ANASTACIA

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

D6.3

Initial End-user Validation and Evaluation Report

This deliverable reports the validation and evaluation of the ANASTACIA integrated framework within the use cases implemented. After the validation of the use cases, the end users will evaluate the results and provide their feedback regarding optimizations

Distribution level	CO
Contractual date	31.10.2018 [M22]
Delivery date	17.12.2018 [M24]
WP / Task	WP6
WP Leader	OdinS
Authors	Rafael (OdinS), Dan (OdinS), Stefano (SOFT), Diego (MONT), Ivan Vaccari, Enrico Cambiaso, Silvia Scaglione, Ivan Vaccari, Elisabetta Punta (CNR), Ruben Trapeiro (ATOS), Giannis Ledakis (UBITECH), Alejandro Molina (UMU), Matteo Filipponi (DG), Piotr Sobonski (UTRC), Miloud Magaad (AALTO)
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 4
1 Introduction 5
1.1 Aims of the document 5
1.2 Applicable and reference documents 5
1.3 Revision History 5
1.4 Acronyms and Definitions 6
2 Contextualization 8
2.1 Scope 8
2.2 End-Users 9
2.2.1 Interviews 9
3 End-User Evaluation of ANASTACIA framework validated according to Use Cases 11
3.1 Use-Case: BMS.2 - Insider Attack on the Fire Suppression System 11
3.1.1 Use case description and benefits from ANASTACIA framework 11
3.1.2 Results of Test-Cases with KPIs and Data Captured 23
3.2 Use-Case BMS.3: Remote Attack on the Building Energy Microgrid 28
3.2.1 Use case description + Benefits from ANASTACIA framework 28
3.2.2 Results of Test-Cases with KPIs and Data Captured 34
3.3 Use-Case BMS.4: Cascade Attack on a Megatall Building 38
3.3.1 Use case description + Benefits from ANASTACIA framework 38
3.3.2 Results of Test-Cases with KPIs and Data Captured 48
3.4 Use Case MEC.3: DoS/DDoS Attacks using IoT Devices 52
3.4.1 Use case description + Benefits from ANASTACIA framework 52
3.4.2 Results of Test-Cases with KPIs and Data Captured 59
3.5 Results of End-User Evaluation 63
Evaluation 63
3.5.1 Questionnaire 63
3.5.2 Results of End-Users Questionnaire 67
4 Conclusions and Future Works 71
5 Annex 1 – Interview Questionnaires 72

List of figures

Figure 1. Initial design of ANASTACIA architecture 8
Figure 3. Bootstrapping phase for BMS.2 12
Figure 4: MSPL policy for PANA traffic enablement for BMS.2 13
Figure 5. Bootstrapping process for BMS.2 13
Figure 6. Network authentication completed for BMS.2 14
Figure 7. ONOS SDN controller allows data traffic from authenticated IoT device for BMS.2 14
Figure 8. Attack triggering for BMS.2 15
Figure 9. Token validation in IoT device for BMS.2 15
Figure 10. Threat notification from IoT device for BMS.2 16
Figure 12: Threat pre-processing in Kafka broker for BMS.2 17
Figure 14. Risk evaluation in Verdict Decision Support System for BMS.2 18
Figure 15. Alert processing in Mitigation Action Service for BMS.2 18
Figure 16. Alert message generated by Verdict Decision Support System for BMS.2 19
Figure 17. MSPL file sent from Mitigation Action Service to Security Orchestrator for BMS.2 19
Figure 18. MSPL file generated by MAS for deploying a virtual IoT-honeynet for BMS.2 20
Figure 19. Reaction deployment for BMS.2 20
Figure 20. Cooja translation of the MSPL file for virtual honeynet by Policy Interpreter for BMS.2 21
Figure 21. VNF instantiation of Cooja honeynet for BMS.2 21
Figure 22. SDN rules for traffic forwarding towards the virtual honeynet for BMS.2 22
Figure 23. Network management inferface of SDN controller with virtual honeynet for BMS.2 22
Figure 24: Complete flow of use case BMS.3 28
Figure 25: Attack triggering for BMS.3 29
Figure 26: Attack triggering of SQL injection for BMS.3 29
Figure 27: Monitoring and Reaction Phase for BMS.3 30
Figure 28: Generated MPSL File for BMS.3 30
Figure 29: Events correlation in Incident Detector for BMS.3 31
Figure 30: MSPL generation in Mitigation Action Service for BMS.3 31
Figure 32: Reception of MSPL file in Security Orchestrator for BMS.3 33
Figure 33: VNF instantiation of Virtual Firewal for BMS.3 33
Figure 34: Filtering rules received in ONOS SDN Controller for BMS.3 33
Figure 35: Network management inferface of SDN controller with Virtual Firewall 34
Figure 36: SQL injection is blocked and the attack is mitigated for BMS.3 34
Figure 38: Temperature sensor measurements for BMS.4 39
Figure 40: Attack script and data analysis detecting attack for BMS.4 41

Figure 44 shows these two exchanged messages for SAS and MAS components 44
Figure 45: Alert message sent from VDSS to SAS component 44
Figure 47: MSPL file to mitigate IoT sensor attack with power off action for BMS.4 46
Figure 49: Configuration reception in IoT Controller to power off the IoT device..... 47
Figure 51: Executed power off action on IoT device 48
Figure 54: Attack triggering of ICMP flooding for MEC.3 54
Figure 56: Generated MPSL File for MEC.3 55
Figure 58: MSPL generation in Mitigation Action Service for MEC.3 56
Figure 59: Deployment of the Reaction for MEC.3 57
Figure 61: VNF instantiation of Virtual Firewal for MEC.3 58
Figure 62: Network management inferface of SDN controller with Virtual Firewall for MEC.3 59

PUBLIC SUMMARY

According to the DoA, the scope of this deliverable is performing the validation and evaluation of the ANASTACIA integrated framework within the Test Cases and according to the methodology defined in T6.2. The performance and the usability of the ANASTACIA test cases will be monitored and validated. After the validation of the Test Cases, the end users will evaluate the ANATACIA performance and provide their feedback regarding any optimizations and further improvements to be performed in the second cycle of ANASTACIA development.

Concretely, this document is focused on the validation of ANASTACIA framework according to four use-cases implemented (BMS.2, BMS.3, BMS.4 and MEC.3). For each use case, it is provided a detailed description of its implementation and the benefits of ANASTACIA framework. This description includes capture data (i.e. web interfaces and log consoles) from ANASTACIA components during the execution and validation of the specific use case. Moreover, key performance indicators (KPIs) of ANASTACIA components are been measured and reported according to the Test-Cases defined in the deliverable D6.2 for each use case.

Moreover, the document presents the results of the evaluation performed by 65 end-users to be meant to help the Consortium in the prioritization of functionalities and features to must be improved. To facilitate the evaluation carried out by end-users, five videos have been provided. The videos show the execution of the first release of integrated ANASTACIA framework according to 4 different use cases in 2 application domains: Building Management System (BMS) and Mobile Edge Computing (MEC). In additional, a video shows the demonstration of seal management module that will be integrated in the next cycle of ANASTACIA development. The evaluation provides end-users feedback based on a questionnaire divided into three parts: general features, specific operations of each implemented use case and related aspects of DSPS Seal management.

1 INTRODUCTION

1.1 AIMS OF THE DOCUMENT

This document provides the results of end-users evaluation and validation of the ANASTACIA framework. The main aims are:

- to describe the use cases implemented and the benefits provided by ANASTACIA framework;
- to validate the test cases defined in D6.2 to verify the integration of ANASTACIA components;
- to provide key performance indicators (KPIs) of ANASTACIA components;
- to provide the results of end-users feedback related to further improvements during the second cycle of ANASTACIA development;

1.2 APPLICABLE AND REFERENCE DOCUMENTS

This document refers to following documents:

- Grant Agreement N°731558 and annexes ("Description of Action").
- D1.2 User-centred Requirement Initial Analysis.
- D1.3 Initial Architecture Design.
- D6.1 Initial Technical integration and validation Report.
- D6.2 Initial Use cases implementation and tests Report.

1.3 REVISION HISTORY

Version	Date	Author	Description
V0.1	14/06/2018	Rafael Marin (OdinS)	ToC and Introduction
V0.2	02/10/2018	Rafael Marin (OdinS)	Description of BMS.2 + Benefits from ANASTACIA
V0.3	05/10/2018	Rafael Marin (OdinS)	Initial contents of BMS.3
V0.4	08/10/2018	Rafael Marin (OdinS)	Initial contents of BMS.4
V0.5	15/10/2018	Diego Rivera (MONT)	Description of MEC.3 + Benefits from ANASTACIA
V0.6	17/10/2018	Rafael Marin (OdinS) Stefano Bianchi (SOFT)	Questionnaire and Contextualization
V0.7	23/10/2018	Piotr Sobonski (UTRC)	Description of BMS.4 + Benefits from ANASTACIA
V0.8	29/10/2018	Rafael Marin (OdinS)	Description of BMS.3 + Benefits from ANASTACIA
V0.9	31/10/2018	Rafael Marin (OdinS)	Revision of test-cases and KPIs for BMS.2 and BMS.3
V0.10	7/11/2018	Rafael Marin (OdinS)	Revision of test-cases and KPIs for BMS.4

V0.11	14/11/2018	Rafael Marin (OdinS)	Revision of test-cases and KPIs for MEC.3
V0.12	16/11/2018	I.Vaccari (CNR), E.Cambiaso (CNR),	First review phase
V0.13	12/12/2018	Rafael Marin (OdinS)	Results of End-users Evaluation in Section 3.5
V0.14	13/12/2018	I.Vaccari (CNR), E.Cambiaso (CNR),	Second review phase

1.4 Acronyms and Definitions

Acronym	Meaning
API	Application Programming Interface
BMS	Building Management Systems
CRUD	Create/Retrieve/Update/Delete
DDoS	Distributed Denial of Service attack
DoA	Description of Action
DPO	Data Protection Officer
DSPS	Dynamic Security and Privacy Seal
FR	Functional Requirement
GDPR	General Data Protection Regulation
HSPL	High-level Security Policy Language
IoT	Internet of Things
Kafka	Message broker used in ANASTACIA framework to enable distributed communication between components
KPIs	Key Performance Indicators
MAS	Mitigation Action Service
MEC	Mobile Edge Computing / Multi-access Edge Computing
MSPL	Medium-level Security Policy Language
MVP	Minimum Viable Product
NFR	Non Functional Requirement

NFV	Network Function Virtualization
OVS	Open Virtual Switch
PI	Policy Interpreter
PR	Privacy Requirement
REST	Representational State Transfer
SAS	Security Alert Service
SEP	Security Enforcement Plane
SDN	Software Defined Network
SO	Security Orchestrator
TRL	Technology Readiness Level
TC	Test Case
UC	Use Case
UML	Unified Modeling Language
VDSS	Verdict Decision Support System
VID	Virtualized Infrastructure Domain
VNF	Virtual Network Function
XML	eXtensible Markup Language

2 CONTEXTUALIZATION

2.1 SCOPE

This document reports the validation and evaluation of the ANASTACIA framework that is able to take autonomous decisions through the use of new networking technologies such as Software Defined Networking (SDN) and Network Function Virtualisation (NFV) and intelligent and dynamic security enforcement and monitoring methodologies and tools.

According to the initial architecture design as shown in Figure 1, ANASTACIA is envisioned as a framework integrated on top of an IoT infrastructure where IoT devices, physical and virtual network elements interact in the Data Plane. On top of that, the Control Plane manages the computing, storage, and networking resources in the Data Plane by leveraging SDN controllers, NFV orchestration platforms, and IoT controllers. The Autonomic Plane includes the components that provide the ANASTACIA framework with its intelligence and dynamic behaviour. The Autonomic Plane can be divided into several sub-planes, which carry out specific activities:

- **Seal Management Plane** provides users with a real-time indicator of the overall security level.
- **User Plane** includes interfaces, applications and tools that help system administrators to manage the IoT platform through the ANASTACIA framework. For example, at this plane system admins are able to edit the security policies that govern the underlying IoT platform.
- **Monitoring plane**. In this process the monitoring information is extracted from the devices through monitoring agents and according to the security controls interpreted from the security policy. In this activity, the monitoring data is filtered and aggregated to carry out its analysis and the detection of anomalies and threats.
- **Reaction plane**. In this process the detected anomalies are evaluated to design counter measures in order to mitigate the effects of attacks and potential threats.
- **Security orchestration plane**. Once the mitigation action has been designed, it is necessary to enforce the controls specified within the security policies. To orchestrate the selected IoT/SDN/NFV-based security enablers, appropriate interactions with the relevant management modules are required in the **Security enforcement plane**.

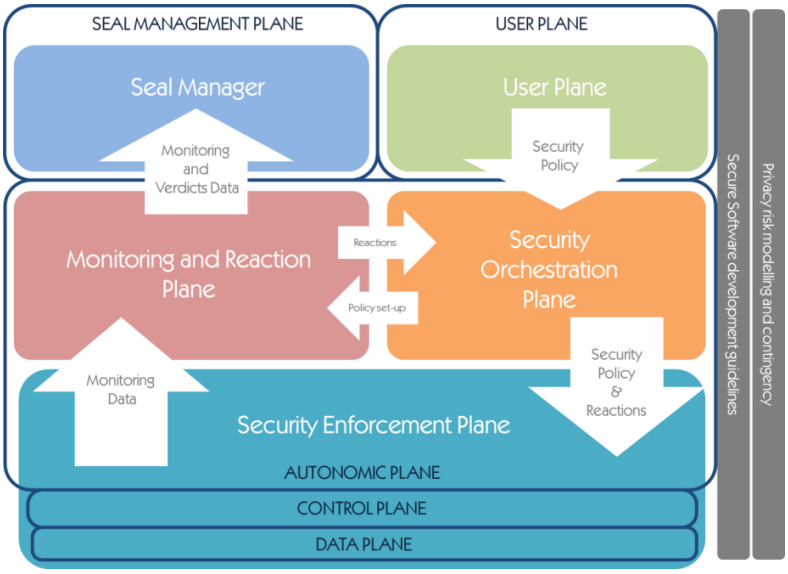


Figure 1. Initial design of ANASTACIA architecture

2.2 END-USERS

As indicated in the DoA, part of the activities of task T6.3 included the evaluation of ANASTACIA framework with potential end-users and stakeholders. The context of use of the main services which will be included in the ANASTACIA framework potentially includes several different user categories, all coping at different levels with security and privacy issues:

- SW developers
- IoT architects/developers
- SDN architects/developers
- NFV architect/developers
- Security managers
- Solution integrators
- Chief Security Officer (CSO)
- Chief Technology Officer (CTO)
- Chief Information Officer (CIO)
- Chief Information and Security Officer (CISO)
- Mobile Edge Computing/Multi Access Edge Computing (MEC) stakeholders
- Building Management System (BMS) stakeholders
- System / Network administrators
- Security professionals/consultants
- Lawyers
- GDPR-associated actors (e.g. Data Protection Officer, Data Processor, Data Controller, etc.)

2.2.1 Interviews

In particular, the evaluation was performed by 65 different users in order to gather meaningful feedback as shown in the following list.

1. Alejandro Perez Mendez (Security Researcher, JISC UK)
2. Christos Iliou (Privacy and Security Researcher, CERTH)
3. José Luis Hernández Ramos (Privacy and Security Researcher, JRC)
4. Ana María García López (Solution integrator in BMS and Energy-efficiency, ENGER)
5. Sotiris Koussouris (IoT Systems Manager, Suite5 Ltd)
6. Daniela Mikeli (IoT developer, SingularLogic)
7. Ramon Sanchez Iborra (IoT researcher, Technische Universität Berlin)
8. Djamel Eddine Bensalem (NFV researcher, Aalto University)
9. Diego Leonel Cadette Dutra (Professor of Mobile Systems, Federal University of Rio de Janeiro (UFRJ), Brazil)
10. Ibrahim Afolabi (SDN developer, Aalto University)
11. Oscar Novo (Senior security researcher, Ericsson)
12. Thu Le Pham (IoT developer, UTRC)
13. Teo Virta (IoT researcher, Ericsson)
14. Tejaswini Darure (IoT SW developer, UTRC)
15. WION Adrien (Security researcher, Telecom ParisTech)
16. Ehsan Ebrahimi (IoT researcher, Thales)
17. Agathe Blaise (IoT researcher, LIP6)
18. Kevin Phemius (IoT researcher, Thales SIG GTS)
19. Naresh Yarlapatiganesh (IoT developer, UTRC)
20. Walid BOUMEZER (SW developer, ComNet Department, Aalto University)
21. Christian Melchiorre (IoT Research Team Leader, Softeco Sismat)
22. Wiktor Buclek (IoT developer, UTRC)

23. Mauro Porrati (SonicWall CSSP - CSSA, Senior ICT Solution Architect, Softeco Sismat)
24. Marcello Torchio (IoT developer, UTRC)
25. Pedro J. Fernández (SDN/NFV researcher, University of Murcia)
26. RUBEN RICART-SANCHEZ (SDN/NFV researcher, UNIVERSITY OF THE WEST OF SCOTLAND)
27. Olivia Döll (IoT security researcher, Archimede Solutions)
28. Juan Antonio Martinez Navarro (IoT security researcher, Odin Solutions)
29. Emilia Evéquoz (IoT security researcher, Archimede Solutions SARL)
30. Alejandro González Jiménez (SDN developer, University of Murcia)
31. Zakaria laaroussi (IoT security researcher, Ericsson)
32. Dan Garcia (IoT security researcher, Odin Solutions)
33. Pedro Gonzalez (SW developer, University of Murcia)
34. Kazi Wali Ullah (IoT security researcher, Ericsson)
35. Jorge Gallego (IoT SW developer, University of Murcia)
36. Marco Boero (Head of Research & Innovation, Softeco Sismat)
37. Ana Hermosilla (Big Data researcher of Building Management System, University of Murcia)
38. Mai Hoang Long (SW developer, Montimage)
39. Javier Ramirez de la Pinta (IoT researcher, ATOS)
40. MOSCA GIOVANNI (Research & Innovation Business Development, Softeco Sismat)
41. Huu Nghia Nguyen (IoT researcher, Montimage)
42. Antonio Álvarez (IoT projects manager, ATOS)
43. GIANNI VIANO (Head of Technology Solutions, Softeco Sismat)
44. Ana Cavalli (IoT Security researcher, Montimage)
45. Adrián Arroyo Pérez (IoT developer, ATOS Research and Innovation)
46. JAVIER PASTOR GALINDO (SW developer, University of Murcia)
47. Pasquale Annicchino (IoT Security researcher, Archimede Solutions)
48. Vinh Hoa La (SW developer, Montimage)
49. Jeremy Davis (SW developer, Device Gateway)
50. Beatriz Gallego-Nicasio Crespo (IoT developer, ATOS)
51. Michael Hazan (IoT Security researcher, Device Gateway)
52. Anna Brekine (Security researcher, Mandat International)
53. Niki Kontoe (Software Engineer, UBITECH)
54. Francesco Malandrino (Software developer, CNR)
55. Edgardo Montesdeoca (IoT researcher, Montimage)
56. David Rojo Antona (Security research engineer, ATOS)
57. Luong Nguyen (IoT researcher, Montimage)
58. Gustavo Gonzalez (Security researcher, ATOS Research & Innovation)
59. Bernard Santoux (IoT Security researcher, Mandat International)
60. Pascal Bisson (IoT researcher, Thales SIX GTS France)
61. Mario Ferdinando Faiella (SW developer, ATOS)
62. Danae Vergeti (Software Engineer, UBITECH)
63. Francesco De Rose (IoT researcher, Device Gateway)
64. Hristo Koshutanski (IoT researcher, Atos Research & Innovation)
65. Petros Mantos (IoT Developer, UBITECH)

The results of the feedback collected from these end-users are presented in Section 3.5 to be considered in the second cycle of ANASTACIA development. Moreover, a table with all answers provided by end-users are included in the Annexe I.

3 End-User Evaluation of ANASTACIA framework validated according to Use Cases

This section reports the validation and end-user evaluation of ANASTACIA framework corresponding to the four use-cases implemented. For each use case, it is provided a detailed description of its implementation and the benefits of ANASTACIA framework. The description subsection includes capture data (i.e. web interfaces and log consoles) from ANASTACIA components during the execution and validation of the specific use case. Moreover, key performance indicators (KPIs) of ANASTACIA components are been measured and reported according to the test-cases defined in the deliverable D6.2 for each use case.

3.1 Use-Case: BMS.2 - Insider Attack on the Fire Suppression System

3.1.1 Use case description and benefits from ANASTACIA framework

The insider attacker tries to activate remotely a fire alarm connected to an IoT device. To execute the attack, the attacker sends a COAP message with the query of fire alarm activation towards the IPv6 address of the IoT device. Figure 2 shows the main operations performed by ANASTACIA components such as allowing IoT-device bootstrapping, secure bootstrapping, attack triggering, monitoring and reaction. The colours of lines indicate the different operations done. Moreover, these operations will be described in detail below.

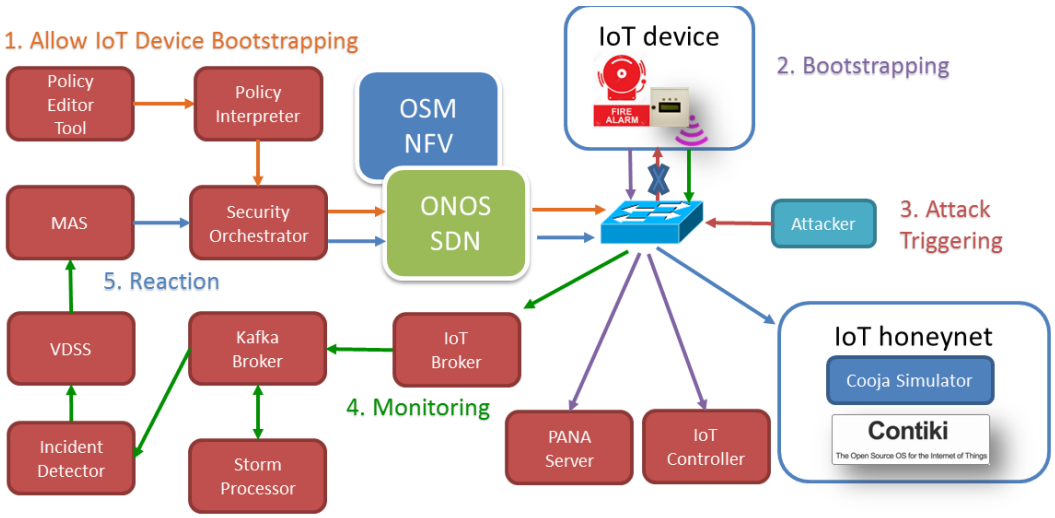


Figure 2. Use case deployment for BMS.2

Thanks to the benefits of ANASTACIA architecture, an insider attacks cannot generate a fire alarm activation to force the evacuation of the building. Any actuation query must be authorized by the requested IoT device using a distributed capability-based technique called DCapBAC. This security-by-design property acts as monitoring component in IoT-devices to enable the attack detection and alert notification, as demonstrated in this use case.

Another advantage of the ANASTACIA architecture is the novel mitigation action based on the creation of a virtual honeynet emulating an IoT network to redirect the attack traffic for further analysis of the threat and avoid any damage in the real infrastructure of smart building.

The complete flow of BMS.2 use case is divided into 4 phases:

1. Allow network authentication and device registration based on IoT bootstrapping technique and SDN traffic management.
2. Insider attack sends a COAP message with an unauthorized actuation to a fire alarm wired to IoT device that uses distributed capability-based access control based on DCapBAC protocol to monitor and detect unauthorized actuation.
3. Threat monitoring and reaction decision based on event correlation and mitigation actions that generates MSPL file with a XML-based language to define the reaction countermeasures.
4. Deploy reaction countermeasures by security-orchestrator coordinating NFV controller and SDN controller:
   - IoT honeynet deployment.
   - Transparent traffic forwarding.

In the next subsections, these different stages of the use case are described in detail.

3.1.1.1 Complete flow of use case BMS.2

3.1.1.1.1 Bootstrapping and Registration

The first stage includes the bootstrapping and registration of an IoT device in the ANASTACIA framework to achieve network connectivity. Following the interactions shown in Figure 3, the network administrator must use the policy editor tool to define a policy rule for enabling the registration and deployment of new IoT devices based on bootstrapping PANA protocol. The policy editor provides a HSPL file with PANA traffic enablement to the policy interpreter translating the HSPL file (High Security Policy Level) to MSPL file (Medium Security Policy Level) shown in Figure 4. This MSPL file is sent to the Security Orchestrator in order to enforce the PANA traffic enablement in the ONOS SDN controller for the IoT network. Once the PANA traffic is allowed, any IoT device can start the bootstrapping process to be authenticated in the IoT network by the PANA Agent that informs to the IoT controller sending the device registration to the Security Orchestrator in order to request the data traffic enablement in SDN network for the new IoT device.

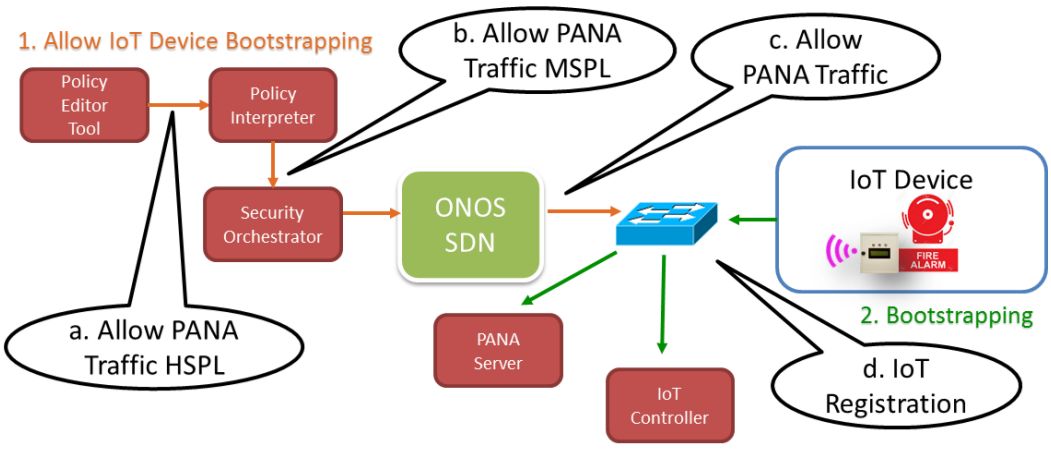


Figure 3. Bootstrapping phase for BMS.2



Figure 4: MSPL policy for PANA traffic enablement for BMS.2

The next figures show the data captured (logs and screens) of the ANASTACIA components (IoT device, PANA agent, IoT controller and Security Orchestrator) to perform the 3 main interactions for the network registration:

1. Figure 5 shows that IoT-device (at the middle console) starts the bootstrapping process with PANA Agent (in the right console).

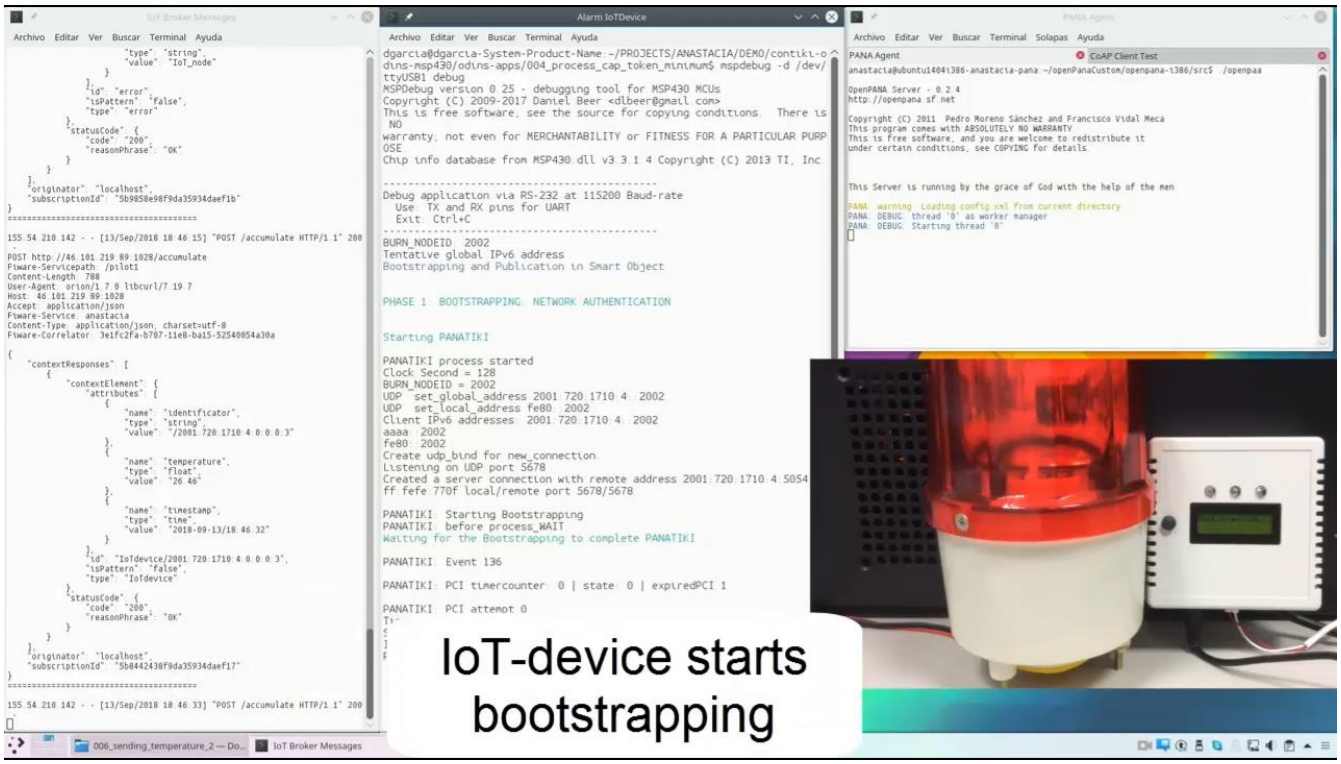


Figure 5. Bootstrapping process for BMS.2

2. Figure 6 shows that IoT-device (at the middle console) finished the bootstrapping process with PANA Agent (at the right console) sending a registration message to the IoT controller.

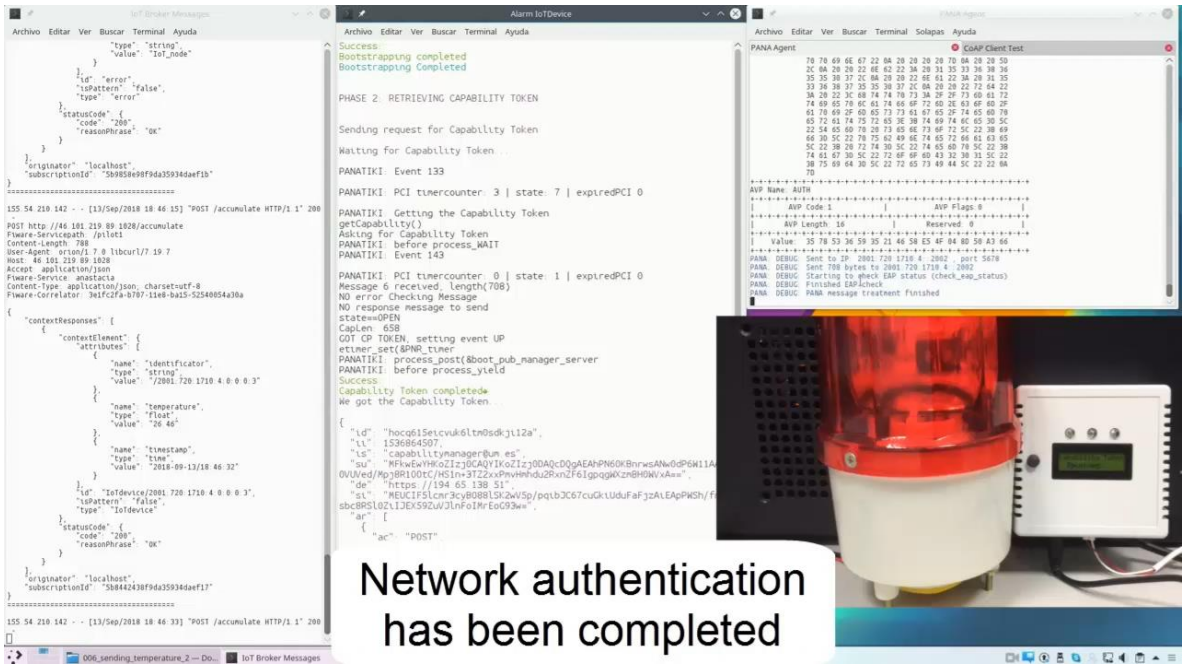


Figure 6. Network authentication completed for BMS.2

3. Figure 7 shows that the Security Orchestrator (at the top-left console) receives the registration from IoT controller and generates a command to ONOS SDN controller (at the right side) to allow that the authenticated IoT device sends data traffic. For instance, ICMP messages are sending through SDN switches as shown at the bottom-left console.

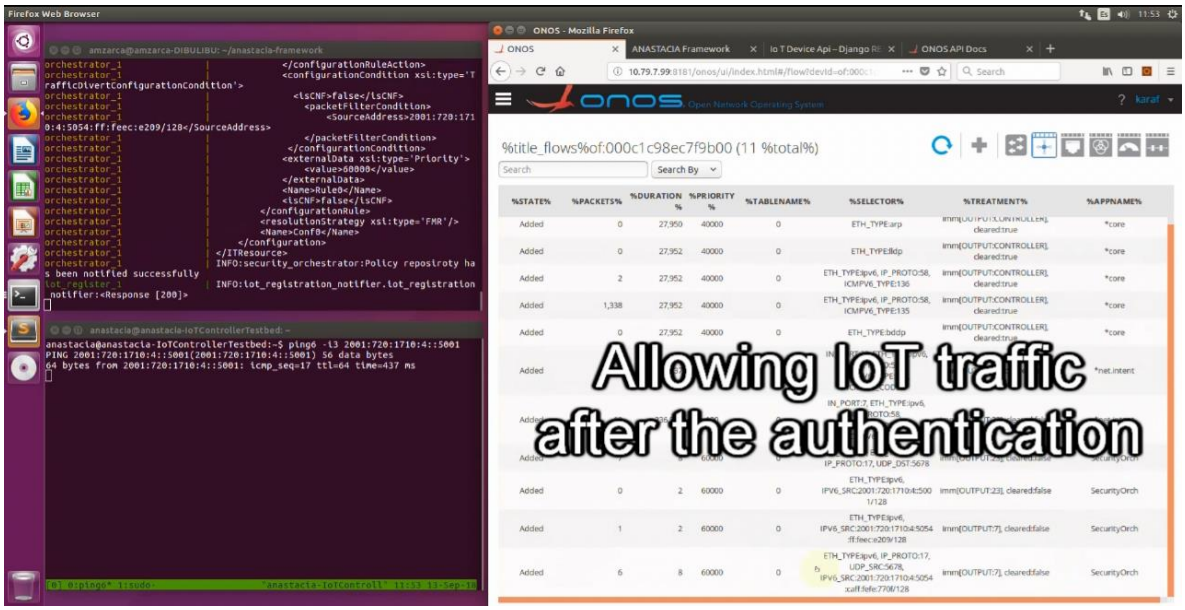


Figure 7. ONOS SDN controller allows data traffic from authenticated IoT device for BMS.2

3.1.1.1.2 Attack triggering

The second stage includes the execution of an attack script to send a query to an IoT device to request the fire alarm activation as shown in Figure 8. First, the attacker sends a COAP message including the fire alarm

activation with an invalid capability token. Before activating the fire alarm, the device verifies that the token is invalid and then the device sends a threat notification towards the IoT broker.

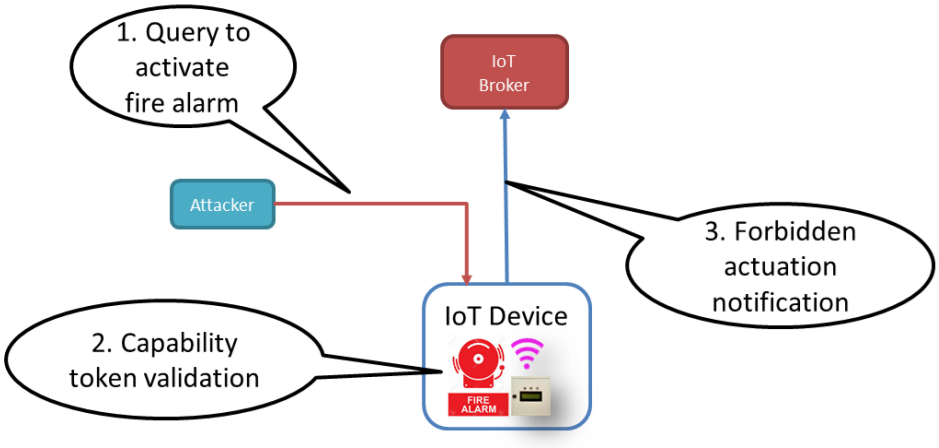


Figure 8. Attack triggering for BMS.2

The next figures show the data captured (logs and screens) from the attacker, IoT device and IoT broker to do the 2 main interactions of the threat detection:

1. Figure 9 shows in the top console in the right side that the attacker sends the COAP request with a capability token.

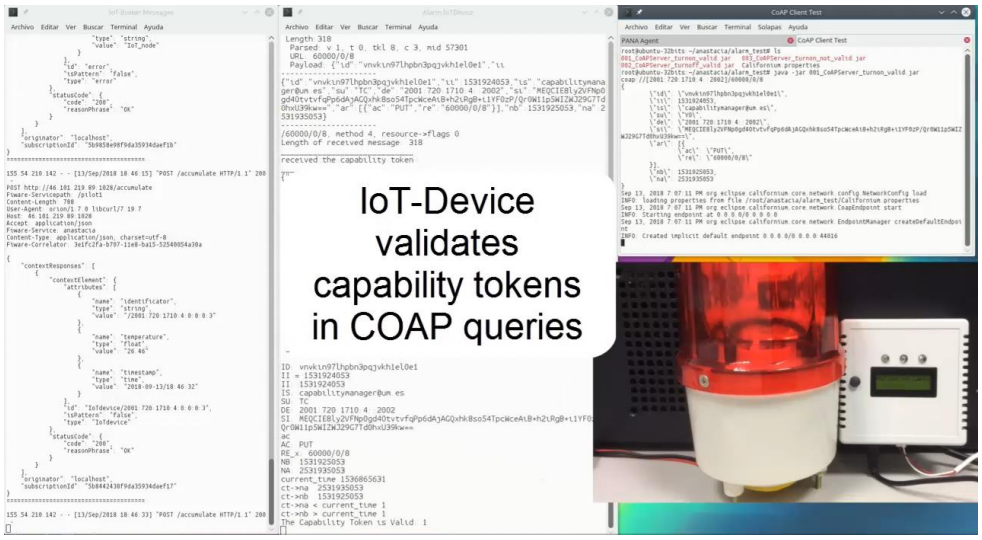


Figure 9. Token validation in IoT device for BMS.2

2. Figure 10 shows in the middle console that IoT-device receives the COAP request and detects the invalid token and notifies the attack to the IoT-broker.

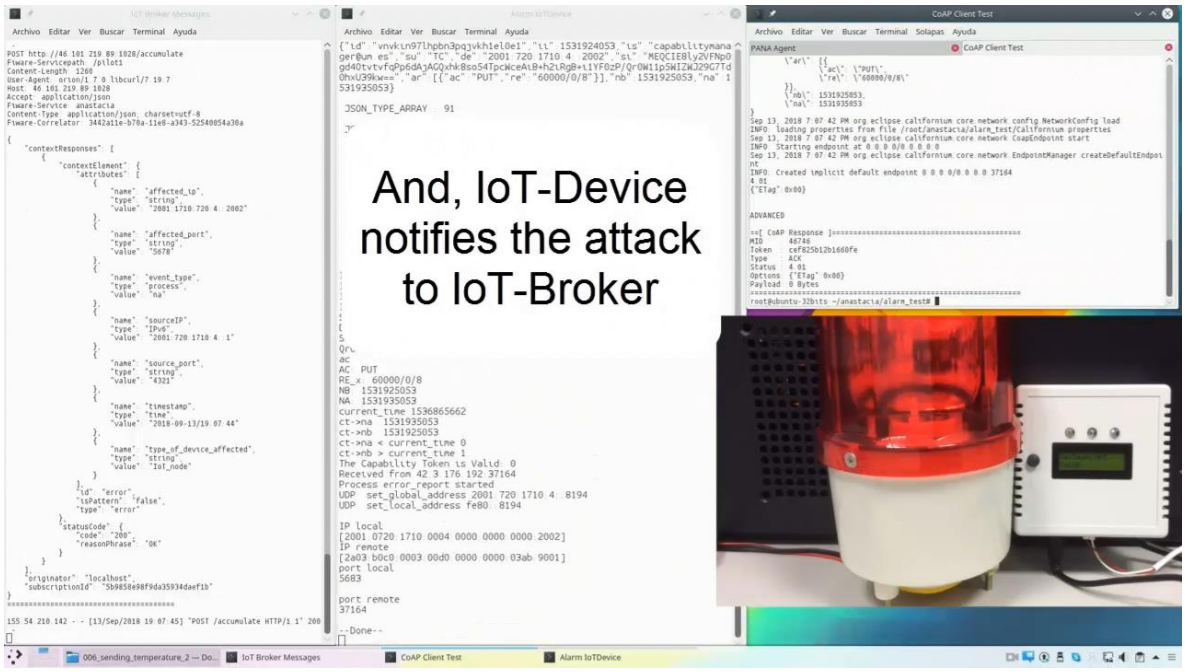


Figure 10. Threat notification from IoT device for BMS.2

3.1.1.1.3 Monitoring and reaction

The third stage is composed of the monitoring and reaction modules to process the threat notification and generate the mitigation actions. Figure 11 shows the interactions on this phase, which starts right after the last one (i.e. attack triggering) has finished. First, the threat notification of forbidden device authorization is sent to Kafka broker where the reported data is aggregated and filtered. Second, these pre-processed data are sent to the Incident detector to make correlation of several events in order to feed to Verdict Decision Support System (VDSS) to generate an alert message. Third, this alert message is received by Mitigation Action Service (MAS) that decides the reaction and creates the MSPL file with the mitigation actions to be sent to the Security Orchestrator.

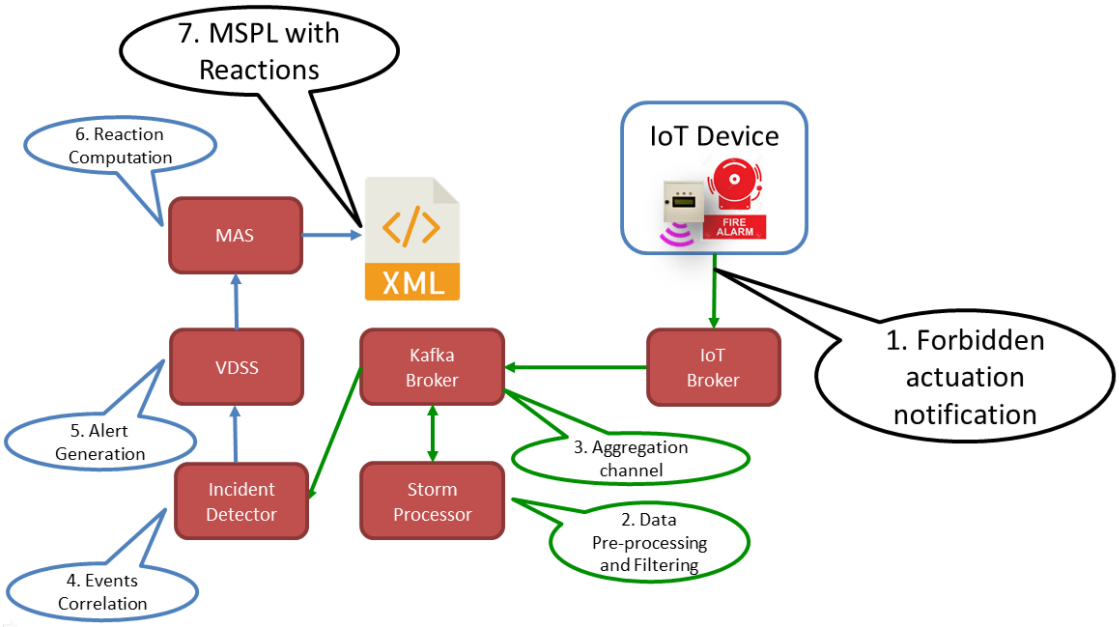


Figure 11: Monitoring and reaction for BMS.2

The next figures show the data captured (logs and screens) from the monitoring and reaction components to do the main interactions of this phase:

1. Figure 12 shows the web interface of Kafka broker with the information pre-processed about the threat description of forbidden device access.

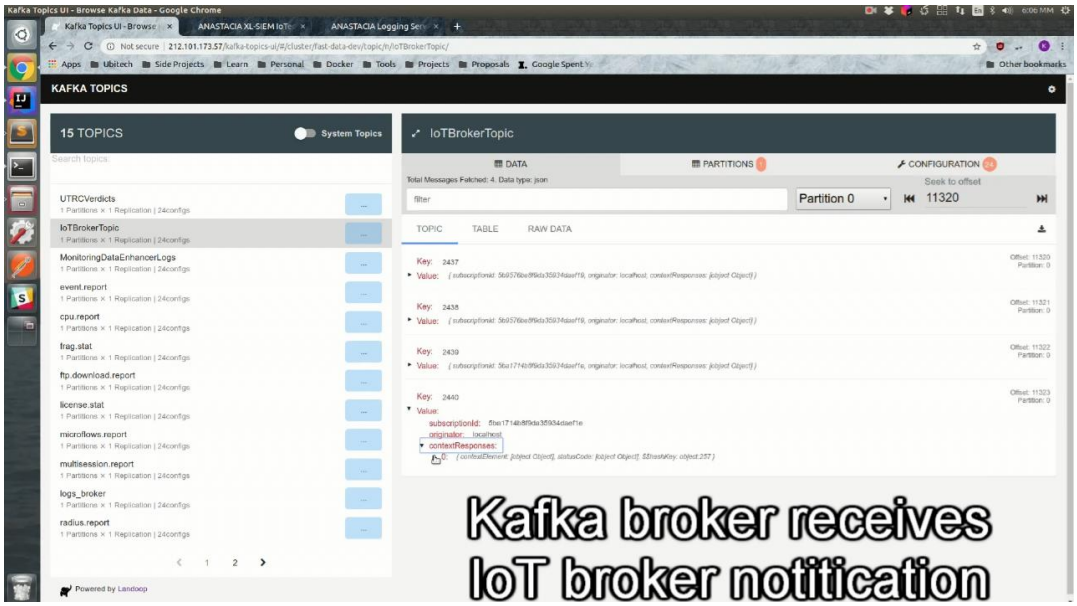


Figure 12: Threat pre-processing in Kafka broker for BMS.2

2. Figure 13 shows the web interface of the Incident Detector to correlate the threat events of "AAA Sensor – Forbidden Device Access" indicating the IPv6 addresses of attack source and victim destination.

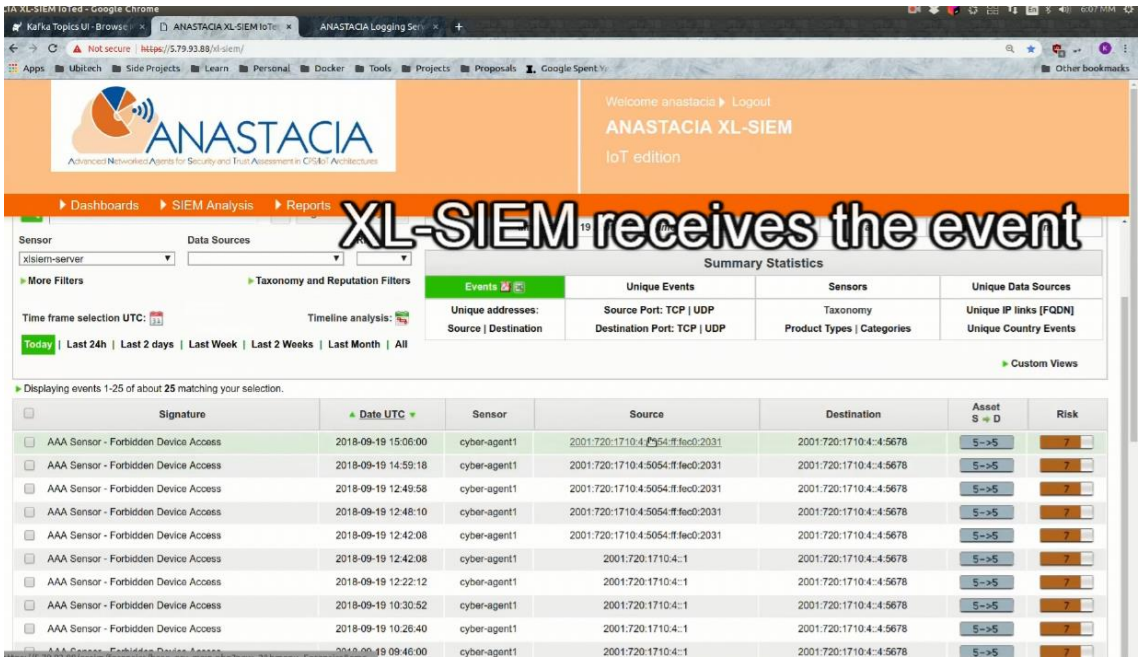


Figure 13. Events correlation in Incident Detector for BMS.2

3. Figure 14 shows the web interface of Verdict Decision Support System to evaluate the risk of the threat events before generating an alert message.

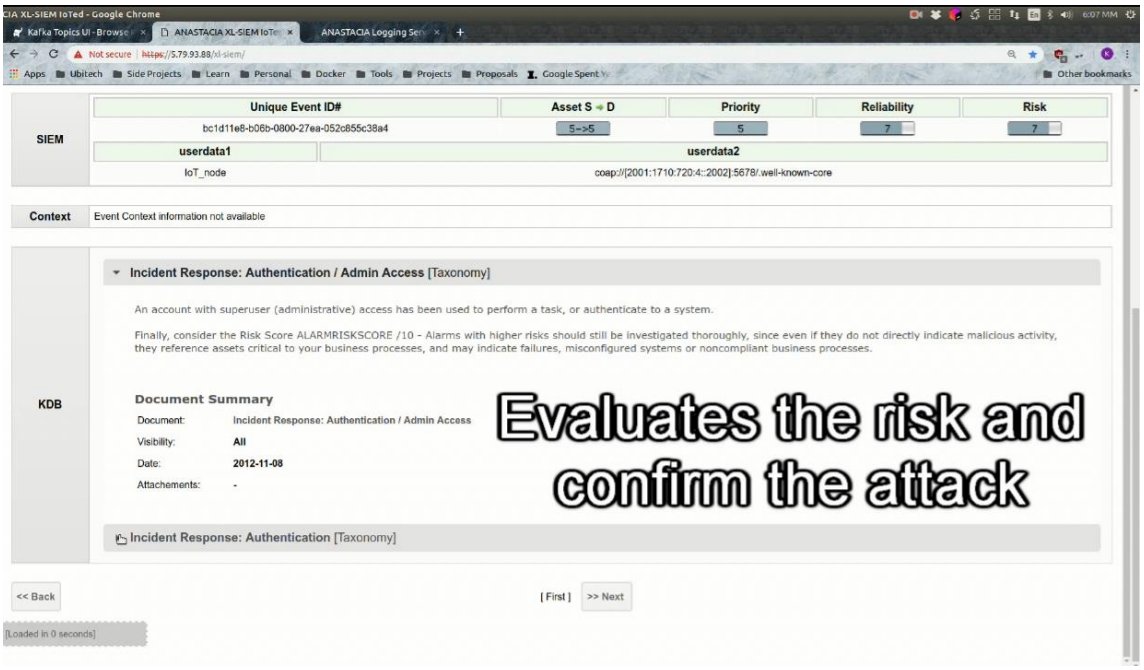


Figure 14. Risk evaluation in Verdict Decision Support System for BMS.2

4. Figure 15 shows the web interface of Mitigation Action Service where an alert message is processed in order to generates a reaction message including MSPL security policies to be sent to Security Orchestrator.

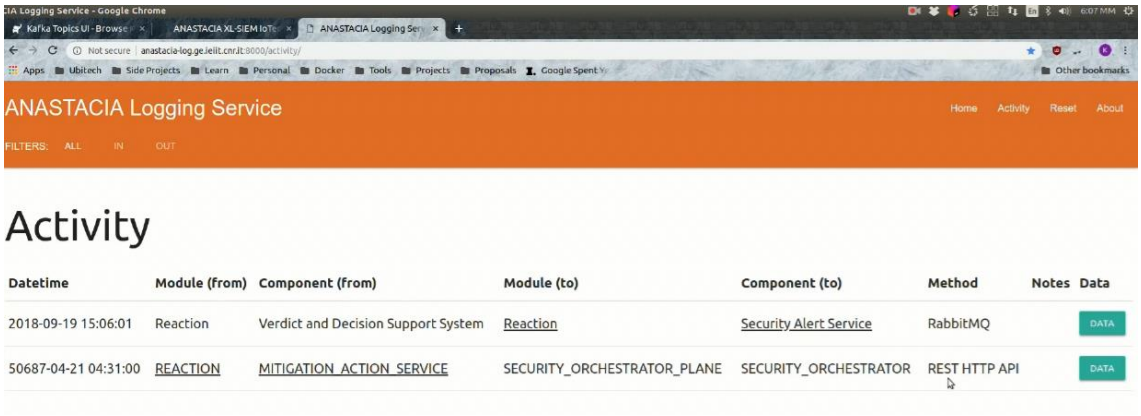


Figure 15. Alert processing in Mitigation Action Service for BMS.2

5. Figure 16 shows the alert message sent by Verdict Decision Support System to Mitigation Action Service.

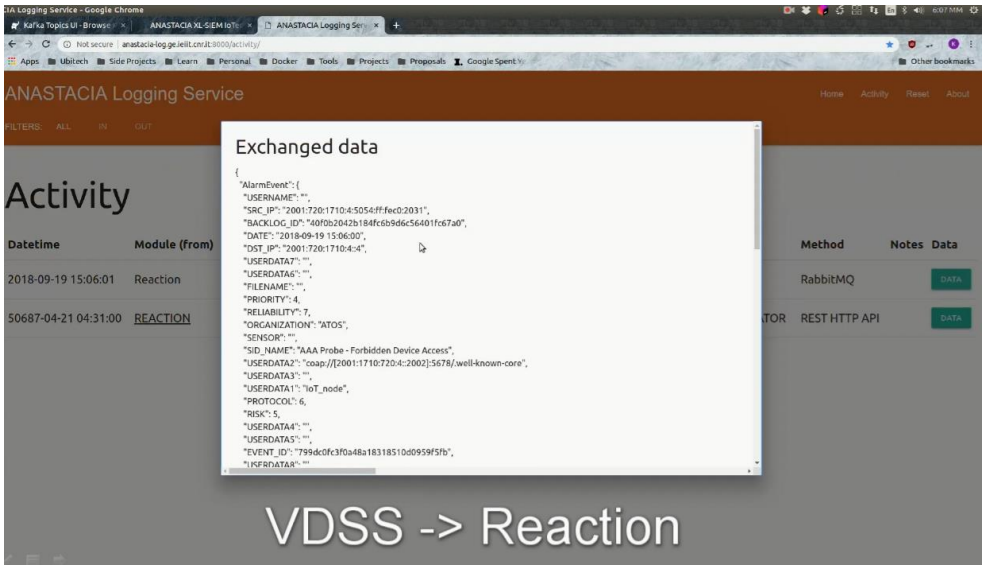


Figure 16. Alert message generated by Verdict Decision Support System for BMS.2

6. Figure 17 shows the MSPL file sent by Mitigation Action Service to Security Orchestrator.

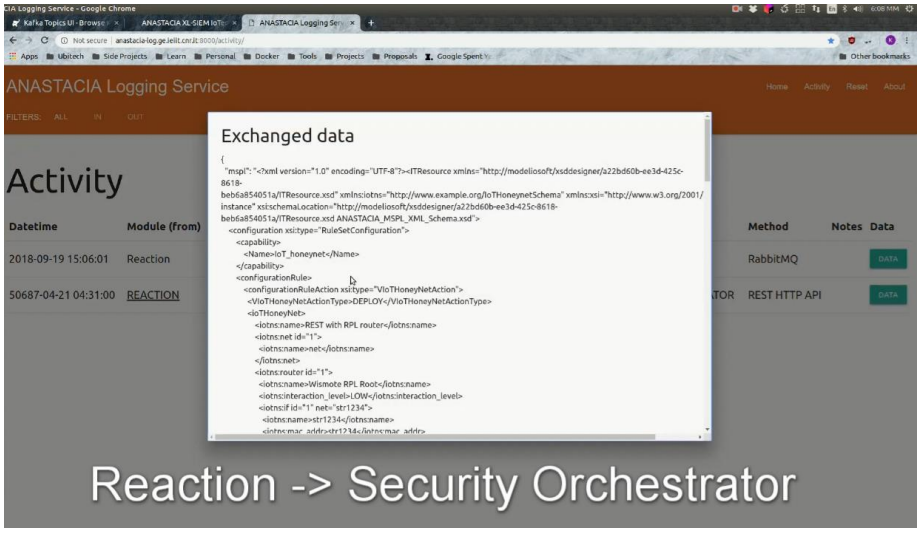


Figure 17. MSPL file sent from Mitigation Action Service to Security Orchestrator for BMS.2

7. Figure 18 shows the full description of the MSPL file generated by Mitigation Action Service to request the deployment of a virtual honeynet including an IoT router and IoT devices.

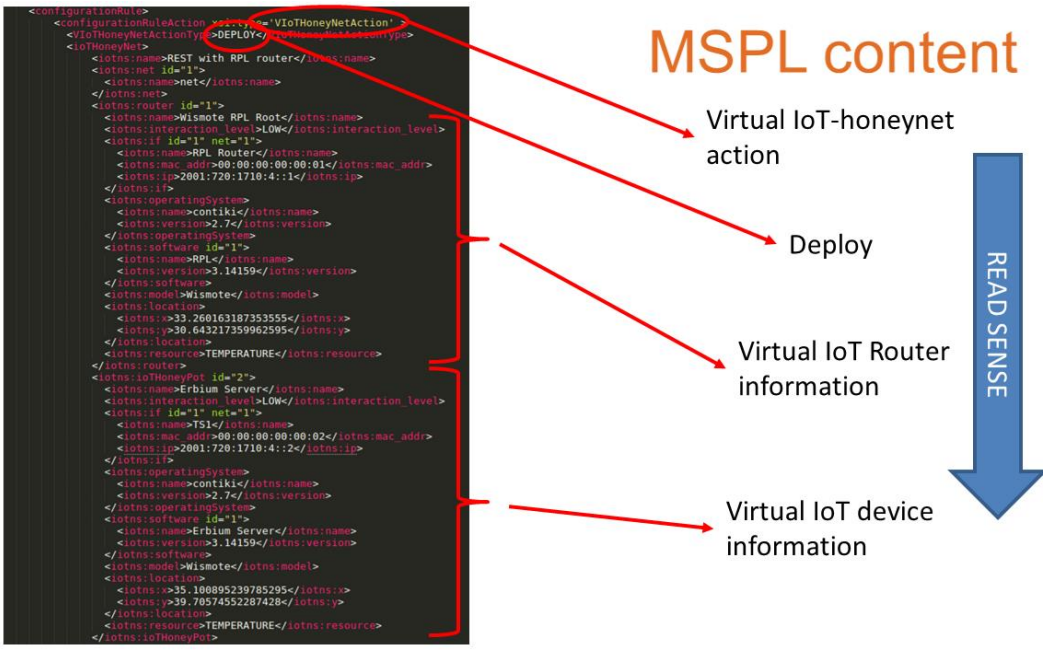


Figure 18. MSPL file generated by MAS for deploying a virtual IoT-honeynet for BMS.2

3.1.1.1.4 Reaction Deployment

The last phase comprises the deployment of the reaction previously generated by the Mitigation Action Service. This reaction (in form of an MSPL file) is received by the Security Orchestrator (SO) and translated by Policy Interpreter in order to start the orchestration process that will deploy the countermeasure. In this use case, the security orchestrator coordinates the OSM NFV controller for deploying a virtual honeynet and the ONOS SDN controller for transparent forwarding the attacker traffic towards the virtual honeynet, as shown in Figure 19.

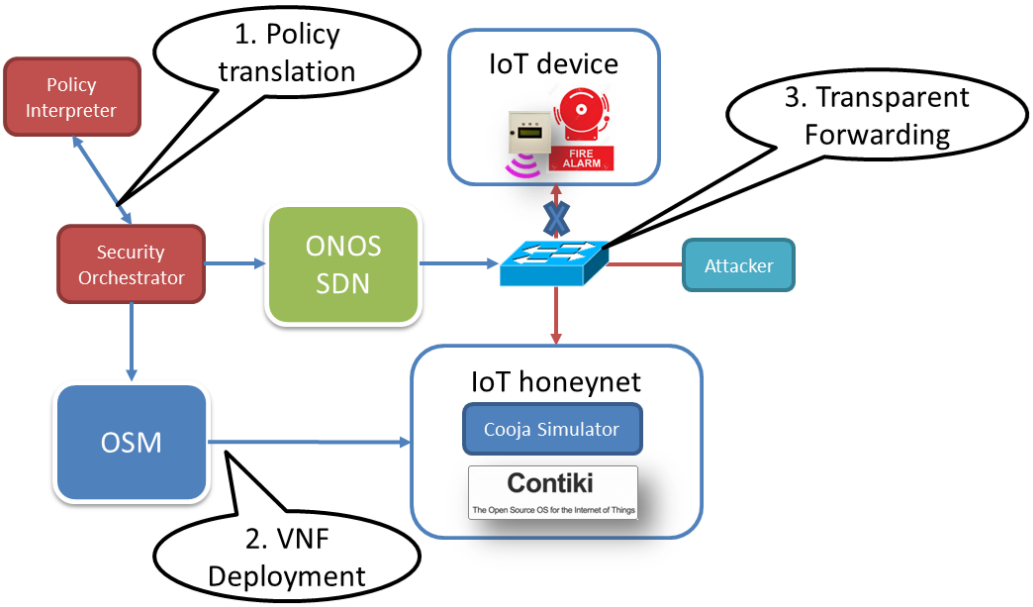


Figure 19. Reaction deployment for BMS.2

The next figures show the data captured (logs and screens) from the Orchestration and Enforcement components to do the main interactions of this phase:

1. Figure 20 shows the logs captured from the Policy Interpreter (at the right side) to process the MSPL file and generate the configuration for the Cooja security enabler (at the left side). In this use case, we use Cooja that is an emulator of IoT network based on Contiki operation system.

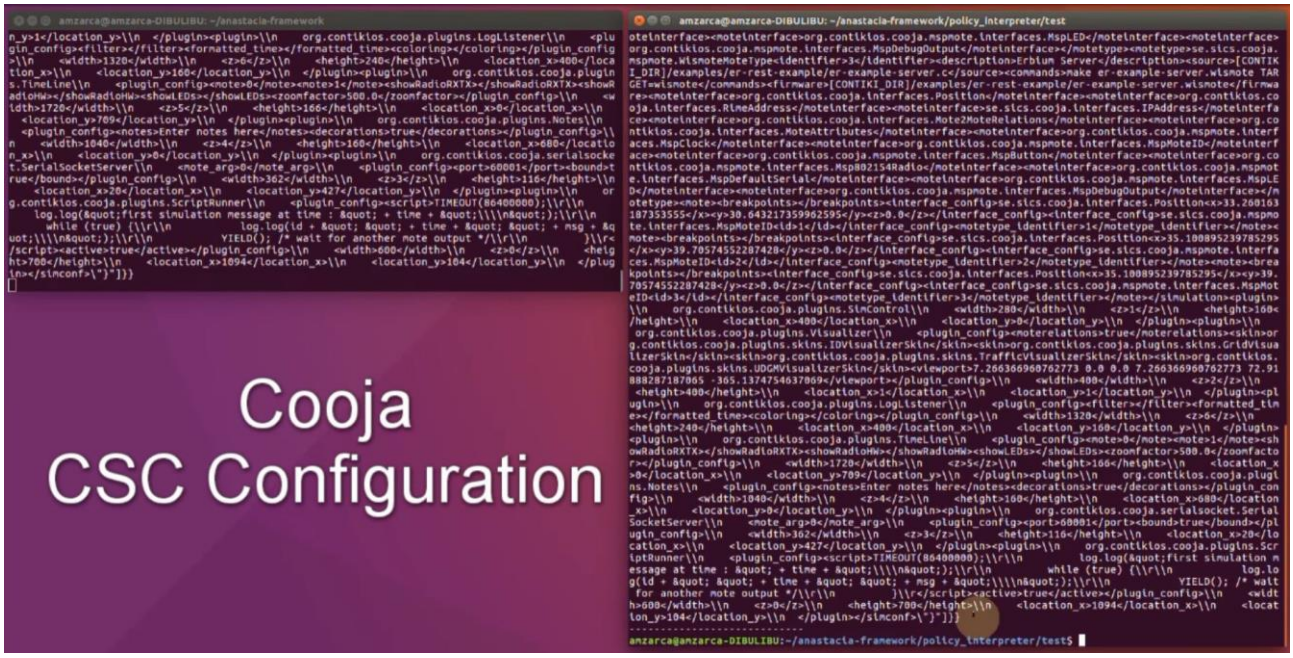


Figure 20. Cooja translation of the MSPL file for virtual honeynet by Policy Interpreter for BMS.2

2. Figure 21 shows the web interface captured from VNF controller (MANO) for starting the VNF Cooja instantiation.

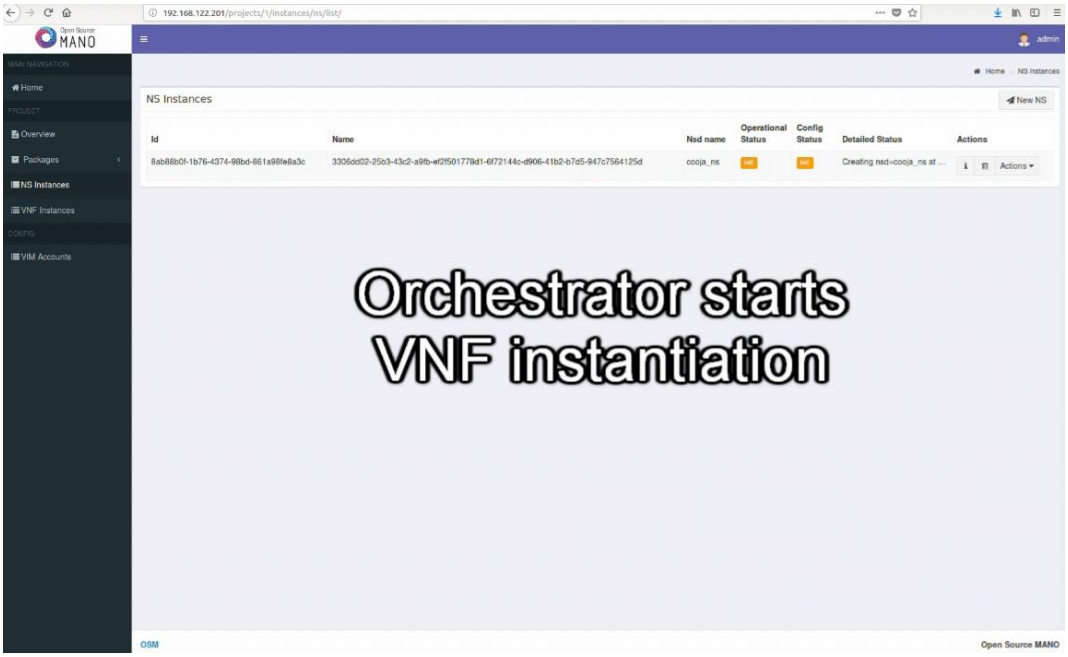


Figure 21. VNF instantiation of Cooja honeynet for BMS.2

3. Figure 22 shows the web interface captured from the SDN controller (ONOS) where is represented the SDN rules to perform the forwarding ordered by Security Orchestrator.

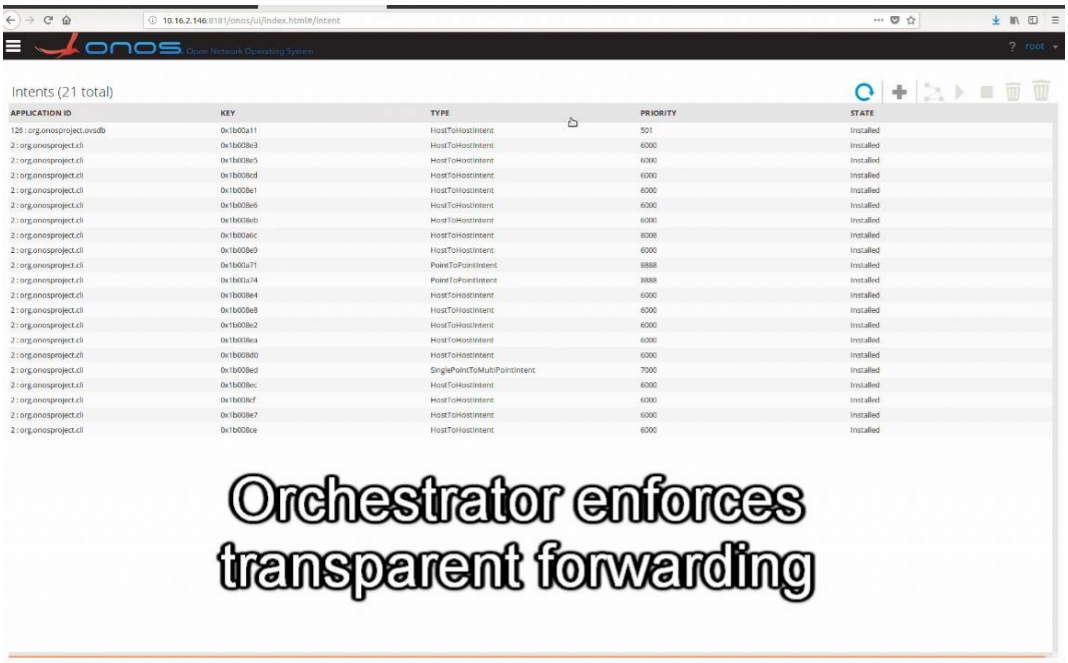


Figure 22. SDN rules for traffic forwarding towards the virtual honeynet for BMS.2

4. Figure 22 shows the network management interface of the SDN controller (ONOS) where is represented the traffic forwarding from the attacker towards the new Cooja simulated network, so called iot-honeynet.

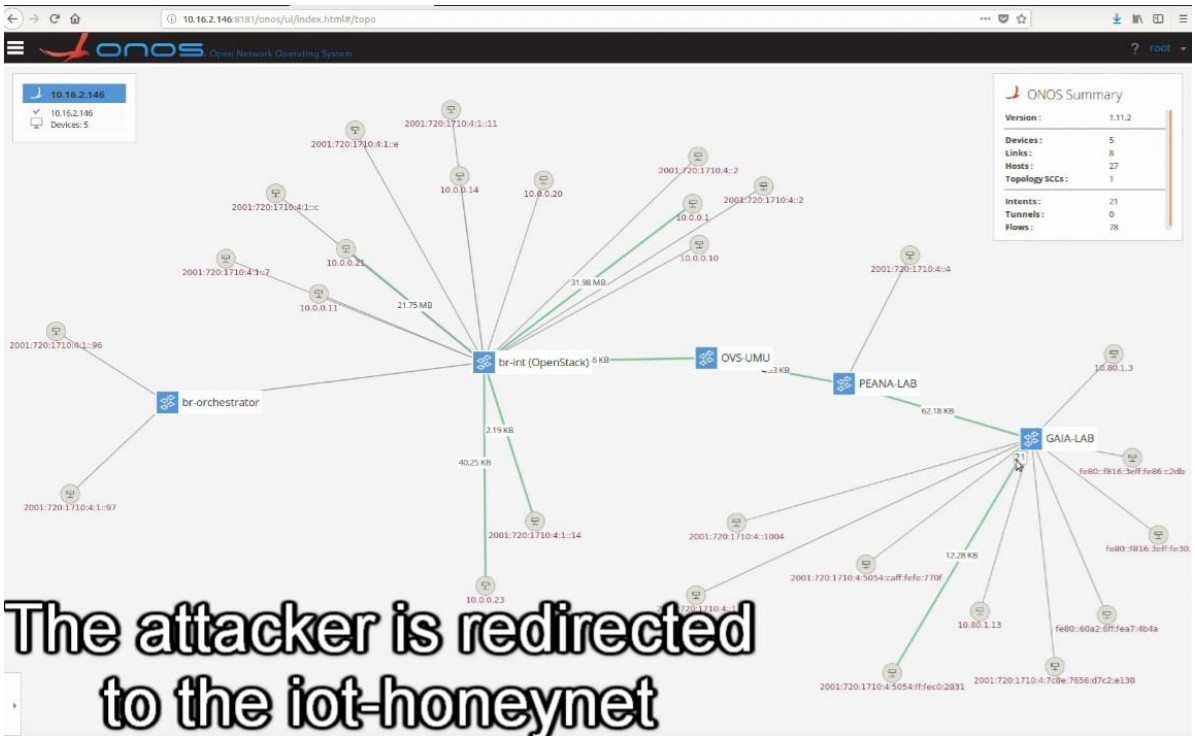


Figure 23. Network management inferface of SDN controller with virtual honeynet for BMS.2

The iot-honeynet emulates a virtual IoT network containing the same IPv6 addressing and firmware of real sensor devices and an IoT router to allow traffic between iot-honeynet and real Ethernet network. So, the SDN controller can redirect the traffic from the attacker towards the virtual honeynet. The attacker cannot realise the change from the real network towards the virtual network.

3.1.2 Results of Test-Cases with KPIs and Data Captured

The following subsection shows the measured KPIs of the validation executions according to the test cases defined in the deliverable D6.2. Below, the tables show the results of test-cases and also the KPI measurements in seconds for 5 executions of the performance tests.

- **TC_BMS.2.1 Validating a capability token for device actuation**

TC_BMS.2.1	Validating a capability token for authorization of device actuation
Execution	· IoT device validates the capability token to authorize the actuation. · Determine if the authorization decision is accept or deny.
KPI(s)	· Processing Time for the validation of capability token in IoT device.

The KPI values measured for the token validation is shown in the next table:

Execution Num.	1	2	3	4	5	Mean
Time1 (seconds)	0.15	0.15	0.17	0.14	0.16	0.15

To validate the capability token, an IoT device requires the processing time to receive the token in JSON format, parse it, and verify the security fields (i.e. lifetime and ECC signature). The current time is measured for the validation of lifetime field in the capability token. This time may be increased with the usage of a specific chip for the validation of the ECC signature.

- **TC_BMS.2.2 Network Authentication and Device Registration**

TC_BMS.2.2	Network Authentication and Device Registration
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)
KPI(s)	1. Time for PANA HSPL -> MSPL policy refinement. 2. Time to select the security enabler. 3. Time for PANA MSPL -> ONOS policy translation. 4. Time for network authentication between IoT device and PANA Agent 5. Time for new device publication between IoT device and IoT Controller 6. Time for the generation of device registration in IoT Controller to send to Orchestrator 7. Time for traffic enabling between Orchestrator and SDN controller

The KPIs measured for the network authentication are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0.3	0.07	0.06	0.09	0.05	0.11
2. Time (seconds)	1,62473	1,605	1,3731	1,4559	1,593	1,53044
3. Time (seconds)	0.24	0.08	0.13	0.12	0.09	0.13
4. Time (seconds)	1.906	1.781	2.445	1.789	1.781	1.802
5. Time (seconds)	0.648	0.640	0.664	0.640	0.664	0.651

6. Time (seconds)	0.027	0.018	0.021	0.020	0.022	0.021
7. Time (seconds)	0,03129	0,034	0,0342	0,0364	0,036	0,0344

Below, the 7 KPIs are described in detail to perform the network authentication among different ANASTACIA components:

1. The Policy Interpreter receives the HSPL and it performs the policy refinement. The time depends on the complexity of the security policy as well as on the amount of system model information required to generate the MSPL. Since right now the system information is obtained from local it is expected that this time increases in the future. All the policy interpreter times has been taken in the UMU´s local testbed. The refinement and translation could vary slightly depending on the network latency.
2. Security Orchestrator requests the list of enablers, and then identify the adequate capability from the Security Enabler Provider component. Moreover, the Security Orchestrator requests the low-level configuration from the Policy Interpreter by sending the MSPL file. The obtained time is a summation of the two precedents actions.
3. The Policy Interpreter receives the MSPL and the selected security enabler. Then it downloads the plugin and performs the translation. This time depends on the complexity of the security policy, being it also dependent on the plugin implementation. In this case ONOS plugin. It is important to highlight that each execution is translating two MSPL in order to allow bidirectional traffic.
4. The current implementation engages in a complete PANA exchange between the PAA and the IoT device (PaC), using the EAP-PSK method as a lightweight authentication method. The current time is not expected to change in the future. Possible increase can be due to retransmission of packet losses.
5. The current implementation encompasses an CoAP exchange between the IoT device and the Controller. In this exchange the information to publish is sent to the Controller in the payload. The time to publish information is subject to increase as security (DTLS and the Capability Token) is sent to complete the publication. It is also subject to the variability due to packet loss and CoAP retransmissions.
6. The IoT Controller receives the IoT notification and then it gets a FWD MSPL template, filling it in order to allow the IoT device presence in the network as a default behaviour. The time could be increased in the future depending on the amount of default security policies have been defined by the security administrator, as well as the quantity of the information required to generate the MSPL.
7. The Security Orchestrator sends requests to the SDN controller to enable IPv6 traffic in the SDN Switch (OVS)s . The computed time represents the difference in time from the reception of the requests and the first packet exchange.

- **TC_BMS.2.3 Unauthorized Actuation Detection**

TC_BMS.2.3	Unauthorized Actuation Detection
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.
KPI(s)	1. Processing time for the reception of unauthorized actuation detection in IoT broker 2. Processing time for threat notification in Kafka broker

The next table shows the KPI values measured for the 2 main operations done to notify the unauthorized actuation:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,405	1.39	0,419	0,419	0,405	0,65475
2. Time (seconds)	0,181	0,035	0,035	0,021	0,037	0,0618

Below, we describe in detail the 2 main operations for the notification of unauthorized action from IoT device towards the Kafka broker through IoT broker:

1. This time represents the notification message sent from the IoT device to the IoT broker through a proxy entity that translates CoAP to HTTP. The time to reception of the notification message is not expected to increase. This time is subject to increase do to loss of packets and the retransmission policy of CoAP.
2. This metric shows the additional time added by the pre-processing (data cleansing, filtering and reformatting) of the events. In next iterations of the platform the pre-processing might include more complex operations and the time could increase.

- **TC_BMS.2.4 Monitoring Management**

TC_BMS.2.4	Monitoring Management
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.
KPI(s)	· Processing time for attack detection in Incident Detector

The next table shows the KPI values measured for the monitoring management of the unauthorized threat:

Execution Num.	1	2	3	4	5	Mean
Time (seconds)	0.107	0.052	0.03	0.041	0.082	0.0624

This KPI includes the processing time for the Incident Detector since the notification is received from Kafka Broker until is sent to VDSS. Communication times between Kafka Broker - Incident Detector and Incident Detector - DSS depends on the network conditions, so they have not been included.

- **TC_BMS.2.5 Reaction Management**

TC_BMS.2.5	Reaction Management
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
KPI(s)	· Processing time to create the MSPL file from MAS when an alert is generated

The next table shows the KPI value measured for the reaction management of the unauthorized attack:

Execution Num.	1	2	3	4	5	Mean
Time (seconds)	0,355	0,33	0,221	0,364	0,04	0,262

This KPI represents the time computed inside the MAS, calculating the time (in seconds) used by the MAS to compute the MSPL since it receives the alert from the VDSS). These values might increase during next iteration of ANASTACIA development due to more complex computations to generate the MSPL file.

- **TC_BMS.2.6 Orchestration Management**

TC_BMS.2.6	Orchestration Management
Execution	· Orchestrator requests the translation of MSPL file to Policy Interpreter · According to the security enablers repository, Policy Interpreter responses a concrete configuration to enable a Honeynet of virtual IoT devices using Cooja emulator.
KPI(s)	· Time for filtering MSPL -> Cooja policy translation.

The next table shows the KPI value measured for the reaction management of the unauthorized attack:

Execution Num.	1	2	3	4	5	Mean
Time (seconds)	0.22	0.12	0.13	0.12	0.11	0.14

The KPI indicates the time of the Policy Interpreter to process the MSPL file according to the repository of security enablers. Then the interpreter downloads the plugin of the Cooja security enabler and performs the translation. This time depends on the complexity of the security policy, being it also dependent on the plugin implementation. In this case Cooja plugin, whose translation is the most complex at the moment.

- **TC_BMS.2.7 Enforcement Management**

TC_BMS.2.7	Enforcement Management
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.
KPI(s)	1. Obtain the response time for VNF deployment between Orchestrator and VNF Controller. 2. Obtain the response time for traffic forwarding between Orchestrator and SDN Controller.

The next table shows the 2 KPIs values measured for the enforcement management from the Security Orchestrator:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	89,7066	78,91	84,179	79,084	79,33	82,2427
2. Time (seconds)	0,03129	0,034	0,0342	0,0364	0,036	0,0344

Below, we describe in detail the 2 KPIs of the enforcement management performed over VNF and SDN controllers:

1. The Security Orchestrator sends a request to the NFV controller to create and deploy (up and running) a VNF instance (Cooja). The computed time represents the difference in time from the reception of the VNF request and the deployment of the Cooja VNF.

2. The Security Orchestrator sends a request to the SDN controller to steer the traffic in the VNF instance (Cooja). The computed time represents the difference in time from the reception of the SDN request and the first traffic forwarded.

- **TC_BMS.2.8 Full integration of ANASTACIA framework**

TC_BMS.2.8	Full integration of ANASTACIA framework for use case BMS.2
Execution	• To complete all steps ordered from TC_BMS.2.2 to TC_BMS.2.8
Expected results	• The completed integration is successful.
KPI(s)	• Verify the complete integration 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.

This test case aims to verify the complete integration of the use case BMS.2 to detect an unauthorized actuation and mitigate the attack with a virtual honeynet.

Execution Num.	1	2	3	4	5	Mean
Verification	PASS	PASS	PASS	PASS	PASS	PASS

3.2 USE-CASE BMS.3: REMOTE ATTACK ON THE BUILDING ENERGY MICROGRID

3.2.1 Use case description + Benefits from ANASTACIA framework

The BMS.3 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. In particular, the use case considers a SQL injection attack from IoT devices towards the SCADA database in order to remove or manipulate all the sensitive data of energy microgrid.

The benefits of ANASTACIA architecture are the automated detection of malicious traffic in order to filter the SQL queries and stop the attack. To cope with this attack, the Montimage Monitoring Tool (MMT) is deployed as a sniffer device in the IoT network to detect the attack raising the corresponding alerts. In addition, the ANASTACIA platform brings the flexibility of using SDN and VNF techniques to deploy the countermeasures. This novelty allows the dynamic creation of a new virtual instance that hosts the firewall to filter the SQL messages towards the SCADA database. Using SDN and NFV techniques, the ANASTACIA platform can deploy different countermeasures along the network to mitigate the attacks.

To evaluate this use case, Figure 24 shows all required interactions of ANASTACIA components. The IoT network has been emulated using the Cooja software, which is part of the Contiki environment[1]. This emulation contains an emulated compromised device (in a first instance), an IoT router (to allow traffic to go outside the IoT network) and an instance of MMT-IoT-Sniffer; a modified Contiki device that allows extracting the packets form the network to be analysed by MMT-Probe.

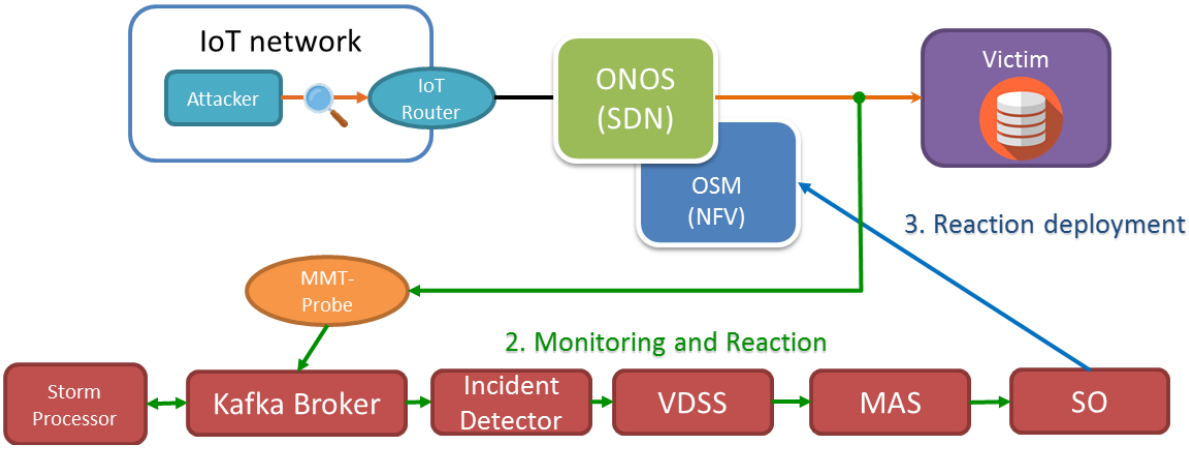


Figure 24: Complete flow of use case BMS.3

3.2.1.1 Complete flow of use case BMS.2

The complete flow of BMS.3 use case is divided into 3 phases:

1. Within IoT network, attacker compromises an IoT device to send COAP messages including SQL injection queries towards a SCADA database.
2. Attack detection is made by the combination of MMT-IoT-Sniffer and MMT-Probe to analyse the traffic inside the IoT network. The threat notification is processed by monitoring components to determine a security alert and generates a mitigation reaction to block the COAP messages from the attacker.

[1] http://www.contiki-os.org/

3. Reaction deployment is coordinated by Security Orchestrator in cooperation with SDN controller (ONOS) and NFV controller (MANO).

In the next subsections, these different stages of the use case are described in detail.

3.2.1.1.1 Attack triggering

The first stage includes the execution of an attack script in a compromised IoT device to start the SQL injection towards the SCADA database as shown in Figure 25. This phase is validated with an emulated IoT network (containing the attacker and the IoT router) and the emulated SCADA database acting as the victim. The communication between these two machines is provided by ONOS – the SDN provider – in order to transfer the packets between the attacker and the victim.

Once the attack has been started, the attacker send continues COAP messages with SQL queries towards the IPv6 address of the victim. To see the attack and the correct mitigation of the attack, the *tcpdump* tool is used on the victim with the goal of following the incoming packets on the victim, therefore being able to watch how the victim receives the attack and how it is blocked by the ANASTACIA platform.

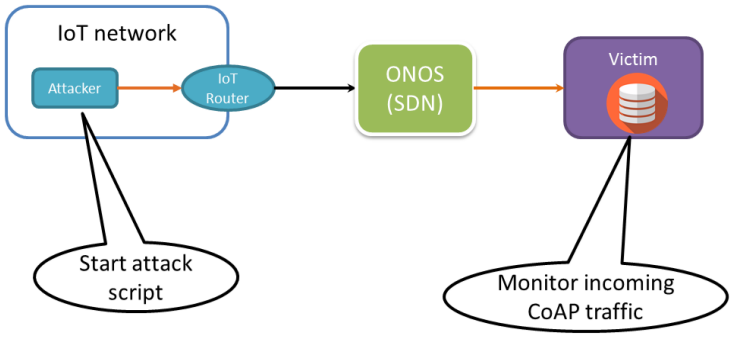


Figure 25: Attack triggering for BMS.3

Figure 26 is a screen captured from the BMS.3 use case showing the attack triggering of SQL injection from an IoT network towards the victim. The *tcpdump* tool is running at the bottom-right console to show the COAP messages including the SQL queries received by the victim. In the figure, the attacker is executing at the top-left console and the Mitigation Action is executing in the bottom-left console to generate the adequate reaction. Moreover, the monitoring agent called MMT-IoT-sniffer is running in the top-right console to capture the network traffic for the analysis of MMT-Probe in order to detect threats.

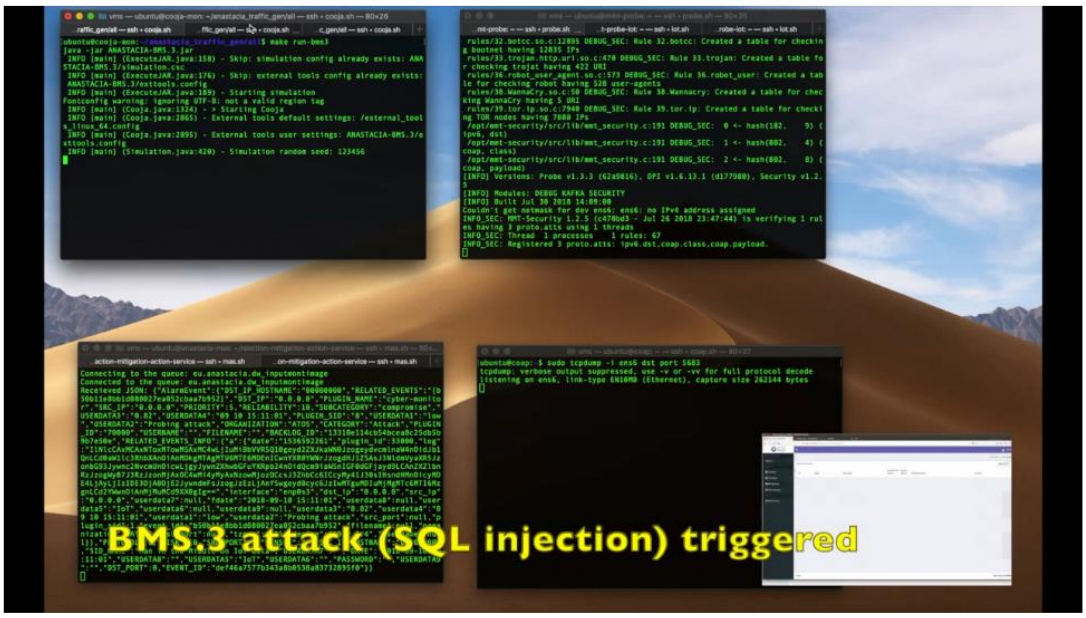


Figure 26: Attack triggering of SQL injection for BMS.3

3.2.1.1.2 Monitoring and Reaction

The second phase of the use case comprises the Monitoring and Reaction stages. Figure 27 shows the interactions on this phase.

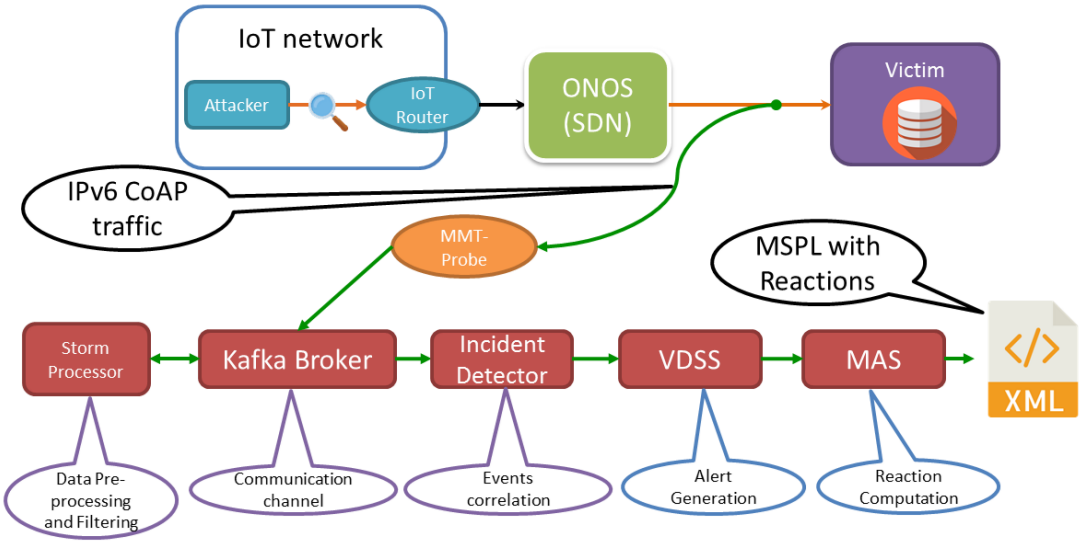


Figure 27: Monitoring and Reaction Phase for BMS.3

This phase starts with the execution of the MMT-IoT-Sniffer software, which enables the extraction of the packets from the IoT network. These packets are analysed by an MMT-Probe instance, which generates periodic network statistics reports and, eventually, security reports regarding any detected attack.

In the ongoing scenario, the COAP/SQL packets inserted by the attacker are analysed by MMT-Probe, which generates a respective security report and send it to the monitoring/reaction chain (shown in red colour in Figure 27) through the Kafka Broker. Using the Storm processor, the Kafka Broker re-formats the MMT report and sends it to the Incident Detector performing the events correlation to feed the VDSS (Verdict Decision Support System) which raises the alert about the detected attack. The alert is sent to the MAS (Mitigation Action Service), which reads the alert and generates the corresponding MSPL file containing the countermeasures to be deployed on the IoT network.

For the BMS.3 use case, the adopted countermeasure specifies the filtering of the SQL queries over COAP messages from the attacker towards the victim. Figure 28 shows the specific syntax of the generated MSPL file with XML-based language.



Figure 28: Generated MPSL File for BMS.3

The next figures show the data captured (logs and screens) from the monitoring and reaction components to perform the 2 main interactions of this phase:

1. Figure 29 shows the web XL-SIEM interface of the Incident Detector to correlate the threat events indicating the IPv6 addresses of attack source and victim destination. In particular, Incident Detector shows the attack notified from MMT-Probe with the name “SQL injection” towards the victim in the IPv6 address: 2001:720:1710:4:1::11

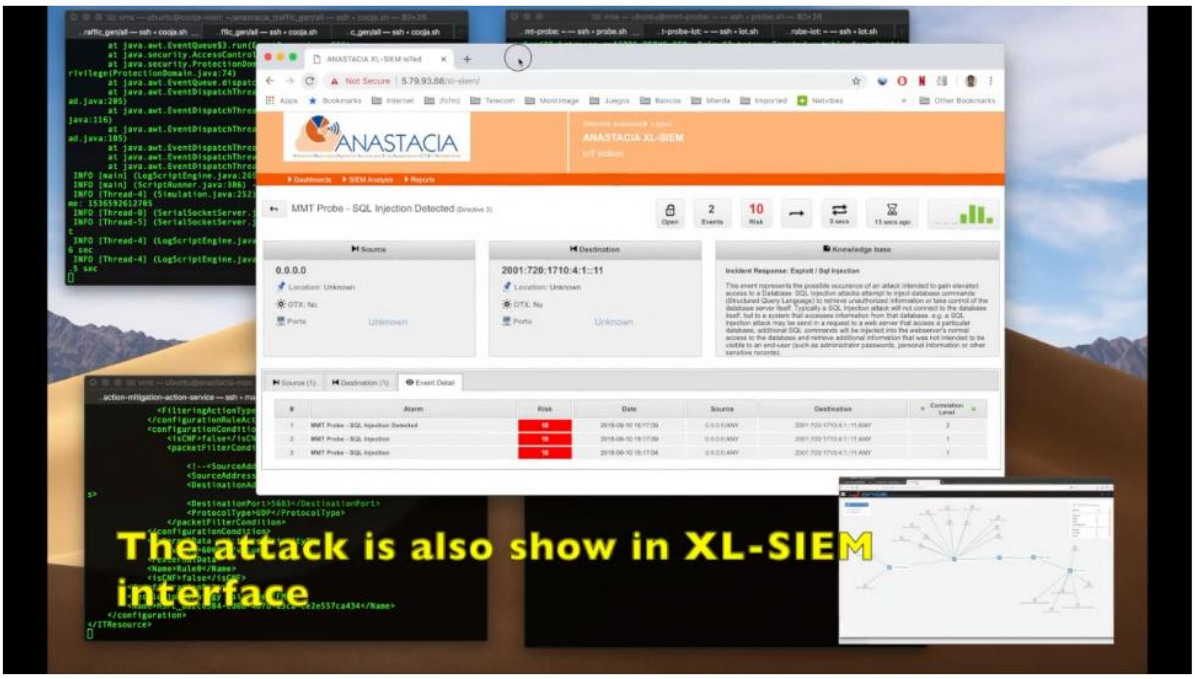


Figure 29: Events correlation in Incident Detector for BMS.3

2. Figure 30 shows the bottom-left console of Mitigation Action Service where the alert is processed and the MSPL file shown in Figure 28 is generated to be sent to Security Orchestrator.



Figure 30: MSPL generation in Mitigation Action Service for BMS.3

3.2.1.1.3 Reaction Deployment

The last phase comprises the deployment of the reaction previously computed by the Mitigation Action Service. This reaction (in form of an MSPL file) is processed by the Security Orchestrator in order to start the management of SDN and NFV controllers to enforce the countermeasures. Upon received the MSPL file, the Security Orchestrator contacts Security Enablers Provider and gets a list of the available Security Enablers capable of deploying the traffic filtering. The Security Orchestrator selects one and contacts the translation service of Policy Interpreter to transform the MSPL file into lower level language – i.e. security enabler specific configurations – that represent the countermeasures specified in the MSPL file. With this translation, the Security Orchestrator is ready to start the orchestration process.Then, the Security Orchestrator manages the specific commands to SDN and NFV controllers in order to enforce the traffic filtering, as shown in Figure 31. In particular, the Security Orchestrator requests to: (1) OSM NFV controller to create a new Virtual Machine instance which will contain the Virtual Firewall; (2) ONOS SDN controller to filter the COAP protocol between the specified IP addresses; and (3) ONOS SDN controller to redirect all the traffic between the specified addresses through the recently-created Virtual Firewall.

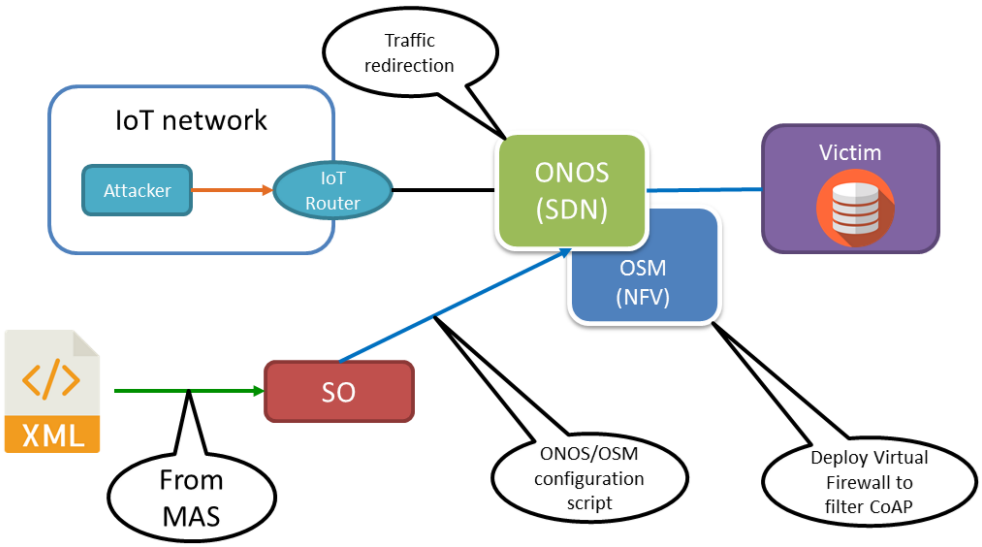


Figure 31: Deployment of the Reaction for BMS.3

Once these tasks have been performed, the traffic is effectively being re-routed and actively filtered by the new firewall. This has the immediate effect of filtering all the SQL queries over COAP packets between the attacker and the victim, which effectively mitigates the on-going attack.

The next figures show the data captured (logs and screens) from the Orchestration and Enforcement components to perform the main interactions of this phase:

1. Figure 32 shows the web interface captured from the MANO NFV Controller when it receives the query of a new VNF instantiation of a Virtual Firewall.

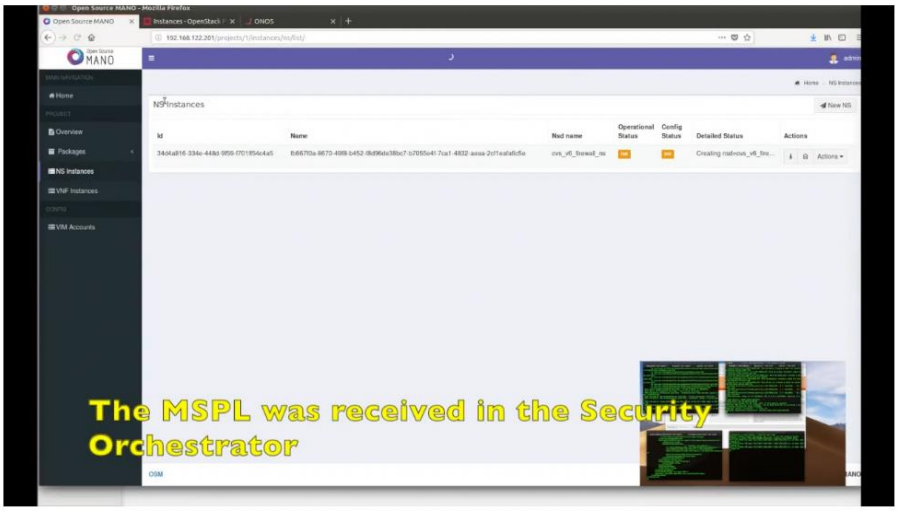


Figure 32: Reception of MSPL file in Security Orchestrator for BMS.3

2. Figure 33 shows the new VM instance created in the web interface of VNF controller (MANO) to deploy the Virtual Firewall as countermeasure. The Virtual Firewall is added dynamically to the topology with the IPv4 address: 10.16.2.146 and name “Device.5”.

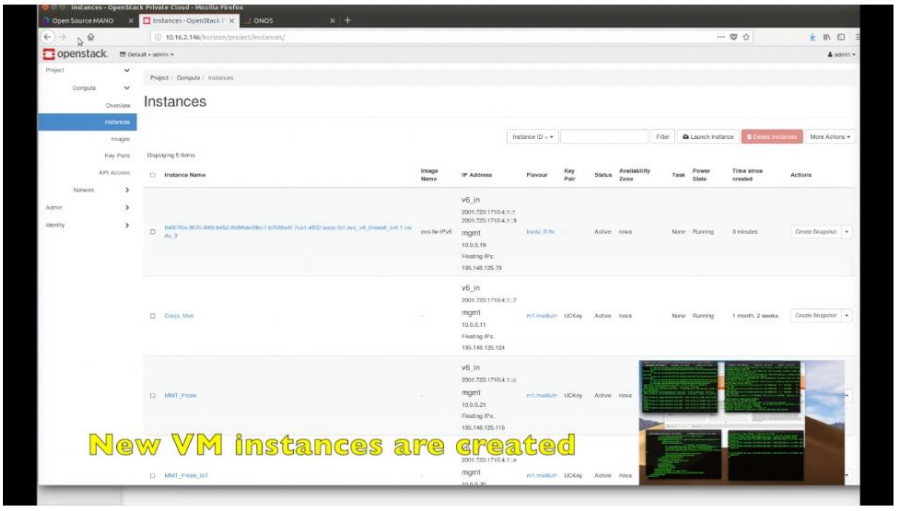


Figure 33: VNF instantiation of Virtual Firewal for BMS.3

3. Figure 34 shows the interfaces captured from the SDN controller (ONOS) where is indicated the SDN rules to perform the filtering ordered by Security Orchestrator.

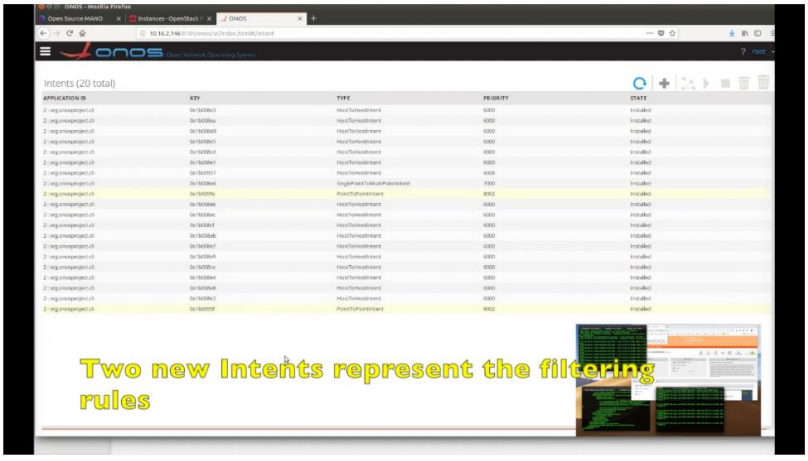


Figure 34: Filtering rules received in ONOS SDN Controller for BMS.3

4. Figure 35 shows the network management interface of the SDN controller (ONOS) where is represented the traffic filtering from the attacker towards the new VNF instance of Virtual Firewall.

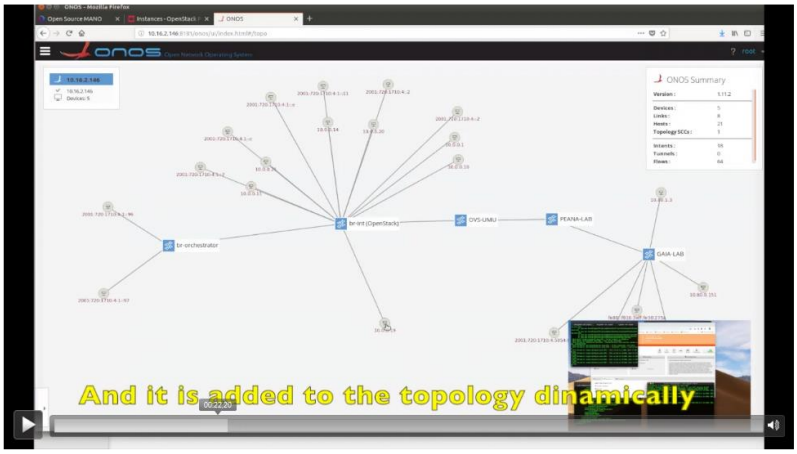


Figure 35: Network management inferface of SDN controller with Virtual Firewall

5. Figure 36 shows the attack is effectively mitigated, after a few seconds of the countermeasures deployed. The COAP messages with SQL queries from the attacker are now filtering. So the attacker (at the top-left console) continues sending malicious messages that are blocked to not arrive to the victim (at the bottom-right console).

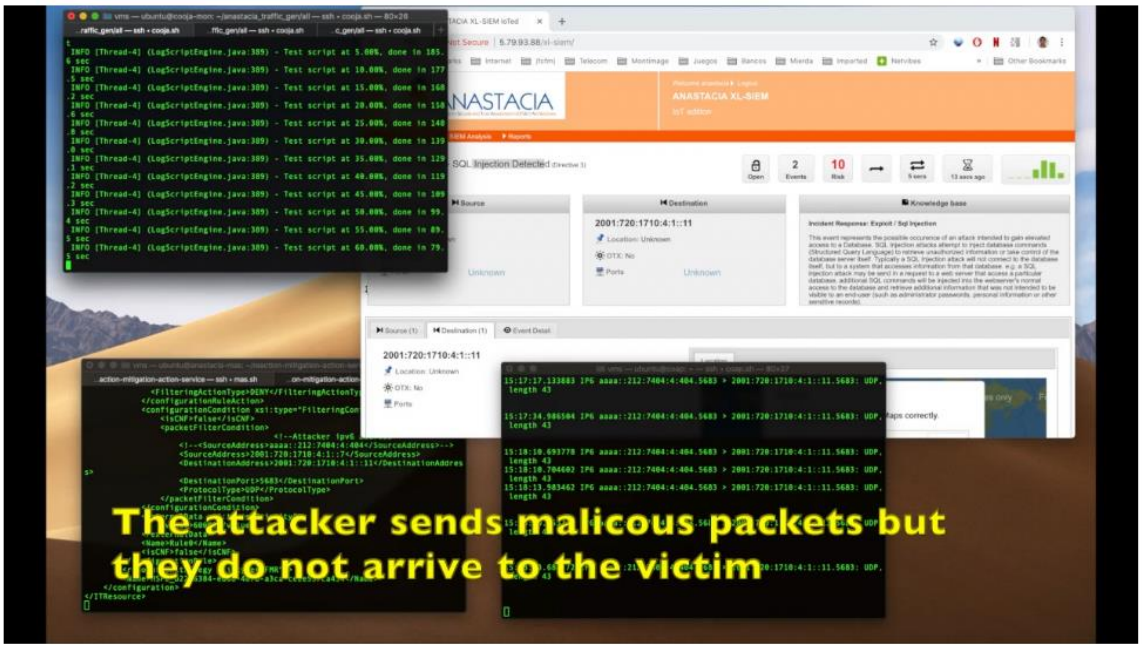


Figure 36: SQL injection is blocked and the attack is mitigated for BMS.3

3.2.2 Results of Test-Cases with KPIs and Data Captured

The following subsection shows the measured KPIs of the validation executions according to the test cases defined in the deliverable D6.2. Below, the tables show the results of test-cases and also the KPI measurements in seconds for 5 executions of the performance tests.

- **TC_BMS.3.1 MMT-Probe basic network analysis**

TC_BMS.3.1	MMT-Probe basic network analysis
Execution	· Start all the components from above in that order · Observe JSON messages coming from MMT-Probe on Kafka message broker
KPI(s)	1. MMT-Probe is correctly started and publishes statistics in Kafka Channel 2. The execution is maintained for 5 minutes and statistics are published during this time.

The next table shows the 2 KPIs values validated for the network analysis of MMT-Probe: **Execution Num.**	**1**	**2**	**3**	**4**	**5**	**Mean**
1. Verification to publish	PASS	PASS	PASS	PASS	PASS	PASS
2. Verification of execution	PASS	PASS	PASS	PASS	PASS	PASS

This test case aims to fulfill a functional requirement rather than providing performance information (fulfilling non-functional requirements). This indicator will complement the last one in order to provide a high-availability of the ANASTACIA platform, fulfilling the requirement FR-10.

- **TC_BMS.3.2 – Monitoring Module detection**

TC_BMS.3.2	Monitoring Module detection
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
KPI(s)	1. Attack detection of SQL injection for the MMT-Probe is detected within 5 sec

The KPI measured for the attack detection of COAP message with SQL injection for the MMT-Probe is shown in the next table:

Execution Num.	1	2	3	4	5	Mean
Time (seconds)	0,014	0,054	0,04	0,109	0,075	0,0584

To measure these values, in one side the starting timestamp is computed with the attack script and on the other side, the detection timestamp is extracted from the MMT-Probe report when the trigger of the security rule is checked. Both the attack script and the MMT-Probe instance were run on the same machine to avoid clock synchronization issues.

- **TC_BMS.3.3 – Reaction to SQL injection attack on SCADA server**

TC_BMS.3.3	Reaction to SQL injection attack on SCADA server
Execution	· Start all the components as stated in TC_BMS.3.2 · Perform attack by running malicious script to inject packets containing SQLi attacks
KPI(s)	1. MSPL file with the countermeasures is generated according to the detected use case. 2. Reaction is computed within 1 sec from receiving the alert from VDSS

The KPIs validated for the reaction to SQLi attack are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Verification	PASS	PASS	PASS	PASS	PASS	PASS
2. Time (seconds)	0,43	0,408	0,42	0,442	0,395	0,419

This test case aims to complement the last one by adding the computation of the MSPL countermeasure without deploying it. TC_BMS.3.3 aims to cover the functional requirement of deploying the reaction and provide KPIs for this process. These values are computed inside the Mitigation Action Service (MAS), calculating the time (in miliseconds) used by the MAS to compute the MSPL since it receives the alert from the VDSS). These values might raise during next iteration due to more complex computations to generate the MSPL file.

- **TC_BMS.3.4 – Monitoring and Reaction with Legit COAP requests**

TC_BMS.3.4	Monitoring and Reaction with Legit CoAP requests
Execution	· Start all the components as stated from above · Inject legit CoAP reequests using automated script · Observe logs on monitoring, reaction components as well as VDSS, MAS, SO
KPI(s)	1. No attack is detected by MMT-Probe (pass/fail)
	2. No alert is generated by VDSS (pass/fail)
	3. No reaction is computed by MAS (pass/fail)
	4. Security Seal stays at the same level

The KPIs validated for the monitoring phase with Legit COAP messages are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Verification	PASS	PASS	PASS	PASS	PASS	PASS
2. Verification	PASS	PASS	PASS	PASS	PASS	PASS
3. Verification	PASS	PASS	PASS	PASS	PASS	PASS
4. Verification	PASS	PASS	PASS	PASS	PASS	PASS

This test case aims to guarantee the proper access (and legit operations) of registered devices and users to send COAP messages to the SCADA server.

- **TC_BMS.3.5 – Monitoring and Reaction**

TC_BMS.3.5	Monitoring and Reaction
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
KPI(s)	1. Time to process the alert received in Security Alert System from VDSS 2. Security and Privacy Seal updated within 3 sec

The KPIs measured for the reaction phase of malicious events detected as SQL injection attack are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,02	0,03	0,03	0,02	0,04	0,028
2. Time (seconds)	2,582	2,48	2,626	2,488	2,491	2,533

Below, the 2 KPIs are described in detail:

1. The measured times refer to the time passing between alerts are received by the VDSS (Verdict Decision Support System) and the relative output (enriched alerts) are sent to the RabbitMQ queue shared with the DSPS (Dynamic Security Privacy Seal).
2. The reason of this relatively slow performance is mainly due to the DSPS storage and its infrastructure. When the DSPS server needs to write the seal on the on-chain storage (blockchain-based with BFT consensus) it takes in average 1.27 seconds. When it needs to write the seal log on the off-chain storage (based on shamir secret schema) it takes in average 0.943 seconds. These times depend a lot from the number of used nodes and their hardware so potentially it could be improved, however we don't have enough resources to update the infrastructure. In next

iterations, more functionalities will be added so the total time will not likely decrease. On the other hand we don't expect to increase it a lot since actions not involving the storage don't have major impacts on the global performance.

- **TC_BMS.3.6 – Attack Detection, Reaction and Security Orchestration**

TC_BMS.3.6	Attack Detection, Reaction and Security Orchestration
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
	1. Time to select the security enabler by the Security Orchestrator 2. Time for filtering MSPL -> OVS policy translation. 3. Time to generate the enforcement of SDN traffic filtering by Security Orchestrator

The KPIs measured for the orchestration and enforcement of traffic filtering are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	1,62473	1,605	1,3731	1,4559	1,593	1,53044
2. Time (seconds)	0.09	0.04	0.04	0.03	0.06	0.052
3. Time (seconds)	0,03129	0,034	0,0342	0,0364	0,036	0,0344

Below, the 3 KPIs are described in detail:

1. Security Orchestrator requests the list of enablers, and then identifies the adequate capability from the Security Enabler Provider component. Moreover, the Security Orchestrator requests the low-level configuration from the Policy Interpreter by sending the MSPL file. The obtained time is a summation of the two precedents actions.
2. The Policy Interpreter receives the MSPL and the selected security enabler. Then it downloads the plugin and performs the translation. This time depends on the complexity of the security policy, being it also dependent on the plugin implementation. In this case OVS Firewall plugin.
3. The Security Orchestrator sends requests to the SDN controller to enable IPv6 traffic in the SDN Switch (OVS)s . The computed time represents the difference in time from the reception of the requests and the first packet exchange.

3.3 USE-CASE BMS.4: CASCADE ATTACK ON A MEGATALL BUILDING

3.3.1 Use case description + Benefits from ANASTACIA framework

In this scenario, the adversary tries to exploit a zero-day vulnerability discovered in IoT devices deployed in the megatall building. By obtaining access to IoT devices and exploiting zero-day flaw, adversary can trigger fire and evacuation alarm causing physical and personal damages in the building. This type of attack might be implemented to divert attention of building security from unauthorized access by implementing security system saturation thus rendering them open to physical breach.

The ANASTACIA framework resolves this issue by addressing the temperature sensor attack. Zero-day vulnerabilities are not identified from signature based detections but rather observing overall system behaviour using Artificial Intelligence (AI) and Machine Learning (ML) techniques to build a model that will be capable of recognizing normal and abnormal system state.

The benefits of ANASTACIA framework are:

- Ability to detect even zero-day attacks by implementing latest techniques in AI and ML for cybersecurity scenarios to discover those threads.
- Capacity to implement mitigation action in fully autonomous manner without security manual operation.

The main idea behind this approach is that attacks are becoming too fast for human reaction and even if human reacts to specific cyberattack the reaction can be prone to error which might expose attacked resources to even greater extent.

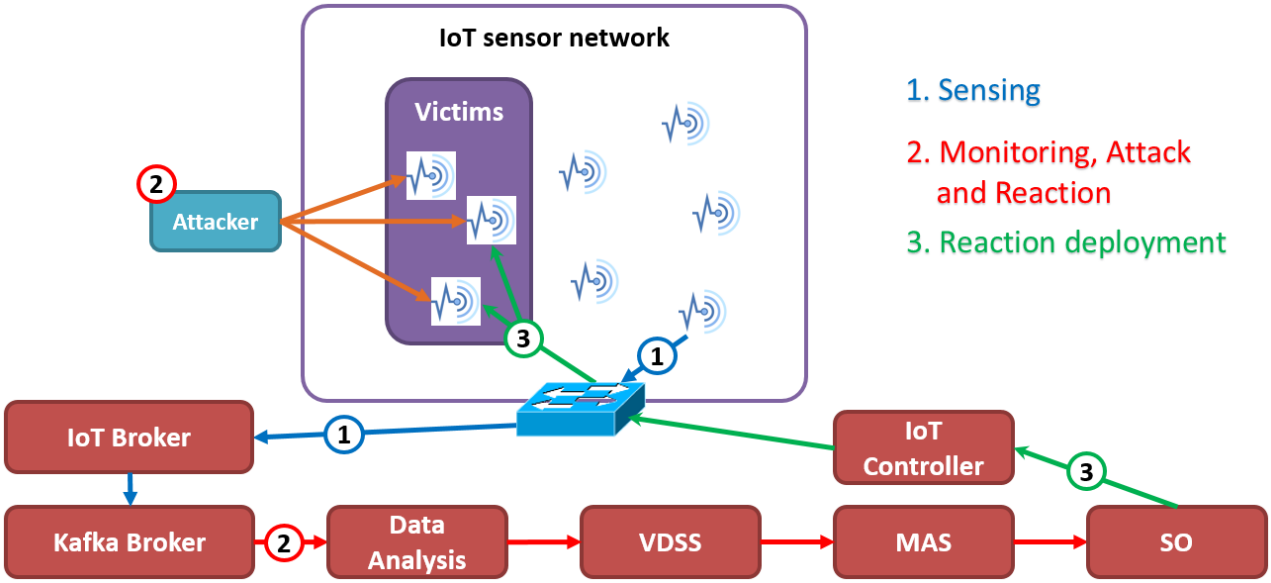


Figure 37: Use case scenario deployment for BMS.4

The use case execution has been illustrated on Figure 37. The scenario is divided into three main steps:

1. Sensing of building operational information (i.e. temperature values) from IoT sensor network employs series of IoT devices (marked blue on the picture). Sensing building information also enables Data Analysis model training phase before ANASTACIA framework is being deployed into protected infrastructure. In this step of BMS.4 scenario only IoT sensing is demonstrated.
2. Monitoring and reaction is running in ANASTACIA framework to detect and prepare mitigation action against attacks over sensor networks. In this step the ANASTACIA framework has been deployed fully on premise and work in parallel to normal building operations. It monitors behaviour

and it will detect malicious behaviour in case of an attack, and will prepare mitigation action to cope with the affected infrastructure.

3. Reaction deployment is executing to enforce mitigation actions or countermeasures in IoT network using SDN and NFV techniques. This action of ANASTACIA framework covers final part of deploying predefined mitigation actions into building infrastructure (i.e. temperature sensor network) to avoid attack and block malicious effect by adversary during execution of malicious activity.

3.3.1.1 Complete flow of use case BMS.4

3.3.1.1.1 Sensing

In the first step of the scenario, temperature measurement from physical device is demonstrated together with passing this information to IoT broker and Kafka broker as final destination point. The information is being published to IoT broker where it can be consumed in a distributed manner by ANASTACIA monitoring components to enable deep analysis of megatall building system and anomaly detection. Figure 38 illustrates measurements obtained on IoT sensor network emulating megatall building behaviour.

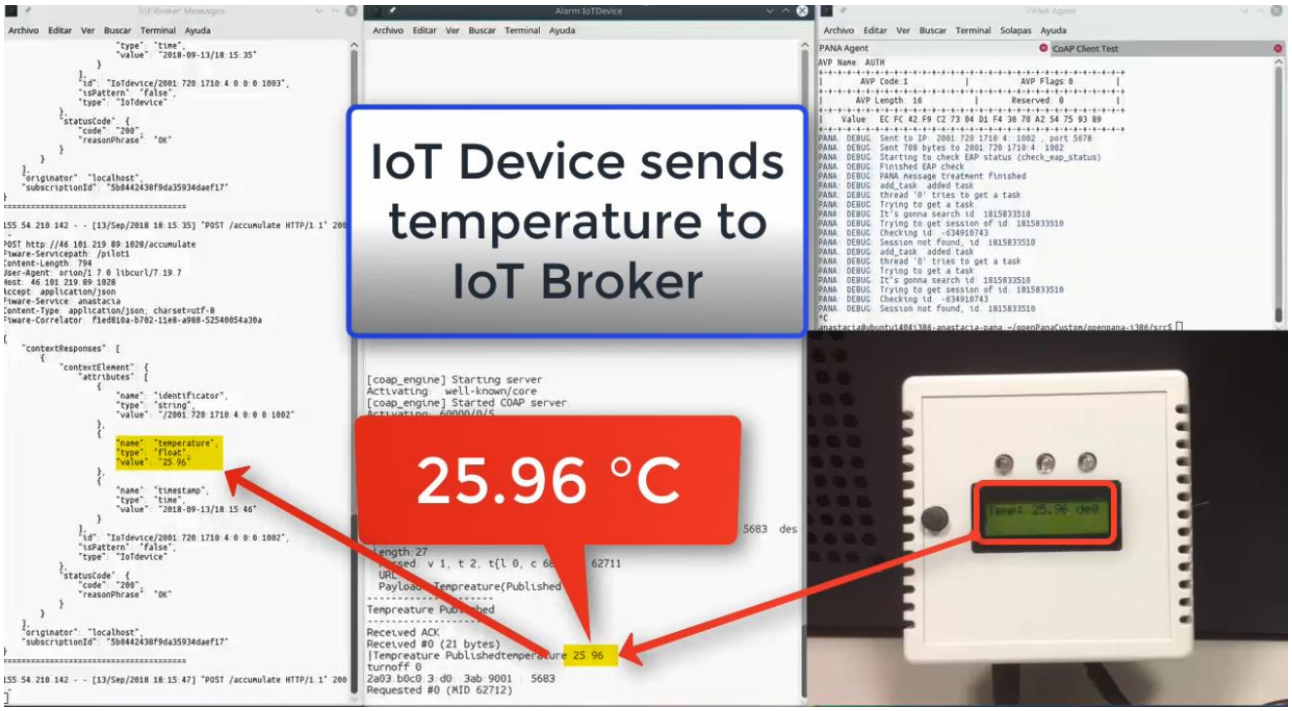


Figure 38: Temperature sensor measurements for BMS.4

The picture represents from the right IoT physical sensor, connection to a wireless IoT gateway where IoT sensor is reporting temperature. On the left side IoT broker console is depicted with log of receiving message from IoT sensor with the same temperature value.

Figure 39 depicts reception of temperature sensor information by Kafka broker. The picture represents a part of scenario where an attack is being triggered and high temperature values are observed. In the next stage, we will observe how this information are processed by Data Analysis agent and converted into attack verdict that are sent to the reaction components of ANASTACIA framework.

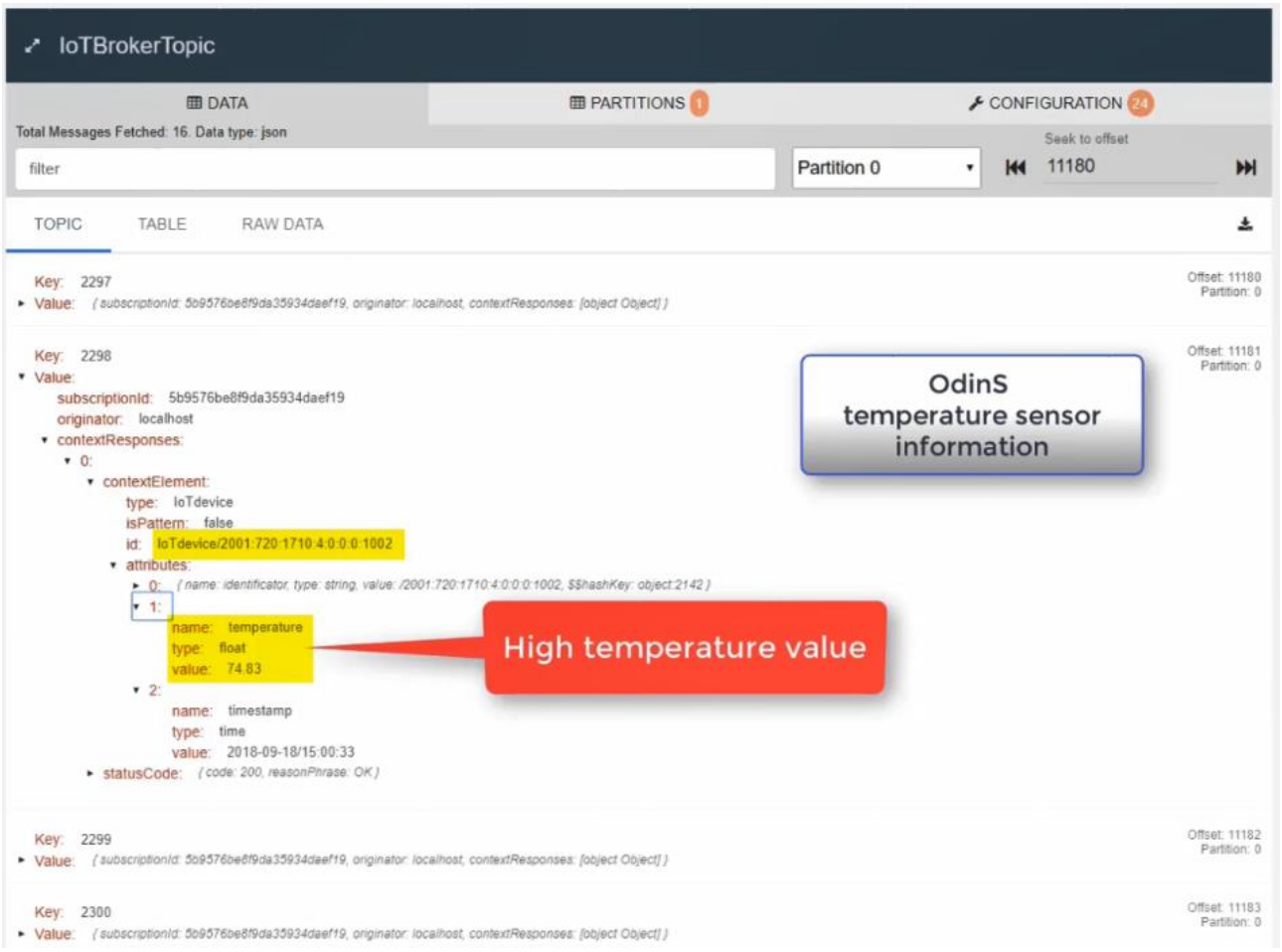


Figure 39: High temperature sensor measurement observed on Kafka broker for BMS.4

3.3.1.1.2 Monitoring and reaction

In this stage, ANASTACIA framework detection and reaction capabilities are presented. First, the monitoring components receive notifications of the temperature sensor information using asynchronous subscriber called Kafka broker. Later, Data Analysis agent buffers information and sent to the Machine Learning model for validation. Once the validation is completed, a new attack event is being generated by this agent and sent to Verdict Decision Support System (VDSS). The next figures show the data captured (logs and screens) from the monitoring and reaction components to do the main interactions of this phase.

Figure 40 illustrates the monitoring part of BMS.4 scenario for attack detection. On the left side, adversary attack script is emulating adversary's malicious behaviour that compromises an IoT sensor to send high temperature values into IoT sensor network. Yellow stripes and arrows represent data that was transmitted through monitoring chain (IoT Gateway, IoT Broker and Kafka Broker) and received by Data Analysis agent.

Right hand side of diagram depicts Data Analysis agent detecting high temperature values and sending attack event. Red marked console output shows attack event generated by the agent and sent to VDSS component with attack detection information.

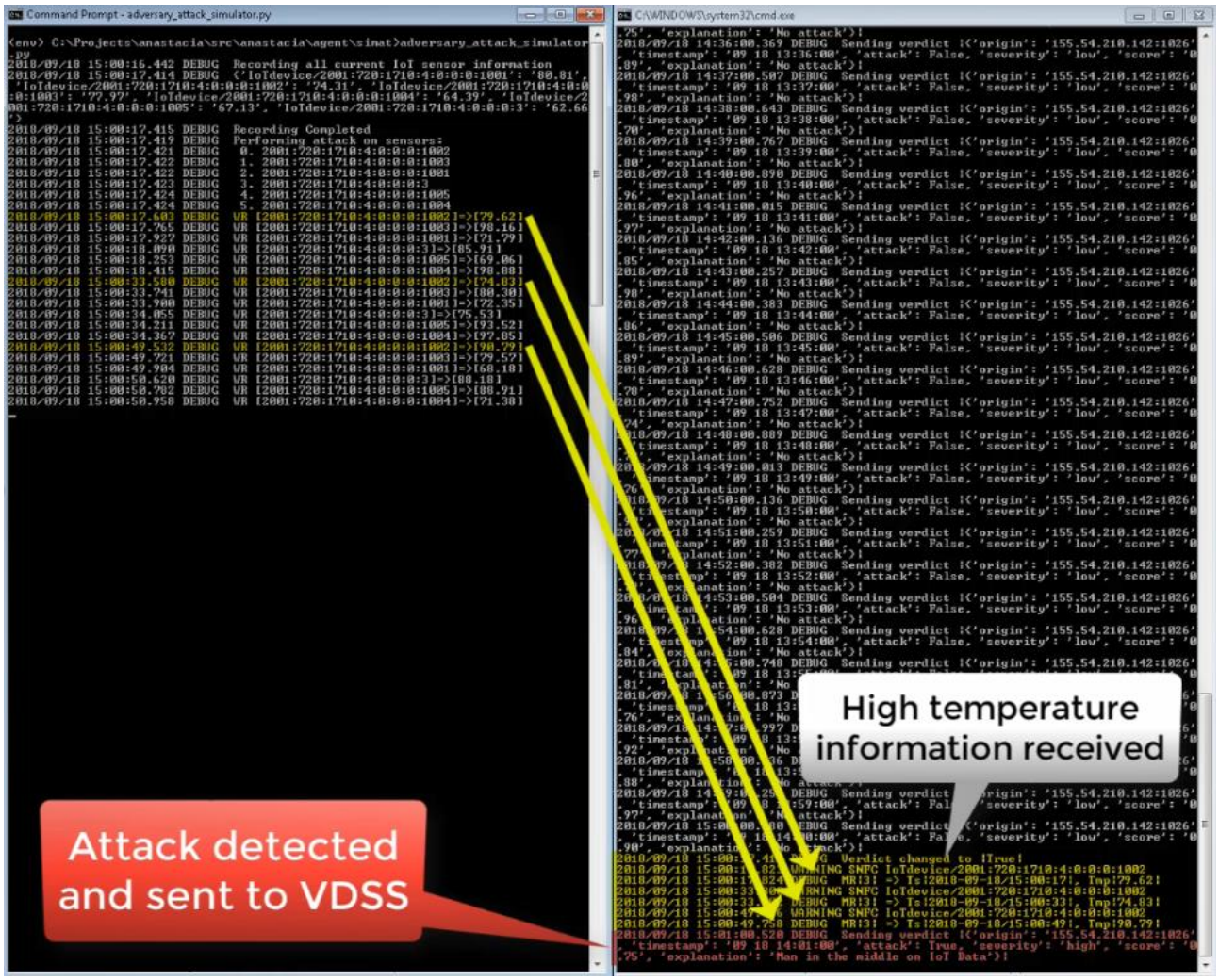


Figure 40: Attack script and data analysis detecting attack for BMS.4

Figure 41 represents the sending of attack event from Data Analysis agent towards Incident Detector (ID) and Verdict Decision Support System (VDSS). The following information provides high level overview of proposed system:

- Device that has been attacked (origin field),
- When attack is happened (timestamp),
- Attack flag determining whether system state is safe (no attack) or unstable (attack),
- Abnormal behaviour severity level (low, medium, high) that might help determine how serious anomaly detection is from monitored system level perspective (IoT sensor failure-battery replacement required vs. large number of devices affected in DDoS attack),
- Probabilistic score of malicious behaviour detection that is computed based on pre-trained data and current monitoring information being fed to the detection model,
- Textual explanation – human readable abnormal behaviour system state description that can be later used to notify system operators about the malicious activity.

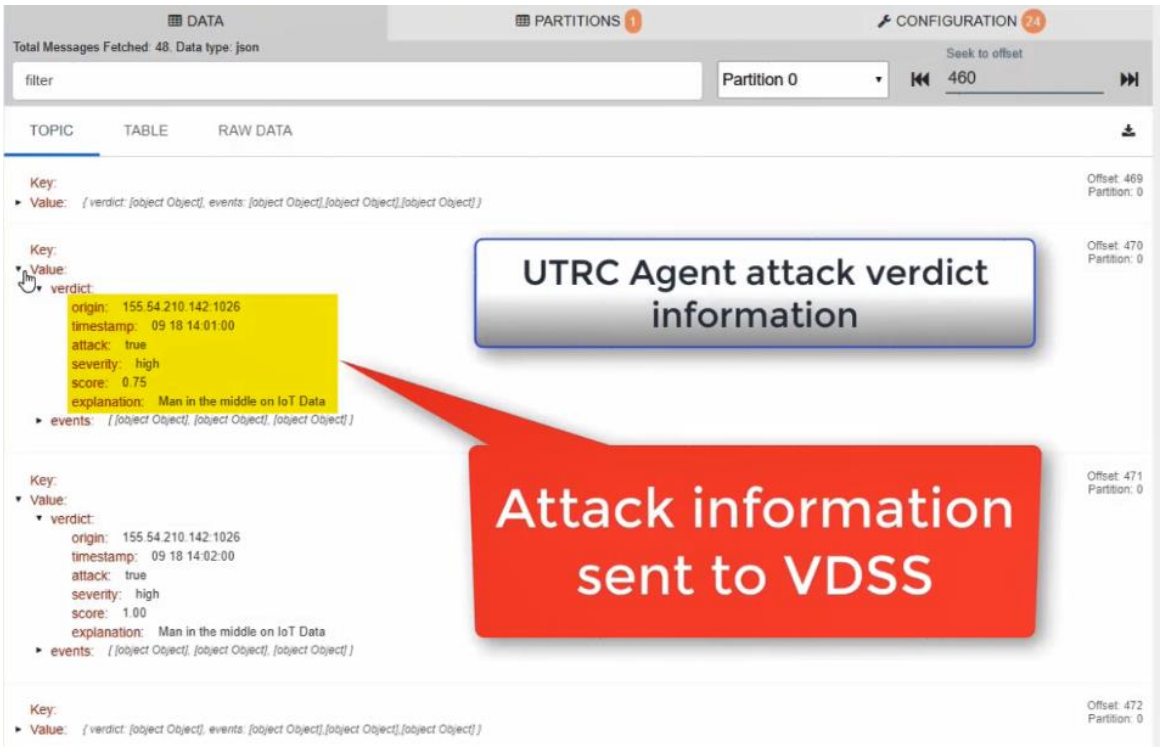


Figure 41: Attack event sent towards VDSS for BMS.4

Figure 42 provides visualization on how Incident Detector is processing attack event message. The Incident Detector shows attack event in its web interface. Next, based on the attack event information, the Incident Detector assigns a risk level, and BMS.4 attack test case is marked as level 10 (highest risk) and given red colour to attract attention of the system operator. The Incident Detector makes attack information and asset correlation for future mapping between events and monitored system elements for higher correlation accuracy. The correlation also enables to build over time attack signatures to enable faster response time and give operator more information about already known threats.

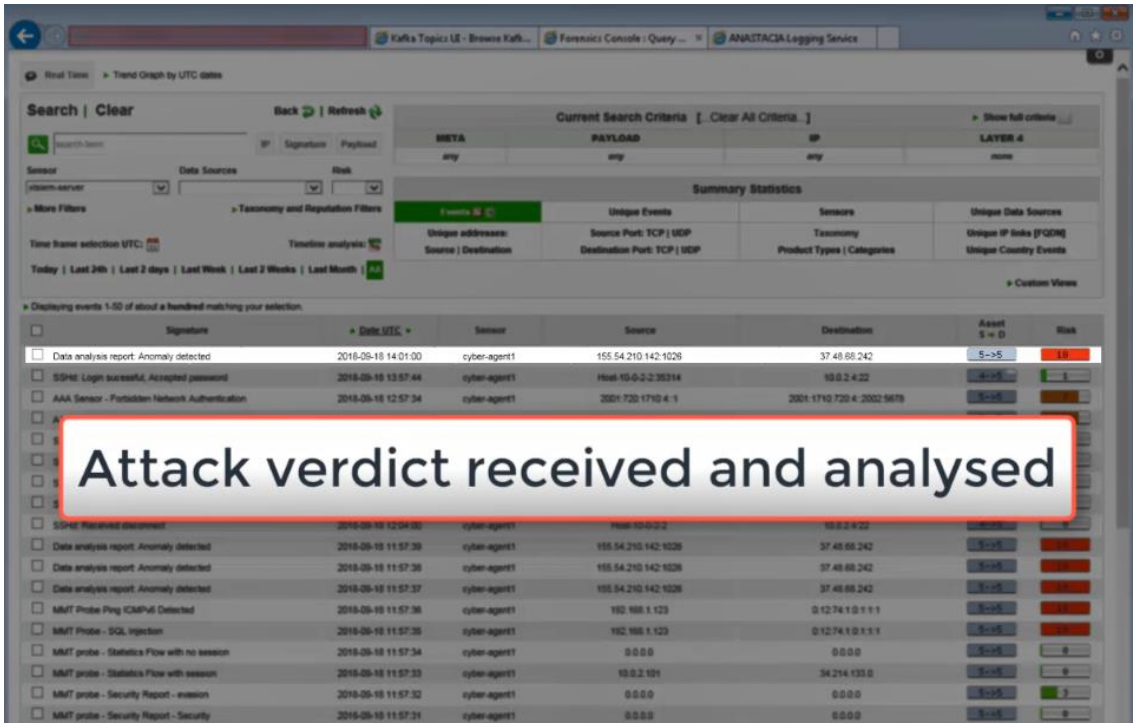


Figure 42: Attack verdict observed on ANASTACIA VDSS component for BMS.

In the next step, the Incident Detector sends the event report to the Verdict Decision Support System (VDSS). After information is being parsed, filtered and categorized accordingly to the predefined decision criteria. Then, VDSS provides a detailed view of the attack verdict presented on Figure 43 in following four sections:

- Normalized event – the normalized information received from attack verdict is presented. Thereby, we can find event signature triggers based on the VDSS component that is generating alerts and mitigation action within ANASTACIA framework,
- SIEM – provides high level event correlation made by VDSS, at this level depending on monitored system specifics more fields can be added for higher correlation accuracy,
- Context – describes information about particular setting in which event happened,
- KDB – Knowledge database information about the event.

In BMS.4 use case, it is zero-day attack and no information is provided on the context and KDB fields as this is being seen as completely new event in the VDSS component. This approach enables ANASTACIA framework to record security incidents to ensure that system operators, cyber-security experts and IT specialists will have access to alert information and will enable them to triage, resolve and remediate potential harm that is caused by detected anomalies.

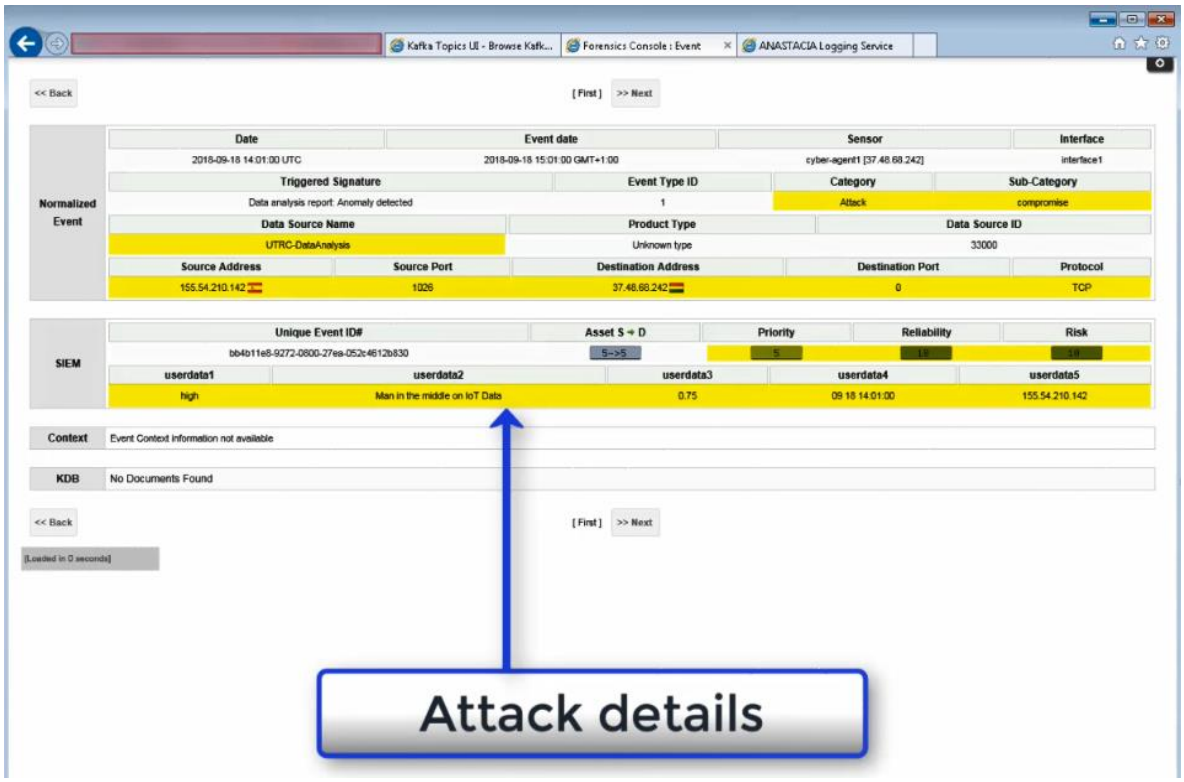


Figure 43: Attack verdict details observed on VDSS component for BMS.4

When the attack is detected by VDSS, two messages are generated:

- Alert message that is processed to initial anomaly detection alert. The alert message will be send to SAS (Security Actuation Service) in order to provide current system state to system operator.
- Mitigation action message with MSPL file is created from MAS (Mitigation Action Service) to send the Security Orchestrator in order to enforce the countermeasures.

Figure 44 shows these two exchanged messages for SAS and MAS components. On the top is alert message sent to SAS while second represents mitigation action sent from MAS towards the Security Orchestrator to enforce the countermeasures.

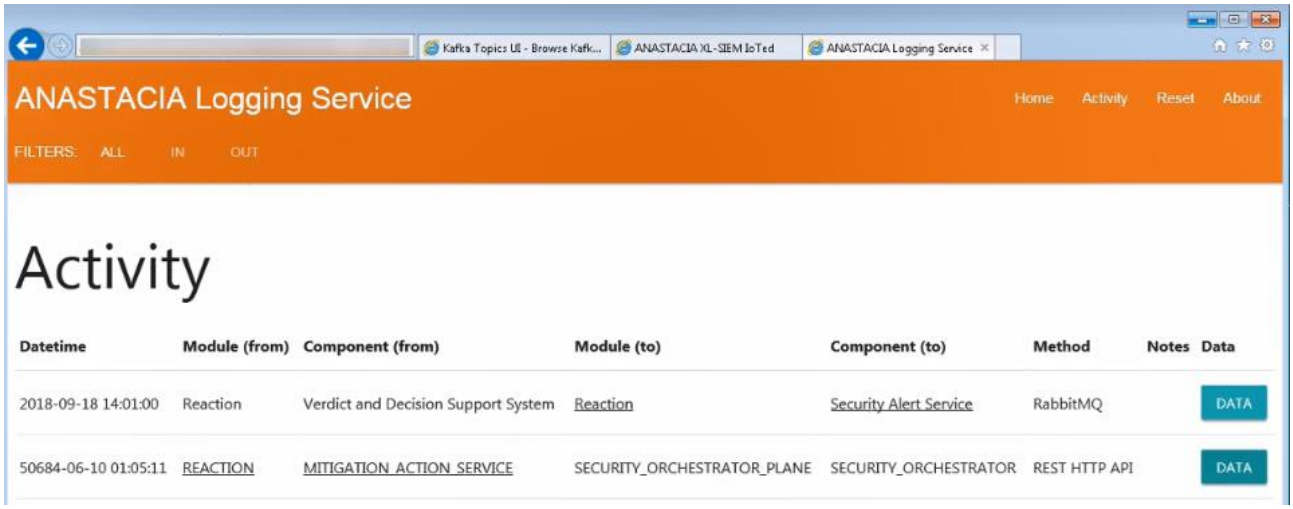


Figure 44: ANASTACIA logging service for SAS and MAS components

Figure 45 shows the alert message sent from VDSS to SAS indicating the attack details like type “Man in the Middle on IoT Data” and IP address for the source and the destination.

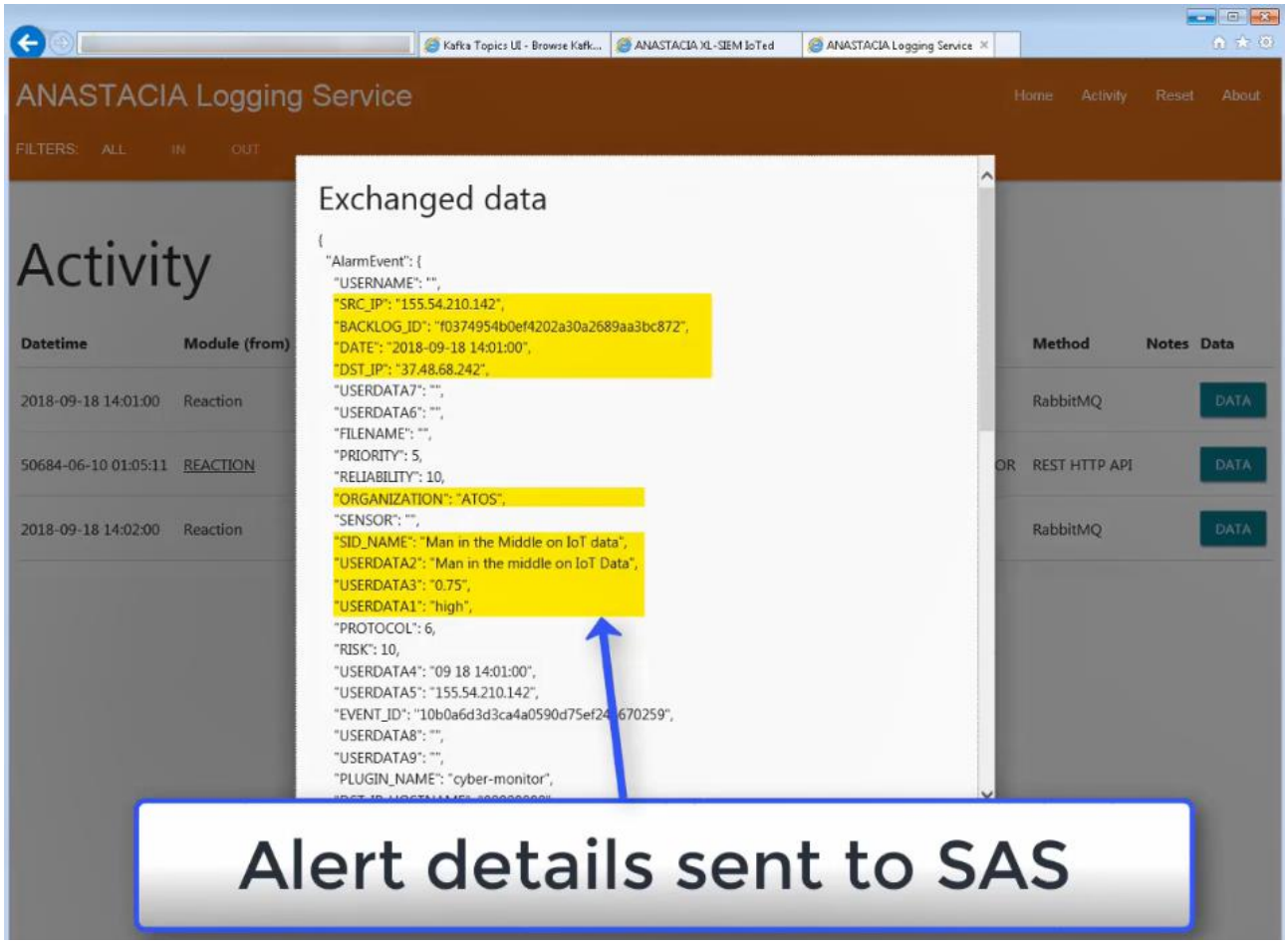


Figure 45: Alert message sent from VDSS to SAS component

Figure 46 shows the message sent by SAS to the Security Orchestrator that includes the MSPL file with the mitigation countermeasures.

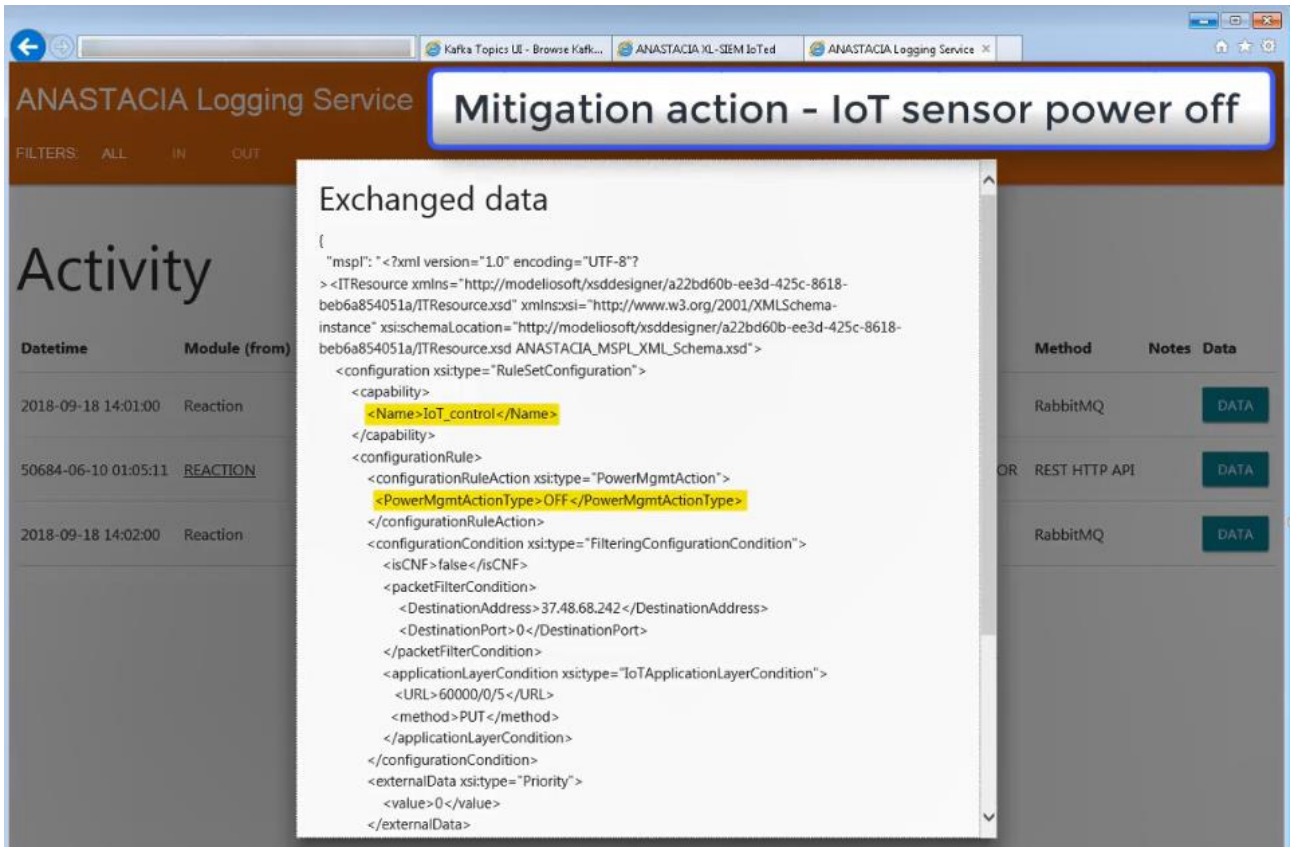


Figure 46: MSPL file sent from MAS to Security Orchestrator for BMS.4

Figure 47 shows the full description of the MSPL file generated by Mitigation Action Service to request to power off the compromised IoT device. There are five main items in the MSPL file:

- Action type – this field represents action that will be performed by underlying HW and SW elements connected to ANASTACIA framework. In BMS.4 use case, the selected action is set to power off affected IoT device.
- Filtering rule – section illustrates capabilities based on which executed action will be filtered and this is characterized in detail by following fields:
- IoT sensor address – network L3 address (IP) of the IoT sensor on which action will be performed. IPv6 and IPv4 addresses are supported.
- GW prefix – represents prefix based on which a specific sensor can be reached via GW REST API.
- Protocol type – Type of the protocol used by IoT Broker. In this case the selected protocol type is REST and IoT Broker provides REST API to manipulate on sensors.

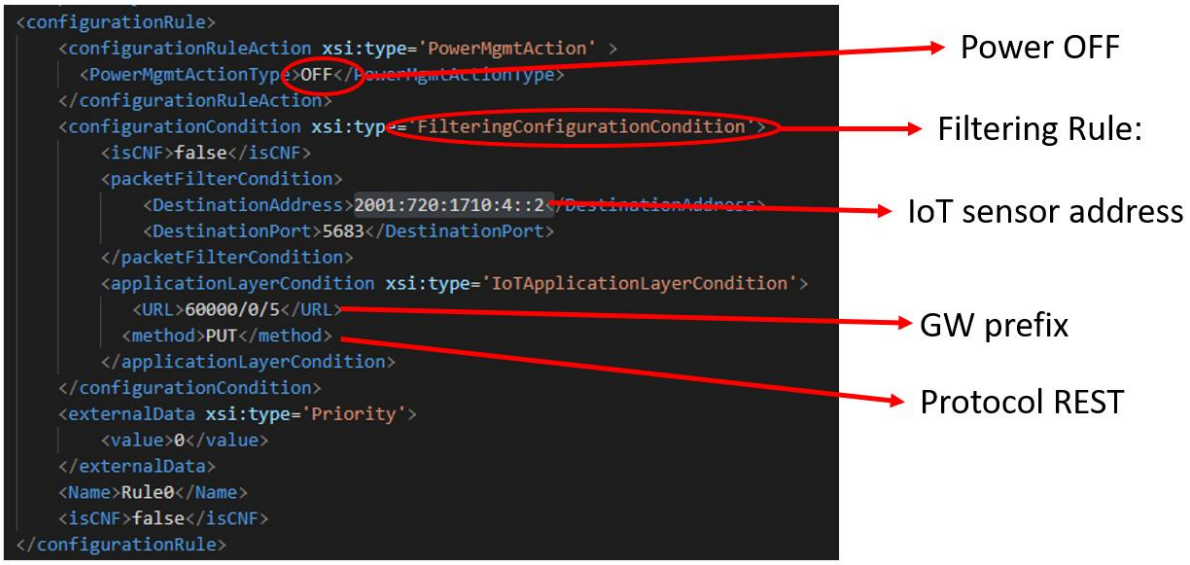


Figure 47: MSPL file to mitigate IoT sensor attack with power off action for BMS.4

On the next step, the MSPL file is received by the Security Orchestrator for further processing and mitigation execution on the IoT sensor network to reduce attack vector on megatall building infrastructure.

3.3.1.1.3 Reaction deployment

Upon reception of MSPL file, the Security Orchestrator requests the translation by Policy Interpreter in order to obtain the specific configuration to deploy the countermeasure. In this case, the translation indicates the specific configuration to send to the IoT controller in order to generate a COAP message to request powering off the compromised IoT device. Figure 48 shows the MSPL file translated by Policy Interpreter in the top-right console.

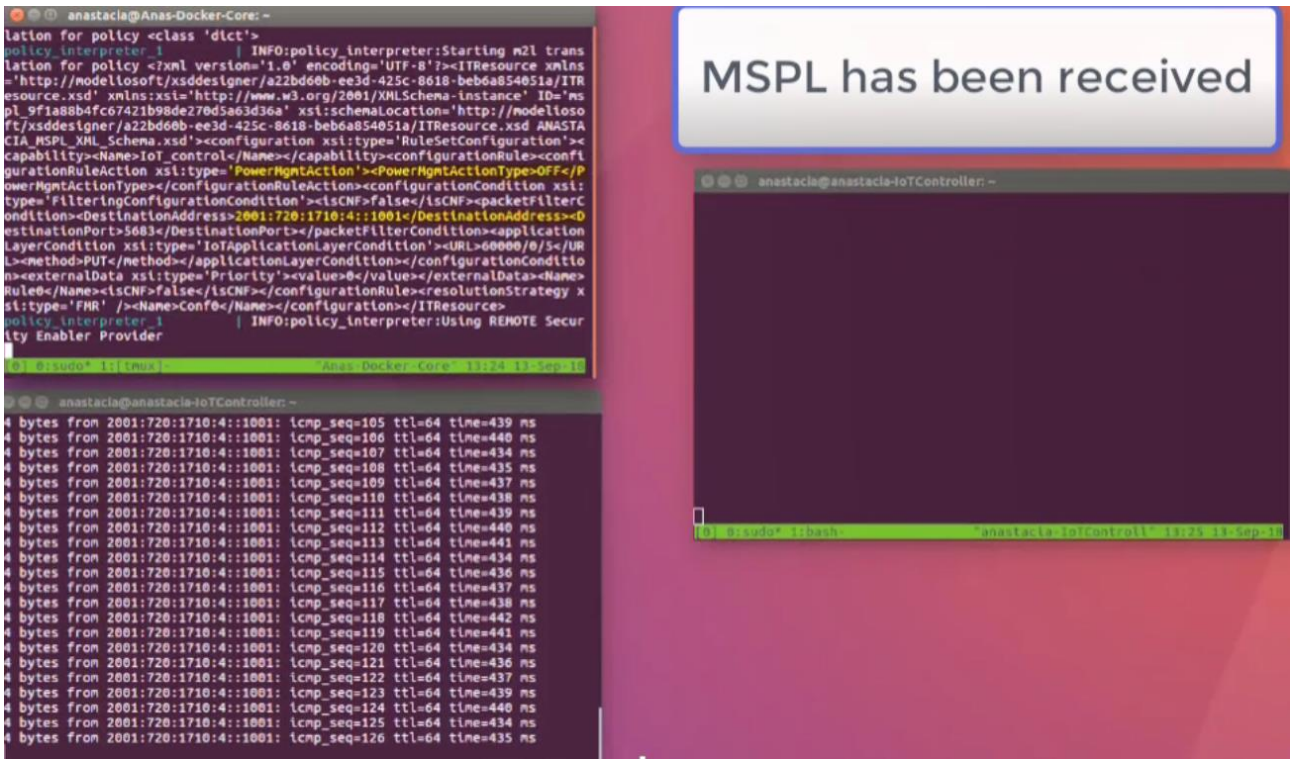


Figure 48: MSPL file translation from Policy Interpreter

In Figure 49, the top-left console shows the configuration reception by IoT controller which indicates the IPv6 address of the compromised IoT device and the COAP resource to send a message of powering off.

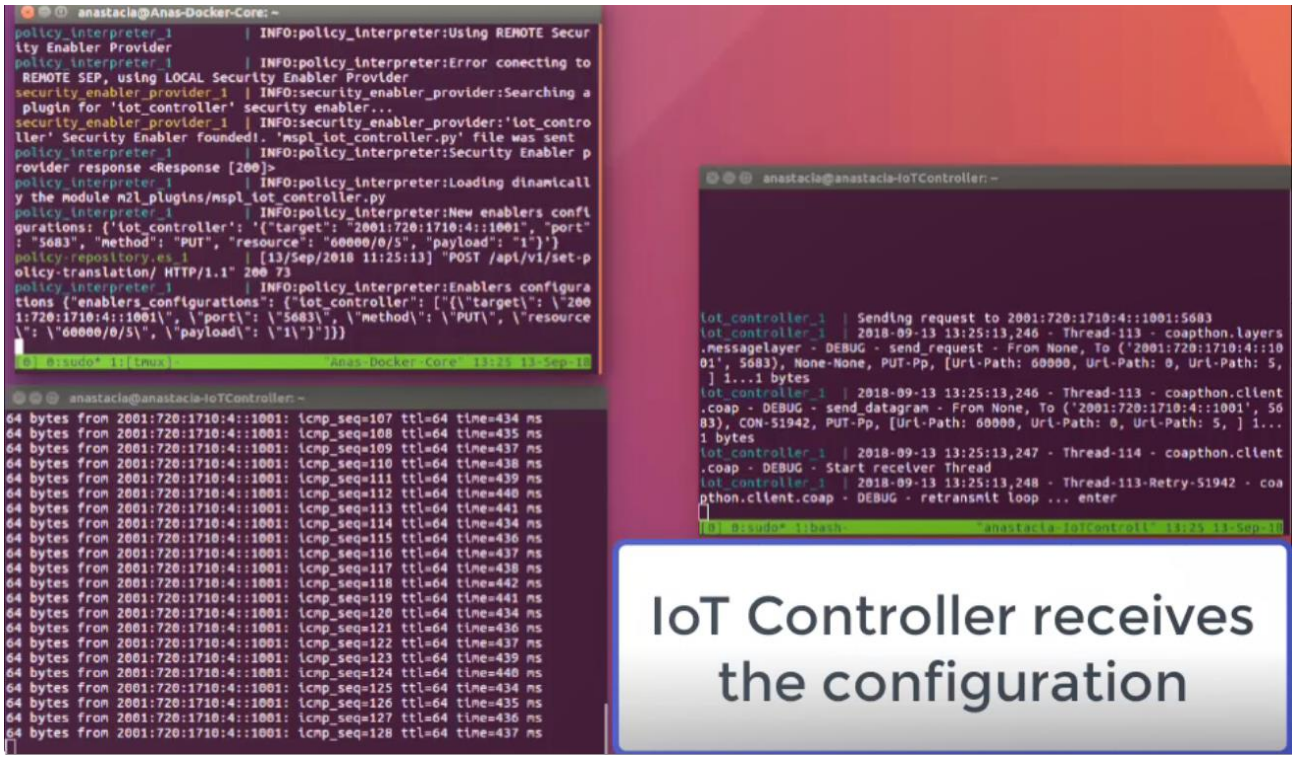


Figure 49: Configuration reception in IoT Controller to power off the IoT device

Once the IoT controller sends the COAP message with the POWER-OFF query, the IoT device is not reachable. Figure 50 shows this result by using ping command from IoT controller to IoT device with its IPV6 address.

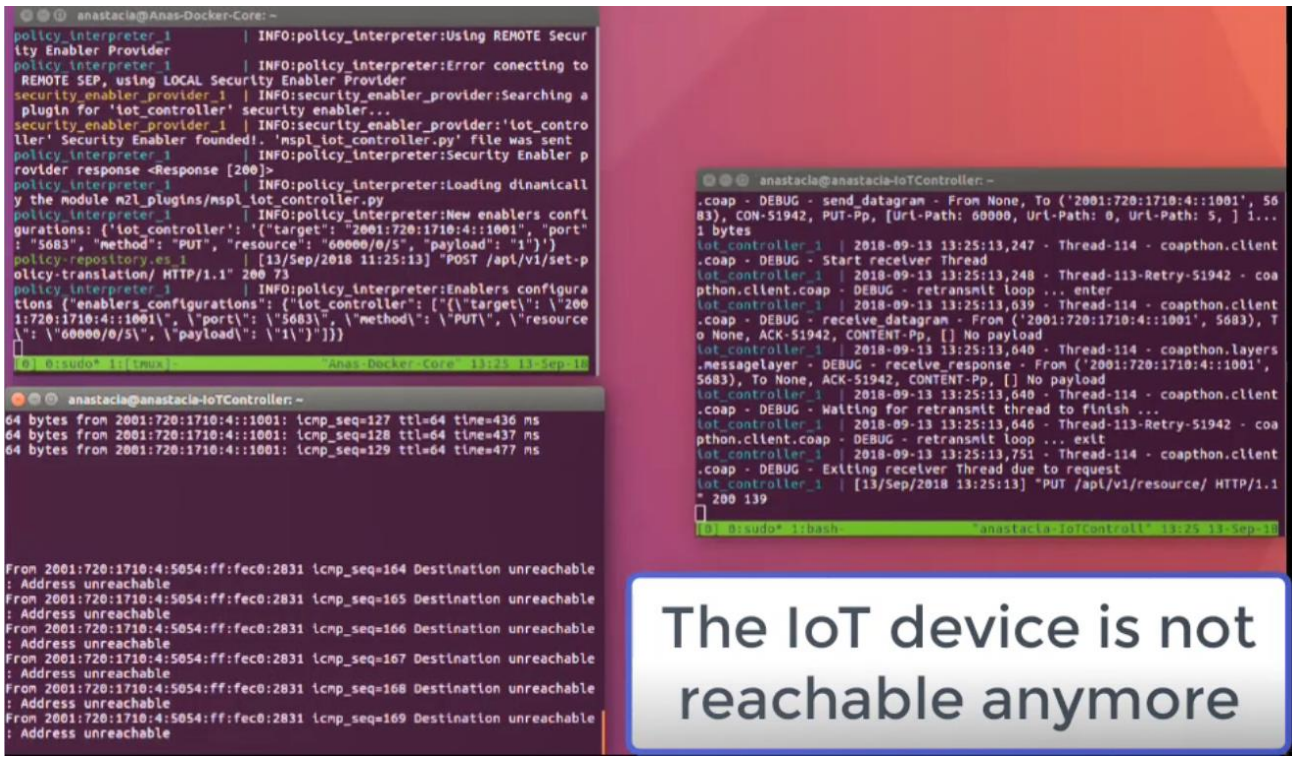


Figure 50: Generated power off action on IoT Controller

Lastly Figure 51 illustrates IoT device view where after reception of CoAP power off message, the device starts shutdown procedure and turns itself off.

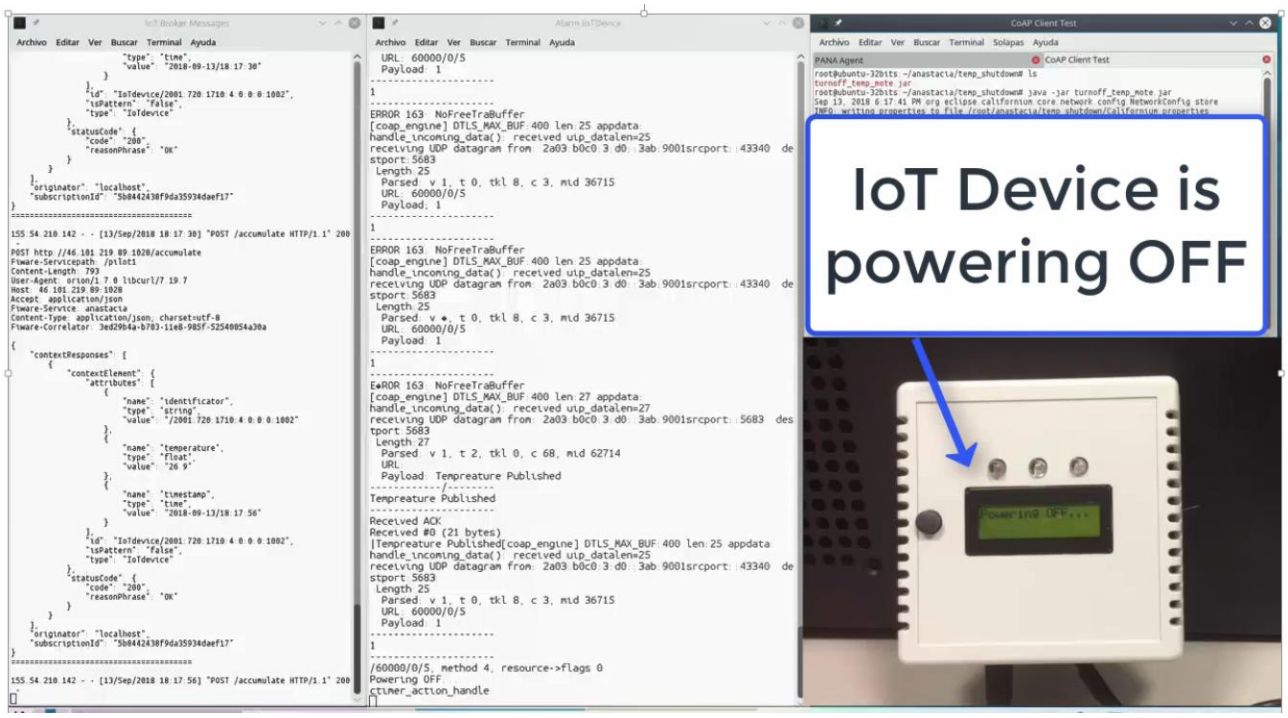


Figure 51: Executed power off action on IoT device

3.3.2 Results of Test-Cases with KPIs and Data Captured

Based on D6.2, the validation of BMS.4 test cases were performed and related KPIs were measured. Below all test cases and metrics are included with results and explanations.

- **TC_BMS.4.1 – Connection with OdinS IoT-broker**

TC_BMS.4.1	Connection with OdinS IoT-broker
Execution	· Start all the components from above in that order · Observe JSON messages coming from OdinS IoT-broker on Kafka Broker being logged on Data Analysis
Expected results	· Data Analysis is consuming physical values from OdinS IoT-broker via Kafka Broker · Data Analysis is storing data to a dataset correctly
KPI(s)	1. Time to activate Data Analysis 2. Time to consume and store information from Kafka broker 3. Time to publish data from IoT-Broker to Kafka broker

The KPI values measured for the data analysis initialization and the data collection from IoT-broker and Kafka-broker is shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,014	0,015	0,013	0,014	0,016	0,0144
2. Time (seconds)	0,156	0,156	0,156	0,156	0,202	0,1652
3. Time (seconds)	0,1628	0,176	0,1765	0,1876	0,169	0,17427

These 3 KPIs are described in details as follows:

1. The measure time refers to Data Analysis component start time. It is measured from starting the component and finishing when agent is ready to receive data and all internal modules are initialized. Data Analysis agent start-up time is minimal and agent requires less than 15 milliseconds to boot up. Since agent is working on continuous basis (24/7 operations mode), the off-line time is minimized to periods where agent needs to be updated. The updates are not happening very frequently and as ANASTACIA framework matures the need for agent restart will be negligible.
2. The time is measured between message reception from Kafka broker and its parsing and processing inside internal buffer. Time required consuming and storing information from Kafka broker per single message is very small. Since agent is written to consume messages in asynchronous fashion system can be scaled up to tens of thousands messages/events per second without much impact on agent internal structure and will solely depend on Kafka broker message processing capacity. Over that limit it is recommended to split events into separate queues on Kafka broker cluster and assign agent to specific physical area. For example higher scale event processing can have larger queues per building, section or area that needs to be monitored. After that agents will provide an independent cover for each area.
3. The IoT Broker is data concentrator of IoT network, as it receives the information from the IoT device, and then sends it to the Kafka broker. In this case, we measure the time it takes from the reception of a message from a data source, to the reception of the ACK of the Kafka Broker that has received this information from the IoT Broker. As above IoT broker to Kafka broker time is minimal especially in case when temperature inside building have very big inertia for temperature change (order tens of minutes). In ANASTACIA framework monitoring is set to report temperature every 5 mins which is enough from temperature measurements perspective. Higher measurement frequency will not provide any additional advantage from monitoring and cyber security perspective.

The data exchange times are also affected by large geographical distribution of ANASTACIA components (Kafka in Greece, IoT-Broker in Spain and Data Analysis in Ireland) thus in the measurements network delay plays significant factor. In real cyber-security deployment, system components will be co-located and data exchange times can be reduced significantly (order of tens of milliseconds).

- **TC_BMS.4.2 – Attack detection based on IoT monitoring data**

TC_BMS.4.2	Attack detection based on IoT monitoring data
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)
KPI(s)	1. Correct detection of malicious activity with precision **(>90%)** 2. Attack detected within 500msec timeframe

The KPIs measured for the attack detection are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Percentage (%)	90,53	90,53	90,53	90,53	90,53	90,53
2. Time (seconds)	0,015	0,015	0,015	0,015	0,015	0,015

Below, these 2 KPIs are explained in detail:

1. This metric represents percentage of correct malicious detections made by Data Analysis agent using Machine Learning technique and sensor data models with historical database.
2. This time refers to amount of time required for Data Analysis agent to detect anomaly after receiving next event. This time is the same as pre-trained model computation for anomaly detection is constant (in principle similar to computing hash function).

Malicious detection performed by Data Analysis model satisfied the required precision for malicious detections in all cases. The false positive were observed during corner cases, where either attacked points were too close to normal behaviour points (order of 10s or 100s of degrees after decimal point) or new normal behaviour of the system was observed in test scenario. In this case, future remedy is to train on larger data sets and fine tune algorithm to provide even higher predictions for system anomalies.

The attack detection was completed within required time. Reason is that regardless amount information fed to model, the validation process in computationally inexpensive thus similar time measurement for attack verdict were observed during test process.

- **TC_BMS.4.3 – Monitoring and Reaction**

TC_BMS.4.3	Reaction to attack on sensor
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
KPI(s)	1. Time to select the security enabler by Security Orchestrator. 2. Time for IoT Control MSPL -> IoT Controller policy translation. 3. Time to generate the enforcement by Security Orchestrator.

The KPIs measured for the monitoring and reaction phase are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	1,62473	1,605	1,3731	1,4559	1,593	1,53044
2. Time (seconds)	0.075	0.035	0.035	0.033	0.038	0.04
3. Time (seconds)	5,21946	5,218	5,2008	5,2132	5,206	5,21151

Below, these 3 KPIs are described in detail:

1. Security Orchestrator requests the list of enablers, and then identifies the adequate capability from the Security Enabler Provider component. Moreover, the Security Orchestrator requests the low-level configuration from the Policy Interpreter by sending the MSPL file. The obtained time is a summation of the two precedents actions. Time to select security enabled by SO was below 2 seconds which enabled ANASTACIA framework to react quickly to IoT sensor attack.
2. The Policy Interpreter receives the MSPL and the selected security enabler and it downloads the plugin and it performs the translation. This time depends on the complexity of the security policy, being it also dependent on the plugin implementation. In this case the IoT Controller plugin.
3. Security Orchestrator sends a request with mitigation action configuration to the IoT controller for enforcing the low-security configuration on the suspected IoT device.

- **TC_BMS.4.4 – Security Alert Management**

TC_BMS.4.4	Full BMS.4 scenario test
Execution	· Start all the components as stated from above · Perform attack by running attack script to change sensor temperature · Observe logs on security management components as well as SAS and DSPS
KPI(s)	1. Security Alert visible within 1.5 sec after detection 2. Security and Privacy Seal is updated within 3 sec

The KPIs measured for the security alert management are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,03	0,04	0,03	0,02	0,03	0,03
2. Time (seconds)	2,582	2,48	2,626	2,488	2,491	2,533

Below, the 2 KPIs are described in detail:

1. The measured values refer to the time passing between alerts are received by the SAS (Security Alert Service) and the relative output (enriched alerts) are sent to the (local) RabbitMQ queue shared with the DSPS (Dynamic Security and Privacy Seal).
2. The reason of this relatively slow performance is mainly due to the DSPS storage and its infrastructure. When the DSPS server needs to write the seal on the on-chain storage (blockchain-based with BFT consensus) it takes in average 1.27 seconds. When it needs to write the seal log on the off-chain storage (based on shamir secret schema) it takes in average 0.943 seconds. These times depend a lot from the number of used nodes and their hardware so potentially it could be improved, however we don't have enough resources to update the infrastructure. In next iterations, more functionalities will be added so the total time will not likely decrease. On the other hand we don't expect to increase it a lot since actions not involving the storage don't have major impacts on the global performance.

Security alert visibility notification's time was fast due to simple message passing and processing on the Service Logging UI. The SAS message is formed in such a way that doesn't require any additional processing from SAS component. This time also exceeds expectations for ANASTACIA framework KPI value.

DSPS is updated with much slower time than expected. Reason for that is behind the scenes Hyperledger[2] blockchain must compute new hash that is matching specific criteria (non-repeatable, amount of zeros or other characters in the hash etc.). The computation of a hash is first part of equation while the latter is that the computation is done within distributed manner. Second condition also adding to caused delay by distributing new hash in the network which slows down the process. There are few methods to speed up computation however this might make blockchain more vulnerable to potential crypto attacks in the future. ANASTACIA designers hope is that in time core blockchain technology computation will become less computationally expensive due to Moore's law and advancements in blockchain technologies.

Overall in BMS.4 test case scenario ANASTACIA framework reacted to IoT temperature sensor attack within less than 10 seconds in total. This time can be further reduced by co-locating all SW components of the same computing infrastructure. Main time contributors were found in reaction part. ANASTACIA team identifies that there is a room for additional optimization to ensure faster response in new version of ANASTACIA framework.

[2] https://www.hyperledger.org/

3.4 USE CASE MEC.3: DOS/DDOS ATTACKS USING IOT DEVICES

3.4.1 Use case description + Benefits from ANASTACIA framework

The MEC.3 use case of ANASTACIA copes with a DoS/DDoS attack using Smart Cameras and IoT Devices. In this scenario, a malicious attacker gains control over multiple IoT devices and uses them such as vector attack to damage a Smart Camera inside the same IoT network. The attack is performed by sending numerous ICMP ping packets towards the victim, with the goal of overwhelming the Smart Camera and leaves it without connectivity. The final goal of this attack is to deactivate the video being transmitted by the camera by not allowing the video packets to go out if the camera due to the high amount of ICMP packets received by the victim.

The benefits of ANASTACIA architecture are the detection of malicious traffic in order to filter the involved protocols and stop the attack. In this sense, architecture adds a security layer that constantly monitors the network and detects any DoS attack performed from within the network.

In addition to this, the ANASTACIA platform brings the flexibility of using NFV and SDN techniques to the deployment of the countermeasures. This novelty allows the creation of a new virtual machine instance – using SDN – that hosts the firewall – implemented using NFV techniques. Despite this particular use case deploys a specific NFV controller; the dynamicity of the ANASTACIA platform allows adapting the countermeasures depending on the deployed capabilities and security policies. Using SDN and NFV the ANASTACIA platform can deploy different countermeasures depending on these constraints to mitigate the attack.

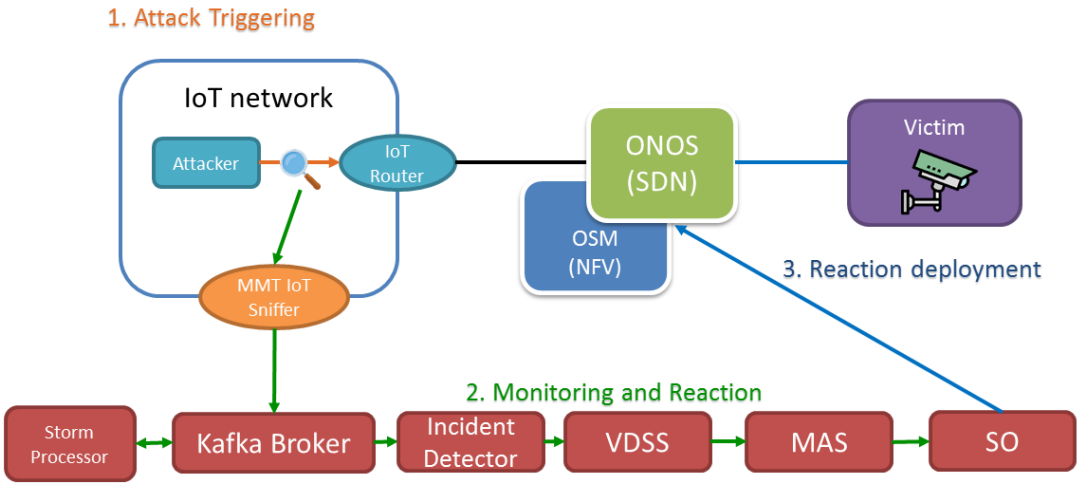


Figure 52: Scenario Deployment for MEC.3

Figure 52 shows the implemented deployment for this attack. The IoT network has been emulated using the Cooja software, which is part of the Contiki environment[3]. This emulation contains an emulated compromised device (in a first instance), an RPL router (to allow traffic to go outside the IoT network) and an instance of MMT-IoT-Sniffer; a modified Contiki device that allows extracting the packets form the network to be analysed by MMT-Probe.

3.4.1.1 Complete flow of use case MEC.3

The complete flow of MEC.3 use case is divided into 3 phases:

1. Within IoT network, attacker compromises an IoT device to start a DoS attack sending ICMP packets towards a victim.

[3] http://www.contiki-os.org/

2. Attack detection is made by the combination of MMT-IoT-Sniffer and MMT-Probe to analyse the traffic inside the IoT network. The threat notification is processed by monitoring components to determine a security alert and generates a mitigation reaction to avoid the DoS attack.
3. Reaction deployment is coordinated by Security Orchestrator in cooperation with SDN controller (ONOS) and NFV controller (MANO).

In the next subsections, these different stages of the MEC.3 use case are described in detail.

3.4.1.1.1 Attack Triggering

The first stage includes the activation of the attack script in the compromised device. Following the deployment shown in Figure 53, the first phase will involve the emulated IoT network (containing the attacker and the RPL router) and the emulated Smart Camera acting as the victim. The communication between these two machines is provided by ONOS – an SDN provider – in order to transfer the packets between the attacked and the victim.

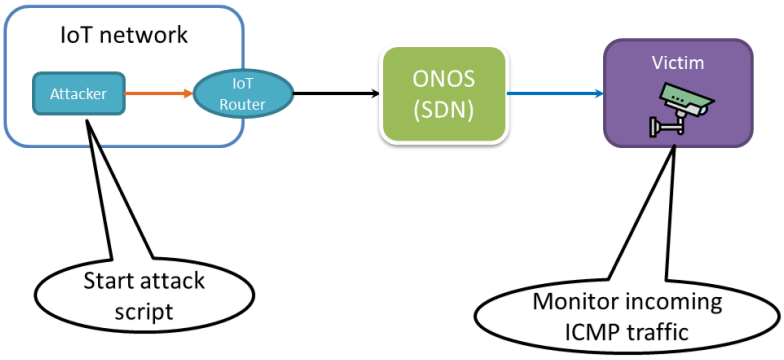


Figure 53: Deployment for Attack Triggering for MEC.3

Once the attack has been started, the attacker will send a burst of 10 ICMP pings packets to the victim. Since the packets are sent as s burst (i.e. as quick as possible), the traffic should be classified as an attack, according to the specifications given in D2.4. Finally, and in order to see the attack and the correct mitigation of the attack, the *tcpdump* tool will be used on the victim with the goal of following the incoming packets on the victim, therefore being able to watch how the victim receives the attack and how it is blocked by the ANASTCIA deployment.

Figure 54 is a screen captured from the MEC.3 video showing the attack triggering of ICMP flooding from a IoT network towards the a victim. The *tcpdump* tool is running in the bottom console at the right side to show the ICMP messages received by the victim. The monitoring agent called MMT-probe is running in the top console at the right side to sniff the network traffic in order to detect and notify the DoS attack based on ICMP messages.



Figure 54: Attack triggering of ICMP flooding for MEC.3

3.4.1.1.2 Monitoring and Reaction

The second phase of the use case comprises the Monitoring and Reaction stages. Figure 55 shows the interactions on this phase, which starts right after the last one (i.e. attack triggering) has finished.

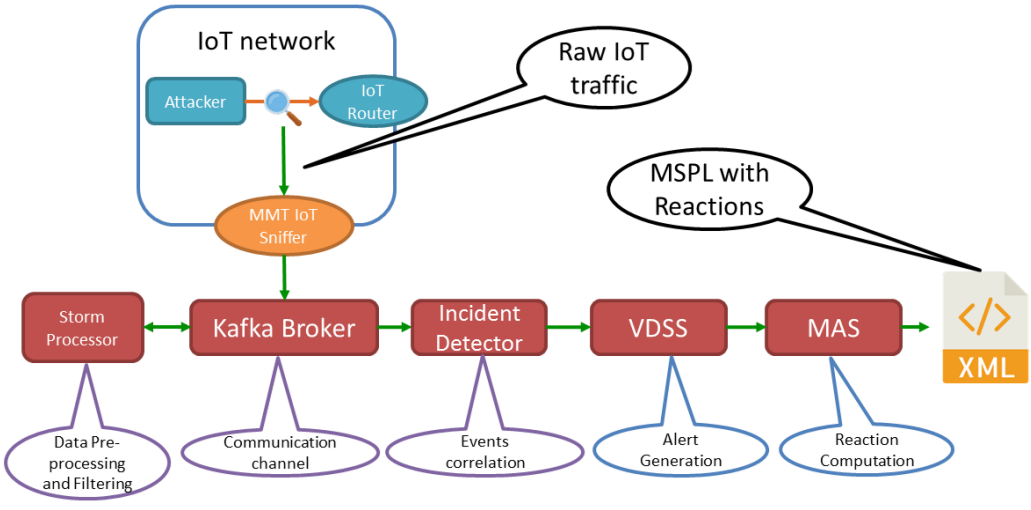


Figure 55: Monitoring and Reaction Phase for MEC.3

The phase starts with the execution of the MMT-IoT-Sniffer software, which will enable the extraction of the packets from the (emulated) IoT network. These packets will be analysed by an MMT-Probe instance, which will generate periodic network statistics reports and, eventually, security reports regarding any detected attack.

In the ongoing scenario, the malicious packets inserted by the attacker will be detected by MMT-Probe, which will generate a respective security report and send it to the whole ANASTACIA chain (shown in red colour in Figure 55) using the Kafka Broker.

The deployed platform will then re-format the MMT report (using the Storm processor) and return it to the Kafka Broker, where it will be read by the Incident Detector. This component will perform the event correlation and send the data to the VDSS which will raise the alert about the detected attack. The alert will then be sent to the MAS, which will read the alert and generate the corresponding MSPL file containing the countermeasures to be deployed on the monitored network.

For the MEC.3 use case, the adopted countermeasure specifies a filtering of the ICMP protocol. Figure 56 shows the specific syntax of the generated MSPL file with XML-based language.

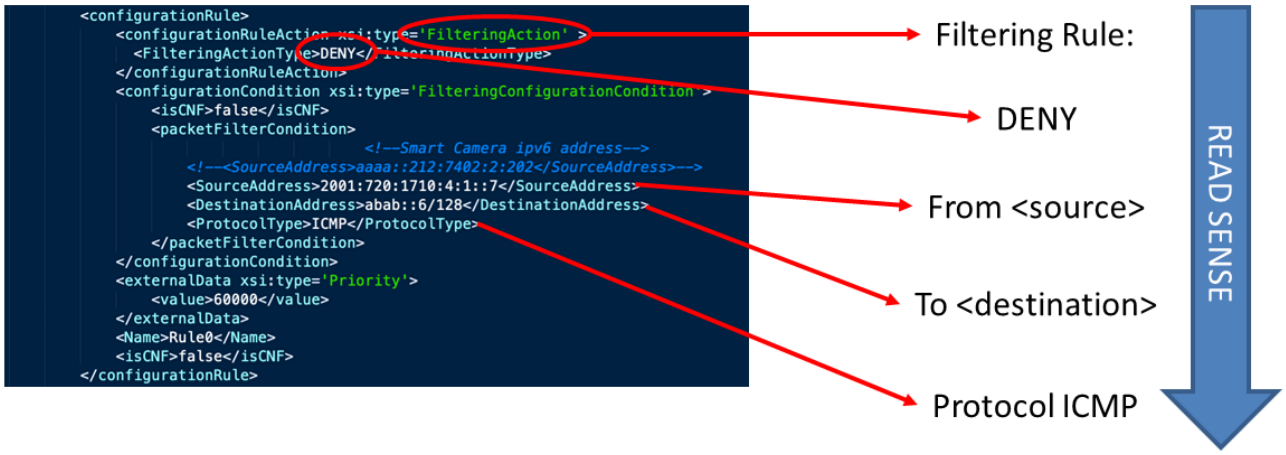


Figure 56: Generated MPSL File for MEC.3

The configuration shown in Figure 56 specifies a filtering rule affective all the traffic coming from a specific source towards a specific destination, concerning the specific ICMP protocol.

The next figures show the data captured (logs and screens) from the monitoring and reaction components to perform the 2 main interactions of this phase:

1. Figure 57 shows the web XL-SIEM interface of the Incident Detector to correlate the threat events indicating the IPv6 addresses of attack source and victim destination. In particular, Incident Detector shows the attack notified from MMT-Probe with the name “ICMP Ping DoS” towards the victim in the IPv6 address.

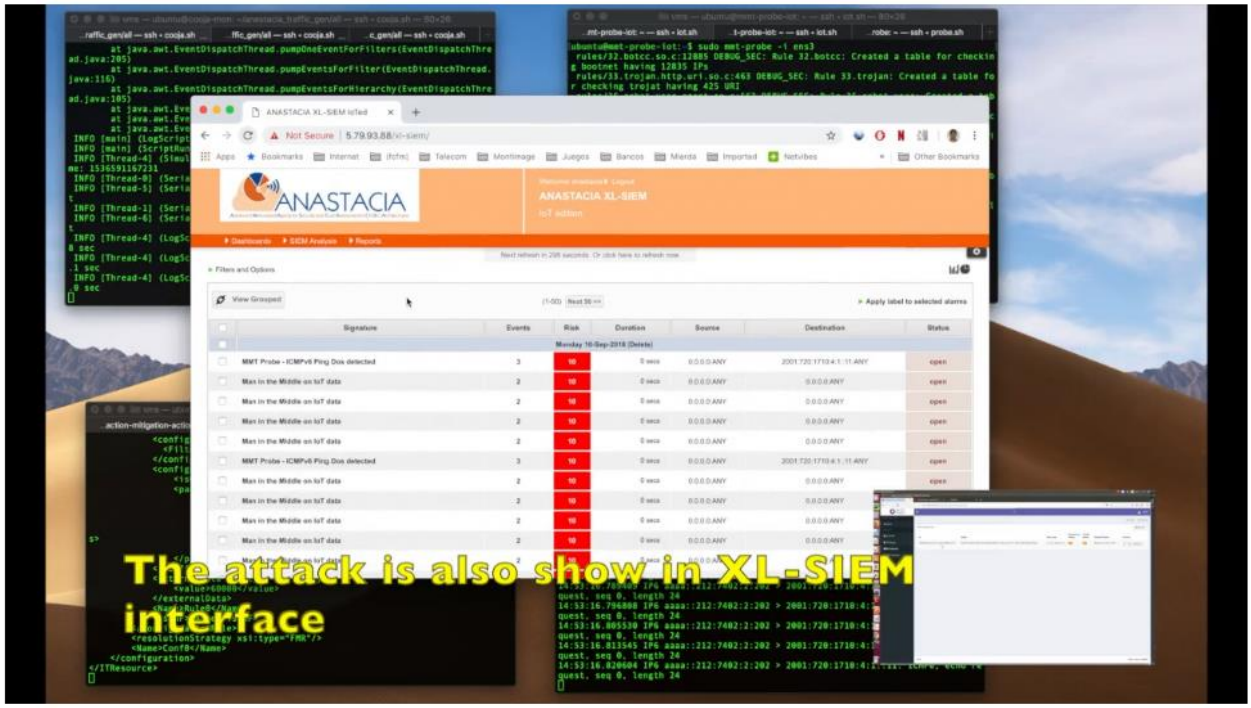


Figure 57: Events correlation in Incident Detector for MEC.3

2. Figure 58 shows the bottom-left console of Mitigation Action Service where the alert is processed and the abovementioned MSPL file is generated to be sent to Security Orchestrator.



Figure 58: MSPL generation in Mitigation Action Service for MEC.3

3.4.1.1.3 Reaction Deployment

The last phase comprises the deployment of the reaction previously computed by the MAS. This reaction (in form of an MSPL file) is received by the Security Orchestrator (SO) in order to start the orchestration process that will deploy the countermeasure.

Figure 59 shows the process followed by the SO to deploy the countermeasure. This phase starts when the Security Orchestrator receives the MSPL file from the Mitigation Action Service, which triggers the deployment of the countermeasures. Upon received the MSPL file, the SO contacts Security Enablers Provider and gets a list of the available Security Enablers capable of deploying the countermeasure. The SO then selects one and contacts the translation service to transform the MSPL file into lower level language – i.e. security enabler specific configurations – that represent the countermeasures specified in the MSPL file. With this translation, the SO is ready to start the orchestration process.

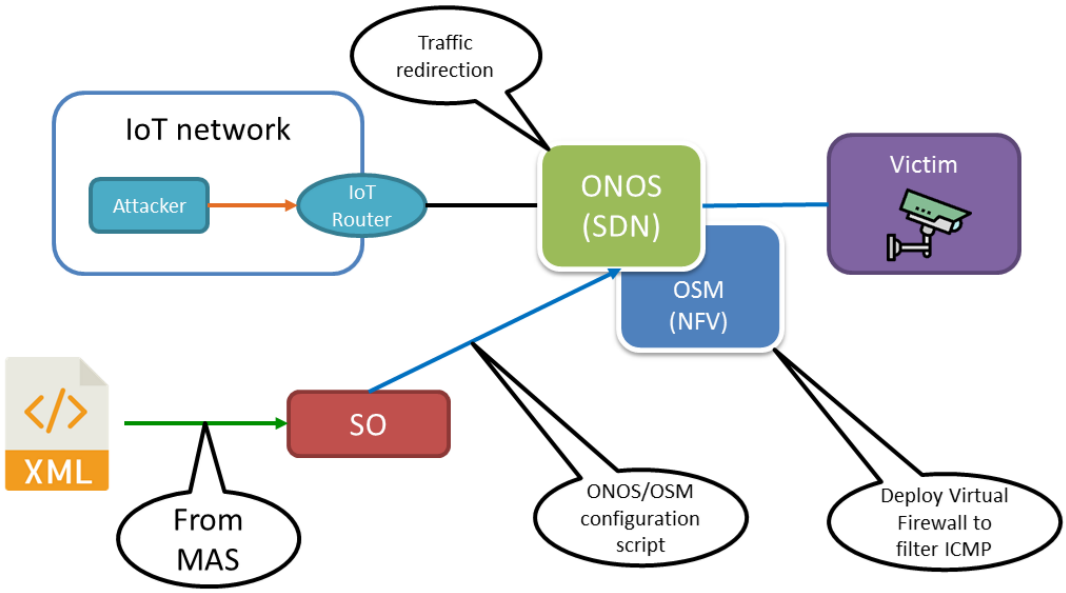


Figure 59: Deployment of the Reaction for MEC.3

The SO starts the orchestration process by executing the low-level configurations specified in the translated MSPL file. In the MSPL file for the MEC.3 use case, the SO needs to: (1) create a new Virtual Machine instance which will contain the Virtual Firewall; (2) deploy ONOS intents using NFV techniques to filter the ICMP protocol between the specified IP addresses; and (3) using SDN techniques, redirect all the traffic between the specified addresses through the recently-created Virtual Firewall.

Once these tasks have been performed by the SO, the traffic is effectively being re-routed and actively filtered by the new firewall. This has the immediate effect of filtering all the ICMP packets between the attacker and the victim, which effectively mitigates the on-going attack.

The next figures show the data captured (logs and screens) from the Orchestration and Enforcement components to perform the main interactions of this phase:

1. Figure 60 shows the web interface captured from the MANO NFV Controller when it receives the query of a new VNF instantiation of a Virtual Firewall.

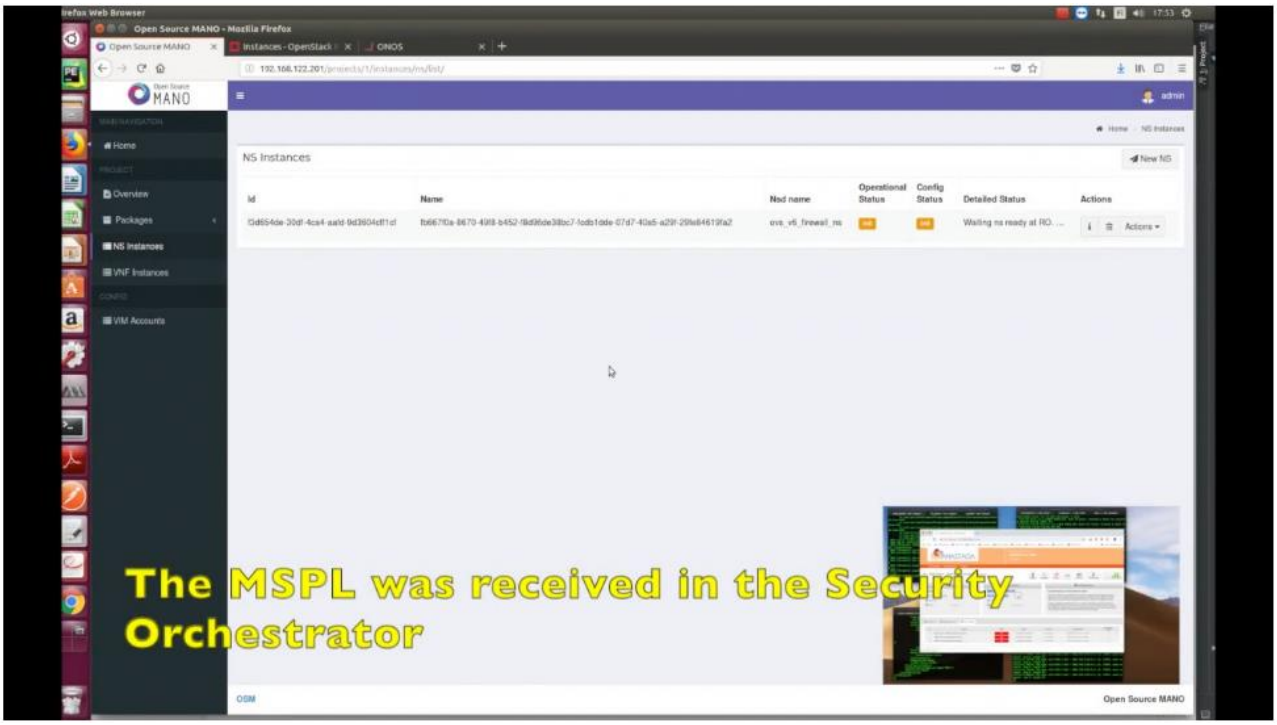


Figure 60: Reception of MSPL file in Security Orchestrator for MEC.3

2. Figure 61 shows the new VM instance created in the web interface of VNF controller (MANO) to deploy the Virtual Firewall as countermeasure.

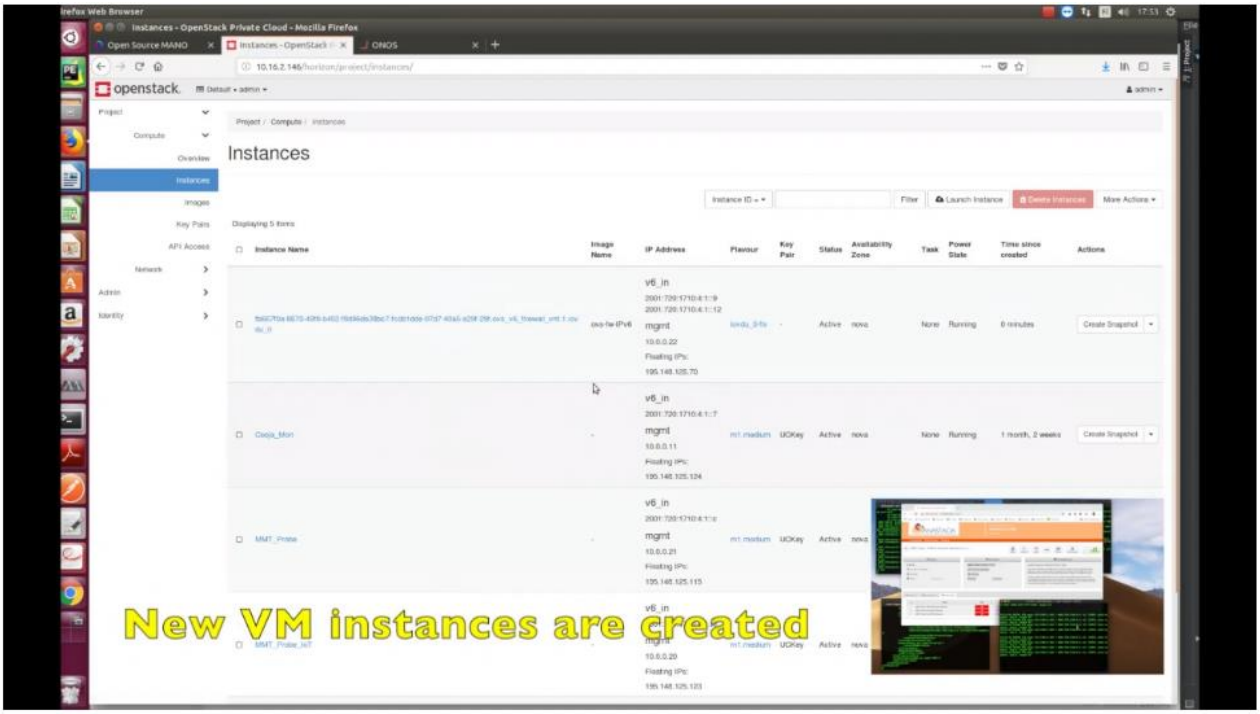


Figure 61: VNF instantiation of Virtual Firewal for MEC.3

3. Figure 62 shows the network management interface of the SDN controller (ONOS) where is represented the traffic filtering from the attacker towards the new VNF instance of Virtual Firewall.

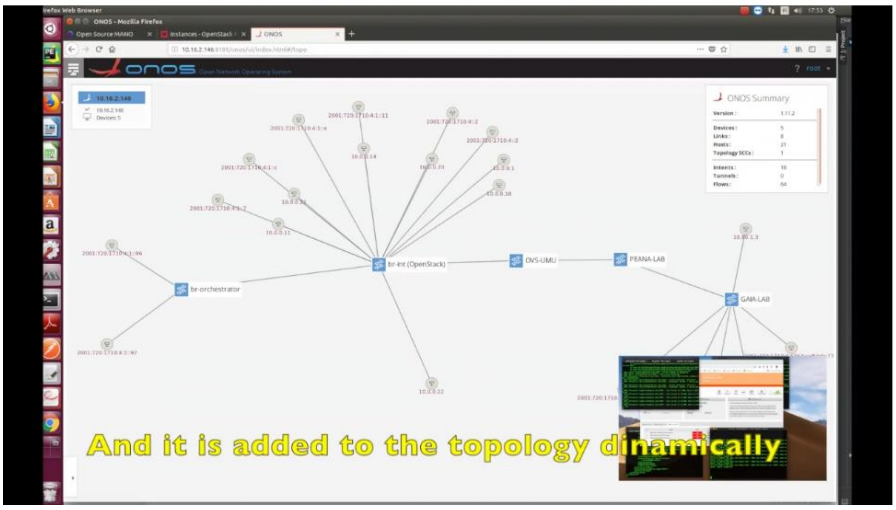


Figure 62: Network management inferface of SDN controller with Virtual Firewall for MEC.3

4. Figure 63 shows the attack is effectively mitigated, after a few seconds of the countermeasures deployed. The ICMP Ping messages from the attacker are now filtering. So the attacker (at the top-left console) continues sending malicious messages that are blocked to not arrive to the victim (at the bottom-right console).

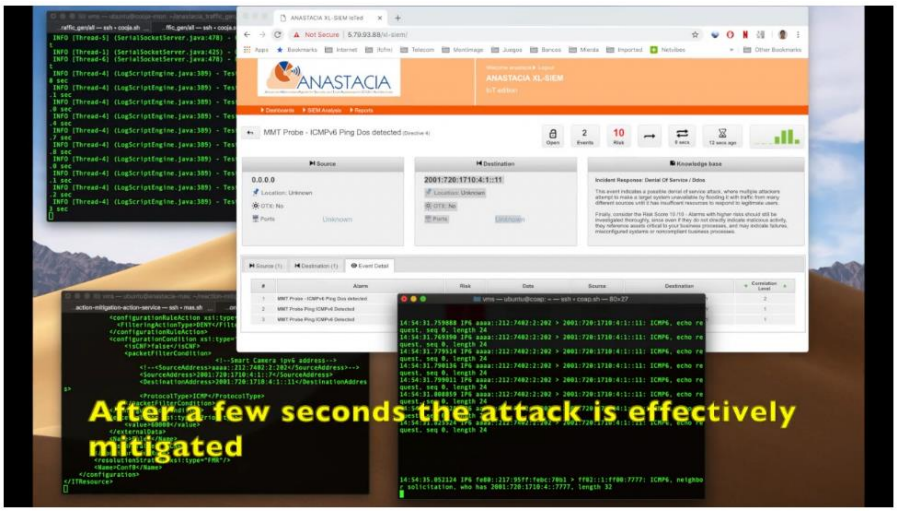


Figure 63: ICMP DoS is blocked and the attack is mitigated for MEC.3

3.4.2 Results of Test-Cases with KPIs and Data Captured

The following subsection shows the measured KPIs of the validation executions according to the test cases defined in the deliverable D6.2. Below, the tables show the results of test-cases and also the KPI measurements in seconds for 5 executions of the performance tests.

- **TC_MEC.3.1 – Attack generation**

TC_MEC.3.1	Device actuation
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.
KPI(s)	· Generating 10 ICMP packets on the network every 10 seconds (pass/fail)

This test case aims to fulfill the functional of generating an attack on the emulated network.

Execution Num.	1	2	3	4	5	Mean
Verification	PASS	PASS	PASS	PASS	PASS	PASS

- **TC_MEC.3.2 – Monitoring processing**

TC_MEC.3.2	Monitoring processing
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.
KPI(s)	1. Detection of the attack is done within 1 sec. 2. The Kafka broker receives the MMT-Probe report (pass/fail)

The KPI measured for the attack detection of ICMP DoS for the MMT-Probe notifying Kafka-Broker is shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,56	0,534	0,431	0,504	0,483	0,5024
2. Verification	PASS	PASS	PASS	PASS	PASS	PASS

These time values are computed in the attacker script and the MMT-Probe. In one side the starting timestamp is computed with the attack script and on the other side, the detection timestamp is extracted from the MMT-Probe report when the trigger of the security rule is checked. Both the attack script and the MMT-Probe instance were run on the same machine to avoid clock synchronization issues. Finally, the corresponding MMT-Probe report is observed in the network.

In this test case, it verifies that MMT-Probe sends correctly a security report to the Kafka broker as soon as the attack is detected. This report must be observed in the broker containing the detection timestamp, which is part of the MMT-Probe report.

- **TC_MEC.3.3 – Reaction processing**

TC_MEC.3.3	Reaction processing
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.
KPI(s)	1. The VDSS sends the information to Security Alert Service within 2 sec. 2. MAS computes the MSPL file within 1 sec 3. The MSPL is sent to the Security Orchestrator (pass/fail)

The KPIs measured for the reaction phase are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	0,042	0,043	0,043	0,042	0,042	0,0424
2. Time (seconds)	0,43	0,408	0,42	0,442	0,395	0,419
3. Verification	PASS	PASS	PASS	PASS	PASS	PASS

Below, the 3 KPIs are described in detail:

1. The current version of the VDSS (Verdict Decision Support System) provides a limited functionality with a simple risk evaluation and a direct selection of the mitigations. A statistical based risk evaluation and automatic decision of the mitigation is expected in the second period of the development, which might increase the current response times of this component.
2. These values are computed inside the MAS (Mitigation Action Service), calculating the time (in seconds) used by the MAS to compute the MSPL file since it receives the alert from the VDSS). These values might increase during next development iteration due to more complex computations to generate the MSPL file.
3. The MAS sends the MSPL file to the SO using an HTTP communication. The transfer of the MSPL file automatically acknowledged by the HTTP protocol, returning a code 200-OK when the transfer has been successful.

- **TC_MEC.3.4 – Seal processing**

TC_MEC.3.4	Seal processing
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
KPI(s)	· Time to show a security attack in the user interface of DSPS

The KPI measured for the seal management is shown in the next table:

Execution Num.	1	2	3	4	5	Mean
Time (seconds)	2,592	2,488	2,633	2,497	2,497	2,541

From these measures, we can see that it takes less than 3 seconds to notify the users from the DSPS-GUI including the time needed to update the seal/log in the storage (See TC_BMS.3.5). This is due to the fact that the DSPS-GUI backend is hosted on the same machine of the DSPS server and the users are notified by WebSocket connection from the backend. In next iterations this measure could slightly increase due to a possible migration of the GUI backend to another Virtual Machine.

- **TC_MEC.3.5 – Orchestration and enforcement processing**

TC_MEC.3.5	Orchestration processing
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
KPI(s)	1. Time to select the security enabler and receive the configuration in the orchestrator 2. Time for Filtering MSPL -> OVS policy translation. 3. Time to generate the instructions to VNF controller and SDN controller from the orchestrator

The KPIs measured for the orchestration and enforcement of traffic filtering are shown in the next table:

Execution Num.	1	2	3	4	5	Mean
1. Time (seconds)	1,62473	1,605	1,3731	1,4559	1,593	1,53044
2. Time (seconds)	0.044	0.03	0.029	0.032	0.031	0.033
3. Time (seconds)	101,772	107,2	110,12	108,46	118,5	109,214

Below, the 3 KPIs are described in detail:

1. Security Orchestrator requests the list of enablers, and then identifies the adequate capability from the Security Enabler Provider component. Moreover, the Security Orchestrator requests the low-level configuration from the Policy Interpreter by sending the MSPL file. The obtained time is a summation of the two precedents actions.
2. The Policy Interpreter receives the MSPL and the selected security enabler and it downloads the plugin and it performs the translation. This time depends on the complexity of the security policy, being it also dependent on the plugin implementation. In this case the OVS Firewall plugin.
3. The Security Orchestrator sends a request to the NFV controller to create and deploy a VNF instance (OVS-Firewall), then the SDN controller steers the traffic through the deployed OVS-Firewall. The computed time represents the summation of these two actions.

3.5 RESULTS OF END-USER EVALUATION

This section presents the main results obtained from the end-users evaluation of ANASTACIA framework according to the use case implemented. The results are meant to help the Consortium in the prioritization of functionalities and features to must be improved in the next cycle of ANASTACIA development.

3.5.1 Evaluation Questionnaire

This end-user questionnaire allows evaluating the first release of the integrated ANASTACIA framework according to 4 different use cases in 2 application domains: **Building Management System** (BMS) and **Mobile Edge Computing** (MEC). The first ANASTACIA release integrates monitoring, reaction, orchestration and enforcement.

To facilitate the evaluation carried out by end-users, five videos have been provided. The videos show the execution of the first release of integrated ANASTACIA framework according to four use cases defined in D6.2, so as the demonstration of seal management module that will be integrated in the next cycle of ANASTACIA development.

- Insider attack on the fire suppression system (BMS.2).
  https://www.youtube.com/watch?v=RPByyMMfQuI
- Remote attack on the building energy microgrid (BMS.3).
  https://www.youtube.com/watch?v=iVKECGs2buw
- Cascade Attack on a Megatall Building (BMS.4).
  https://seafile.gruppoitaleaf.com/f/d07022a0c6/
- DoS/DDoS attack using smart cameras and IoT devices (MEC.3).
  https://www.youtube.com/watch?v=pdHBbK0_CPM
- Dynamic Security and Privacy Seal (DSPS).
  https://www.youtube.com/watch?v=opb8fWSP_-w

After seeing the videos, the end users evaluate the results and provide their feedback in the following questionnaire divided into three parts:

- The first part contains general questions that have been prepared in accordance to the end-users requirements defined in deliverable D1.2. In addition, few more questions were added to map the measured KPIs with the end-users feedback.
- The second part contains specific questions about the seal management plane (DSPS) that is been developing to be integrated to ANASTACIA framework in the next cycle of development.
- The third series of questions is created separately against each integrated use case to better reflect specific test conditions and features that the proposed scenario is validating.

Below, a set of general and specific questions were provided to end-users to gather their feedback and evaluation.

3.5.1.1 General questions

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Is the ANASTACIA framework **easy to use**?	
2	Does the ANASTACIA framework have **intuitive user interfaces**?	
3	Does the ANASTACIA framework provide **real-time feedback**?	
4	Does the ANASTACIA framework provide **powerful reporting**?	
5	Is the ANASTACIA framework **modular and scalable**?	
6	Does the ANASTACIA framework provide **automatic reactions to threats**?	
7	Are you agreed with the **response time of monitoring module**?	
8	Are you agreed with the **response time of reaction module**?	
9	Are you agreed with the **response time of orchestration module**?	
10	Are you agreed with the **response time of enforcement module**?	

1. **What do you like the most about the ANASTACIA framework?**

Free answer to be provided

2. **What do you dislike the most about the ANASTACIA framework?**

Free answer to be provided

3.5.1.2 Specific questions of Dynamic Security/Privacy Seal (DSPS) Management

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	DSPS management based on blockchain technique makes you feel more protected against cyber-attacks?	
2	Do you agree that the web user interface of DSPS is easy to understand?	
3	Do you agree that the DSPS management is too complex to be fully appreciated?	
4	Do you agree that the DSPS management provides a powerful reporting about real-time indicators of the overall security/privacy level?	

3.5.1.3 Specific questions for Use Case BMS.2: Insider Attack on the Fire Suppression System (Unauthorized Device Access)

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	Network authentication of new IoT/CPS devices using secure bootstrapping make you feel more protected against cyber-attacks?	
2	Distributed capability-based access control for IoT resources make you feel more protected against cyber-attacks?	
3	Kafka broker of monitoring module provides alert information that is easy to understand?	
4	Security alert service of reaction module provides a powerful reporting about the threat detection and the mitigation MSPL file?	
5	Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated?	
6	Do you agree that IoT network simulation and traffic forwarding by SDN/NVF Controllers are useful countermeasures?	

3.5.1.4 Specific questions for Use Case BMS.3: Remote Attack on the Building Energy Microgrid (SQL Injection)

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	The attack of SQL injection included in COAP message is easy to understand?	
2	Do you agree that Deep Packet Inspection of monitoring module is a powerful tool to detect a SQL injection attack?	
3	XIEM-tool of monitoring module provides SQL alert notification that is easy to understand?	
4	The log console of Mitigation Action Service for monitoring module is enough to understand what is happening?	
5	Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated?	
6	Do you agree that traffic filtering by SDN/NVF Controllers is a useful countermeasure?	

3.5.1.5 Specific questions for Use Case BMS.4: Cascade Attack on a Megatall Building (Temperature Sensor Manipulation)

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	The attack of temperature sensor manipulation is easy to understand?	
2	Do you agree that Data Analysis based on Machine learning for monitoring module is a powerful tool to detect data manipulation?	
3	Kafka-broker of monitoring module provides alert notification that is easy to understand?	
4	Security alert service of reaction module provides a powerful reporting about the threat detection and the mitigation actions?	
5	Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated?	
6	Do you agree that the enforcement of turning off the sensor device by IoT controller is a useful countermeasure?	

3.5.1.6 Specific questions for Use Case MEC.3: DoS/DDoS Attacks using IoT Devices

Objectives	Question: Can you please rate from 1 (low) to 5 (high) the relevance of the objectives?	Assessment level (1-5)
1	The attack of DDoS generated by the simulation of a sensor network sending ICMP messages is easy to understand?	
2	Do you agree that Deep Packet Inspection for monitoring module is a powerful tool to detect DDoS attacks?	
3	XIEM-tool of monitoring module provides DDoS alert notification that is easy to understand?	
4	Security alert service of reaction module provides a powerful reporting about the threat detection and the mitigation actions?	
5	Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated?	
6	Do you agree that the filtering the ICMP traffic from the sensor network by SDN controller is a useful countermeasure?	

3.5.2 Results of End-Users Questionnaire

Interviewees were asked to rate relevant features to be considered by ANASTACIA partners in the second cycle of framework development. End-users rated each question from 1 to 5 in accordance to a Liker scale:

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

The results of general questions show that end-users are partially agreed with all main features of ANASTACIA framework as shown the following table. Although, the easy use of ANASTACIA framework obtained the worst result. Meanwhile, the automatic reaction to threats was the most appreciated by the end-users.

Results of General Questions about ANASTACIA framework	MEAN	DEVIATION
1. Easy to use	3,461	0,9341823
2. Intuitive user interfaces	3,663	0,9482306
3. Real-time feedback	4,139	0,7338593
4. Powerful reporting	3,969	0,8587863
5. Modular and scalable	4,076	0,820435
6. Automatic reactions to threats	4,522	0,6562033
7. Response time of monitoring module	4,014	1,0883958
8. Response time of reaction module	4,043	1,2334641
9. Response time of orchestration module	4,135	1,3574919
10. Response time of enforcement module	4,088	1,5762936

For the DSPS evaluation, the results indicate that end-users are partially satisfied with the web user interface and reporting indicators. Moreover, the end-users are neutral with the use of blockchain in the DSPS management and are partially disagreed that the DSPS is too complex.

Results of Specific Questions about DSPS Module	MEAN	DEVIATION
1. DSPS based on blockchain makes you feel more protected	3,511	1,0128034
2. Web user interface of DSPS is easy to understand	4,239	0,8981518
3. DSPS is too complex to be fully appreciated	2,457	1,0286712
4. DSPS provides a powerful reporting about real-time indicators	4,084	0,6603119

According to the evaluation of BMS.2 use case, the next table indicates that end-users are partially agreed with the protection provided by secure bootstrapping and distributed access control of IoT resources. Moreover, end-users are partially satisfied with the ANASTACIA components such as Kafka broker, Security Alert Service and SDN/NVF Controllers.

Results of Specific Questions about BMS.2 Use Case	MEAN	DEVIATION
1. Network authentication using secure bootstrapping makes you feel more protected	3,867	1,1171463
2. Distributed access control for IoT resources makes you feel more protected	3,663	0,9864918
3. Kafka broker of monitoring module provides alert information that is easy to understand	3,679	0,9603531
4. Security alert service of reaction module provides a powerful reporting	3,881	0,7382432
5. Mitigations actions managed by the Security Orchestrator are too complex	2,998	1,0572622
6. IoT network simulation and traffic forwarding by SDN/NVF Controllers are useful countermeasures	4,201	0,9525338

Regarding the evaluation of BMS.3 use case, the results show that end-users are partially agreed with Deep Packet Inspection for SQL-injection detection, XIEM-tool for Incident Detector and Traffic filtering as countermeasure. The end-users indicate its neutral opinion about the understandable console of Mitigation Action Service and the complexity of Security Orchestrator.

Results of Specific Questions about BMS.3 Use Case	MEAN	DEVIATION
1. The attack of SQL injection included in COAP message is easy to understand	3,701	1,1048542
2. Deep Packet Inspection of monitoring module is a powerful tool to detect a SQL injection attack	4,02	0,8902943
3. XIEM-tool of monitoring module provides SQL alert notification that is easy to understand	3,755	0,8712932
4. The log console of Mitigation Action Service is enough to understand what is happening	3,509	0,8795386
5. Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated	2,79	1,0413221
6. Traffic filtering by SDN/NVF Controllers is a useful countermeasure	4,318	0,8407218

According to the evaluation of BMS.4 use case, the next table indicate that end-users are partially satisfied with the ANASTACIA components such as Data Analysis Agent based on machine learning, Kafka Broker of Monitoring Module and Security Alert Service for threat reporting. Moreover, the end-users express their neutral position for the complexity of Security Orchestrator.

Results of Specific Questions about BMS.4 Use Case	MEAN	DEVIATION
1. The attack of temperature sensor manipulation is easy to understand	4,068	0,9998606
2. Data Analysis based on Machine learning is a powerful tool to detect data manipulation	3,939	0,8992535
3. Kafka-broker of monitoring module provides alert notification that is easy to understand	3,939	0,8154375
4. Security alert service provides a powerful reporting about the threat detection	3,937	0,6584405
5. Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated	2,83	1,0534558
6. The enforcement of turning off the sensor device by IoT controller is a useful countermeasure	3,893	1,0255922

Regarding the evaluation of MEC.3 use case, the next results show that end-users are partially agreed with Deep Packet Inspection for DDoS detection, XIEM-tool for Incident Detector and Traffic filtering as countermeasure. The complexity of mitigation actions managed by Security Orchestrator is considered neutral by the end-users.

Results of Specific Questions about MEC.3 Use Case	MEAN	DEVIATION
1. The attack of DDoS generated by the network simulation with ICMP messages is easy to understand	4,164	1,1493945
2. Deep Packet Inspection for monitoring module is a powerful tool to detect DDoS attacks	4,002	0,9694761
3. XIEM-tool of monitoring module provides DDoS alert notification that is easy to understand	3,761	0,8666695
4. Security alert service provides a powerful reporting about the threat detection	3,96	0,8475656
5. Mitigations actions managed by the Security Orchestrator are too complex to be fully appreciated	2,817	1,1139065
6. The filtering the ICMP traffic from the sensor network by SDN controller is a useful countermeasure	4,198	0,8714848

3.5.2.1 Main highlights

Main valuable highlights from the collected questionnaires are reported here according to the most appreciated aspects and the most unsatisfied ones.

1. **What do you like the most about the ANASTACIA framework?**

- The overall idea for mitigation based on predefined policies. The scenarios seemed to work properly.
- Ambitious goal to manage security aspects by detecting, monitoring and reacting to the identification of vulnerabilities or attacks. Also, the integration of different components towards this end.
- The IoT network simulation and the respective traffic forwarding when an attack occurs.

- It provides understandable user interfaces that can easily be used by non-specialized users.
- Network simulation and traffic forwarding.
- It's very useful for privacy and security concerns.
- The end-user interfaces that facilitate understanding of the different modules of actions.
- You can dynamically and automatically act against different kind of attacks.
- It is an interesting approach to modulate the different actions in a network, from the network access, to the detection of the different possible attacks and alert about them in close real time to notify the correspondent agent to mitigate the attacks by using automation processes.
- Intuitive user interfaces.
- The variety of components which allows for a dynamic setup of the network topology to derive traffic to specific nodes using NFVs, treat security and provide countermeasure for different security threats or attack.
- The novel, interesting and promising way of facing cybersecurity on IoT environments.
- The reaction module that generates and enforces security policies in the system handled by the security orchestration and enforcement planes.
- Its scalability and modularity
- It is an awesome framework for security mitigation relating to both intrusion prevention and detection.
- Fast response time as well as providing with complete countermeasures for the security treats.
- The organization in several planes, each of them dedicated to a particular goal, and interacting each other. It can be useful for a very large number of attacks, with appropriate detection and mitigation methods. The use cases presented here are interesting.
- Its holistic approach on IoT security
- Successful integration of key technologies and standards into a unified cybersecurity framework. From a technology standpoint, the project is challenging, ambitious and state of the art, and of high potential if complexity is properly managed. The technologies are complementary and well positioned to offer adequate monitoring, detection and reaction to cyber-attacks. Especially the SDN/NFV way is used to offer mitigation environment/infrastructure against network level attacks.
- The capacity to provide automatic protection against potential threats.
- Despite the usage of access control and security policy creation, which usually requires some manual tasks to be performed, the ANASTACIA framework seems to work well in terms of automatic detection of threats and, especially, identification of possible countermeasures. Not so sure about real-time detection of possible zero-day attacks, when no previous knowledge of the attack itself is available.
- Its modularity and scalability. The novelty introduced regarding the monitoring of IoT devices.
- The intelligence of the combination of SDN and NFV paradigms. The network is able to detect and stop an attack fastly.
- The automated adaptability through online monitoring and testing techniques.
- Ambitions of the ANASTACIA Framework and the fact it targets actionable results (useful and usable).
- The capability to autodetect attacks and deploys countermeasures. It also isolates the attacker emulating a virtual copy of the victim IoT network that could allow us to study the attack.
- DSPS GUI is very user friendly and the alert information is complete and easy to understand.
- The fast and powerful response provided for cyber-attacks in IoT and Cloud architectures.

2. **What do you dislike the most about the ANASTACIA framework?**

- The framework could be easier to use if the platform was more integrated.
- The intelligence behind the Incident detector is not clear. The attacks presented were straight forward. What would happen if the attack was a more sophisticated one?

- The architecture might be a little complex to understand and requires expertise to be properly deployed.
- I don't understand why a blockchain is used in the framework. I think a centralized server would be a better idea and provide much more benefits.
- Turning off IoT sensor as a countermeasure.
- It can be perceived as a very large and complicated framework.
- The terminology in the videos is sometimes to complex – not easy to understand for people outside the project.
- Maybe it could be too complex to understand if it is the first time you see the architecture.
- In general it is complex to understand.
- It is difficult to understand how the orchestration of different components is carried out. In addition, the definition of security policies at different levels, as well as debugging information seems a bit complex to understand.
- It is complicated that it may add too much overhead and complexity to the IoT environment, adding possible failure points.
- It is difficult to understand ANASTACIA and to use the whole framework.
- The current implementation level of the DSPS Seal Management.
- The attack vectors used to demonstrate the effectiveness of the framework are too common, for that reason it's hard to validate the robustness of the framework in that regards I suggest investigating on more complex vectors that compromise the different components and give a more detailed assessment.
- ANASTACIA system might be quite complicated for system administrators with no or little experience with network security attacks.
- We have no information about the overhead generated during the detection and mitigation of attacks. Though it may be quite low, thanks to the organization into several planes.
- From security standpoint, it remains unclear what IoT threats the project is capable/aiming to address. For instance, malware threat is a serious concern for IoT and the project should better stress its capability to address such threat. From adoption standpoint, complexity and framework settings may be a killing factor for the project. Interactions among various domain components (monitoring, orchestration, enforcement) remain complex and difficult to understand.
- The complexity that seems to be required to configure all the components to be deployed, and all the rules that should be applied.
- Not so clear why the blockchain is used and what benefits it really brings to the project with respect to other solutions
- The problem of monitoring in real-time IoT end-devices, due to their limited networking and processing capabilities.
- The GUIs can be improved a bit to be more effective.

In summary, end-users identified several critical aspects to be improved in the next cycle of ANASTACIA development and evaluation:

1. More intelligence in Incident Detector.
2. Justification of blockchain in DSPS Seal management.
3. More ambitious use cases with novelty attacks and threats.
4. More explanation of ANASTACIA framework and integrated components.
5. More measurements about the processing overhead of ANASTACIA components.

4 CONCLUSIONS AND FUTURE WORKS

This document reports the validation and end-users evaluation of ANASTACIA framework regarding the four use-cases implemented (BMS.2, BMS.3, BMS.4 and MEC.3). It presents the detailed description of each use case implemented, the benefits of ANASTACIA framework and so as capture data (i.e. web interfaces and log consoles) from ANASTACIA components during the validation of the use cases. Key performance indicators (KPIs) of ANASTACIA components were measured and reported according to the Test-Cases defined in the deliverable D6.2. These KPIs show the fast response time of ANASTACIA components in the modular architecture planes: monitoring, reaction, orchestration and enforcement. The response time is usually less than 3 seconds in the most of components. Even the DSPS Seal management takes less than 3 seconds to notify the users from the DSPS-GUI including the time needed to update the seal/log in the storage based on blockchain.

In additional, the document presents the results of the evaluation performed by 65 end-users to provide their feedback regarding any optimizations and improvements of ANASTACIA framework. To facilitate the evaluation carried out by end-users in multiple workshops, five videos and a wide questionnaire have been provided.

The evaluation results of general questions show that end-users are partially satisfied with all main features of ANASTACIA framework. Although, the easy use of ANASTACIA framework obtained the worst result. Meanwhile, the holistic approach for automatic reaction to attacks was the most appreciated by the end-users. In the 4 use cases, the end-users are partially agreed with the performance and fast response of the ANASTACIA components. On the other hand, the results show a neutral opinion with respect to the blockchain use in the DSPS Seal management and the complexity of Security Orchestrator to manage the mitigation actions.

As future works, the end-users require some improvements such as more intelligence in Incident Detector, more ambitious use cases with novelty attacks and threats, better justification of blockchain in DSPS Seal management, integrated components and more analysis about the processing overhead of ANASTACIA components. Also, they request more information to understand better the ANASTACIA framework and the integrated components in order to be used and deployed in real applications.

Finally, the end-users highlight the novelty approach of ANASTACIA framework for monitoring, reaction, orchestration and enforcement with the combination of SDN and NFV paradigms in order to mitigate automatically attacks and threats in IoT/CPS and Cloud scenarios.

5 ANNEX 1 – INTERVIEW QUESTIONNAIRES

The following table presents all answers of 65 end-users to the abovementioned questionnaire of evaluation. There are some white spaces that represent N/A (no answers) from end-users.

