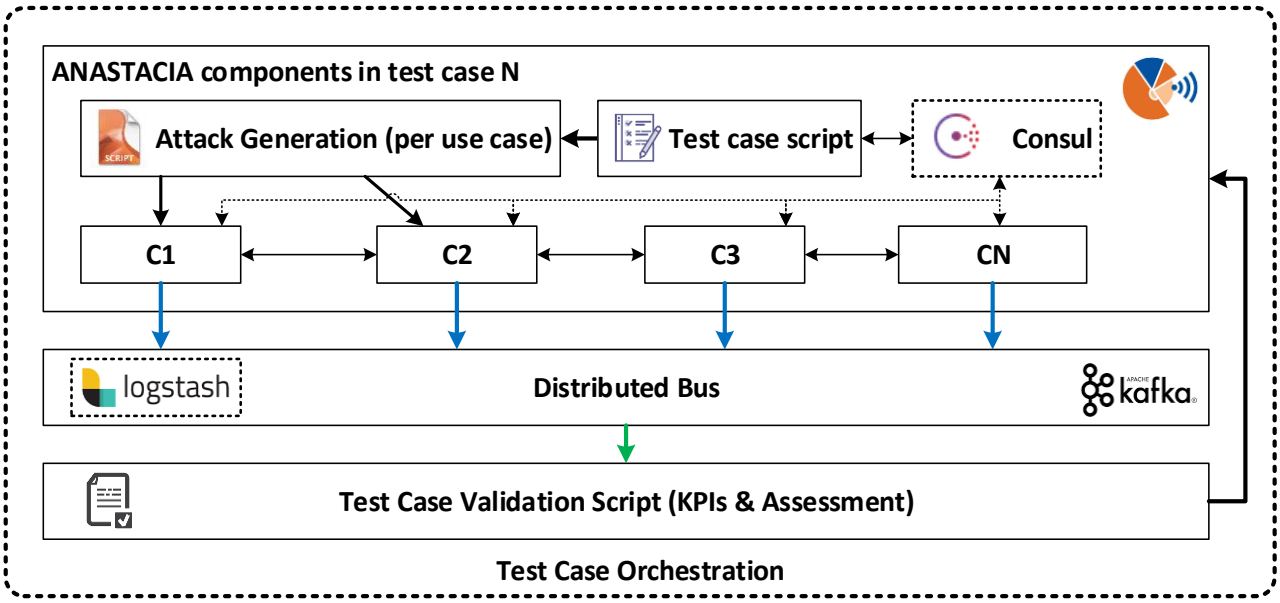

Figure 2. ANASTACIA test case KPI measurement architecture.

ANASTACIA team is considering using Consul Service *[2]* technology to help resolve distributed service availability check problem and enable faster component coordination during test cases execution. If further development and agreement among project partners will be achieved the test scripts can use it to pre-check and orchestrate test conditions in ANASTACIA system for running test cases.

Another considered technology for potential use in next iteration is distributed logging system called logstash *[3]*. It offers out of the box not only distributed logging capability but as well tailored UI that can help users to see test cases performance, KPIs during test cases execution.

Both technologies can be successfully used in conjunction with test framework to simplify testing and enhance test case results visibility to end users. Increased automation will also provide reproducibility which will be very useful due to component updates and changes that will have to be requalified. Removing manual test process is critical for successful ANASTACIA system demonstration, thus team effort in test process is focused around maximum reduction of manual input in test scenarios.

3 TEST USE-CASES

This section defines ANASTACIA test-cases corresponding to the each use-case. Considering each component has multiple functions that can be tested, and then the following section groups together the test-cases that belong to specific use-case, defined in deliverable D1.2. The aim of test-case is to go as granular as possible, so that each functionality can be properly validated. The test-cases are typically grouped in three main groups, as follows, where each test-case level evaluated against identified KPI.

a) **Individual component performance:** test-cases are designed to evaluate individual components preference, where these components are used to perform the corresponding use-case.
b) **Components interaction:** in order to examine two connected components, a set of test-cases are identified to evaluate the performance of the connected components and their connection reliability.
c) **Use-case integration:** finally, an integration test-case(s) is identified to evaluate the full use-case performance.

3.1 TEST CASE NAMING CONVENTION

Test cases naming convention reflect use case naming convention. The test case convention is as follows:

TC_<*ANASTACIA use case identifier*>.<tc_number>

Where:

- TC_ – abbreviation for test case,
- <name> – ANASTACIA use case identifier,
- .<tc_number> – local identifier for test-case

3.2 TEST CASE IDENTIFICATION PROCESS

This chapter describes the test case identification process to develop test case scenarios that will be used to test the ANASTACIA components. The test case identification process has been divided into three main steps:

1. **Input and dependencies**: identifying inputs and their dependency for each component used in performing the test case.
2. **Internal component features:** this step illustrates component feature(s) that will be tested by using a given test case. Normally, each test case will focus on a particular feature or group of features that are atomic from the component's perspective.
3. **Output requirements:** finally each test case scenario will describe how to evaluate results, expected outcome of the test case and KPIs that are required by the component to meet ANASTACIA framework requirements (i.e. processing speed, supported I/O formats, response time, etc.).

The test case identification methodology steps are illustrated on Figure 3. From the test-case perspective, it is important to focus on component functionality that will be tested as part of predefined use-case as reported in D1.2. Since test-cases will be executed multiple times to validate and verify the component's behavior and ensure current implementation stability, therefore we plan to use *Kafka* message broker to facilitate distributed outcome tracking.

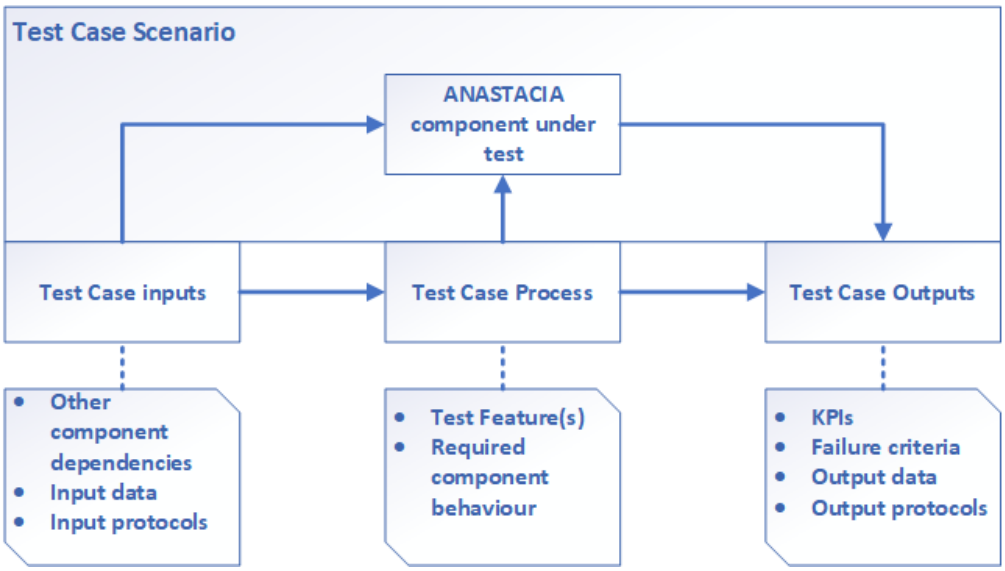

Figure 3. ANASTACIA test case identification methodology steps.

3.3 Test-Cases TC_BMS.2: Insider Attack on the Fire Suppression System

This test-case focuses on testing use-case *UC_BMS.2: Insider attack to a fire suppression system*. The main objective of this use-case is evaluate ANASTCIA framework against protecting the system from an insider attack and avoid any damage to the building assets. In this use-case, the attacker exploits the building operations workstation to request the activation of fire alert system managed by IoT device. Hence, this section identifies a set of test-cases (TC_BMS.2) to implement and evaluate UC_BMS.2 on Smart Building testbed.

3.3.1 Individual Component Test-Cases

This section presents a test-case to evaluate the performance of IoT devices for the validation of the authorization with capability token using the security enabler of AAA architecture.

The following table shows the description of the test case to evaluate and validate the capability token before authorizing any action over IoT devices using AAA architecture of ANASTACIA framework in the context of use case BMS.2.

Table 1. Test case TC_BMS.2.1 – Validating a capability token for authorization of device actuation

TC_BMS.2.1	Validating a capability token for authorization of device actuation
Preconditions	• IoT device is working - COAP API is available online • IoT device receives an actuation query with a capability token.
Components	• IoT Device
Execution	• Validate the capability token to authorize the actuation. • Determine if the authorization decision is accept or deny.
Expected results	• IoT Device generates an authorization decision (accept or deny).

Expected completion	Month 18 – June of 2018
KPI(s)	• Obtain the elapsed time for the validation of capability token
Fail criteria	• Exception caught during the token validation • Excessive time for token validation.

3.3.2 Components Interaction Test-Cases

This subsection presents a set of test-cases to evaluate the interaction among components of ANASTACIA framework grouped in different modules such as IoT Infrastructure, Monitoring Module, Reaction Module, Seal Manager, Orchestration Plane and Control Domain. Concretely, this subsection describes the following test-cases involved in use case BMS.2:

- TC_BMS.2.2 Network Authentication and Device Registration
- TC_BMS.2.3 Unauthorized Actuation Detection
- TC_BMS.2.4 Monitoring Management
- TC_BMS.2.5 Seal Management
- TC_BMS.2.6 Reaction Management
- TC_BMS.2.7 Orchestration Management
- TC_BMS.2.8 Enforcement Management

3.3.2.1 TC_BMS.2.2 Network Authentication and Device Registration

This test case implements the evaluation of the operations required when a new device is connected in an IoT network using ANASTACIA framework. For this test-case, the next figure shows the sequence diagram with the messages exchanged by IoT Device, PANA Agent, Orchestrator, IoT Controller, SDN Controller and SDN Switch.

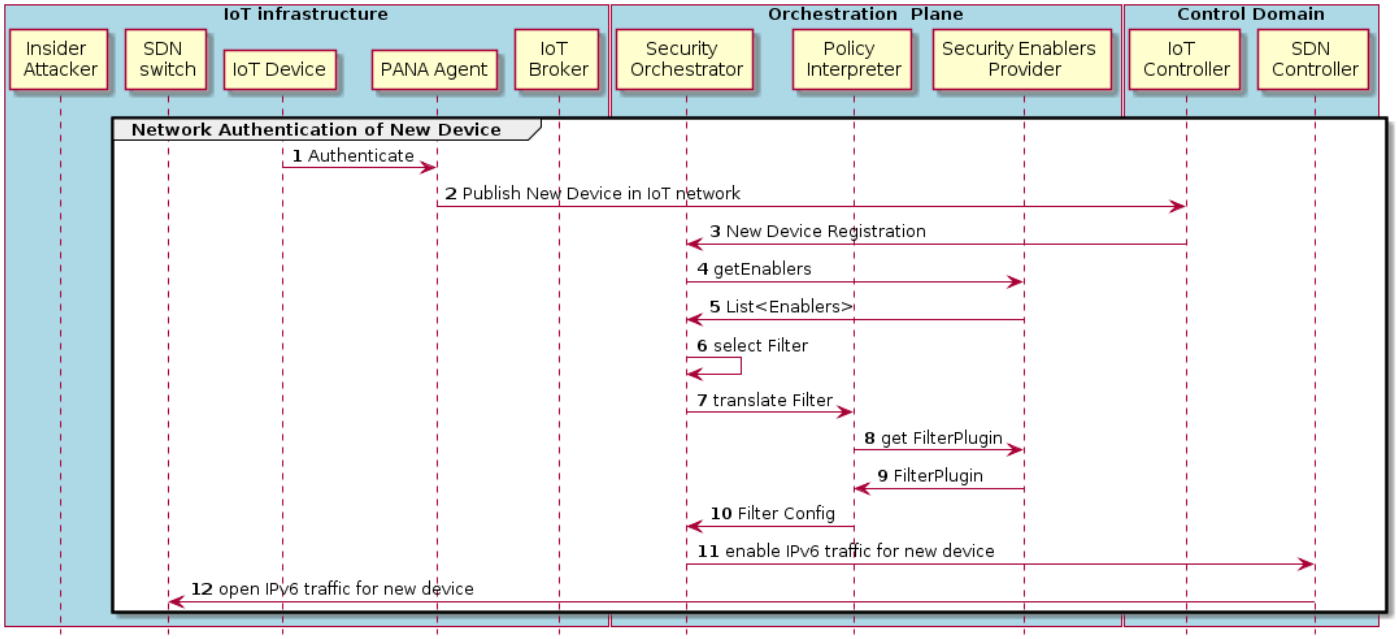

Figure 4 Sequence diagram for TC_BMS.2.2 – network Authentication and device registration

The following table shows the description of the test case to validate the ANASTACIA framework when a new device is connected in an IoT network.

Table 2. Test case TC_BMS.2.2 – Network Authentication and Device Registration

TC_BMS.2.2	Network Authentication and Device Registration
Preconditions	• PANA Agent is ready to authenticate new devices in IoT network. • Orchestrator is ready to receive data registration from PANA Agent. • IoT Controller is ready to receive publication of new devices in IoT network. • SDN Controller is ready to enforce actions over SDN switches. • SDN Switch (OVS) is ready to forward IP traffic in IoT network.
Components	• IoT Device • PANA Agent • Orchestrator • IoT Controller • SDN Controller • SDN Switch (OVS)
Execution	• Turn on the IoT device in order to start the bootstrapping process • IoT device sends a query for network access to PANA Agent • PANA Agent evaluates if the identity of new device is valid or invalid • If it is valid, PANA Agent sends a publication of new device to IoT Controller • IoT Controller sends a message to Orchestrator for new device registration including relevant data (IPv6 of device, IPv6 of IoT controller and SDN switch for new device) • Orchestrator requests to SDN controller to enable the IPv6 traffic for new device • SDN controller enforces the open of IPv6 traffic in SDN Switch (OVS)
Expected results	• Network authentication is successful • New device publication is successful • Device registration is successful • Traffic enabling is successful
Expected completion	Month 19 – July of 2018
KPI(s)	• Obtain the response time for network authentication between IoT device and PANA Agent • Obtain the response time for new device publication between PANA Agent and IoT Controller • Obtain the response time for device registration between IoT Controller and Orchestrator • Obtain the response time for traffic enabling between Orchestrator and SDN controller
Fail criteria	• Excessive time for network authentication. • Excessive time for new device publication. • Excessive time for device registration. • Excessive time for traffic enabling.

3.3.2.2 TC_BMS.2.3 Unauthorized Actuation Detection

This test case implements the evaluation of the operations required when an insider attacker is detected due to unauthorized actuation over an IoT device using the security enabler of AAA architecture provided by ANASTACIA framework. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Insider Attacker, IoT Device, IoT Broker and Kafka Broker.

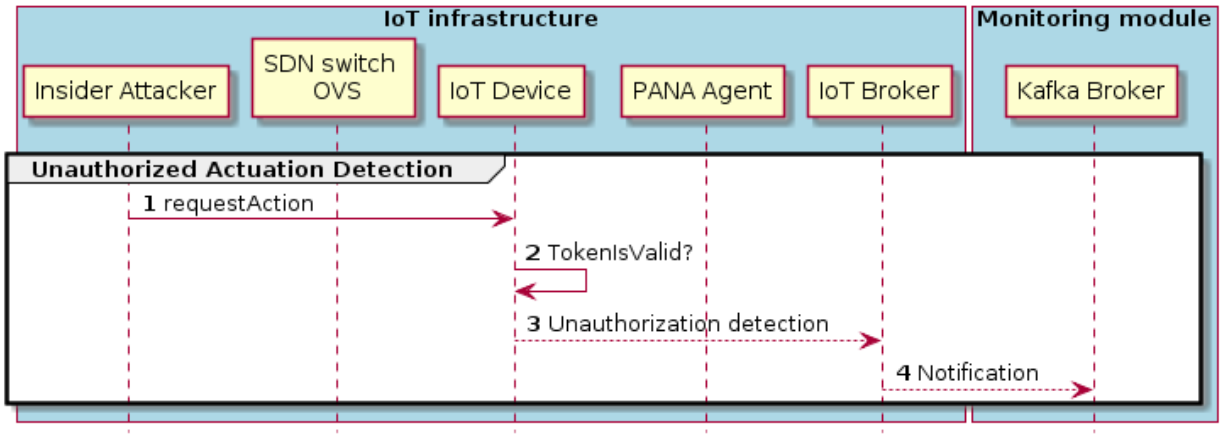

Figure 5 Sequence diagram for TC_BMS.2.3 – unauthorized actuation detection

The following table shows the description of the test case to validate the IoT infrastructure and AAA architecture when an unauthorized actuation is detected and notified by an IoT device.

Table 3. Test case TC_BMS.2.3 – Unauthorized Actuation Detection

TC_BMS.2.3	Unauthorized Actuation Detection
Preconditions	• Script is ready to emulate the attack query. • IoT Device is authenticated in the IoT network. • IoT broker is ready to receive data messages from IoT devices via COAP API. • Kafka broker is ready to receive monitoring data via REST API.
Components	• Script of Insider Attacker • IoT Device • IoT Broker • Kafka Broker
Execution	• Run the script emulating the attack for the activation of fire alarm managed by IoT device. • IoT device receives the activation query and validate the capability token to authorize the remote actuation. • If the token is invalid, IoT device sends a message of unauthorized actuation detection to IoT Broker. • IoT Broker notifies the threat detected to Kafka broker.
Expected results	• Unauthorized actuation detection is successful • Threat notification is successful
Expected completion	Month 19 – July of 2018
KPI(s)	• Obtain the response time for unauthorized actuation detection from IoT device to IoT broker

	• Obtain the response time for threat notification from IoT broker to Kafka broker
Fail criteria	• Excessive time for unauthorized actuation detection. • Excessive time for threat notification.

3.3.2.3 TC_BMS.2.4 Monitoring Management

This test case implements the evaluation of the monitoring management when an attack notification is generated due to unauthorized actuation detection. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Kafka Broker, Incident Detector and Verdict and Decision Support System (VDSS).

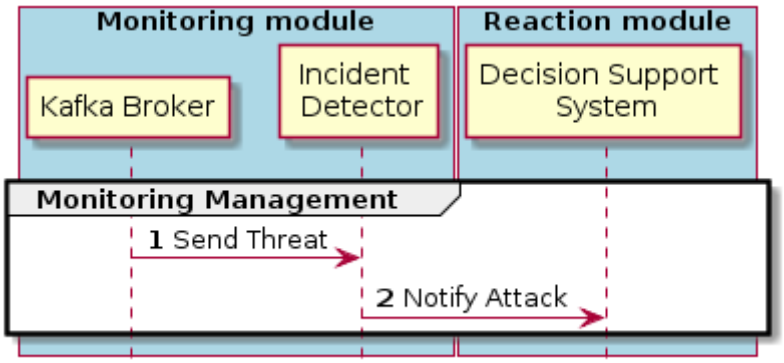

Figure 6 Sequence diagram for TC_BMS.2.4 – monitoring management

The following table shows the description of the test case to validate the monitoring module when an attack notification of unauthorized actuation is generated by the Incident Detector.

Table 4. Test case TC_BMS.2.4 – Monitoring Management

TC_BMS.2.4	Monitoring Management
Preconditions	• IoT Device has detected an unauthorized actuation and IoT broker has notified the threat towards Kafka broker. • Kafka broker is ready to communicate via REST API and MQTT. • Incident detector is ready to receive threats notifications. • Verdict and Decision Support System is ready to process attack notifications.
Components	• Kafka Broker • Incident Detector • Verdict and Decision Support System (VDSS)
Execution	• Kafka Broker sends the treat notification towards the Incident Detector. • If the threat is considered as an attack, Incident Detector sends a message of attack detection to VDSS.
Expected results	• Attack detection is successful
Expected completion	Month 19 – July of 2018

KPI(s)	• Obtain the response time for attack detection from Kafka broker to VDSS
Fail criteria	• Excessive time for attack detection.

3.3.2.4 TC_BMS.2.5 Seal Management

This test case implements the evaluation of the seal management when an alert of unauthorized actuation must be shown to end-users. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Verdict and Decision Support System (VDSS), DSPS Agent, DSPS User-Interface.

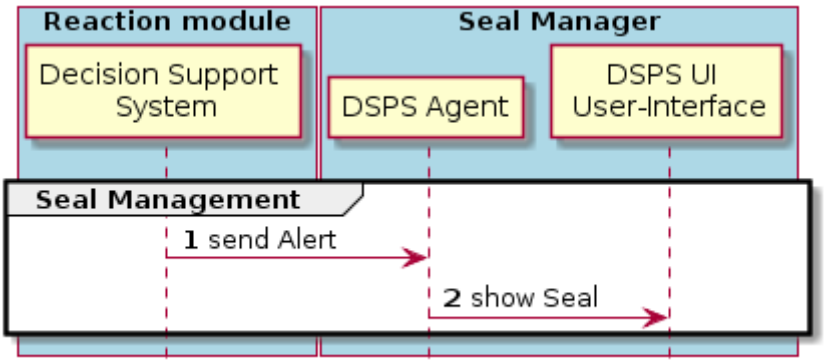

Figure 7 Sequence diagram for TC_BMS.2.5 – seal management

The following table shows the description of the test case to validate the seal manager when an alert message of unauthorized actuation is showed by the DSPS User-Interface.

Table 5. Test case TC_BMS.2.5 – Seal Management

TC_BMS.2.5	**Seal Management**
Preconditions	• VDSS has received a notification for attack detection of remote actuation over IoT Device connected to fire alarm system. • DSPS Agent is ready to receive alert notifications. • DSPS User-Interface (UI) is ready to show the seal state and alert notification.
Components	• Verdict and Decision Support System (VDSS) • DSPS Agent • DSPS User-Interface (UI)
Execution	• VDSS notifies the alert towards DSPS Agent for seal management. • According to the alert detected, DSPS Agent generates the seal to show in DSPS User-Interface.
Expected results	• Seal management is successful
Expected completion	Month 19 – July of 2018
KPI(s)	• Obtain the response time for seal management from VDSS to DSPS User-Interface.

Fail criteria	• Excessive time for seal management.

3.3.2.5 TC_BMS.2.6 Reaction Management

This test case implements the evaluation of the reaction management when mitigation actions must be proposed to the Orchestrator in order to cope with the unauthorized actuation. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Verdict and Decision Support System (VDSS), Mitigation Action Service (MAS) and Orchestrator.

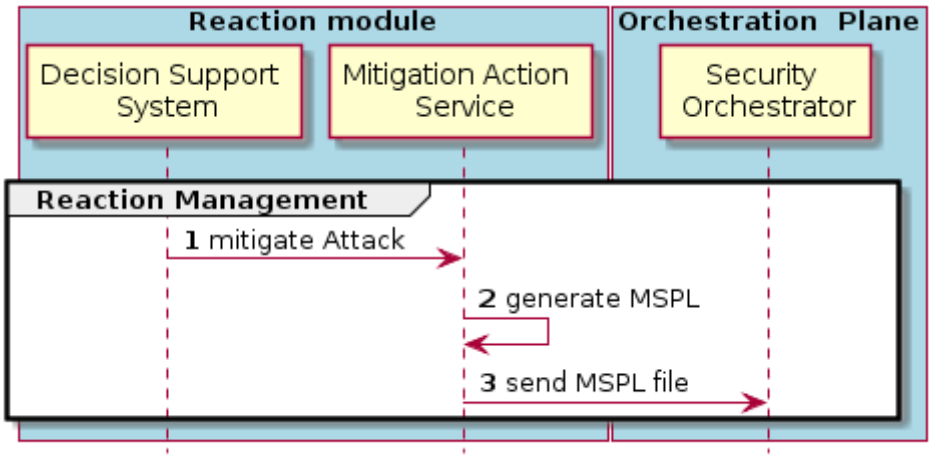

Figure 8 Sequence diagram for TC_BMS.2.6 – reaction management

The following table shows the description of the test case to validate the reaction management when the mitigation actions are proposed by Mitigation Action Service (MAS).

Table 6. Test case TC_BMS.2.6 – Reaction Management

TC_BMS.2.6	Reaction Management
Preconditions	• VDSS has received a notification for attack detection of remote actuation over IoT Device connected to fire alarm system. • Mitigation Action Service is ready to manage the reaction according to the attack. • Orchestrator is ready to receive the reaction
Components	• Verdict and Decision Support System (VDSS) • Mitigation Action Service (MAS) • Orchestrator
Execution	• VDSS notifies the alert towards Mitigation Action Service (MAS) • According to the attack detected, MAS generates a MSPL file with the reaction proposed. • MAS sends the MSPL file towards Orchestrator
Expected results	• Reaction management is successful
Expected completion	Month 19 – July of 2018

KPI(s)	• Obtain the response time for reaction management from VDSS to Orchestrator.
Fail criteria	• Excessive time for reaction management.

3.3.2.6 TC_BMS.2.7 Orchestration Management

This test case implements the evaluation of the orchestration management when concrete configurations of the mitigation actions are generated. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Orchestrator and Policy Interpreter.

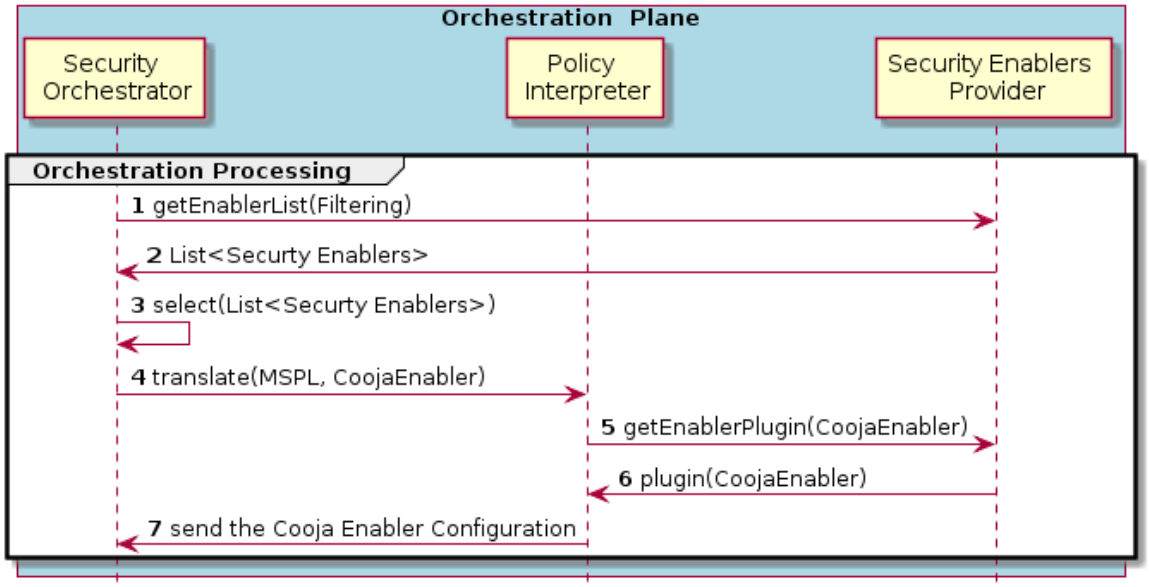

Figure 9 Sequence diagram for TC_BMS.2.7 – orchestration management

The following table shows the description of the test case to validate the orchestration management when security enabler configurations for the mitigation actions are provided by Policy Interpreter.

Table 7. Test case TC_BMS.2.7 – Orchestration Management

TC_BMS.2.7	Orchestration Management
Preconditions	• Orchestrator has received the reaction proposal to mitigate the attack of remote actuation over IoT Device connected to fire alarm system. • Policy Interpreter is ready to translate MSPL file to concrete configuration to enforce the mitigation actions. • Security Enablers Provider is ready to provide the configuration of security enablers.
Components	• Orchestrator • Policy Interpreter • Security Enablers Provider
Execution	• Orchestrator requests the translation of MSPL file to Policy Interpreter • According to the security enablers provider, Policy Interpreter responses a concrete configuration to enable a Honeynet of virtual IoT devices using Cooja emulator.

Expected results	• Orchestration management is successful
Expected completion	Month 19 – July of 2018
KPI(s)	• Obtain the response time for orchestration management between Orchestrator, Policy Interpreter and Security Enablers Provider.
Fail criteria	• Excessive time for orchestration management.

3.3.2.7 TC_BMS.2.8 Enforcement Management

This test case implements the evaluation of the enforcement management when concrete mitigation actions are carried out. For this test-case, the next figure shows the sequence diagram with the messages exchanged by Orchestrator, VNF Controller (OSM), SDN Controller and SDN Switch (OVS).

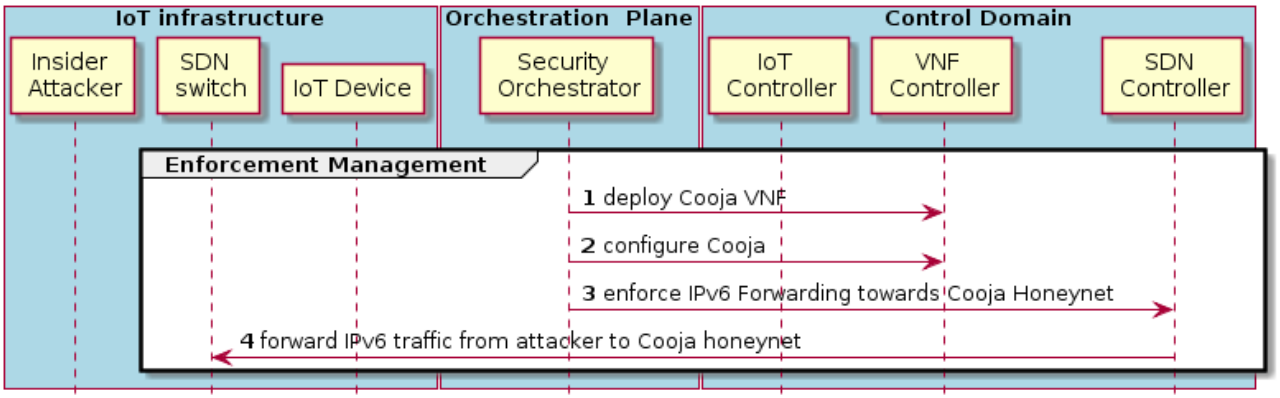

Figure 10 Sequence diagram for TC_BMS.2.8 – enforcement management

The following table shows the description of the test case to validate the enforcement management when security enablers such as VNF deployment and SDN forwarding are applied to mitigate the unauthorized actuation over an IoT device.

Table 8. Test case TC_BMS.2.8 – Enforcement Management

TC_BMS.2.8	Enforcement Management
Preconditions	• Orchestrator has received the configuration to enforce the mitigation actions of a Honeynet with virtual IoT devices using Cooja emulator. • VNF Controller (OSM) is ready to deploy virtual machines with network functions. • SDN Controller is ready to apply filtering/forwarding rules • SDN Switch (OVS) is ready to forward IP traffic in IoT network.
Components	• Orchestrator • VNF Controller (OSM) • SDN Controller • SDN Switch (OVS)
Execution	• Orchestrator requests to VNF Controller for the deployment of a Honeynet with virtual IoT devices using Cooja emulator.

	• Orchestrator requests to SDN Controller for the traffic forwarding from the insider attacker towards the virtual Honeynet in VNF deployment.
Expected results	• VNF Deployment is successful • Traffic Forwarding is successful
Expected completion	Month 19 – July of 2018
KPI(s)	• Obtain the response time for VNF deployment between Orchestrator and VNF Controller. • Obtain the response time for traffic forwarding between Orchestrator and SDN Controller.
Fail criteria	• Excessive time for VNF deployment. • Excessive time for traffic forwarding.

3.3.3 Integration Test-Case

This subsection shows the test-case required to evaluate the full integration of all components of ANASTACIA framework involved in the use case BMS.2 of "*Insider Attack on the Fire Suppression System*". This test-case implements the integration of the test-cases abovementioned from TC_BMS.2.2 to TC_BMS.2.8.

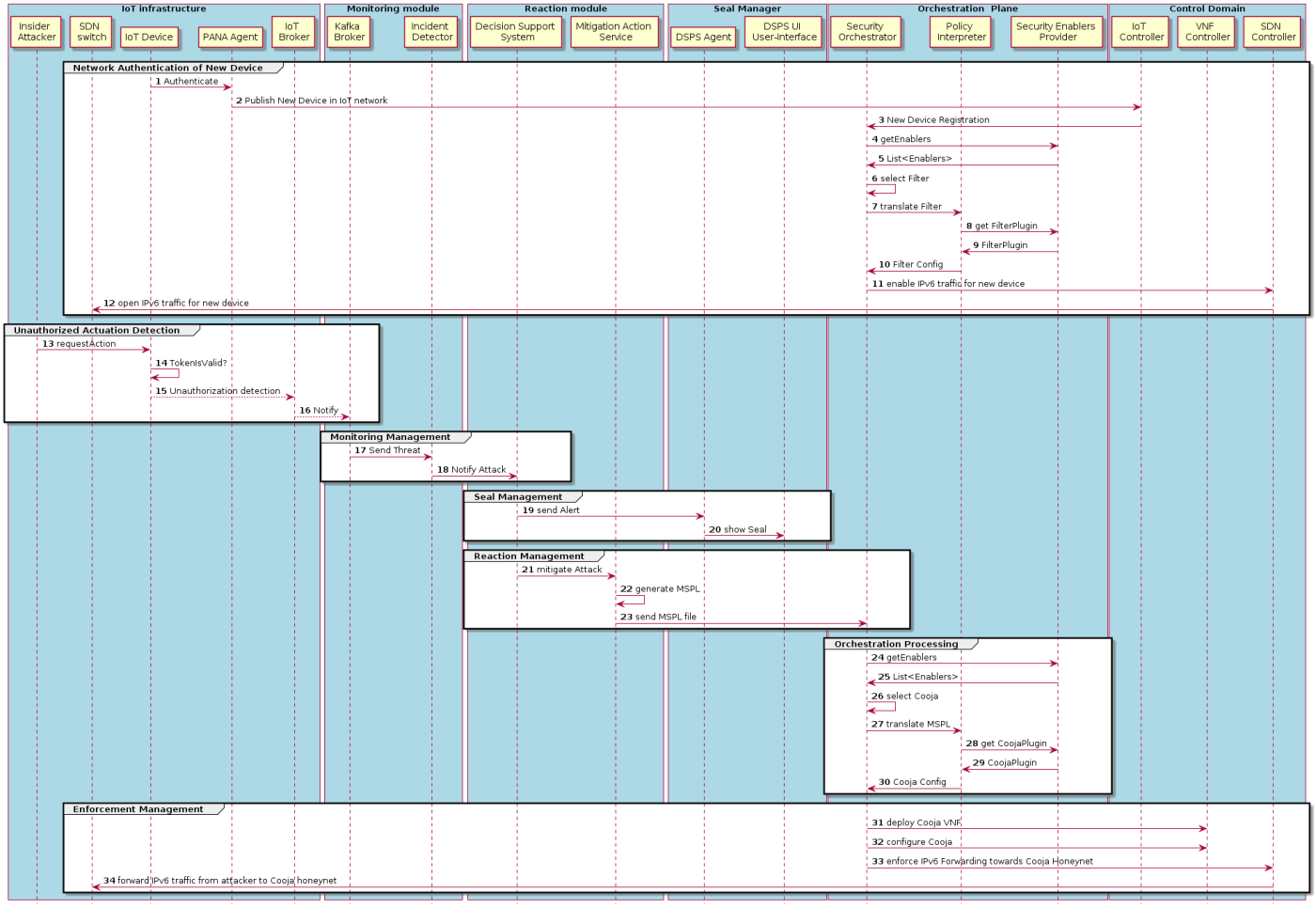

Figure 11 Sequence diagram for TC_BMS.2.9 – full integration of ANASTACIA framework for use case BMS.2

For this test-case, the following figure shows the sequence diagram with the messages exchanged by the components of ANASTACIA framework. The diagram shows the components grouped in the different

modules defined in ANASTACIA architecture such as IoT Infrastructure, Monitoring Module, Reaction Module, Seal Manager, Orchestration Plane and Control Domain.

The following table shows the description of the test case to evaluate and validate the full integration of ANASTACIA components in use case BMS.2.

Table 9. Test case TC_BMS.2.9 – Full integration of ANASTACIA framework for use case BMS.2

TC_BMS.2.9	Full integration of ANASTACIA framework for use case BMS.2
Preconditions	• Preconditions from TC_BMS.2.2, TC_BMS.2.3, TC_BMS.2.4, TC_BMS.2.5, TC_BMS.2.6, TC_BMS.2.7 and TC_BMS.2.8
Components	• Script for Insider Attack • SDN Switch • IoT Device • PANA Agent • IoT Broker • Kafka Broker • Incident Detector • Decision Support System • Mitigation Action Service • DSPS Agent • DSPS User-Interface UI • Security Orchestrator • Policy Interpreter • Security Enablers Provider • IoT Controller • VNF Controller • SDN Controller
Execution	• To follow all steps ordered from TC_BMS.2.2 to TC_BMS.2.8
Expected results	• The completed integration is successful.
Expected completion	Month 20 – August of 2018
KPI(s)	• Obtain the total time from unauthorized actuation detection in IoT device to the enforcement of SDN Controller for the traffic forwarding of the insider attacker towards the virtual Honeynet in VNF deployment.
Fail criteria	• Excessive total time for the detection and mitigation of the unauthorized actuation for insider attacker.

3.4 Test-Case TC_BMS.3: Remote Attack on the Building Energy Microgrid

In this test-case, we examine the UC_BMS.3: *Remote Attack on the Building Energy Microgrid*. The use-case aims to evaluate ANASTACIA platform towards protecting an energy microgrid against an ex-employed

remote intrusion, where an ex-employee plans to remotely attack the plant by accessing the SCADA server and violating the stored data. Hence, ANASTACIA platform should be tested within this section to provide protection for all the sensitive data.

In this particular scenario, the ANASTACIA technologies will be used to detect an SQL Injection (SQLi) attack towards the SCADA server. In this particular scenario, the Montimage Monitoring Tool (MMT) will be used to detect the attack raising the corresponding alerts and triggering the Reaction mechanism to cope with the ongoing attack. More information about this test case can be found in D1.2 and D2.2, which present a description of the use cases requirements and a detailed description of the detection and mitigation techniques.

The general strategy of the Use Case testing will follow an incremental way, aiming to test the integration capabilities of the different modules brought by the ANASTACIA partners. According with the ANASTACIA objectives, it is important to remark that the test cases here described do not aim to bring unitary tests for each integrated module, but rather test the interaction among different modules. In Figure 12 it is shown the main organization of the testing process of the use case, where 4 incremental steps can be identified:

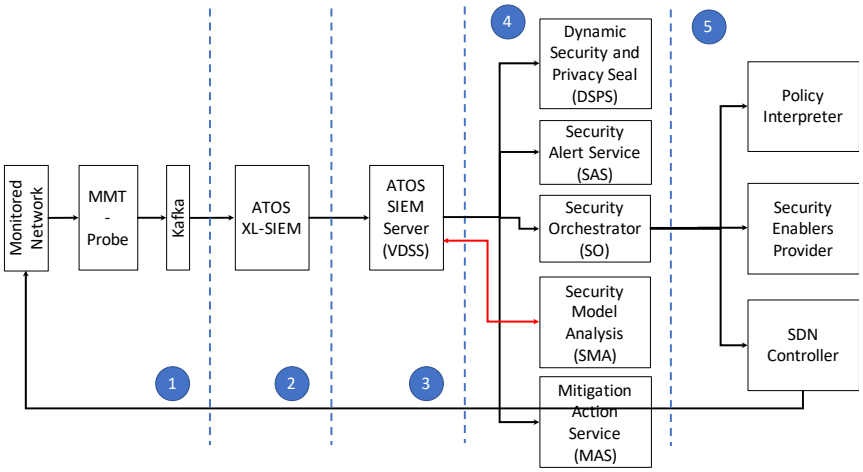

Figure 12 BMS.3 Test Case General Workflow

1. *Basic data extraction (step 1)*: This step aims to test the extraction feature of ANASTACIA through the MMT-Probe component. Specifically, this stage will test the capability of MMT to extract statistics from the IoT ANASTACIA network.
2. *Monitoring test (Steps 1 and 2)*: Once the data is extracted and a first security analysis is performed by the MMT-Security library (embedded in MMT-Probe). To this end, a SQLi attack will be injected in the monitored network. The detection data is sent to the ATOS SIEM Server to be further analyzed. This step aims to test this Monitoring Feature of ANASTACIA.
3. *Reaction Test (Steps 1 through 3)*: This step will test the capability of the reaction module to propose the countermeasures to the Security Orchestrator. In this test case, a SQLi attack will be generated in the analyzed network which will be detected by the MMT-Probe. This module will report the alert to the ATOS XL-SIEM for further analysis, which will send the final detection verdicts to the XL Server for risk assessment.
4. *Monitoring and Reaction Test (Steps 1 through 4)*: This step will test the monitoring and reactions components in the BMS.3 scenario.
5. *Legit SQL Traffic Test*: This scenario will test the behavior of the platform when a legit SQL traffic is injected in the network.
6. *Security Orchestration Test (Steps 1 through 5)*: This final configuration will test the interactions between the Monitoring and Reaction and the Security Orchestration Planes. This includes testing the implementation of the decided countermeasures on the monitored network.

3.4.1 Individual Component Test-Cases

3.4.1.1 Test-Case TC_BMS.3.1

The Test Case TC_BMS.3.1 comprises the first milestone of the general BMS.3 ANASTACIA use case. It is intended to provide an initial proof of concept, showing that the proposed tool (MMT-Probe in this case) can be used to monitor a SCADA server against SQLi attacks. In particular, this test case aims to obtain the periodic network statistics reports generated by the DPI Module (MMT-Probe for this use case) about the connections present in monitored network without an ongoing attack.

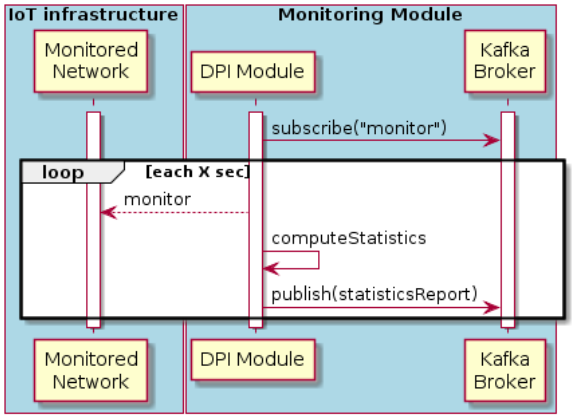

Figure 13 Sequence diagram for TC_BMS.3.1 – MMT-Probe basic network analysis

Figure 13 shows the interactions of the different components when the DPI Module (MMT-Probe in this use case) is monitoring the network. In the diagram, the MMT-Probe subscribes to the "monitor" channel on the Kafka Broker. Right after this action, the component actively monitors the Monitored Network, computing the statistics of all the open connections that go through the network. The collected information is aggregated and then published to the Kafka channel in form of a statistics report.

Table 10. Test case TC_BMS.3.1 – MMT-Probe basic network analysis

TC_BMS.3.1	MMT-Probe basic network analysis
Preconditions	• The Monitored network is working - REST API is available online • Kafka message broker is configured and up and running • The DPI Module (MMT-Probe) is configured to sniff the traffic of the Monitored Network (either in the IoT network or in the incoming link of the attacked server) • MMT-Probe is sending network reports to the Kafka Channel in JSON format
Components	• The Monitored Network • Kafka message broker • MMT-Probe (as DPI Module)
Execution	• Start all the components from above in that order • Observe JSON messages coming from MMT-Probe on Kafka message broker
Expected results	• MMT-Probe statistics reports are being published in the Kafka broker
Expected completion	Month 18 – June of 2018
KPI(s)	• MMT-Probe instance is stable • All the information produced by the MMT-Probe (statistics reports) is published on the Kafka Channel in accordance to JSON message specification

Fail criteria	• Exception during the active monitoring of MMT-Probe • Statistics reports lost during the active monitoring of MMT-Probe

3.4.1.2 Test-Case TC_BMS.3.2

Once the monitoring capabilities of MMT have been tested in the ANASTACIA Network, the next step is testing the ability of MMT-Probe to send the security reports on the common ANASTACIA Kafka Channel.

In this particular test case, the MMT-Probe will be deployed to allow monitoring the network to which the SCADA server is connected (in other words, the Monitored Network). This configuration will allow the MMT-Probe to detect any malicious activity towards the server, in particular, the SQLi attack concerned by the use case.

To achieve this goal, this test case expands TC_BMS.3.1 with the introduction of an attack in the Monitored Network, which will be detected and reported by the MMT-Probe module deployed in the ANASTACIA infrastructure.

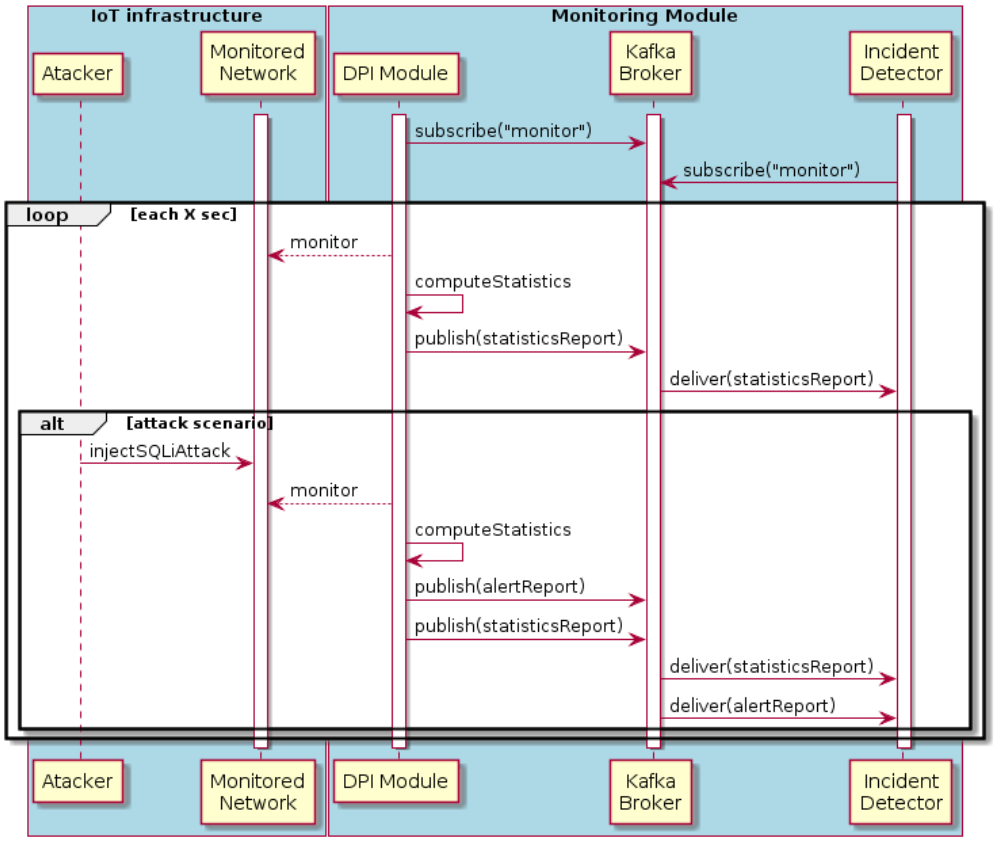

Figure 14 Sequence diagram for TC_BMS3.2 – Monitoring Module detection

Figure 14 shows the sequence diagram of the TC_BMS.3.2 test case. At the start of the test case, both the Kafka Broker and the Incident Detector (the ATOS XL-SIEM tool in this case) have to subscribe to the "monitor" channel in order to listen to the statistics and alert reports generated by the MMT-Probe module. At this stage, the DPI Module (MMT-Probe) will actively monitor the network, generating statistics about the packets traversing the network. At certain point, the SQLi attack is triggered by executing the attack script, which will inject network packets containing the stack. These packets will be analyzed by the MMT-Probe and detect the ongoing attack. After this, the monitoring instance will generate and publish in the Kafka channel an alert report in addition to the statistics report. The former will contain the details of the detected attack, including the affected IP addresses and the type of attack detected.

Table 11. Test case TC_BMS.3.2 – Monitoring Module detection

TC_BMS.3.2	Monitoring Module detection
Preconditions	• TC_BMS.3.1 completed and verified • Attack script that will emulate malicious behavior and is capable of injecting packets on the Monitored Network containing SQLi attacks
Components	• The Monitored Network • Kafka message broker • MMT-Probe (as DPI Module) • XL-SIEM Agent (as Incident Detector)
Execution	• Start all the components from above in that order • Observe JSON messages (statistics reports) coming from MMT-Probe on Kafka message broker • Start the attack by executing the generation script • Observe JSON messages (alert reports) coming from MMT-Probe on Kafka message broker
Expected results	• MMT-Probe statistics reports are being published in the Kafka broker • MMT-Security alert reports are published in the Kafka broker when the attack is detected in the network
Expected completion	Month 18 – June of 2018
KPI(s)	• All KPIs stated in TC_BMS.3.1 • Attack is detected within 5 sec
Fail criteria	• Exception during the active monitoring of MMT-Probe • Statistics reports lost during the active monitoring of MMT-Probe • Alert Reports lost during the active monitoring of MMT-Probe while a SQLi attack is conducted.

3.4.2 Components Interaction Test-Cases

3.4.2.1 Test-Case TC_BMS.3.3

The TC_BMS.3.3 test case aims to transmit the computed alerts to the Reaction Module of the ANASTACIA platform. In this scenario, the MMT-Probe will detect a SQLi attack, and transfer this information to the Reaction module to compute the pertinent countermeasures.

Following the incremental testing strategy, the TC_BMS.3.2 is expanded with the ATOS XL-SIEM capabilities to aggregate the information from different sources (such as different security modules deployed in the Monitored Network), process that is used to generate the verdicts that will be sent to the ANASTACIA's Reaction Module. Then this module which is in charge of starting the process of determining the countermeasures that will be deployed in order to cope with the injected attack.

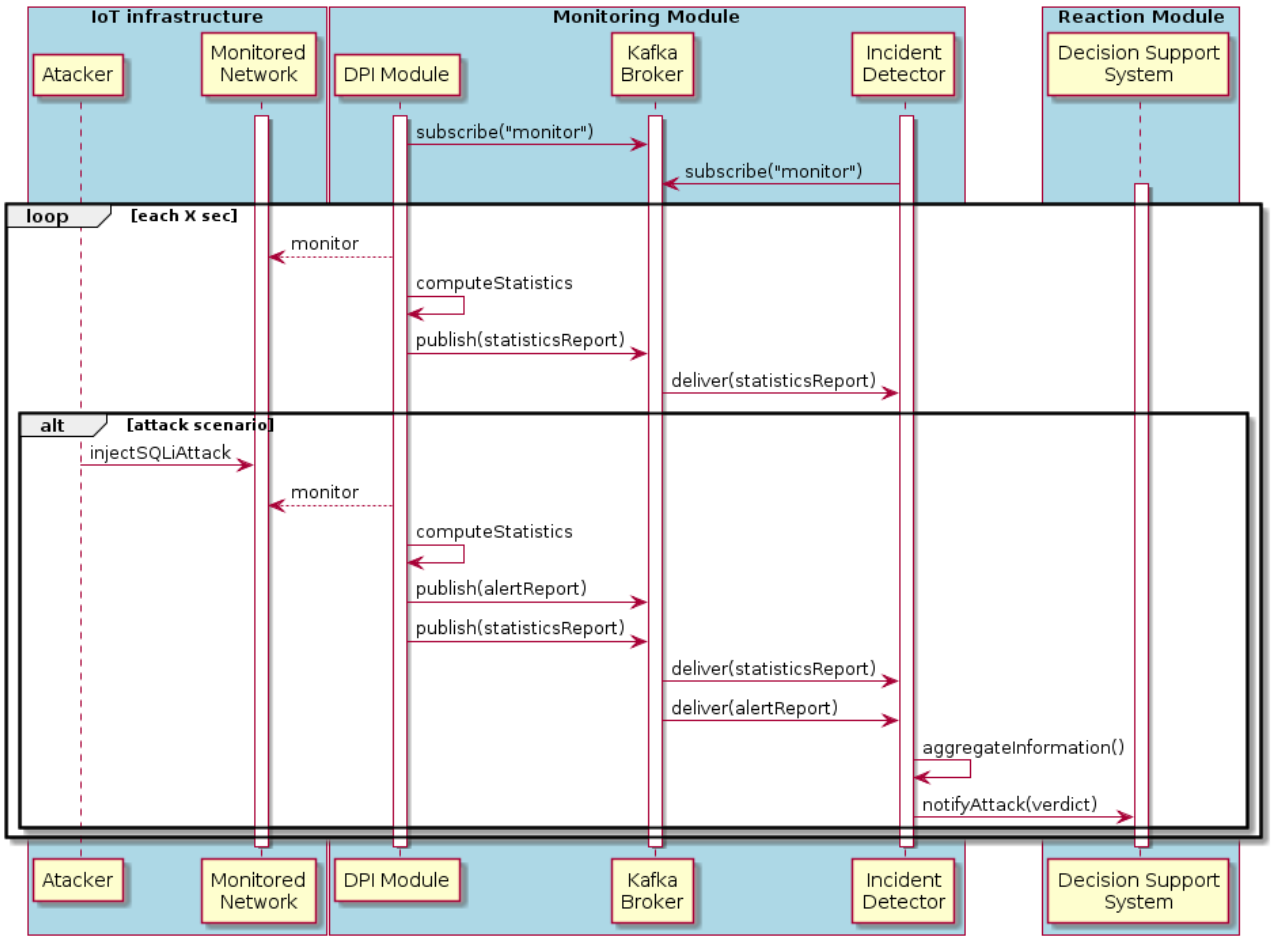

Figure 15 Sequence diagram for TC_BMS.3.3 – Reaction to SQLi attack on SCADA network

Figure 15 shows the sequence diagram for this particular use case. In the first stage, the DPI Module (MMT-Probe) and the Incident Detector (ATOS XL-SIEM) modules have to subscribe to the "monitor" channel of the Kafka Broker in order to be able to publish and receive messages respectively. Next, the MMT-Probe starts monitoring the network, generating statistics reports periodically about the flows on the monitored network. Once the attack has been triggered, the MMT-Probe module detects the malicious activity and generates the corresponding alert report. This report is published in the Kafka channel, and sent to the XL-SIEM tool for its further analysis. Later, the XL-SIEM tool processes this report and generates a verdict about the SQLi attack. This verdict is sent to the SIEM Server by using the Monitoring Verdicts Interface (MVI), which is composed of Apache Storm and Zookeeper elements. The SIEM Server is in charge of analyzing the detection report and starts the computation of the reaction process.

Table 12. Test case TC_BMS.3.3 – Reaction to SQLi attack on SCADA network

TC_BMS.3.3	Reaction to SQLi attack on SCADA network
Preconditions	• Satisfied preconditions from TC_BMS.3.2 • Decision Support System (DSS) is ready to consume reaction verdicts • DSS is ready to publish alerts • DSS is ready to publish changes to Dynamic Security and Privacy Seal service • DSS is ready to publish changes to Mitigation Action Service
Components	• Components from TC_BMS.3.2 • ATOS SIEM Server (as Decision Support System) • Attack script emulating adversary
Execution	• Start all the components as stated in TC_BMS.3.2

	• Perform attack by running malicious script to inject packets containing SQLi attacks
Expected results	• Result from TC_BMS3.2 • Attack on the SCADA Server blocked successfully by reaction component
Expected completion	Month 18 – June of 2018
KPI(s)	• KPIs as stated in TC_BMS.3.2 • Reaction applied within 1 sec from detection on SEP
Fail criteria	• Fail criteria as stated in TC_BMS.3.2 • Failure one of the components during test • The detected attack is notified to the DSS with more than 1 sec delay

3.4.2.2 Test-Case TC_BMS.3.4

The TC_BMS3.4 test case complements TC_BMS.3.5 in the sense of testing a legit SQL traffic in the monitored network. Following this description, the ANASTACIA platform has to be able to allow the legit traffic and do not raise an alert about this allowed action. In particular, this test aims to lower the false-positive number of detections, and thus raising the reliability of the whole ANASTACIA platform.

Figure 16 shows how the ANASTACIA platform behaves in the case a normal SQL traffic is detected in the monitored network. At the start, both the MMT-Probe and the XL-SIEM modules subscribe to the "monitor" channel of the Kafka Broker. After this, the MMT-Probe starts monitoring the network, generating the statistics reports periodically. At a certain point, the legit SQL script is triggered, inserting a valid SQL query in the network. Since these instructions do not represent a threat for the Security Enforcement Plane, it is not detected by the MMT-Probe, which continues to generate the periodic statistics reports. As the monitoring module did not generate any alert report, the XL-SIEM component does not generate an attack verdict, which avoids triggering the reaction module and all the mechanisms involved in it.

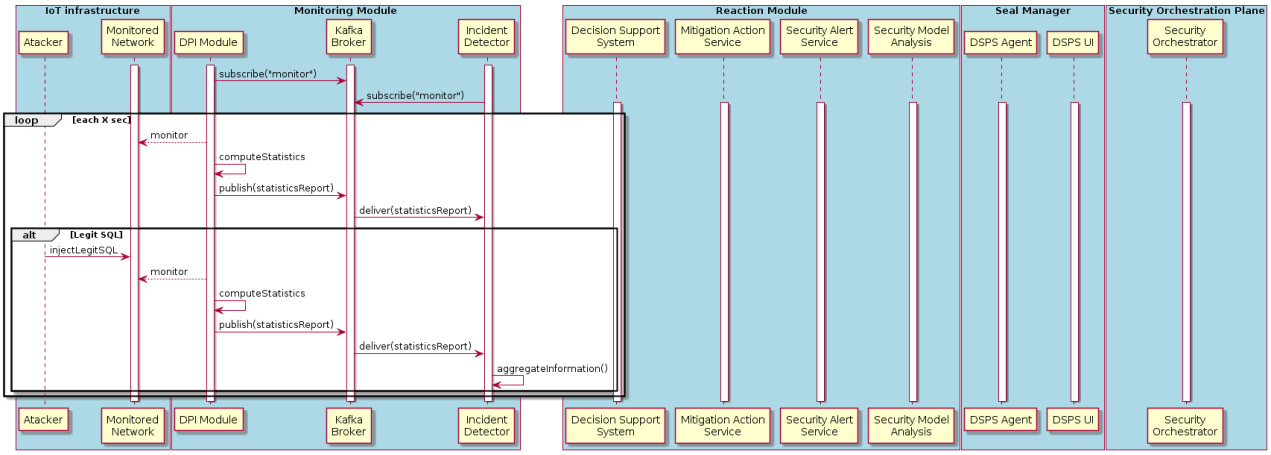

Figure 16 Sequence diagram for TC_BMS.3.4 – Full BMS.3 scenario test with Legit SQL traffic

Table 13. Test case TC_BMS.3.4 – Monitoring and Reaction BMS.3 scenario test with Legit SQL traffic

TC_BMS.3.4	**Monitoring and Reaction BMS.3 scenario test with Legit SQL traffic**
Preconditions	• TC_BMS.3.3 preconditions completed successfully • Decision Support System is ready to publish mitigation action to Kafka broker • Mitigation Action Service is ready to consume messages from Kafka broker • Security service orchestrator is up and running

Components	• As stated in TC_BMS.3.3 • Dynamic Security and Privacy Seal (DSPS) • Security Alert Service (SAS) • Decision Support System (DSS) • Mitigation Action Service (MAS) • Security Orchestrator (SO) • Script emulating legit SQL traffic in the network.
Execution	• Start all the components as stated from above • Perform attack by running attack script to inject packets containing SQLi attacks • Observe logs on monitoring, reaction components as well as DSS, MAS, SO
Expected results	• Monitoring component do not detect any attack • Alert is not raised in the frontend • Not a single reaction is triggered by the Reaction Plane
Expected completion	Month 18 – June of 2018
KPI(s)	• Security Seal stays at the same level
Fail criteria	• An attack was detected (false negative)

3.4.3 Integration Test-Cases

3.4.3.1 Test-Case TC_BMS.3.5

The TC_BMS.3.5 is one of the final tests of the whole use case. It aims to show the whole monitoring and reaction process for the BMS.3 use case.

This test case complements TC_BMS.3.3 in order to include all the interactions the Reaction Module has to use in order to compute the set of countermeasures, update the Security and Privacy Seal, raise alerts on the ANASTACIA front-end and send the results of the analysis (in form of an MSPL file) to the Security Orchestrator in order to be deployed.

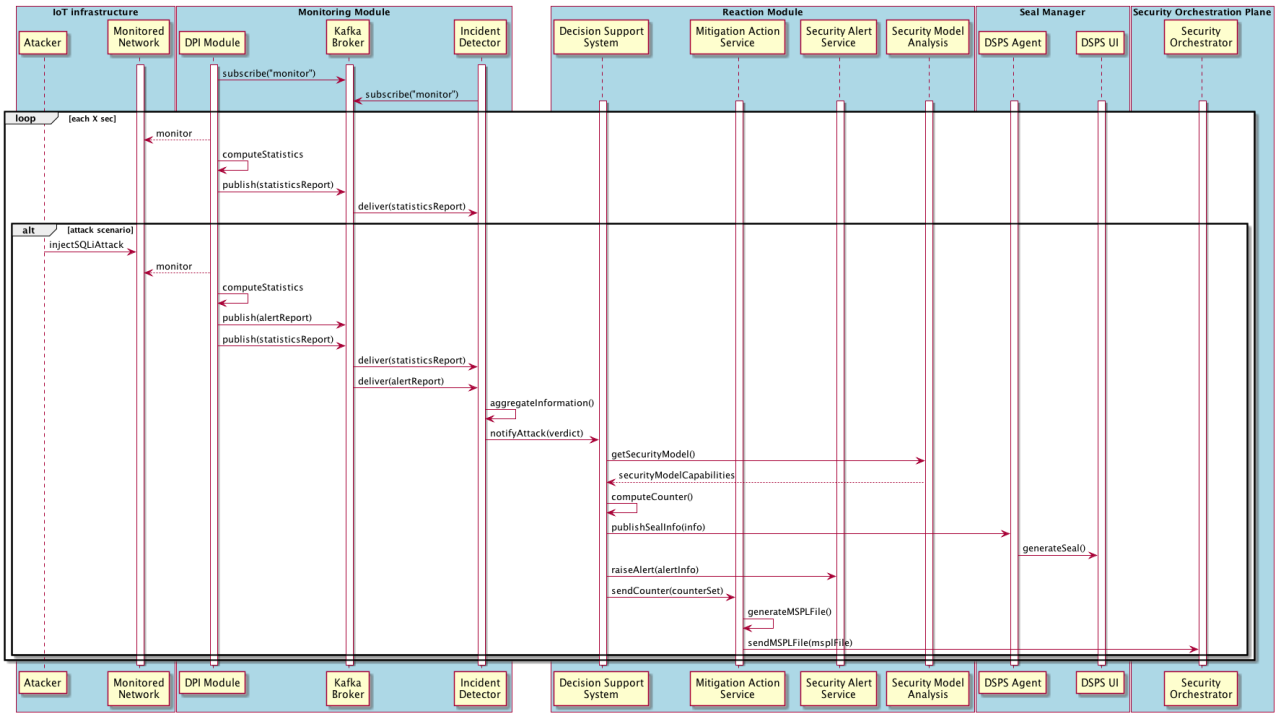

Figure 17 Sequence diagram for TC_BMS.3.5 – Full BMS.3 scenario test

Figure 17 shows the sequence diagram for this test case. In the first stage, the MMT-Probe and XL-SIEM tool subscribe to the "monitor" Kafka Channel. After this initialization phase, MMT-Probe starts monitoring the network, generating periodic statistic reports. Once the SQLi attack is fired, the Montimage module detects the threat and generates an alert report, which is published in the subscribed Kafka Channel. This report is read by the XL-SIEM tool which analyses all the available information at the moment and generates a verdict about the ongoing threat. The verdict is sent to the Decision Support System (ATOS SIEM Server) using the Monitoring Verdicts Interface. Then, the SIEM Server correlates the information and performs a security assessment in order to compute the countermeasures to tackle the attack. This computation is done with the support of the Security Model Analysis module (red arrow in Figure 12), which is in charge of precomputing the policies deployed and the capabilities of the monitored network. Considering this information, the SIEM Server finally computes the set of countermeasures and performs a series of activities:

- Informs the Dynamic Security and Privacy Seal (DSPS): In order to re-compute the value of the Security and Privacy Seal, the SIEM Server puts all the required information to the Security Seal Plane by using the Seal Manager Metadata Interface (SMMI).
- Display an alert: To do so, the SIEM Server sends the information by using the Security Alert Service (SAS).
- Sends the countermeasures to the Security Orchestrator: Finally, the SIEM Server has to inform the Security Orchestrator about the countermeasures to be deployed on the network. To do so, the SIEM Server makes use of the Mitigation Action Service (MAS), which is in charge of translating the response into a MSPL file that will be used by the Security Orchestrator to enforce the security policy

Table 14. Test case TC_BMS.3.5 – Monitoring and Reaction BMS.3 scenario test

TC_BMS.3.5	Monitoring and Reaction BMS.3 scenario test
Preconditions	• TC_BMS.3.3 preconditions completed successfully • Verdict and Decision Support System is ready to publish mitigation action to Kafka broker • Mitigation Action Service is ready to consume messages from Kafka broker • Security service orchestrator is up and running
Components	• As stated in TC_BMS.3.3

	• Dynamic Security and Privacy Seal (DSPS) • DSPS User Interface (UI) • Security Alert Service (SAS) • Decision Support System (DSS) • Mitigation Action Service (MAS) • Security Orchestrator (SO) • Attack script emulating adversary
Execution	• Start all the components as stated from above • Perform attack by running attack script to inject packets containing SQLi attacks • Observe logs on monitoring, reaction components as well as VDSS, MAS, SO
Expected results	• Monitoring component successfully detects all generated attacks • Alert messages visible on user UI • Reaction recommendation sent to the Security Orchestrator in form of an MSPL file • Value of the Security and Privacy Seal updated correspondingly. • Packets containing the SQLi attack successfully dropped by reaction component
Expected completion	Month 18 – June of 2018
KPI(s)	• KIPs as stated in TC_BMS.3.3 • Security Alert visible within 1.5 sec after detection • Security and Privacy Seal updated within 1.5 sec
Fail criteria	• Fail criteria as stated in TC_BMS.3.3 • DSS not propagated requests to MAS • MAS not propagated to SO • Alerts not propagated by SAS • DSPS didn't changed seal level during attack

3.4.3.2 Test-Case TC_BMS.3.6

The TC_BMS3.6 test case extends TC_BMS.3.4 in order to include the execution of the orchestration process triggered by the Reaction Module. In this particular scenario, the Security Orchestrator will process the received MSPL file (representing the computed reaction), determine the correct security enabler to enforce the security policy, and send the commands to the security enabler in order to apply the reaction on the monitored network.

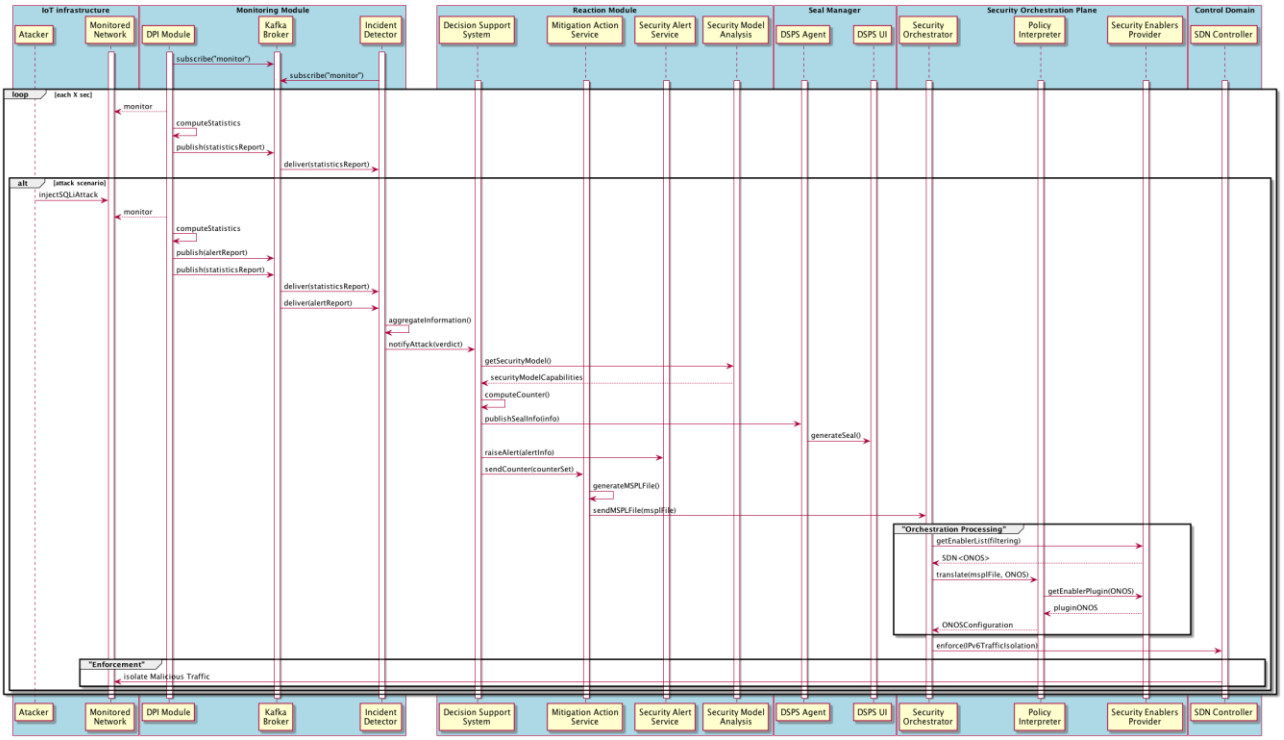

Figure 18 Test case TC_BMS.3.6 – Reaction Enforcement Process

Figure 18 shows the interactions of the TC_BMS.3.6 test case. The first stage starts with the subscription of MMT-Probe and the XL-SIEM tool to the "monitor" channel of the Kafka Broker. After this, MMT-Probe periodically publishes the statistics reports to the channel, which is also delivered to the XL-SIEM. In case an attack is detected by the DPI Module (MMT-Probe), an alert is raised, and an alert report is published in the Kafka channel along with the periodic statistics report. The Incident Detector (XL-SIEM) will aggregate al the received information in order to generate a final verdict about the detected incident. This verdict report will be transmitted to the Decision Support System (DSS), in the Reaction Module. The DSS will start the computation of the reaction, by making the respective calls to the Security Model Analysis (to obtain the capabilities allowed by the security policy loaded in the platform), to the Seal Manager (to update the value of the ANASTACIA security seal), to the Security Alert Service (to raise an alert about the detected attack on the network) and, finally, to the Mitigation Action Service (to transmit the computed reaction to the Security Orchestrator).

Once the Security Orchestrator (SO) receives the MSPL file containing the reaction, it starts the orchestration process in order to deploy the reaction to counter the attack. First, the SO retrieves the Security Enablers allowed by the security policy, being in this case, an ONOS-based SDN controller. Using the MSPL from the Reaction Module and the list of security enablers, the SO calls the Policy interpreter to translate the MSPL file, obtaining a set of instructions that will be sent to the security enabler to counter the attack. In this case, a set of ONOS configurations to filter IPv6 traffic is returned. Finally, these instructions are sent to the SDN Controller, in the control domain, component in charge of executing the instructions in order to isolate the malicious detected traffic.

Table 15 Test Case TC_BMS.3.6 – Attack Detection, Reaction and Security Orchestration in BMS.3 scenario test

TC_BMS.3.6	Attack Detection, Reaction and Security Orchestration in BMS.3 scenario test
Preconditions	• TC_BMS.3.4 preconditions completed successfully • Policy Interpreter
Components	• As stated in TC_BMS.3.4 • Policy Interpreter (PI) • Security Enablers Provider (SEP) • SDN Controller (SDN-Cont)
Execution	• Start all the components as stated from TC_BMS.3.4 • Perform attack by running attack script to inject packets containing SQLi attacks • Observe logs on monitoring, reaction components as well as DSS, MAS, SO, PI, SEP and SDN-Cont
Expected results	• Monitoring component detect the ongoing attack • An Alert is raised in the frontend • A Reaction is triggered by the Reaction Plane • The SO translates and communicates the instructions to the SDN controller • Network configuration changes according the applied countermeasure • Affected network traffic is isolated according with the reaction computed
Expected completion	Month 18 – June of 2018
KPI(s)	• Security Seal changes to a lower level • An alert is raised on the ANASTACIA UI
Fail criteria	• Fail Criteria according to TC_BMS.3.4 • The Security Seal does not change accordingly • An Alert about the attack is not raised on the ANASTACIA UI • Security Orchestrator does not send the instructions to SDN Controller • SDN controller does not execute the instructions sent by the SO

3.5 TEST-CASE TC_BMS.4: CASCADE ATTACK ON A MEGATALL BUILDING

The objective in this test-case (TC_BMS.4) is examining ANASTACIA platform for protecting the system from a remote data tempering for sensitive sensor and actuation data. This test is exploring UC_BMS.4: *Cascade attack on a megatall building*, where a hacker plans to gain control over critical temperature sensor to manipulate the temperature value and hence triggering the fire and evacuation alarms. This test is covering attack models implementation, monitoring that uses data analysis to detect anomaly behavior of temperature value, and then the test uses security orchestrator to stop the malicious traffic by requesting to the SDN controller and IoT controller.

Test-cases have been divided into 4 steps. Reason for division is to enable team to properly validate parts ANASTACIA framework separately first and as components will start working increase level of test case complexity by adding more components to test scenario. The test cases division has been illustrated on Figure 19.

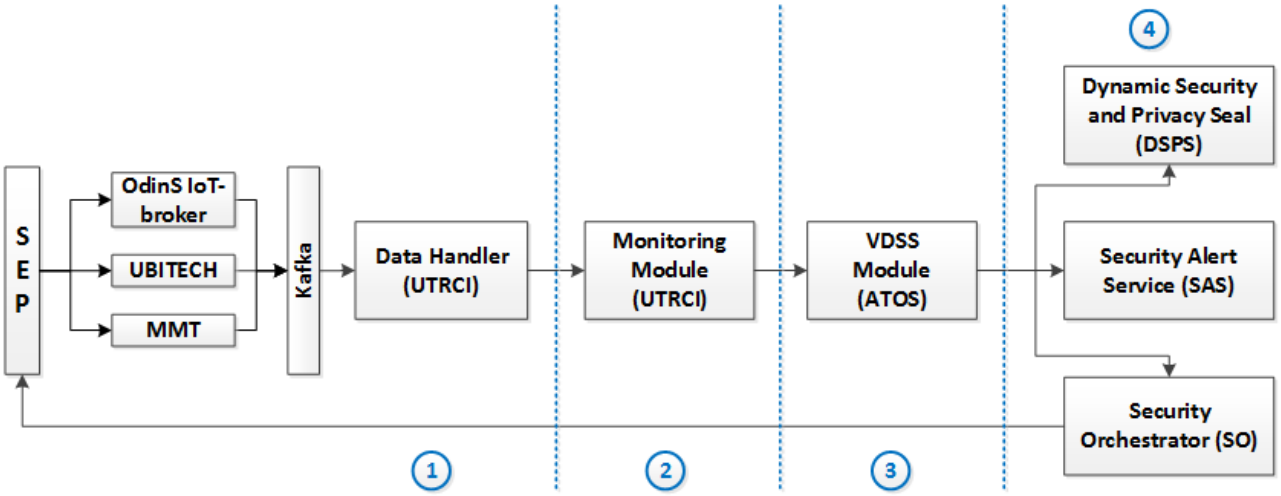

Figure 19. SW architecture for test case TC_BMS_4

Designed test steps are following:

1. Basic input validation – aim of this step is to validate if all required inputs are being received through Kafka broker and/or via REST API from OdinS IoT-broker and MMT agents.
2. Monitoring test – will check UTRCI component capability to detect attacks performed by attack script emulating adversary.
3. Reaction test – will evaluate ability of VDSS service to react to performed attacks within specified time limit.
4. Full test – will look into full TC_BMS.4 test case scenario using ANASTACIA framework components and check how each component behave when SEP infrastructure is being attacked.

Test-cases will use attack script to generate temperature sensor attacks in SEP network. The script will emulate malicious behavior of adversary that will be used to validate and verify UTRC and other components behavior during performed attack on SEP. The script algorithm was illustrated on Figure 20. The adversary attack algorithm will start with SEP topology mapping to get full SEP topology. On next step algorithm will store all current temperature sensors to restore them after attack completion. Back process is performed in order to minimize impact of the attack script into underlying infrastructure.

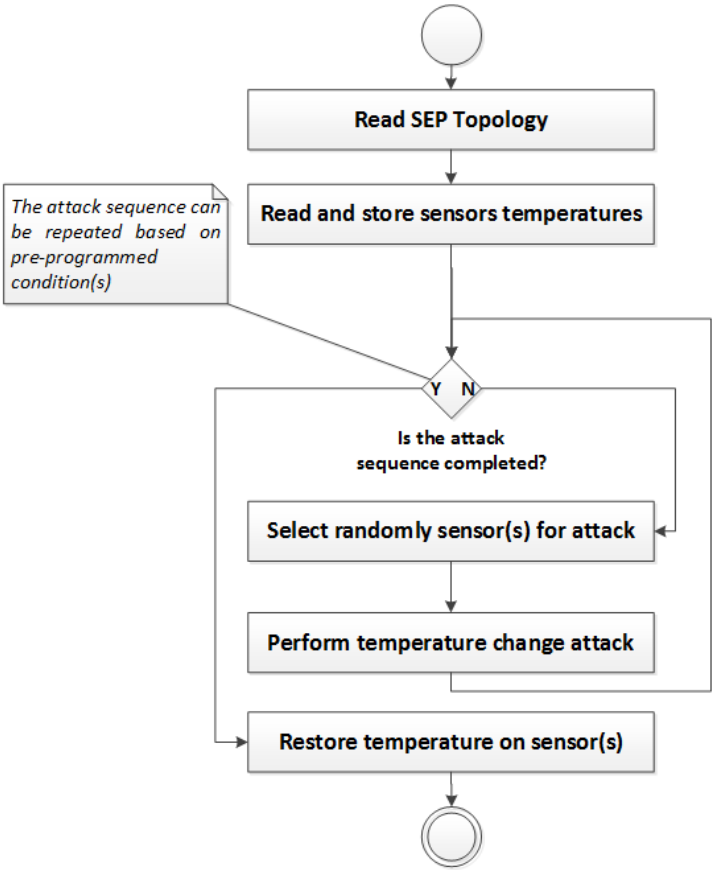

Figure 20. Temperature script change algorithm that will emulate adversary in final test case TC_BMS_4.4.

Then depend on the algorithm pre-programmed logic algorithm will perform temperature sensor attack. First algorithm will select sensors to be attacked, next it will attempt to change temperature on them. After first round of attack depend on the loop logic algorithm might decide to run another round of attack on different type of sensors or continue attack on the current set. After multiple iterations once programmed sequence of attack has been completed algorithm will reset all attacked sensors to their original temperature values. Reason is that in case of component failure test completion will not affect sensor operation after test execution.

3.5.1 Individual Component Test-Cases

3.5.1.1 Test-Case TC_BMS.4.1 – Connection with OdinS IoT broker

This test case will verify connection established between OdinS and UTRC components and ability of UTRC Data Analysis component to consume SEP data. Figure 21 illustrates sequence flow for test case TC_BMS4.1.

The test case is divided into two phases. During connection initialization OdinS IoT Broker starts publishing SEP data to Kafka broker. SEP data represents any type of sensor information that is collectable by IoT Broker on SEP. UTRC Data Analysis component will subscribe to the broker topic to consume messages published by OdinS IoT broker.

In next phase of TC_BMS4.1 test SEP data events will be consumed by UTRC Data Analysis component. Each SEP data event will be reported via Kafka Broker as SEP data received test event (**TE**) to Unit Test TC_BMS4.1 script that will store and analyses test information (KPIs calculation etc.). TC_BMS_4.1 sequence diagram has been depicted on Figure 21.

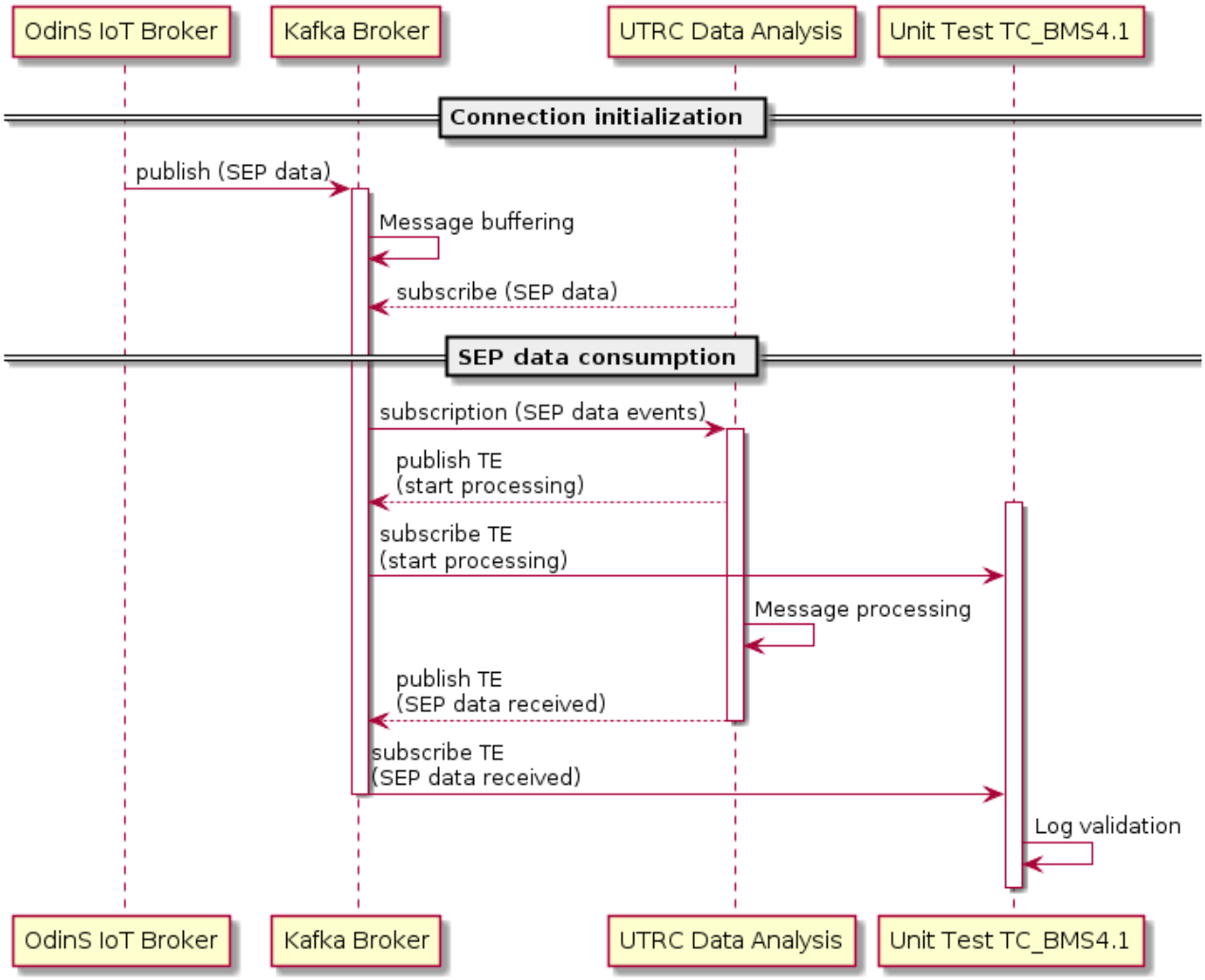

Figure 21. Sequence diagram for TC_BMS_4.1 – connection with OdinS IoT broker.

Table 16. Test case TC_BMS.4.1 – Connection with OdinS IoT-broker

TC_BMS.4.1	Connection with OdinS IoT-broker
Preconditions	• SEP component is working - REST API is available online • Kafka message broker is configured and up and running • OdinS IoT-broker is publishing network information to Kafka message broker • UTRC Data Analysis is subscribed to the network monitoring data topic on the broker
Components	• SEP • Kafka Broker • OdinS IoT Broker • UTRC Data Analysis
Execution	• Start all the components from above in that order • Observe JSON messages coming from OdinS IoT-broker on Kafka Broker being logged on UTRC Data Analysis
Expected results	• UTRC Data Analysis is consuming physical values from OdinS IoT-broker via Kafka Broker • UTRC Data Analysis is storing data to a dataset correctly

Expected completion	Month 19 – July of 2018
KPI(s)	• UTRC Data Analysis is stable during test • All information consumed from SEP is stored • All information consumed from SEP is published to Kafka topic in accordance to JSON message specification
Fail criteria	• Exception caught during UTRC Data Analysis execution • Data records not stored on a local store • Some part of data lost during conversion and not published on Kafka message broker

3.5.1.2 Test-Case TC_BMS.4.2 – Attack detection based on IoT monitoring data

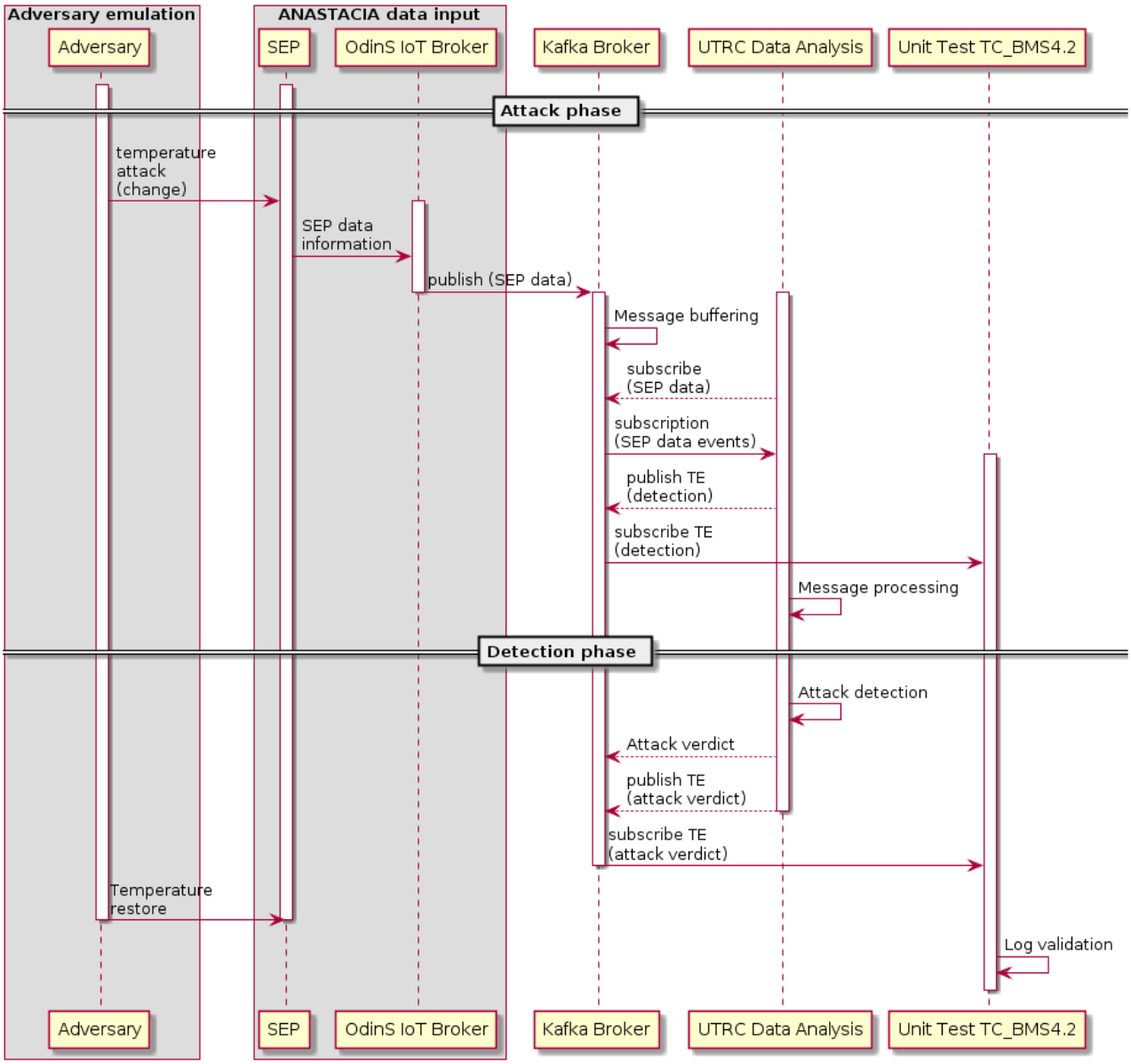

Figure 22. Sequence diagram for TC_BMS_4.2 – attack detection based on IoT monitoring data

In this test scenario (Figure 22) test script emulating adversary will be used to perform attack on randomly selected sensor in SEP plane. Next OdinS IoT Broker will deliver changes in temperatures observed in SEP via

Kafka broker to UTRC Data Analysis component. On the component side internal model will send initial test event to Unit Test TC_BMS.4.2, process messages and perform attack detection after which attack verdict will be delivered via Kafka broker to other ANASTACIA components. At the time when attack verdict is being delivered new message will be generated for Unit Test TC_BMS.4.2 to enable KPIs processing.

Table 17. Test case TC_BMS.4.2 – Attack detection based on IoT monitoring data

TC_BMS.4.2	Attack detection based on IoT monitoring data
Preconditions	• TC_BMS.4.1 completed successfully • SEP operational • OdinS IoT-broker API is operational (messages published on Kafka broker or available via REST API) • MMT agent operational (messages published on Kafka broker) • OdinS IoT-broker is publishing network information to Kafka message broker • MMT agent is monitoring network traffic • MMT agent is publishing information to Kafka message broker • Attack script that will emulate malicious behavior and is capable of changing temperature sensor on continuous manner
Components	• SEP • Kafka message broker • OdinS IoT-broker • MMT agent • Monitoring agent (*UTRC Data Analysis*) • Attack script emulating adversary
Execution	• Start all the components from above in that order • Start attack script that emulates adversary behavior • Wait for temperature change and detection report • Observe component behavior (logs dumped by unit test)
Expected results	• Successful temperature change detection by UTRC Data Analysis module • Detection information is being passed to Kafka topic accordance to JSON message specification.
Expected completion	Month 19 – July of 2018
KPI(s)	• Requirements as stated in TC_BMS.4.1.2 • Correct detection of malicious activity with precision **(>90%)** • Attack detected within 500msec timeframe
Fail criteria	• Requirements as above in TC_BMS.4.1.2 • Detection precision criteria below expected level from above • Detection performed above 500msec timeframe

3.5.2 Components Interaction Test-Cases

3.5.2.1 Test-Case TC_BMS.4.3

This test case will be further extension of TC_BMS.4.2 where UTRC Data Analysis cooperation with ATOS VDSS will be demonstrated. As in previously, test case will start from attack emulation generated by adversary script. Next monitoring SEP data will be delivered to Kafka Broker. Afterward UTRC Data Analysis component will subscribe and start processing monitoring information in order to compute attack verdict. Start of the component processing will marked by test event (TE) send to Unit Test TC_BMS.4.3 script that will use this information to compute test cases KPIs. Once attack verdict is computed and sent to VDSS Module via Kafka Broker, a reaction recommendation is calculated by VDSS Module and send to Kafka Broker for further processing. On different queue VDSS component will send TEs to Unit Test TC_BMS.4.3 script that will use them for test case evaluation and validation during log validation step. As previously the adversary emulation script will restore temperatures on SEP to ensure that underlying infrastructure will be left in the same state as before the test. Whole test case sequence diagram has been illustrated on Figure 23.

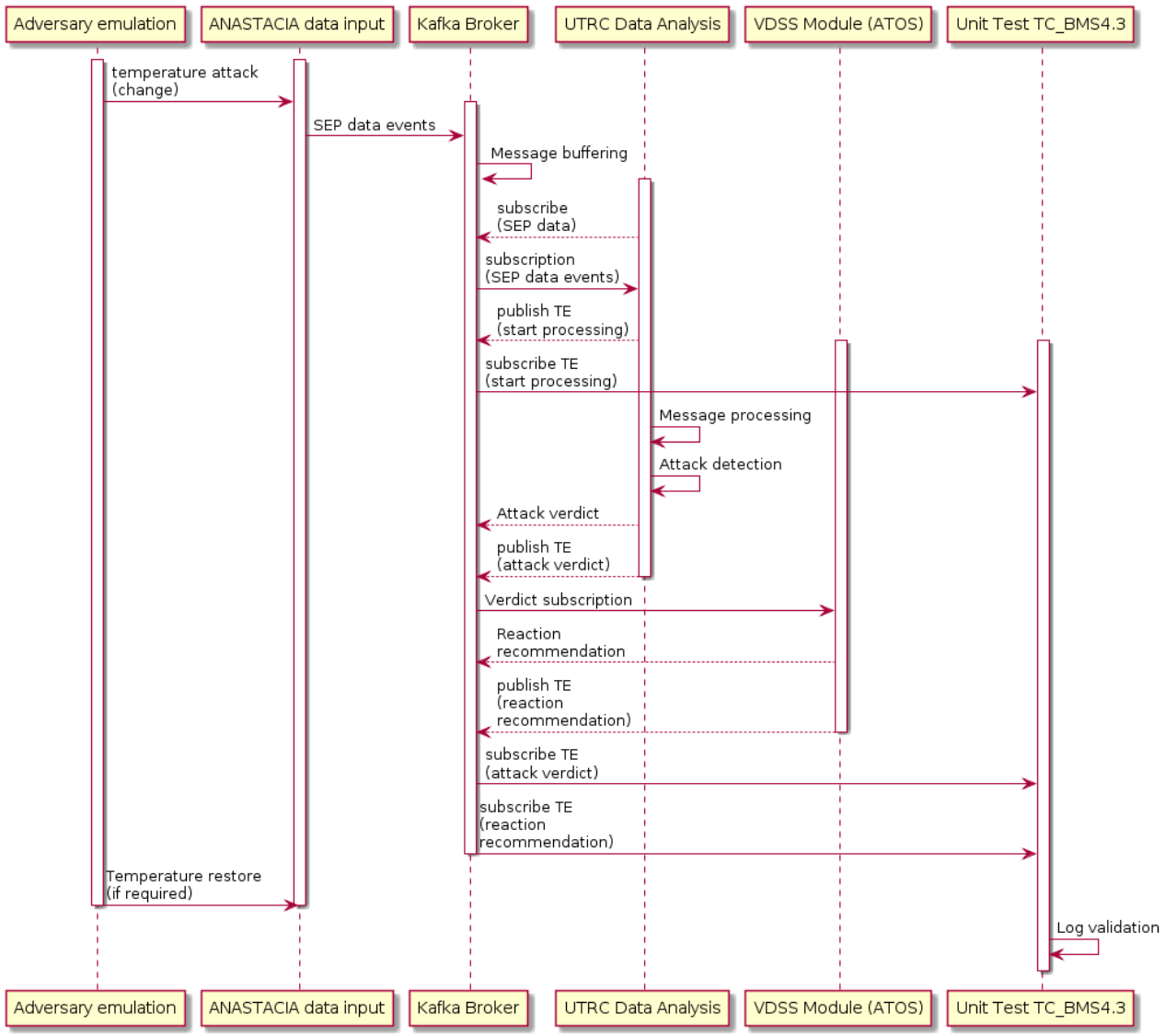

Figure 23. Sequence diagram of TC_BMS_4.3 – reaction to attack on sensor.

Table 18. Test case TC_BMS.4.3 – Reaction to attack on sensor

TC_BMS.4.3	Reaction to attack on sensor
Preconditions	• Satisfied preconditions from TC_BMS.4.2 • Verdict and Decision Support System (VDSS) is ready to consume reaction verdicts • VDSS is ready to publish alerts • VDSS is ready to publish changes to Dynamic Security and Privacy Seal service • VDSS is ready to publish changes to Security Orchestrator
Components	• Components from TC_BMS.4.2 • VDSS • Attack script emulating adversary
Execution	• Start all the components from above in that order (SEP, Monitor and Reaction components) • Start attack script that emulates adversary behavior • Perform attack by running malicious script to change sensor temperature
Expected results	• Monitoring component successfully detected • Attack on a sensor blocked successfully by reaction component
Expected completion	Month 20 – August of 2018
KPI(s)	• KPIs as stated in TC_BMS.4.2 • ANASTACIA infrastructure stable during test • Reaction applied within 1 sec from detection on SEP
Fail criteria	• Fail criteria as stated in TC_BMS.4.2 • Failure one of the components during test • Reaction applied on SEP with more than 1 sec delay

3.5.3 Integration Test-Case

3.5.3.1 Test-Case TC_BMS.4.4 – Full BMS_4 test case scenario

Overall in accordance to test steps described for use case BMS_4 this is the last step that will use all previously described steps of component interactions between adversary emulation, ANASTACIA data input, Kafka broker, UTRC Data Analysis component, VDSS Module (ATOS) and combine them to close loop with other reaction components (Security Alert Service, Dynamic and Privacy Seal) and Security Orchestrator. Detailed sequence diagram is illustrated on Figure 24 while test case description has been placed in Table 19.

After processing all steps from TC_BMS.4.3 the VDSS module is sending three signals to other ANASTACIA components:

- Reaction recommendation (MSPL file) to Orchestration service,
- Security seal level change request to Dynamic Privacy and Security Seal component,
- Alerts to Security Alerts Service.

Once this process is completed as in previous test cases Unit Test TC_BMS.4.4 script will perform log analysis to calculate test KPIs for further analysis. Depending on the outcome of the scenario Adversary emulation script might restore automatically temperature on SEP to restore clean state on SEP. Details on how other reaction components will execute reaction recommendations, security seal level change and display alerts

are out of the scope of this test case. The only requirement here is that all reaction component should publish status of their actions to Kafka broker for further test case analysis.

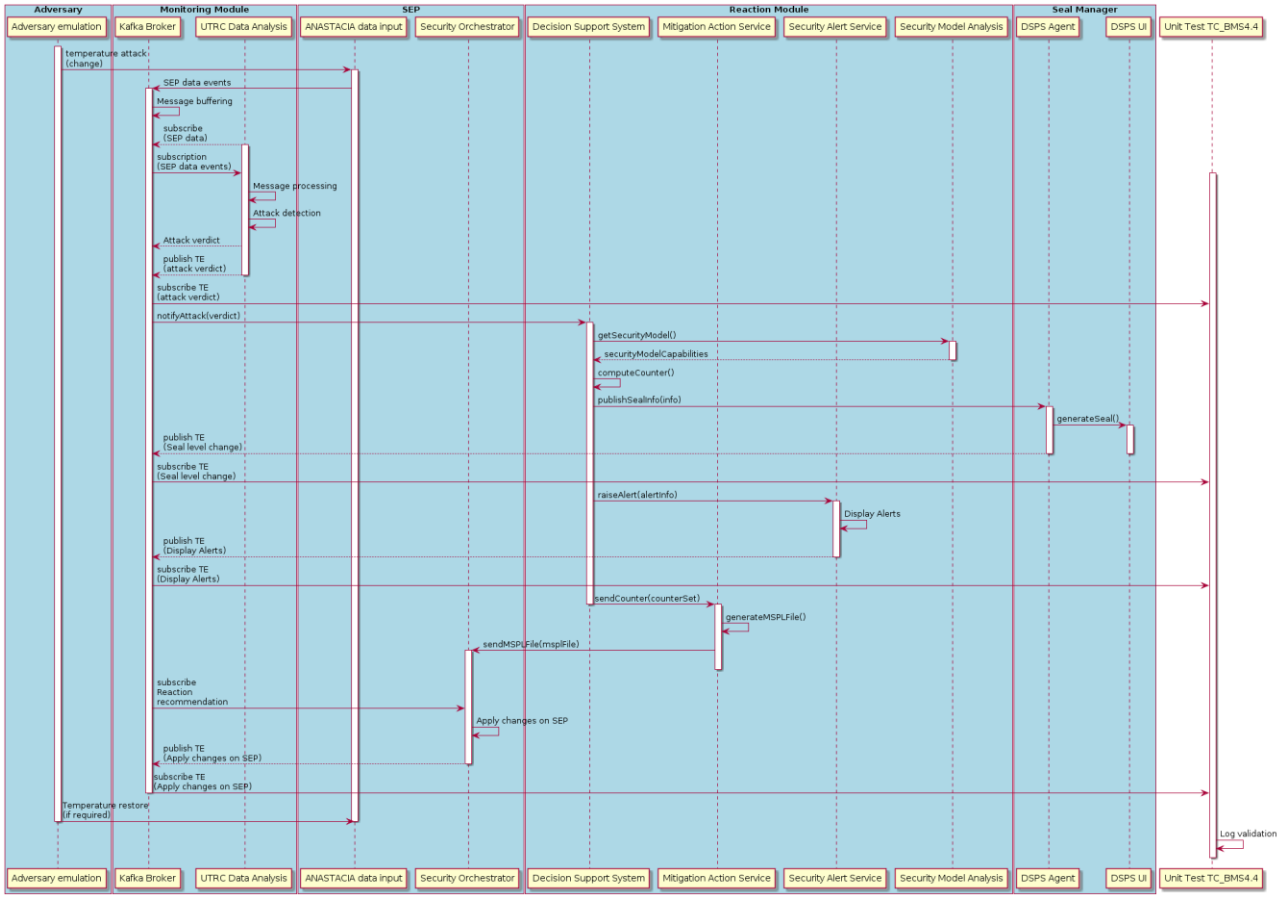

Figure 24. Sequence diagram of TC_BMS_4.4 – Full BMS.4 scenario test.

Looking into more detail at Orchestration Processing level there are few additional steps that are executed in order to apply MSPL policy file into SEP. Detailed interaction between Orchestrator components and SEP are illustrated on Figure 25. After Mitigation Action Service generates MSPL file and send it to SO (steps 1-2), the component will check IoT enablers list (3) and receives from SO IoT controls list (4) which are then locally processed in order to select correct IoT control (5). Next MSPL file and IoT control will be translated to appropriate plugin for IoTContol (6-8) by Policy Interpreter and Security Enablers Provider. In step (9) IoTControl configuration is send to SO and in step (10) IoTControl action is sent to stop the traffic to IoT Controller. Finally in step (11) IoT Controller send request to turn off IoT device on SEP.

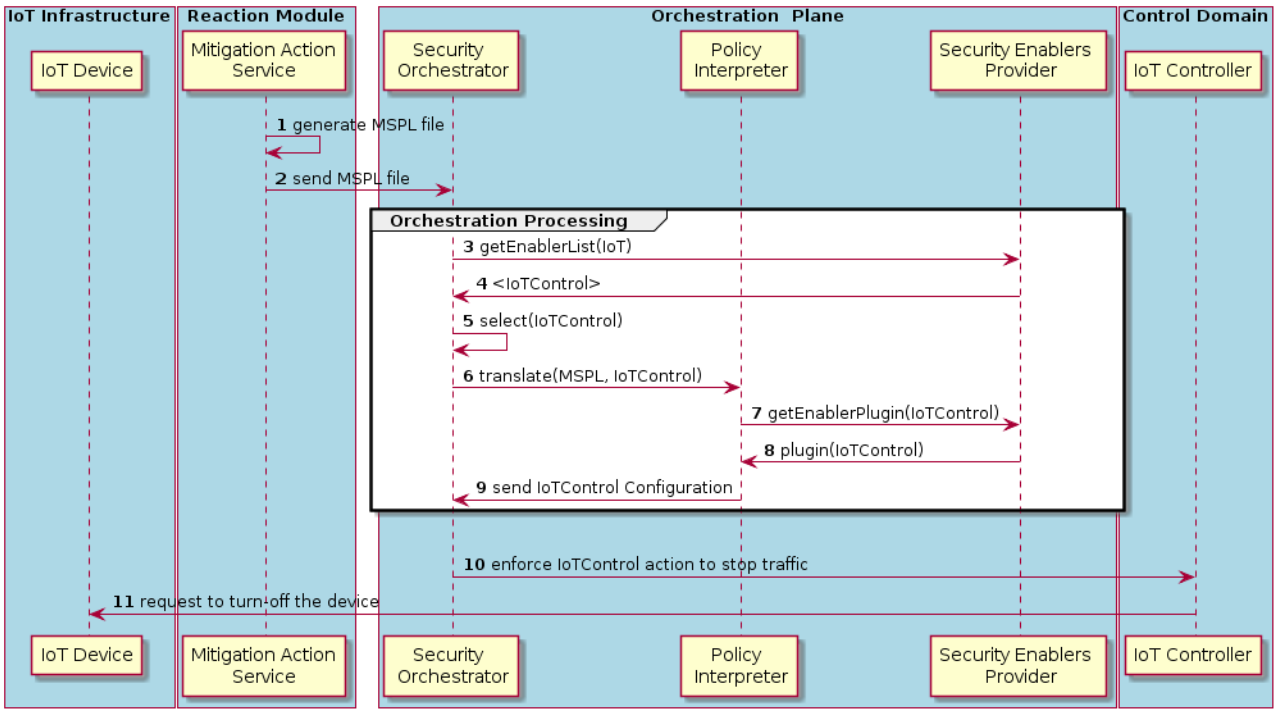

Figure 25. Sequence diagram of TC_BMS_4.4 – Details of orchestration processing section.

Table 19. Test case TC_BMS.4.4 – Full BMS.4 scenario test

TC_BMS.4.4	Full BMS.4 scenario test
Preconditions	• TC_BMS.4.3 preconditions completed successfully • Verdict and Decision Support System is ready to publish mitigation action to Kafka broker • Mitigation Action Service is ready to consume messages from Kafka broker • Security service orchestrator is up and running
Components	• As stated in TC_BMS.4.3 • Dynamic Security and Privacy Seal (DSPS) • Security Alert Service (SAS) • Verdict and Decision Support System (VDSS) • Mitigation Action Service (MAS) • Security Orchestrator (SO) • Attack script emulating adversary
Execution	• Start all the components as stated from above • Perform attack by running attack script to change sensor temperature • Observe logs on monitoring, reaction components as well as VDSS, MAS, SO
Expected results	• Monitoring component successfully detects all generated attacks • Alerts messages visible on user UI • Reaction recommendation taken • Attacks on sensors blocked successfully by reaction component
Expected completion	Month 20 – August of 2018

KPI(s)	• KIPs as stated in TC_BMS.4.3 • Security Alert visible within 1.5 sec after detection • Security and Privacy Seal updated within 1.5 sec
Fail criteria	• Fail criteria as stated in TC_BMS.4.3 • VDSS not propagated requests to MAS • MAS not propagated to SO • SO didn't applied mitigation to SEP • Alerts not propagated by SAS • DSPS didn't changed seal level during attack

3.6 Test-Case TC_MEC.3: DoS/DDoS Attacks using Smart Cameras and IoT Devices

This section describes the test case for the use case UC_MEC.3. The main objective of the use case is to validate ANASTACIA system against malicious IoT attackers that will attempt to denial of service through cameras system. In this use case a group of hackers gets the IP address of IoTs and cameras and use if for DDoS attack. Thus, this section identifies the different steps of test-cases to implement and evaluate UC_MEC.3 on Mobile (Multi-access) Edge Computing testbed.

3.6.1 Individual Component Test-Cases

This section presents a test-case to evaluate the performance of Anastacia system against DDoS attack performed by IoT devices including video cameras. The following table shows the description of the test case to evaluate and validate the capability of dynamically detect and cope a DDoS attack enabled by the ANASTACIA framework.

3.6.1.1 TC_ MEC.3.1 Device Actuation

The following sub test case implements the IoT device attack. When an attack is initiated the attacker launches a high amount of ICMP requests through the corrupted IoT device. Table 20 shows the description of the sub test to validate the Anastacia Framework while sequence diagram has been illustrated on Figure 26.

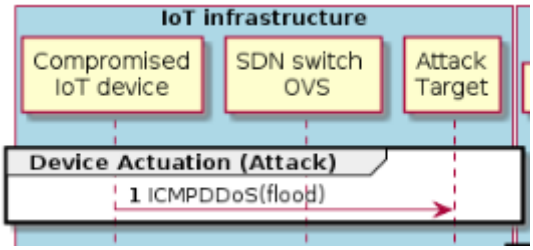

Figure 26 Sequence diagram for TC_MEC3.1 – Device actuation

Table 20. Test case TC_MEC.3.1 – Device actuation

TC_MEC.3.1	Device actuation
Preconditions	• IoT devices has been hacked by an attacker, and is ready to send an ICMP flooding attack

Components	• Compromised IoT device (in form of an attack-generating script or modified device firmware to launch an attack) • Attack target device
Execution	• Attacker instructs the compromised IoT devices to launch a DoS attack using ICMP requests.
Expected results	• The IoT device(s) begin the attack by sending a high amount of ICMP requests.
Expected completion	Month 15 – March 2018
KPI(s)	• An abnormal amount of ICMP ping requests (more than 10 ICMP request every 5 seconds) is observed in the network.
Fail criteria	• The ICMP ping packets are in normal levels, meaning no attack.

3.6.2 Components Interaction Test-Cases

This subsection presents the different test case components and their interaction, where each component belongs to the following module: IoT Infrastructure, Monitoring Module, Reaction Module, Seal Manager, Orchestration Plane and Control Domain.

In this subsection we give a description of the test-cases involved in use case MEC.3:

- TC_ MEC.3.2 Monitoring processing
- TC_ MEC.3.3 Reaction Processing
- TC_ MEC.3.4 Seal Processing
- TC_ MEC.3.5 Orchestration Processing
- TC_MEC.3.6 Enforcement

3.6.2.1 TC_ MEC.3.2 Monitoring processing

The following sub test case implements the monitoring processing. When an attack is launched the monitoring module – and more specifically one of the instances of the deployed Monitoring Agents – should detect the current attack and alert the incident detector component. For the MEC.3 scenario and its respective test cases, we will use a MMT-Probe instance as the Monitoring Agent of the network, which is capable of detecting ICMP flooding attacks.

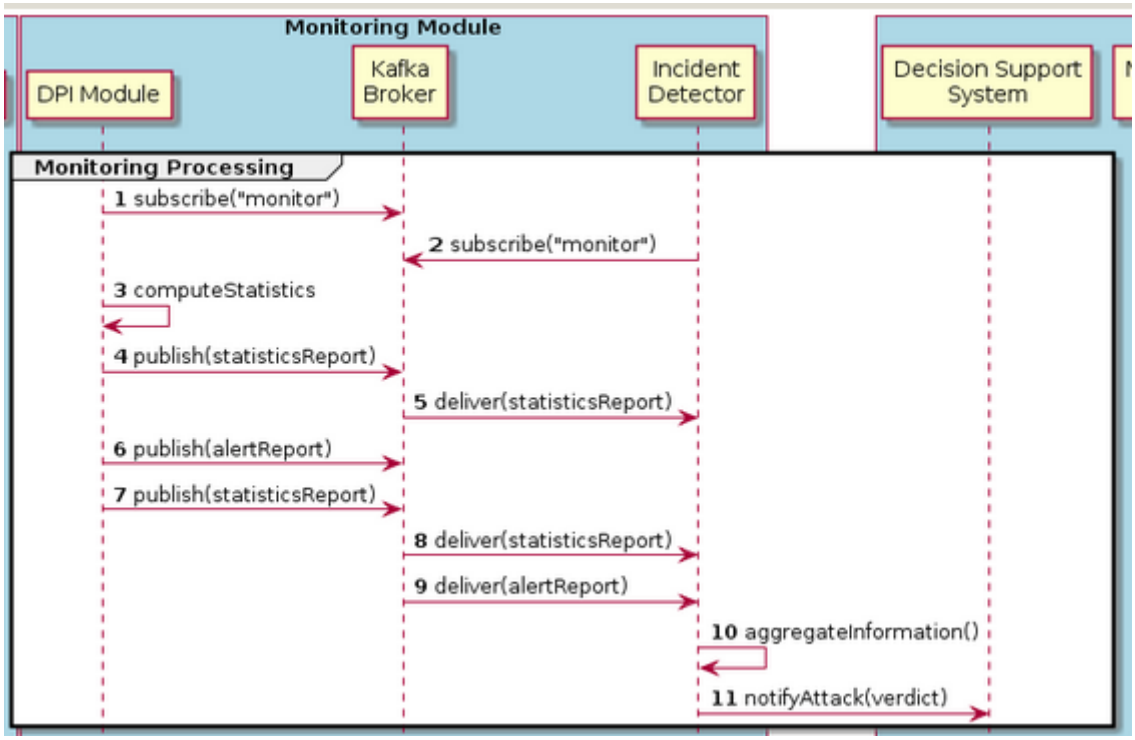

Figure 27 Sequence diagram for TC_MEC3.2 – Monitoring processing

Table 21. Test case TC_MEC.3.2 – Monitoring processing

TC_MEC.3.2	Monitoring processing
Preconditions	• Preconditions as in TC_BMS.3.1 • The Monitoring Agent (MMT-Probe) is deployed and running in the monitored network. • The Kafka Broker is started and available to receive messages • The Incident Detector is running and ready to receive data form the Monitoring Agents
Components	• MMT-Probe instance (as Monitoring Agent) • Kafka Broker • ATOS XL-SIEM (as Incident Detector)
Execution	• The Monitoring Agent is started accordingly. • The Kafka Broker is started accordingly • The Incident detector is started accordingly • The attack scenario is started depending on it implementation (either starting the attacking script or triggering the attack on the modified devices.
Expected results	• The Monitoring agent actively monitors the traffic of the IoTs infrastructure • The Kafka Broker receives the alert from the monitoring agents and delivers the message to the Incident Detector.
Expected completion	Month 17 – May of 2018
KPI(s)	• The Incident Detector has been adverted about the anomaly
Fail criteria	• The Monitoring Agent did not detect the ongoing attack • The Kafka Broker did not deliver the alert to the Incident Detector.

3.6.2.2 TC_ MEC.3.3 Reaction Processing

The following sub test case implements the reaction processing. The monitoring module and more specifically the Incident Detector component alerts the reaction module about security anomaly. The reaction module should select the mitigation action and process it with the support of the Security Model Analysis module. Once the countermeasure has been decided, an alert is risen transmitted to the ANASTACIA UI using the Security Alert Service, the information about the detected attack is made available to the Seal Manager and the Reaction is communicated (in form of an MSPL file) to the Security Orchestrator.

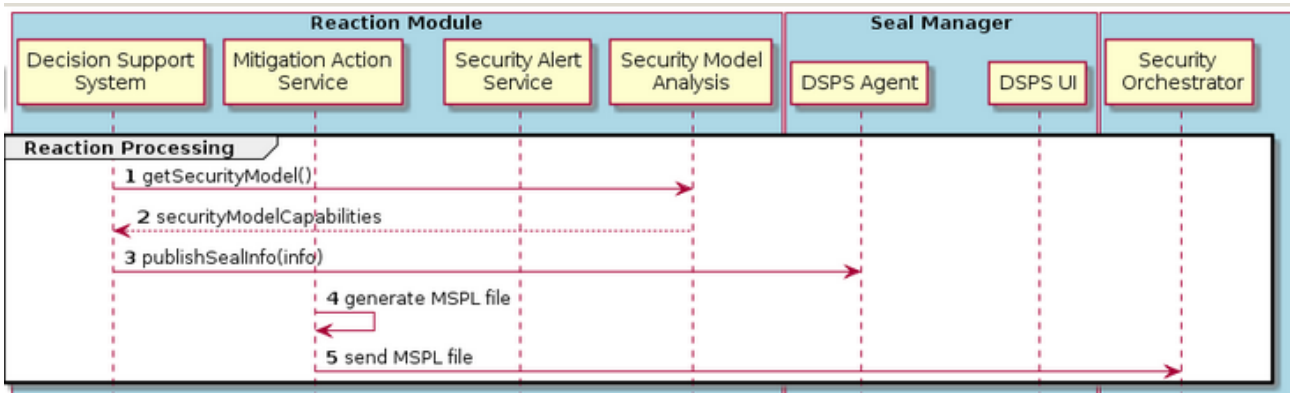

Figure 28 Sequence diagram for TC_MEC3.3 – Reaction processing

Table 22. Test case TC_MEC.3.3 – Reaction processing

TC_MEC.3.3	Reaction processing
Preconditions	• The Incident Detector has been alerted about the DoS attack
Components	• Monitoring Module: Incident Detector • Decision Support System • Mitigation Action service • Security Model Analysis • Security Alert Service • DSPS Agent
Execution	• The Incident Detector component alerts the Decision Support System component about the DoS attack • The Decision Support System component studies the alert (by calling the Security Model Analysis) and notifies the Mitigation Action Service, the Security Alert Service and the DSPS Agent in the Seal Manager module • Finally, the mitigation Action service will generate an MSPL file. The MSPL file will be sent to the Security Orchestrator.
Expected results	• The Seal Manger is alerted by the security attack • The Security Alert Service is alerted about the detected attack in order to show the notification in the ANASTACIA UI. • The Mitigation Action Service is alerted about the security attack and is already to send the MSPL file to the orchestrator
Expected completion	Month 17 – May of 2018

KPI(s)	• The Seal Manager gets the notification • The Security Alert Service receives the information about the detected attack • The Mitigation Action Service selects the accurate reaction • The Security Orchestrator receives the MSPL file
Fail criteria	• Decision Support System component does not alert the other components • The Security Model Analysis does not inform the Decision Support System about the available security capabilities specified in the security policy

3.6.2.3 TC_ MEC.3.4 Seal Processing

The following sub test case implements the seal processing. The reaction module advertises the Seal Manager about the security attack and the Seal Manager will alert the end user through the user interface.

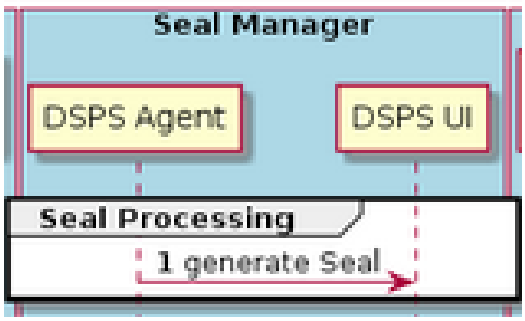

Figure 29 Sequence diagram for TC_MEC3.4 – Seal processing

Table 23. Test case TC_MEC.3.4 – Seal processing

TC_MEC.3.4	Seal processing
Preconditions	• The DSPS Agent receives the alert from the Decision Support System component
Components	• DSPS Agent • DSPS UI
Execution	• DSPS Agent generates the accurate seal and sends it to DSPS User Interface in order to post it to the end user.
Expected results	• DSPS User Interface receives the accurate seal
Expected completion	Month 18 – June of 2018
KPI(s)	• The user Interface shows an security attack
Fail criteria	• DSPS Agent generates does not generate the seal or the accurate seal • DSPS Agent generates does not alert the DSPS User Interface. • The DSPS User Interface doesn't react to the alert

3.6.2.4 TC_ MEC.3.5 Orchestration Processing

The following sub test case implements the orchestration processing. When the Mitigation Action Service reaction receives a security alert it will generate an MSPL file that contains the accurate reaction (defense capability), here in our example the reaction capability is “Filtering”. This file is sent to Security Orchestrator component. When the security orchestrator component receives the MSPL file from the reaction module, it requests the enablers list for the identified capabilities from Security Enabler Provider component. The component selects the list of the enablers that fit with the sent reaction capability, as an example, for filtering capability the enablers can be “Open vSwitch (OVS), Snort, IpTable, etc”. This list is sent to the Security Orchestrator where a sub module in the Security Orchestrator called the Resource planner will select the adequate enabler among the send enablers list. The Security Orchestrator sends the MSPL file and the selected enabler to the Policy Interpreter. The Policy Interpreter will request the proper plugin to the security enabler provider and once the plugin has been received, the Policy Interpreter will use the plugin in order to translate it from the medium level policy to a specific configuration “low level translation” for the selected security enabler. Finally, the Policy Interpreter sends the low level configuration to the Orchestrator as a response of the translation request.

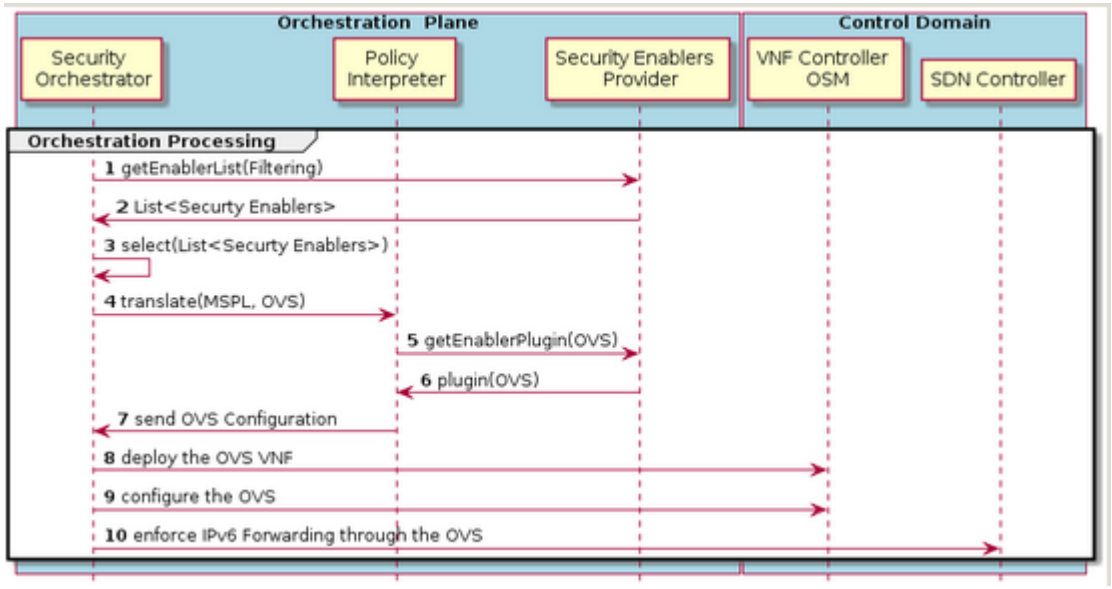

Figure 30 Sequence diagram for TC_MEC3.5 – Orchestration processing

Table 24. Test case TC_MEC.3.5 – Orchestration processing

TC_MEC.3.5	Orchestration processing
Preconditions	• The Orchestration plan receives an MSPL file with a specific reaction
Components	• Reaction Module: Mitigation Action Services • Security Orchestrator • Policy Interpreter • Security Enablers Provider
Execution	• The Mitigation Action Service reaction sends a MSPL file that contains the accurate reaction (defense capability), • Security Orchestrator requests the enablers list for the identified capabilities from Security Enabler Provider component, • Resource planner (in the Orchestrator) receives the enablers list and it decides the enabler to use, according on the system model information, • The Security Orchestrator sends to the Policy Interpreter the MSPL file and the selected enabler, requesting a policy translation,

	• The Policy Interpreter requests to the Security enabler provider the specific plugin for the selected enabler, • The Policy Interpreter receives the enabler plugin and then, it performs the MSPL to low level translation, • The Policy Interpreter sends the low level configuration to the Orchestrator as a response of the translation request • The security orchestrator instruct the VNF controller "OSM" to deploy VNF enabler • The security orchestrator sent the adequate VNF configuration to the VNF controller "OSM" • The security orchestrator instruct the SDN Controller to enforce IPv6 forwarding through the OVS-Firewall
Expected results	• The Security Enablers Provider generates the adequate list of enablers • The translation of Interpreter Policy is done with success • The security orchestrator selects the enabler with success • The security enablers provider provides the enabler configuration with success
Expected completion	Month 18 – June of 2018
KPI(s)	• The security plugin enabler have been generated and sent to the orchestrator to cope with the attack
Fail criteria	• No security enabler has been selected

3.6.2.5 TC_ MEC.3.6 Enforcement

The following sub test case implements the enforcement. The security orchestrator module advertises the control domain that will deploy the accurate enabler to stop the DoS attack.

Figure 31 Sequence diagram for TC_MEC3.6 – Enforcement

Table 25. Test case TC_MEC.3.6 – Enforcement

TC_MEC.3.6	Enforcement
Preconditions	• The security plugin enabler have been generated and sent to the security orchestrator to cope with the attack
Components	• Security Orchestrator • VNF controller OSM • SDN Controller • SDN switch OVS
Execution	• The SDN Controller steer the IPv6 traffic from the attacker to the device through the OVS-Firewall.

Expected results	• The selected plugin (the OVS firewall) will be deployed • The attacker's traffic will be rerouted through the OVS-Firewall
Expected completion	Month 19 – July of 2018
KPI(s)	• The security plugin enabler is deployed and the attack is mitigated
Fail criteria	• No enabler has been deployed • The deployed enabler does not stop the attack

3.6.3 Integration Test-Case

This subsection shows the test-case required to evaluate the full integration of all components of ANASTACIA framework involved in the use case MEC.3 of "DoS/DDoS Attacks using Smart Cameras and IoT Devices". The following figure shows the sequence diagram and the exchanged messages between the different components of ANASTACIA framework. The diagram shows the components grouped in the different modules defined in ANASTACIA architecture such as IoT Infrastructure, Monitoring Module, Reaction Module, Seal Manager, Orchestration Plane and Control Domain.

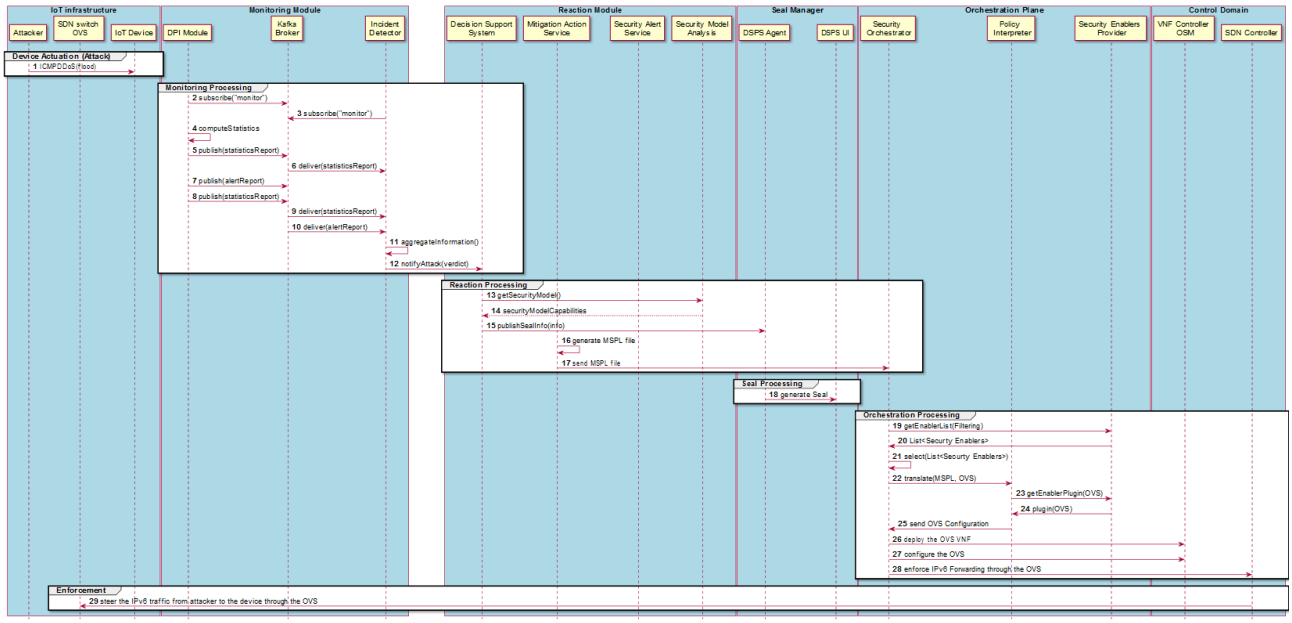

Figure 32 Sequence diagram for TC_MEC3.7 – Full MEC.3 test case

Table 26. Test case TC_MEC.3.7– Full MEC.3 test case

TC_MEC.3.7	Full MEC.3 test case
Preconditions	• Preconditions from TC_MEC.3.1 to TC_MEC.3.6
Components	• Corrupted IoT device • SDN Switch • Attack Target • Kafka Broker • Incident Detector • Decision Support System

	• Mitigation Action Service • DSPS Agent • DSPS User-Interface UI • Security Orchestrator • Policy Interpreter • Security Enablers Provider • IoT Controller • VNF Controller • SDN Controller
Execution	• To follow all steps ordered from TC_MEC.3.1 to TC_MEC.3.6
Expected results	• The completed integration is successful.
Expected completion	Month 19 – July of 2018
KPI(s)	• Detect the attack • Select the reaction with success • Select the accurate enabler to cope with the attack • Deploy the enabler (security plugin) • Mitigate the attack
Fail criteria	• Fail criteria mentioned from TC_MEC.3.1 to TC_MEC.3.6

3.7 END-USER QUESTIONNAIRE

This chapter provides generic user questionnaire that will be used for each test case to validate results outcome. Each question is rated in accordance to grading scale already presented in D1.2.

1. Very Low – fully disagree,
2. Low – partially disagree,
3. Medium – neutral,
4. High – partially agree,
5. Very High – fully agree.

The questionnaire has been dived into two parts. Frist will contain generic questions that have been prepared in accordance to the requirements defined in D1.2. In addition few more questions were added to map the measured KPIs with the end-users feedback. Second series of questions will be created separately against all integration test cases defined in sections above in order to better reflect specific test conditions and features that given scenario is validating. Table 27 represents generic questions for each of the integration test cases.

Table 27. Generic test case questionnaire

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Are you agreed that ANASTACIA framework is easy to use?	
2	Are you agreed that ANASTACIA framework is intuitive user interfaces?	
3	Are you agreed that ANASTACIA framework provides real-time feedback	
4	Level of test case configurability (attack generation, policies, reaction solution).	
5	Are you agreed that ANASTACIA framework provides automations reaction to threats?	
6	Are you agreed with the response time of monitoring module?	
7	Are you agreed with the response time of reaction module?	
8	Are you agreed with the response time of seal module?	
9	Are you agreed with the response time of orchestration module?	
10	Are you agreed with the response time of enforcement module?	

In addition to graded questions each test case will provide to users one open question that will help ANASTACIA consortium gather all additional information about each test case. It is as follows:

Could you provide additional feedback and observations from running test case?

The feedback provided in this question will enable ANASTACIA project partners to improve SW components and test process. User observations are very valuable as they provide independent information about test cases which should help improve ANASTACIA framework and uncover potential new features required in new iteration of ANASTACIA system.

3.7.1 Questionnaire for TC_BMS.2.9

Questions built around TC_BMS.2.9 are focused around checking each step of test case process.

Table 28 encloses all user questions related to full test case of BMS.2 scenario.

Table 28. Questionnaire for TC_BMS.2.9

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Has new device authentication was successful and device was granted network connectivity?	
2	Is unauthorized device authentication failed as expected?	
3	Was monitoring management sent notification about unauthorized access?	
4	Is seal Management component send notification to DSPS Agent?	
5	Was reaction module generated MSPL file for Security Orchestrator component?	
6	How successful Orchestration Management phase was in TC_BMS.2.9?	
7	How successful Enforcement Management phase was in TC_BMS2.9?	

3.7.2 Questionnaire for TC_BMS.3.5

Table 29. Questionnaire for TC_BMS.3.5

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Has subscription to DPI Module was completed?	
2	Has each loop iteration received statisticsReport?	
3	During LegitSQL phase has the attack been successful?	
4	Is Incident Detector correctly recognized iSQL attacks?	
5	Are security alerts being delivered to DSPS Agent?	
6	During iSQL attack has security seal been changed in DSPS UI?	

3.7.3 Questionnaire for TC_BMS.3.6

Table 30. Questionnaire for TC_BMS.3.6

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	All questions from TC_BMS.3.5	
2	Has the attack been mitigated by Mitigation Action Service?	

3.7.4 Questionnaire for TC_BMS.4.4

Table 31. Questionnaire for TC_BMS.4.4

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Has the attack been detected by UTRC Data Analysis module?	
2	Has security alerts been published to Security Alert Service and are visible?	
3	Has attack mitigation been completed by Orchestration Service?	

4	Is Security Seal level been changed during attack and revert after mitigation action?	
5	Are all components involved in TC_BMS.4.4 reported test events into Unit Test TC_BMS4.4 log?	

3.7.5 Questionnaire for TC_MEC.3.7

Table 32. Questionnaire for TC_MEC.3.7

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Has attack effects been observed on ANASTACIA IoT infrastructure?	
2	Is attack been detected by monitoring ANASTACIA components?	
3	Has ANASTACIA Reaction Module applied attack mitigation?	
4	Are security alerts visible?	
5	Is security and privacy seal been changed during attack?	
6	Is security and privacy seal been restored after mitigation actions been performed by ANASTACIA framework?	

4 SUMMARY AND FUTURE WORK

This document focus was about providing ground work on how to execute testing, validation and evaluation of ANASTACIA use cases defined in D1.2. The consortium preselected for demonstration four use cases that will be prepared by AALTO, ATOS, OdinS, MMT, UTRCI and THALES. Initial work provided in D6.1 was guiding element during work on this document.

The document described in chapter 2 generic methodology on how to measure KPIs during test cases validation. The section looked at potential alternatives that can be implemented in the future in Y3 of ANASTACIA project for this purpose.
All test cases were presented in chapter 3 where project partners defined test cases and provided insight on how they will be executed within ANASTACIA framework. They are following:

- TC_BMS.2 – Insider Attack on the Fire Suppression System – will also help test SEP infrastructure and monitoring components using OdinS IoT Broker,
- TC_BMS.3 – Remote Attack on the Building Energy Microgrid – this scenario will showcase MMT tool set integration with ANASTACIA framework,
- TC_BMS.4 – Cascade Attack on a Megatall Building – demonstrate detection part of UTRC Data Analysis component and integration with reaction part of ANASTACIA framework,
- TC_MEC.3 – DoS/DDoS Attacks using Smart Cameras and IoT Devices – provide insight on how SDN/VNF orchestration developed by AALTO and THALES can be used to react to active threads running in SEP.

Last part of the chapter provides user questionnaire that will be used for test case evaluation and results validation. Work completed in this document will become basis for T.6.3 and deliverable D6.3.

All information about test cases provides good test coverage of components used in ANASTACIA framework. After completing work designed in D6.3 the consortium will have good understanding how components are performing, what is the level of integration among them and what needs to be improved on next iteration of ANASTACIA framework implementation.

To summarize D6.2 provides definition of test cases that will be evaluated and validated on next step of WP6 work by ANASTACIA project partners. Evaluation and validation of test cases defined in this deliverable will give first insight to ANASTACIA framework test approach and help identify challenges, gaps that will be addressed on next iteration of ANASTACIA system implementation. Questionnaires from chapter 3.7 will help acquire user feedback on first iteration of ANASTACIA platform. Users opinions gathered in this iteration of WP6 work will be used as input to refined test cases definition, implementation as well as their evaluation and validation in final version of ANASTACIA framework that will be demonstrated in M36.

5 REFERENCES

1. Docker - https://www.docker.com/
2. Consul service – https://www.consul.io/docs/agent/services.html
3. Logstash distributed logging – https://www.elastic.co/products/logstash

6 APPENDIX

6.1 EXAMPLE OF TEST EVENT MESSAGE FOR EXTERNAL KPI REPORT


```
{
   "component":"UTRC Data Analysis",
   "test_case":"TC_BMS_4.3",
   "ts":"2018-04-26 9:00",
   "triggers":[
      {
         "id":"05",
         "name":"attack detected"
      }
   ],
   "actions":[
      {
         "name":"Verdict sent"
      }
   ],
   "kpis":{
      "accuracy":"96.18%",
      "reaction_time":"0.34"
   }
}
```